Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Nejc Potrebuješ

Težave pri prehodu interneta s protokola IPv4 na IPv6

diplomsko delo

univerzitetni študij računalništva in informatike

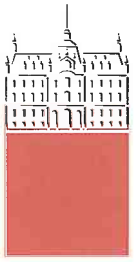
Mentor: doc. dr. Mojca Ciglarič

Ljubljana 2013

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si



Št. naloge: 01963 / 2013
Datum: 10.10.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **NEJC POTREBUJEŠ**

Naslov: **TEŽAVE PRI PREHODU INTERNETA S PROTOKOLA IPV4 NA IPV6**
PROBLEMS OF THE INTERNET TRANSITION FROM PROTOCOL IPV4 TO IPV6

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Opišite razloge, zaradi katerih protokol IPv4 več ne zadošča potrebam sodobnega interneta ter pojasnite, v čem je IPv6 boljši. Nato analizirajte današnje mehanizme in specifikacije (RFC), ki omogočajo sobivanje obeh različic protokola IP v internetu. Opišite težave pri prehodu prometa iz sveta IPv4 v svet IPv4 in nazaj ter navedite aplikacije, ki jih tovrstne težave najbolj prizadanejo. Izberite eno od implementacij prehodnih mehanizmov in na pilotni postavitvi preizkusite, koliko pripomore k združljivosti obeh različic protokola IP. Izsledke kritično komentirajte.

Mentor: M. Ciglarič
doc. dr. Mojca Ciglarič

Dekan:
prof. dr. Nikolaj Zimic

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Nejc Potrebuješ, z vpisno številko **63060271**, sem avtor diplomskega dela z naslovom:

Težave pri prehodu interneta s protokola IPv4 na IPv6

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mojce Ciglarič,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 17. oktober 2013 Podpis avtorja:

Zahvaljujem se mentorici doc. dr. Mojci Ciglarič za pomoč, nasvete ter izkazano potpežljivost pri izdelavi diplomske naloge. Posebna zahvala gre tudi moji družini in vsem ostalim, ki so me tekom študija podpirali in mi stali ob strani.

Kazalo

Povzetek

Internet trenutno sestoji iz starih omrežij IPv4, novih omrežij IPv6 ter omrežij, ki podpirajo oba omenjena protokola. Protokola IPv4 in IPv6 sta na žalost nezdružljiva, zato potrebujemo mehanizme, ki omogočijo prenos podatkov tako preko omrežja IPv4 kot tudi preko omrežja IPv6 in ki omogočajo dostop napravam, ki podpirajo samo protokolni sklad IPv4, dostopati do naprav, ki podpirajo samo protokolni sklad IPv6 in obratno. Ti mehanizmi so znani kot »Prehodni mehanizmi IPv4/v6«. V zadnjih nekaj letih je bilo razvitih mnogo takih mehanizmov.

V nadaljevanju sta najprej predstavljena protokola IPv4 in IPv6, nato sledi splošen pregled prehodnih mehanizmov IPv4/v6, na koncu pa se osredotočim na mehanizem, ki je uporaben v primeru, ko imamo na voljo aplikacijo, ki podpira le protokolni sklad IPv4 in je nameščena na napravi, ki pa ima direkten dostop le do omrežja IPv6. Z uporabo omenjeega mehanizma aplikaciji omogočimo dostop bodisi do omrežja IPv6 bodisi do omrežja IPv4. Primer take aplikacije je Skype.

Ključne besede

IPv4, IPv6, internet, prehodni mehanizmi IPv4/v6, BIH

Abstract

Internet currently coexist of older IPv4 networks, new IPv6 network and networks that supports both IPv6 and IPv4 stack. IPv4 and IPv6 networks are unfortunately incompatible. To enable communication between these two networks several mechanisms were developed. These mechanisms are so called »IPv4/v6 transition mechanisms«.

In this thesis first there are IPv4 and IPv6 protocols presented, disadvantages of an IPv4 protocol and advantages of an IPv6 protocol. Further there are some transition mechanisms presented (Dual-Stack, tunneling mechanisms and translation mechanisms). At the end there is a focus on a translation mechanism called Bump-in-the-Host which enables Dual-Stack hosts with IPv4-only applications installed to communicated over IPv6 network with IPv4 or IPv6 hosts. An example of an application that currently supports only IPv4 stack is Skype.

Keywords

IPv4, IPv6, internet, IPv4/v6 transition mechanisms, BIH

Poglavje 1

Protokol IPv4

1.1 Splošno o protokolu IPv4

Začetki medsebojnega povezovanja naprav segajo v leto 1957, ko v ZDA ustanovijo organizacijo ARPA [1] (Advanced Research Project Agency), katere namen je bil Ameriki prinesti prednost v znanosti in tehnologiji, uporabni v vojaške namene. V obdobju med letoma 1962 in 1966 je prihajalo do številnih raziskav o zmožnostih povezovanja računalniških sistemov, izdelalo pa se je tudi nekaj poskusnih mrež. Tako je bil že leta 1967 izdelan prvi načrt za omrežje ARPANET, ki pa je pričelo delovati leta 1969. Prvo vozlišče je bilo postavljeno v univerzi UCLA v Los Angelesu, isto leto pa so se priključila še tri dodatna vozlišča. Do leta 1971 je bilo v ARPANET povezanih že 15 vozlišč, leta 1972 pa so ARPANET povezali še z enim omrežjem, ALOHAnet-om, zasnovanim na havajski univerzi. Tekom sedemdesetih let se je ARPANET razširli tudi v Evropo (Anglija, Norveška), izdelani pa so bili tudi že načtri za povezovanje vozlišč v lokalna omrežja.

