UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA U NOVOM SADU

Petar D. Bojović

Neprekidnost sesije IP servisa kod heterogenih mobilnih mreža primenom softverski definisanih mreža

DOKTORSKA DISERTACIJA

Mentori:
Prof. dr Ilija Bašičević
Prof. dr Vojin Šenk

Novi Sad, 2018

УНИВЕРЗИТЕТ У НОВОМ САДУ ● **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:		
Идентификациони број, **ИБР**:		
Тип документације, **ТД**:		Монографска документација
Тип записа, **ТЗ**:		Текстуални штампани материјал
Врста рада, **ВР**:		Докторски рад
Аутор, **АУ**:		Петар Бојовић
Ментор, **МН**:		Проф. Др. Илија Башичевић, Проф. Др. Војин Шенк
Наслов рада, **НР**:		Непрекидност сесије IP сервиса код хетерогених мобилних мрежа применом софтверски дефинисаних мрежа
Језик публикације, **ЈП**:		Српски
Језик извода, **ЈИ**:		Српски
Земља публиковања, **ЗП**:		Република Србија
Уже географско подручје, **УГП**:		Војводина
Година, **ГО**:		2018.
Издавач, **ИЗ**:		Ауторски репринт
Место и адреса, **МА**:		Нови Сад; Трг Доситеја Обрадовића 6
Физички опис рада, **ФО**:		6 поглавља / 130 страна / 56 референци / 64 слике
Научна област, **НО**:		Електротехничко и рачунарско инжењерство
Научна дисциплина, **НД**:		Рачунарска техника
Предметна одредница/Кqучне речи, **ПО**:		Бежичне мреже, софтверски дефинисано умрежавање, мобилне мреже
УДК		
Чува се, **ЧУ**:		У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, **ВН**:		
Извод, **ИЗ**:		Ова дисертација се бави истраживањем проблема континуитета IP мрежних сесија у области комуниксција у мобилним рачунарским мрежама. Циљ истраживања у оквиру ове докторске дисертације је да се дефинише решење проблема мобилности примењиво на хетерогене бежичне мреже применом методе софтверски дефинисаног умрежавања. У оквиру истраживања приказана је и практична имплементација предложеног решења. Током истраживања су добијени резултати који указују на потребу интеграције постојећих традиционалних бежичних мрежа са софтверски дефинисаним мрежама. Основу предложеног решења представља инкрементслан приступ у погледу увођења нових SDN функционслности у бежичне IP мреже. Кроз имплементацију минималног сета SDN функционалности гради се тзв. хибридни модел SDN мреже. Главни допринос овог истраживања се огледа у дефинисању поступка који ће омогућити да се превазиђе проблем мобилности у актуелном концепту хетерогених бежичних рачунарских мрежа. Овакав модел решења, пружа значајан допринос и са аспекта улагања у промену инфраструктуре у бежичним мрежама. Имплементацијом хибридног модела, редукује се потреба за потпуним, али и значајним, увођењем виртуелне инфраструктуре базиране на флексибилним софтверски дефинисаним мрежама.
Датум прихватања теме, **ДП**:		
Датум одбране, **ДО**:		
Чланови комисије, **КО**:	Председник:	Проф. др. Никола Теслић, редовни професор
	Члан:	Проф. др. Мило Томашевић, редовни професор
	Члан:	Проф. др. Мирослав Поповић, редовни проф.
	Члан:	Проф. др. Војин Шенк, редовни професор
	Члан, ментор:	Проф. др. Илија Башичевић, ванредни проф.

Потпис ментора

UNIVERSITY OF NOVI SAD ● **FACULTY OF TECHNICAL SCIENCES**
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, **ANO**:	
Identification number, **INO**:	
Document type, **DT**:	Monographic publication
Type of record, **TR**:	Textual printed material
Contents code, **CC**:	PhD Thesis
Author, **AU**:	Petar D. Bojović
Mentor, **MN**:	Ilija Bašičević and Vojin Šenk
Title, **TI**:	IP Session continuity in heterogeneous mobile networks using Software Defined Networking
Language of text, **LT**:	Serbian
Language of abstract, **LA**:	Serbian
Country of publication, **CP**:	Republic of Serbia
Locality of publication, **LP**:	Vojvodina
Publication year, **PY**:	2018.
Publisher, **PB**:	Author's reprint
Publication place, **PP**:	Novi Sad, Dositeja Obradovica sq. 6
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)	6 chapters / 130 pages / 56 references / 64 images
Scientific field, **SF**:	Electrical and Computer Engineering
Scientific discipline, **SD**:	Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, **S/KW**:	Wireless networks, software defined networking, mobile networks
UC	
Holding data, **HD**:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, **N**:	
Abstract, **AB**:	This dissertation investigates the problem of IP networking communication sessions continuity in mobile computer networks. The aim of the research within this doctoral dissertation is to define a solution to the mobility problem applicable to heterogeneous wireless networks using the software-defined networking method. The research also demonstrates the practical implementation of the proposed solution. During the research were obtained results that indicate the need for integration of software-defined networks into existing traditional wireless networks. The basis of the proposed solution is an incremental approach in terms of introducing new SDN functionality into wireless IP networks. Through the implementation of the minimal set of SDN functionality, the so-called hybrid model of the SDN network is being built. The main contribution of this research is reflected in the definition of a process that will allow to overcome the problem of mobility in the current concept of heterogeneous wireless computing networks. This solution model, also provides a significant contribution from the aspect of investing in the change of infrastructure in wireless networks. Implementation of the hybrid model reduces the need for a complete replacement with a virtual infrastructure based on flexible software-defined networks.
Accepted by the Scientific Board on, **ASB**:	
Defended on, **DE**:	

Defended Board, **DB**:		
President:	PhD Nikola Teslić, Full professor	
Member:	PhD Milo Tomašević. Full professor	
Member:	PhD Miroslav Popović, Full professor	Menthor's sign
Member:	PhD Vojin Šenk, Full professor	
Member, Mentor:	PhD Ilija Bašičević, Associate professor	

Sadržaj

Indeks slika

Akronimi

Akronim	*Engleski naziv*	*Srpski naziv*
3G	Third Generation Networks	Mreže treće generacije
5G	Fifth Generation Networks	Mreže pete generacije
5GLTE	Fifth Generation Long-Term Evolution Networks	Peta generacija mreža dugoročne evolucije
AAA	Authentication, Authorization and Accounting	Autentifikacija, autorizacija i knjigovodstvo
ACL	Access Control List	Lista za kontrolu pristupa
ALG	Application Layer Gateway	Posrednik na aplikacionom sloju
AP	Access Point	Pristupna tačka
API	Application Programming Interface	Interfejs za programiranje aplikacija
ARP	Address Resolution Protocol	Protokol za razrešavanje adresa
BGP	Border Gateway Protocol	Protokol graničnog mrežnog prolaza
BOOTP	Boot Protocol	Protokol za podizanje računara
CAPsMAN	Controlled Access Point system Manager	Menadžer za kontrolu pristupnih tačaka
CALEA	The Communications Assistance for Law Enforcement Act	
CCoA	Collocated-Care-of-Address	Privremena adresa pristupa
CCR	Controller for Core Router	Kontroler Core sloja
CDN	Content Delivery Network	Mreža za isporuku sadržaja
CN	Corresponding Node	Čvor korespodent
CoA	Care-of-address	Adresa nadležnog uređaja
CoS	Class of Service	Klasa servisa
CPU	Central Processing Unit	Centralni procesor
CR	Core Router	Ruter Core sloja
CRC	Cyclic Redundancy Check	Ciklična provera redundanse
DHCP	Dynamic Host Configuration Protocol	Protokol za dinamičko podešavanje računara
DiffServ	Differentiated services	Razlikovanje servisa
dNAT	Destination Network Address Translation	Prenošenje od mrežne adrese
DNS	Domain Name Service	Servis domenskih imena

DR	Distribution Router	Ruter diistribucionog sloja
DST	Destination	Odredište
FTP	File Transfer Protocol	Protokol za prenos fajlova
GPRS	General Packet Radio Service	Opšte usluge za paketnu radiokomunikaciju
GRE	Generic Routing Encapsulation	Generička enkapsulacija pri rutiranju
GSM	Global System for Mobile Communications	Opšti sistem za mobilne komunikacije
GUTI	Globally Unique Temporary Identifier	Globalno jedinstveni privremeni identifikator
GW	Gateway	Mrežni prolaz
HD	Host Discovery	
HTTP	HyperText Transfer Protocol	Protokol za prenos hiperteksta
HTTPS	Secured HyperText Transfer Protocol	Zaštićeni protokol za prenos hiperteksta
IANA	Internet Assigned Numbers Authority	Autoritet za dodelu internet brojeva
ICMP	Internet Control Message Protocol	Protokol za upravljanje porukama na internetu
IEEE	Institute of Electrical and Electronic Engineers	Institut inženjera elektrotehnike i elektronike
IETF	Internet Engineering Task Force	Inženjerske snage interneta
IMSI	International Mobile Subscriber Identity	Međunarodni mobilni pretplatnički identifikator
IoT	Internet of Things	Internet stvari
IP	Internet Protocol	Internet protokol
IPsec	Internet Protocol Security	Bezbednost internet protokola
IPv6	Internet Protocol version 6	Internet protokol verzija 6
ISO	International Standardization Organization	Međunarodna organizacija za standardizaciju
ISP	Internet Service Provider	Provajder internet servisa
IXP	Internet eXchange Point	Tačka razmene internet sobraćaja
JDK	Java Development Kit	Java razvojni paket
JSON	JavaScript Object Notation	JavaScript oznaka objekta
L2	Datalink Layer	Link sloja podataka
L3	Network Layer	Mrežni sloj
L5	Application Layer	Aplikacioni sloj

LAN	Local Area Connection	Lokalna mreža
LMA	Local Mobility Anchor	Čvorište lokalne mobilnosti
LTE	Long-Term Evolution Networks	Mreže dugoročne evolucije
MAC	Media Access Control	Upravljanje pristupom medijumima
MAG	Mobile Access Gateway	Čvorište mobilnog pristupa
MD	Mobility Discovery	
MIP	Mobile Internet Protocol	Mobilni internet protokol
MN	Mobility Node	Mobilni čvor
MS	Mobility Service	
MST	Mobility Service Table	
MXP	Mobility eXhange Point	Tačka razmene mobilnih korisnika
NAPT	Network Address Port Translation	Translacija mrežnih adresa pomoću portova
NAT	Network Address Translation	Translacija mrežnih adresa
NCP	Network Control Protocol	Protokol za upravljanje mrežom
OF	OpenFlow	
ONF	Open Networking Fundation	Fondacija za slobodno umrežavanje
OS	Operating System	Operativni sistem
OVS	OpenVSwitch	
OVSDB	OpenVSwitch DataBase	OpenVSwitch baza podataka
PAT	Port Address Translation	Prenošenje mrežnih adresa putem portova
PBA	Proxy Binding Ack	Potvrda mape posrednika
PBU	Proxy Binding Update	Ažuriranje mapa posrednika
PMIPv6	Proxy Mobile IPv6	Internet protokol verzije 6 za posredničku mobilnost
PPP	Point-to-Point Protocol	Protokol od tačke do tačke
pvIP	permanent virtual Internet Protocol Address	Stalna virtualna internet protokol adresa
PXE	Preboot eXecution Environment	Okruženje za preliminarno pokretanje sistema
QoE	Quality of Experience	Kvalitet doživljaja
QoS	Quality of Service	Kvalitet servisa
RA	Router Advertisement	Oglašavanje rutera

RAF	The School of Computing	Računarski fakultet
REST	REpresentational State Transfer	Prenos reprezentativnih stanja
RFC	Request for Comment	Zahtev za komentare
rIP	real Internet Protocol Address	Realna internet protokol adresa
RIS	Remote Instalation Service	Servis za daljinsku instalaciju
RS	Router Solicitation	Pretraga rutera
RSA	Rivest,Shamir,Adlema algorithym	RSA algoritam
RTT	Round-Trip Time	
SDN	Software Defined Networking	Softverski definisane mreže
SMB	Server Message Block	Blok poruke servera
SMTP	Simple Mail Transfer Protocol	Protokol za prenos elektronske pošte
sNAT	Source Network Address Translation	Translacija izvorišne mrežne adrese
SOX	Serbian Open eXchange	
SPF	Shortest Path First	Algoritam najkraće putanje
SRC	Source	Izvorište
SSH	Secure SHell	
SW	Switch	Svič
SYN	Synchronization flag	Oznaka sinhronizacije
TCP	Transmission Control Protocol	Protokol za upravljanje prenosom
TFTP	Trivial FTP	
TLS	Transport Layer Security	Bezbednost transportnog sloja
ToS	Type of Service	Tip servisa
TS	Tap Service	Tap servis
TTL	Time To Leave	Vreme zadržavanja zapisa
UDP	User Datagram Protocol	Protokol za korisničke datagrame
UEFI	Unified Extensible Firmware Interface	Unificirani nadogradiv interfejs za firmver
UML	Unified Modeling Language	Objedinjeni jezik za modelovanje
UMTS	Universal Mobile Telecommunications System	Univerzalni sistem mobilnih telekomunikacija
USB	Universal Serial Bus	Univerzalna serijska magistrala
VLAN	Virtual Local Area Connection	Virtualna lokalna mreža

VM	Virtual Machines	Virtualne mašine
VoIP	Voice over Internet Protocol	Prenos glasa kroz internet protokol
VPN	Virtual Private Network	Virtualne privatne mreže
WAN	Wide Area Network	Mreže širokog područja
WiFi	Wireless Fidelity	Bežična mreža
WiMax	Worldwide Interoperability for Microwave Access	Svetska interoperabilnost za mikrotalasne pristupne mreže
WLAN	Wireless Local Area Connection	Bežične lokalne mreže
WNV	Wireless Network Virtualisation	Bežična virtualizovana mreža

Uvod

Poslednje dve decenije svedoci smo ogromnog napretka u oblasti računarstva zasnovanog na bežičnim, i to, pre svega, mobilnim komunikacijama. Prava eksplozija u implementaciji novih tehnologija i rešenja u ovoj oblasti dovela je do realizacije različitih (heterogenih) računarskih okruženja. Suštinska karakteristika novonastalih okruženja ogleda se u mogućnosti da mobilnim korisnicima pruže nove, naprednije servise i usluge. Međutim, brz razvoj mobilnog računarstva doneo je određene probleme. Ti problemi se, pre svega, odnose na činjenicu da u heterogenim bežičnim računarskim okruženjima postoji evidentan nedostatak interoperabilnosti. Osnovna karakterisika svakog od njih je različit nivo kvaliteta servisa (engl. *Quality of service* – QoS). Razlog za postojanje heterogenosti u pogledu ponuđenog nivoa kvaliteta servisa leži u nepostojanju jedinstvenog skupa standarda u korišćenju različitih mrežnih pristupa. To je faktor koji presudno utiče na problem obezbeđivanja potpune mobilnosti korisnika koji prolaze kroz heterogene bežične mreže.

Prelazak iz jedne bežične mreže u drugu može dovesti do značajnih promena sa aspekta korisnika, kao što su raspoloživ propusni opseg ili promena nekog od drugih parametara koji bitno utiču na kvalitet servisa. Zbog toga se mora kontinuirano raditi na razvoju i primeni različitih modela adaptivnog računarstva. Implementacija ovih modela treba da omogući potpunu mobilnost korisnika i obezbedi potreban nivo kvaliteta servisa pri prolasku korisnika preko više, često različitih, bežičnih mreža. Glavni cilj implementacije različitih modela adaptivnog računarstva mora da bude u vezi sa obezbeđivanjem potpune mobilnost korisnika, zbog čega je neophodno da postoji kontinuitet u odvijanju IP sesije pri prelasku kroz različite mreže.

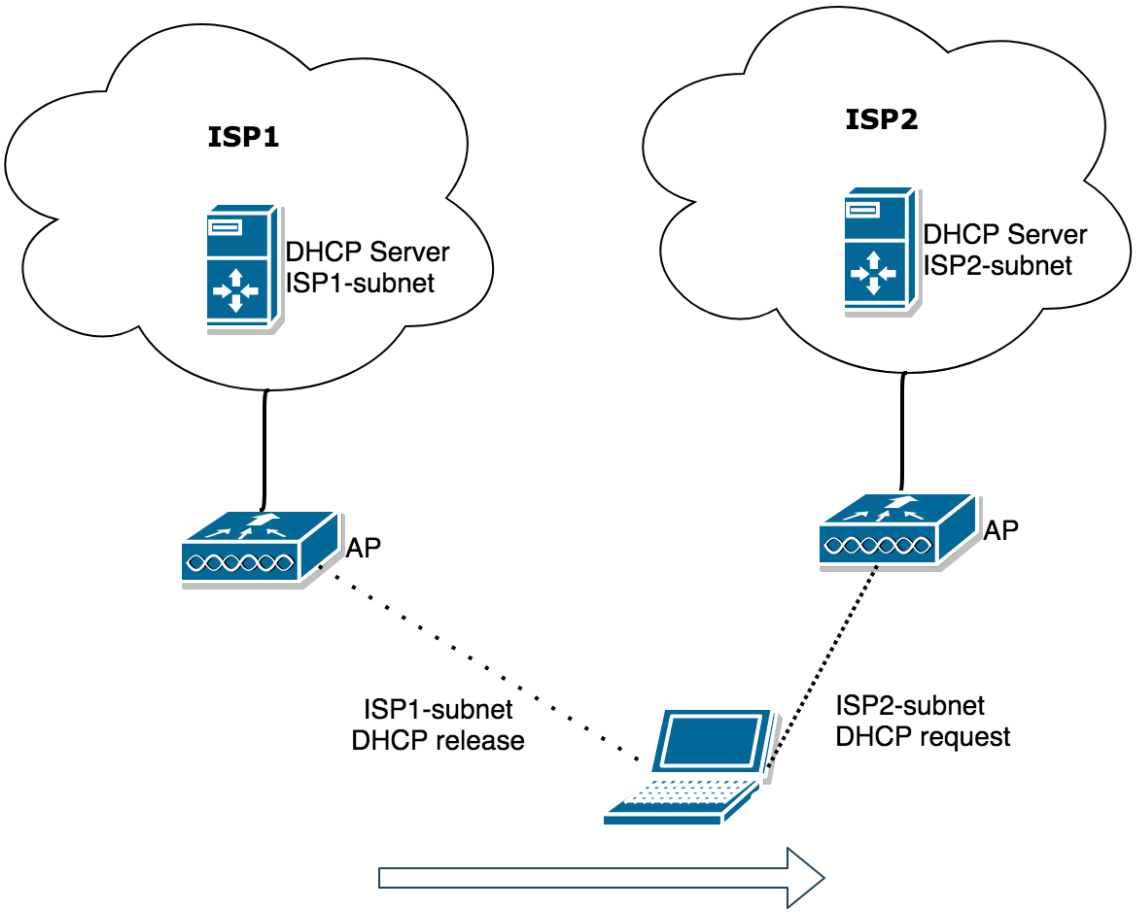
Da bi se kontinuitet u odvijanju IP sesije omogućio, potrebno je da se korisnicima pri prelasku kroz heterogene bežične mreže obezbedi neprekidna dostupnost servisa. To u praksi znači da u komunikaciji moraju da se koriste isključivo standardizovani protokoli, definisani na odgovarajućim slojevima TCP/IP referentnog komunikacionog modela [1]. Posmatrano sa stanovišta L3 (internet) sloja TCP/IP referentnog modela, internet protokol (engl. *Internet Protocol* – IP) predstavlja osnovu svake mrežne komunikacije koja je u vezi sa adresiranjem računara i stvaranjem potrebnih uslova za određivanje optimalne putanje paketa od izvora do željene destinacije.

Početna uloga IP protokola [2] bila je u vezi sa idejom, odnosno zahtevom, da svi računari i drugi uređaji unutar globalne internet mreže poseduju jedinstvenu IP adresu kako bi paketi, upućeni sa bilo kog računara/uređaja, uvek stigli na „pravo mesto". Ovakav pristup, zasnivao se na analogiji – kao što ljudi menjaju adresu stanovanja, ali uvek zadržavaju jedinstveni matični broj, i korisnicima unutar internet mreže treba omogućiti da promene fizičku lokaciju, a na svom uređaju zadrže IP adresu. Međutim, brz rast broja korisnika odnosno računara/uređaja na globalnoj internet mreži, jasno je ukazao na činjenicu da ovakav pristup nije praktičan i da značajno opterećuje funkcionisanje protokola za rutiranje saobraćaja, posebno onda kada se radi o mobilnim korisnicima. Uzrok je u tome što su protokoli za rutiranje saobraćaja primenjeni samo nad mrežnim grupama, a ne nad pojedinačnim računarima. Dakle, jasno je da se koncept jedinstvene IP adrese nekog računara nije mogao primeniti kod mobilnih korisnika.

U početnim fazama razvoja interneta, problematika dodele IP adresa mobilnim korisnicima nije bila u fokusu. Ovom pitanju nije se poklanjalo previše pažnje jer su u tom momentu tehnologije za mobilni pristup bile prilično nerazvijene. Međutim, poslednjih decenija 20. veka, brzim razvojem privrede i društva u celini, došlo je do bitnih promena u oblasti komunikacija koje su u suštini uslovljene strategijom razvoja naše civilizacije zasnovane na:

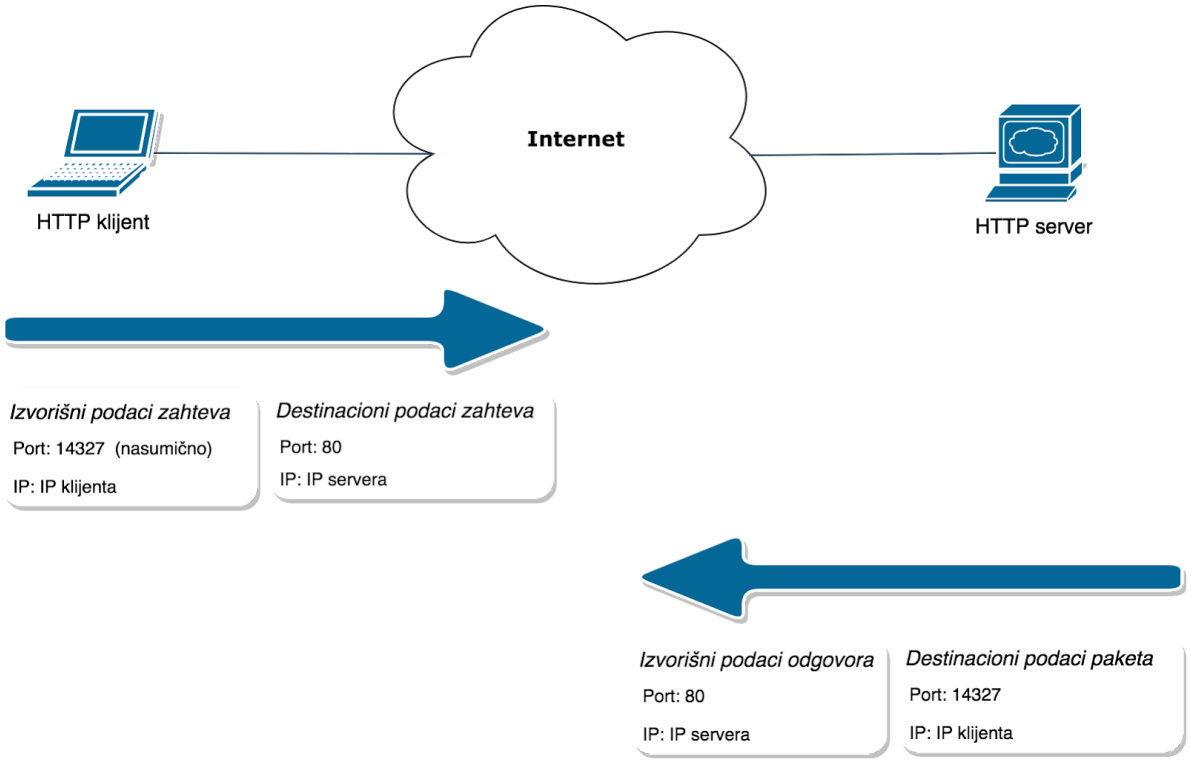
- globalizaciji – zahtevu i potrebi da se komunikacione veze uspostave između bilo koje dve tačke na Zemlji, u stanju mirovanja ili u pokretu, i
- integraciji – obezbeđenju mogućnosti prenosa različitih servisa i integraciji različitih komunikacionih sistema.

U skladu s tim, sve se veći značaj pridaje rešavanju problema mobilnosti koji se u mrežama povećava zajedno s brojem mobilnih korisnika i aplikacija za mobilne uređaje. I pored činjenice da danas postoje određena rešenja, ne može se još uvek reći da postoji široko prihvaćeno rešenje za mobilnost na internetu. Problem mobilnosti je teško rešiti sa aspekta heterogenih mreža, jer se on proširuje na druge aspekte kao što su vlasništvo, administracija, tehnologije i dr. Osnovni problem mobilnog korisnika ogleda se u tome što prilikom promene mreže mora da promeni vrednosti svojih parametara, odnosno da se adaptira na parametre mreže u koju dolazi. To konkretno znači da mobilni korisnici pri „dolasku" u drugu mrežu moraju npr. da promene IP adresu na svom računaru/uređaju.



Slika 1.1. Promena IP adrese usled prelaska u drugu mrežu

Navedeni problem koji se odnosi na promenu IP adrese prilikom prelaska u drugu mrežu je od suštinskog značaja ako se ima u vidu dvosmernost svake komunikacije na relaciji mobilni klijent – server. Naime, komunikacija se odvija tako što mobilni klijent sa određene adrese u toj mreži (IP, MAC i sl.) šalje paket na adresu servera. Na klijentov zahtev, server odgovara na adresu mobilnog klijenta sa koje je zahtev poslat.



Slika 1.2. Primer komunikacije između HTTP klijenta i servera

Pitanje promene IP adrese pri prelasku u drugu mrežu od velikog je značaja i sa aspekta transporta paketa, jer se većina servisa koji se danas koriste na internetu oslanja na uspostavljanje neke forme komunikacije između klijenta i servera. TCP [3] predviđa uspostavljanje i održavanje konekcije između klijenta i servera da bi se obezbedila pouzdanost prenosa podataka i kontrolisao njihov tok između dve strane u komunikaciji. UDP [4] sa druge strane ne predviđa uspostavljanje konekcije, ali mnogi servisi koriste izvornu IP adresu kako bi identifikovali korisnika. Primeri su VoIP komunikacija, mrežne igre i sl. Dakle, komunikacija između klijenta i servera uslovljena je time da oba entiteta (klijent i server) razgovaraju uvek preko iste IP adrese.

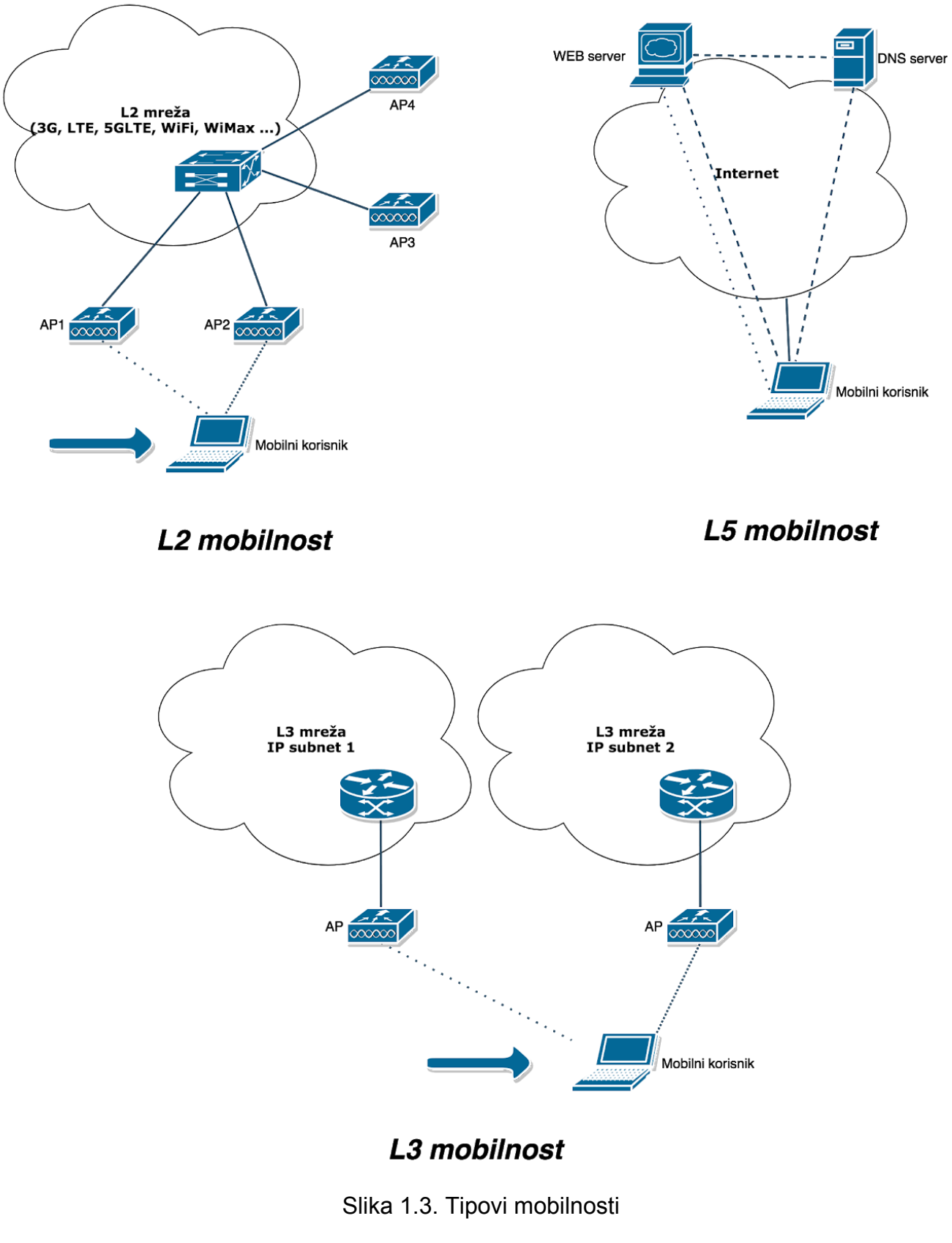
Kada mobilni korisnik – klijent pređe u drugu mrežu, on menja IP adresu (slika 1.1). Sa promenom IP adrese dolazi do prekida svih konekcija koje je uspostavio sa udaljenim serverima. Razlog za prekid uspostavljenih konekcija leži u tome što mobilni klijent, nakon prelaska u novu mrežu, sve pakete udaljenim serverima šalje sa nove IP adrese. Pakete koji mu dolaze sa nove adrese server tretira kao deo nove konekcije i ne dozvoljava klijentu da nastavi rad, kao da do promene nije došlo. Dakle, sa stanovišta tradicionalnih IP mreža, ovaj problem se rešava ponovnim uspostavljanjem konekcije/relacije između mobilnog klijenta i servera, što znači da klijent ponovo inicira uspostavljanje TCP konekcije ili ponovo pokreće komunikaciju po UDP protokolu, poništavajući sva prethodna stanja.

Posledica ovakvog rešenja u oblasti mobilnosti korisnika je da isti privremeno gubi pristup servisu koji je koristio pre prelaska u drugu mrežu. Kao primeri gubitka servisa koji nastaje usled prelaska u drugu mrežu mogu se navesti:

- Preuzimanje fajlova putem HTTP ili FTP konekcije korišćenjem TCP protokola na transportnom nivou praćeno je prekidom servisa, odnosno prekidom prenosa podataka pri prelasku mobilnog korisnika u drugu mrežu (protokoli kao što su HTTP i FTP poseduju rešenja koja omogućavaju nastavak preuzimanja određenog fajla pod određenim uslovima).
- Prekid VoIP komunikacije, koja je uglavnom zasnovana na korišćenju UDP protokola, nakon prelaska u novu mrežu (dolazi do potpunog gubitka paketa i prekida veze). Potrebno je napomenuti da je zavisno od implementacije VoIP servisa nekad potrebno i nekoliko minuta da se VoIP konekcija prema serveru ponovo inicira sa nove adrese klijenta (razlog je u tome što neki VoIP protokoli periodično vrše registraciju, tj. povezivanje korisnika sa IP adresom).
- Prekid TCP konekcije i obustava slanja nekog većeg imejla u trenutku prelaska u drugu mrežu (SMTP protokol koji se koristi za slanje imejlova nema podršku za nastavak slanja, pa je potrebno ponovo uspostaviti TCP konekciju i opet poslati ceo sadržaj imejla).
- Kod internet *streaminga* (uglavnom se koristi UDP na transportnom nivou), nakon prelaska u novu mrežu, prestaće da dolaze paketi posmatranog *streama* jer se oni još neko vreme isporučuju na staru adresu. Potrebno je ponovo poslati UDP zahtev za isporuku *streama* kako bi primerak paketa stigao i na novu adresu.

Praktično se svaki protokol, a time i svaki servis, iznad mrežnog sloja oslanja na perzistentnost klijentske i serverske IP adrese, pa promenom IP adrese bilo kog učesnika u komunikaciji dolazi do narušavanja toka podataka na višim slojevima TCP/IP modela i prekida komunikacije. Zbog toga se, razvojem interneta i novih pristupnih tehnologija (WiFi, WiMax, GPRS, 3G, LTE, 5GLTE), pojavila i potreba da se pronađe rešenje za pomenuti problem. Predložena su različita rešenja sa ciljem da se održi neprekidnost komunikacije (sesije). Ova rešenja su zavisno od mesta, odnosno sloja implementacije u TCP/IP komunikacionom modelu, klasifikovana [5] na sledeći način:

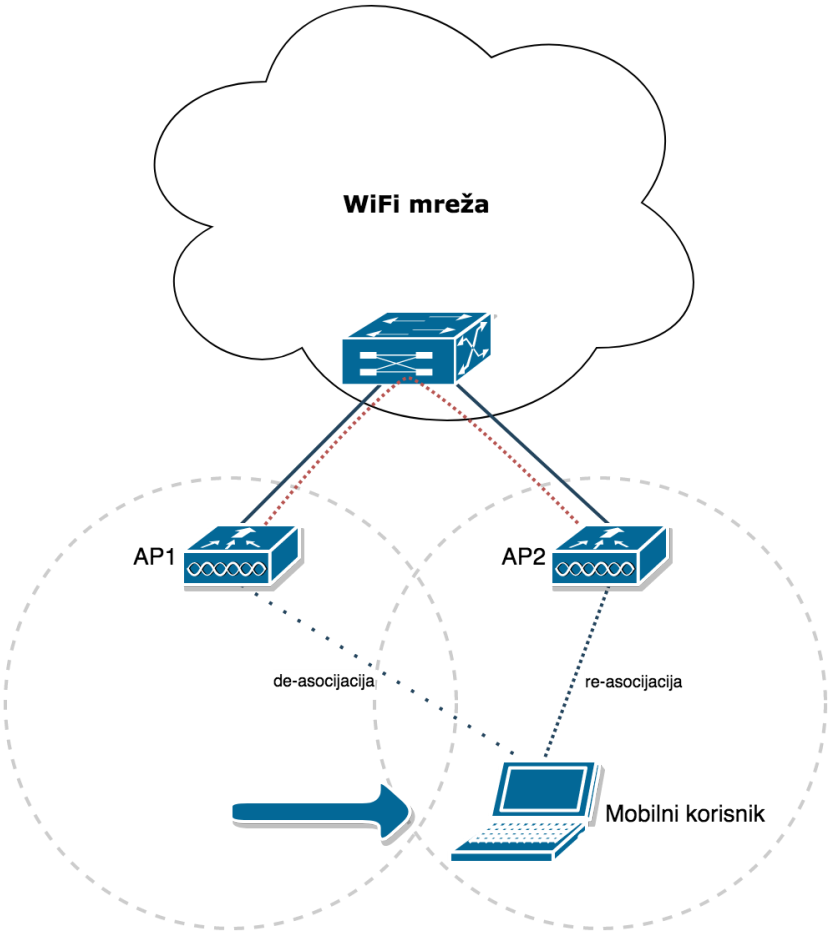
- L2 mobilnost – rešenja za mobilnost unutar tehnologije pristupnog nivoa,
- L3 mobilnost – rešenja za mobilnost između različitih mrežnih celina, i
- L5 mobilnost – rešenja za mobilnost na aplikativnom nivou



Slika 1.3. Tipovi mobilnosti

L2 mobilnost

Mobilnost unutar tehnologija pristupnog sloja, tzv. „L2 mobilnost", podrazumeva implementaciju rešenja koja treba da omoguće prelazak korisnika preko različitih zona pokrivenosti u mreži nekog provajdera u kojoj se koristi ista tehnologija. Ova procedura je u mrežama mobilne telefonije sa ćelijskim pokrivanjem poznata pod terminom *handoff*. Mora se naglasiti da su popularne pristupne tehnologije za bežični pristup kao što su WiFi, WiMax, GSM, 3G, LTE, 5GLTE već primenile određena rešenja koja se odnose na problem prelaska iz jedne u drugu zonu pokrivenosti. Njihova primena omogućava prelazak korisnika iz jedne u drugu zonu, a da tom prilikom ne dolazi do velikih gubitaka u prenosu paketa. Imajući u vidu da je ova mobilnost rešena na L2 sloju (*Network Access* sloj TCP/IP modela), jasno je da ne postoji potreba za promenom IP adrese i time se otklanja opasnost od prekida servisa višeg sloja. Mora se, međutim, još jednom naglasiti da su ovakva rešenja izvodljiva samo unutar mreže istog provajdera i iste tehnologije.

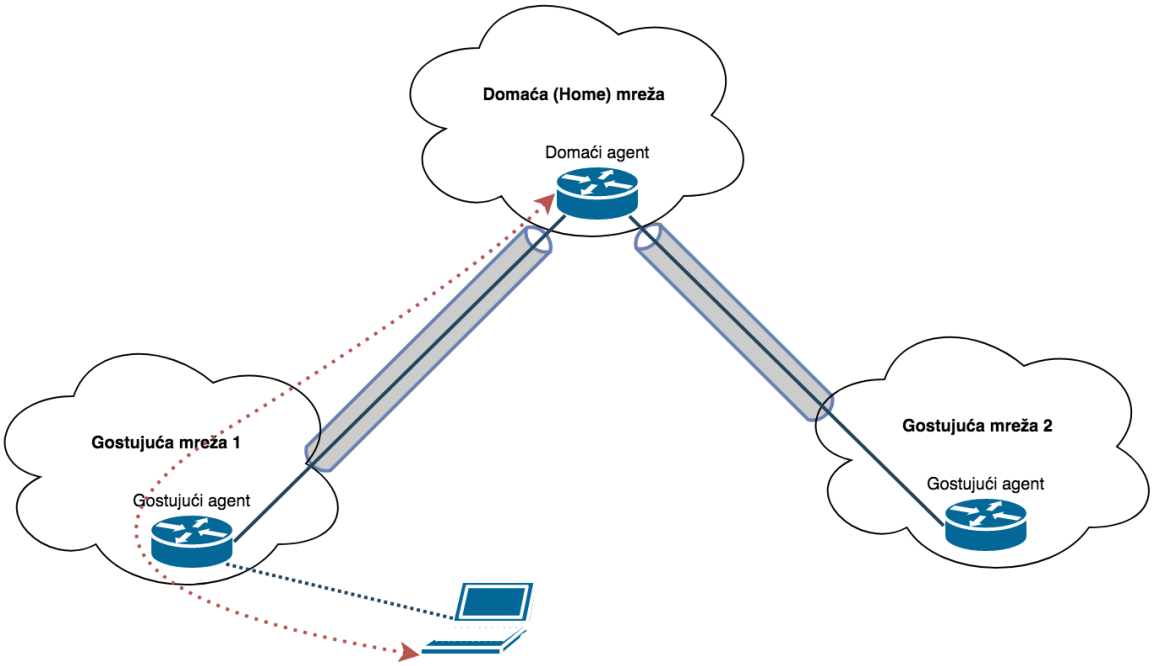


Slika 1.4. Primer L2 mobilnosti kod WiFi mreže

L3 mobilnost

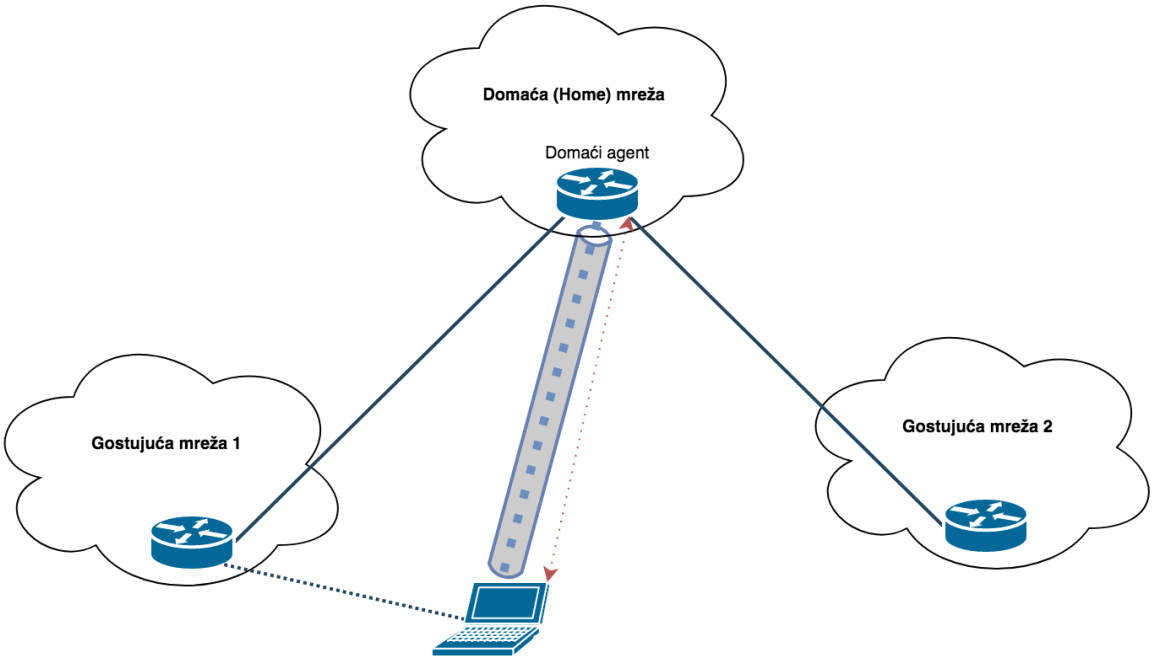
Mobilnost na mrežnom sloju ili L3 mobilnost predstavlja mogućnost obezbeđivanja neprekidnosti jedne konekcije (toka paketa) prilikom prelaska korisnika iz jedne u drugu mrežu. U suštini, ove mreže mogu da budu deo istog administrativnog domena ili deo različitih domena (deo mreže istog ili različitog provajdera servisa) i na L2 sloju mogu da budu izgrađene u istoj ili u drugačijoj pristupnoj tehnologiji. U uvodnom delu je opisana problematika zbog koje dolazi do prekida servisa kada mobilni korisnik promeni mrežu pristupa. Iz navedenog se može lako zaključiti da iznalaženje rešenje za problem mobilnosti na L3 sloju nije nimalo jednostavno i lako za implementaciju, posebno imajući u vidu već široko rasprostranjene standarde. Važno je naglasiti da, i pored toga što je sam standard za IP [2] definisao način rešavanja IP mobilnosti i što postoje drugi standardi kao što je RFC3344 [6]), u praksi još nije zaživela neka tehnika, odnosno rešenje mobilnosti na L3 sloju. Problem mobilnosti na L3 sloju je ponovo intenziviran zbog potrebe prelaska na veći adresni prostor putem IPv6 protokola [7]. Međutim, šira primena IPv6 protokola (čija je važnost nesporna u stručnim i u naučnim krugovima) i dalje je spora, čak i pomalo neizvesna u pogledu potpune zamene aktuelnog IPv4 protokola. Sve su ovo razlozi zbog kojih postoji veliki broj rešenja kojima se predlažu specifični pristupi ovom problemu.

Jedan od pristupa podrazumeva postojanje domaće mreže (engl. *Home Network*) i agenta (engl. *Home Agent*). Prilikom prelaska u drugu mrežu, korisnik se kod svog agenta registruje putem nove, privremene, gostujuće IP adrese. Kad god neko šalje paket ka uvek istoj IP adresi klijenta iz domaće mreže, agent taj paket prosleđuje do privremene gostujuće adrese. Takođe važi i obrnuto, kada klijent šalje paket do udaljenog servera, to čini preko agenta. Sličan pristup se koristi i u ćelijskoj (mobilna telefonija) tehnologiji gde je ova procedura poznatija pod terminom *roaming*.



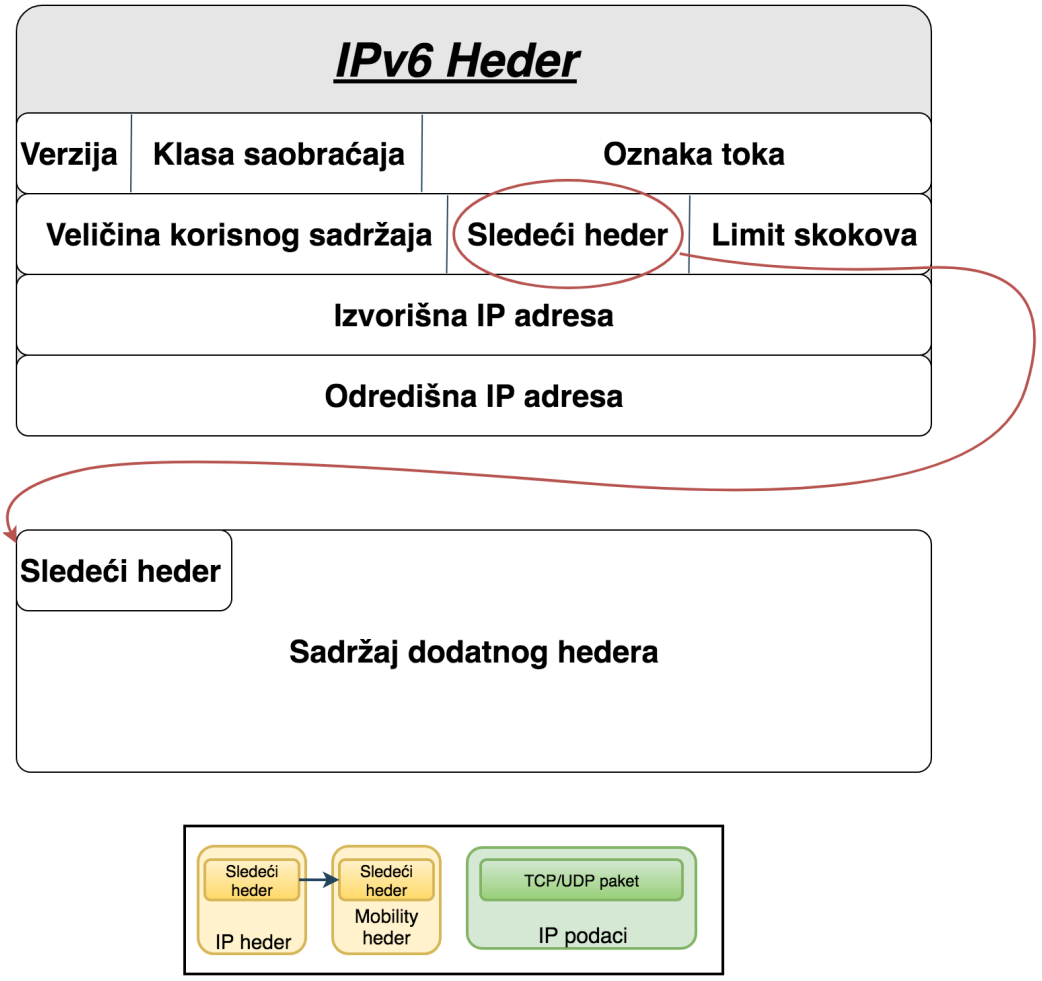
Slika 1.5. Mobilni IP sa domaćom i gostujućim mrežama

Drugi pristup podrazumeva realizaciju neke vrste virtualnog tunela (eng. *Virtual Private Network* – VPN) između računara, gde god da se nalazi (bilo u domaćoj ili gostujućoj mreži), i domaće mreže, tako da je klijent uvek virtualno prisutan u domaćoj mreži.