Časovni pregled razvoja IPv4:

- 1969: zagon DARPA za razvoj in raziskovanje omrežij
- 1981: standardizacija IPv4 (RFC 791)

- 1991: razvoj svetovnega spleta (WWW)
- 1993: objavljen CIDR
- 1994: razvit NAT
- 1996: uvedba zasebnega naslovnega prostora IP

1.2 Pomankjivosti protokola IPv4

Protokol IPv4 je bil ustvarjen v časovnem obdobju z drugačnimi omrežnimi zahtevami – omrežje je bilo v tistem času dostopno par tisoč uporabnikom. Na začetku je bil to vojaški protokol, ki je zagotavljal odporne komunikacije in neodvisno izbiro poti. Danes IP predstavlja podporo omrežjem po celem svetu in je standard sodobnih komunikacij. Velika omrežja pomenijo tudi veliko odjemalcev, kar pa pomeni nove zahteve in težave.

Za razporejanje globanih IP naslovov skrbi organizacija IANA (Internet Address and Numbering Authority). Na začetku je bil celotni prostor IPv4 razdeljen na razrede A, B in C, ki so jih nato uporabljali za dodeljevanje IP naslovov. Ravno ta razdelitev naslovnega prostora IPv4 po razredih je povzročila, da je sčasoma poraba naslovov IPv4 postala manj učinkovita, saj je bilo uporabnikom dodeljenih veliko več naslovov IPv4, kot pa so jih dejansko potrebovali. V spodnji tabeli so prikazani začetni trije razredi naslovov IPv4, ki so se uporabljali pri razdeljevanju posameznim organizacijam.

Tabela 1.1: Razredi naslovov IPv4

Razred	Prvi oktet naslova IPv4	Število omrežij	Število naslovo IPv4
A	0-127	129	16 777 216
B	128-191	16348	65536
C	192-223	2097152	256

Največji problem pri porabi naslovov IPv4 je vsekakor predstavljala nepravilna in neenakomerna razporejenost naslovov (npr. veliko nalovov ra-

zreda A je nerazporejenih in neuporabljenih). Teoretično je 32-bitni naslovni prostor zagotavljal približno 4.3 miljarde naslovov IPv4, v praksi pa je ta številka veliko manjša (okoli 250 milijonov), prav zaradi zgoraj omenjega problema. Danes se v omrežje priklaplja vedno več naprav (npr. brezžični telefoni, dlančniki, potovalni računalniki, hišne naprave, ipd) in vsaka od njih za uspešno povezavo v globalni internet potrebuje unikaten naslov IPv4. S pojavitvijo svetovnega spleta (angl. World Wide Web) leta 1991, se je močno povečala tudi poraba naslovov IPv4. Da bi podaljšali življensko dobo naslovov IPv4 je bilo razvitih več mehanizmov, ki so podrobneje opisani v nadaljevanju. Ti mehanizmi so:

- *DHCP*,
- *zasebni naslovni prostor*,
- *NAT za pretvarjanje naslovov IPv4* ter
- *CIDR in VLSM za segmentacijo razredov in izboljšano usmerjanje.*

1.3 Začasni mehanizmi za podaljšanje življenske dobe naslovnega prostora IPv4

1.3.1 VLSM

VLSM [2] (Variable Length Subnet Mask) je metoda, ki omogoča delitev podomrežja IPv4 na poljubno velika podomrežja. Omogoča, da nek naslovni blok razdelimo na različno velika podomrežja, kar omogoča večji izkoristek pri porabi naslovov IPv4, saj lahko velikost omrežja prilagodimo glede na potrebe.

1.3.2 CIDR

CIDR [3] (Classless Inter-Domain Routing) je metoda za alociranje naslovov IP in usmerjanje paketov IP. Cilj metode je upočasniti tako rast usmerje-

Tabela 1.2: Primer različno dolgih mask omrežja

Predpona omrežja	Maska omrežja	Št. uporabnih naslovov IPv4
/26	255.255.255.192	62
/27	255.255.255.224	30
/28	255.255.255.240	14
/29	255.255.255.248	6
/30	255.255.255.252	2

valnih tabel na usmerjevalnikih kot tudi porabo naslovov IP. Uporablja se v kombinaciji z metodo VLSM, ki je opisana zgoraj. Omogoča združevanje več blokov naslovov IP v en večji blok, kar zmanjša velikost usmerjevalnih tabel na usmerjevalnikih ini zmanjša promet v omrežju, saj se oglašuje samo eno omrežje.

1.3.3 Zasebni naslovni prostor

Zasebni naslovni prostor je bil razvit leta 1996 in je podrobneje opisan v RFC 1918 ("Address Allocation for Private Internets"). Naslovi IP iz tega naslovnega prostora se običajno uporabljajo v primerih, ko ne potrebujemo globalnega usmerjanja, npr. med napravami v domačih omrežjih oziroma v omrežjih posameznih podjetij. Zasebno omrežje je definirano s tremi bloki naslovov IPv4, ki so prikazana v spodnji tabeli.

Tabela 1.3: Privatni naslovni prostor

Razred	Začetni naslov IPv4	Končni naslov IPv4	Št. naslovov IPv4
A	10.0.0.0	10.255.255.255	16777216
B	172.16.0.0	162.31.255.255	1048576
C	192.168.0.0	192.168.255.255	65536

V domačih omrežjih ponavadi uporabljamo naslove razreda C, v večjih podjetjih pa uporabljajo naslove razredov A in B.

1.3.4 NAT

NAT [5] (angl. Netword Address Translation) je metoda, ki se uporablja za preslikovanje naslovov IP. Osnovna zamisel NAT-a je, da napravam, ki so znotraj nekega omrežja, ni potrebno dodeliti javnega naslova IPv4, temveč le zasebnega. Preslikave med javnimi in zasebnimi naslovi IPv4 se odvijajo na usmerjevalnikih, ki povezujejo domače (zasebno) omrežje z javnim (internetnim) omrežjem. Poznamo več vrst NAT preslikav:

- *Statičen NAT*,
- *NAT z uporabo "bazenov"naslovov IP* in
- *PAT*.

Statičen NAT je najpreprostejša oblika NAT-a. Primer je lahko nek strežnik v zasebnem omrežju, ki ima določen statičen naslov IPv4 (pomeni, da se v prihodnosti ne bo spreminjal). V primeru, da nekdo iz javnega omrežja želi dostopati do tega strežnika, mu s statičnim NAT-om zagotovimo, da bo ta strežnik vedno imel enak javni naslov IP.

Pri NAT-u, ki pa za preslikavo uporavlja množico javnih naslovov IP, pa se preslikava izvede dinamično. Pomeni, da moramo imeti prav toliko javnih IP naslovov kot imamo naprav v zasebnem omrežju, ki se želijo povezati v internet. Pri tej vrsti NAT-a pa naprava v zasebnem omrežju ne dobi vedno enakega javnega naslova IPv4. Primer te vrste NAT-a je na prikazan spodnji shemi.

PAT, ki ga poznamo tudi pod imenom »NAT overload«, je nadgradnja dinamičnega NAT-a, saj translacija med javnimi in zasebnimi naslovi IPv4 ni »ena na ena« ampak lahko en javni naslov IPv4 preslikamo v več zasebnih naslovov IPv4, saj se pri translaciji upošteva tudi številka vrat (port), na katerega se povezujemo. PAT je bolj primeren tam, kjer imamo na voljo malo javnih naslovov IPv4 in mnogo naprav v zasebnem omrežju, ki želijo dostopati do interneta.

1.3.5 DHCP

DHCP [4] (angl. Dynamic Host Configuration Protocol) je protokol, ki skrbi za dinamično dodeljevanje, uporavljanje in odvzemanje naslovov IP. Definiran je v RFC 2131 ("Dynamic Host Configuration Protocol"). Odjemalec DHCP preko protokola DHCP pridobi omrežni (IP) naslov, privzet prehod ter enega ali več strežnikov DNS, ki skrbijo za preslikavo domeniskih imen v naslove IP. Vlogo strežnika DHCP, ki dodeluje naslove IP odjemalcem v domačih omrežjih običajno opravlja usmerjevalnik v večjih omrežjih pa v ta namen namenjen strežnik.

Poglavje 2

Protokol IPv6

2.1 Novosti in izboljšave protokola IPv6

IPv6 vsebuje kar nekaj novosti in izboljšav, ki so podrobneje opisane v nadaljevanju:

- *Večji naslovni prostor*,
- *enostavnejša oblika glave paketa IPv6*,
- *varnost*,
- *mobilnost*,
- *agregacija* ter
- *spremembe v konfiguraciji naslova IPv6.*

2.1.1 Večji naslovni prostor

Večji naslovni prostor je eden od glavnih razlogov za uvedbo protokola IPv6. Naslov pri starem protokolu IPv4 je dolg 32 bitov, kar pomeni nekaj več kot 4 milijarde (4,294,967,296) naslovov. Ta številka se je v času uvedbe protokola IPv4 leta 1981 zdela nedosegljiva, a so že slabih 10 let kasneje začeli razmišljati o novem protokolu, ki bi imel na voljo več naslovov. Največje

vprašanje je predstavljala dolžina naslova pri novem protokolu. Predlogi so bili 64-, 128-, ali celo 160-bitni naslovni prostor.

Tako je bil leta 1995 izdan prvi RFC, ki opisuje novi protokol IP verzije 6. Dolžina naslova pri novem protokolu je 128-bitov, kar pomeni 340.282.366.920.938.463.463.374.607.431.768.211.456 različnih naslovov.

2.1.2 Enostavnejša oblika glave paketa IPv6

Večja dolžina nalova IP pomeni tudi večjo glavo paketa IPv6. Ker vsaka glava paketa IP vsebuje izvorni in ponorni naslov IP, je pri starem protokolu ta številka znašala 64 bitov, pri novem pa kar 256 bitov.