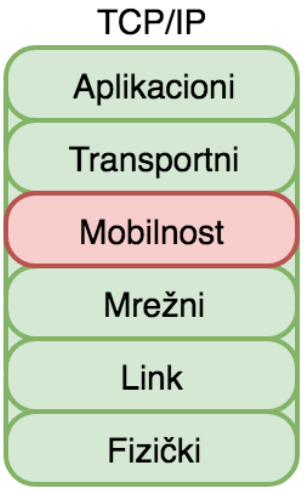
Slika 1.6. Mobilni IP formiranjem VPN tunela između mobilnog korisnika i domaće mreže

Može se takođe navesti i pristup čiji je cilj iskorišćavanje mogućnosti nadogradnje postojećih ili budućih protokola (kao što je IPv6 protokol) kako bi se efikasnije rešio problem mobilnosti.



Slika 1.7. IPv6 heder i ekstenzija za IP mobilnost

Neka od mogućih rešenja predlažu i modifikaciju trenutno aktuelnog TCP/IP komunikacionog modela „ubacivanjem" novog sloja između mrežnog i transportnog sloja, čiji bi primarni zadatak bio da se bavi problemom mobilnosti.



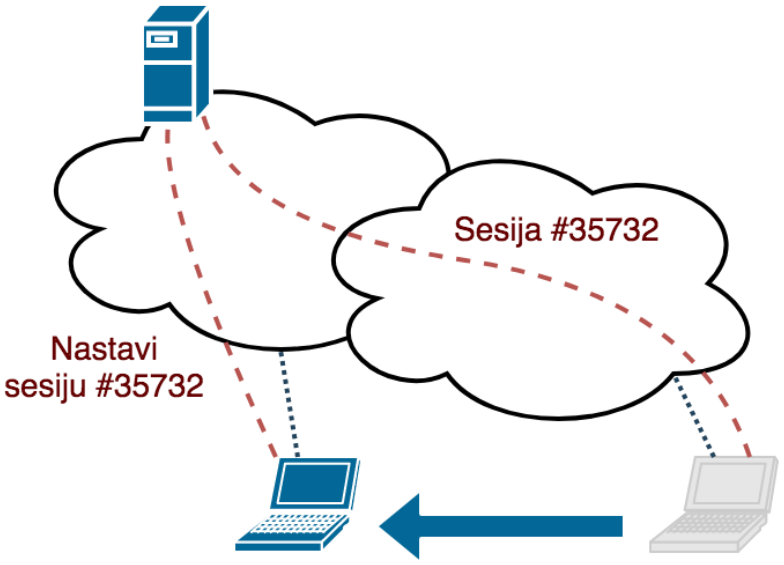
Slika 1.8. Sloj mobilnosti kao poseban sloj TCP/IP modela

U narednom poglavlju „Srodna istraživanja" obrazložićemo konkretna rešenja za probleme mobilnosti.

L5 mobilnost

Problem mobilnosti može se rešavati i na aplikacionom sloju (L5 mobilnost) implementacijom protokola koji bi imao zadatak da minimalizuje efekte gubitaka i što pre uspostavi prenos kao da do prelaza iz jedne u drugu mrežu nije ni došlo (podrazumeva mogućnost prevazilaženja problema koji nastaju usled mobilnosti na nižim nivoima putem aplikacionog protokola). Ovakav pristup zahteva od dizajnera aplikativnog softvera i protokola za komunikaciju između aplikacija da predvidi mogućnost nastanka problema mobilnosti, tj. kratkog prekida komunikacije sa njenim ponovnim uspostavljanjem preko druge adrese. UDP protokoli se ne oslanjaju na konekcije pa se može efikasnije napraviti otpornost na promenu izvorišne adrese. TCP konekcije, zbog svoje procedure u uspostavljanju veze, sporije ponovo uspostavljaju vezu, ali je ipak putem aplikacionog protokola moguće predvideti da nastave započetu sesiju (primer je upravo mogućnost nastavka preuzimanja fajla preko HTTP-a).

Efikasnost rešenja koje bi omogućilo neprekidnost komunikacije na aplikacionom sloju usko je povezana sa brzinom prelaska između mreža, tj. sa tehnologijama nižeg nivoa. Zbog toga, i pored primene mehanizama L5 mobilnosti, neprekidnost zavisi od tehnologija nižih slojeva.



Slika 1.9. L5 mobilnost putem podrške aplikacionog protokola da nastavi postojeću sesiju

Srodna istraživanja

Standardi

Jedan od najznačajnijih pokušaja rešavanja problema mobilnosti u mrežama sa internet protokolom je onaj koji se odnosi na izradu Mobile IP (MIP) standarda [8]. Prvi pokušaj u pravcu izrade ovog standarda napravljen je 1996. godine, kada je donet IETF dokument RFC 2002 [9]. Ovaj dokument predstavljao je osnovnu verziju MIP standarda, a razvojni trendovi u vezi sa obezbeđivanjem potpune mobilnosti u IP mrežama inicirali su potrebu da se još intenzivnije nastavi rad na njegovom daljem razvoju. To je 2002. godine rezultirao novim RFC 3344 dokumentom pod kojim je MIP i danas poznat.

Osnovni elementi bežične IP mreže u kojoj je implementiran MIP standard su:

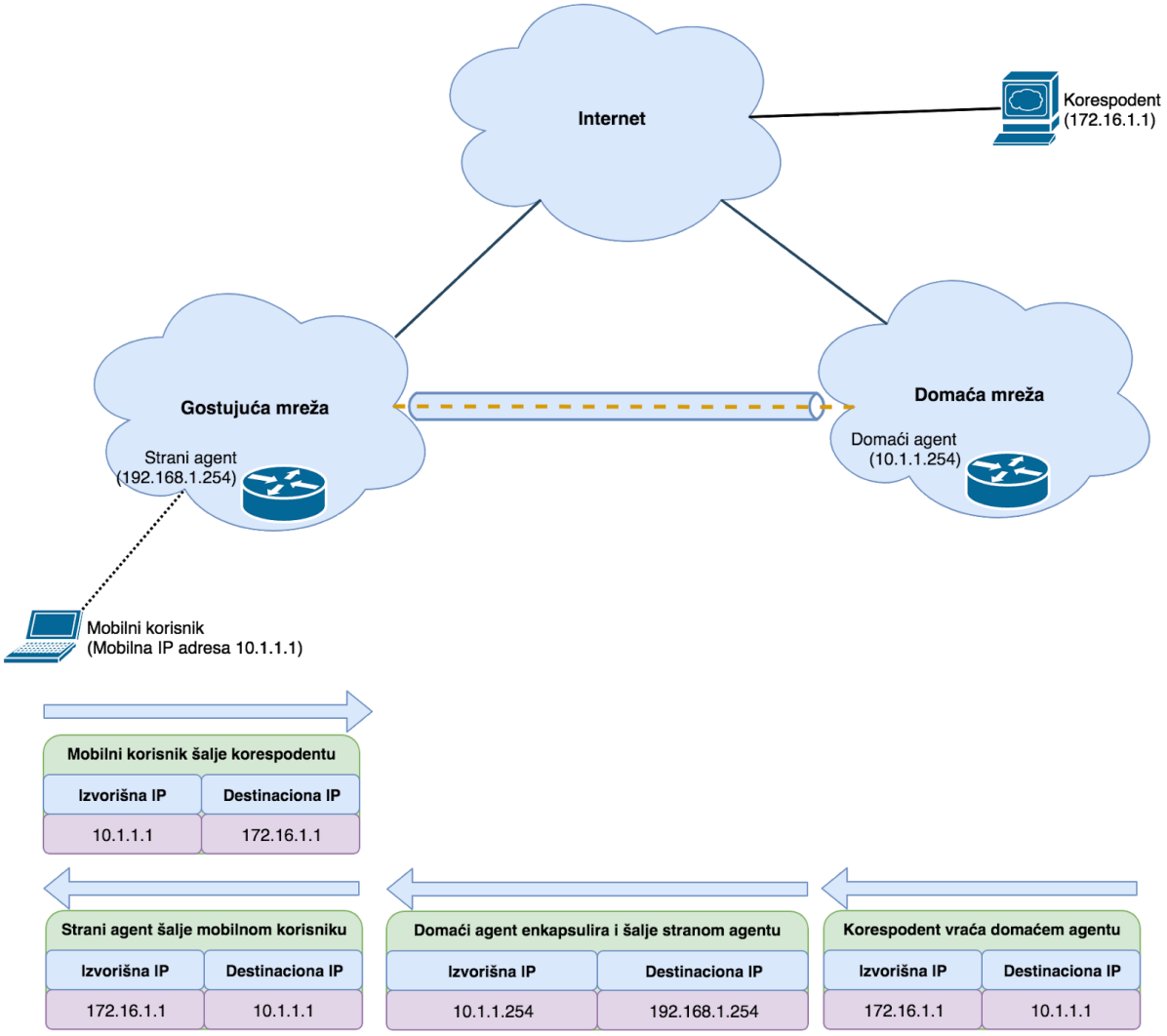
- domaća mreža (engl. *Home Network*) – poseduje mrežni prefiks koji odgovara mrežnoj adresi mobilnog korisnika. Saobraćaj će se od mobilnog korisnika i za njega rutirati kao i svaki drugi običan saobraćaj kada je korisnik u domaćoj mreži;
- virtuelna mreža na domaćem agentu (engl. *Home Agent*), koja uglavnom ne postoji u fizičkoj formi. Ruter koji opslužuje tu virtuelnu mrežu oglašava njeno postojanje drugim mrežama preko dinamičkog protokola za rutiranje saobraćaja; i
- strana mreža (engl. *Foreign Network*), koja je svaka mreža izvan domaće mreže mobilnog korisnika. Mobilni korisnik može transparentno da pređe u drugu mrežu, a da računar sa kojim komunicira to i ne primeti.

Primena MIP standarda u ovim mrežama zasniva se na implementaciji softverskih entiteta koji se u domaćoj i stranoj mreži nazivaju agentima. Njihov primarni zadatak je da omoguće evidenciju mobilnih korisnika i obezbede efikasno usmeravanje njihovih paketa. U tom smislu, MIP standardom definisane su sledeće operacije:

- *Mobile IP Agent Discovery* – procedura za pronalaženje agenata u trenutnoj mreži koristeći ICMP RA (engl. *Router Advertisement*) poruke.
- *Mobile IP Registracija* i AAA (autentifikacija, autorizacija i *acounting*) – procedura za evidenciju mobilnih korisnika i povezivanje njihovog CoA (engl. *Care-of-address*, obično adresa stranog agenta), kako bi se postavio podatak adrese preko koje je korisnik dostupan u gostujućoj mreži.

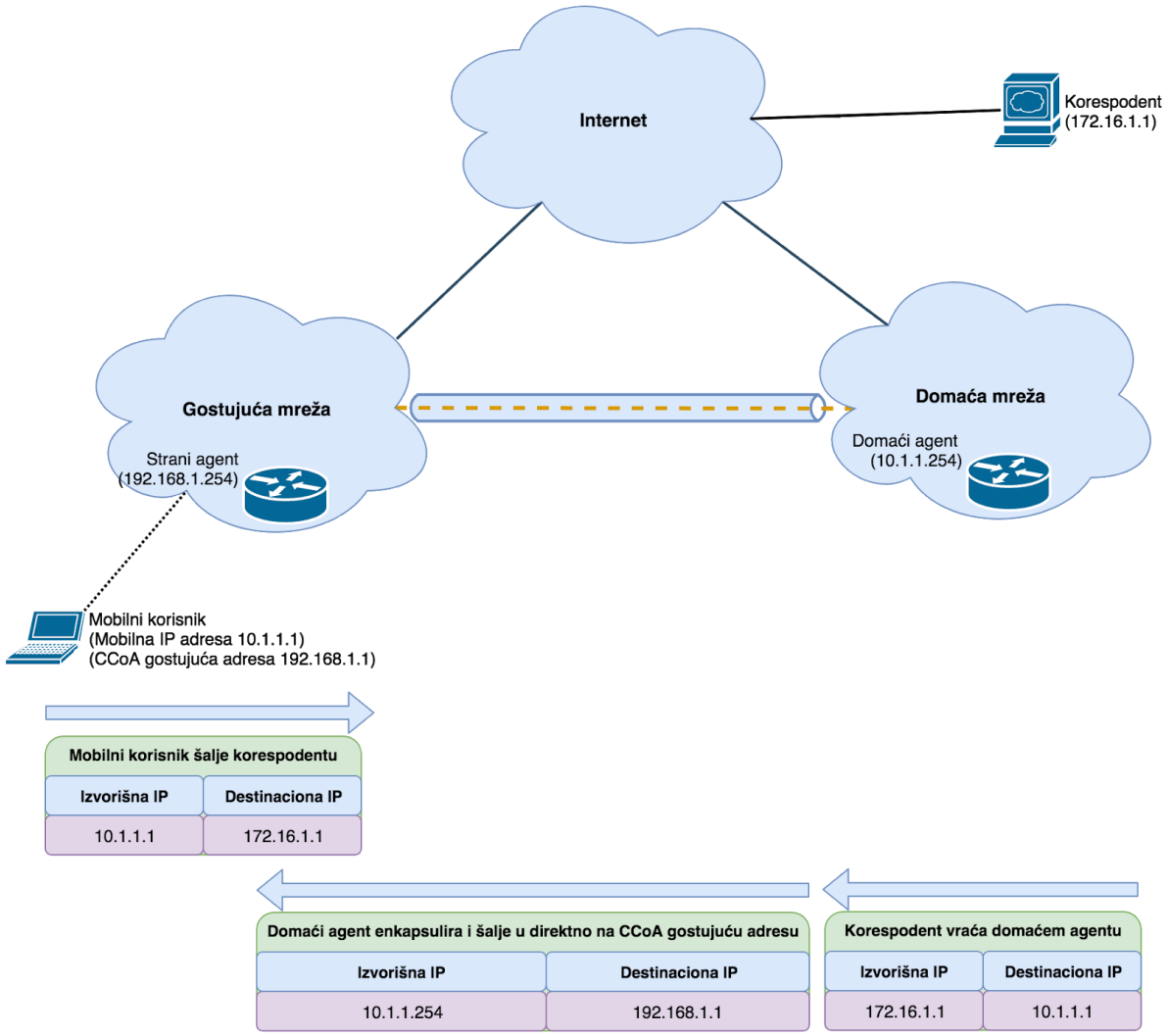
- *Mobile IP Tunnel*, *Binding* i *Datagram Forwarding* – procedure koje definišu način usmeravanja paketa do mobilnog korisnika kada je on u domaćoj ili stranoj mreži.

Funkcionisanje MIP standarda zasniva se na konceptu koji podrazumeva da se mobilnom korisniku nakon uspešne registracije kod domaćeg agenta, omogući da pakete šalje svom korespodentu. Slanje paketa se odvija direktno, korišćenjem *home* adrese mobilnog korisnika nezavisno od njegove lokacije (bilo da se nalazi u domaćoj ili stranoj mreži) i primenom standardnih protokola za rutiranje saobraćaja. Zadatak protokola za rutiranje saobraćaja je da obezbede da svaki paket mobilnog korisnika bude prosleđen do korespodenta, bez obzira na polaznu tačku. U primeni ovog koncepta mogu nastati određeni problemi, i to u slučajevima kada korespodent vraća paket na *home* adresu mobilnog korisnika. Ukoliko je mobilni korisnik u domaćoj mreži, standardni protokoli za rutiranje saobraćaja omogućiće da paket bude prosleđen do domaćeg agenta koji će ga uputiti lokalnom mobilnom korisniku. Međutim, ako je mobilni korisnik u stranoj mreži, proces prosleđivanja je komplikovaniji. Naime, registracijom domaći agent poseduje podatak o CoA korisnika u stranoj mreži. Formira se GRE tunel između domaćeg agenta i CoA (obično gostujućeg, stranog agenta). Paket namenjen mobilnom korisniku u stranoj mreži prenosi se kroz formiran tunel do stranog agenta. CoA deenkapsulira paket koji je primljen od domaćeg agenta i prosleđuje ga do mobilnog korisnika u gostujućoj mreži.



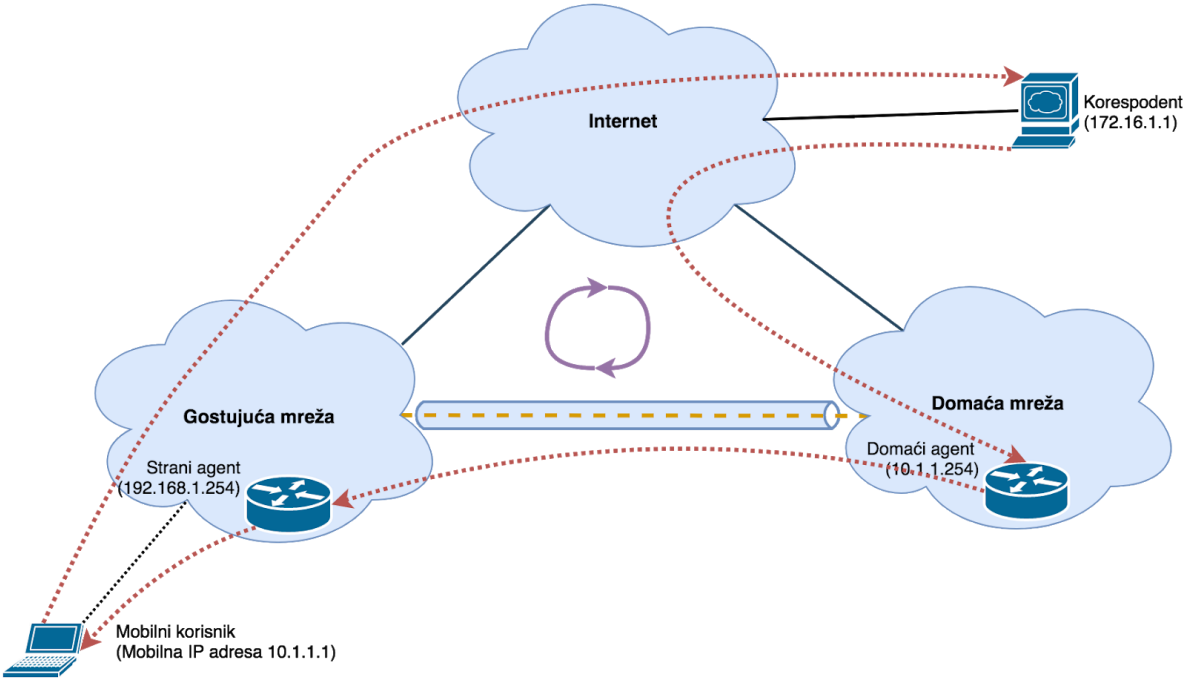
Slika 2.1. MobileIP standard

Na slici 2.1. prikazano je kretanje paketa i dodela IP adresa u Mobile IP rešenju. Interesantno je navesti i činjenicu da rešenje zasnovano na Mobile IP standardu predviđa mogućnost da i sam mobilni korisnik bude CoA kada se njegova adresa u gostujućoj mreži koristi kao CCoA parametar prilikom registracije kod domaćeg agenta. Na taj se način preskače još jedan korak u prosleđivanju paketa sa gostujućeg agenta na mobilnog korisnika.



Slika 2.2. MobileIP standard direktno slanje na CCoA gostujuću IP adresu

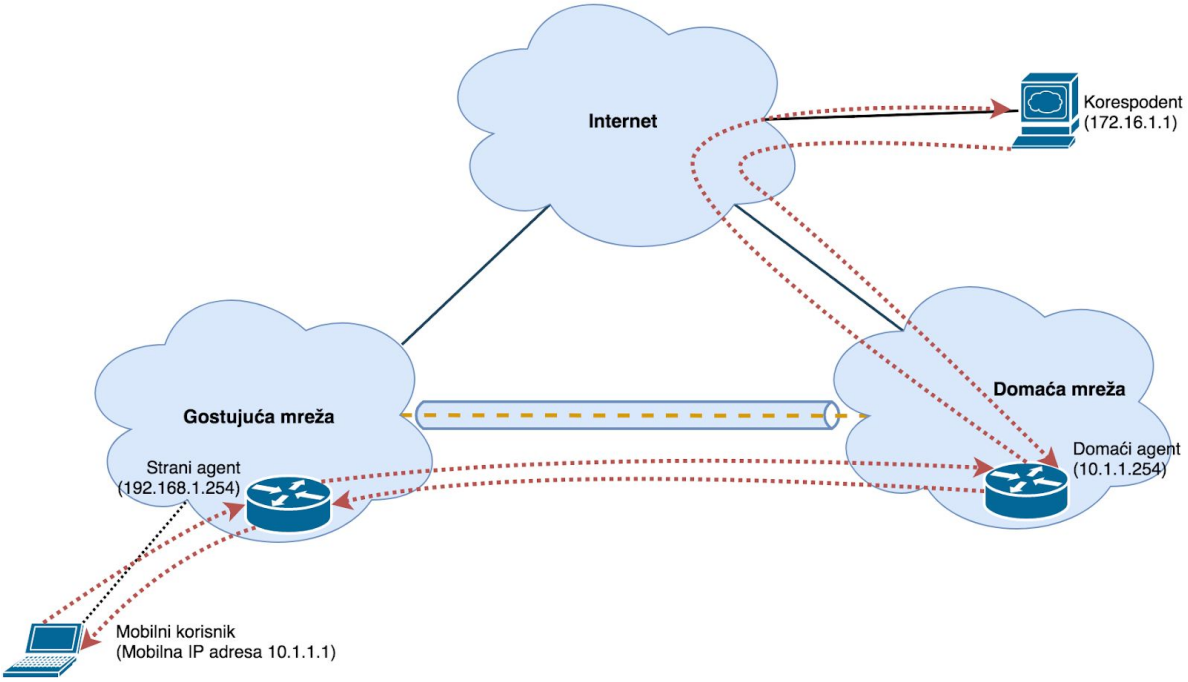
Ovakav pristup stvara trougaoni tok podataka u svakoj korespodenciji između mobilnog korisnika i udaljenog korespodenta.



Slika 2.3. MobileIP standard – problem trougaonog prenosa

Ovde je važno naglasiti da mobilni korisnik pakete šalje uvek direktno svom korespodentu, dok prijem ide postpuno drugačijom putanjom. Takav pristup u korespodenciji ne samo da nije efikasan već stvara i brojne probleme u radu. Neki od tih problema su u načinu na koji *firewall* uređaji tretiraju unidirekcione konekcije. Naime, *state-full* ruter neće propustiti dolazne konekcije ukoliko ne vidi adekvatnu odlaznu konekciju za isti mrežni tok.

Rešenje problema trougaonog rutiranja predloženo je dokumentom RFC 3024 [10] i podrazumeva primenu tzv. „reverznog rutiranja" kod mobilnog IP. Ovaj pristup se zasniva na ideji tunelovanja odlaznog saobraćaja do domaćeg agenta, čime se obzebeđuju identične putanje u odlaznom i dolaznom smeru.



Slika 2.4. MobileIP – reverzni tunel

Suštinski posmatrano, primenom Mobile IP koncepta modifikuje se proces „tretiranja" ARP paketa sa ciljem da se izvrši pravilno mapiranje domaće IP adrese u gostujućoj mreži. Upravo zbog rešavanja problema sa mapiranjem MAC i IP adresa, koristi se koncept Proxy ARP-a kao potprocesa Mobile IP rešenja. Uloga Proxy ARP-a je da obezbedi direktnu, virtuelnu vezu između udaljenog mobilnog korisnika i domaćeg agenta i omogući uspešno mapiranje domaće IP adrese mobilnog korisnika sa njegovom MAC adresom. Važno je naglasiti da se, zbog korišćenja *broadcast* transmisije, ARP kao rešenje za mapiranje adresa sa mrežnog sloja i adresa data link sloja, kao veoma zastarelo rešenje, retko koristi. Razlog za to leži u činjenici da se dodatno opterećuje propusni opseg računara kojima ne treba usluga konkretnog ARP mapiranja.

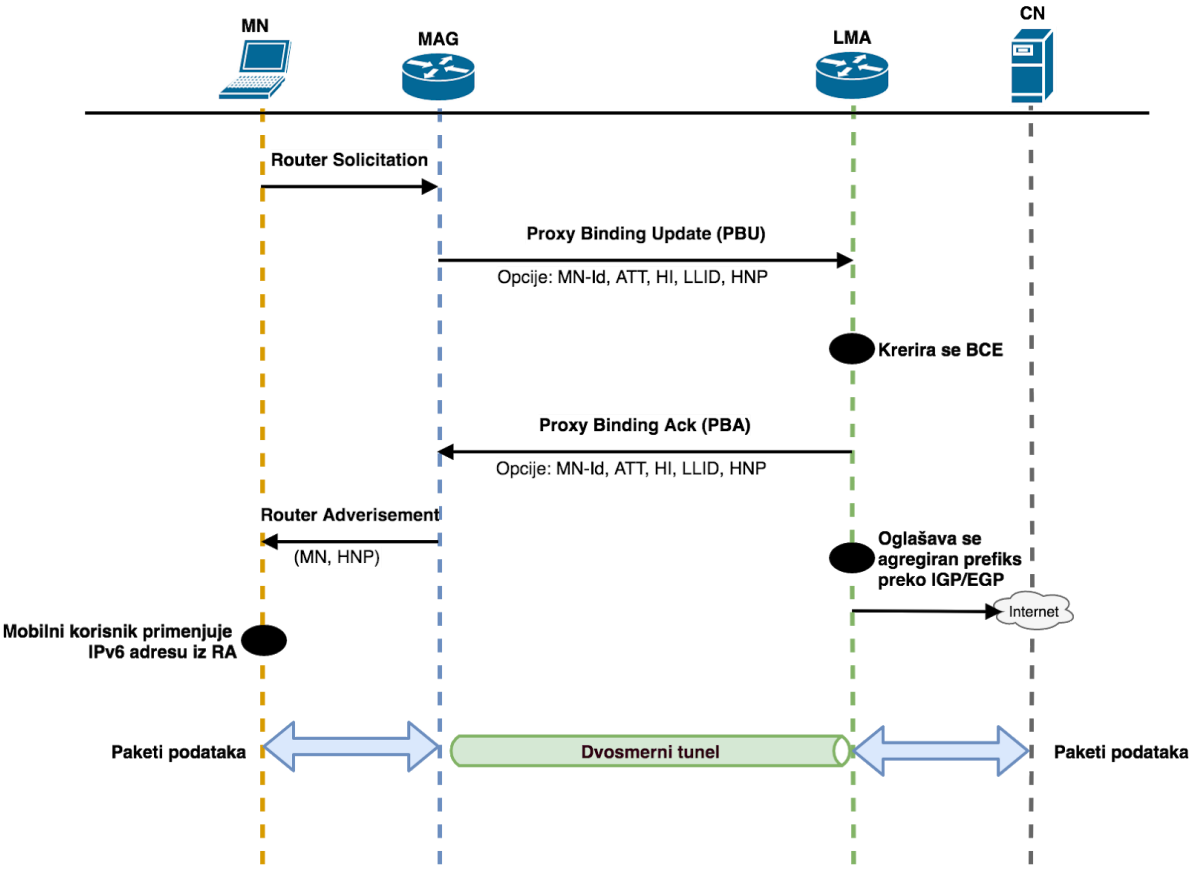
Implementacijom nove verzije internet protokola – IPv6 [7], odbačen je ARP mehanizam i uveden nov metod za razrešavanje adresa. Predloženo je novo rešenje za IP mobilnost koje se zasniva na korišćenju IPv6 protokola i primeni Mobile IP standarda [11], poznato pod nazivom Proxy Mobile IPv6 (PMIPv6) [12]. Ovaj protokol podrazumeva korišćenje sledećih entiteta [13]:

- LMA (*Local Mobility Anchor*) – predstavlja lokalno mobilno čvorište (sinonim za domaćeg agenta nekog korisnika), čija je uloga da održava mapiranje korisnika sa IPv6 adresom domaće mreže;

- MAG (*Mobile Access Gateway*) – uređaj zadužen za PMIP signalizaciju linka mobilnog korisnika. MAG je pristupni ruter u mreži gde se nalazi mobilni korisnik, pa ga možemo smatrati gostujućim agentom ako je korisnik u stranoj mreži. MAG obavlja sledeće zadatke:
    - od LMA uzima adresu koju mobilni korisnik treba da koristi,
    - zadržava mapiranu adresu kada se mobilni korisnik kreće između MAG-ova (pristupnih linkova), i
    - tuneluje saobraćaj između mobilnog korisnika i LMA.

Procedura dodeljivanja IP adrese mobilnom korisniku je sledeća:

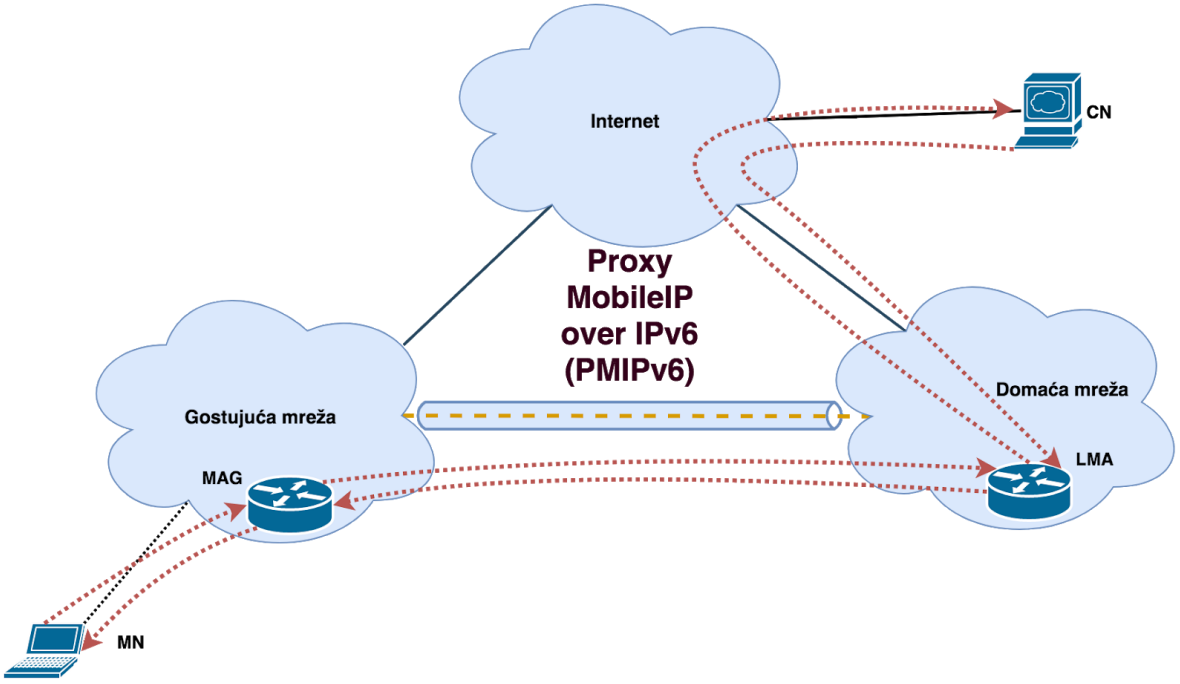
- Mobilni korisnik, nakon povezivanja na pristupni deo mreže (engl. *access part of network*), šalje paket ruterske solicitiacije (RS).
- Lokalni MAG prihvata RS paket, proverava autorizacije korisnika i prosleđuje PBU (engl. *Proxy Binding Update*) poruku do LMA korz tunel između LMA i MAG-a. Tom porukom MAG zahteva alokaciju mobilne IPv6 adrese za korisnika sa određenom MAC adresom.
- LMA procesira PBU paket i generiše PBA (engl. *Proxy Binding Ack*) paket sa podacima IP adrese koju korisnika treba da koristi.
- MAG koristi podatke iz PBA kako bi sastavio poruku ruterskog oglašavanja (RA) i prosledio do mobilnog korisnika.
- Mobilni korisnik koristi podatke RA paketa da populiše parametre domaće IP mreže.
- LMA oglašava prefiks domaćih IPv6 adresa u ruting protokolu kako bi bio dostupan iz udaljenih mreža.



Slika 2.5. PMIPv6 signalizacija tokova

Tok paketa između mobilnog korisnika i korespodenta izgleda tako što:

- mobilni korisnik sastavlja paket i isporučuje lokalnom MAG ruteru;
- koristeći formirani dvosmerni tunel, MAG ruter prosleđuje pakete do LMA rutera;
- LMA ruter dalje prosleđuje pakete prema standardnim pravilima rutiranja do pripadajućeg korespodenta;
- korespodent odgovara na primljen paket tako što ga upućuje LMA ruteru, koji isti prosleđuje tunelom do registrovanog MAG rutera zaduženog za traženu mobilnu IPv6 adresu;
- MAG ruter paket prosleđuje mobilnom korisniku.



Slika 2.6. Proxy MobileIP over IPv6 – tok podataka

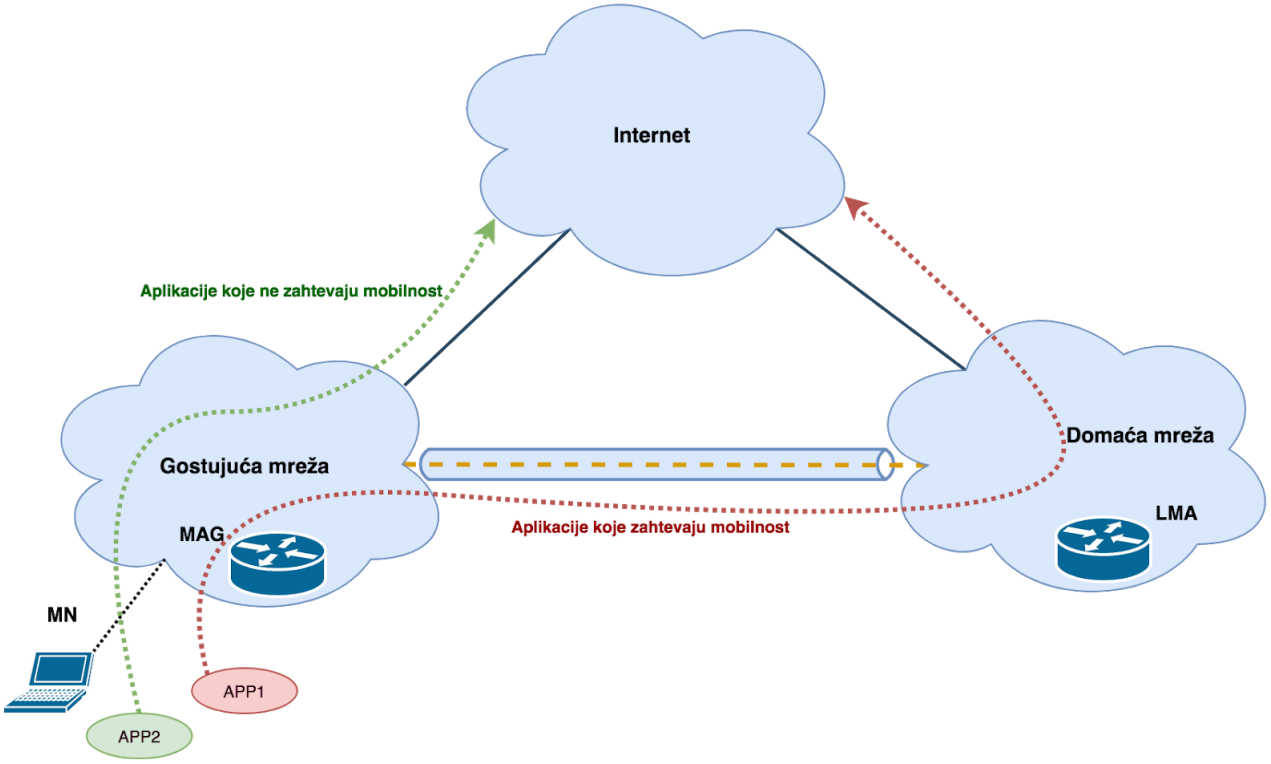
PMIPv6 protokol je otvoren standard [12] čije su najznačajnije karakteristike:

- transparentnost sa aspekta korisnika kome nije potreban poseban softver,
- konzistentnost IP adrese,
- održivost komunikacije na transportnom nivou TCP/IP modela,
- PMIPv6 podržava *dual-stack* rad u IP mreži, tj. istovremenu primenu IPv4 i IPv6 adresa,
- veza između LMA i MAG entiteta može biti omogućena preko IPv4 ili IPv6 mreže,
- svi mobilni korisnici za transfer paketa koriste jedan deljeni tunel između LMA i MAG entiteta (uglavnom se koristi GRE tunel, a ređe u kombinaciji sa IPsec-om),
- signalizacija je standardizovana i može se odvijati po IPv6 i IPv4 protokolu [14],
- podržan je u standardima kao što su 3GPP, 3GPP2, WiMax.

Korišćenjem tunela forsira se odvijanje saobraćaja između mobilnog korisnika i korespodenta preko domaće mreže i tunela između domaće mreže i udaljenog gostujućeg agenta. Ovakav pristup može u većoj meri da optereti domaću mrežu i ugrozi optimalnost rutiranja kod aplikacija i servisa koji ne zahtevaju perzistentnu IP adresu.

U cilju saobraćajnog rasterećenja (*offloading*) primenjuju se različite polise kojima se servisi koji zahtevaju IP mobilnost odvajaju od ostalih servisa [15]. Drugim rečima, saobraćaj servisa koji ne zahtevaju perzistentnost IP adrese mogao bi da se prosledi

direktno preko gostujuće mreže do korespodenta (umesto preko tunela i domaće mreže). Ovakvo rešenje doprinelo bi poboljšanju performansi kod servisa baziranih na HTTP protokolu, dok bi servisi kao što su VoIP i drugi koji zahtevaju perzistentnost IP adresa i dalje koristili tunel do domaće mreže.



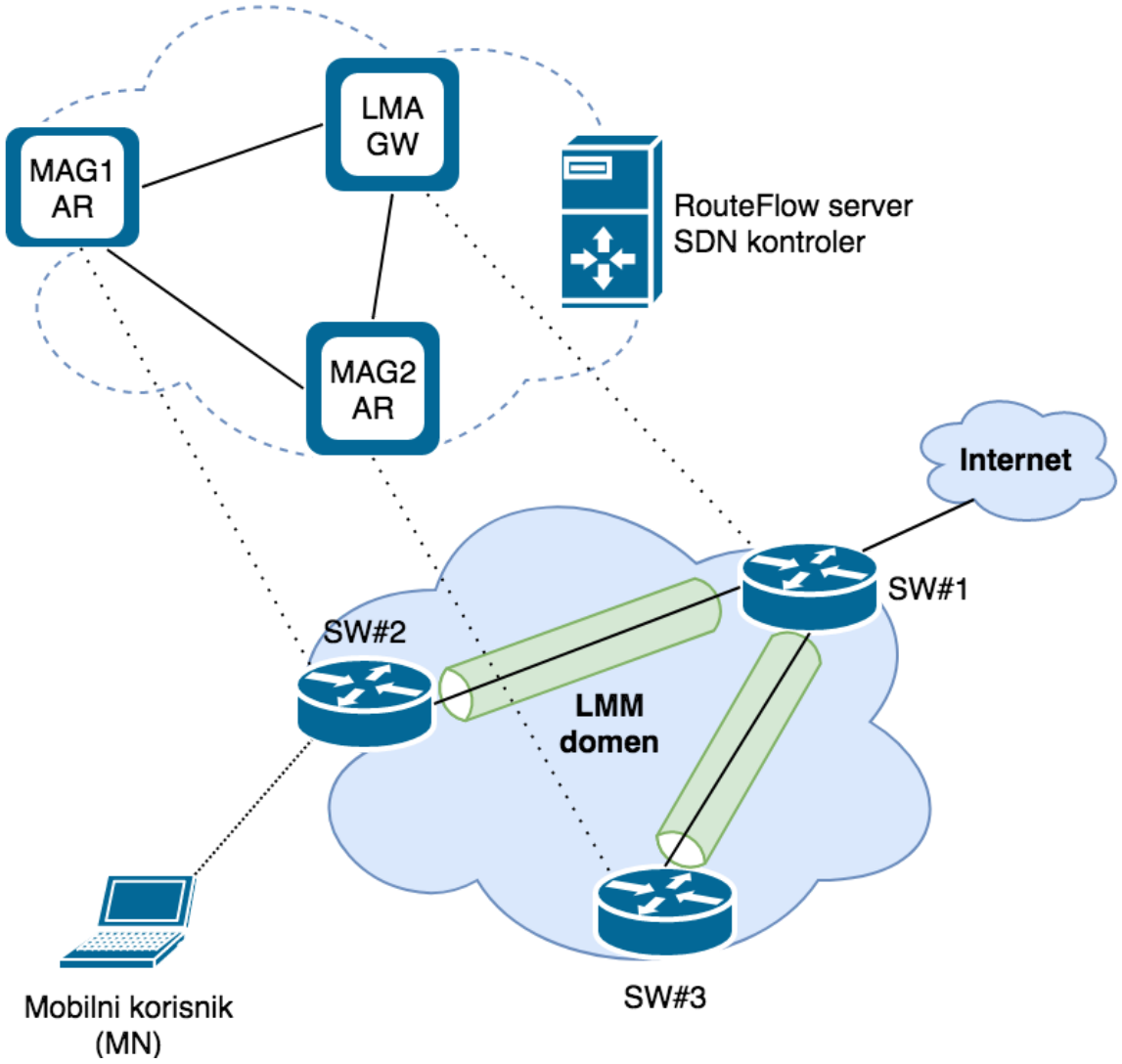
Slika 2.7. PMIPv6 rasterećenje *(offloading)*

SDN radovi

Kao što se vidi iz prethodno navedenog, PMIPv6 protokol problem mobilnosti korisnika rešava implementacijom specijalizovanih entiteta u mreži kao što su LMA i MAG kontroleri. U njihovoj nadležnosti su kontrola pristupa i realizacija tunela, funkcije koje često nisu implementirane u postojećim ruterima. Da bi se protokol u potpunosti primenio u praksi i da bi se time proširila postojeća funkcionalnost uređaja, neophodno je da se izvrši zamena postojeće opreme ili ubace dodatni uređaji u IP mrežu. Važno je naglasiti da i pored standardizacije ovakvo rešenje nije našlo masovnu primenu. Za to postoje dva osnovna razloga:

- Ovakav pristup rešavanju problema mobilnosti zahteva zamenu postojeće infrastrukture i/ili implementaciju specijalizovanog hardvera sa podrškom standardnim MIP protokolima, što posmatrano sa aspekta operatora iziskuje veće troškove.
- Porastom broja mobilnih korisnika raste i opterećenje domaće mreže, čak i ako se korisnici nalaze u gostujućim mrežama. Ovo je posledica neefikasnog rutiranja korisničkog saobraćaja kojim se paketi prvo rutiraju do domaće mreže, pa tek onda prosleđuju dalje, čime se unosi dodatno kašnjenje i loše utiče na QoE (engl. *Quality of Experience*).

Potreba da se problem mobilnosti korisnika u IP mrežama reši uvođenjem većeg stepena programabilnosti, odnosno primenom softverski definisanog umrežavanja (engl. *Software Defined Networking*), bila je predmet istraživanja autora u [16]. U tom istraživanju autori su pošli od osnovne karakteristike SDN umrežavanja, koja podrazumeva razdvajanje kontrolne ravni od ravni za prosleđivanje podataka. Njihova ideja bila je da problem mobilnosti reše sa oba aspekta, odnosno iz perspektive kontrolne signalizacije (kontroler je nadležan za realizaciju većine mrežnih funkcija kao što su rutiranje, menadžment, mobilnost i kvalitet servisa) i iz perspektive prosleđivanja podataka. Predloženo je rešenje koje podrazumeva da svaka mreža poseduje SDN svičeve (rutere – iz SDN perspektive ne postoji razlika) i kontroler, radi uređivanja mrežnih funkcija. Da bi prezentovali funkcionalnost predloženog rešenja, autori su u svom eksperimentu koristili virtuelne mašine (engl. *Virtual Machines* – VM) na kojima je implementiran PIMPv6 protokol kako bi se omogućila LMA ili MAG funkcionalnost. Važno je naglasiti da je za upravljanje svičevima u mreži (svičevi su deo jednog SDN domena) nadležan SDN kontroler. On na svim uređajima postavlja adekvatne tokove (engl. *flows*) u skladu sa tabelom ili polisama rutiranja. Kontroler je takođe zadužen i da „pripremi" tokove za tunel između različitih mreža kako bi se ostvarila komunikacija između LMA i MAG entiteta (ovi tokovi daju instrukciju sviču da izvrši enkapsulaciju paketa u IP-to-IP tunel i prosledi do MAG ili LMA entiteta).



Slika 2.8. PMIPv6 arhitektura preko SDN-a

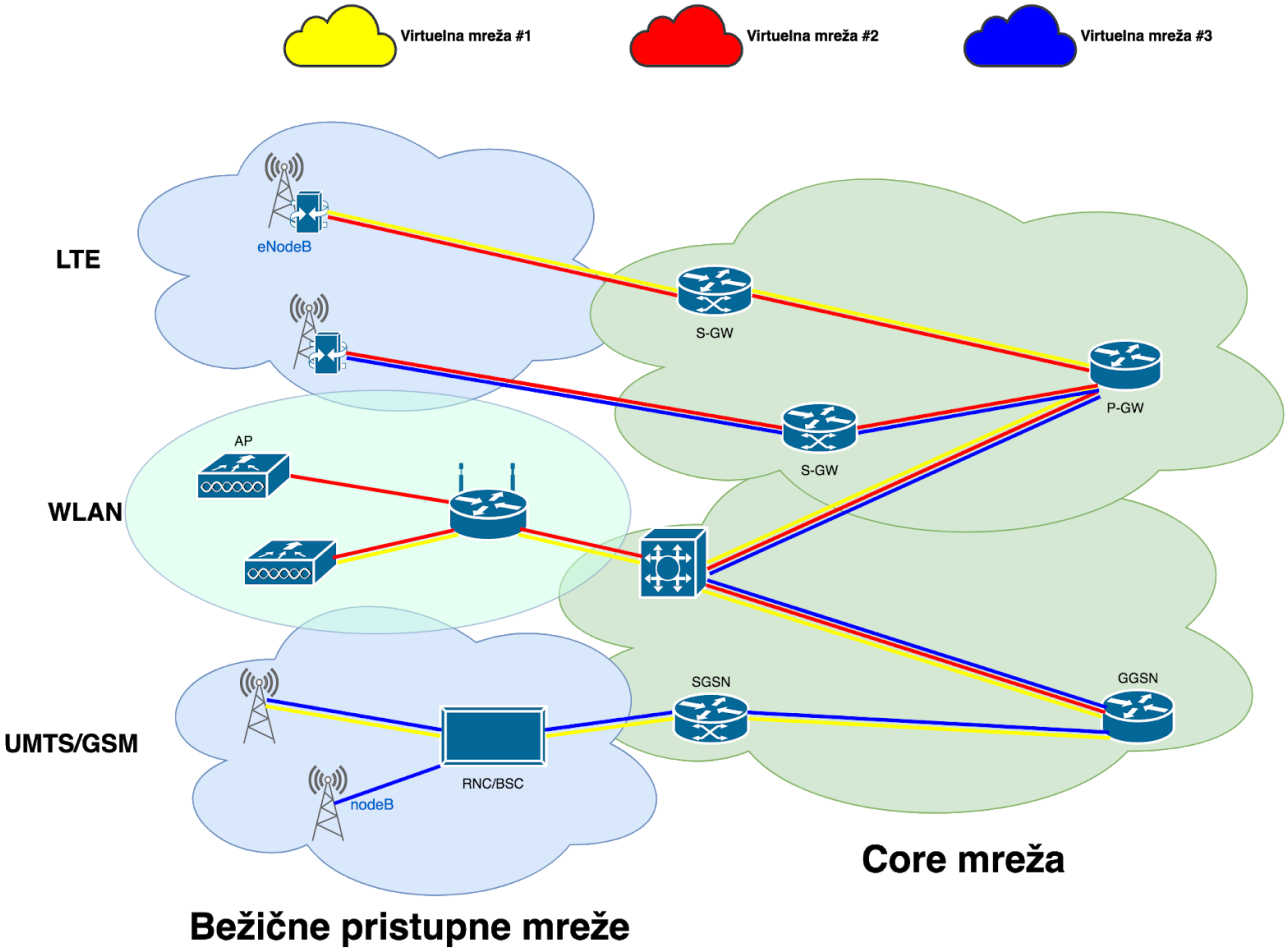
Veoma je važno naglasiti da se procedura prelaska mobilnog korisnika iz jedne mreže u drugu ne razlikuje mnogo od procedure definisane PMIPv6 protokolom. Ona se sastoji od sledećih koraka:

- korisnik promeni mrežu – npr. pređe sa SW1 na SW2;
- MAG2 (MAG kod SW2) inicira L3 handover proces slanjem PBU poruke ka LMA (*home* agentu) kako bi obavestio da je došlo do promene lokacije mobilnog korisnika;
- LMA ažurira zapise tako da MAG2 sad predstavlja agenta mobilnog korisnika i uzvraća PBA porukom kao potvrdom o tranziciji;
- SDN kontroler, na kom se izvršava i MAG, translira promene dobijene od LMA u instrukcije tokova i prosleđuje ih SDN svičevima;

- SDN svičevi odmah nakon prijema novih instrukcija prosleđivanja, nezavisno od kontrolera, počinju prosleđivanje podataka između mobilnih korisnika i LMA direktno enkapsulacijom paketa u IP-to-IP tunel.

Može se zaključiti da je procedura komunikacije između MAG i LMA entiteta praktično ostala nepromenjena u odnosu na PIMPv6 proceduru. Implementacijom SDN metode umrežavanja, dobijena je potpuna fleksibilnost i programabilnost SDN svičeva koji sada ne moraju da prate strog dizajn IP mreža. Drugim rečima, ne postoji potreba da se između mobilnog korisnika i MAG-a kreira virtualna mreža, jer će svič proslediti sve što mu kontoler naloži. Svič koji ima ulogu MAG-a može direktno da enkapsulira podatke korisnika u IP-to-IP tunel i da to obavlja mimo daljeg angažovanja kontrolera. Međutim, ovakvim pristupom ne rešava se problem optimalnog rutiranja saobraćaja između mobilnog korisnika i udaljenog korespodenta. Taj problem nije lako rešiti jer se u praksi i dalje susrećemo sa tradicionalno dizajniranim računarskim mrežama u kojima je implementiran koncept adresnog plana koji podrazumeva lokalnost radi lakše skalabilnosti.

Jasno je da se kao suštinski problem koji treba rešavati nameće problem optimizacije putanja u mobilnim mrežama. U tom smislu, u radu [17] predložen je koncept potpune virtualizacije bežičnih IP mreža, kojima se može pristupiti sa bilo koje lokacije.



Slika 2.9. SDN sa virtualnim mrežama

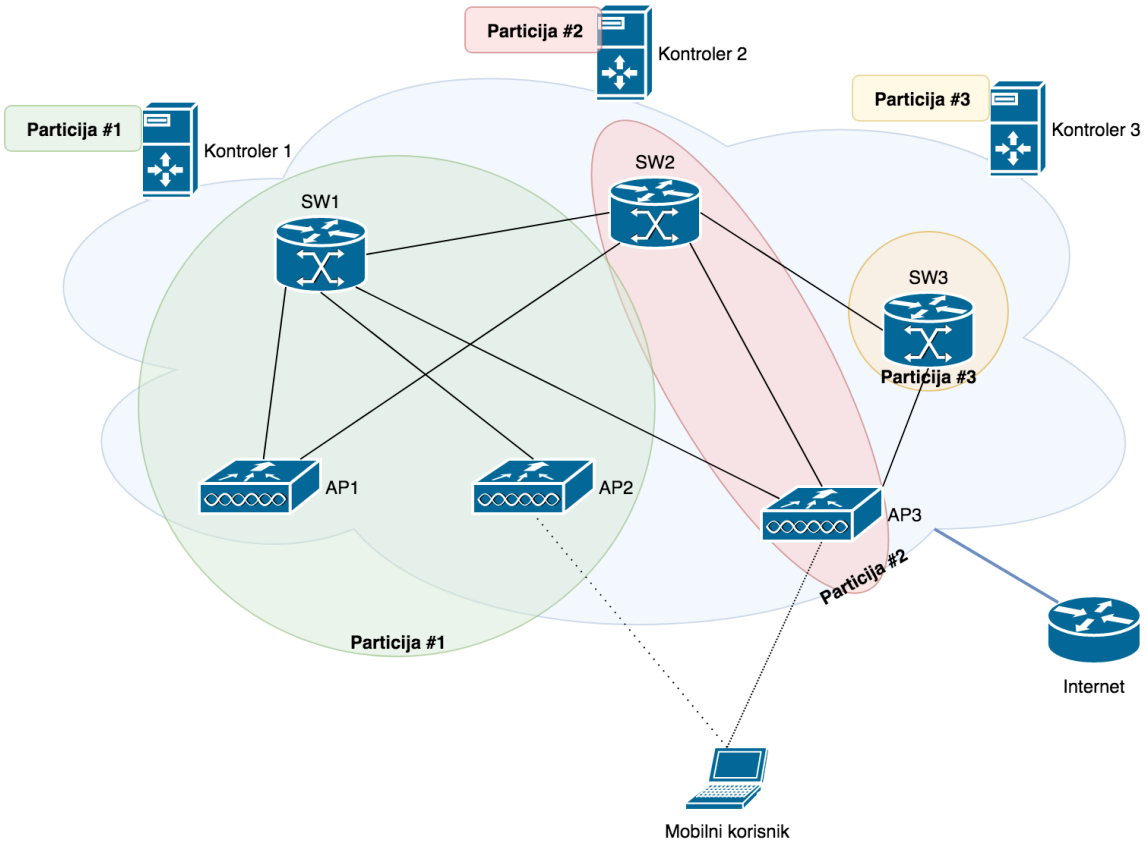
Koncept potpune virtualizacije u heterogenim mobilnim (*cell phone*) mrežama podrazumeva i implementaciju SDN kontrolera koji upravlja svim domenskim svičevima. Ovaj koncept autori su predstavili pod nazivom Bežična virtualizovana mreža (engl. *Wireless Network Virtualisation* – WNV), koji je prikazan na slici 2.9.

U okviru istraživanja, autori su predložili rešenje koje se zasniva na postojanju više virtualnih bežičnih mreža koje koriste deljenu bežičnu infrastrukturu. Ova infrastruktura koja se sastoji od virtualizovanih fizičkih resursa poseduje i SDN funkcionalnost, i odlikuje se visokim stepenom programabilnosti, koji se može posmatrati sa aspekta velike moći upravljanja tokovima podataka. Dakle, ovo rešenje zasniva se na potpunom prelasku sa tradicionalnog umrežavanja na SDN pristup umrežavanju. Ono otvara mogućnost rešavanja problema rutiranja saobraćaja mobilnih korisnika, ali istovremeno ističe i problem visoke cene realizacije navedenog koncepta.

S druge strane, u radovima [18] i [19] autori su predložili rešavanja problema mobilnosti korisnika po novom metodu koji se zasniva na primeni SDN mreža u tehnologiji ćelijskih pokrivanja. Međutim, prikazano rešenje je moguće primeniti samo na homogeni tip tzv. „ćelijske mreže", a problem mobilnosti se rešava na L2 sloju TCP/IP modela. Ovi radovi znatno su doprineli rešavanju problema mobilnosti u IP mrežama. U njima je prvi put, na konzistentan način, kombinacijom funkcionalnosti SDN tehnologije i primenom koncepta virtualizacije mreža, pokušano da se nađe rešenje problema mobilnosti.

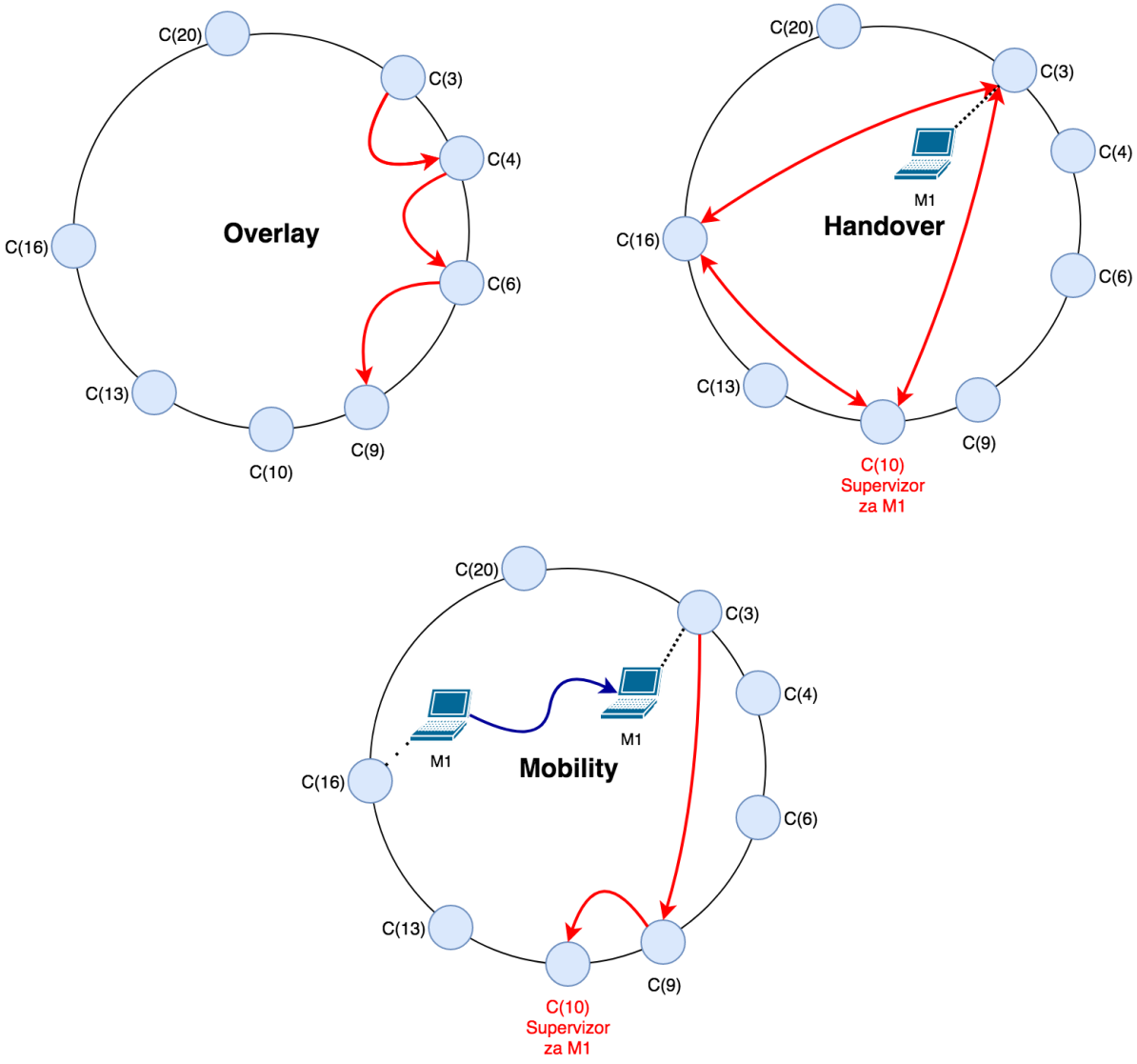
Problem mobilnosti u IP mrežama, krajem prošlog veka, naglo je dobio na intenzitetu i značaju sa pojavom novih, naprednijih tehnologija kao što je *Internet of Things* (IoT). Zato su u radu [20] autori upravo i razmatrali ovaj problem sa aspekta heterogenih bežičnih mreža u kojima su integrisani IoT uređaji. Naglasili su činjenicu da sa razvojem novih tehnologija raste i broj korisnika i povećava se fizička rasprostranjenost bežičnih mreža. Ukazali su na neophodnost rešavanja problema IP mobilnosti u smeru integracije različitih tehnologija. Naime, korisniku nije previše važno da li je mrežni servis dobio preko WiFi-ja, LTE-a ili Bluetootha, već samo njegova dostupnost. Autori su predložili da se primenom SDN tehnike umrežavanja reši problem integracije heterogenih mreža, uz održanje perzistentnosti u IP komunikaciji. Predložili su takođe i metodu alokacije resursa primenom SDN funkcionalnosti kako bi se korisnicima mobilnih mreža obezbedio veći kvalitet servisa. Nedostatak ovog rada je u tome što autori nisu uspeli da prikažu praktičnu primenu opisanog rešenja u većim mrežnim okruženjima.

U radu [21], autori su predložili sopstveno rešenje pod nazivom *UbiFlow,* tačnije softverski definisani IoT sistem za uređivanje mobilnih mrežnih tokova.



Slika 2.10. UbiFlow rešenje za hibridne mreže

Arhitektura UbiFlow rešenja prikazana je na slici 2.10. i predviđa postojanje više mreža različitih tehnologija kao što su WiFi, WiMax, LTE i sl. (na slici su označene različitim bojama). Za svaku od njih definisane su pristupne tačke (engl. *Access Point* – AP) i nadležni SDN kontroler particije. Da bi se obezbedila potpuna mobilnost korisnika pri prolasku kroz heterogene bežične mreže, unutar UbiFlow arhitekture uspostavljena je komunikacija između odgovarajućih kontrolera (kontroleri u svom lokalnom domenu, particiji, uređuju tokove na SDN svičevima). Rešenje koje su autori predložili koristi konzistentno heširanje za identifikaciju i održavanje stanja kontrolera.



Slika 2.11. UbiFlow prikaz tehnika mobilnosti

Kao što je prikazano na slici 2.11. kontroleri su numerički obeleženi i vezani u prsten kako bi se razlikovala tri stanja mobilnosti:

- *Overlay* – mobilni korisnik svojim kretanjem prelazi sa jednog na drugi, susedni kontroler,
- *Mobility* – mobilni korisnik prelazi iz jedne particije (tehnologije) na kontroler druge particije, i
- *Handover* – mobilni korisnik prelazi sa udaljenog kontrolera iste particije na drugi nesusedni kontroler.

Značaj pomenutog istraživanja nije samo u tome što se problem mobilnosti rešava na L2 i L3 sloju, već i u tome što se u eksperimentu koriste različite, trenutno aktuelne tehnologije. Međutim, osnovni nedostatak ovog rešenja je što se različite tehnologije

posmatraju sa aspekta vlasništva jedne organizacije. Time se nivo saradnje između kontrolera smanjuje, a rešenje je teško primeniti u praksi na mreže koje pripadaju različitim organizacijama.

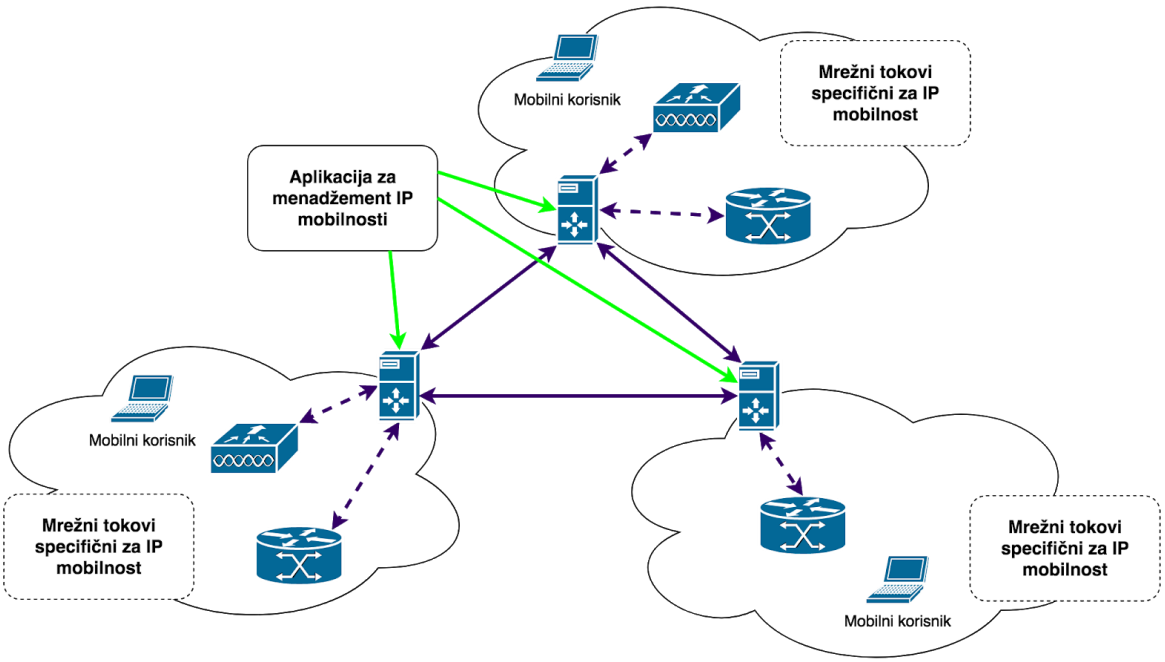
Virtualizacija adresnog prostora

Analiza prethodno pomenutih istraživanja i ograničenja u predloženim rešenjima jasno nameću potrebu za pronalaženjem univerzalnog rešenja za problem mobilnosti koje bi se:

- moglo primeniti na heterogene bežične mreže i
- implementiralo uz minimalne promene u postojećoj mrežnoj infrastrukturi, što bi iziskivalo i minimalne troškove.

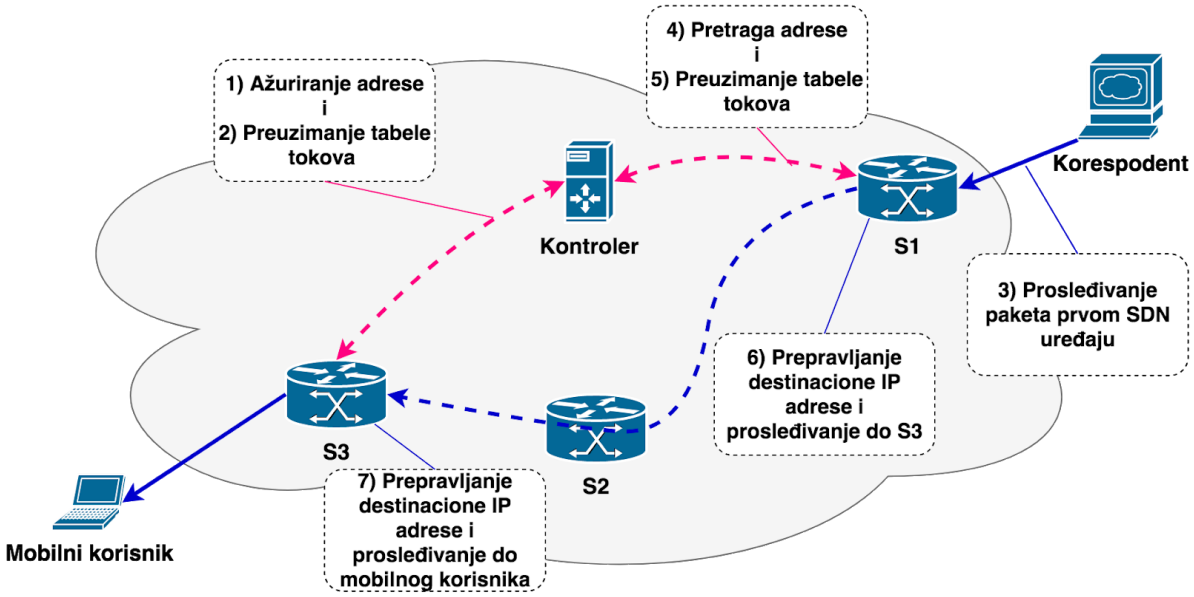
Polazeći od navedenih zahteva, autori su u radu [22] sproveli eksperiment u kome su koristili SDN tehnologiju umrežavanja kao bazu za rešavanja problema, ali su pri tome vodili računa da postoje različiti mrežni domeni, tj. provajderi usluga. Osnovu istraživanja čini pretpostavka da je moguće uspostaviti međusobnu saradnju između različitih provajdera kako bi se mobilnim korisnicima omogućila neprekidnost sesije unutar svih njihovih mreža. Pomenuti rad je veoma značajan jer se njime uvodi nov koncept tzv. „prepisivanja (engl. *re-write*) IP adresa" kako bi se postigla veća transparentnost u komunikaciji. Autori su u istraživanju pošli od toga da svaki mrežni domen (mreža internet provajdera) poseduje sopstveni kontroler koji upravlja saobraćajnim tokovima u tom domenu, te da je neophodno uspostaviti komunikaciju između kontrolera kako bi se obezbedila potpuna mobilnost i kako bi se „dogovorilo" o prelasku korisnika iz zone pokrivanja jedne u zonu pokrivanja druge mreže. U takvoj konstelaciji, mrežni kontroler ima dva zadatka:

- konstantno „mapiranje" korisnika sa njegovom trenutnom lokacijom (trenutnom gostujućom mrežom) i
- uređivanje, odnosno ažuriranje tokova ka SDN mrežnim uređajima za svakog mobilnog korisnika.



Slika 2.12. Koncept SDN mobilnih tokova

Posmatrano sa aspekta putanje paketa, tok podataka je korišćenjem SDN tehnologije (data plan) praktično već uređen u skladu sa pravilima koje je mrežni kontroler prethodno ažurirao u tabeli tokova (engl. *Flow table*). U eksperimentu su korišćeni Pox SDN kontroler, OpenFlow protokol za komunikaciju sa mrežnim uređajima i TCP protokol za komunikaciju između SDN kontrolera različitih mrežnih domena. Glavna ideja ovog istraživanja bila je da se korisniku omogući da prilikom prelaska iz jedne u drugu mrežu zadrži svoju originalnu IP adresu, ali da se za njega alocira i jedna dodatna rutabilna adresa u gostujućoj mreži. Potrebno je naglasiti da mobilni korisnik nema nikakve informacije o adresi koja je za njega alocirana u gostujućoj mreži, niti će je direktno primeniti. Uloga rutera na koji je povezan mobilni korisnik jeste da izvrši konverziju stalne IP adrese u gostujuću rutabilnu adresu.



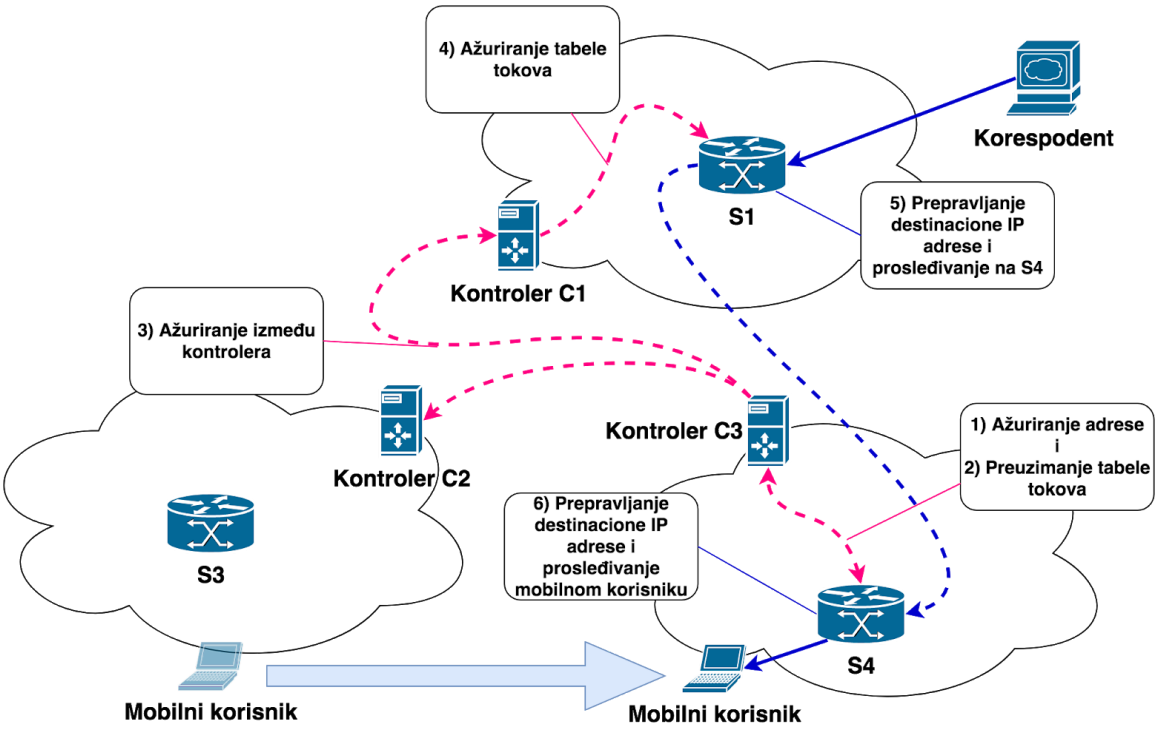
Slika 2.13. SDN mobilnost – tok procesa

Ako se pođe od trenutka kada se mobilni korisnik prvi put pojavljuje u mreži, onda se procedura obezbeđivanja potpune mobilnosti sastoji od sledećih koraka (slika 2.13):

- pristupni ruter (S3) preko kontrolne ravni prosleđuje SDN kontroleru zahteve za dobijanje IP adrese;
- SDN kontroler alocira rutabilnu IP adresu za mobilnog korsnika i čuva informacije koje se odnose na mapiranje korisnikove IP adrese i njegove lokacije;
- SDN kontroler ažurira tokove na svim drugim SDN ruterima kako bi omogućio prolaz paketa mobilnog korisnika ka korespodentima i obrnuto;
- kada od udaljenog korespodenta stigne paket namenjen mobilnom korisniku, ulazni ruter (prvi ruter u mreži koji je povezan sa internetom _ S1) zahteva od SDN kontrolera da obezbedi informaciju/instrukciju neophodnu za konkretni tok kako bi se izvršio prenos pristiglog paketa;
- SDN kontroler vraća instrukciju da bi se originalna destinaciona IP adresa (jedinstvena IP adresa korisnika) prepravila u rutabilnu IP adresu koju korisnik ima u toj mreži;
- pošto je paket sad sa rutabilnom IP adresom, on će bez problema proći kroz mrežu do pristupnog rutera (S3);
- pristupni ruter mora ponovo da prepravi destinacionu IP adresu paketa kako bi je vratio u originalno stanje, tj. u jedinstvenu IP adresu koju koristi mobilni korisnik.

Mobilni korisnik praktično uvek zadržava istu jedinstvenu IP adresu, ali se svaki njegov paket privremeno „prepravlja" kako bi prošao kroz gostujuću mrežu sve dok ne izađe iz nje, kada se vraća u originalno stanje i prosleđuje na internet u skladu sa tradicionalnim pravilima rutiranja.

U slučaju mobilnosti korisnika unutar istog mrežnog domena, jedan SDN kontroler je zadužen da reažurira tokove na SDN uređajima i da obezbedi ispravno mapiranje stalne IP adrese korisnika sa privremenom rutabilnom adresom.



Slika 2.14. SDN mobilnost – procesi usled mobilnosti

Važno je ponovo napomenuti da je prelazak mobilnog korisnika iz jedne u drugu mrežu moguć samo ako postoji efikasna komunikacija između njihovih SDN kontrolera (svaki SDN kontroler je zadužen samo za svoju mrežu i zato je neophodno da oni međusobno razmenjuju instrukcije o mobilnosti korisnika). Na slici 2.14. se vidi da kada mobilni korisnik pređe iz mreže 2 u domen mreže 3, pristupni ruter obaveštava kontrolera mreže 3 o novom korisniku i započinje prethodno definisanu proceduru. Da bi predloženo rešenje imalo smisla, neophodno je da se kontroler ulazne (domaće) mreže obavesti o privremenoj rutabilnoj adresi preko koje je mobilni korisnik dostupan. Gostujući kontroler obaveštava sve kontrolere sa kojima sarađuje da je određeni mobilni korisnik, sa njegovom

univerzalnom IP adresom, sada dostupan preko nove privremene rutabilne adrese. Kontroler domaće mreže ažurira tokove tako što pristigli paket, poslat na univerzalnu IP adresu mobilnog korisnika, prepisuje na novu rutabilnu adresu u drugom mrežnom domenu.

Radi testiranja predloženog rešenja, autori su izveli odgovarajuću simulaciju u Mininet okruženju [23], u kome su implementirali opisanu višedomensku infrastrukturu sa Pox kontrolerima. Merenjem različitih performansi, odnosno indikatora u mreži, autori su uporedili svoje rešenje sa rešenjem koje se bazira na PIMPv6 protokolu i dobili mnogo bolje rezultate i brži odziv u trenucima prelaska mobilnog korisnika iz jedne u drugu mrežu. Ova prednost postignuta je upravo kombinacijom visoke programabilnosti SDN tehnologije i primene tehnike maskiranja permanentne IP adrese nekom privremenom adresom.

Predlog novog rešenja

Detaljnom analizom postojećih rešenja koja se odnose na problem IP mobilnosti, može se izvesti zaključak da kod većine autora preovlađuje pristup zasnovan na podeli mrežne infrastrukture na domaću (home) mrežu i gostujuću mrežu. Takav pristup omogućava da se standardni principi rutiranja primene na domaće adrese mobilnih korisnika kada su u domaćoj mreži, a da se u slučaju prelaska u neku gostujuću mrežu primeni prosleđivanje paketa do nove lokacije. Dakle, posledica ovakvog rešenja je potreba za prerutiranjem saobraćaja iz domaće u gostujuću mrežu, što ukazuje na problem nepostojanja optimalnog algoritma za rutiranje saobraćaja. S druge strane, da bi se došlo do fleksibilnijeg rešenja, koje između ostalog treba da podrži i heterogene mreže, protokoli za upravljanje (registraciju korisnika u domaćoj mreži, prijavu korisnika u gostujućoj mreži, formiranje tunela između gostujućeg i domaćeg agenta i sl.) postaju znatno kompleksniji i teži za praktičnu implementaciju. Kompleksnost realizacije protokola se ogleda u njegovom slabom prodoru i nedovoljnoj zastupljenosti u bežičnim mrežama. Može se sasvim opravdano reći da nijedan od opisanih protokola nije široko zastupljen u bežičnim mrežama, a razlog za to je u najvećoj meri kompleksnost njihove implementacije i konfiguracije.

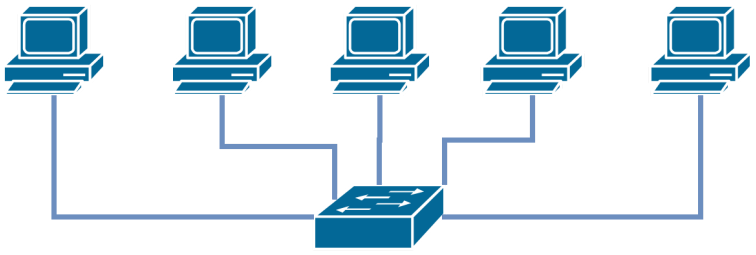
Imajući u vidu sve gorenavedeno, novo rešenje problema IP mobilnosti u heterogenim bežičnim mrežama mora biti znatno jednostavnije za implementaciju. Samo će tako biti moguće da se unutar internet zajednice obezbedi njegova šira primena. Jednostavnost implementacije predstavlja važan, ali ne i presudan faktor koji može da utiče na primenu novog rešenja. Važno je naglasiti da novo rešenje u materijalnom smislu mora u što manjoj meri da optereti provajdera. To je moguće samo ako primena novog rešenja zahteva minimalne promene u postojećoj bežičnoj mrežnoj infrastrukturi i ako postoji kooperativnost između mrežnih provajdera. Može se zaključiti da realizacija ovakve ideje zahteva napuštanje koncepta domaća–gostujuća mreža i tranziciju razmišljanja u internet zajednici ka konceptu postojanja tzv. „celokupne hibridne federativne mreže".

Dakle, ključni faktor u definisanju novog, jednostavnijeg rešenja za problem IP mobilnosti u heterogenim bežičnim mrežama jeste teza da se problem može rešiti korišćenjem postojećih, već dokazanih tehnologija i rešenja. Zato je neophodno da se pre iznošenja predloga novog rešenja opišu tehnologije na kojima se ono zasniva.

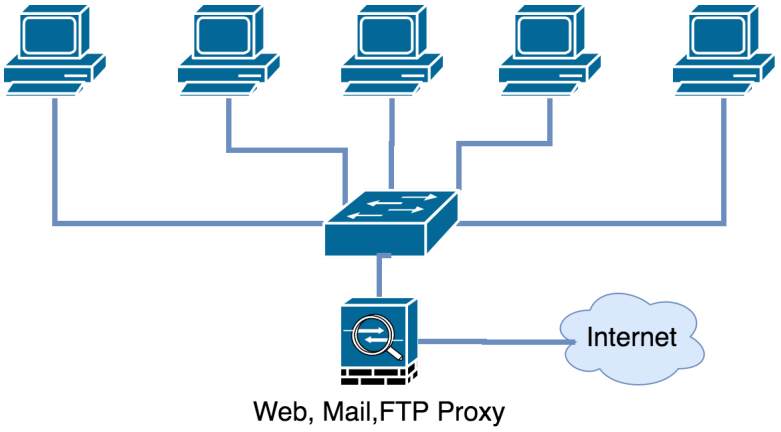
Network Address Translation (NAT)

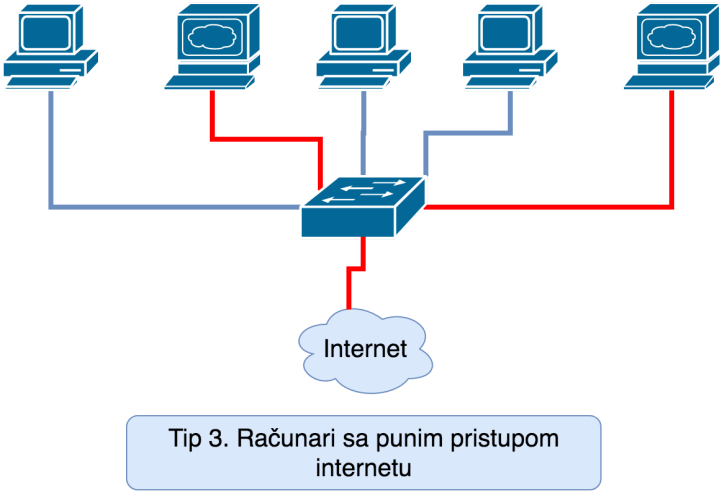
Tehnologija translacije mrežnih adresa (engl. *Network Address Translation* – NAT) je standardna [24] tehnika koja se koristi za preslikavanje, odnosno translaciju IPv4 adresa jednog u IPv4 adrese drugog domena. Nastala je kao posledica istrošenosti IPv4 adresnog prostora (poslednja IPv4 adresa je dodeljena), ali i prave eksplozije u pogledu rasta broja korisnika na internetu i pojave novih tehnologija. Time je ona dobila na značaju jer su njenom primenom postavljeni jasni standradi kojima se definiše način alokacije IP adresa u privatnim mrežama i na javnom internetu [25]. Tako je omogućeno da bilo koji uređaj u LAN mreži obavezno poseduje jedinstvenu IPv4 adresu, a da s druge strane za komunikaciju na javnom Internetu koristi javne, globalno rutabilne IPv4 adrese [26]. Standardom [25], predviđena je podela računara/uređaja unutar lokalnih, tj. LAN mreža, u skladu s njihovim često različitim potrebama, na:

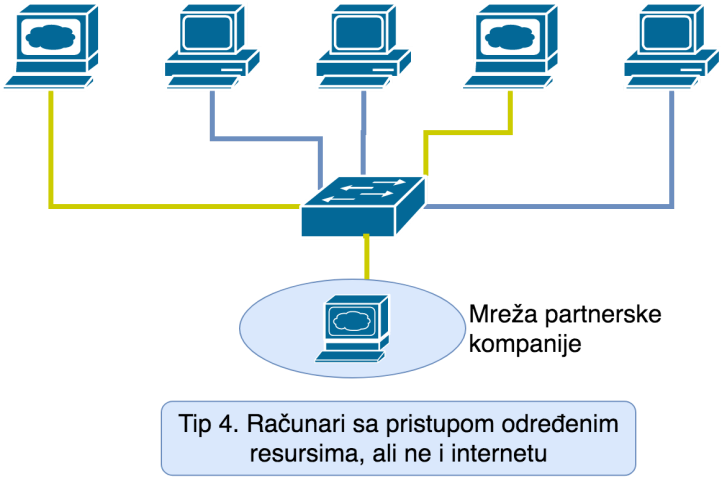
1. računare koji nemaju potrebu za komunikaciju sa uređajima u drugim mrežama, odnosno na internetu,
2. računare koji imaju potrebu da pristupe samo malom broju aplikativnih servisa (kao npr. mail, ftp i sl.), što se može omogućiti preko gejtveja na aplikacionom sloju,
3. računare kojima je potrebna puna IP konektivnost ka spoljnim mrežama i internetu, i
4. računare koji treba da pristupe samo određenim spoljnim mrežama, ali ne i celom internetu.



Tip 1. Računari sa isključivo lokalnim pristupom



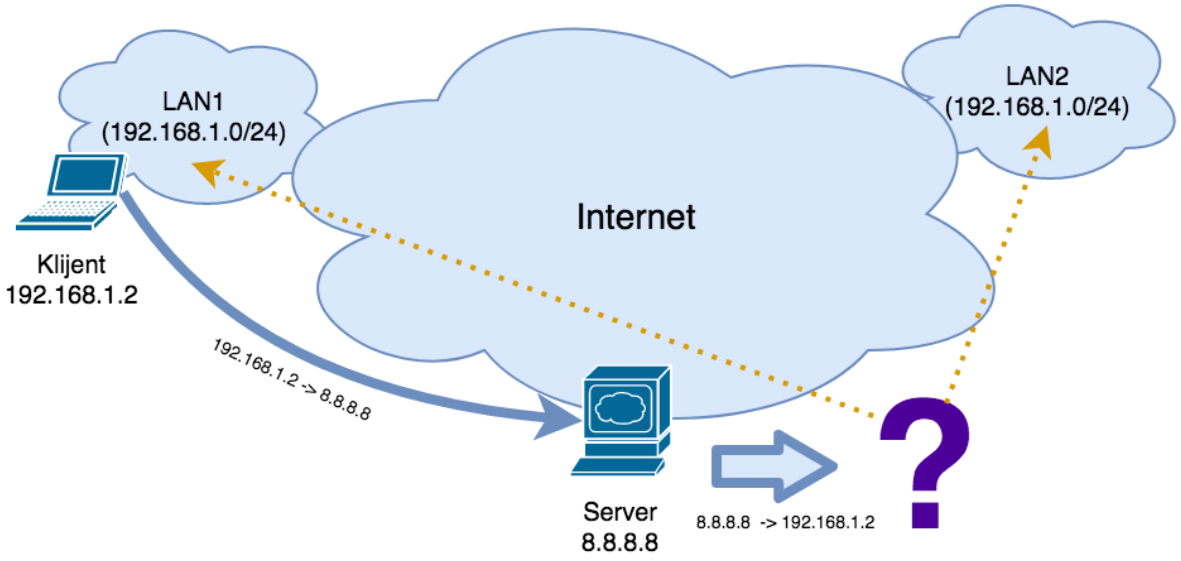




Slika 3.1. Tipovi računara u privatnim mrežama

Da bi se primenila politika definisana standardom [25], odnosno da bi računari/uređaji u LAN mreži mogli da komuniciraju u skladu sa navedenim pravilima, neophodno je svakom uređaju/računaru u mreži dodeliti adekvatnu IP adresu. Ako se računaru dodeli privatna IP adresa, on može biti u ulozi 1, 2 ili 4, a ukoliko mu se se dodeli javna IP adresa, onda on može da komunicira sa svima na internetu (uloga 3). Ako se dogodi da se na dva interfejsa postave privatna i javna IP adresa, onda taj računar može imati ulogu 3 i biti posrednik uloge 2 za ostale računare bez pristupa internetu. Slučaj 4 je situacija u kojoj postoje dve privatne mreže sa različitim privatnim adresnim prostorom i sa međusobnom vezom. One tada mogu da razmenjuju podatke, ali ne preko interneta.

Bitno je naglasiti da računari koji poseduju IP adrese iz privatnog opsega mogu da komuniciraju samo sa drugim računarima, u istom ili različitiom privatnom opsegu. Oni ne mogu da komuniciraju sa računarima na internetu, iz prostog razloga što računar sa interneta ne zna kome treba da pošalje odgovor (privatne IP adrese nisu globalno jedinstvene jer može postojati veliki broj računara sa istom privatnom IP adresom).



Slika 3.2. Problem komunikacije između računara u privatnoj i javnoj mreži

Stroga klasifikacija na računare sa ulogama 1–4 napravila je mreže prilično statičnim i nefleksibilnim. Svaka promena njihove uloge u mreži tražila je velike intervencije kako na infrastrukturnom, tako i na planu softverske konfiguracije. Zato se kao rešenje pojavila NAT tehnologija definisana standardom [24].

Može se reći da je osnovna uloga NAT servisa da omogući dinamičko (automatsko) prevođenje IP adresa iz jednog u drugi domen (NAT servis se izvršava na ruteru koji je jednim interfejsom spojen na jedan, a drugim na drugi IP domen). To se prvenstveno odnosi na translaciju privatnih IP adresa u javnu, tj. globalno jedinstvenu IP adresu (bilo koji računar iz LAN mreže sa privatnom IP adresom komunicira sa računarom na internetu posredstvom NAT servisa). Primenom NAT servisa, omogućava se fleksibilnije funkcionisanje računarskih mreža i njihovih korisnika, a time se ukida i potreba za strogom klasifikacijom uloga prikazanih na slici 3.1. Međutim, može se desiti da iz različitih razloga (npr. bezbednosne svrhe), ipak postoji potreba za podelom uloga u LAN mreži. Primenom NAT servisa, takve zahteve je moguće izvršiti na jednom ruteru, što značajno pojednostavljuje menadžment mrežnih uređaja.

Radi zadovoljavanja različitih mrežnih potreba, NAT servis se može klasifikovati [24] u dve grupe:

- Bazični NAT
    - statički
    - dinamički, i
- *Network Address Port Translation* – NAPT.

Bazični NAT

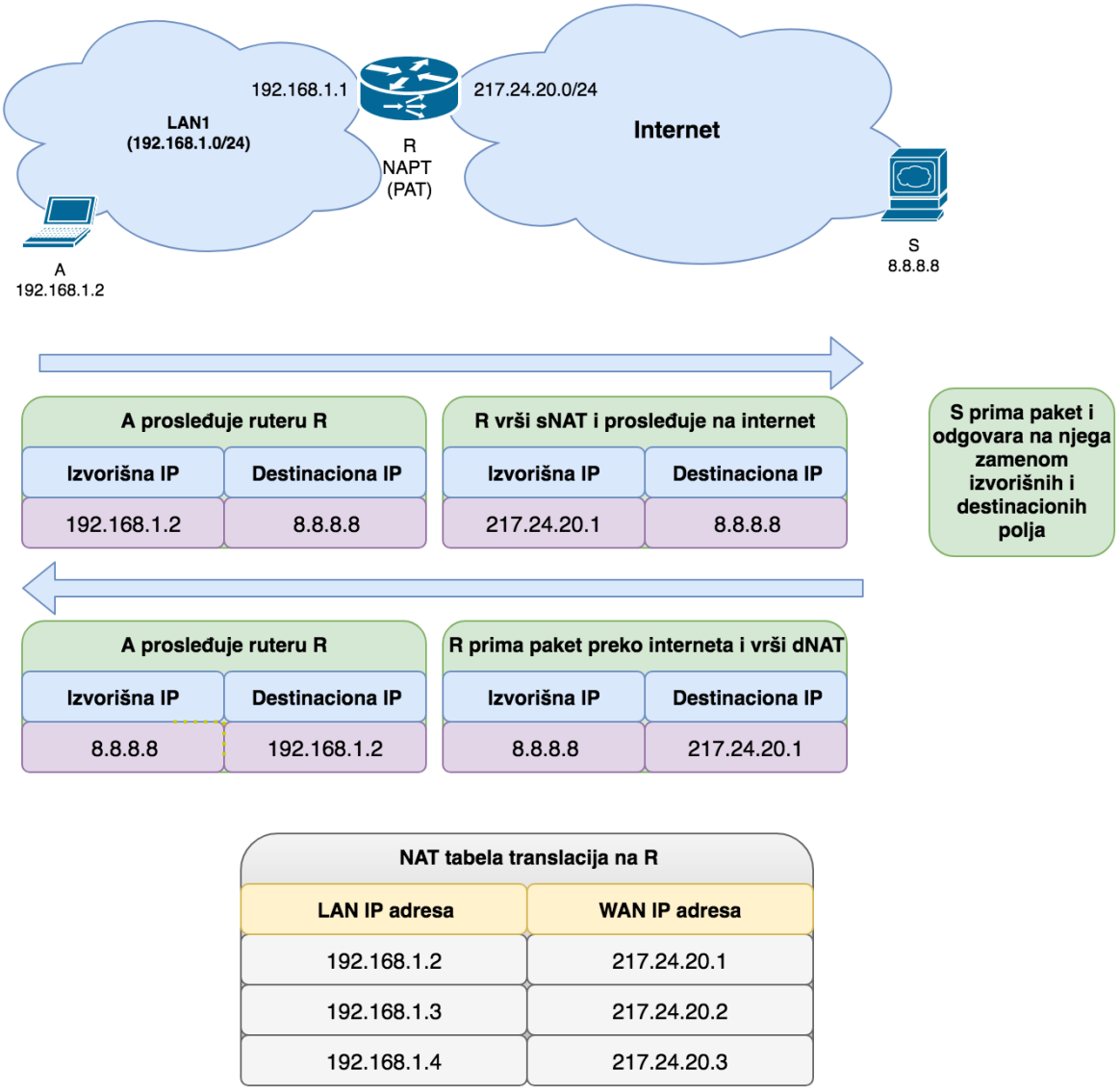
Osnovni ili bazični NAT servis izvršava se na mrežnom sloju TCP/IP referentnog modela, odnosno na ruteru, i podrazumeva translaciju, tj. prepravku IP adresa u zaglavlju (engl. *header*) IP paketa iz jednog domena (npr. privatne mreže) u jedinstvenu IP adresu drugog domena (npr. internet). U praksi, NAT servis možemo tretirati kao proceduru koja se sprovodi nad izvornom ili odredišnom IP adresom u IP paketu. Zato razlikujemo:

- *Source NAT* (tzv. sNAT) – procedura translacije se sprovodi nad izvornom IP adresom u paketu, i
- *Destination NAT* (tzv. dNAT) – procedura translacije se sprovodi na odredišnoj IP adresi paketa.