Glava paketa IPv4 vsebuje 12 polj. Tem dvanajstim poljem sledi še opcijsko polje poljubne dolžine. Osnovna dolžina glave IPv4 je 20 oktetov (bajtov), ki pa se zaradi opcijskega polja lahko poljubno poveča. V glavi paketa IPv6 je od 12 polj ostalo le še 6 polj, razlogi za odstranitev teh polj so naslednji:

- Dolžina glave (HD Len) je pri paketu IPv6 fiksna in znaša 40 oktetov (bajtov). Enako velika glava pri vseh paketih IPv6 tudi poenostavi procesiranje paketka (pri paketu IPv4 je bilo dolžino glave zaradi opcijskega polja potrebno vedno posebej izračunati)
- Fragmentacija je v protokolu IPv6 drugačna, zato polja v glavi paketa IPv6 ne potrebujemo. Pri protokolu IPv6 se usmerjevalniki ne ukvarjajo več s fragmentacijo, kar tudi zmanjša probleme, ki so se pojavili zaradi fragmentacije v protokolu IPv4. Podatki o fragmentaciji so glavi IPv6 dodani kot razširitvena glava (opis v nadaljevanju) le, če je do fragmentacije paketa dejansko prišlo.
- Odstranjeno je bilo tudi polje kontrolne vsote, saj večina protokolov na drugem nivoju (Ethernet, ...) že opravlja to operacijo. Pravtako s tem prisilimo protokole na višjem sloju (npr. UDP – User Diagram Protocol), da, do sedaj opcijsko, računanje checksuma postane obvezno.

Opcijsko polje je pri protokolu IPv6 odstranjeno iz glave in dodano na konec glave, s čimer se izognemo različnimi dolžinami glave in s tem omogočimo lažje (hitrejše) procesiranje paketov IPv6 med usmerjanjem. Ostala polja so bodisi ostala nespremenjena bodisi je prišlo do manjših sprememb.

Dolžina glave pri paketu IPv6 je 40 bajtov (oktetov). Ima manj polj kot v paketu IPv4, poravnana pa je na 64-bitov, kar omogoči hitrejše procesiranje. Vsebuje 8 polj:

- *Verzija (angl. version)*: to 4-bitno polje vsebuje številko 6 namesto številke 4 v glavi IPv4.
- *Vrsta prometa (angl. traffic class)*: 8-bitno polje enako polju »Type of service (ToS)« pri protokolu IPv4, kjer lahko označimo tip prometa, ki se pretaka skozi omrežje.
- *Oznaka podatkovnega toka (angl. flow label)*: To, novo, 20-bitno polje se uporablja za označevanje individualnih pretokov prometa z unikatnimi številkami.
- *Dolžina tovora (angl. payload length)*: To polje je podobno polju »Total Length« v protokolu IPv4, vendar je navedena le velikost podatkov v paketu, brez velikosti glave (saj ima le ta fiksno dolžino).
- *Naslednja glava (angl. next header)*: To polje opisuje tip podatkov, ki sledijo fiksni glavi IPv6. Lahko označuje paket na transportnem nivoju (TCP, UDP), lahko pa nam pove, da osnovni glavi sledijo še opcijske glave.
  - *Hop-by-Hop Option* (Next Header Value = 0): Uporablja se za prenos neobveznih informacij, ki jih pregleda vsako vozlišče na poti paketa do ponora.
  - *Fragment* (Next Header Value = 44): Z njim lahko izvorno vozlišče IPv6 pošlje paket, ki je večji, kot ga določa MTU (Maximum

Transfer Unit) na poti. MTU predstavlja največjo velikost okvirja, ki se še lahko prenese na poti do ponora.

  - *Routing* (Next Header Value = 43): Izvorno vozlišče IPv6 uporabi razširitveno glavo za navajanje vmesnih vozlišč na poti paketa do ponora.
  - *Destination Options* (Next Header Value = 60): Uporablja se za prenos neobveznih informacij, ki so namenjene ponoru določenega paketa.
  - *Encapsulation Security Payload* (Next Header Value = 60): Protokol za šifriranje vsebine.
  - *Authentication* (Next Header Value = 51): Protokol avtentikacijskega čela.

- *Omejitev skokov (angl. hop limit)*: V tem polju je navedena številka, ki določa maksimalno število preskokov med usmerjevalniki, preden se uniči.

- *Izvorni naslov (angl. source address)*: 128-bitno polje, ki označuje izvor paketa IPv6.

- *Ponorni naslov (angl. destination address)*: 128-bitno polje, ki označuje ponor paketa IPv6.

2.1.3 Varnost

Poleg večjega naslovnega prostora je razlog za uvedbo protokola IPv6 tudi varnost. Vsak sklad IPv6 ima podprto enkripcijo, kar pomeni, da lahko vsak paket IPv6 kriptiramo z uporabo varnostnega protokola IPSec (Internet Protocol Security).

2.1.4 Mobilnost

Mobilnost na tretjem nivoju je edina rešitev za naprave, ki se premikajo med omrežji, in za katere hočemo, da imajo vedno enak naslov IPv6. Princip mobilnosti je rešen s pomočjo t.i domačih agentov in mobilnih odjemalcev. Klienti, ki se pomikajo po omrežju sodelujejo z agenti, lociranimi v njihovem domačem omrežju, ki omogočajo usmerjanje paketov do njih, ne glede na lokacijo.

2.1.5 Agregacija

Večji naslovni prostor omogoča bolj strukturirano alociranje naslovov IPv6 kot pri protokolu IPv4. Celotni naslovni prostor v internetu je trenutno določen s predpono 2000::/3. Vsak ponudnik internetnih storitev dobi naslovni prostor z masko /64, ta vsaki stranki določi naslovni prostor z masko /48, ta pa nato vsakemu segmentu v omrežju določi blok naslovov IPv6 z masko /64. Tako lahko določena stranka ponudniku internetnih storitev oglašuje samo en naslov z masko /48, ta pa v Internet oglašuje samo en naslov z masko /64. Ta način se imenuje agregacija in močno zmanjša velikost usmerjevalne tabele na usmerjevalnikih.