Sam proces implementacije NAT servisa može biti statičkog i dinamičkog karaktera. Statički NAT se izvršava centralizovano i zahteva da administrator na ruteru „ručno” konfiguriše pravilo (zapis) statičke prirode za mapiranje između privatnih i javnih IP adresa. Time računari obuhvaćeni definisanim pravilom translacije iz uloge 1 prelaze u ulogu 3.

Ostali računari u LAN mreži, koji nisu obuhvaćeni NAT servisom, ostaju u ulozi 1. Ovakav vid translacije IP adresa često se naziva „jedan na jedan mapiranje" i garantuje perzistentnost mapiranih adresa (garancija da će se računar neke privatne mreže uvek videti „spolja" sa mapirane javne adrese).

Procedura NAT translacije „s-kraja-na-kraj" prikazana je na slici 3.3.



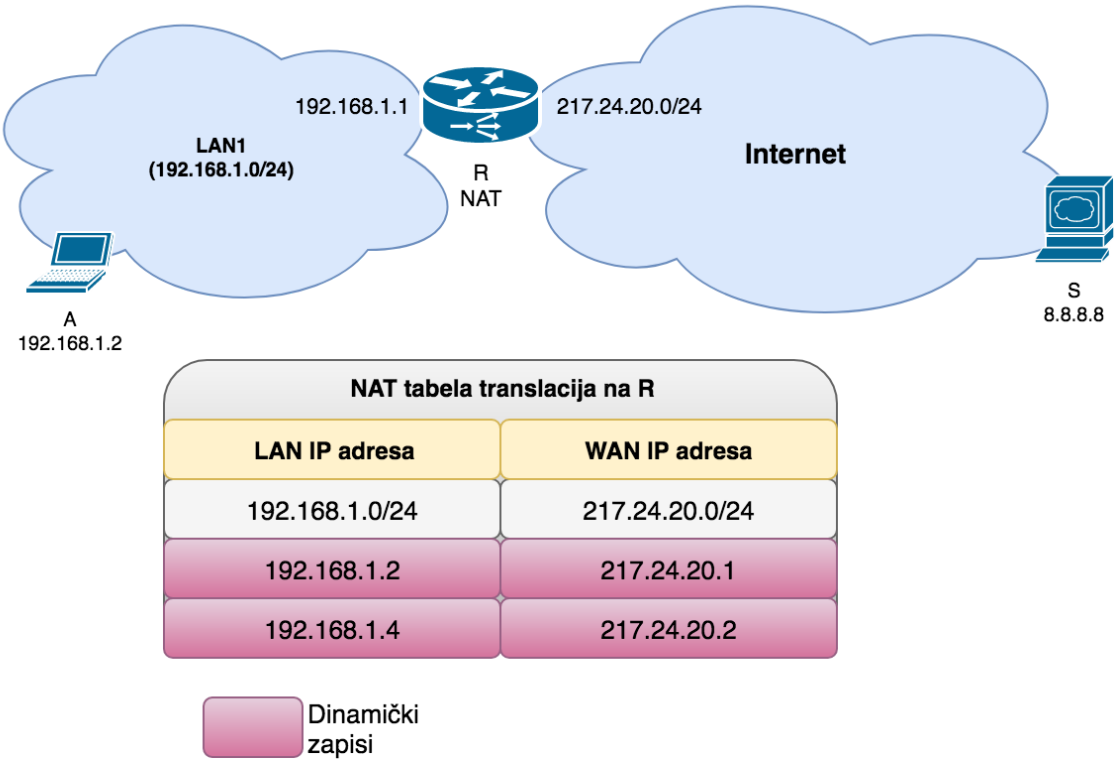
Slika 3.3. Bazni NAT sa statičkim zapisima u NAT tabeli

Zbog brzog rasta interneta i potrebe za sve većom integracijom različitih mreža u jednu globalnu mrežu, moralo je da se započne sa restriktivnim dodeljivanjem javnih (jedinstvenih) IP adresa. Već onda je bilo jasno da nije praktično da svi računari u nekoj LAN mreži imaju pristup internet servisima. Kompanije i organizacije dobijale su manji broj javnih IP adresa nego što je bio njihov plan razvoja i proširenja. Stvorila se potreba za što

fleksibilnijm uređenjem uloge računara u LAN mreži, tako da čak i računari koji uglavnom koriste samo resurse lokalne mreže mogu privremeno dobiti i uslugu servisa sa interneta.

Dinamički NAT nastao je kao rezultat modifikacije prethodno opisane procedure statičkog povezivanja privatnih i javnih adresa, koja se ogleda u tome što svaki računar u skladu sa svojim potrebama može privremeno dobiti javnu adresu kako bi koristio resurse interneta (slika 3.4). Umesto fiksnih (statičkih) veza između privatnih i javnih IP adresa, uvode se opsezi privatnih i javnih adresa između kojih se može vršiti translacija (definiše se opseg privatnih IP adresa koje imaju pravo na mapiranje sa nekom javnom adresom iz unapred definisanog opsega). Uloga rutera na kojem je konfigurisan dinamički NAT servis jeste da uređaju kojem je dozvoljen pristup internetu alocira jednu od slobodnih javnih IP adresa iz raspoloživog opsega/bazena (engl. *pool)* javnih IP adresa. Privremen karakter translacije obezbeđuje se tako što na ruteru postoji definisan period (*timeout*) u kom će posmatrana translacija biti validna i nakon kojeg će ista automatski biti izbačena iz NAT tabele (tabela sa svim zapisima o translaciji).

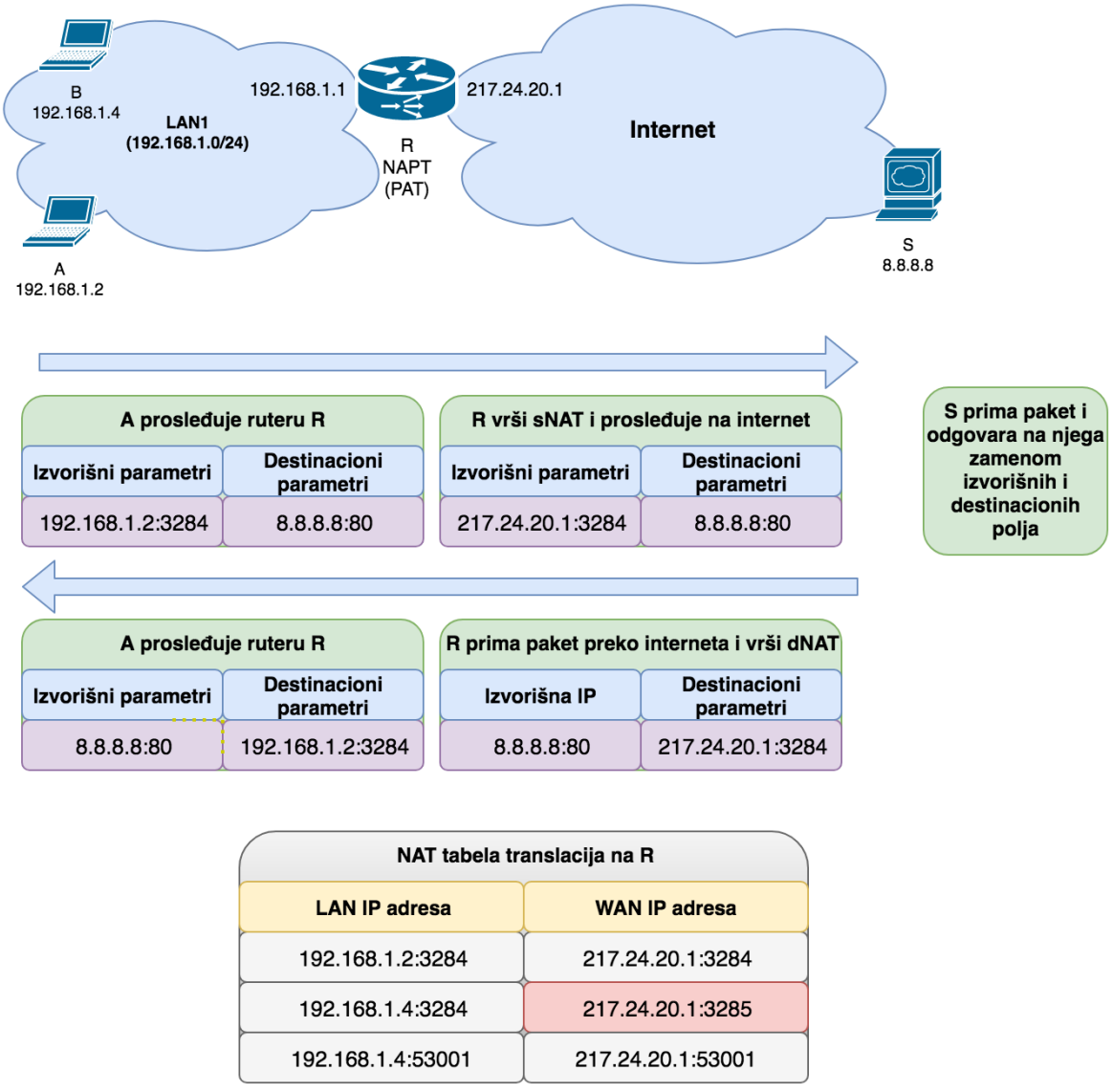
Važno je istaći da NAT servis ne može biti realizovan ako zahtev dođe od računara koji nemaju pravo na NAT servis, ili ako na raspolaganju nema više slobodnih javnih IP adresa u definisanom bazenu javnih IP adresa.



Slika 3.4. Dinamički NAT

Network Address Port Translation – NAPT

Osnovni nedostatak bazičnog NAT servisa je taj što ovaj servis omogućava translaciju adresa tipa „jedan-na-jedan". Primenom ovog principa, broj preslikavanja, odnosno kapacitet NAT servisa ograničava se na raspoloživ broj javnih IP adresa. Praktično, bazični NAT ne daje suštinski doprinos rešavanju problema koji se odnosi na to kako usporiti potrošnju javnih IP adresa, a da se što većem broju korisnika obezbedi pristup internetu. Da bi se pronašlo rešenje, izvršena je dodatna modifikacija NAT servisa. Ova promena se ogleda u tome što se osim podataka sa mrežnog sloja (IP adresa), translacija sprovodi i na podacima sa transportnog sloja (TCP/UDP portovi). Takav koncept zasniva se na ideji, ili bolje rečeno potrebi, maksimalnog iskorišćavanja svake javne IP adrese, tako što se NAT proces sprovodi na nivou konekcija i tokova transportnog sloja, a ne samo na nivou mrežnog sloja. Drugim rečima, NAT tabela sadrži translacije u kojima se vodi računa i o preslikavanjima portova na transportnom sloju. Ovde je važno napomenuti i to da kod TCP komunikacije postoji konekcija pa održavanje zapisa u NAT tabeli prosleđivanja predstavlja jasnu evidenciju o stanju konekcije. UDP komunikacija je malo komplikovanija zbog činjenice da ne poznaje konekcije i da u njoj ne mora da postoji perzistentan prenos između dve strane (problem UDP komunikacije putem NAPT-a rešen je tako što je od poslednjeg prenosa paketa postavljen timeout period [27]).



Slika 3.5. NAPT ili PAT translacija

Postupak prenosa paketa sa NAPT translacijom prikazan je na slici 3.5. Računar A, koji poseduje adresu iz privatnog opsega, želi da pošalje paket do računara S koji je na internetu. A prosleđuje paket ruteru R na kome se izvršava NAPT servis. R hvata izvorišnu IP adresu i broj porta iz pristiglog paketa, i upoređuje sa NAT tabelom translacije. Ukoliko zapis već postoji, vrši se sNAT procedura po parametrima koji se nalaze u tabeli. Ako zapis ne postoji, NAPT servis pokušava da alocira zapis sa javnom IP adresom koju ruter poseduje i brojem porta identičnim dolaznom paketu. Ukoliko je par javna IP adresa i željeni broj porta već alociran za neku drugu konekciju, ruter inkrementuje broj porta dok ne dođe do slobodnog porta. Tada R vrši sNAT, tj. prepravlja izvorišnu IP adresu (ponekad i port) primljenog paketa u javnu IP adresu i broj porta, koji su dobijeni iz NAT tabele prosleđivanja. Prepravljeni paket rutira na osnovu ruting protokola i predaje svom ISP-u.

Paket putem interneta pristiže do računara S jer on ima jedinstvenu javnu IP adresu. Računar S odgovara na zahtev računara A uzvraćajući paket na javnu IP adresu mapiranu za A. Tehnički, S zamenjuje mesta izvorišnih i odredišnih parametara paketa (MAC, IP adresu, broj porta) i šalje ih svom ruteru. Kako je povratni paket takođe adresiran za javnu IP adresu, takav paket bez problema dolazi do rutera R. Ruter R proverava paket i nalazi da se IP adresa i port koji se nalazi u destinacionim poljima paketa poklapaju sa zapisom NAT tabele translacija. Tada R započinje dNAT proceduru i zamenjuje destinacionu IP adresu i port paketa parametrima iz NAT tabele, tj. privatnom adresom računara A i njegovim portom. Paket se rutira po ruting tabeli, tj. prosleđuje u privatnu mrežu.

NAPT je omogućio da privatne mreže imaju skoro neograničen pristup javnim internet resursima. Svaki računar (osim ako je firewallom drugačije konfigurisano) može imati dinamički pristup javnoj IP adresi, i to samo privremeno. Dovoljna je samo jedna javna IP adresa da ceo LAN, bez obzira na njegovu veličinu, može da pristupi internetu. To rešenje je značajno usporilo alokaciju javnih IPv4 adresa i privremeno odložilo hitnost rešavanja problema prelaska na bolji adresni plan.

NAPT može da translira 65536 konekcija u jednom trenutku (TCP/UDP portovi su 16-bitni brojevi koji se koriste u procesu translacije) putem jedne javne IP adrese. Naravno, NAPT je moguće podesiti i tako da translaciju istovremeno vrši na više javnih IP adresa, čime se povećava broj istovremenih konekcija.

NAT problemi

NAT servis na određen način izoluje LAN mrežu od ostatka interneta. U mrežama sa implementiranim NAT servisom pri adresiranju nije potrebno voditi računa o tome koje će uloge dobijati računari, već samo o korišćenju odgovarajućeg spektra adresa namenjenog LAN mrežama [27]. Ako se ipak ukaže potreba za primenom specifične politike (npr. u vezi sa načinom korišćenja LAN mreže), sve operacije vrše se samo na ruteru. Ovde se mora uzeti u obzir da dinamički NAT i NAPT, zbog svoje dinamičke prirode, praktično funkcionišu i kao *Statefull* firewall [29]. Zbog neperzistentnosti zapisa u NAT tabeli preslikavanja, NAPT predstavlja jednu vrstu jednosmernog firewalla, koji omogućava da korisnici iz lokalne mreže pristupaju spoljnim internet resursima, kao i da prime odgovor na svoje zahteve. Međutim, NAPT ne dopušta da prvi, inicijalni paket (npr. TCP SYN) prođe kroz ruter u lokalnu mrežu jer ne postoji adekvatan zapis u NAPT translaciji.

Ovo je pozitivna pojava zato što samo korišćenje NAPT servisa, pored svoje osnovne uloge, omogućava i osnovnu zaštitu LAN mreža od spoljnih napada. Za netehnološki orijentisane kompanije to predstavlja značajnu prednost jer automatski imaju neku bazičnu vrstu zaštite svoje LAN mreže.

S druge strane, jednosmerni firewall pravi mnogo problema u servisima koji se oslanjaju na konekciju tipa „s-kraja-na-kraj" (engl. *end to end*). Ti servisi očekuju da mogu da uspostave direktnu vezu između krajnjih računara, bez posrednika koji menja sadržaj paketa. Primer za to je i problem koji se javlja kod VoIP, tj. SIP/RTP komunikacije. Kada se u VoIP komunikaciji uspostavlja veza između dve strane, konekcija direktno od pozivaoca pokušava da pristupi računaru kojeg poziva kako bi se uspostavio komunikacioni kanal. Tada problem pravi NAT jer ne dopušta translaciju paketa sa interneta ukoliko prethodno ta konekcija nije potekla iz lokalne mreže. Postoji veći broj radova koji diskutuju o različitim načinima rešavanja ovog problema u VoIP okruženju.

Sličan problem, nastao usled jednosmernog firewalla, postoji i u primeni VPN tehnologije na L2 i L3 sloju. Očekivano je da se komunikacioni kanal zbog bezbednosti uspostavi isključivo „s-kraja-na-kraj" i da ne sme da postoji posrednik koji menja pakete.

Za rešenje pomenutih problema najčešće se koristi jedna od sledeće dve tehnike:

- Statički dNAT – *Port Forwarding* – koji dodaje permanentni zapis u NAT tabelu prosleđivanja, i na taj način omogućava da se za paket koji dolazi sa interneta uvek može obaviti translacija i da se on može prosleditи tačno određenom računaru u LAN mreži.
- Aplikacioni Gateway (engl. *Aplication Level Gateway* – ALG) – predstavlja Gateway/Proxy aplikaciju koja bi bila aktivna na ruteru sa NAT servisom i koja bi na specifičan način obradila i prosledila paket mimoilazeći pravila NAT procesa. ALG podrška se pravi za svaki aplikativni protokol posebno, pa je njegova rasprostranjenost relativno skromna.

Zbog visokog nivoa programabilnosti i proširene funkcionalnosti, NAT je u praksi najčešće implementiran u softveru rutera, tj. izvršava se procesorom, a ne zasebnim hardverom. Posledica toga je da NAT procesiranje dovodi do velikog kašnjenja u prenosu podataka. Zbog svoje softverske implementacije, a i sve raznovrsnijom primenom interneta (pre svega korišćenja *peer-to-peer* sistema), NAT je sve češće „usko grlo" u komunikaciji.

Sve to, kao i gotovo iscrpljenost raspoloživih javnih IPv4 adresa u određenim regionima sveta, čini da se ponovo forsira hitno rešavanje problema dostupnosti javnih

adresa. Već u ranim fazama rešavanja problema adresnog prostora (i drugih) nastao je protokol IPv6. Do danas je ovaj protokol imao nekoliko modifikacija, a poslednja verzija je dostupna putem standarda [30]. Prelazak na IPv6 je nešto što se mora desiti kako bismo rešili problem javne dostupnosti, ali se sa realizacijom već poprilično kasni. IPv6 predviđa dovoljno veliki adresni prostor tako da ne samo sadašnje LAN mreže već i bilo koje buduće mreže mogu biti adresirane javnim IPv6 adresama koje bi bile globalno dostupne. Više praktično neće biti potrebno koristiti NAT u svrhu translacije privatnih u javne IP adrese.

Sa IPv6 protokolom NAT će postati suvišan, čime će se smanjiti vreme procesiranja na ruteru, tj. ruter će se baviti samo rutiranjem.

Softver Defined Networks (SDN)

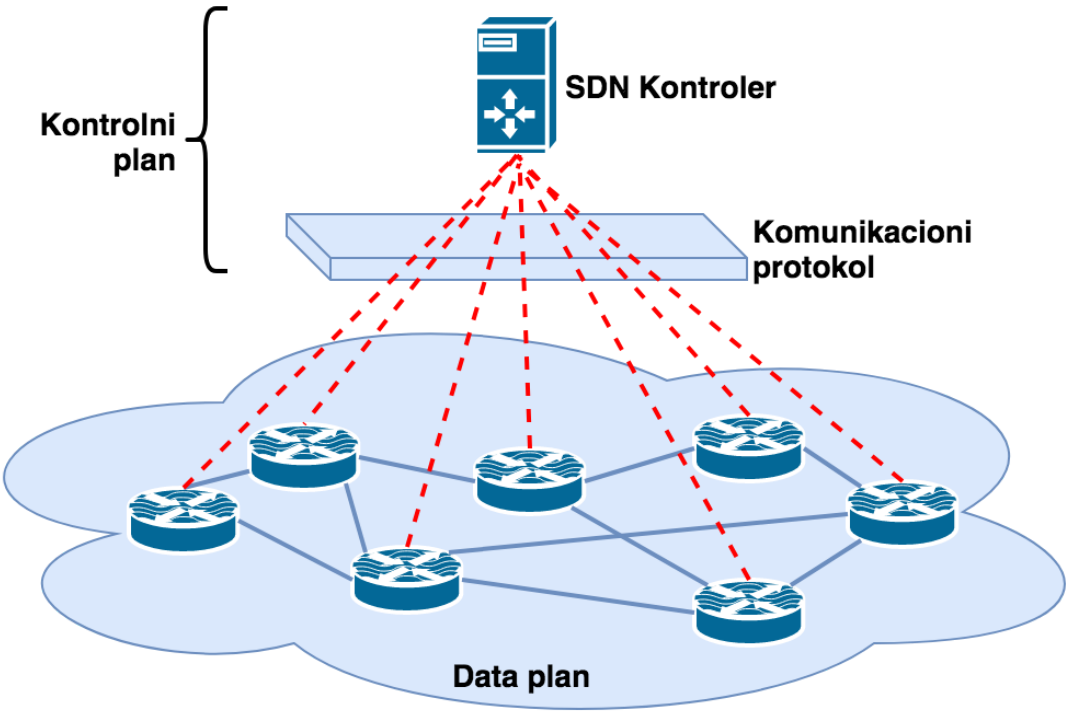
Standardizacija različitih tehnologija i rešenja izvršava se putem jedne od dveju organizacija nadležnih za objavu standarda IEEE i IETF, i neophodna je da bi se precizno definisao skup pravila kojima se uređuje određen proces u mreži i time omogućila kolaboracija opreme različitih proizvođača. Taj problem je u velikoj meri eskalirao, tj. dobio na značaju u oblasti računarskih mreža, jer je umnogome otežao primenu novih tehnologija i njihovih rešenja. Očigledan primer je implementacija VoIP servisa kod koje je bio izražen problem kolaboracije između opreme različitih proizvođača, pogotovo u domenu komunikacije preko SIP protokola (engl. *Session Initiation Protocol*). Detaljno je opisan u radu [31] koji za predmet istraživanja ima interkonekciju VoIP uređaja različitih proizovđača. Svaki proizvođač je „ispoštovao" standard na svoj način, što je nametalo potrebu pronalaženja rešenja koje bi moglo da usaglasi različita tumačenja standarda VoIP komunikacije. Dakle, jasno se može zaključiti da problem standardizacije, koji se ogleda u postojanju brojnih dokumenata u kojima se opisuju različite procedure, i manjak kolaboracije između opreme različitih proizvođača, predstavlja problem u računarskim mrežama sa tradicionalnom arhitekturom koji otežava, a ponekad i onemogućava, primenu novih, efikasnijih i pametnijih servisa. Primena ovih servisa jednostavno ne može da sačeka da standard „sazri", jer za to nema vremena u eri brzog tehnološkog napretka.

Implementacija naprednih servisa iziskuje da se u računarskim mrežama primene dinamičnija, fleksibilnija rešenja za brže i efikasnije rešavanje određenih mrežnih problema.

Zato je neophodno da se iz oblasti razvoja hardvera (koji nije baš podložan lakoj izmeni radi unapređenja), pređe na oblast unapređenja softvera gde su mogućnosti za pronalaženje rešenja veće (putem patch ili softverske nadogradnje lako se može dodati ili primeniti neka nova funkcionalnost, i uz znatno manje troškove). U naučnoj i stručnoj literaturi postoje brojni radovi u kojima istraživači pokušavaju da pronađu odgovor, odnosno rešenje za razne izazove u oblasti mrežnih komunikacija kao što su: rutiranje saobraćaja, mobilnost u mreži, brzina svičovanja, veća bezbednost i sl. Sa sigurnošću se može reći da je većina tih rešenja bolja od postojećih, standardom propisanih procedura. Međutim, retko je koje od njih pretočeno u standard, a još ređe primenjeno u praksi na hardverskim uređajima. Razlog za to je prisustvo standarda koji su zaživeli i široko su rasprostranjeni, čija svaka promena izaziva veliku diskusiju o njihovim prednostima i nedostacima. Pritom treba uzeti u obzir da svaka promena postojećeg standarda zahteva da novo rešenje bude kompatibilno sa prethodnim, što u nekim slučajevima nije moguće sprovesti. Novi standardi, koji treba da zamene starije, moraju biti znatno bolji da bi opravdali zamenu. Pošto se u praksi to retko sreće, najčešće se čeka da se pronađe više rešenja za neke probleme, pa se tek onda objavljuje novi standard sa akumuliranim rešenjima različitih problema (primer je niz nadogradnji koje donosi IPv6 u odnosu na IPv4 protokol).

Zbog potrebe da se ubrza rešavanje određenih mrežnih problema, uvođenjem većeg stepena programabilnosti u računarske mreže, nastao je koncept softverski definisanih mreža (engl. *Software Defined Networking* – SDN). On se zasniva na unošenju veće dinamike i primeni fleksibilnijih rešenja, na svemu onome što nudi razvoj softvera. S druge strane, SDN se oslanja na hardver u onom delu koji treba da omogući visoke performanse.

Koncept SDN arhitekture računarskih mreža zasniva se na primeni tehnologije kojom se kontrolna ravan (engl. *control plane*) razdvaja od ravni za prosleđivanje podataka (engl. *data plan*e).



Slika 3.6. SDN koncept

Kontrolna ravan je zadužena za upravljanje računarskom mrežom i sastoji se od specijalizovanog protokola kojim komuniciraju SDN sposobni uređaji. Njom se definiše način rada neke računarske mreže, a u okviru koncepta SDN mreže ona je realizovana softverski kako bi se omogućila visoka fleksibilnost upravljanja. Za razliku od nje, u ravni za prosleđivanje podataka primenjen je mehanizam za prosleđivanje samih paketa u SDN mreži. U njemu SDN uređaji koriste instrukcije postavljene na kontrolnoj ravni za prosleđivanje i potrebnu manipulaciju određenim paketom. Da bi se postigle maksimalne performanse u prosleđivanju saobraćaja, ova ravan je realizovana hardverski.

Glavne komponente svake SDN mreže su:

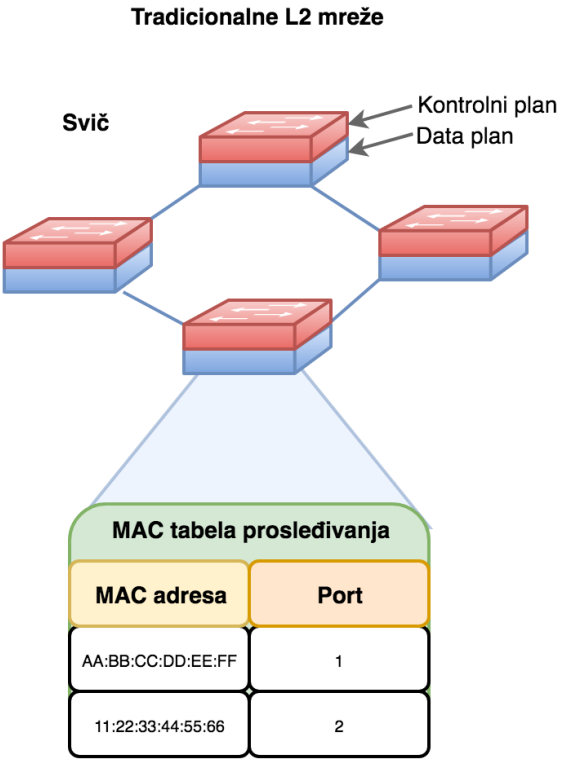
- SDN sposobni uređaji (SDN svič) – uređaj koji prosleđuje paket, tj. procesira data plan, i
- SDN kontroler – uređaj koji SDN svičevima daje instrukcije o načinu na koji treba proslediti konkretni paket.

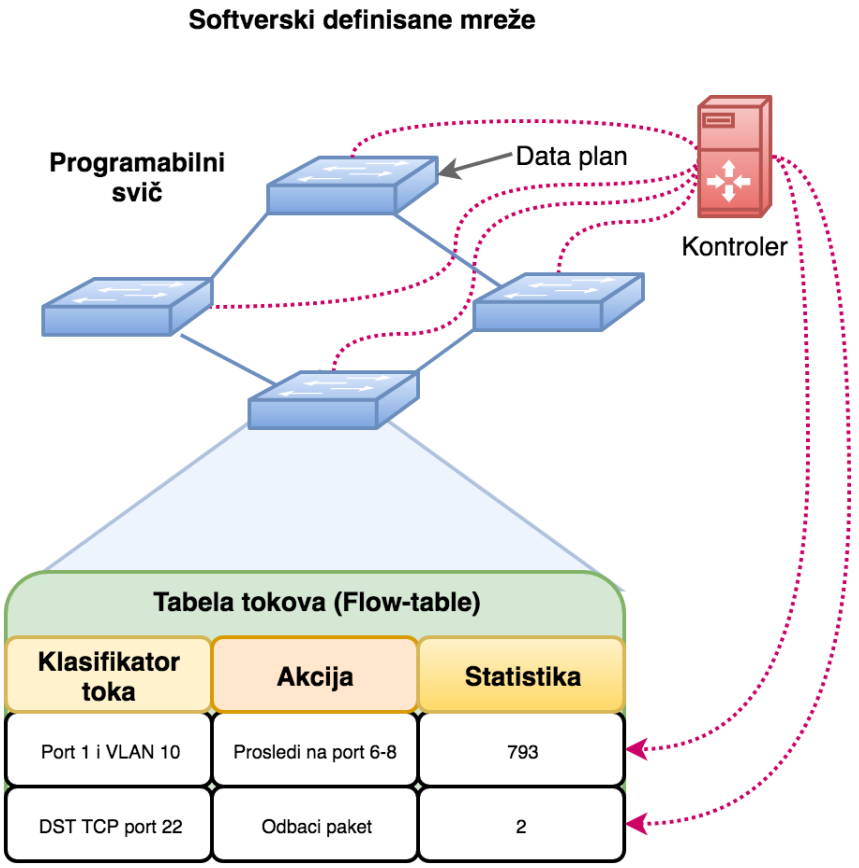
Data plan

SDN koncept mreža predviđa maksimalnu redukciju funkcija SDN uređaja na data-link (L2) sloju TCP/IP modela. Zbog jednostavnosti, uređaj koji vrši prosleđivanje u ovoj ravni često se naziva SDN sposoban uređaj ili SDN svič. Odlikuju ga sledeće funkcije:

- poseduje tabelu tokova (engl. *Flow table*) sa instrukcijama kako treba tretirati određeni paket,
- prosleđuje ili modifikuje paket na osnovu instrukcije iz tabele tokova,
- uspostavlja vezu sa SDN kontrolerom kako bi slao i primao instrukcije, i
- ažurira tabelu tokova prema instrukcijama dobijenim od kontrolera.

S obzirom na to što se osnovne funkcije u ovoj ravni izvršavaju na L2 sloju TCP/IP modela, komunikacija između dve MAC adrese (ili bilo kojih drugih adresa na L2 sloju, ako se ne koriste Ethernet bazirani porotokoli) predstavlja jedan *Flow,* tj. tok.





Slika 3.7. SDN data plan sa tabelom tokova

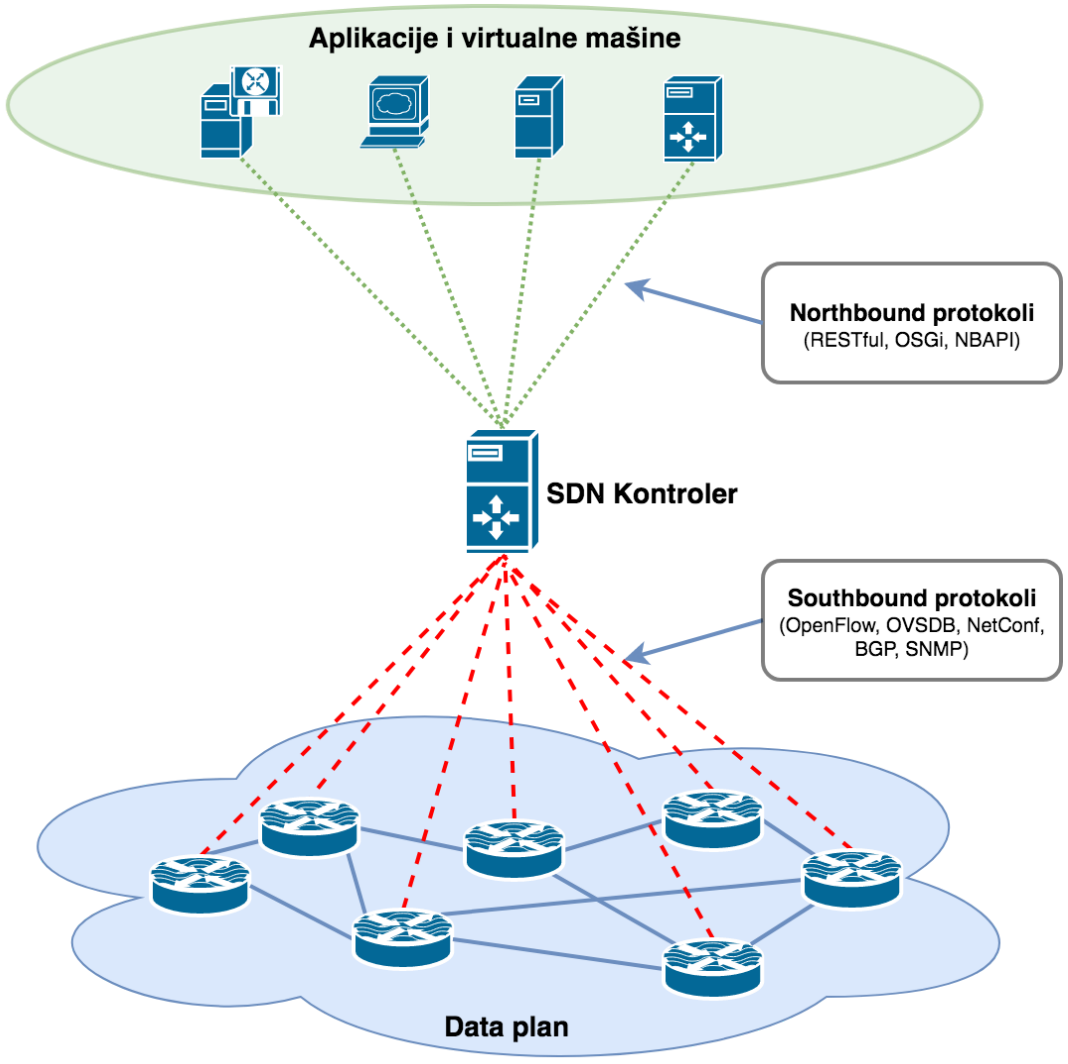
Važno je naglasiti da SDN svičevi ne treba da ponavljaju klasične procedure Ethernet prosleđivanja (L2 forwarding) niti učenja MAC adresa, već da se za realizaciju svojih funkcija oslone isključivo na SDN kontroler i tabelu tokova. Na taj način se primenom SDN koncepta omogućava korišćenje jednostavnih i jeftinih hardverskih uređaja sa visokim performansama prosleđivanja u računarskim mrežama. Trenutno stanje na ovom tržištu je takvo da se tek narednih godina očekuje pojava većeg broja pristupačnih SDN uređaja.

Ovde je potrebno ukazati na to da je primarni zadatak SDN sviča da na osnovu tabele tokova prosleđuje paket. Isti zadatak trenutno obavljaju i klasični svičevi, koji na osnovu MAC tabele prosleđivanja donose odluku o načinu procesiranja paketa. Suštinska razlika je u tome što je kod klasičnih svičeva postupak ažuriranja MAC tabele – poznat kao postupak „učenja" – prilično strog i statičan jer je definisan standardom, dok je tabela tokova kod SDN svičeva podložna promeni u skladu sa bilo kojom instrukcijom sa kontrolera. Jednostavno, u ravni za prosleđivanje paketa, SDN svičevi prosleđuju paket brzinom hardvera, a kontrolu mesta i načina prosleđivanja paketa definiše SDN kontroler.

Kontrolni plan

Kontrolna ravan je u okviru SDN koncepta zadužena za softversko upravljanje mrežnim tokovima. Takav pristup omogućava implementaciju različitih procedura koje se odnose na to kako se može uticati na rad mreže. Na ovoj ravni definisana su dva API-a:

- *Southbound interface* – koristi se u komunikaciji SDN svičeva sa SDN kontrolerima, i
- *Northbound interface* – koristi se između SDN kontrolera i drugih procesa koji utiču na rad mreže.



Slika 3.8. Southbound i Northbound protokoli (API)

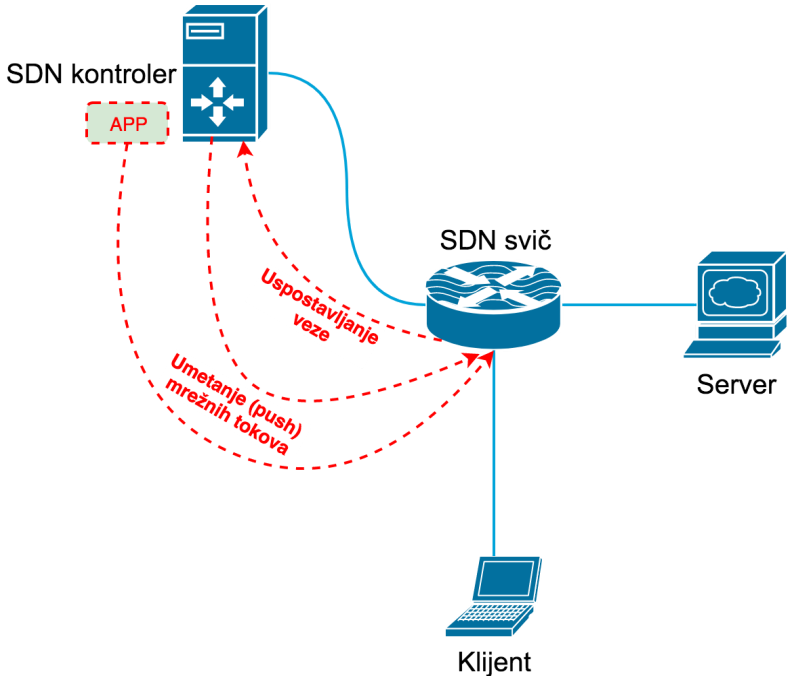
U okviru Northbound interfejsa implementirani su uglavnom tzv. *proprietary* protokoli koji se koriste za komunikaciju između SDN kontrolera i softvera proizvođača SDN kontrolerskog softvera. Oni mogu koristiti otvorene interfejse poput JSON-a, REST-a i sl.

kao mehanizme za razgovor sa kontrolerom (u *cloud* sistemima Northbound interfejs se koristi za kreiranje virtualnih mreža između virtualnih mašina). S druge strane, u okviru Southband interfejsa implementirani su bolje uređeni protokoli koji su definisani u brojnim naučnim radovima i standardizovani. U radu [32] opisan je najvažnijii protokol u okviru ovog interfejsa – OpenFlow protokol, koji se još razvija (aktuelna je verzija 1.5.1), što je glavni razlog za odsustvo masovne hardverske podrške. Ovaj protokol se koristi u komunikaciji između SDN sviča i SDN kontrolera kako bi se dobile instrukcije za obradu tokova, i detaljnije će biti opisan u narednom poglavlju.

Konceptom softverski definisanih mreža definisana su dva načina na koje se mogu ažurirati podaci u tabeli tokova:

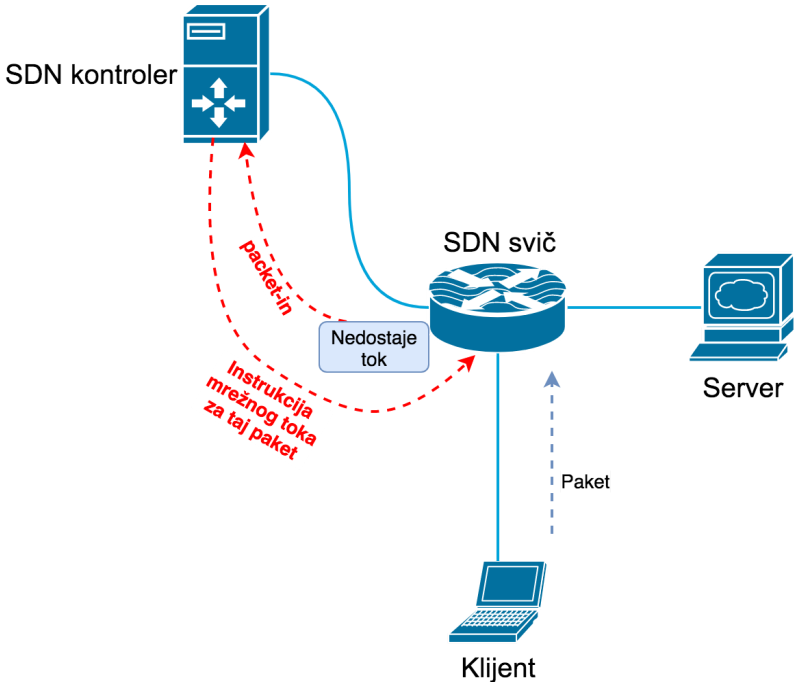
- proaktivni režim i
- reaktivni režim.

Proaktivni režim podrazumeva da se SDN svič priključuje (konektuje) na SDN kontroler i da odmah dobija sve potrebne tokove kojima može da vrši prosleđivanje. Tada se SDN svič ponaša slično kao tradicionalni L2 svič, tj. poseduje sve tokove u tabeli prosleđivanja. SDN kontroler može, na osnovu toga što ima podatke o celokupnoj mreži i svest o najoptimalnijim putanjama, da unapred pošalje sve najoptimalnije tokove do svakog novog SDN sviča. Tako se u mrežu implementira centralizovana logika, koja se ogleda u tome što je SDN kontroler svestan cele slike na svakom pojedinačnom SDN sviču. Ono na šta posebno treba obratiti pažnja je potreba da se tabela tokova ne „prepunjava” (svako prosleđivanje zahteva pretragu zapisa u tabeli, što sa većim brojem zapisa unosi dodatno kašnjenje u obradi). Proaktivno umetanje mrežnih tokova moguće je takođe inicirati nekom aplikacijom unutar SDN kontrolera ili preko protokola Northbound interfejsa. Tada SDN kontroler šalje novi tok SDN sviču, bez prethodne komunikacije.



Slika 3.9. Proaktivno umetanje mrežnih tokova

Reaktivni režim podrazumeva drugačiju situaciju u kojoj za paket koji stiže na SDN svič ne postoji zapis u tabeli tokova. U toj situaciji, SDN svič prosleđuje primerak paketa SDN kontroleru na dalju analizu. SDN kontroler poseduje mogućnost da na osnovu primljenog paketa izvršiti samostalnu analizu ili da čak, preko protokola Northbound interfejsa, konsultuje druge procese. Nakon analize, kontroler formira instrukciju kojom se definiše šta treba uraditi sa tokom, i prosleđuje je odgovarajućem SDN sviču. SDN svič ažurira pristiglu instrukciju u tabeli tokova i procesira paket prema aktuelnom stanju. Svaki budući paket koji bude deo istog toka biće prosleđen direktno sa sviča, bez ponovne konsultacije sa SDN kontrolerom. Inicijalno prosleđivanje kontroleru i čekanje na njegov odgovor unosi malo kašnjenje prilikom prosleđivanja prvog paketa toka. Ipak, to smanjuje potrebu da se u tabeli tokova čuvaju oni tokovi koji se ne koriste aktivno, čime se ubrzava procesiranje svih drugih paketa aktivnih tokova.



Slika 3.10. Reaktivno umetanje mrežnih tokova

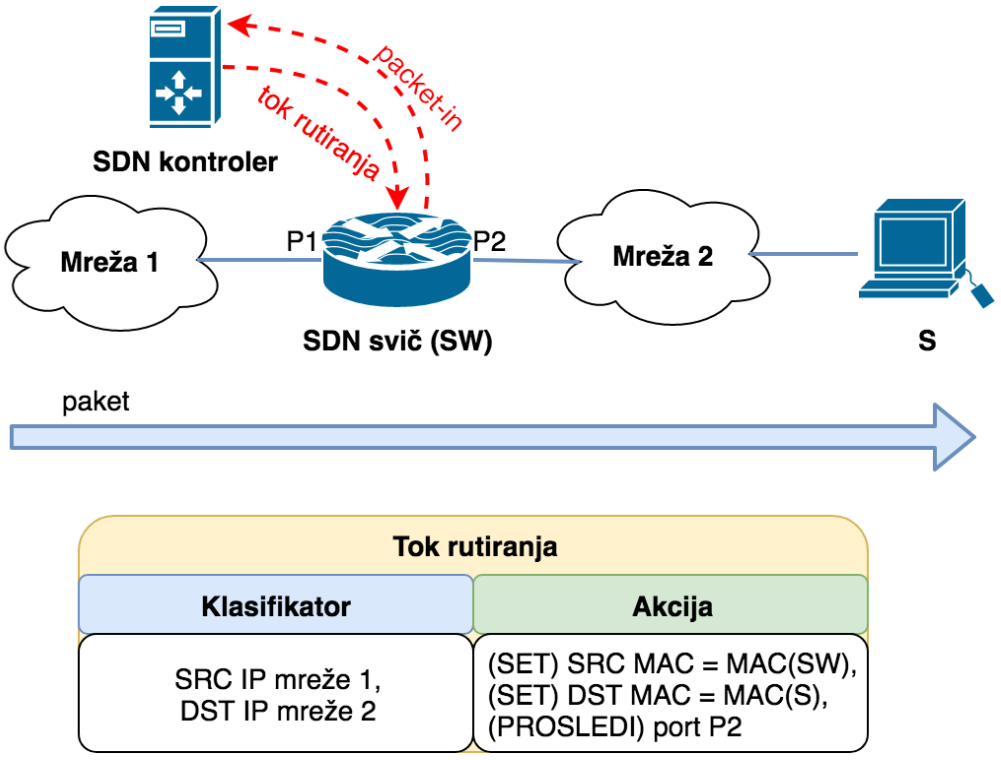
Na SDN kontroleru se zapravo implementiraju sve osnovne i dodatne funkcionalnosti neke SDN mreže, kao što su:

- kontrola mrežnog pristupa na L2/L3/L4/L5 slojevima TCP/IP modela,
- L2 svičovanje uz SPF optimizaciju celokupne mreže,
- L3 rutiranje uz SPF optimizaciju celokupne mreže,
- NAT,
- dinamički Queuing i alokacija propusnog opsega,
- nadgledanje mreže – Network ussage and monitoring,
- skladištenje mrežnih tokova u forenzičke svrhe,
- dinamička pristupačnost VLAN segmentima, i
- uređivanje multicast tokova.

Dakle, primenom SDN koncepta, svaka funkcionalnost može da se realizuje softverski na strani SDN kontrolera i da se fizički primeni preko SDN svičeva (funkcionalnost se definiše putem programskog koda na kontroleru i proizvodi u flow instrukciju). Sam kontroler ne menja paket, već šalje instrukciju o izmeni do SDN sviča. SDN svič treba, pored svoje osnovne uloge prosleđivanja, da bude u stanju da izvrši i modifikaciju prema jednostavnoj instrukciji kontrolera.

Primer takve instrukcije je, recimo, modifikacija ethernet headera tako da u polje izvorišne i destinacione MAC adrese unese tačno određene parametre prema instrukciji kontrolera. Time SDN svič praktično vrši deo procesa rutiranja, a da pri tome nije imao ruting tabelu niti je morao da zna bilo šta o mrežnom sloju.

Takva realizacija omogućava da SDN svič vrši prosleđivanje, tj. rutiranje svojom hardverskom brzinom, a da se kontrolom rutiranja bavi SDN kontroler.