2.1.6 Spremembe v konfiguraciji naslova IPv6

Zaradi večje dolžine naslova IPv6 so uvedli nov način nastavljanja le tega, ki avtomatizira nastavljanje, tudi če v omrežju nimamo nastavljenega protokola DHCP. Temeljna mehanizma za nastavljanje naslovov IPv6 sta SLAAC (angl. Stateless Address Auto Configuration) in DHCPv6 (angl. Dynamic Host Configuration Protocol Version 6). SLAAC je protokol, ki se uporablja za avtokonfiguracijo naslova IPv6, obenem pa poskrbi še za globalno unikatnost naslova. Usmerjevalnik, ki omogoča SLAAC, periodično v lokalnem omrežju oglašuje podatke o tem omrežju in jih pošilja vsem napravam v omrežju. Oglašujeta se predpona omrežja (na lokalni ravni) in privzeta pot (angl. Default gateway) omrežja. Poleg autokonfiguracije SLAAC protokol

omogoča tudi detekcijo podvojenih naslovov IPv6 v omrežju.

Koraki SLAAC:

- *Generiranje lokalnega Link-local naslova*: Link-local je naslov, ki se uporablja za usmerjanje znotraj omrežja in ga ne moremo uporabljati kot izvorni/ponorni naslov za usmerjanje med omrežji. Za ta naslov je rezerviranih prvih 10 bitov naslova in so vedno enaki (FE80). Tem desetim bitom sledi 54 ničel, zadnjih 64 bitov pa si določi vsaka naprava sama bodisi na podlagi MAC naslova bodisi na podlagi kakšnega algoritma.

- *Testiranje zgeneriranega Link-local naslova*: Ko si naprava zgenerira Link-local naslov, s pomočjo protokola za odkrivanja sosedov (angl. Neighbor Discovery - ND) vsem napravam v omrežju pošlje t.i. »Neighbor Solicitation« sporočilo s svojim Link-local naslovom. Če katera od ostoječih naprav že vsebuje tak naslov, to sporoči z »Network Advertisement« sporočilom, in sprašujoča naprava si mora zgenerirati nov Link-local naslov in postopek ponoviti. Ta korak lahko smatramo kot varnostno luknjo v omrežju, saj lahko neko napravo prekonfiguriramo tako, da ob vsakem takem sporočilu vrne odgovor, kar pomeni, da si bo sprašujoča naprava vedno znova izračunavala svoj Link-local naslov.

- *Dodelitev Link-local naslova*: Če test ugotavljanja podvojenega Link-local naslova uspe, si naprava ta naslov nastavi.

- *Kontaktiranje usmerjevalnika*: Ko imamo nastavljen Link-local naslov, si lahko nastavimo še globalni naslov, ki se uporablja za naslavljanje med omrežji. Ta naslov si naprava lahko skonfigurira na dva načina. Lahko počaka, da usmerjevalnik pošlje »Router Advertisement« sporočilo (spet preko protokola za odkrivanje sosedov), ali pa usmerjevalniku pošlje »Router Solicitation« sporočilo, ki prisili usmerjevalnik, da predčasno v omrežje pošlje »Router Advertisement« sporočilo.

- *Odgovor usmerjevalnika*: Odgovor, ki ga sprašujoča naprava dobi od usmerjevalnika med drugim vsebuje tudi način nastavljanja globalnega naslova IPv6.

- *Konfiguracija globalnega naslova*: V primeru SLAAC, si naprava nastavi globalni naslov IPv6 s pomočjo predpone omrežja, ki jo dobi od usmerjevalnika, za spodnjih 64 bitov pa se uporabi npr. naslov MAC naprave oziroma kakšen drugačen algoritem. Paziti moramo le, da je zgenerirani naslov globalno unikaten.

Drugi protokol, ki se uporablja za nastavljanje naslovov IPv6 je DHCPv6, ki je definiran v RFC 3315 ("Dynamic Host Configuration For IPv6") Nekatere lastnosti protokola:

- Omogoča več kontrole kot SLAAC (usmerjevalnik ve, katera naprava ima določen naslov IP).

- Lahko se uporablja tudi v okolju, kjer nimamo nobenega usmerjevalnika.

- Lahko se uporablja vzporedno s SLAAC.

- Uporablja se lahko tudi za preštevilčevanje omrežja (omrežje zamenja predpono).

Proces nastavljanja naslova IPv6 preko protokola DHCPv6 je podoben tistemu v IPv4. Naprava, ki se želi povezati v omrežje lahko zazna prisotnost usmerjevalnika v omrežju. Če najde vsaj en usmerjevalnik, preuči njegov RA (angl. Router Advertisement), da ugotovi ali se uporablja DHCPv6.

Koraki DHCPv6:

- V primeru, da je DHCPv6 omogočen ali v primeru, da odjemalec ne najde nobenega usmerjevalnika v omrežju, odjemalec prične fazo DHCP solicit, s katero poišče strežnik DHCPv6 v omrežju. Sporočilo pošlje na FF02::1:2 - multicast naslov za vse strežnike DHVPv6 v omrežju.

- Strežnik DHCPv6 odjemalcu sporoči svoj naslov.
- Odjemalec na vrnjeni naslov pošlje zahtevo za dodelitev naslova IPv6.
- Strežnik DHCPv6 zgradi naslov in ga pošlje odjemalcu.
- Odjemalec si ta naslov nastavi za svojega.

2.2 Naslavljanje v protokolu IPv6

Naslovi IPv6 so predstavljeni z osmimi 16-bitnimi polji, kar skupaj znaša 128 bitov. Primer takega naslova je 2001:0db0:1234:0078:0000:0000:0000:a345. Osnovni zapis lahko skrajšamo tako, da izpustimo vodilne ničle in dobimo 2001:db0:1234:78:0:0:0:a345. Ta zapis pa lahko še bolj skrajšamo tako, da zaporedje več ničelnih polj zamenjamo z znakom "::", vendar to lahko naredimo le enkrat. Najkrajši pravilen zapis zgornjega naslova IPv6 je torej 2001:db0:1234:78::a345.

Poznamo tudi dva posebna načina zapisa naslovov IPv4 znotraj naslova IPv6:

- *Kompatibilni naslovi IPv4*: Ta vrsta zapisa se uporablja pri vzpostavljanju avtomatskih tunelov preko naslovnega prostora IPv4. Primer zapisa: 0:0:0:0:0:0:192.0.2.100 = ::C000:0264
- *Preslikani naslovi IPv4*: Ta vrsta zapisa omogoča predstavitev naslova IPv4 znotraj naslova IPv6. Primer zapisa: 0:0:0:0:0:FFFF:192.0.2.100 = ::FFFF:C000:0264

2.2.1 Tipi naslovov pri protokolu IPv6

Protokol IPv6 pozna tri tipe naslovov: unicast, anycast in multicast. Za razliko od protokola IPv4 pri protokolu IPv6 ne obstajajo broadcast naslovi, vendar podobno funkcionalnost lahko dosežemo z multicast naslovi.

Unicast naslovi

Unicast naslovi se uporabljajo za naslavljanje točno določene naprave/vmesnika v omrežju. Poznamo več vrst unicast naslovov:

- *Globalni unicast naslovi*: Omogočajo možnost naslavljanja celotnega globalnega omrežja IPv6. Sestavljen je iz treh delov:
  - *Usmerjevalna predpona*: običajno 48 bitov, ki določa posamezno organizacijo. Prvi trije biti so enaki 001, ostali biti pa enolično določajo registrarja, ponudnika internetnih storitev in organizacijo.
  - *Podomrežje*: Tipično rezerviranih 16 bitov.
  - *Naslov vmesnika*: Zadnjih 64 bitov naslova, ki določajo posamezen vmesnik. Naslov vmesnika ni nujno dolg 64 bitov, vendar je priporočena takšna dolžina saj SLAAC predvideva 64-bitni naslov vmesnika, pravtako pa nekateri operacijski sistemi, npr. Microsoft Windows XP, ne dopuščajo sprememb dolžine maske omrežja.
- *Link local naslovi*: Vsi vmesniki, na katerih je omogočen IPv6, imajo določen ta naslov. Namenjen je naslavljanju znotraj nekega omrežja oziroma znotraj nekega segmenta. Prepoznamo ga po predponi FE80::/10 in 64-bitnem identifikatorju vmesnika. Uporablja se pri avtomatskem nastavljanju globalnih naslovov IPv6, iskanju sosedov v omrežju (ND – Network Discovery), uporabljajo pa ga tudi nekateri usmerjevalni protokoli. Zadostujejo za povezovanje naprav v istem omrežju, torej ne potrebujejo globalnih naslovov.
- *Unique local naslovi*: So podobni site-local naslovom. Namenjeni so usmerjanju znotraj organizacije, s tem da so, glede na druge organizacije, enolično določeni. Te vrste naslovi se v internetnem omrežju ne usmerjajo. ULA naslovi so dveh tipov:
  - *FC00::/8*: še nedefinirani, vendar planirano, da bojo globalno usmerjani,

  - *FD00::/8*: uporabljajo se za naslavljanje v neki zasebnem omrežju IPv6, ki se ne priklaplja v internet (npr. mreža bankomatov, za katero nočemo, da je dosegljiva z interneta).

- *Posebni unicast naslovi*:

  - *Nedoločeni naslovi*: uporabljajo se v primeru, ko nimamo na voljo nobenega drugega naslova, npr. pri pošiljanju naslova DHCPv6 za globalni naslov oziroma v primeru, ko je poslan paket za dupliciran naslov v omrežju (angl. DAD Duplicate Address Detection).
  - *Loopback naslovi*: ekvivalenten naslovu IPv4 127.0.0.1.
  - *IPv4 preslikani naslovi*: Uporabljajo jih nekateri tranzicijski mehanizmi, tudi 6PE in 6VPE, ki sta opisana v nadaljevanju, ko se naslov IPv4 uporablja kot next hop v predponi IPv6.

Anycast naslovi

Anycast naslov IPv6 je naslov, ki ga dodelimo vmesniku na eni ali več napravah. Ko nek paket pošljemo na anycast naslov, se ta dostavi na najbližji vmesnik s tem naslovom. Najbližji vmesnik je določen glede na mero distance nekega usmerjevalnega protokola. To lahko pomeni število hopov, hitrost povezave in druge. Vse naprave, ki si delijo ta naslov, naj bi omogočale enake storitve, zato je vseeno, katero napravo/vmesnik izbere za najbližjo. Ideja o anycast naslovu sega v leto 1993 in je definirana kot način, da pošljemo paket na najbližji naslov, ki je član neke anycast skupine. Ta tehnika nam omogoča najti najbližjo napravo/vmesnik, ki pripada neki grupi. Anycast naslovi spadajo v isti naslovni prostor kot unicast naslovi, zato jih od slednjih ne ločimo. Ko tak naslov določimo vmesniku, moramo eksplicitno povedati, da gre za anycast naslov. V nasprotnem primeru bi se smatralo, da gre za podvojem naslov v omrežju. Trenutno se anycast naslovi uporabljajo pri domenskih strežnikih.