Slika 3.11. SDN rutiranje

OpenFlow

OpenFlow protokol je prvi put predstavljen u okviru istraživanja objavljenog u radu [32] sredinom 2008. godine. Dalji rad na razvoju ovog protokola i implementaciji rešenja kojim se kontrolna (upravljačka) ravan razdvaja od puta za prosleđivanje podataka (engl. *datapath*), nastavljen je pod okriljem organizacije *Open Networking Fundation* (ONF) (organizacija je deo organizacije *Linux Fundation*). ONF projekat standardizacije i uređivanja OpenFlow protokola vodi se pod nazivom *Open Datapath* [33].

Prva verzija OpenFlow protokola sa konkretnom specifikacijom kako on treba da funkcioniše objavljena je 31.12.2009. godine pod nazivom *OpenFlow V1.0* [34]. Ovim standardom nije samo specifikovana procedura odvajanja kontrolne ravni i ravni za prosleđivanje podataka, već su definisani i pojmovi u vezi sa mrežnim tokovima (engl. *flows)* kao entitetima mrežne komunikacije. U tom smislu, OpenFlow svičevi (SDN svičevi) su klasifikovani kao *datapath* mrežni uređaji koji imaju zadatak samo da fizički prosleđuju podatke (pakete) na osnovu instrukcija sadržanih u tabelama mrežnih tokova. Primer tabele mrežnog toka dat je na slici 3.7, na kojoj se vidi da se ova tabela sastoji od:

- zaglavlja (engl. *header*) – kriterijumi za selekciju paketa (engl. *packet matching*),
- brojača – statistika za svaki selektovan paket, i
- akcije – instrukcija u pogledu radnje koju treba izvršiti sa selektovanim paketom.

U okviru zaglavlja tabele mrežnih tokova nalaze se polja čije vrednosti predstavljaju kirterijume u skladu sa kojima se može izabrati i drugačije obraditi neki paket. Ta polja sadrže:

- ulazni port na sviču (engl. *Ingress Port*),
- izvorišnu MAC adresu (engl. *Source address*),
- destinacionu MAC adresu (engl. *Destination MAC address*),
- tip protokola na datalink sloju,
- VLAN ID (engl. *Virtual Local Area Identifier),*
- VLAN prioritet (engl. *Virtual Local Area Priority),*
- izvorišnu IP adresu (engl. *Source IP address),*
- destinacionu IP adresu (engl. *Destination IP address*),
- tip protokola mrežnog sloja,
- IP QoS bitove (ToS, DiffServ),
- izvorišni port (engl. *Source port*) transportnog sloja,
- destinacioni port (engl. *Destination port*) transportnog sloja,
- tip ICMP (engl. *Internet Control Message Protocol*) poruke, i
- ICMP kod.

Vrednost bilo kojeg od pomenutih polja može se koristiti kao kriterijum u procesu mečovanja paketa sa tabelom mrežnih tokova. Moguće je korisiti i više kriterijuma u jednoj instrukciji kada se primenjuje operator „i” (engl. *and*), pa je potrebno da svi kriterijumi budu zadovoljeni kako bi se koristila instrukcija tog mrežnog toka.

OpenFlow V1.0 takođe propisuje i brojače za svaki tok tako da se može dobiti pregledna statistika. Minimalna konfiguracija predviđa postojanje:

- brojača po tabeli koji sadrže:
    - broj aktivnih tokova,
    - broj pregleda paketa, i
    - broj paketa koji zadovoljavaju neki od kriterijuma (packet matches);
- brojača za pojedinačne tokove sa informacijama o:
    - broju primljenih paketa,
    - broju primljenih bajtova,
    - dužini trajanja toka u sekundama, i
    - dužini trajanja toka u nanosekundama;
- brojača za svaki port OpenFlow sviča sa podacima o:
    - broju primljenih paketa,
    - broju poslatih paketa,
    - broju primljenih bajtova,
    - broju poslatih bajtova,
    - broju izgubljenih paketa na prijemu,
    - broju izgubljenih paketa pri slanju,
    - broju primljenih paketa sa greškom,
    - broju poslatih paketa pristiglih sa greškom,
    - broju primljenih paketa sa greškom u poravnanju (engl. *Frame alignment error*),
    - broju primljenih prevelikih paketa (engl. *Overrun error*),
    - broju primljenih paketa sa CRC greškama, i
    - broju kolizija;
- brojača za redove na OpenFlow svičevima koj sadrže:
    - broj poslatih paketa,
    - broj poslatih bajtova, i
    - broj grešaka nastalih zbog prepunjavanja (engl. *Overrun error*).

OpenFlow V1.0 standardom definisan je veliki broj akcija koje bi OpenFlow svič trebalo da podrži. Mora se međutim naglasiti da svaka akcija nije obavezna u svakoj

implementaciji i da postoje akcije koje su markirane kao obavezne i one koje su markirane kao opcione.

Danas postoje dva modela implementacije OpenFlow protokola. Prvi model se odnosi na proces razvoja namenskog hardvera/softvera koji će funkcionisati isključivo kao OpenFlow svič i koji će se za svaku operaciju obraćati SDB kontroleru. Drugi model se odnosi na slučaj implementiranja OpenFlow protokola kao dodatne funkcionalnosti postojeće opreme koja već poseduje određen hardver/softver.

OpenFlow implementacijom definisane su sledeće akcije:

- **FORWARD – obavezne:**
  - **ALL** – slanje paketa na sve interfejse osim na onaj s kog je paket došao,
  - **CONTROLLER** – paket se enkapsulira i prosleđuje kontroleru na dalju analizu,
  - **LOCAL** – paket se predaje na lokalno sviching procesiranje, i
  - **IN_PORT** – paket se šalje preko istog porta sa kojeg je pristigao;
- **ENQUEUE** – **opciona** – prosledi paket na red koji je povezan sa izlaznim portom;
- **DROP** – **obavezna** – odbaci paket; i
- **MODIFY-FIELD** – **opciona** – uvodi mogućnost da OpenFlow svič izvrši jednostavnu modifikaciju određenog polja hedera paketa.

Interesantno je istaći da su implementacijom OpenFlow protokola kao dodatne funkcionalnosti na postojećoj opremi podržane akcije:

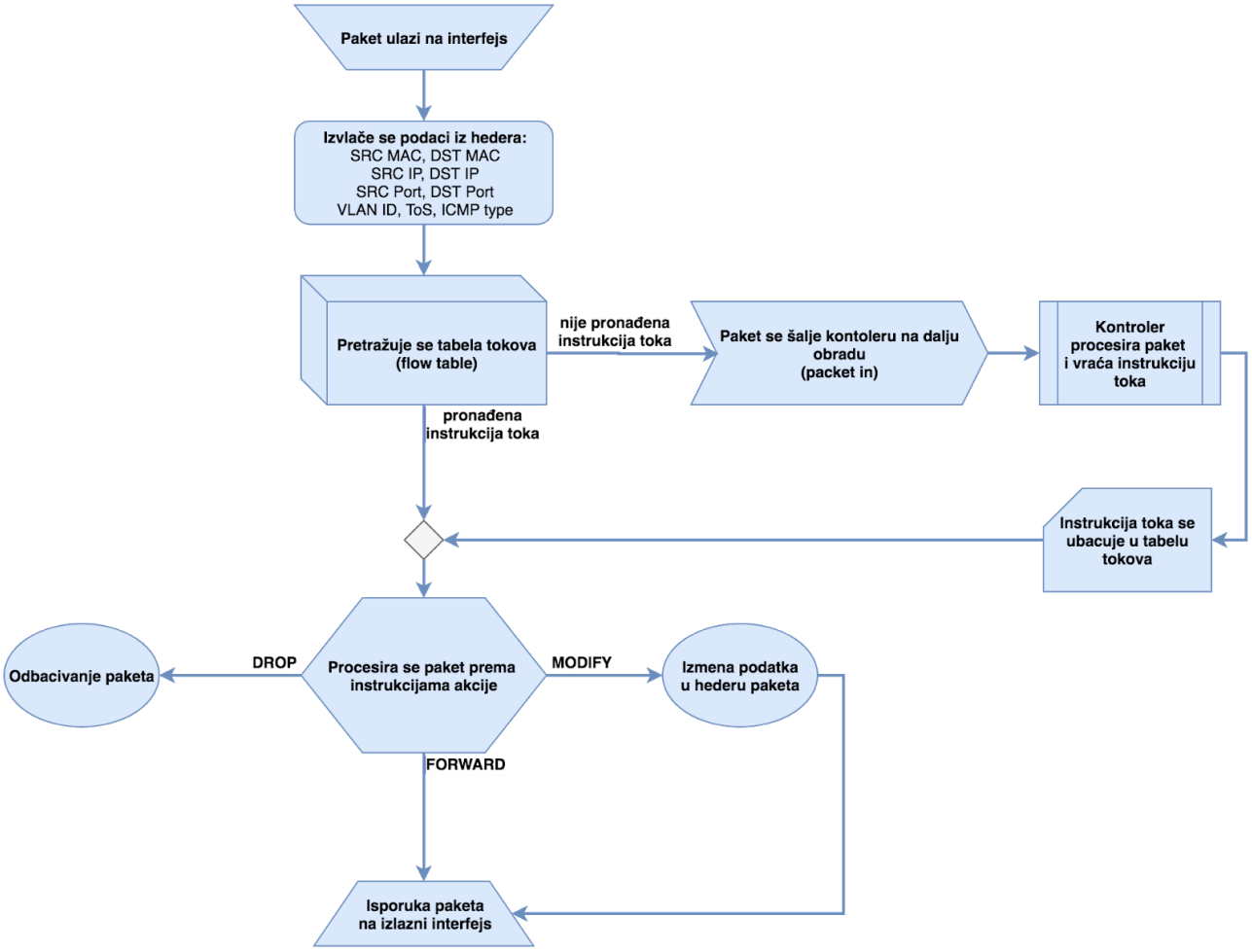
- **NORMAL** – realizacijom ove akcije, kontrola nad paketom vraća se OpenFlow sviču koji sprovodi tradicionalnu obradu paketa, a specijalno za svičeve sa OpenFlow podrškom; i
- **FLOOD** – paket se šalje na sve izlazne interfejse (sa ulaznog).

Pomenute akcije kod tzv. *OpenFlow* svičeva, u kojim realizacija akcije zavisi od instrukcije kontrolera, nemaju mnogo značaja jer ne sadrže konkretnu instrukciju.

Modifikacijom pojedinih polja stvaraju se mogućnosti za primenu naprednih funkcionalnosti na jednostavnim OpenFlow svičevima. Ove modifikacije uključuju:

- promenu VLAN ID paketa (čak i kreiranje VLAN hedera ako ne postoji),
- promenu VLAN prioriteta,
- uklanjanje VLAN hedera,
- promenu izvorne MAC adrese,
- promenu destinacione MAC adrese,

- promenu izvorne IP adrese,
- promenu destinacione IP adrese,
- promenu ToS (DiffServ) parametara,
- promenu izvornog porta transportnog sloja i
- promenu destinacionog porta transportnog sloja.



Slika 3.12. Tok procesiranja paketa preko OpenFlow protokola

Ovde je veoma značajno precizno opisati proceduru koja treba da se sprovede kada paket pristigne na ulazni port OpenFlow sviča. Ona se sastoji iz sledećih koraka:

1. Paket se prihvata sa nekog od ulaznih portova;
2. Parsiraju se podaci zaglavlja paketa i izvlače parametri za dalju proveru;
3. Na osnovu određenih kriterijuma pretražuje se (engl. *lookup*) tabela mrežnih tokova kako bi se utvrdilo da li postoji poklapanje mrežnih tokova sa podacima unutar primljenog paketa;

4. Ukoliko se pronađe poklapanje sa nekim od mrežnih tokova, izvršava se adekvatna akcija;
5. Ako nema poklapanja između kriterijuma za pretragu i parametara iz paketa, prelazi se na proveru poklapanja sa sledećom instrukcijom u tabeli mrežnih tokova;
6. Ukoliko se nijedan kriterijum (zapis) iz tabele mrežnih tokova ne poklapa sa podacima iz paketa, paket se enkapsulira i prosleđuje kontroleru na dalju obradu.

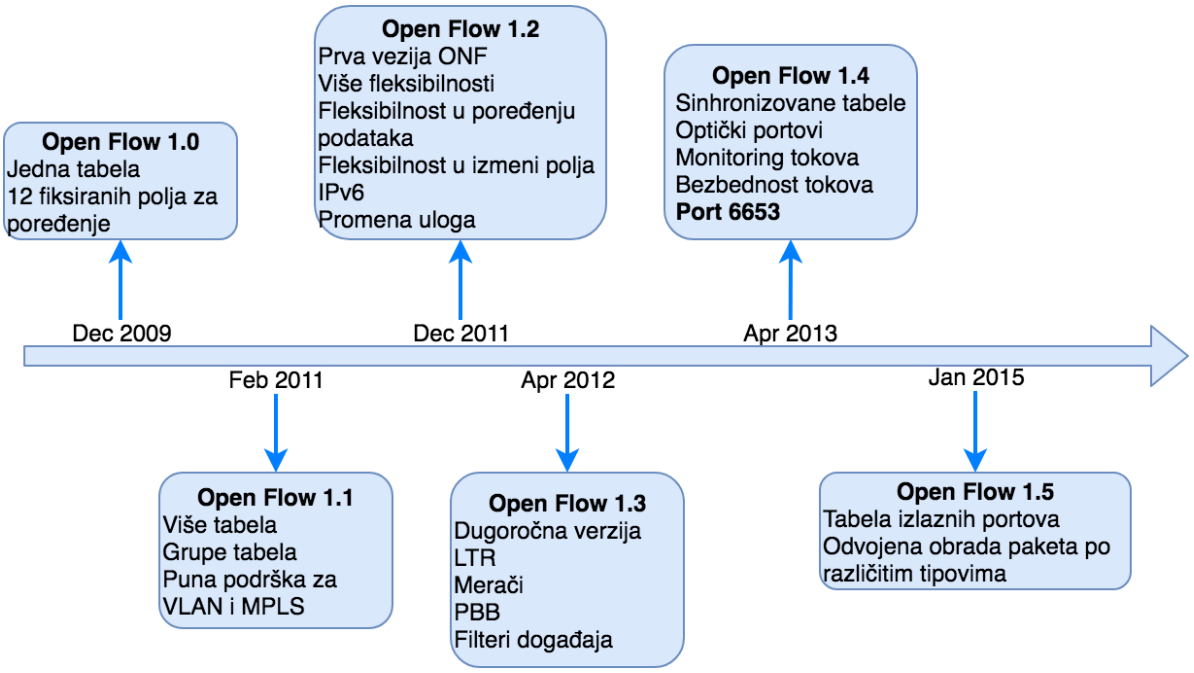
SDN kontroler ima zadatak da nakon prijema izvrši deenkapsulaciju paketa i da ga procesira prema sopstvenoj softverskoj logici, što uključuje i kreiranje odgovora sa instrukcijom toka i vraćanje paketa OpenFlow sviču koji mora da obradi paket prema novoj instrukciji.

Da bi se omogućilo normalno funkcionisanje OpenFlow uređaja u SDN mreži, neophodno je da se osigura bezbedna komunikacija između OpenFlow sviča i kontrolera. U tom smislu, primenjuje se komunikacija u skladu sa TLS standardom kako bi se garantovala visoka bezbednost. U cilju stvaranja uslova za maksimalnu bezbednost ove komunikacije, predviđeno je da i kontroler i svič poseduju svoj par RSA ključeva, te da u procesu uspostavljanja veze razmene sertifikate radi osiguranja autentičnosti.

Specifikacija OpenFlow 1.0 predviđa da se za komunikaciju koristi TCP protokol na portu 6633. Međutim, sredinom 2013. godine, organizacija zadužena za standardizaciju TCP i UDP portova IANA, dodelila je OpenFlow porotokolu portove 6653 za TCP i UDP komunikaciju [35]. Ova promena postaće deo OpenFlow specifikacije tek u njenim kasnjim verzijama.

OpenFlow V1.0 predstavlja samo početak razvoja protokola koji je pokretač SDN arhitekture. Modifikovan je putem velikih revizija pod brojevima V1.1, V1.2, V1.3, V1.4 i V1.5. Poslednja, najaktuelnija specifikacija OpenFlow protokola, objavljena je u septembru 2016. godine u verziji V1.5.1.

Na slici 3.13. dat je pregled najbitnijih promena u okviru različitih verzija OpenFlow protokola:



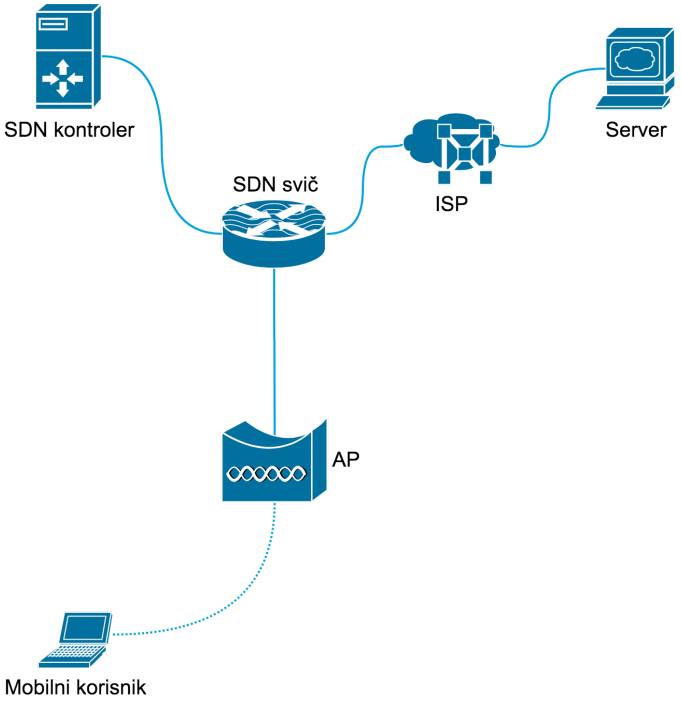
Slika 3.13. Pregled razvoja OpenFlow porotokola

Koncept novog rešenja za obezbeđivanje neprekidnosti IP sesije pri prolasku kroz heterogene bežične mreže

Koncept novog rešenja

U prethodnim poglavljima opisana su postojeća rešenja koja se primenjuju da bi se obezbedila neprekidnost servisa u heterogenim, bežičnim IP mrežama. Obrazloženi su nedostaci svakog rešenja ponaosob, ali su takođe naznačene i osnovne smernice u pogledu daljeg razvoja jednog novog i efikasnijeg koncepta za rešavanje ovog problema. Jasno je ukazano da osnovu novog koncepta mobilnosti u heterogenim bežičnim IP mrežama treba da čini SDN tehnologija, kao i da novo rešenje mora da bude relativno lako za implementaciju i bez mnogo materijalnih/finansijskih ulaganja u promenu tradicionalne arhitekture u bežičnim IP mreža. Drugim rečima, primena novog rešenja ne sme da zahteva izmenu tradicionalnog načina komunikacije između klijenta i servera, niti instalaciju bilo kakvog specifičnog softvera od strane mobilnog korisnika i servera (korespodenta). Zadatak

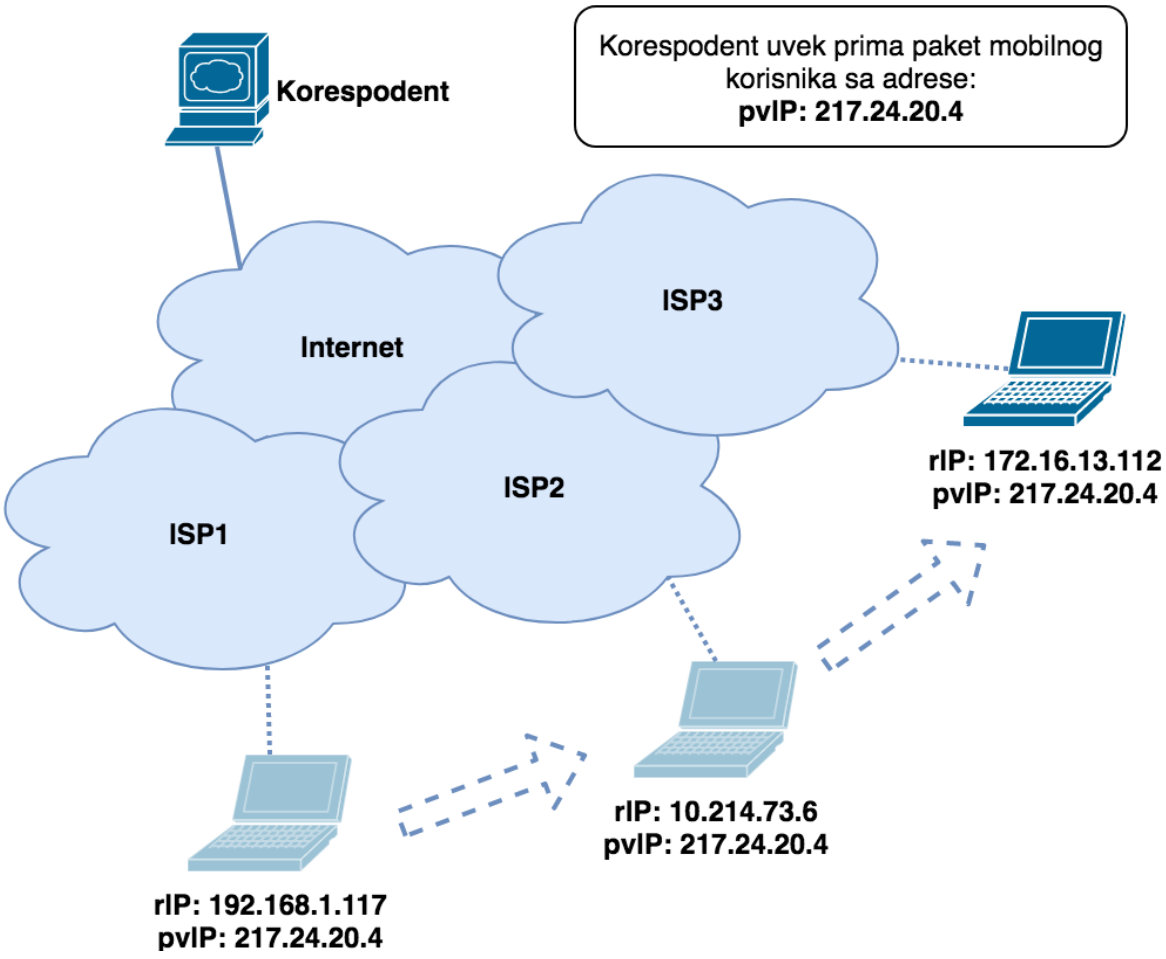
novog rešenja je da uz implementaciju minimalnih promena u mreži pružaoca servisa bežičnog pristupa omogući IP mobilnost. Imajuću u vidu ovako postavljen zadatak, predloženo rešenje je publikovano u međunarodnom časopisu [36], a u postupku je i zaštita patentnog prava [37].



Slika 4.1. Osnovni SDN koncept

Analizom postojećih rešenja za IP mobilnost lako se može doći do zaključka da je princip maskiranja IP adrese dobar put ka obezbeđenju potpune IP mobilnosti. U tom smislu, potrebno je uvesti koncept virtualne IP adrese koju treba dodeliti svakom mobilnom korisniku. Svaki mobilnik korisnik sa virtuelnom IP adresom mora biti vidljiv ostalom delu interneta, ali to ne znači da taj korisnik mora posedovati istu virtualnu IP adresu konfigurisanu na svojim interfejsima u svakoj mreži. Predlog novog rešenje je da mobilni korisnik poseduje:

- **realnu privremenu IP adresu** (engl. *real IP Address – rIP*) – IP adresa koju mobilni korisnik koristi u lokalnoj mreži bežičnog provajdera, i
- **permanentu virtualnu IP adresu** (engl. *permanent virtual IP Adress – pvIP*) – IP adresa kojom se mobilni korisnik predstavlja kada komunicira sa drugim korisnicima izvan domena mreže mobilnosti, tj. sa serverom na internetu.



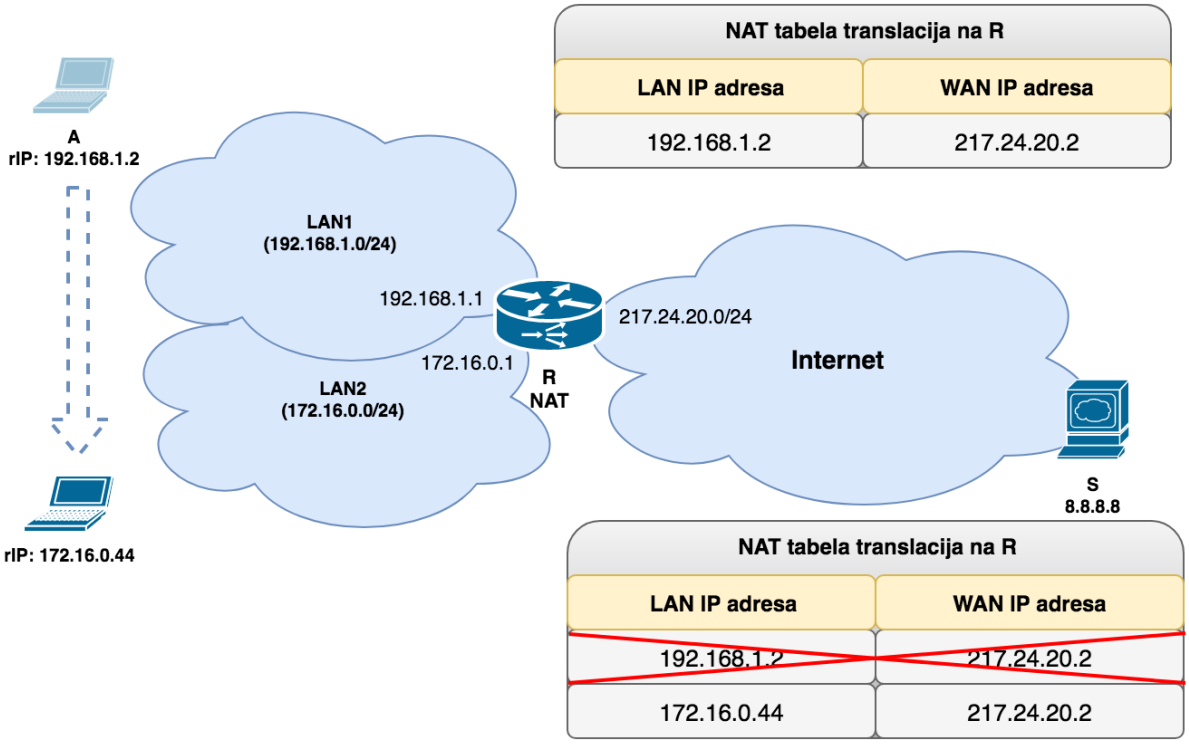
Slika 4.2. Razlika između rIP i pvIP adrese

Osnovna ideja je da mobilni korisnik dobija *rIP* adresu preko DHCP protokola (bilo da se koristi IPv4 ili IPv6) od strane provajdera bežične mreže. Adresa može biti čak i privatna IP adresa, iz opsega koji koristi provajder u čijoj se bežičnoj mreži trenutno nalazi korisnik. Dakle, upotrebom *rIP* adrese, mobilni korisnik može da komunicira sa drugim računarima u lokalnoj mreži provajdera (npr. sa lokalnim DNS serverom), ili da pakete prosleđuje ka ruteru koji ih šalje van mreže (*default gateway*). Važno je napomenuti da prelaskom u mrežu drugog provajdera mobilni korisnik mora da dobije drugu IP adresu (koja može biti privatna IP adresa) iz opsega te druge, gostujuće mreže. Ukoliko bi mobilni korisnik koristio samo *rIP* adresu, onda ne bi bilo moguće obezbediti njegovu potpunu IP mobilnost. Prelaskom u drugu mrežu, usled promene IP adrese, bile bi prekinute sve aktivne sesije tog mobilnog korisnika. Zato je neophodno da se uvede maskiranje *rIP* adrese nekom permanentom virtualnom IP adresom – *pvIP* adresom.

Sam proces maskiranja *rIP* adrese *pvIP* adresom izvršava se na ruteru koji predstavlja izlaznu tačku iz zajedničkog domena mobilnosti više bežičnih provajdera (analiza mesta u

mreži na kojem je moguće sprovoditi proces maskiranja adresa biće opisan dalje u tekstu). Predlog u vezi sa procesom maskiranja jeste da se u paketu mobilnog korisnika na izlasku iz bežične mreže jednog provajdera izvrši translacija u formi dinamičkog NAT-a njegove *rIP* u određenu *pvIP* adresu (*pvIP* adresa pojedinog korisnika predstavlja njegovu jedinstvenu javnu IP adresu sa kojom je dostupan spoljnim mrežama). Perzistentnim prevođenjem *rIP* adrese u *pvIP* adresu, korišćenjem NAT servisa, omogućava se da spoljnim serverima deluje kao da paketi uvek dolaze sa iste javne adrese. U suštini, ova procedura nije novina jer dinamički NAT servis već postupa na isti način kada mapira privatne u javne IP adrese i obrnuto.

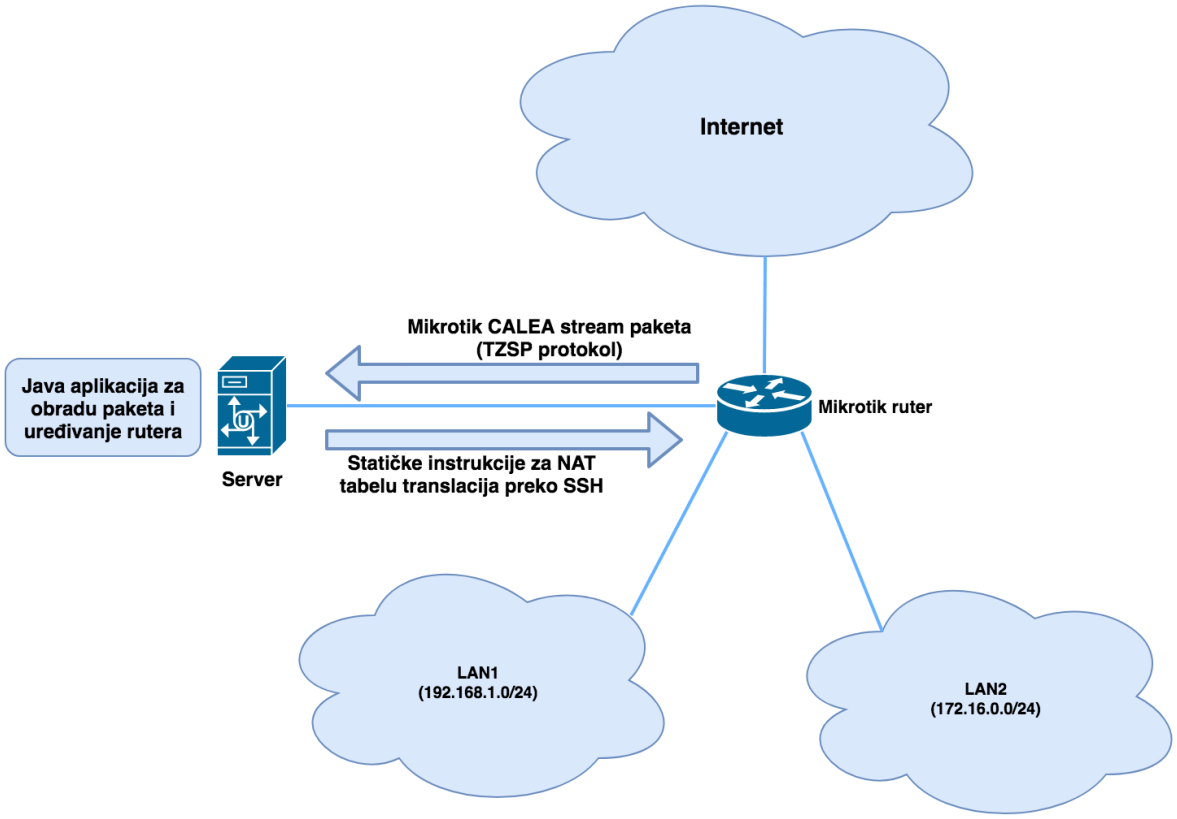
Važno je znati da se prelaskom mobilnog korisnika u drugu gostujuću mrežu menja *rIP* adresa korisnika koji istu treba da dobije od DHCP servera gostujuće mreže. Zato je neophodno da se definiše na koji način treba uticati na NAT servis i izmeniti postojeću translaciju u NAT tabeli. Potrebno je da se u NAT tabeli zapis stare *rIP* adrese zameni novom, gostujućom *rIP* adresom, ali i da se sačuva ista *pvIP* adresa. Imajući u vidu sva pravila NAT mehanizma, ovaj zadatak nije lako realizovati. Mora se uzeti u obzir činjenica da se kod dinamičkog NAT-a, dolaskom paketa za koji ne postoji zapis u NAT tabeli, alocira uvek nova spoljna IP adresa i dodaje novi zapis. Zato dinamički NAT ne rešava problem L3 mobilnosti.



Slika 4.3. Promena u NAT tabeli kada korisnik pređe u drugu mrežu

Da bi se pronašlo rešenje, mora se izraditi specifičan softver koji bi preko protokola za upravljanje (Telnet, SSH ili HTTPS JSPN API) omogućio spoljno dodavanje statičkih NAT translacija. Realizacijom takvog softvera bio bi onemogućen dinamički, a koristio bi se isključivo statički NAT i sve potrebne translacije dodavale bi se iz spoljnog servisa (slika 4.4).

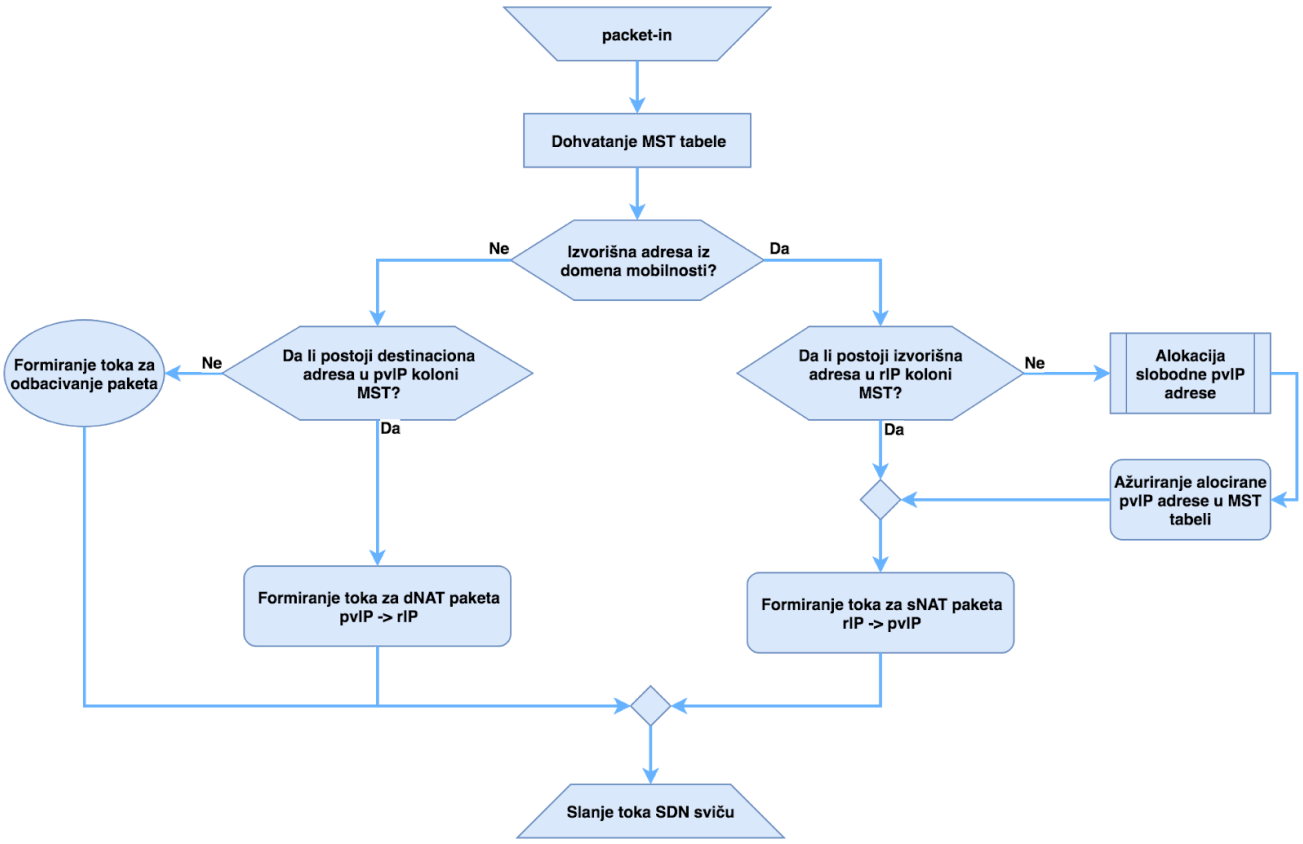
Realizaciju odgovarajućeg softvera opterećuje više problema. Najveći problem je to što bi ovakvu vrstu softverske podrške morao da poseduje svaki vendor mrežne opreme. Ovaj problem je dodatno uvećan činjenicom da svaki vendor koristi svoj komandni jezik za konfiguraciju uređaja i svoj sopstveni, a ne unificirani API, što značajno utiče na kompleksnost softverskog rešenja (potrebno je napraviti modularni sistem sa podrškom za opremu različitih vendora). Uz sve to, moraju se uzeti u obzir i eventualne promene u načinu konfiguracije prilikom sistemskih nadogradnji (softverske module je potrebno konstantno održavati i nadograđivat)i. Drugi bitan problem je to što se može desiti da prilikom dolaska prvog paketa korisnika nije ubačena nova instrukcija za translaciju jer je za to zadužen spoljni servis. Mora se imati u vidu da su kašnjenja u postavljanju instrukcija za translacije relativno česta. Naime, svičevi i ruteri znatno brže procesiraju pakete nego što bi to uradio neki servis na aplikacionom sloju, što za posledicu ima da se početni paketi ne mogu translirati, pa stoga bivaju odbačeni.



Slika 4.4. NAT eksperiment sa eksternim softverom za kontrolu

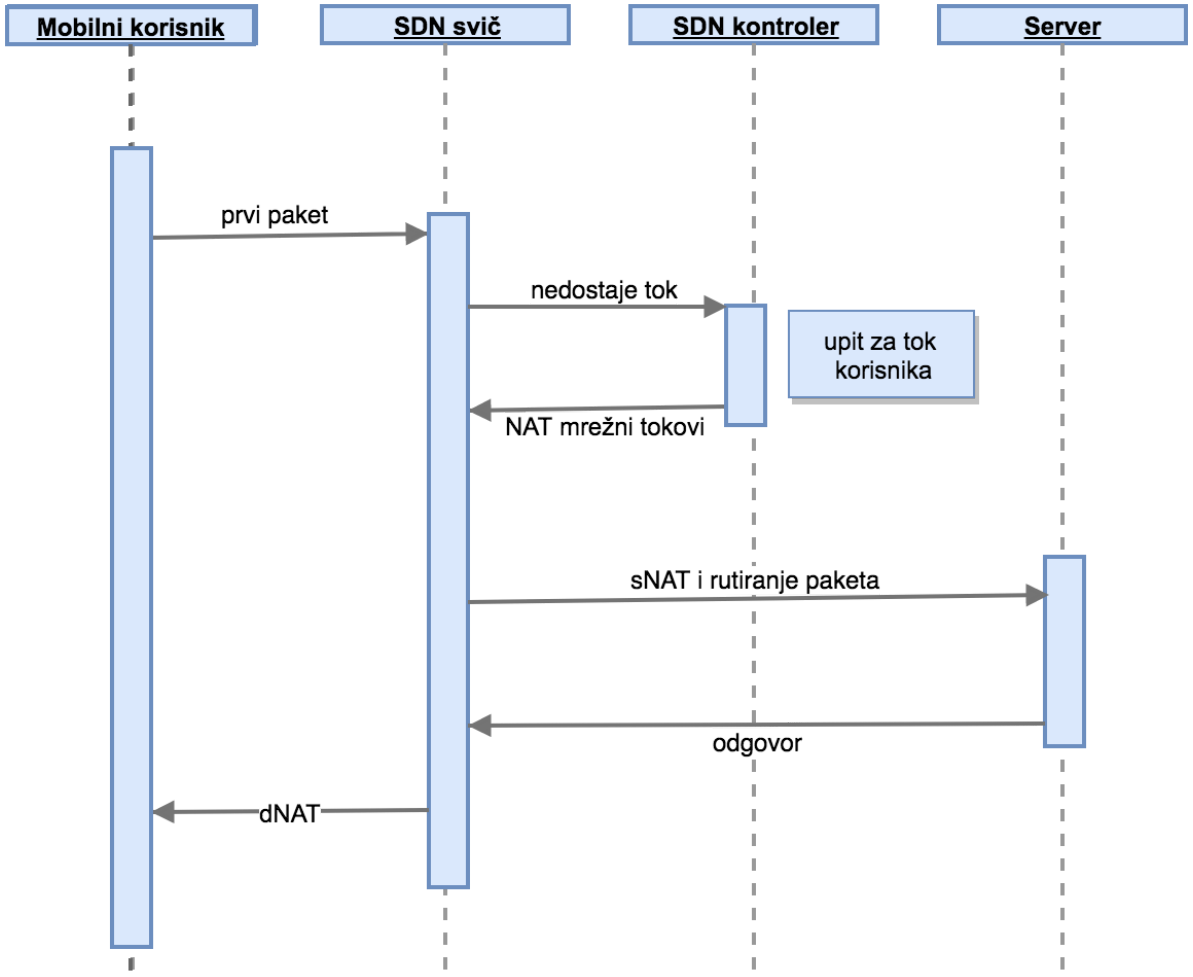
Uvođenjem posebnog servisa promenjen je tradicionalan način rada mreže u pogledu NAT servisa i u mrežu je uveden veći stepen programabilnosti i fleksibilniji pristup u rešavanju problema, što je osnovna karakteristika SDN tehnologije umrežavanja. Implementacija OpenFlow protokola za komunikaciju između rutera i SDN kontrolera (koji obezbeđuje servis za kontrolu), bio je sledeći logičan korak u razvoju rešenja za mobilnost u bežičnim IP mrežama. Analizom funkcionisanja OpenFlow protokola može se utvrditii da on, osim što utiče na prosleđivanje paketa, predviđa i proceduru u skladu s kojom je moguće vršiti izmene u paketu. Sposobnost OpenFlow protokola da omogući modifikaciju izvorne i destinacione IP adrese pruža mogućnost da se pravila translacije uređuju mnogo lakše nego u slučaju statičkog NAT-a.

Sa aspekta servisa za kontrolu translacija, neophodno je da se novim rešenjem predvidi implementacija OpenFlow kontrolera. Uloga kontrolera je da upravlja mrežnim tokovima u procesu translacije *rIP* u *pvIP* u odlaznom smeru (od mobilnog klijenta ka korespodentu) i *pvIP* u *rIP* u povratnom smeru.



Slika 4.5. Dijagram procesiranja paketa na SDN kontroleru

Novina u odnosu na postojeća rešenja je u tome što OpenFlow svič (SDN svič) primljeni paket, za koji nema instrukciju u tabeli tokova, odmah prosleđuje kontroleru na dalju analizu. Pošto procesira paket, kontroler vraća instrukciju koja se smešta u tabelu tokova (engl. *Flow table)* na sviču. Dakle, ako ne može trenutno da obradi paket, OpenFlow svič ga ne odbacuje, već ga privremeno baferiše dok ne stigne odgovor od OpenFlow kontrolera. Tako se izbegava gubitak prvih paketa koji je postojao u dosadašnjim implementacijama. Treba napomenuti da procesiranje prvog paketa unosi određeno kašnjenje i da će svi naredni paketi biti procesirani direkno od strane SDN sviča preko instrukcija koje se nalaze u tabeli tokova na sviču.

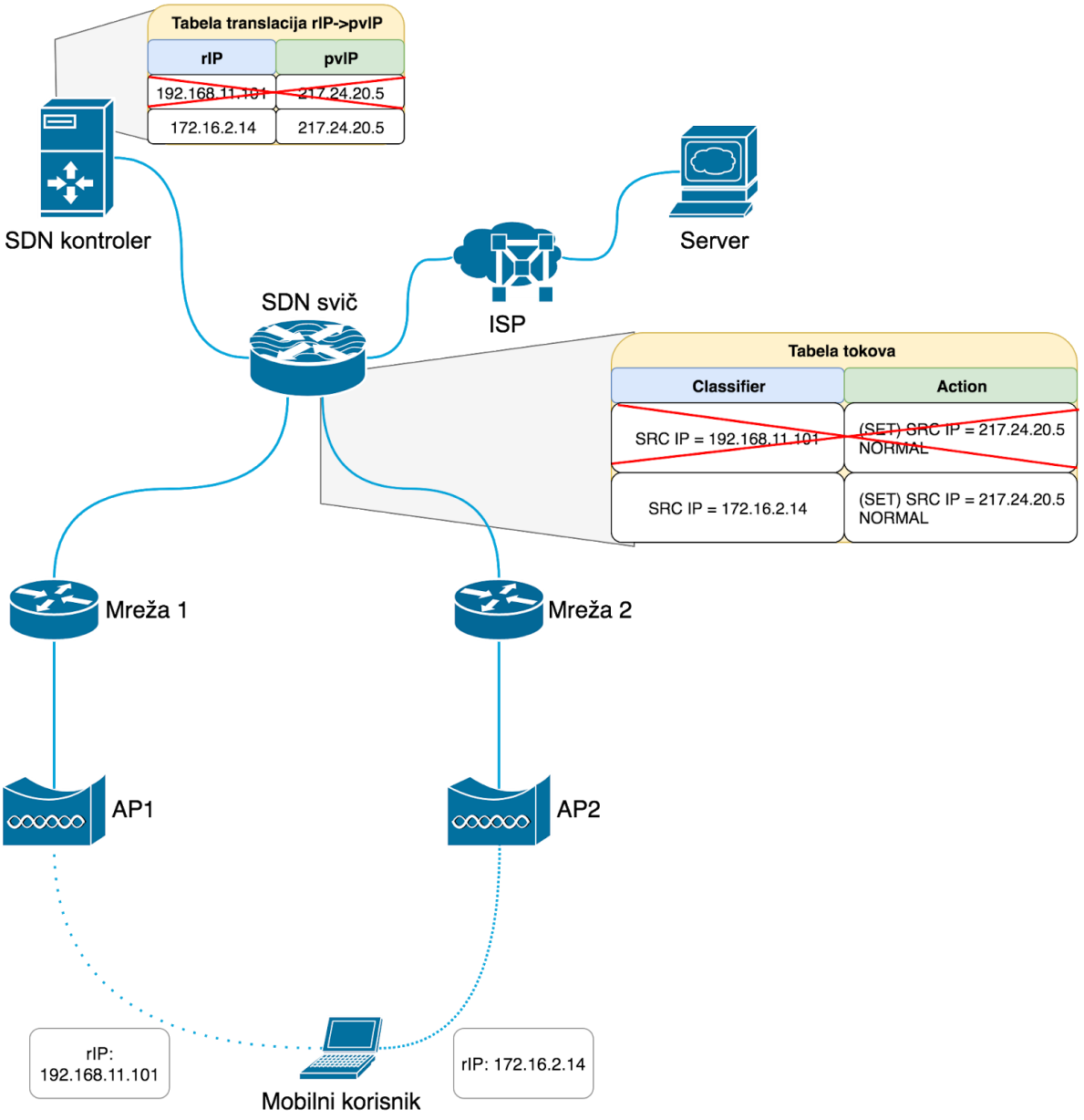


Slika 4.6. Dijagram toka prvog paketa

Perzistentnost komunikacije mobilnog korisnika sa udaljenim serverom

Predlogom novog rešenja za mobilnost u bežičnim IP mrežama, svakom mobilnom korisniku se alocira jedinstvena *pvIP* adresa kojom se on predstavlja spoljnim serverima. Mobilni korisnik, asocirajući se na mrežu bežičnog provajdera, dobija *rIP* adresu putem koje može da komunicira u lokalnoj mreži tog provajdera. Kada mobilni korisnik pošalje paket namenjen korisniku izvan domena mreže bežičnog provajdera, vrši se translacija adresa iz *rIP* u *pvIP* adresu. Za translaciju je zadužen SDN sposoban svič, koji se oslanja na lokalnu tabelu tokova sa instrukcijama koje su definisane od strane SDN kontrolera. Svaki paket, za koji se pretraživanjem u Flow tabeli ustanovi da ne postoji instrukcija, prosleđuje se SDN kontroleru na obradu. Cilj je da se napravi instrukcija, ažurira Flow tabela i stvore uslovi da se izvrši translacija adresa unutar IP paketa.