Multicast naslovi

Ta vrsta naslovov služi usmerjevalnikom, da pošiljajo sporočila množici vmesnikov. Posamezen vmesnik lahko pripada več multicast skupinam. Definirani so na naslednji način:

- Prvih 8 bitov je enakih FF, kar označuje multicast naslove.
- Naslednji 4 biti predstavljajo zastavice.
- Naslednji 4 biti poredstavljajo doseg multicast naslova.
- Sledi 112 bitov, ki določajo multicast skupino.

Poglavje 3

Pregled IPv4/IPv6 prehodnih mehanizmov

Navkljub vsem prizadevanjem za zmanjševanje izrabe naslovnega prostora IPv4 je njegova popolna izčrpanost v bližnji prihodnosti neizbežna. Edina dolgoročna rešitev je prehod na protokol IPv6, ki ponuja dovolj velik naslovni prostor za vse projekcije prihodnje rabe. IPv6 je protokol, ki nastaja in se neprestano izboljšuje že več kot 10 let, uporaba pa se v zadnjih letih močno povečuje.

Idealni model prehoda na protokol IPv6 bi bilo podaljšano obdobje, v katerem bi naprave v internetu podpirale tako protokolni sklad IPv4 kot IPv6 do točke, ko bi protokol IPv6 podpiralo dovolj naprav, da protokol IPv4 ne bi bil več potreben. Temeljni problem je namreč, da sta protokola IPv4 in IPv6 v osnovi nezdružljiva, kar pomeni, da naprave, ki podpirajo samo protokolni sklad IPv4, in naprave, ki podpirajo samo protkolni sklad IPv6, med seboj ne morejo komunicirati brez pomoči t.i. prehodnih mehanizmov IPv4/v6.

Prehodne mehanizme v grobem delimo v tri skupine:

- Dvojni sklad
- Tunelski mehanizmi

- Translacijski mehanizmi

V splošnem delimo prehodne mehanizme IPv4/v6 na način, ki je opisan zgoraj. Vendar moramo upoštevati dejstvo, da se tako naslovi IPv4 kot naslovi IPv6 pojavljajo ne le na omrežnem nivoju ampak tudi na višjih nivojih (npr. neka aplikacija izvede DNS poizvedbo). Tudi če bi zgoraj omenjeno »podaljšano obdobje« podaljšali za toliko časa, da bi vse naprave podpirale protokolni sklad IPv6 in bi posledično izklopil protokolni skladi IPv4, to še vedno ne bi zadostovalo.

Problem nastane pri aplikacijah, saj nekatere aplikacije še vedno ne podpirajo protokolnega sklada IPv6. Primer takih aplikacij so Skype, Trillian, rTorrent, Outlook Express in Novell Directory [3]. To pomeni, da tudi če naše celotno omrežje podpira protokolni sklad IPv6 (na omrežnem nivoju - usmerjevalniki, požarne pregrade, ...), moramo še vedno imeti mehanizme, ki bodo omogočili aplikacijam, ki podpirajo le protokolni sklad IPv4, komunicirati z napravami, ki se nahajajo v omrežju IPv6.

V tem dokumntu so najprej predstavljene vse tri zgoraj naštete skupine prehodnih mehanizmov, njihove prednosti in slabosti, na koncu pa se osredotočim na mehanizem, ki omogoča aplikaciji, ki podpira le protokolni sklad IPv4, komunikacijo s strežniki, ki se nahajajo v omrežju IPv6.

3.1 Dvojni sklad

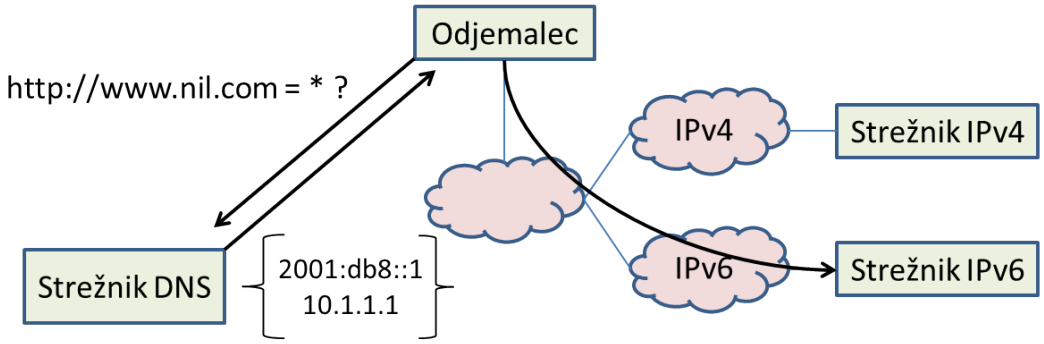
Ključ do uspešnega prehoda protokola IPv4 na novejši protokol IPv6 je ohranjanje kompatibilnosti s starim omrežjem IPv4. Najbolj enostaven način za dosego tega cilja je, da naprave, ki jih nadgradimo na omrežje IPv6 še vedno podpirajo omrežje IPv4. Takim napravim, ki podpirajo obe omrežji, pravimo naprave z dvojnim skladom. Naprave, ki podpirajo dvojni sklad, torej lahko pošiljajo tako pakete IPv4 kot tudi pakete IPv6.