Dakle, jasno je da se prelaskom u drugu mrežu mobilnom korisniku menja i *rIP* adresa. Promenu lokacije mobilnog korisnika mora da detektuje SDN kontroler, čija je obaveza da SDN sviču pošalje novu instrukciju u pogledu translacije adresa. Nova instrukcija u stvari predstavlja izmenu koja se odnosi na translaciju adresa, i nova *rIP* adresa se mapira u postojeću *pvIP* mobilnog korisnika. Iz perspektive mobilnog korisnika, paketi postojeće komunikacije dolaze na novu IP adresu, kao da je preko ove adrese mobilni korisnik oduvek bio konektovan na javni internet. S druge strane, sa aspekta korespodenta, paketi korisnika dolaze sa *pvIP* adrese koja se nije menjala. Ovakvim predlogom rešenja obezbeđuje se perzistentnost komunikacije između mobilnog korisnika i udaljenog servera.

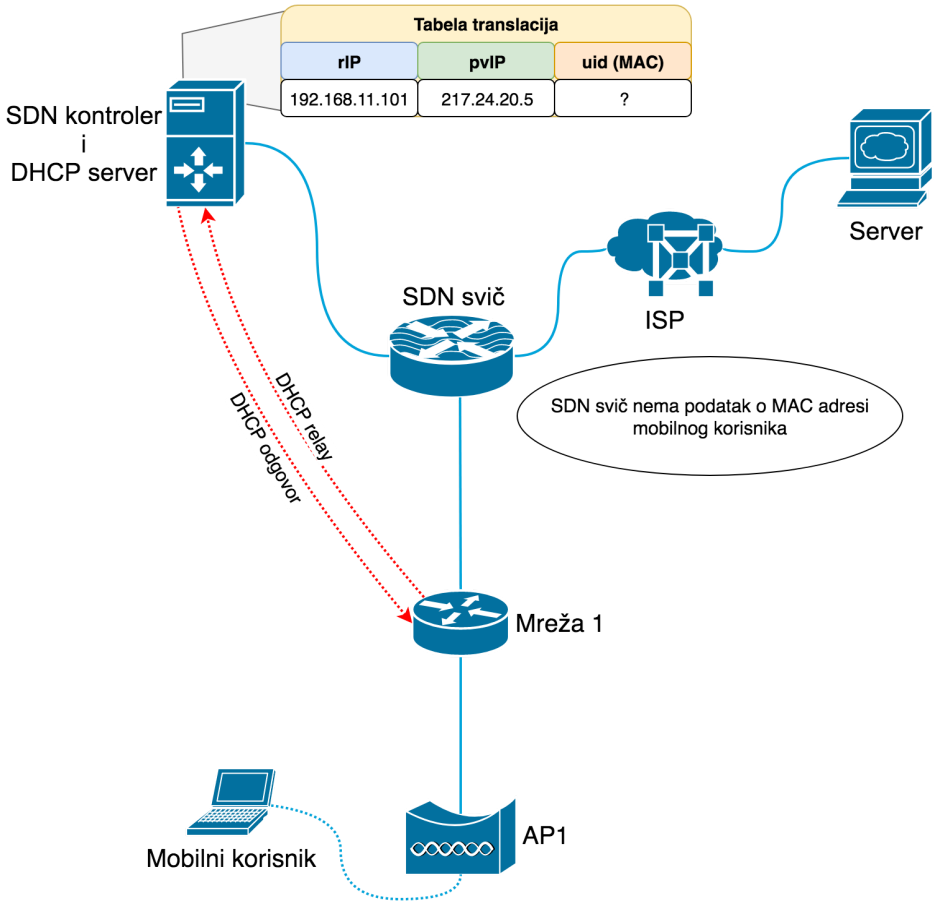


Slika 4.7. Prikaz prelaska u drugu mrežu

Identifikacija korisnika

Jedna od intencija u dizajniranju arhitekture tradicionalnih računarskih mreža je jasno razdvanje funkcija Network Access sloja od funkcija internet sloja. Cilj razdvajanje je da se omogući lakša distribucija saobraćaja. Takav pristup, međutim, potencijalno može dovesti do problema prilikom implementacije novog rešenja za IP mobilnost korisnika putem heterogenih bežičnih IP mreža. Problem je u vezi sa potrebom ili, bolje rečeno, jasnim zahtevom da se precizno definiše način na koji se jedinstveno identifikuje svaki mobilni korisnik.

Kada bi se kontrola paketa obavljala samo na nivou SDN sviča (na poziciji rutera koji treba da radi NAT), onda bi od unikatnih parametara korisnika bila vidljiva samo njegova izvorna IP adresa, a ona nije gotovo nikakav jedinstven identifikator mobilnog korisnika (jer je dodeljuje DHCP server, i može istu adresu dati nekom drugom ukoliko korisnik nije prisutan neko vreme). Problem jedinstvene identifikacije je praktično nemoguće rešiti samo na mrežnom sloju TCP/IP modela (ako izuzmemo DHCP i RA). Neophodno je pristupiti parametrima ili nižeg sloja (npr. MAC adresa) ili višim slojevima (npr. ime računara kroz SMB protokol). Potrebno je istaći da identifikacija kroz aplikativni sloj nije praktična, jer zahteva instalaciju posebnog softvera na klijentskom uređaju ili nadgledanje funkcionisanja postojećih protokola na mobilnom uređaju, što je veoma spor proces.



Slika 4.8. Dosezanje univerzalnog identifikatora korisnika (MAC adrese)

Najjednostavniji način rešavanja problema identifikacije verovatno je prosleđivanje DHCP ili RA zahteva do SDN kontrolera, iz kojeg SDN kontroler može da sazna MAC adresu mobilnog korisnika kao njegov univerzalni identifikator. Tada SDN kontroler kontroliše alokacije IP adresa u lokalnim pristupnim mrežama i prikuplja podatke o vezi između MAC i *rIP* adresa. Problem u ovom pristupu je to što SDN kontroler ne bi trebalo da bude DHCP server za svaki pojedinačni provajder bežične mreže, već bi trebalo da se svaki provajder oslanja na sopstveni servis za dodelu adresa. Važno je znati i to da kada bi SDN kontroler samo primao DHCP zahteve, a da na njih ne odgovara, mogao bi da sazna koji su korisnici prisutni u njegovoj mreži, ali ne i koju su *rIP* adresu dobili od DHCP servisa. Time kontroler ne može da poveže MAC adresu sa *rIP* adresama. Ovaj problem dodatno otežava i to što se u svim mrežama ne koriste DHCP servis i RA za alokaciju adresa korisnicima (npr. PPP mrežama se adrese alociraju u toku NCP procedure).

Problem identifikacije mobilnih korisnika u okviru nove platforme za IP mobilnost rešava se uvođenjem novog, tzv. *Tap* servisa u računarske mreže. Primena *Tap* servisa zasniva se na implementaciji servera, čija je osnovna funkcija da interno analizira sve

pakete koji stižu na ulazni ruter posmatrane bežične IP mreže. Princip rada *Tap* servera je takav da nakon što identifikuje paket sa MAC adresom koju prethodno nije registrovao, *rIP* adresu iz tog paketa i ostale podatke prosleđuje SDN kontroleru. Povezivanje *Tap* servera sa ruterom može se izvršiti preko hardverskog mini haba ili softverskim tehnikama kao što su PortMirroring, SoftwareTAP, CALEA i sl. Paket se na *Tap* serveru analizira istovremeno sa normalnom obradom paketa na ruteru. Svrha implementacije *Tap* servisa je prikupljanje informacija o mobilnom korisniku (npr. veza između MAC adrese i *rIP* adrese), privremeno keširanje i osvežavanje nakon određenog vremena. Važno je napomenuti i to da se procesiranje svakog paketa može izbeći primenom algoritma kojim se procesiranje svodi na ciljane pakete kao što su DHCP, RA ili PPP paketi.

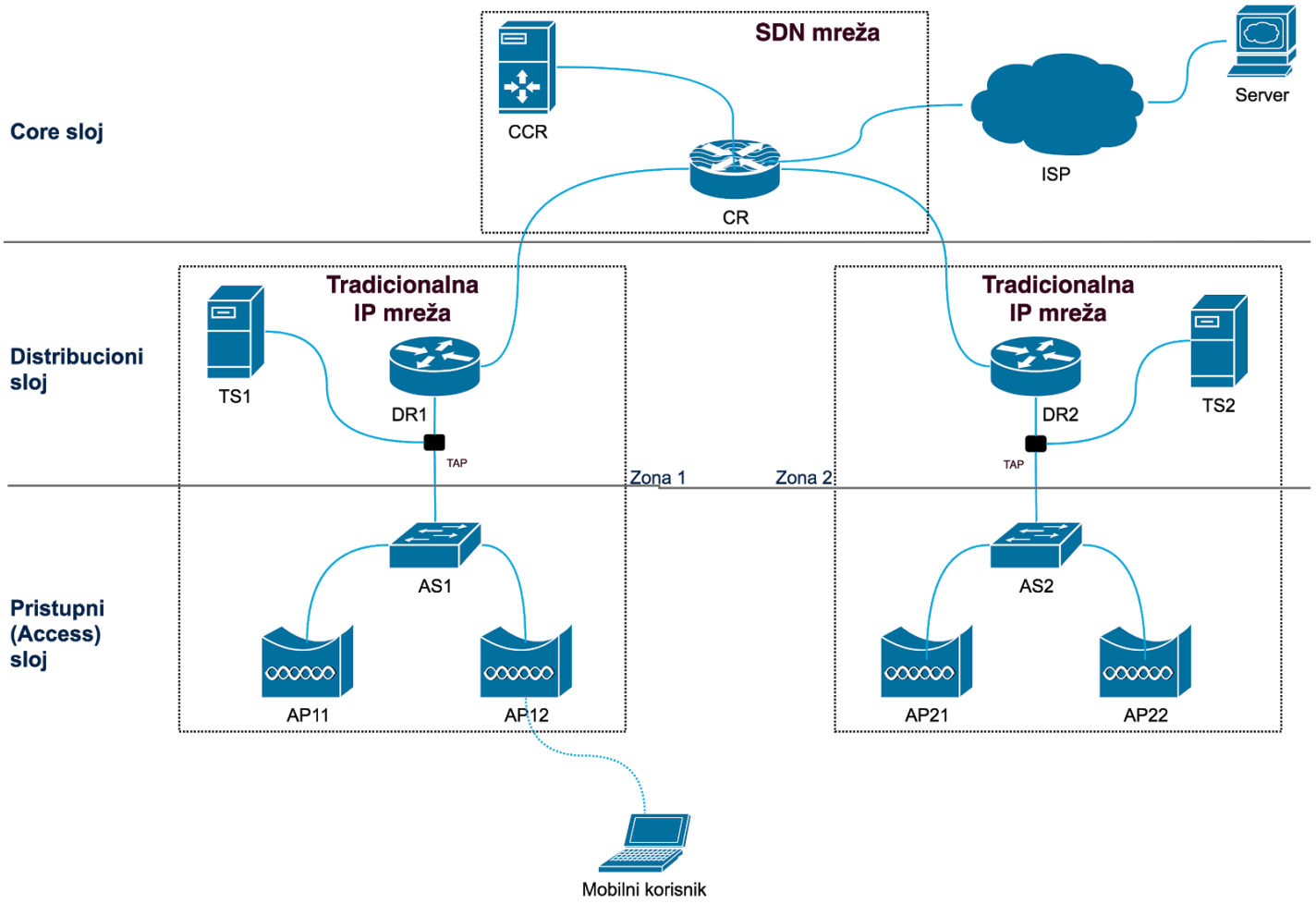
Motiv za izradu novog koncepta rešenja za IP mobilnost bilo je definisanje univerzalnog rešenja kako bi se prevazišle razlike koje postoje zbog heterogenosti bežičnih mreža (izgrađene su u različitim tehnologijama). Bez obzira na to što se, grubo posmatrano, na IP sloju gube razlike, zbog jedinstvene identifikacije korisnika, novo rešenje mora da obuhvati i niže slojeve TCP/IP modela. Zato osim MAC adrese kao jedinstvenog identifikatora treba uzeti u obzir i druge identifikatore istog korisnika. Posmatrano sa aspekta L2 tehnologija, to mogu biti:

- GSM, 3G – IMSI (*International Mobile Subscriber Identity*),
- LTE, LTE5G – IMSI ili GUTI (*Globally Unique Temporary Identifier*),
- 802.11 WiFi – MAC,
- 802.16 WiMax – MAC, i
- 802.15 Bluetooth, LiFi, ZigBee, LoRa – MAC.

U zavisnosti od toga koja se L2 tehnologija koristi, *Tap* servis prihvata jedinstvene identifikacione podatke korisnika, njegovu *rIP* adresu, i prosleđuje ih SDN kontroleru. Da bi odgovorio na izazov heterogenosti bežičnih mreža, SDN kontroler mora da poseduje vezu između jedinstvenih identifikatora u okviru različitih tehnologija (npr. IMSI i MAC adresa), što je trenutno izvan domašaja ovog istraživanja. Pretpostavka je da bi korisnik morao inicijalno da poveže svoje jedinstvene identifikatore preko neke registracione forme ili web aplikacije, ali samo prilikom prvog pristupa domenu bežičnih mreža (otvaranje naloga i sl.).

Dizajn novog rešenja

Da bi se predloženo rešene za IP mobilnost u heterogenim bežičnim mrežama integrisalo sa postojećim arhitekturama velikih računarskih mreža, koncipiran je dizajn troslojne mrežne arhitekture (slika 4.9).



Slika 4.9. Arhitektura rešenja IP mobilnosti

Uloga pristupnog (*Network Access*) sloja na slici 4.9. jeste da se korisnicima omogući prijem signala bežične mreže i fizički pristup mreži. Ovaj sloj podržava različite tehnologije pristupa kao što su LTE, 5GLTE, WiFi, WiMax i sl. Sastoji se od većeg broja pristupnih tačaka (AP) i repetitora. L2 mobilnost između AP-a uređena je tehnologijom koja se koristi i odvija na istom sloju.

Distribucioni sloj omogućava mobilnim korisnicima pristup bežičnoj IP mreži provajdera. Na ovom sloju nalaze se uređaji koji osim servisa alokacije IP adresa (DHCP,

RA i dr.) pružaju i *Tap* servis, koji omogućava jedinstvenu identifikaciju mobilnog korisnika i prenos informacije o njegovoj adresi do SDN kontrolera.

Core sloj predstavlja tačku u kojoj se objedinjavaju bežične mreže različitih provajdera. Na ruteru unutar ovog sloja (CR – core router) vrši se transliranje adresa radi održanja permanentne virtualne IP adrese za svakog mobilnog korisnika. CR je SDN sposoban uređaj koga kontoliše CCR (*Controller for Core Router*) kontroler.

Uloga svakog provajdera ovog domena L3 mobilnosti jeste da obezbedi funkcionalnost pristupa korisnicima na Access i distribucionom sloju. Eksterna, nezavisna instanca zadužena je da objedini različite ISP-ove i preko Core sloja omogući perzistentno mapiranje IP adresa korisnika pri izlasku iz domena L3 mobilnosti.

Glavna prednost ovog rešenja je u tome što ne zahteva značajne promene u infrastrukturi postojećih bežičnih mreža, već samo dodavanje jednog novog uređaja (*Tap* servera). Ovaj server zadužen je samo za prikupljanje potrebnih podataka i on je uređaj koji ne utiče na tok paketa i funkcionisanje tradicionalne mrežne infrastrukture. Dakle, SDN funkcionalnost dodata je samo u Core sloju, koji bi takoreći predstavljao posebnu organizaciju. Zadržavanjem postojeće infrastrukture i delimičnom implementacijom SDN funkcionalnosti, suštinski se realizuje hibridna SDN mreža koja obezbeđuje potpunu funkcionalnost IP mobilnosti, uz minimalne promene (finansijska ulaganja) u postojećoj tradicionalnoj infrastrukturi.

Istraživanje – eksperimenti

Sastavni deo ovog istraživanja predstavljaju eksperimenti koji su doveli do praktične realizacije opisanog rešenja za IP mobilnost. Međutim, realizacija je zahtevala da prvo rešimo određene praktične probleme koji se odnose na dostupnost SDN tehnologije.

U prvom eksperimentu smo pokušali da napravimo SDN sposoban svič sa minimalnom cenom implementacije, koji bismo u daljim eksperimentima koristili za demonostraciju SDN mogućnosti.

Drugi eksperiment predstavlja istraživanje u kojem smo pokušali da iskoristimo postojeću mrežnu opremu i nadogradimo je SDN funkcijama. Taj eksperiment nas je poveo u smeru istraživanja mogućnosti postojanja hibridnih mreža koje se sastoje i od tradicionalnih i od SDN elemenata.

Treći eksperiment je tehnička realizacija postavljenog rešenja za IP mobilnost. Koristeći iskustva iz prvog eksperimenta, u pogledu korišćenja OVS, i drugog eksperimenta, tj. tehnika koje kombinuju tradicionalnu i SDN mrežu, postavili smo praktičnu demonstraciju opisane funkcionalnosti IP mobilnosti po arhitekturi sa slike 4.9.

Eksperiment 1 – OpenFlow svič

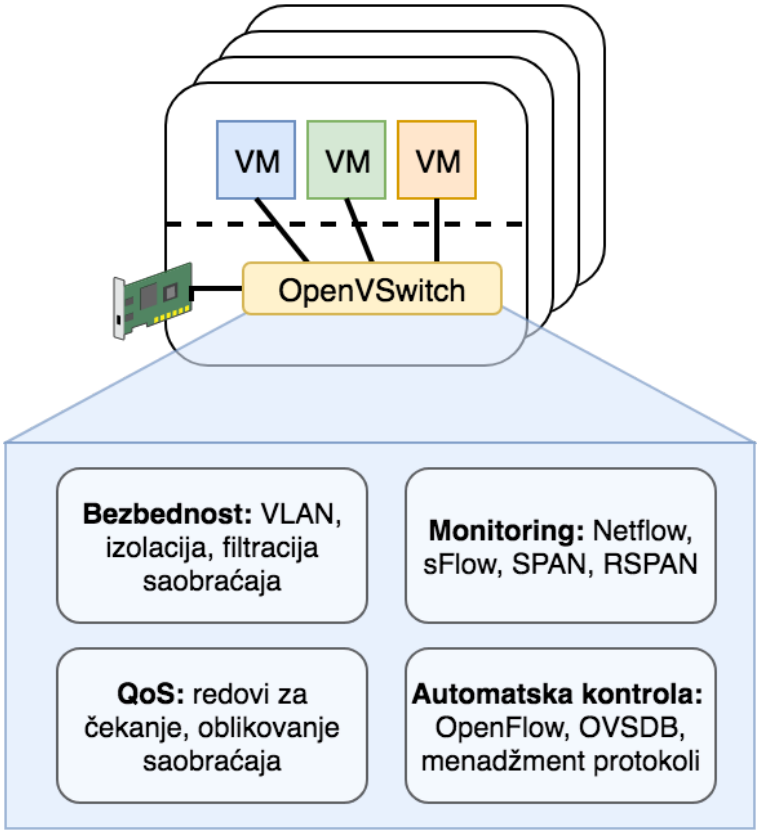
Hardverski svičevi čija je glavna funkcija prosleđivanje u skladu sa tabelom tokova, koji podržavaju OpenFlow porotokol kao standard, postoje već neko vreme. Međutim, primena ovih uređaja još nije na nivou koji se očekuje, a razlog tome su visoke cene ovih uređaja. Njihova primena se danas uglavnom svodi na uređivanje BGP tokova na Core sloju u mrežama internet provajdera. Na tržištu još nema adekvatnih uređaja koji bi hardverskim putem prosleđivali pakete na osnovu tabele tokova, a OpenFlow koristili za kontrolu zapisa tabele tokova.

Imajući u vidu da je podrška za OpenFlow protokol, a time i SDN, još uvek u fazi razvoja, prilično je teško doći do uređaja koji podržavaju OpenFlow standard. Zato je u prvoj fazi istraživanja zadatak bio da se napravi adekvatan OpenFlow svič baziran na OpenVSwitch softverskom rešenju. Izveden je eksperiment sa ciljem da se dobije fizički uređaj koji se može kontrolisati preko OpenFlow protokola. U toku realizacije ovog eksperimenta korišćeni su:

- OpenVSwitch – softver za upravljanje tokovima preko OpenFlow protokola,
- Soekris net4801 – hardverska razvojna platforma sa podrškom za Linux OS i većim brojem mrežnih portova, i
- Preboot eXecution Environment (PXE) – postupak podizanja operativnog sistema preko računarske mreže.

OpenVSwitch (OVS)

Danas postoji više softverskih projekata sa ciljem realizacije SDN sviča koji podržava OpenFlow protokol i SDN funkcionalnost. Verovatno najpoznatiji takav projekat je OpenVSwitch [38] (OVS), koji je realizovan u formi Open Soruce projekta od strane Linux fondacije sa Apache 2.0 licencom. Dizajniran je da obezbedi mehanizme za automatizaciju kao što su: OpenFlow, NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.11ag i sl.



Slika 5.1. OpenVSwitch (OVS)

Osnovne funkcije koje su implementirane na ovom softverskom sviču su:

- komunikacija između više vritualnih mašina putem NetFlow, sFlow, IPFIX, RSPAN ili GRE,
- podrška za LACP protokol,
- VLAN podrška 802.11Q i za pristupne i za trunk linkove,
- Multicast snooping,
- LLDP podrška,
- 802.1ag link monitoring podrška,

- STP i RSTP,
- QoS kontrola,
- HFSC qdisc podrška,
- NIC bonding i load balancing (slično kao Cisco EtherChannel),
- OpenFlow,
- IPv6,
- različiti tuneling protokoli – GRE, VXLAN, STT, Geneve, IPSec,
- interfejsi ka spoljnom upravljanju preko C i Python programskih jezika,
- forvardovanje paketa preko korisničkog ili Kernel režima,
- podrška za više istovremenih tabela prosleđivanja i virtualnih svičeva, i
- mogućnost potpune virtualizacije, tj. odvajanja hardverskog od softverskih resursa.

OVS rešenje danas je uglavnom sastavni deo računarske infrastrukture data centara i Cloud servisa. Razlog za njegovu sve veću implementaciju je potreba da se obezbede redundantni linkovi u sistemima visoke pouzdanosti i da se pri tome troškovi svedu na što manju meru. S druge strane, prednost ovog rešenja je u tome što je dovoljno da se koristi softverski svič kako bi se obezbedila redundansa linkova u slučaju kada postoji više mrežnih adaptera. Sa aspekta Cloud sistema, veoma je važno da postoji veza između različitih hostovanih virtualnih mašina, pa je zato neophodno izgraditi mrežnu infrastrukturu između njih. OVS predstavlja rešenje koje može da obezbedi više instanci virtualnih svičeva sa naprednim mogućnostima i tako napravi virtualnu mrežnu infrastrukturu.

Prva, danas verovatno i najčešća primena OVS OpenFlow svičeva, jeste u Clould sistemima kao što je Openstack [39]. Ovo Open Source rešenje koristi OpenFlow i OVSDB protokol preko svog modula za mrežnu kontrolu – Neutron, kako bi se konfigurisali i uredili tokovi u virtualnoj mreži OpenVSvitch procesa. OVSDB je standardizovan protokol, razvijan upravo za OVS, koji se koristi za uređivanje konfiguracije OpenVSwitcha [40]. Preko njega je moguće programabilnim putem menjati izgled virtualne mreže. Dakle, u SDN primeni, OpenVSwtich možemo koristiti putem dva protokola:

- OVSDB – protokola za kreiranje i konfiguraciju instanci virtualne mreže, i
- OpenFlow – protokola za uređivanje mrežnih tokova u svakoj instanci.

Soekris net4801

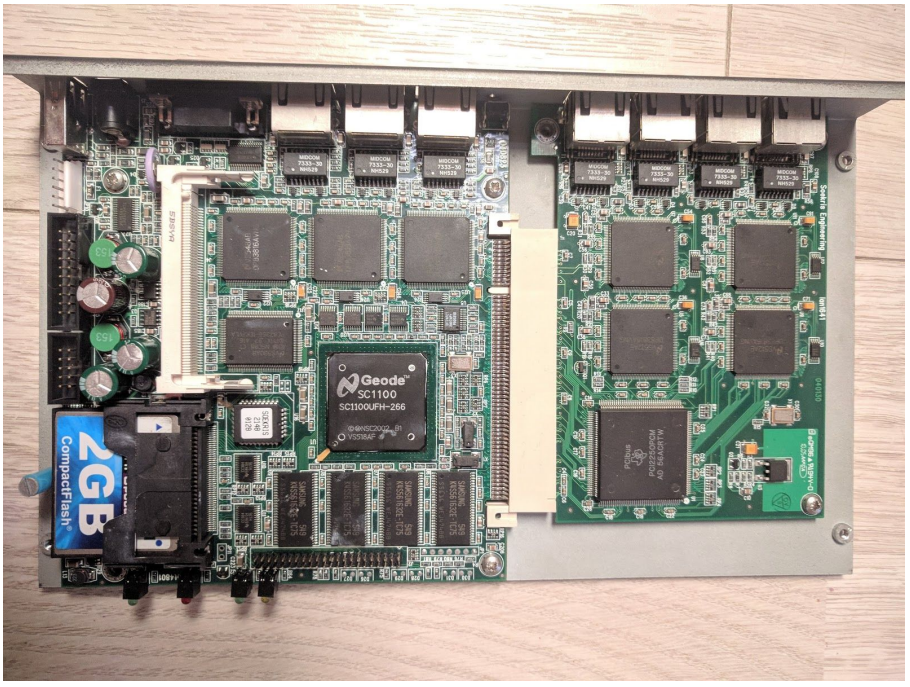
Hardverska platforma Soekris net4801 [41] ima sledeće karakteristike:

Procesor	AMD SC1100 na 266 MHz
RAM	128MB PC133 SDRAM
BIOS	512KB FLASH Soekris comBIOS
Magistrale	1x PCI slot, 1x miniPCI
Ethernet	3x FastEthernet DP83816
Disk	CompactFlash
Napajanje	6-28V DC, 15W max

Tabela 5.1. Soekris net4801 karakteristike

Uređaj je nadograđen dodatnom LAN karticom sa četiri dodatna Fast Ethernet porta, tj. sa Soekris lan1641 PCI karticom. Sve je to upakovano u namensku kutiju kako bi se dobilo celovito rešenje.

CompactFlash kartica od 2GB poslužila je za smeštanje Linux operativnog sistema.



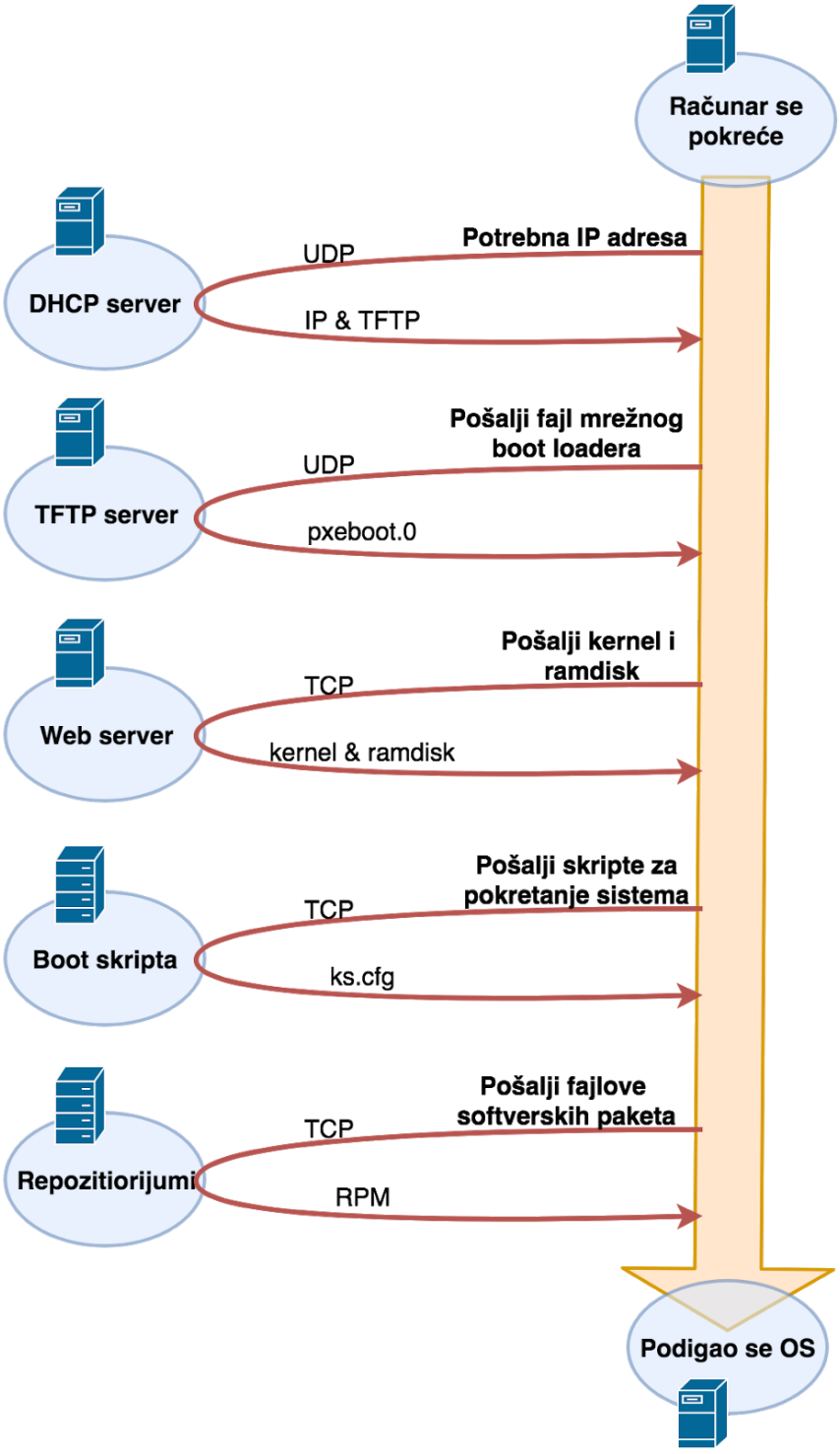
Slika 5.2. Soekris net4801 sa lan1641 karticom

Slika 5.3. Soekris net4801 pregled portova

PXE

Preboot eXecution Environment (PXE) predstavlja standardizovan postupak za podizanje sistema na klijentskim računarima preko računarske mreže. Postupak je propisan i publikovan 1999. godine, od kada je počela njegova impelementacija u mrežnim karticama. Mrežno butovanje sistema računara postiže se pomoću standardnih servisa kao što su DHCP i TFTP (TrivialFTP).

U početku se PXE koristio u konceptu *diskless* terminalnih računara koji bi OS podizali sa mreže prilikom svakog restarta računara. Danas se PXE prilično često koristi u data centrima u okviru RIS (engl. *Remote Instalation Service*) implementacije [46]. Često je PXE jedino rešenje za podizanje sistema na *headless embedded* sistemima, gde je dostupan samo terminalni (serijski) pristup. Danas je PXE procedura uključena i u *Unified Extensible Firmware Interface* (UEFI) [47] standarda.



Slika 5.4. Podizanje operativnog sistema preko mreže

PXE butovanje sa mreže podrazumeva sledeću proceduru:

- BIOS klijentskog računara u okviru PXE *Boot* procesa izborom opcije za mrežno butovanje,
- PXE *Boot* kojom se obavlja BOOTP/DHCP komunikacija i pribavlja IP adresa mrežnog segmenta,

- DHCP server isporučuje mrežne parametre i podatke o IP adresi TFTP servera i naziv fajla koji predstavlja *network boot loader* operativnog sistema,
- PXE proces primenjuje parametre dobijene od DHCP servera,
- kontaktira TFTP server tražeći fajl čije je ime dobio u DHCP procesu,
- preuzima fajl i započinje izvršavanje instrukcija iz tog fajla.

Po završetku PXE procedure, *network boot loader* preuzima podizanje sistema tako što u formi ramdiska preko TFTP ili nekog drugog protokola (HTTP, FTP i sl.) „preuzima" kernel. S preuzetim kernelom, sistem može da nastavi podizanje i da koristi spoljne konfiguracione resurse i repozitorijume, ili da uđe u proceduru instalacije sistema. Važno je istaći da konfiguracija PXE boot okruženja nije posebno komplikovana, a problem eventualno može da nastane ako u okruženju postoji DHCP server koji ne podržava konfiguraciju neke od dodatnih opcija kao što su:

- opcija 66 – adresa TFTP boot servera, i
- opcija 67 – naziv fajla mrežnog boot loadera.

TFTP servis se danas vrlo retko koristi. Kao veoma bazičan protokol za razmenu fajlova, koristi UDP kao komunikacioni kanal, i podržava samo dve komande – GET i PUT. I sama činjenica da se oslanja na UDP govori dovoljno o tome zašto se ne koristi često. Njegova primena u današnje vreme je upravo kod kratkih, pouzdanih prenosa, obično u lokalnoj mreži. Koristi se kod nadogradnji *firmwarea* kod nekih *dedicated* uređaja, kao i kod raznih *provisioning* (*auto-configuring*) sistema kao što su VoIP autokonfiguracija, PXE boot i sl.

Pošto TFTP servis ne stiže u standardnim verzijama savremenih operativnih sistema, za njegovo korišćenje neophodno je instalirati dodatni softver. Pod OS Windows, portabilni DHCP i TFTP serveri mogu se relativno lako koristiti preko *open-source* projekta TFTPD32 [48]. U Linuxu se TFTP klijent može dohvatiti iz repozitorijuma kao paket „tftp", dok je server dostupan kao „tftpd", „tftp-server" ili „tftpd-hpa". Nakon instalacije TFTP servera, potrebno je u *tftp-root* folder ubaciti *network boot loader* fajl kao i druge fajlove koji su potrebni za mrežno podizanje sistema. Važno je da se napomene da nemaju sve instalacije *network boot loader* uključen u ISO fajl ili instalacioni disk. U tim slučajevima, preuzima se *network boot loader* fajl drugog sistema i u konfiguraciji se nalaže korišćenje *ramdisk kernela* željenog operativnog sistema.

Opis eksperimenta

Instalacija operativnog sistema na *headless* uređajima bez grafičke karte predstavlja složen zadatak. U tom smislu, koristi se neka od sledećih metoda:

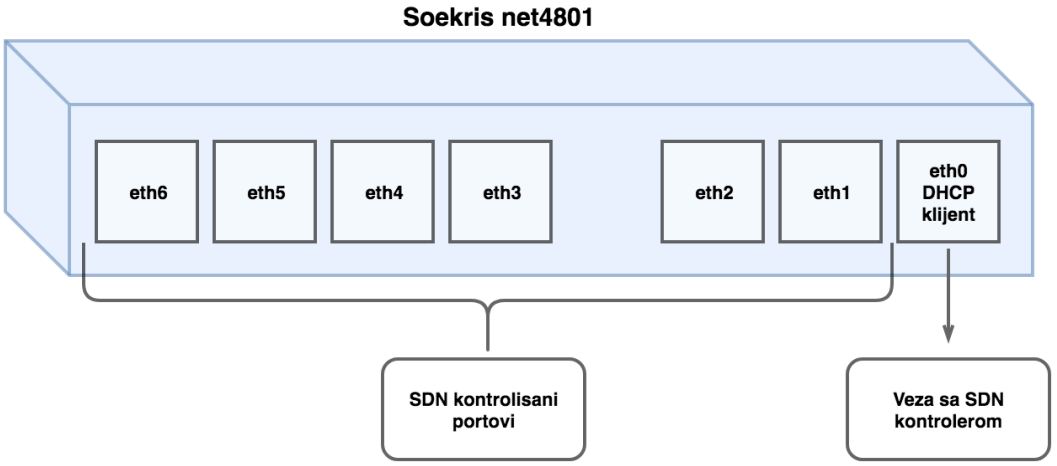
- Snimanje na CF/SD karticu unapred pripremljenog klona (engl. *clone image*) sistem – to je najjednostavniji metod jer se izbegava instalacija sistema i problemi koji tokom nje mogu nastati. Metoda se primenjuje kada postoji klon gotovo identične hardverske platforme (najmanja promena u hardverskoj platformi može da utiče na podizanje operativnog sistema).
- Instalacija sistema pre eksternog medijuma (USB flash ili USB CDrom) – pošto ne postoji grafička kartica, procesoru se pristupa serijskom vezom (*Null modem* serijskim kablom), čime se aktivira terminalni interfejs za komunikaciju sa BIOS-om. Time se omogućuje da se iz BIOS-a aktivira butovanje sa eksternog uređaja (instalacioni CD disk ili *Flash* drajv sa ISO *image* fajlom), a problem eventualno može nastati ako BIOS ne prepozna npr. USB uređaj kao *storage medium* pa sa njega ne dozvoljava butovanje.
- Instalacija sistema preko mreže, tj. PXE *boot* [42] – pruža se mogućnost da se deo operativnog sistema ili celokupan operativni sistem butuje preko lokalne mreže. Ona se oslanja na DHCP i TFTP servise kako bi uređaj povukao minimum operativnog sistema za butovanje.

U ovom eksperimentu, postojanje klona sistema nije bilo moguće jer je platforma već poprilično zastarela, pa su već i stari klonovi sistema postali nedostupni za preuzimanje. Opcija instalacije preko eksternih medijuma bila je adekvatno rešenje, s obzirom na to što platforma poseduje USB port. Testiranje je pokazalo da ugrađeni BIOS nema gotovo nikakvu podršku za eksterni USB storage i zato nije bilo moguće pokrenuti instalaciju sa eksternog USB CDroma niti sa desetak različitih vrsta USB flash diskova.

Treća varijanta, preko PXE boota, nakon rešavanja svih izazova mrežne konfiguracije, uspešno je pokrenula instalaciju. Zbog zastarelosti hardvera bilo je neophodno pokrenti instalaciju prilično stare verzije Linux OS. Najsvežija verzija za koju je uspešno započeta instalacija je Linux Ubuntu Server 10.04.4 LTS. Novije varijante nisu mogle da se pokrenu na ovom procesoru.

Linux Ubuntu je izabran zvog velikog repozitorijuma već pripremljenih paketa. Stoga se instalacija OpenVSwitch paketa svela na prosto pozivanje instalacije repozitorijuma. Instalirana je i IPref3 paket alatka za aktivno merenje i stres test mrežnih resursa [43]. Taj alat će se koristiti kako bi se izmerile performanse ovog OpenFlow sviča.

Konfigurisana je procedura tako da prvi port ovog uređaja bude port koji dobija adresu preko DHCP servisa i komunicira sa OpenFlow kontrolerom. Ostalih šest portova biće kontrolisani instrukcijama tabele tokova OVS-a.



Slika 5.5. Logička raspodela uloga portova

PC računar S predstavlja IPref3 server kojim će se testirati propusni opseg mreže. On će takođe imati i ulogu OpenFlow kontrolera. Za softver OpenFlow kontrolera korišćen je POX projekat [44]. U pitanju je vrlo jednostavan OpenFlow kontroler pisan u Pythonu. Od dodatnih funkcija, u kontroler je učitan modul za L2 učenje i prosleđivanje kako bi se obezbedilo da OpenFlow svič ima istu funkcionalnost kao tradicionalni L2 svič. Bilo je takođe neophodno promeniti i port za osluškivanje konekcije, jer je prema poslednjim specifikacijama standardizovan TCP port 6653 umesto prvobitnog 6633.

POX je pokrenut komandom:

```
./pox.py forwarding.l2_learning --port=6653
```

Na strani OpenFlow sviča, konfigurisan je OVS korišćenjem lokalne OVSDB komande:

sudo ovs-vsctl add-br of-switch	**kreiramo virtualni svič of-switch**
sudo ovs-vsctl add-port of-switch eth1 *sudo ovs-vsctl add-port of-switch eth2* *sudo ovs-vsctl add-port of-switch eth3* *sudo ovs-vsctl add-port of-switch eth4* *sudo ovs-vsctl add-port of-switch eth5* *sudo ovs-vsctl add-port of-switch eth6*	**dodajemo fizičke interfejse u virtualni svič of-switch**
sudo ifconfig eth1 promisc up *sudo ifconfig eth2 promisc up* *sudo ifconfig eth3 promisc up* *sudo ifconfig eth4 promisc up* *sudo ifconfig eth5 promisc up* *sudo ifconfig eth6 promisc up*	**prebacujemo fizičke interfejse u promiscuous mod kako bismo preskočili TCP/IP stack operativnog sistema**
sudo ovs-vsctl set-fail-mode of-switch standalone	**vraćamo rad u regularni režim u slučaju da kontroler nije dostupan**
sudo ovs-vsctl set-controller of-switch tcp:172.16.1.1:6653	**postavljamo of-switch u kontrolisan režim rada i postavljamo adresu i port SDN kontrolera**

OVS projekat podržava dva načina obrade dolaznih paketa. Jedan je prosleđivanjem iz procesa iz kog je pokrenut OVS *daemon* (tzv. *user-space data path* ili *netlink data path*), a drugi je preko kernela operativnog sistema (*netdev data path*). Testiranje je pokazalo da svičovanje preko kernela OS pruža mnogo bolje performanse jer prosleđivanje paketa nema direktno „takmičenje" sa ostalim procesima u OS. Da bi se to postiglo bilo je neophodno napraviti specifičan modul za kernel operativnog sistema koji je instaliran na uređaju.

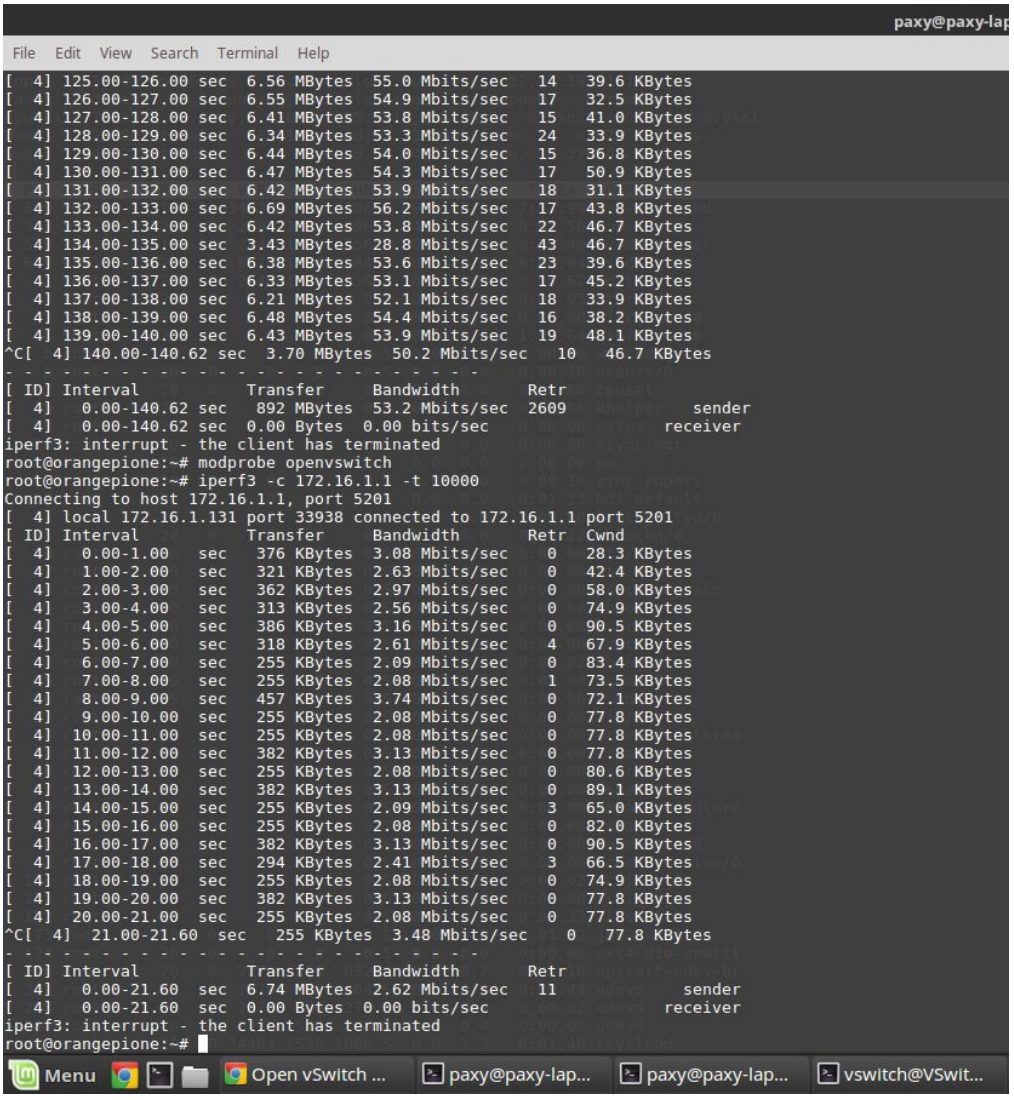
Cilj eksperimenta bio je da se dobiju što bolje performanse uređaja i zato je izvršeno kompajliranje kernel modula za *netdev datapath*. Ovaj postupak predstavlja pokušaj da se uređaj osposobi za kompajliranje bilo kojeg projekta, preuzimanje izvornog koda OVS projekta, kompajliranje samog modula za *netdev datapath*, kao i instalaciju modula u operativni sistem. Kada je ovaj proces kojim se instalira kernel drajver za netdev završen, stvoreni su uslovi da se implementiraju različite metode upravljanja paketima preko komande:

```
sudo ovs-vsctl set bridge of-switch datapath_type=netdev
```

ili vraćanja preko:

```
sudo ovs-vsctl set bridge of-switch datapath_type=netlink.
```

Rezultati eksperimenta



Slika 5.6. IPref3 merene performanse preko dva tipa *datapath*

Na slici 5.6. prikazane su performanse u *netdev* (preko kernela) i *netlink* (iz user-spacea) režimima rada. U gornjem delu slike nalaze se merenja koja ukazuju na ukupan propusni opseg u *netdev* režimu rada (izmeren je maksimalan propusni opseg od oko 55Mb/s), dok se u donjem delu nalaze merenja koja ukazuju na propusni opseg u *netlink* režimu rada, gde su performanse znatno lošije i postiže se maksimum od oko 3 Mb/s.

Analizom dobijenih rezultata može se zaključiti da i pored optimizacije procesa prosleđivanja paketa korišćenjem kernel drajvera, procesor u Soekris net4801 nije u stanju da podrži *throughput* ni približno brzini linka. Ukupan propusni opseg u *netdev* režimu rada je oko 55Mb/s i deli se na broj portova koji istovremeno imaju mrežne tokove. Ukupni propusni opseg linkova je 7x100Mb/s tako da performanse CPU predstavljaju „usko grlo”.

Važno je istaći i to da dobijeni rezultati eksperimenta ozbiljno navode na razmišljanje o tome da li vredi razvijati ovakva hibridna hardversko–softverska rešenja. Jasno je da postoje platforme s mnogo jačim procesorima, ali i dalje je pitanje da li se može postići tzv. *wire-speed* (maksimalna brzina tehnologije, npr. za FastEthernet 100mb/s). Naredna istraživanja u pogledu budućnosti hardverskih OpenSwitch uređaja moraju da vode ka jeftinijoj realizaciji sa hardverskim modulima koji *datapath* obavljaju u potpunosti brzinom linkova i sa podrškom za metodu kojom se eksternim magistralama menja sadržaj tabele prosleđivanja.

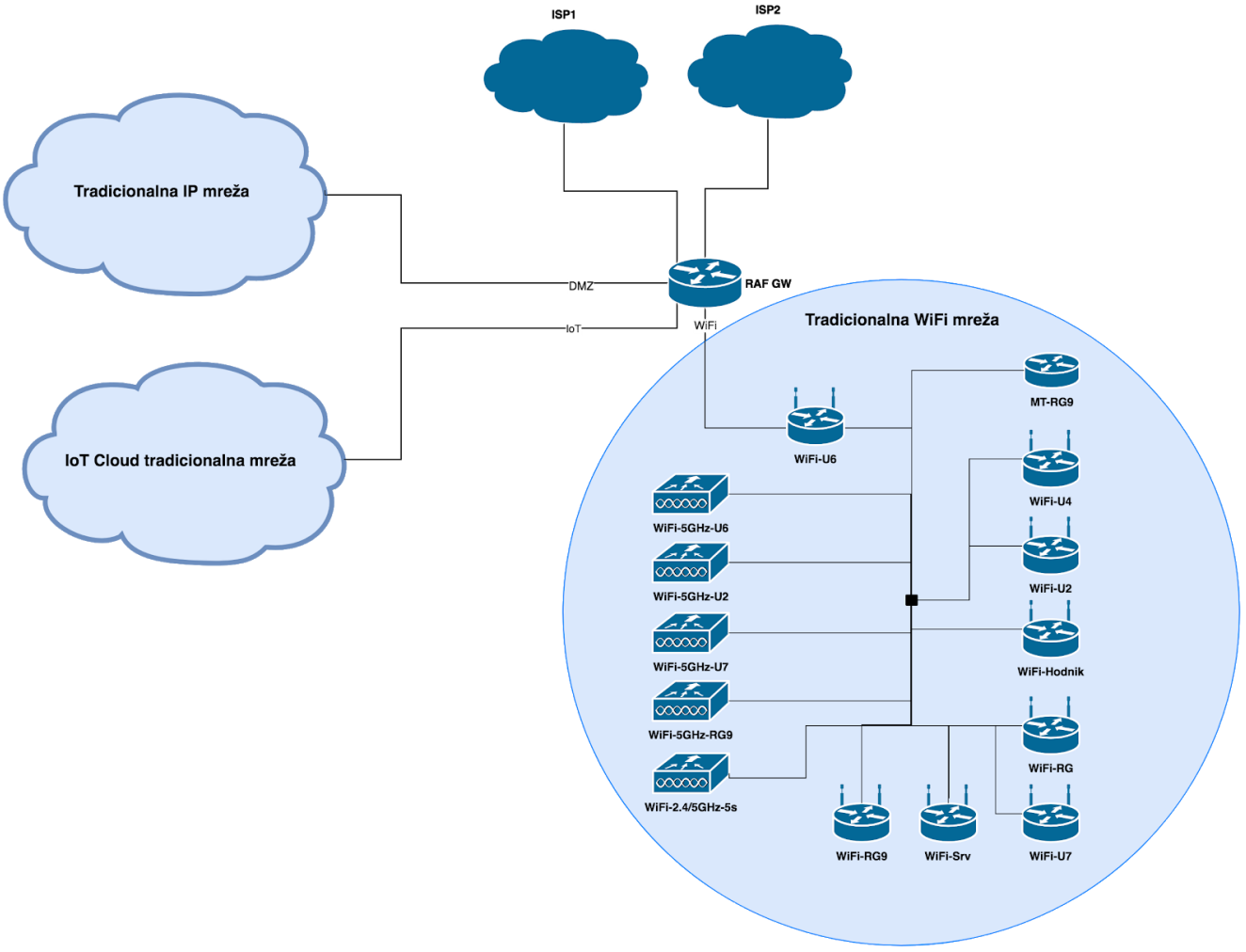
Eksperiment 2 – primena SDN okruženja u realnim mrežama

Implementacija OpenFlow protokola u zasebnom hardverskom uređaju i primena SDN funkcionalnosti još uvek predstavljaju težak zadatak. Sa stanovišta teorije, trebalo bi da je SDN svič funkcionalno i tehnički znatno jednostavniji za implementaciju nego tradicionalni svič ili ruter, što u realnom okruženju najčešće nije slučaj. To je bio razlog zbog kojeg je tok ovog istraživanja usmeren ka onim SDN uređajima koji su već dokazani kao funkcionalni i pouzdani u tradicionalnim mrežama, a ipak u određenoj meri imaju podršku za OpenFlow protokol (Mikrotik ruteri imaju podršku za OpenFlow V1.0.[49] i predstavljaju pretežno softversko rešenje za gotovo sve funkcije savremenih mreža).

Parcijalna integracija OpenFlow V1.0 specifikacije omogućava sledeće SDN funkcionalnosti na Mirkotik ruteru:

- uređivanje prosleđivanja tokova na osnovu bilo kog parametra hedera paketa,
- implementaciju Access Control Listi (ACL) i realizacija naprednih firewall tehnika,
- upravljanje multicastom,
- pravljanje VLANovima,
- upravljanje ICMP paketima,
- izmenu QoS vrednosti u paketima,
- izmena MAC adresa i
- izmenu tipa protokola na L2 i L3.

U trenutnoj realizaciji OpenFlow protokola na Mikrotik ruterima nije obezbeđena podrška za izmenu izvornih i destinacionih IP adresa u paketu. Međutim, i pored pomenutih nedostataka, ova platforma nudi mnogo mogućnosti u pogledu razvoja SDN mreža, što se posebno odnosi na okruženja gde je potrebno na fleksibilan način integrisati široko rasprostranjene bežične mreže. To je i bio zadatak eksperimenta iz rada [49], u kom je realizovano prebacivanje dela mreže na OpenFlow kontrolisan segment radi implementacije naprednih mogućnosti i izvršena integracija sa ostatkom postojeće, tradicionalne računarske mreže.

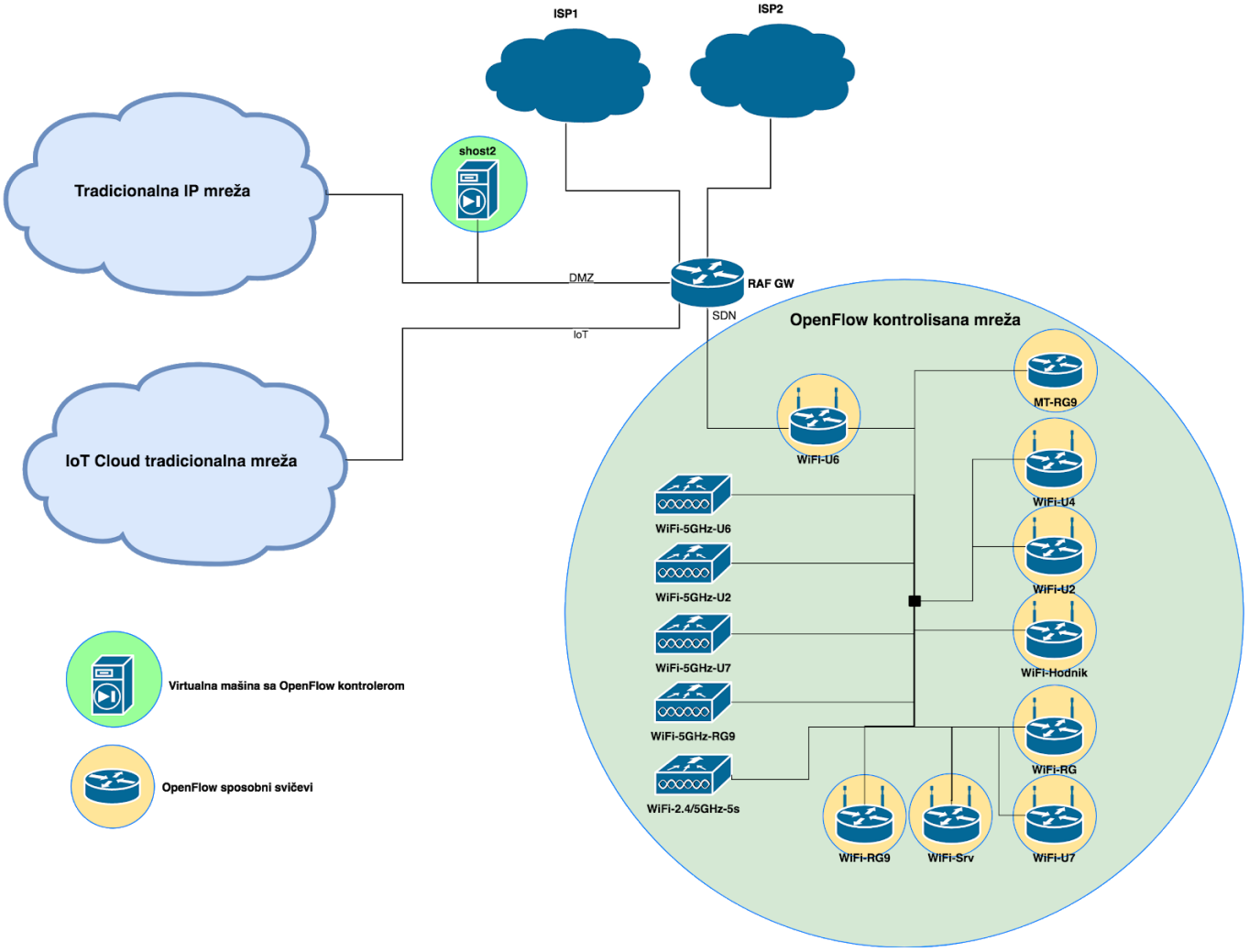


Slika 5.7. Šema prvobitnog izgleda RAF WiFi mreže

Eksperiment je izveden na mrežnoj infrastrukturi Računarskog fakulteta Union univerziteta u Beogradu (RAF). Ova infrastruktura se sastoji od većeg broja različitih mrežnih segmenata (prilično je heterogena i sastavljena je od 23 mrežna segmenta), a jedan od njih je upravo segment za bežični pristup mreži (namenjen studentima,

zaposlenima, IT administraciji i gostima). Da bi se zadovoljile potrebe različitih organizacionih celina unutar kompanije i obezbedila puna operabilnost u njihovom radu, mrežni segmenti su grupisani po funkcionalnostima, a realizovani su primenom različitih tehnologija: *Ethernet*, virtualizacije (VLAN,VPN), WiFi, *Link Agregation* i dr. Realizacija segmenta bežične mreže imala je za cilj obezbeđenje fizičke pokrivenosti signalom svake prostorije u okviru ustanove. Trenutna koncepcija je podrazumevala pokrivanje tri sprata sa učionicama, kancelarijama, čitaonicom, kabinetima i sl. Potpuna pokrivenost je postignuta sa osam Mikrotik WiFi rutera koji su podržavali IEEE 802.11b/g/n standarde. Vremenom se, međutim, pokazalo da je frekventni spektar od 2.4GHz prilično zagušen pa je bilo potrebno dodati podršku za IEEE 802.11a/ac standard sa frekvencijskim spektrom od 5GHz (u tu svrhu su dodati TP-Link AP uređaji).

Ovakva koncepcija bila je pogodna da se izvrši rekonfiguracija Mikrotik WiFi rutera u OpenFlow sposobne svičeve, i to u delu funkcionalnosti. Da bi se to realizovalo, bilo je neophodno da se WLAN interfejs na koji se povezuju korisnici i TRUNK interfejs, kao veza sa ostatkom mreže, „stave pod kontrolu” OpenFlow kontrolera.



Slika 5.8. Šema hibridne RAF mreže

Kao platforma za realizaciju različitih projekata u svrhu eksperimentisanja nad SDN mrežama, poslužio je open-source kontroler *Floodlight* [50]. Ovaj kontroler je izabran zato što je razvijen u Java programskom okruženju. Još jedan bitan argument za izbor *Floodlight* kontrolera (umesto danas poznatijeg *OpenDaylight* kontrolera) jeste taj što *Floodlight* ima ažurnu i kvalitetnu dokumentaciju podržanih klasa i metoda, i veliki broj dobro ilustrovanih primera za implementaciju dodatnih modula. Naime, u početnoj fazi ovog eksperimenta bilo je veoma korisno to što je *Floodlight* već posedovao ugrađene module za određene funkcionalnosti u svom izvornom kodu, koje je bilo potrebno samo aktivirati (npr. modul za *switching,* tj. L2 *learning and forwarding* koji vraća tradicionalnu funkcionalnost prosleđivanja VLAN paketa i omogućava optimizaciju prosleđivanja Dijkstrinim algoritmom). Sam SDN OpenFlow kontroler smešten je na virtualno hostovanu mašinu (VM) na jednom od servera u DMZ zoni. Ta VM izvršava Ubuntu Server OS i ima instaliran softver *Java* (JDK). Floodlight kontroler je pripremljen za rad kloniranjem akutelne verzije sa GitHub-a i

kompajliranjem, a nakon toga je izmenjen default *shell script* fajl *floodlight.sh* kojim se pokreće *Floodlight*. Ova promena podrazumeva da se u parametar pokretanja ubaci i konfiguracioni fajl modula *switching,* što će omogućiti kontroleru da vrši *L2 learning i forwarding.* Modifikuje se takođe i poslednja linija *floodligh.sh* tako što se ispred *$@* dodaje:

```
cf src/main/resources/learningswitch.properties
```

Posebna pažnja tokom konfiguracije se mora posvetiti podešavanju aktuelne verzije OpenFlow protokola na Floodlight kontroleru (usled promene broja porta, mora se proveriti da li su usaglašene verzije na OpenFlow svičevima i OpenFlow kontroleru). Imajući u vidu da je Floodlight aktivan projekat koji se konstantno nadograđuje (podržava najnoviju specifikaciju OpenFlow standarda) i činjenicu da Mikrotik podržava tek parcijalnu implementaciju (OpenFlow V1.0), bilo je neophodno da se u konfiguraciji Floodlight kontrolera definiše korišćenje OpenFlow specifikacije V1.0 i promeni broj porta sa TCP 6653 na TCP 6633. Ova konfiguracija dobijena je izmenom parametara u default konfiguracionom fajlu *src/main/resources/le arningswitch.properties*. Dodati su takođe i parametri:

```
net.floodlightcontroller.core.internal.OFSwitchManager.openFlowPort=6633,
net.floodlightcontroller.core.internal.OFSwitchManager.openFlowAddresses=0.0.0.0, i
net.floodlightcontroller.core.internal.OFSwitchManager.supportedOpenFlowVersions=1.0.
```

Pokretanjem Floodlight kontrolera pomenutom konfiguracijom nisu rešeni svi problemi jer su OpenFlow svičevi često gubili vezu sa kontrolerom i pokušavali ponovo da je uspostave. U daljem toku istraživanja, fokus je bio na razlogu za pojavu ovog problema. Analizom se došlo do zaključaka da je razlog nepotpuna implementacija OpenFlow specifikacije u Mikrotik modulu. Naime, u razvojnoj fazi, Mikrotik je uspeo da podrži samo OpenFlow V1.0 specifikaciju i nije podržana mogućnost pregovaranja oko OpenFlow verzije specifikacije. S druge strane, ta mogućnost je implementirana na Floodlight kontroleru, koji može da izvrši usaglašavanje čak i po specifikaciji V1.0. Dakle, zbog neusaglašenosti specifikacija, „razgovor” između Floodlighta i Mikrotik OpenFlow modula je prekinut i nije mogao da bude uspešno uspostavljen. Međutim, zahvaljujući tome što je Floodlight kontroler *Open-source* projekat, problem je rešen modifikacijom izvornog koda Floodlight kontrolera [51] u izvornom fajlu:

src/main/java/net/floodlightcontroller/core/in ternal/OFChannelHandler.java

- iznad koda:

  setState(new WaitFeaturesReplyState());

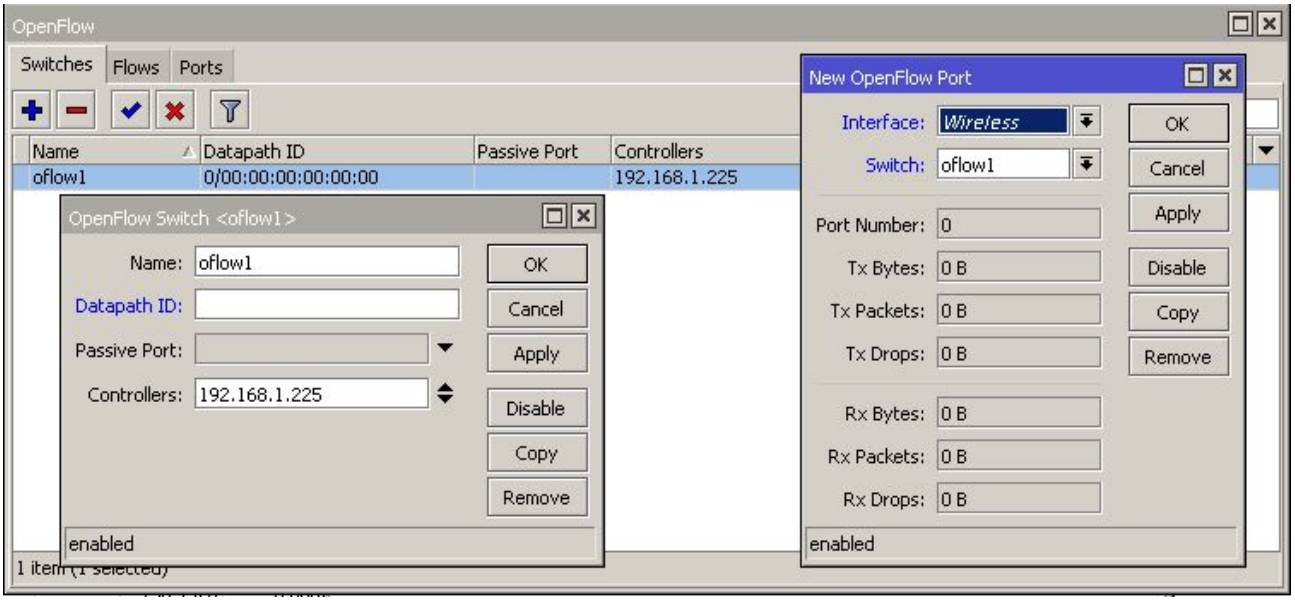
  ubacuje se:

  sendHelloMessage();

- naredno pojavljivanje poziva funkcije *sendHelloMessage() se* briše ili se gasi komentarom.

Posle ove modifikacije, potrebno je ponovo kompajlirati Floodlight kontroler i ponovo ga pokrenuti *shell* skriptom *floodlight.sh*. Nakon rešavanja problema na kontroleru, izvršena je konfiguracija OpenFlow modula na Mikrotik uređajima. Konfiguracija podrazumeva sledeće korake:

- otvaranje OpenFlow opcije unutar Winbox menadžera,
- kreiranje instance OpenFlow sviča, npr. *oflow1,*
- podešavanje IP adrese kontrolera u okviru opcije *Controllers, i*
- pod tabom Ports, dodavanje portova koji se predaju na kontrolu OpenFlow kontroleru.



Slika 5.9. Mikrotik konfiguracija OpenFlow sviča

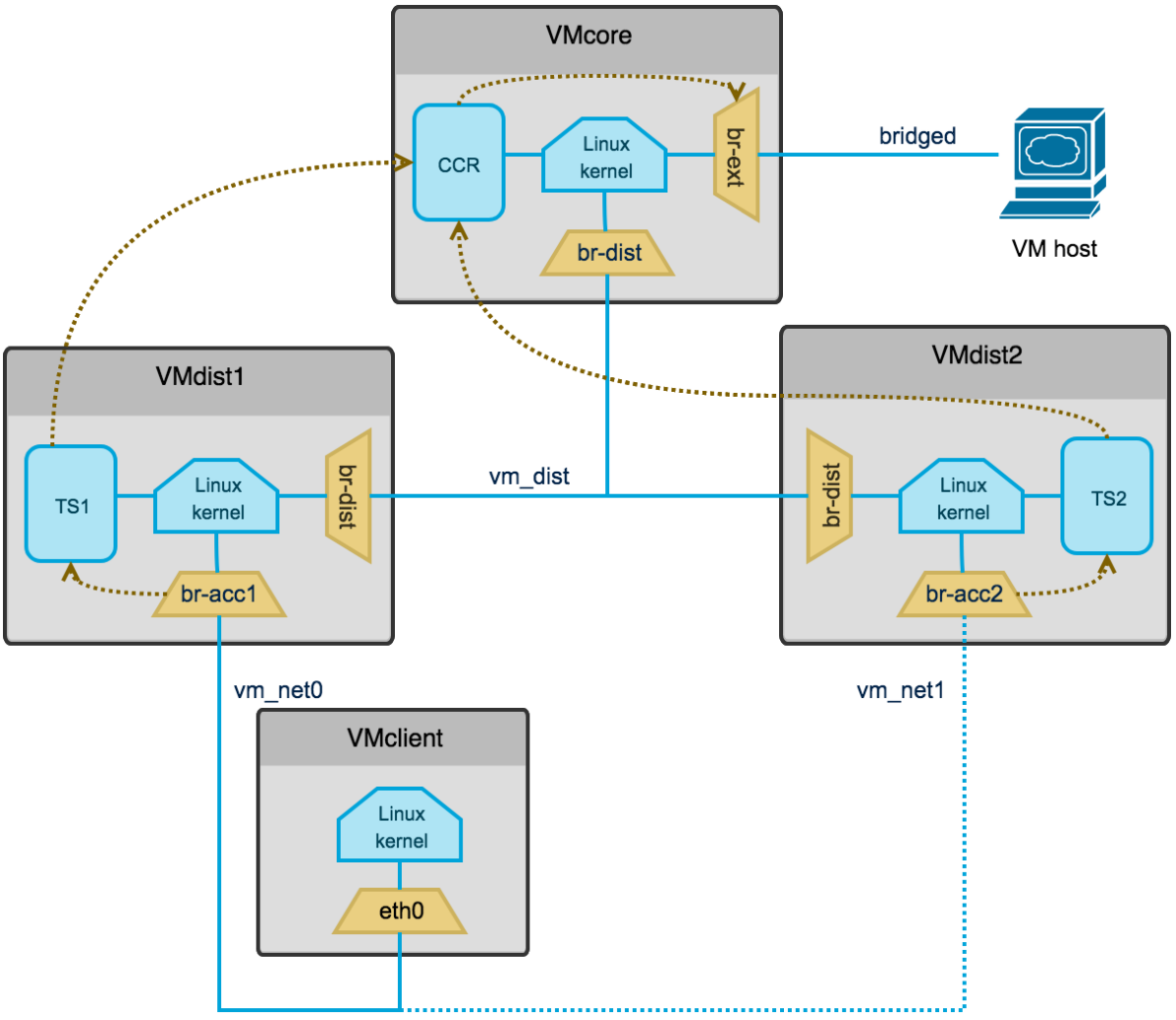
Rezultati do kojih se došlo ovim eksperimentom pomogli su da se napravi poligon za testiranje SDN funkcionalnosti i implementaciju naprednih mogućnosti. U tom smislu je u nastavku istraživanja razvijen modul na Floodlight kontroleru, koji vrši dinamičku kontrolu pristupa bežičnoj mreži i omogućava sprovođenje visokokonfigurabilnih politika pristupa mreži. Osim modula namenjenih sprovođenju politika bezbednosti, razvijeni su i moduli za uređivanje QoS politika. Takvi moduli imaju zadatak da modifikuju CoS polje paketa kako bi mreža razlikovala tipove saobraćaja. Sastavni deo QoS modula je i modul za monitoring koji prikuplja podatke sa brojača različitih tokova na SDN svičevima i obrađuje ih u statističke svrhe.

Najvažniji rezultat ovog eksperimenta je praktičan dokaz da je moguće realizovati hibridnu SDN mrežu koja se delom oslanja na tradicionalne uređaje i paketski način obrade podataka, a delom na koncept obrade mrežnih tokova sa implementiranom funkcionalnošću programabilnih mreža.

Eksperiment 3 – SDN rešenje za L3 mobilnost

Prethodna dva eksperimenta su pokazala da je moguće napraviti softverski SDN svič nad nekom *embeded* platformom, ali da rešenje ima nedostatak jer zahteva značajne procesorske performanse da bi se podržale LAN brzine tokova. Takođe je pokazano i da prelazak na SDN mrežu i integracija OpenFlow protokola kao dodatne funkcionalnosti u postojeće softverske rutere, jesu mogući. Međutim, eksperiment je pokazao i to da je u ovoj fazi istraživanja veliki izazov kako za testiranje napraviti adekvatno okruženje. Naime, prethodni eksperiment je pokazao da aktuelna implementacija OpenFlow specifikacije podrazumeva samo V1.0 specifikaciju, i to bez bilo kakve mogućnosti prepravke paketa (eksplicitno je naznačeno da prepravka IP adresa u paketu nije implementirana).

Ovde je važno naglasiti da je u poglavlju u kome je prezentovano rešenje za L3 mobilnost ukazano na potrebu da se za realizaciju programabilnog NAT-a iskoriste napredne SDN funkcionalnosti. Da bi se testiralo predloženo rešenje, odnosno da bi se izveo adekvatan eksperiment, napravljeno je virtualno okruženje. Razlog za to je potreba da se eksperiment može ponoviti i da njegova realizacija ne zavisi ni od jedne konkretne hardverske platforme.



Slika 5.10. Dijagram eksperimenta sa virtualnim mašinama

Eksperiment je koncipiran kao laboratorija virtualnih mašina (VM) čije su funkcionalnosti podeljene prema predloženim slojevima arhitekture našeg rešenja. Korišćen je *Oracle VirtualBox* kao rešenje za eksperiment nad virtualnim mašinama i virtualnim mrežama.

Definisane su sledeće uloge VM:

- VMclient – VM koja predstavlja korisnika mobilne mreže sa mogućnošću prelaska u različite mreže;
- VMdist (1,2) – dve instance VM koje obavljaju funkciju distribucionog sloja predložene arhitekture; i
- VMcore – VM sa funkcionalnošću Core sloja predložene arhitekture.

VMclient predstavlja instancu VM sa Ubuntu OS, koja se sastoji samo od osnovne instalacije sistema, bez dodatnih paketa. U inicijalnoj postavci, VMclient je povezan na virtualnu mrežu *vm_net0,* tj. mrežu ISP1. Dalja simulacija zahtevala je fizički prelazak sa ISP1 na ISP2, pa je omogućeno da vezu sa ISP2 klijent realizuje preko *vm_net1,* kada to bude potrebno.

VMdist mašine su napravljene tako da se sastoje od svih predviđenih komponenti distribucionog sloja arhitekture našeg rešenja. Njihova realizacija je bazirana na Ubuntu serveru OS. U prvom koraku su formirana dva virtualna mrežna interfejsa, jedan ka Access sloju, drugi ka Core sloju. Intefejs *br-acc* za ISP1 je povezan na *vm_net0* virtualnu mrežu, dok je isti interfejs ISP2 povezan na virtualnu mrežu *vm_net1.* Intefejs *br-dist* povezan je linkom ka Core sloju, tj. povezan je na zajedničku virtualnu mrežu svih ISP distribucija *vm_dist*. Linux kernel je zadužen za rutiranje paketa između intefejsa na klasičan način – ruting tabela, ruting proces. TS servis je servis koji osluškuje pakete *br-acc* interfejsa, i kao Tap servis o unikatnim klijentima obaveštava kontroler Core sloja (CCR). Obe instance VMdist su konfigurisane sa adekvatnim IP adresama koje su alocirane tom konkretnom ISP-u. Na linku ka Core sloju, instance imaju adrese u istom adresnom prostoru kako bi se ostvarila veza sa Core slojem. DHCP (RA) servis je konfigurisan na *br-acc* kako bi mobilni korisnici dobili realnu privremenu IP adresu, čim ostvare asocijaciju. Konfigurisane su default rute tako da upućuju ka Core sloju, i dodate statičke rute ka susednim ISP-ovima radi optimizacije putanje.

VNcore mašina se sastoji od komponenti Core sloja arhitekture. Implementacija je realizovana na Ubuntu serveru OS. Linux Core se koristi za tradicionalno rutiranje paketa na osnovu lokalne ruting tabele. Interfejs *br-dist* je povezan na virtualnu mrežu čiji su članovi distribucioni slojevi drugih bežičnih ISP-ova, t. *vm_dist*. Interfejs *br-ext* je intefejs povezan sa izlaznom tačkom iz domena IP mobilnosti. U praksi je to verovatno neki natprovajder ili više njih. U našoj laboratoriji, to je bridžovan intefejs sa fizičkom mrežom u kojoj se laboratorija realizuje. Intefejs *br-ext* je ujedno OpenFlow kontrolisani svič koji ostvaruje instrukcije OpenFlow kontrolera. CCR je kontroler zadužen za implementaciju predložene funkcionalnosti i uređivanje instrukcija translacije za izlazni interfejs *br-ext*.

Kao korespodent ili računar izvan domena IP mobilnosti mogli su biti korišćeni host računar ili računar iz fizičke mreže sa servisima za testiranje funkcionalnosti i performansi mreže.

Na VMdist mašinama realizovana je Java aplikacija koja obavlja funkcionalnost Tap servera.

Na VMcore konfigurisan je OpenVSwitch softver tako da kontrolu rada nad *br-ext* interfejsom prepusti OpenFlow kontoleru. Floodligh je open-source platforma korišćena za CCR i razvoj kontrolerske logike. Za Floodlight su razvijena dva Java modula, jedan za komunikaciju sa TS-om i proaktivnu konfiguraciju, a drugi za reaktivno procesiranje paketa.

Tehnička realizacija – opis Java modula

Pokretanjem Floodlight kontrolera, inicira se *MobilityDiscovery* (MD) modul kao i *MobilityService* (MS). MD je zadužen za osluškivanje, prihvatanje i obradu TCP konekcija pristiglih od *HostDiscovery* (HD) procesa koji se izvršava na TS (distribucionom sloju). Nakon uspostavljanja komunikacije, HD isporučuje podatak u formi *uid#rIP*, gde *uid* predstavlja unikatnu identifikaciju korisnika, a *rIP* njegovu trenutno realnu IP adresu.

Prilikom procesiranja podatka, MD proverava da li postoji zapis u tabeli – *MobilityServiceTable* (MST), koji odgovara *uid* identifikatoru korisnika. Ukoliko zapis ne postoji, radi se o novom korisniku koji se prvi put (posle određenog perioda) prijavljuje u domen L3 mobilnosti. Za tog korisnika alocira se nova, unikatna *pvIP* adresa i u MST tabelu upisuje zapis sačinjen od tripleta *uid-rIP-pvIP.* (Napomena: ako su mreže heterogene, za istog korisnika može postojati više *uid-a*, čije su međusobne veze unapred definisane i predstavljaju deo istog zapisa). Nakon toga, MD kreira proaktivnu OpenFlow instrukciju za sNAT kako bi se izvorišna adresa prepravila iz *rIP* u *pvIP*, kao i dNAT, kako bi se u povratku destinaciona IP adresa prepravila iz *pvIP* u *rIP.* Tokove ubacuje (*push*) u SDN svič, tj. *br-ext* u ovom primeru. Oni su vremenski limitirani kako bi se omogućilo automatsko čišćenje tabele tokova usled neaktivnosti korisnika. U eksperimentu je definisan vremenski „limit” od deset minuta. Pored toga, dolaskom nove instrukcije za postojeći tok, SDN svič uklanja stare tokove.

SDN svič (u ovom slučaju OVS) prilikom podizanja uspostavlja komunikaciju sa Floodlight kontrolerom koji dalje upravljanje prepušta MS modulu. MS modul je zadužen da svakom SDN sviču (mada je u našem primeru samo jedan) odmah vrati instrukciju da svaki paket procesira sa akcijom **NORMAL**, što znači da ga prepusti Linux kernelu na tradicionalno procesiranje. Ova instrukcija se šalje sa veoma niskim prioritetom izvršenja

kako bi se izvršila samo ako ne postoji nijedna druga instrukcija za isti tok. Pomoću NORMAL instrukcije, kroz SDN nije potrebno da se napravi kompletan ruting modul, već samo deo funkcionalnosti kojom treba proširiti mrežu. Svaki mrežni tok koji proaktivno šalje MD ima viši prioritet izvršenja tako da će se ta instrukcija izvršiti uvek pre nego inicijalna NORMAL komanda.

Pošto funkcionalno CCR ne može sam da dođe do unikatne identifikacije korisnika, za te podatke oslanja se na MH proces. HD proces predstavlja nezavisnu Java aplikaciju koja se izvršava na TS uređaju (u našem primeru nalazi se unutar VMdist-a). Ovaj proces dobavlja pakete sa Access sloja i analizira heder data link sloja. Dohvatanje paketa se obavlja bilo hardverskim ili softverskim tap-om. U eksperimentu je korišćen JPcap [52] koji predstavlja Java biblioteku za snifovanje paketa (prihvatanje paketa sa fizičkog sloja zaobilaženjem TCP/IP pravila procesiranja) sa mrežnog interfejsa.

Procesiranje paketa od strane HD-a, podrazumeva proveru da li je paket došao sa poznate MAC adrese (ili drugi unikatni identifikator). Ukoliko je to nova identifikacija, čita se podatak o izvorišnoj IP adresi i formira par *uid#rIP*. Umesto analize svakog paketa, zbog opterećenja servisa, može se analizirati samo određeni filtrirani paket kao što je BOOTP/RA/DHCP ili možda ARP. Međutim, pouzdanost je znatno veća ako se analizira svaki paket.

Ukoliko je uhvaćen par *uid#rIP* identifikator novog korisnika (tj. nije primećen ranije), MH obaveštava zasebnom perzistentnom TCP konekcijom MD CCR o uhvaćenom paru. Takođe, MH taj par kešira u lokalnu keš tabelu zajedno sa sistemskim vremenom. Po dolasku novog paketa već poznatog para, proverava se lokalni keš a MD obeveštava samo ako je prošlo više vremena nego što je definisano parametrima procesa. Konkretno, obaveštavanje o postojećim tokovima se u našem primeru vrši na svakih pet minuta.

Prilikom svake obnove postojećeg para *uid#rIP*, MD će uvideti da postoji zapis u MST tabeli i pripremiti apdejtovan proaktivni tok. Time će OpenFlow sviču poslati instrukciju da taj tok još uvek važi.

Tako, kada HD primeti aktivnost korisnika, za njega se formiraju tokovi translacije sa *timeout* periodom od deset minuta.

Ukoiliko korisnik ne bude aktivan neko vreme, tj. nema protok paketa, SDN svič će izgubiti njegove tokove translacije radi što optimalnijeg rada. Međutim, problem može da nastane u trenutku kada se korisnik reaktivira. Njegov paket će ponovo uhvatiti HD i javiti MD-u, koji će napraviti proaktivan tok. Međutim, sve to procesiranje se obavlja asinhrono od

procesiranja paketa koji se prosleđuje od korisnika do Core sloja. Zato će korisnički paket stići do Core rutera znatno pre nego što se završi asinhrono procesiranje i stigne proaktivni tok.

Da se ne bi desilo da se korisnički paket onda prosledi pod instrukcijom NORMAL, jer njegova instrukcija za translaciju još nije dostupna, dodaje se instrukcija koja takav paket šalje na reaktivnu obradu SDN kontroleru.

Konkretno, osim inicijalnih NORMAL instrukcija, MS proces SDN sviču inicijalno ubacuje i instrukciju **FORWARD CONTROLLER,** koja ima malo veći prioritet od instrukcije NORMAL, ali opet znatno manji nego ostale proaktivne instrukcije. Ova instrukcija se dodaje samo za *rIP* spektar IP adresa ISP provajdera.

Tada MS, osim ubacivanja inicijalnih parametara, obrađuje i pakete koji stignu reaktivno. Ta obrada je vrlo slična postupku za obradu koju vrši MD, a svodi se na dobavljanje podataka iz MST tabele i formiranje adekvatnih tokova.

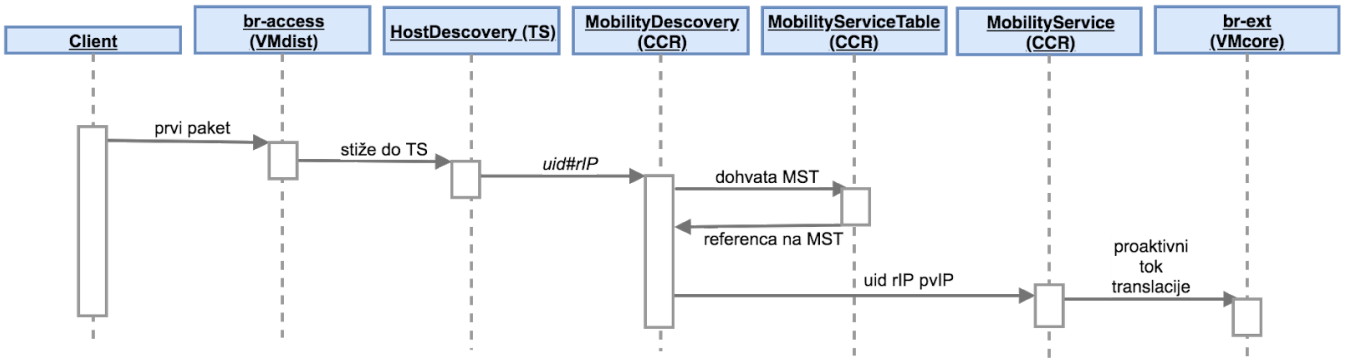
Čak i ako paket reaktiviranog korisnika stigne do Core rutera pre proaktivnog toka, Core ruter će reaktivno proslediti paket do kontrolera i sačekati njegov odgovor pre nego što obradi pristigli paket. Takvo inicijalno odlaganje je produkt SDN koncepta, ali je obično znatno manje nego slične tehnologije L3 mobilnosti.

Implementacija ove funkcionalnosti je raspoloživa kao open-source projekat hostovan na GitHub-u [55].

Simulacija L3 mobilnosti

Pre nego što se analiziraju procedure koje se izvršavaju kada korisnik pređe iz zone pokrivenosti jednog u zonu prokrivenosti drugog provajdera, potrebno je analizirati tok paketa dok se on nalazi u matičnoj mreži.

UML dijagram toka na slici 5.11. prikazuje procedure koje se asinhrono izvode prilikom dolaska prvog paketa korisnika.



Slika 5.11. UML dijagram toka za proaktivne tokove

Važno je znati da se onda kada mobilni korisnik želi da komunicira sa udaljenim korespodentom, odigravaju sledeći procesi:

- Paket mobilnog korisnika pristiže do AP uređaja unutar Access sloja predložene arhitekture;
- Isti paket se prosleđuje do distribucionog sloja gde Tap servis prihvata primerak i asinhrono vrši obradu koju čine procesi:
    - HD proces unutar TS ekstraktuje *uid#rIP*;
    - HD kešira zapis i po potrebi isporučuje podatak MD modulu CCR kontrolera;
    - MD modul proverava postojanje *uid* korisnika u MST tabeli i za njega dobavlja ili alocira *pvIP*;
    - MD posredstvom MS-a kreira proaktivne tokove translacija *rIP* u *pvIP* i obratno; i
    - kreirani tokovi se ubacuju u OpenFlow svič, tj. *br-ext* u ovom slučaju.
- Paralelno sa asinhronim procesiranjem, paket se isporučuje Core sloju arhitekture:
    - ukoliko postoje instrukcije tokova za pristigli paket, Core ruter rutira paket i vrši sNAT translirajući izvorišnu IP adresu iz *rIP* u *pvIP* po instrukciji toka;
    - ukoliko ne postoji instrukcija za taj tok (jer asinhrona obrada još nije završena), paket se prosleđuje OpenFlow kontroleru, CCR, na reaktivnu obradu. Rezultat obrade su reaktivni tokovi za sNAT i dNAT. Na osnovu tih tokova, Core ruter vrši rutiranje i transliranje izvorišnih adresa.

Prilikom povratka paketa, od korespodenta do mobilnog korisnika, proces obrade je sledeći:

- Paket pristiže do Core rutera koji je spektar *pvIP* adresa oglasio preko ruting protokola kao svoju mrežu.

- Core ruter proverava da li postoji instrukcija u tabeli tokova za konkretnu traženu *pvIP*:
    - ukoliko instrukcija postoji (najšešći slučaj), obaviće se dNAT, tj. translacija destinacione *pvIP* adrese u *rIP* po instrukcijama;
    - ukoliko instrukcija ne postoji, paket se prosleđuje CCR-u na dalju obradu, gde reaktivno MS modul proverava MST tabelu i ako nađe zapis, vraća reaktivne instrukcije tokova;
    - ukoliko se u MST tabeli ne nađe zapis koji odgovara *pvIP* adresi, paket se odbacuje jer je poslat na adresu koja još nije alocirana.
- Nakon dNAT obrade na Core ruteru, paket se dalje rutira ka odgovarajućem distribucionom ruteru, tj. provajderu, na osnovu *rIP* adrese.
- Distribucioni ruteri prenose paket u L2 tehnologiju koju koristi bežični korisnik.
- Odgovarajući AP (što je uređeno specifičnom L2 tehnologijom) prosleđuje paket krajnjem korisniku.

Da bi mogao da se testira postavljeni koncept IP mobilnosti, omogućeno je i to da VMclient u toku postojeće komunikacije promeni virtualnu mrežu na koju je zakačen. Sa mogućnošću takve promene mreže, praktično se vrši apstrakcija tehnologije na L2 sloju.

Prelazak iz jedne u drugu virtualnu mrežu (*Roaming*) realizovan je *shell* skritpom sa komandama:

```
VBoxManagecontrolvm "Client" setlinkstate1 off
VBoxManagecontrolvm "Client" nic1 intnet vm_net1
VBoxManagecontrolvm "Client" setlinkstate1 on
```

Pomenute komande simuliraju logičko isključivanje linka, prebacivanje na drugu mrežu i ponovno uključivanje linka. To je nešto što se svakako dešava u realnom okruženju, jer prilikom svakog L3 rominga dolazi do prekida i ponovnog uspostavljanja veze sa drugom mrežom. Kratak prekid koji postoji u ovom procesu je nešto što ne može da se izbegne, ali su posledice prelaska nešto što može da se ublaži.

Prikazana skripta na najbrži mogući način prebacuje link iz virtualne mreže IPS1 na virtualnu mrežu ISP2. Postupak povratka je identičan, samo što bi se koristio identifikator *vm_net0* za povratak u ISP1 mrežu.

Analiza toka paketa u trenutku kada se aktivira skript, odnosno kada mobilni korisnik pređe iz zone pokrivenosti signalom jednog ISP-a u zonu pokrivenosti drugog ISP-a, sastoji se od sledećih procesa:

- AP drugog ISP-a povezuje premeštenog mobilnog korisnika (u našoj laboratoriji to je apstrahovano procedurom iz skripte).
- Mobilni korisnik traži DHCP/RA parametre za novu mrežu, a ruter distribucionog sloja tog ISP-a vraća IP parametre te mreže.
- Asinhrono sa DHCP/RA odgovorom, HD proces unutar TS detektuje „novog" korisnika na osnovu njegovog *uid-a* i
    - ekstrahuje njegov par *uid#rIP* gde je *rIP* nova, tek dobijena IP adresa u mreži aktuelnog ISP;
    - pošto je to za HD novi zapis, odmah prosleđuje par do MD procesa unutar CCR-a;
    - MD proces proverava MST tabelu u kojoj nalazi da postoji zapis sa tim *uid* parametrima, te ažurira zapis sa novim *rIP* parametrom;
    - MD, posredstvom MS-a, formira proaktivni tok za sNAT i dNAT sa instrukcijom translacije tako da se nova *rIP* adresa translira na postojeću *pvIP* adresu;
    - MS prenosi instrukcije na OpenFlow svič koji ih zbog njihove prirode prepisuje pre već postojećih (zastarelih).
- Kada mobilni korisnik primeni novu *rIP* adresu i pošalje paket ka udaljenom korespodentu, Cloud ruter će već imati proaktivne tokove sa ažurnom adresom za sNAT translaciju. Ukoliko se čak dogodi da asinhroni poces ne bude završen kada paket već stigne do Cloud rutera, reaktivni proces će „dohvatiti" nove ažurne tokove.

U povratnom smeru je situacija vrlo slična onoj kada je korisnik još uvek u istoj mreži. Core ruter će imati ažurne proaktivne tokove vrlo brzo nakon što se mobilni korisnik pojavi sa prvim paketom u bilo kojoj mreži u domenu mobilnosti.

Kao i sva druga rešenja, i ovo predloženo rešenje nije sasvim rešilo probleme koji se mogu naći u nekoj komunikaciji. Najveći problem u praktičnoj realizaciji je vreme ponovnog povezivanja u L2 tehnolgijama pristupa. U našem eksperimentu korišćeni su skript i procedura prelaska u drugu mrežu koja ceo proces obavlja veoma brzo. Reasocijacija se izvršava za nešto manje od 100ms, što je daleko od realnih pristupnih tehnologija i bežičnih mreža.

Ipak, ovo je rad na rešenju L3 mobilnosti koji se oslanja na IP orijentisane tehnologije, pa se nije ulazilo dublje u načine optimizacije različitih L2 tehnolgija za bržu reasocijaciju.

Sam DHCP takođe uvodi određeno kašnjenje u reasocijaciji na L3 sloju. Dok računar ne dobije parametre sa kojima može da radi u toj mreži, on ne šalje niti prima pakete. Zbog svoje kompleksnosti i zastarelosti, DHCP se u novijim verzijama IP protokola, IPv6, zamenjuje sa RA procedurom koja je znatno brža i koja dodatno optimizuje vreme reasocijacije na L3 sloju.

Oba opisana problema reasocijacije imaju za posledicu izgubljene pakete dok se ne obavi ponovno povezivanje. TCP mehanizmi ublažavaju ovaj efekat, dok se kod UDP komunikacije mogu primetiti određeni artifekti izgubljenih paketa. U ovom eksperimentu reasocijacija je trajala veoma kratko, pa su gubici bili minimalni i gotovo neprimetni u testovima sa različitim servisima.

Još jedan negativan uticaj ovog rešenja je i to što će procesiranje paketa, ukoliko paket stigne do Core rutera pre proaktivnog toka, biti odloženo dok ne dođe reaktivni tok. To uvodi inicijalno povećano kašnjenje u isporuci prvog paketa, ali ne utiče bitno na performanse samog sistema jer se većina tokova obrađuje proaktivnim tokovima.

Rezultati testiranja i performanse rešenja

Za potrebe testiranja postavljenog eksperimenta korišćen je alat za testiranje raspoloživog opsega i performanci *iperf3* [43]. Serverska strana je postavljena na računar korespodenta, tj. host računar izvan domena mobilnosti. Klijentski deo test servisa je pokretan na VMclient mašini. Osim rezultata merenja propusnog opsega, radi detaljnije analize performansi rešenja, aktiviran je i *tcpdump* servis za prihvatanje i beleženje paketa u *pcap* formatu.

Alat *iperf3* korišćen je kod UDP prenosa jer je UDP dodatno osetljiv na gubitke paketa zbog nepostojanja mehanizama za pouzdanu isporuku kao kod TCP-a.

Na serverskoj strani, *iperf3* alat je pokrenut komandom:

```
iperf3 -s -D
```

dok su na VMclientu testovi izvršeni komandama:

```
iperf3 -c <ip host računara> -u -i 1
i
iperf3 -c <ip host računara> -u -i 1 -R
```

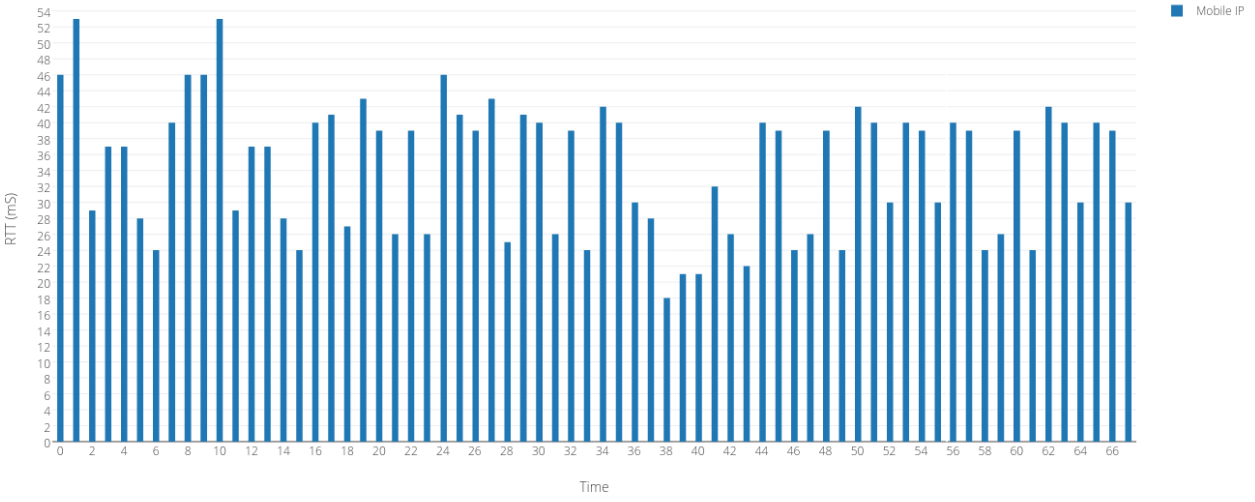
Prilikom klijentskog poziva *iperf3* servisa, osim IP adrese *iperf3* servera, aktivirano je UDP testiranje (-u) kao i izveštavanje na svaku sekudnu (-i 1). Poziv bez reverznog parametra (-R) izaziva testiranje u jednom smeru i to uploada, tj. propusnog opsega sa VMclienta na korespodenta. Korišćenjem reverznog parametra, testira se download smer, tj. propusni opseg od korespodenta ka VMclientu.

Snimanje saobraćaja na VMclientu i korespodentu, radi dalje analize, obavljeno je komandom:

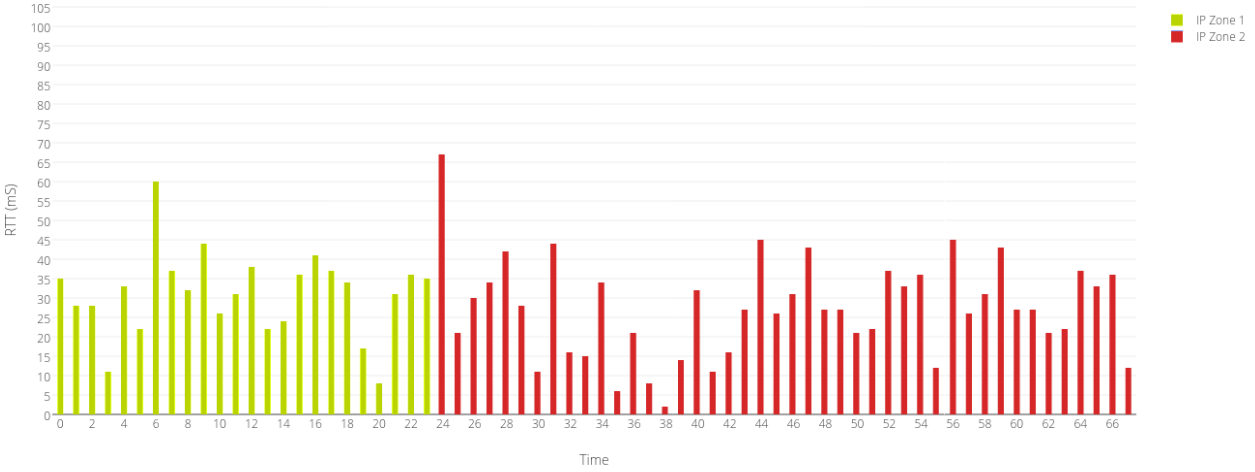
```
tcpdump -i eth0 -w recoding.pcap
```

Snimljeni paketi *recording.pcap* kasnije su korišćeni u analizi podataka procesiranjem kroz Wireshark [53].

Podaci su prikupljeni za vreme aktiviranja simulacija prelaska VMclient računara između dve mreže. Prvi rezultati su pokazali da se predloženim postupkom može dobiti neprekidiva kontinualna komunikacija između klijenta i servera.



Slika 5.12. Prikaz varijacija RTT vrednosti iz perspektive korespodenta



Slika 5.13. Prikaz varijacija RTT vrednosti iz perspektive mobilnog korisnika

Predstavljeni dijagrami su grafovi toka paketa u simuliranom okruženju koji prikazuju RTT, tj. vreme isporuke paketa iz perspektive VMclienta i perspektive korespodenta.

Ovi prvi testovi pokazuju da postoji kontinualnost u prenosu UDP paketa, i da iz perspektive korespodenta u načinu prenosa nije došlo do bilo kakve promene u načinu prenosa paketa.

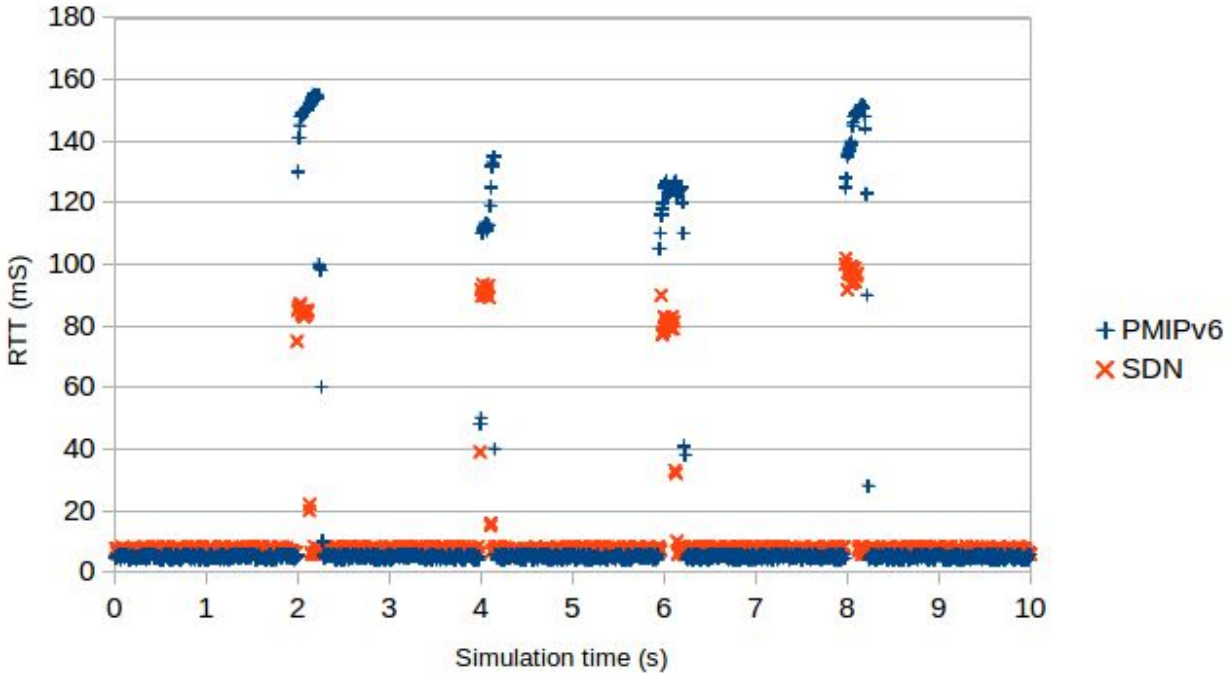
Prvi, inicijalni test, urađen je sa namerom da se proveri i ustanovi postojanje transparentne mobilnosti i neprekidivosti sesije. Međutim, njime nije dobijeno dovoljno podataka sa kojima bi kvalitet rešenja mogao da se uporedi sa kvalitetom drugih rešenja i standarda.

Zato je test ponovljen primenom snimanja u mnogo višoj rezoluciji, tj. snimanjem većeg broja paketa u sekundi i poređenjem performanse sa drugim rešenjem opisanim u poglavlju o sličnim istraživanjima. Konkretno, rezultat testa upoređen je sa standardom PMIPv6 [12], koji je opisan u drugom poglavlju.

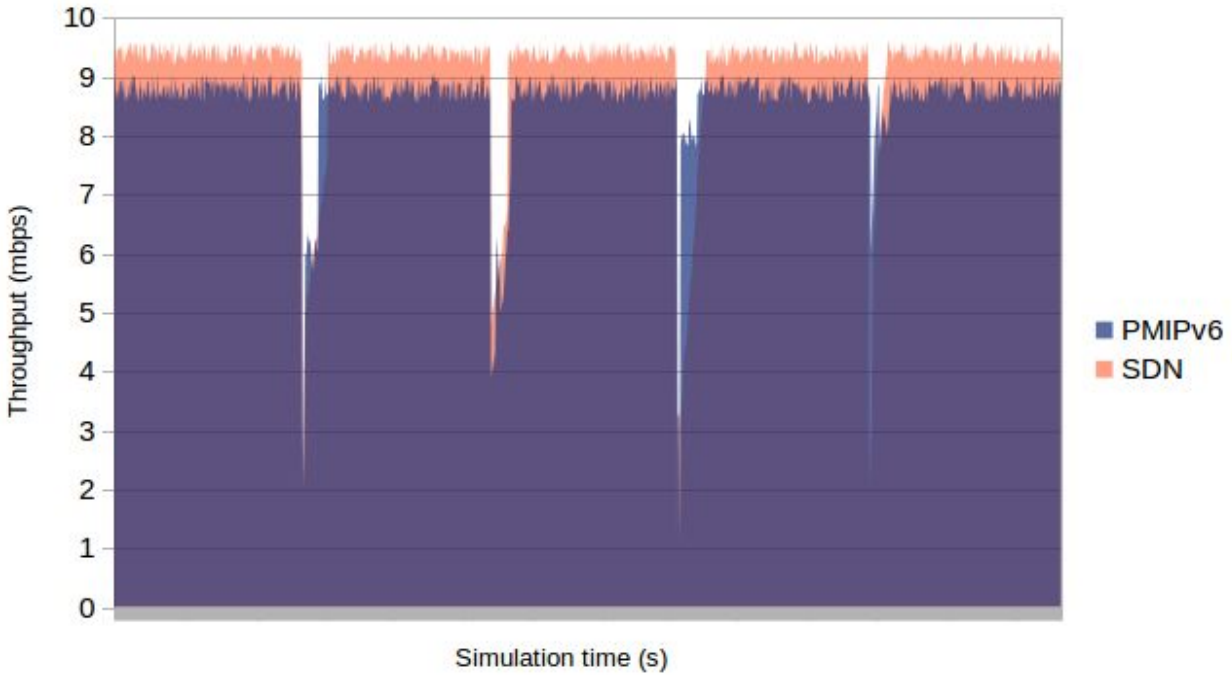
Da bi se u istom okruženju i na isti način testirala PMIPv6 implementacija i novo rešenje za IP mobilnost, izvršena je migracija testnog scenarija u Mininet okruženje [23] i u tu svrhu je korišćen Mcproxy [54] open-source projekat. Migracija novog rešenja za IP mobilnost u Mininet bila je prilično jednostavna zbog mogućnosti da se kroz Mininet koriste OVS i eksterni kontroler. Konfiguracija novog rešenja u Mininet okruženju dostupna je na GitHub-u [55].

Prilikom testiranja korišćena je i identična procedura *iperf3* alatom. Kreiran je skript koji vrši četiri prelaska iz jedne u drugu mrežu simulirajući kretanje mobilnog korisnika. Skript bi prilikom pokretanja čekao dve sekunde, zatim bi prešao u mrežu 2, sačekao dve sekunde, pa bi se vratio u mrežu 1, sačekao dve sekunde, prešao bi u mrežu 2 i posle dve sekunde opet bi se vratio u mrežu 1.

Ista simulacija obavljena je i za PMIPv6 i novo rešenje, a dobijeni rezultati prikazani su na slikama 5.14. i 5.15.



Slika 5.14. Poređenje kašnjenja između našeg i PMIPv6 rešenja



Slika 5.15. Poređenje maksimalnog propusnog opsega između dva rešenja

Na slici 5.14. nalazi se uporedni prikaz dva algoritma za L3 mobilnost gde se može videti da PMIPv6 znatno više kasni u ažuriranju stanja pre nastavljanja regularnog toka podataka. S druge strane, kašnjenje postoji i kod novog rešenja, jer asinhroni proaktivni tok ne stiže uvek da se ažurira pre prvog paketa nakon tranzicije. Tada reaguje reaktivni mehanizam i šalje kontroleru paket na obradu, što dovodi do određenog kašnjenja. Ono je ipak mnogo kraće od PMIPv6 jer je procedura mobilnosti znatno jednostavnija.

Na slici 5.15. nalazi se uporedni prikaz maksimalnog propusnog opsega u trenutku simulacije. Zbog zakasnelih inicijalnih paketa, dok se stanja ne ažuriraju, oba mehanizma nakratko uvode pad u maksimalnom propusnom opsegu. Manja kašnjenja novog rešenja prave relativno malu prednost u odnosu na PMIPv6 kada se radi o maksimalnom propusnom opsegu u trenutku L3 mobilnosti. Međutim, novo rešenje ima značajnu prednost naspram PMIPv6 u trenutku stabilnih stanja (tj. kada korisnik koristi mrežu u jednoj zoni), zbog toga što se ne koriste dodatna enkapsulacija i tunelovanje. *Overhead* koji dodaje *tuneling* protokol kod PMIPv6 direktno utiče na maksimalni propusni opseg protokola i smanjuje njegovu efikasnost iskorišćenja linka.

Iz priloženih rezultata možemo zaključiti da su prednosti novog rešenja naspram PMIPv6 protokola jasno vidljive u pogledu bržeg postizanja stabilnog stanja i normalizacije kašnjenja isporuke paketa, kao i da je značajno bolje iskorišćenje raspoloživih linkova.

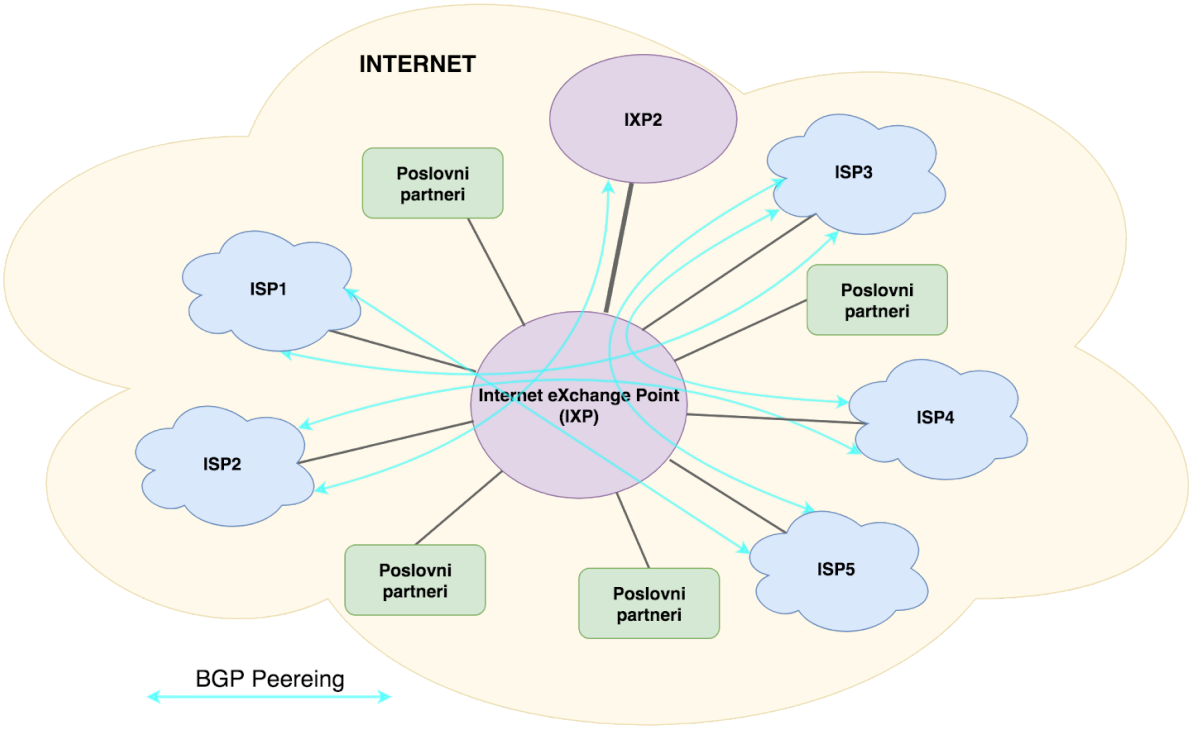
Buduća istraživanja

Mobility eXhange Point (MXP)

Fokus ovog istraživanja morao je da bude usmeren ka razvoju modela rešenja koje bi moglo biti jednostavno primenjeno. Osim tehnološkog rešenja, akcenat u okviru istraživanja koja su u vezi sa rešavanjem problema mobilnosti bio je na usklađivanju novog rešenja sa poslovnom logikom internet provajdera. Činjenica je da upravo u tome leži glavni problem primene predloženog tehničkog rešenja za L3 mobilnost u heterogenim bežičnim mrežama.

Uprkos očiglednim prednostima koje donosi SDN koncept umrežavanja (u pogledu većeg stepena programabilnosti, efikasnijeg upravljanja, većeg nivoa automatizacije i dr.), nova rešenja bazirana na ovoj tehnologiji nisu zaživela jer još uvek ne postoji njena masovna primena. Pitanje kolaboracije između opreme različitih proizvođača u bežičnim mrežama ključni je razlog zašto se u koncipiranje novog rešenja išlo sa strogim zahtevom za minimalne promene u mrežnoj infrastrukturi ISP-ova kako bi se postigla L3 mobilnost.

Za razliku od drugih rešenja baziranih na SDN tehnologiji, koja su propisana brojnim standardima, predloženim rešenjem se ne traži specijalizovani hardver i ne insistira se na primeni uređaja za upravljanje SDN mrežom koji u praksi nisu potvrđeni. Kada se pažljivo analiziraju predlozi za realizaciju drugih rešenja koja se baziraju na SDN konceptu, može se jasno zaključiti da je za njihovu implementaciju potrebna zamena tradicionalne mreže, odnosno njenih uređaja, SDN opremom. To predstavlja veliku investiciju, tj. ogroman trošak za kompanije koje su već uložile znatna sredstva u nabavku opreme tradicionalnih mreža. Umesto toga, rešenje koje je rezultat ovog istraživanja zahteva instalaciju samo jednog dodatnog fizičkog ili virtuelnog računara, tzv. *Tap* servera. Ovaj bi računar izvršavao softver za kolekciju parametara korisnika, kao što su njegova MAC i IP adresa, i bio bi dovoljan za celu mrežu ISP-a.



Slika 6.1. Intenret eXchange Point (IXP)

Analizirani su poslovni modeli različitih organizacija, kako bi se ISP-ovi međusobno povezali i na taj način optimizovali međusobne putanje. *Internet eXchange Point* (IXP) predstavlja prilično bitan pomak u objedinjavaju pružalaca internet usluga u jednu veliku heterogenu sredinu. Zahvaljujući IXP organizacijama, saobraćaj unutar jedne države ne mora da napušta granice te države iako su ISP-ovi povezani sa različitim inostranim provajderima. IXP predstavlja tu tačku saradnje među konkurentnim ISP kompanijama. Ukoliko još i IXP pruži dodatne usluge, kao što je keširanje nekih CDN sadržaja, ISP kompanije imaju dodatni motiv da ostvare kvalitetnu vezu sa njim i omoguće svojim korisnicima bolje performanse internet saobraćaja.

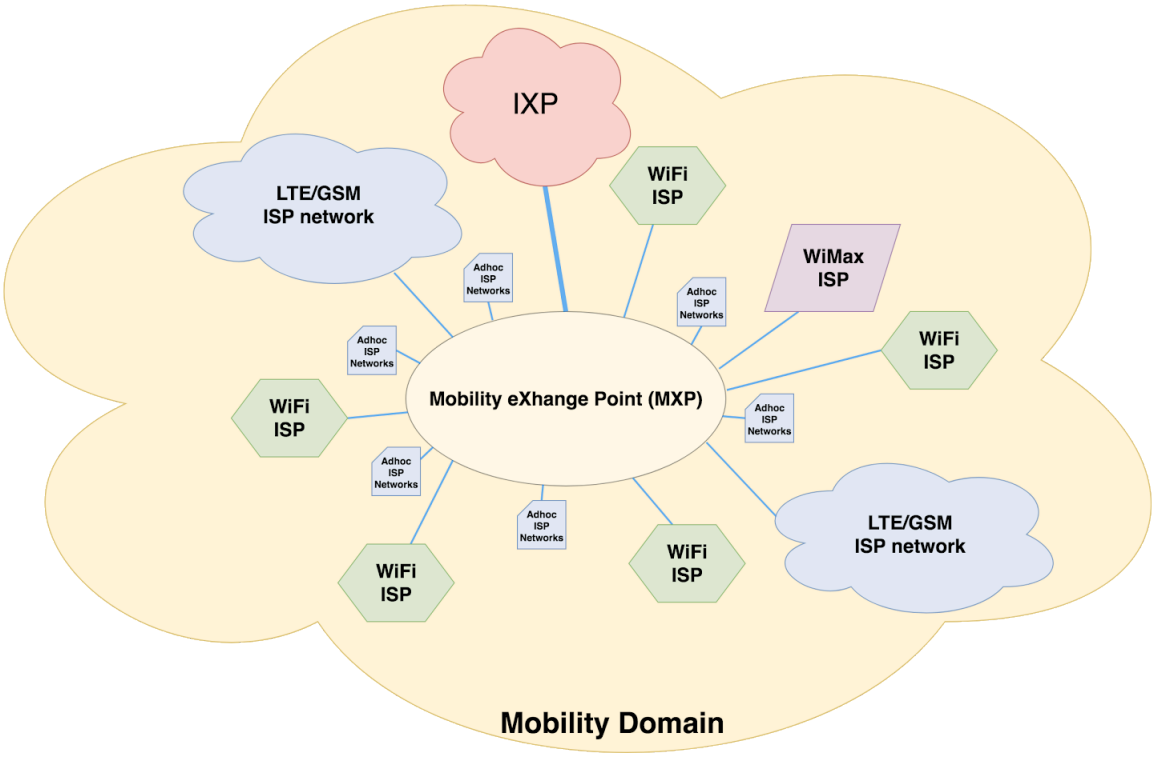
U Srbiji, kompanija SOX [56] pruža usluge IXP-a i veliki deo domaćih ISP-ova su već njihovi korisnici. SOX ostvaruje partnerstvo sa velikim kompanijama kao što su Nethood i ICANN, RIPE i Vrisign, čiji su kvalitetni linkovi bitni zbog hostovanja root DNS servera, kao i sa CDN kompanijama: Google, LimeLight i Akamai. Takođe je veoma važno napomenuti i da je SOX povezan i sa velikim mrežnim čvorištima kao što su Hurricane Electric, CloudFlare i Valve. Upravo ovakva povezanost sa velikim internet mrežnim čvorištima motivisala je ISP kompanije da i same postanu korisnici IXP usluga.

Može se smatrati da je biznis model kakav je SOX koristio dobro rešenje za prevazilaženje problema lokalnosti internet saobraćaja (da tokovi između provajdera u Srbiji ostanu u Srbiji, umesto da koriste međunarodne linkove). I taj problem je doskora predstavljao veliki izazov iz prostog razloga što je uspostavljanje saradnje među konkurentskim kompanijama prilično problematično.

Zato, prateći slična iskustva, razvijamo sopstveni model koji bi, koristeći naše rešenje za IP mobilnost, trebalo da omogući heterogenu bežičnu mobilnost korisnika između različitih ISP-ova.

Model se sastoji od koncepta formiranja treće organizacije koja bi bila zadužena za IP mobilnost. Mobility eXhange Point (MXP) bi bila organizacija koja bi izvršavala Core sloj našeg rešenja. Uvođenjem treće instance, izbegava se direktna poslovna saradnja među konkurentnim kompanijama.

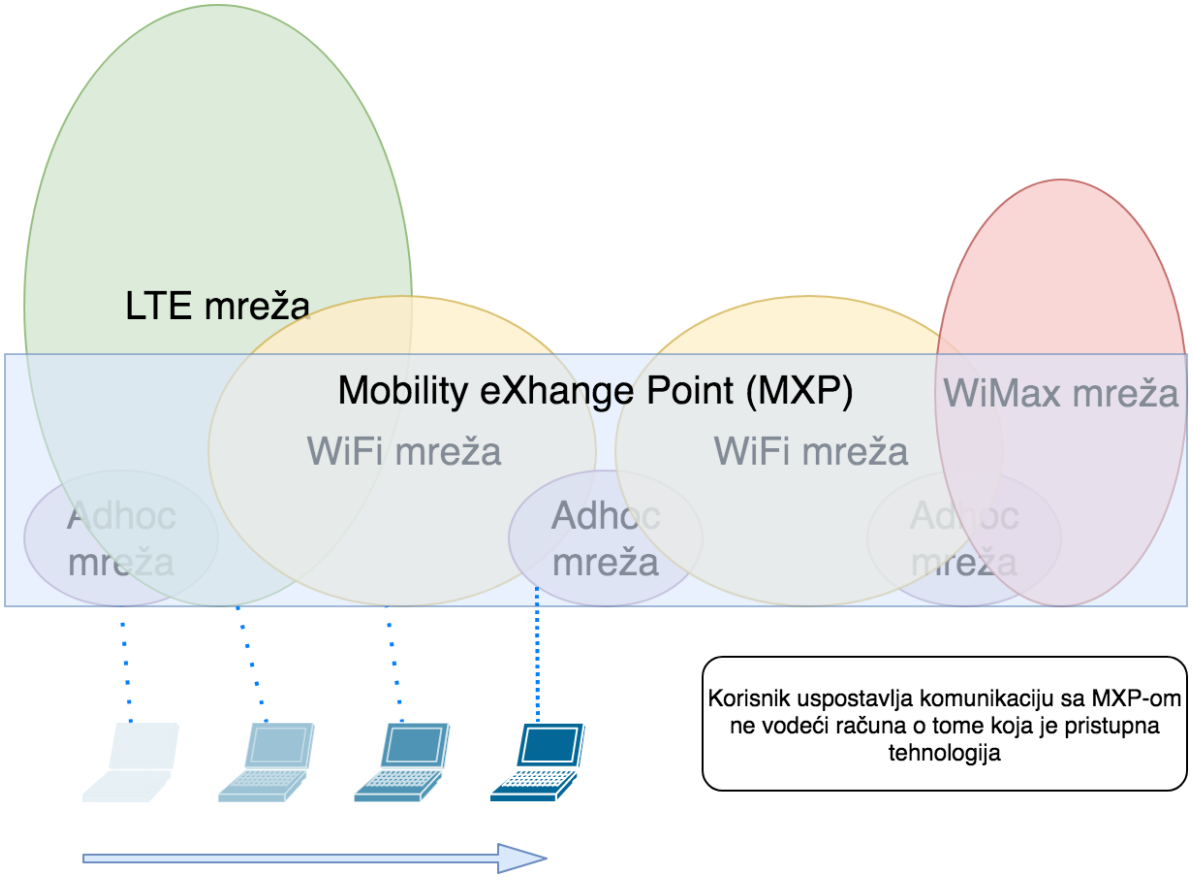
Tehnološki se sva potrebna SDN implementacija primenjuje samo u MXP-u, čime se izbegavaju gotovo svi dodatni troškovi ISP-a.



Slika 6.2. Koncept Mobility eXhange Point (MXP)

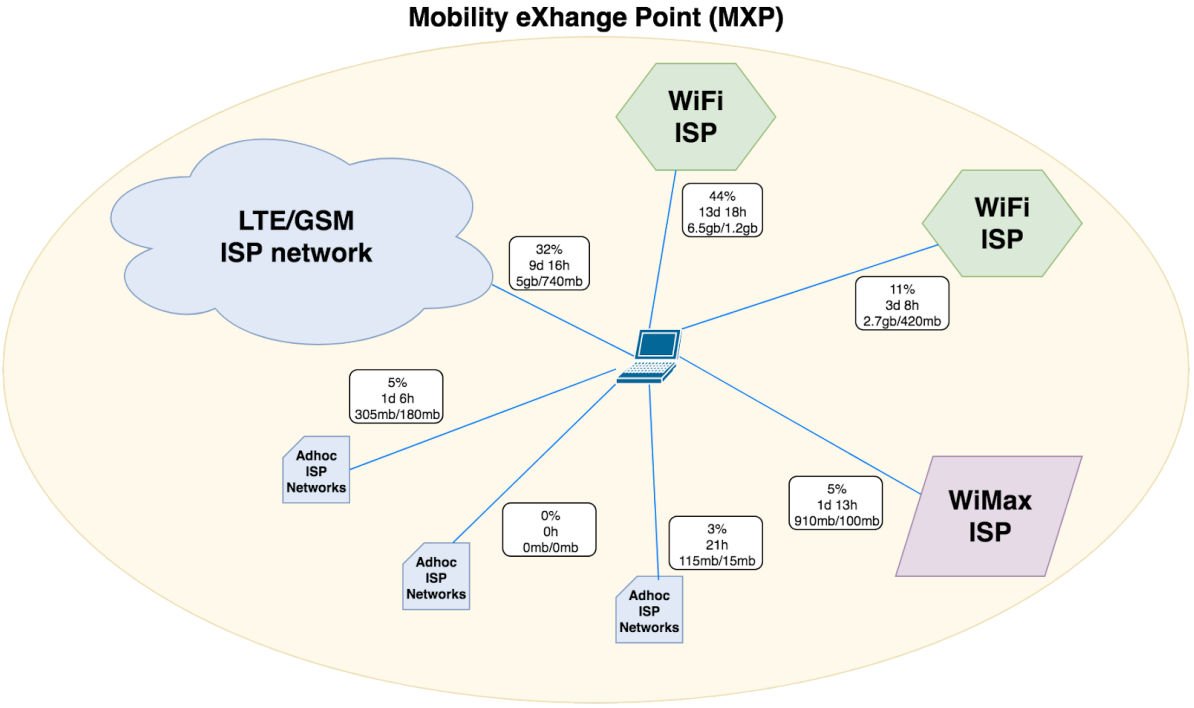
Ulaskom svakog novog ISP-a u mrežu MXP omogućava se proširenje domena pokrivenosti L3 mobilnosti.

MXP bi kao koncept mogao potpuno da odvoji uslugu od tehnologije pristupne mreže. Za korisnika je onda sasvim nebitno preko koje se tehnologije povezuje. Takođe mu je prilično nebitno i preko kog se ISP-a povezuje.



Slika 6.3. Odvajanje usluge od transportne mreže

Sistem naplate bi mogao da se realizuje tako da korisnik plaća korišćenje celog domena bežične mobilnosti (bez obzira na ISP). MXP bi posredstvom SDN-a vodio detaljnu analitiku mobilnosti, prenosa podataka, vremena korišćenja mreže i sl., kako bi adekvatno rasporedio profit ostvaren korisničkom pretplatom prema ISP članicama MXP domena.



Slika 6.4. Primer MXP sistema naplate usluge

Sve je ovo još uvek ideja u razradi, tako da se i dalje radi na analizi modela iz različitih poslovnih perspektiva.

Postoje još neki tehnološki izazovi na tom putu, a jedan od njih je i problem mapiranja različitih unikatnih parametara istog korisnika za različite tehnologije pristupa (WiFi MAC, WiMax MAC, IMSI i sl.).

Zaključak

Potreba za održanjem mrežne komunikacije čak i u trenutku kretanja korisnika kroz heterogene mreže ima visok prioritet u rešavanju problema IP mobilnosti, posebno sa aspekta mreža sledeće generacije, tzv. pametnih mreža (engl. *Smart networks*). Analizirajući postojeća rešenja ovog problema, može se zaključiti da svako od njih poseduje određene nedostatke koji mogu biti tehničke i/ili fiinansijske (investicione) prirode. Upravo iz tih razloga, nijedno od ponuđenih rešenja nije masovno zaživelo u praksi i problem mobilnosti i dalje ostaje prisutan kod mobilnih korisnika koji se premeštaju iz jedne u drugu bežičnu mrežu.

Tradicionalne, hijerarhijski organizovane mreže, karakteriše funkcionisanje u skladu sa ustaljenim pravilima TCP/IP protokola i velika zavisnost primene bilo kojeg rešenja ili novog servisa od karakteristika specijalizovanog hardvera. Drugim rečima, nemogućnost da se u mrežu unese veći stepen programabilnosti, odnosno odsustvo fleksibilnosti u primeni efikasnijeg rešenja, predstavlja osnovni problem za rešavanje IP mobilnosti.

S druge strane, SDN pruža visok nivo fleksibilnosti u pogledu upravljanja mrežom, ali i iziskuje troškove nabavke i zamene postojeće opreme sa SDN sposobnim svičevima i kontrolerima. Danas za IP mobilnost postoji veći broj rešenja koja se baziraju na SDN konceptu, ali nijedno od njih nije šire primenjeno jer zahteva značajne promene u mrežnoj infrastrukturi (što često predstavlja složen proces) i znatna finansijska ulaganja.

Opisano rešenje na prvom mestu rešava tehnološki problem IP mobilnosti, ali to čini vodeći računa o troškovima realne implementacije rešenja. Ono kombinuje rasprostranjenost tradicionalnih mreža i fleksibilnost SDN mreža stvarajući veliku hibridnu mobilnu mrežu. Provajderi bežičnih usluga ne moraju dodatno da ulažu u SDN sposobnu opremu da bi svojim korisnicima pružili IP mobilnost. Potrebno je samo da budu deo domena mobilnosti (slika 6.3) kako bi korisnicima omogućili punu IP mobilnost.

Arhitekturom rešenja (slika 4.9) predviđeno je da SDN funkcionalnost IP mobilnosti bude realizovana na Core sloju koji može biti i zasebna organizacija (slika 6.2). Održanje komunikacije usled prelaska između različitih mreža postiže se perzistentnim mapiranjem mobilnog korisnika na njegovu jedinstvenu virtualnu IP adresu, bez obzira na to u kojoj se bežičnoj mreži nalazi.

Rezultati testiranja opisanog rešenja, prikazani u eksperimentu 3, ukazuju na veću efikasnost rešenja u odnosu na PMIPv6 i druge standardizovane protokole. Veća efikasnost se ogleda u tome što se ne koristi dodatna enkapsulacija i time ne uvodi dodatni *overhead* kao u rešenju sa tunelima između domaće i strane mreže. Praktično, opisani koncept u potpunosti izbacuje pojmove domaće i strane mreže, čime se rešavaju brojni problemi u optimizaciji rutiranja.

Reference

[1] R. Braden, *Requirements for Internet Hosts Communication Layers*, Internet Engineering Task Force, RFC1122, October 1989, https://tools.ietf.org/html/rfc1122

[2] J. Postel, *Internet Protocol – DARPA Internet Program _ Protocol Specification*, Defense Advanced Research Projects Agency, Information Sciences Institute University of Southern California, RFC791, September 1981, https://tools.ietf.org/html/rfc791

[3] J. Postel, *Transmission Control Protocol _ DARPA Internet Program _ Protocol Specification*, Defense Advanced Research Projects Agency, Information Sciences Institute University of Southern California, RFC793, September 1981, https://tools.ietf.org/html/rfc793

[4] J. Postel, *User Datagram Protocol*, Information Sciences Institute University of Southern California, RFC768, August 1980, https://tools.ietf.org/html/rfc768

[5] L. Zhang, R. Wakikawa, Z. Zhu, *Support mobility in the global internet*, MICNET '09 Proceedings of the 1st ACM workshop on Mobile internet through cellular networks; 2009, DOI 10.1145/1614255.1614257, https://www.researchgate.net/publication/228917008_Support_mobility_in_the_global_Internet

[6] C. Perkins, *IP Mobility Support for IPv4*, Nokia Research Center, RFC3344, August 2002, https://tools.ietf.org/html/rfc3344

[7] S. Deering, R. Hinden, *Internet Protocol, Version 6 (IPv6) _ Specification,* Cisco and Nokia, RFC2460, December 1998, https://tools.ietf.org/html/rfc2460

[8] M. Grayson, K. Shatzkamer, K. Wierenga, *Building the Mobile Internet*, Cisco Press, Januar 2011, Part of the Networking Technology series.

[9] C. Perkins, *IP Mobility Support*, IBM, RFC2002, October 1996, https://tools.ietf.org/html/rfc2002

[10] G. Montenegro, *Reverse Tunneling for Mobile IP, revised,* Sun Microsystems, RFC3024, January 2001, https://tools.ietf.org/html/rfc3024

[11] D. Johnson, C. Perkins, J. Arkko, *Mobility Support in IPv6*, Rice University and Nokia Research Center and Ericsson, RFC3775, June 2004, https://tools.ietf.org/html/rfc3775

[12] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, *Proxy Mobile IPv6*, Cisco and Wichorus and Starent Networks and Nokia, RFC2513, August 2008, https://tools.ietf.org/html/rfc5213

[13] *PMIP: Multipath Support on MAG and LMA*,Cisco IOS XE 3S _ Configuration Guides, Cisco, April 2016, https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mob_pmipv6/configuration/xe-3s/mob-pmipv6-xe-3s-book/imo-pmipv6-multipath-support.html

[14] R. Wakikawa, S. Gundavelli, *IPv4 Support for Proxy Mobile IPv6*, Toyota ITC and Cisco, RFC5844, May 2010, https://tools.ietf.org/html/rfc5844

[15] S. Gundavelli, X. Zhou, J. Korhonen, G. Feige, R. Koodli, *IPv4 Traffic Offload Selector Option for Proxy Mobile IPv6*, Cisco and ZTE Corporation and Renesas Mobile, RFC6909, April 2013, https://tools.ietf.org/html/rfc6909

[16] L. Kyoung-Hee, *Mobility Management Framework in Software Defined Network*, *International Journal of Software Engineering & Its Applications*, 2014, DOI 10. 14257/ijseia.2014.8.8,01, https://www.researchgate.net/publication/289919646_Mobility_management_framework_in_software_defined_networks

[17] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, A. Vasilakos, *Software-Defined and Virtualized Future Mobile and Wireless Networks*: A Survey, Journal *Mobile Networks and Applications*, February 2015, DOI 10.1007/s11036-014-0533-8, https://www.researchgate.net/publication/265252430_Software-Defined_and_Virtualized_Future_Mobile_and_Wireless_Networks_A_Survey

[18] L. Li, Z. Mao, J. Rexford, *Toward Software-Defined Cellular Networks*, Software Defined Networking (EWSDN), 2012 European Workshop, DOI 10.1109/EWSDN.2012.28, https://www.researchgate.net/publication/261075906_Toward_Software-Defined_Cellular_Networks

[19] L. Valtulina, M. Karimzadeh, G. Karagiannis, G. Heijenk, A. Pras, *Applying SDN /OpenFlow in Virtualized LTE to support Distributed Mobility Management (DMM)*, Globecom Workshops (GC Wkshps), 2014, DOI 10.1109/GLOCOMW.2014.7063379, https://www.researchgate.net/publication/276289611_Performance_Evaluation_of_a_SDNOpenFlow-Based_Distributed_Mobility_Management_DMM_Approach_in_Virtualized_LTE_Systems

[20] Z.Qin, G. Denker, C. Giannelli, P. Bellavista, N.Venkatasubramanian, *A Software Defined Networking architecture for the Internet-of-Things*, Network Operations and Management Symposium (NOMS), 2014, DOI 10.1109/NOMS.2014.6838365, https://www.researchgate.net/publication/269300754_A_Software_Defined_Networking_architecture_for_the_Internet-of-Things

[21] D. Wu, D. Arkhipov, E. Asmare, Z. Qin, J. McCann, *UbiFlow: Mobility Management in Urban-scale Software Defined IoT*, Computer Communications (INFOCOM), 2015 IEEE Conference, DOI 10.1109/INFOCOM.2015.7218384, https://www.researchgate.net/publication/282723441_UbiFlow_Mobility_Management_in_Urban-scale_Software_Defined_IoT

[22] Y. Wang, J. Bi, K. Zhang, *Design and Implementation of a Software-Defined Mobility Architecture for IP Networks*, Journal *Mobile Networks and Applications*, February 2015, DOI 10.1007/s11036-015-0579-2, https://www.researchgate.net/publication/272401605_Design_and_Implementation_of_a_Software-Defined_Mobility_Architecture_for_IP_Networks

[23] Mininet: an instant virtual network on your laptop (or other PC), http://mininet.org/

[24] P. Srisuresh, M. Holdrege, *IP Network Address Translator (NAT) Terminology and Considerations,* Lucent Technologies, RFC2663, August 1999, https://tools.ietf.org/html /rfc2663

[25] Y. Rekhte, B. Moskowitz, D. Karrenberg, G. de Groot, *Address Allocation for Private Internets,* IBM and Chrysler and RIPE NCC, RFC1597, March 1994, https://tools.ietf.org/html/rfc1597

[26] E. Gerich, *Unique Addresses are Good*, Merit Network, RFC1814, June 1995, https://tools.ietf.org/html/rfc1814

[27] F. Audet, C. Jennings, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*, Nortel Networks and Cisco Systems, RFC4787, January 2007, https://tools.ietf.org/html/rfc4787

[28] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, *Address Allocation for Private Internets,* Cisco Systems and Chrysler Corp and RIPE NCC and Silicon Graphics, RFC1918, February 1996, https://tools.ietf.org/html/rfc1918

[29] M. Smith, R. Hunt, *Network security using NAT and NAPT*, Proceedings 10th IEEE International Conference on Networks (ICON 2002), 2002, DOI:10.1109/ICON.2002.10 33337, https://www.researchgate.net/publication/3967922_Network_security_using_NAT_and_NAPT

[30] S. Deering, R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification,* Check Point Software, RFC8200, July 2017, https://tools.ietf.org/html/rfc8200

[31] Ž. Bojović, V. Šenk, D. Dobromirov, P. Bojović, *Intervendor working of voip networks*, *The ITP Journal*, Volume 5, Part 3, 2011, https://www.researchgate.net/publication/298957307_INTERVENDOR_WORKING_OF_VOIP_NETWORKS

[32] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, *OpenFlow: enabling innovation in campus networks,* SIGCOMM Comput. Commun. Rev. 38, April 2008, 69_74, DOI 10.1145/1355734.1355746, https://www.researchgate.net/publication/220195143_OpenFlow_Enabling_innovation_in_campus_networks

[33] *Open Datapath _ OpenFlow specification,* Open Networking Foundation, Member of the Linux Foundation, https://www.opennetworking.org/technical-communities/areas/specification/open-datapath/

[34] *OpenFlow Switch Specification _ Version 1.0.0,* Open Networking Foundation, December 2009, http://www.opennetworking.org/wp-content/uploads/2013/04/openflow -spec-v1.0.0.pdf

[35] *Service Name and Transport Protocol Port Number Registry,* Internet Assigned Numbers Authority, OpenFlow assignment for 6653 TCP/UDP port, July 2013, https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?search=6653

[36] P. D. Bojović, Ž. Bojović, D. Bajić, V. Šenk, *Intervendor working of voip networks IP Session continuity in heterogeneous mobile networks using Software Defined* Networking, *Journal of Communications and Networks* 19(6): 563_568, December 2017. DOI 10.1109/JCN.2017.000096, https://www.researchgate.net/publication/321095480_IP_Session_continuity_in_heterogeneous_mobile_networks_using_Software_Defined_Networking

[37] P. Bojović, Ž. Bojović, V. Šenk, *Method of providing seamless communication in heterogenous mobile networks; Postupak obezbeđivanja neprekidne komunikacione sesije u heterogenim mobilnim mrežama*, Zavod za intelektualnu svojinu Republike Srbije, patent pod brojem П-2016/0585, *Glasnik intelektualne svojine,* br. 1/2018, 31.1.2018, str.19_20, http://www.zis.gov.rs/upload/documents/pdf_sr/pdf/glasnik/GIS_2018/Glasnik_2018_1.pdf

[38] *OpenVSwitch (OVS),* Multilayer Open Virtual Switch, Linux Foundation Collaborative Project, https://www.openvswitch.org/

[39] *OpenStack,* Cloud operating system, https://www.openstack.org/

[40] B. Pfaff, B. Davie, *The Open vSwitch Database Management Protocol,* VMWare, RFC7047, December 2013, https://tools.ietf.org/html/rfc7047

[41] *Soekris Engeneering net4801 series boards and systems,* Soekris Engineering, April 2004, http://soekris.com/media/manuals/net4801_manual.pdf

[42] W. Miller, *The Desktop Files Network-Booting Windows,* Microsoft TechNet, July 2008, https://technet.microsoft.com/en-us/library/2008.07.desktopfiles.aspx

[43] J. Dugan, S. Elliott, B. A. Mah, J. Poskanzer, K. Prabhu, *iPerf3, tool for active measurements of the maximum achievable bandwidth on IP networks,* https://iperf.fr/

[44] Project *POX,* Networking software platform written in Python, https://github.com/noxrep o/pox

[45] M. Henry, D. Koeppen, E. Dittert, V. Viswanathan, *Intel Preboot Execution Environment*, Intel and Bootix, RFC DRAF document, https://tools.ietf.org/html/draft-henry-remote-boot-protocol-00

[46] L. Avramov, *The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology*, Cisco Press. p. 43, December 2014.

[47] *Unified Extensible Firmware Interface*, Software interface between an operating system and platform firmware, http://www.uefi.org/uefi

[48] *Tftpd32,* Small Windows open-source service for DHCP, TFTP, DNS, SNTP and Syslog servers as well as a TFTP client, http://tftpd32.jounin.net/

[49] Ž. Bojović, P. D. Bojović, J. Šuh, *The implementation of Software Defined Networking in enterprise networks, The ITP Journal*, March 2018, https://www.researchgate.net/publication/324476332_The_implementation_of_Software_Defined_Networking_in_enterprise_networks

[50] *Project Floodlight,* Open SDN Controller, http://www.projectfloodlight.org/

[51] *Fix compatibility with Mikrotik an OF1.0,* Floodlight fix by GitHub user Neby1992, Feb 2016, https://github.com/floodlight/floodlight/pull/639

[52] Project *jpcap,* network packet capture library for applications written in Java, Oct 2011, https://github.com/jpcap/jpcap

[53] *Wireshark,* open-source packet analyzer, https://www.wireshark.org/

[54] Project *Mcproxy and Mininet,* PMIPv6 Mininet implementation, Apr 2014, https://github.com/ mcproxy/mcproxy/tree/master/debug/mininet

[55] Project *SDN-IP-Mobility,* Implementation of SDN based IP mobility solution, July 2018, https://github.com/Paxy/SDN-IP-Mobility

[56] *Serbian Open eXchange*, Otvorena platforma za saradnju operatera i kontent provajdera, http://www.sox.rs