Kljub temu, da lahko naprava na istem vmesniku podpira tako protokol IPv4 kot protokol IPv6, pa ni nujno, da sta na tem vmesniku oba omogočena. Tako lahko vozlišča, ki podpirajo oba protokolna sklada, razdelimo na tri skupine:

- tiste, ki imajo omogočen protokolni sklad IPv4 in onemogočen protokolni sklad IPv6
- tiste, ki imajo omogočen protokolni sklad IPv6 in onemogočen protokolni sklad IPv4
- tiste, ki imajo omogočena oba protokolna sklada

Konfiguriranje naslovov IP na vmesnikih poteka pri omrežju IPv4 drugače kot pri omrežju IPv6. Naslovi IPv4 se konfigurirajo bodisi statično bodisi s pomočjo protokola DHCP, naslovi IPv6 pa tudi bodisi statično bodisi s pomočjo protokola DHCPv6 bodisi s pomočjo metode SLAAC.

Primer delovanja aplikacije, ki uporablja dvojni sklad:



Slika 3.1: Dvojni sklad: primer delovanja

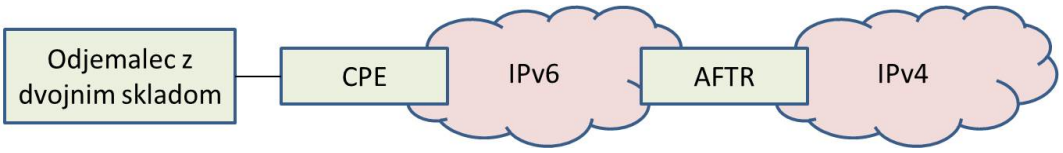
Aplikacija, ki podpira oba protokolna sklada, pošlje zahtevo DNS tako za naslov IPv4 kot za naslov IPv6 za določeno domeno (npr. www.nil.com). Strežnik DNS vrne oba naslova IP, aplikacija pa se nato odloči preko katerega protokolnega sklada bo pošiljala zahteve. Privzeto se uporabi naslov IPv6, lahko pa se do strežnika dostopa tudi preko naslova IPv4.

3.1.1 Dual Stack Lite

Pred več kot desetimi leti je bilo predvideno, da bo prehod iz starega omrežja IPv4 na novo omrežje IPv6 potekalo predvsem preko naprav, ki podpirajo dvojni sklad. Predvidevano je bilo, da naj bi večina naprav prešla na IPv6 povezljivost še preden bo zmanjkalo naslovov IPv4. Žal se to ni zgodilo, posledica tega pa je, da morajo biti vsebine, ki so dosegljive preko IPv4 povezljivosti, dosegljive tudi tistim uporabnikom, ki imajo samo IPv6 povezljivost.

Terminologija, uporabljena pri DS-Lite protokolu:

- *CPE (Customer Premise Equipment)*: Usmerjevalnik, ki ga ponudnik internetnih storitev ponudi stranki, in preko katerega se stranka poveže v omrežje ponudnika storitev.
- *Address Family Transition Router (AFTR) element*: element, ki omogoča odljemalcem, ki imajo omogočen le protokolni sklad IPv6, da dostopajo do vsebin, ki so dostopne le preko protokola IPv4.



Slika 3.2: Dual Stack Lite: shema omrežja

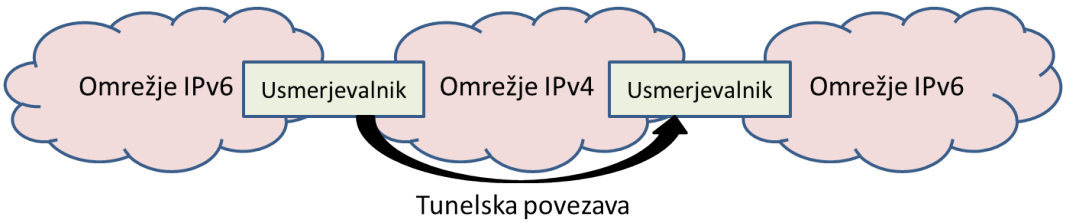
Predvidimo, da lahko odjemalec v strankinem omrežju do internetnih vsebin dostopajo tako preko IPv4 kot preko IPv6 povezljivosti. Ob priklopu stranke v omrežje ponudnika internetnih storitev ta od ponudnika dobi globalni naslov IPv6 ter zasebni naslov IPv4. Omenimo pa še, da omrežje ponudnika internetnih storitev podpira le IPv6 povezljivost.

Ob taki topologiji je v primeru, ko odjemalec želi dostopati do vsebin, ki

so dostopne le preko IPv4 povezljivosti, potrebno paket IPv4 na usmerjevalniku CPE enkapsulirati v paket IPv6, nato ga prenesti do usmerjevalnika AFTR, dekapsulirati nazaj v paket IPv4, opraviti NAT translacijo (iz odjemalčevega zasebnega naslova IPv4 v globalni naslov IPv4 željene vsebine) ter ga poslati v omrežje IPv4.

3.2 Tunelski mehanizmi

Tunelski mehanizmi sami po sebi ne omogočajo neposredne komunikacije med napravami IPv4 in napravami IPv6, temveč omogočajo povezavo več bodisi otokov IPv6 bodisi otokov IPv4, ne da bi bilo potrebno nadgraditi celotno omrežje. Na spodnji sliki je prikazan osnovni princip delovanja tunelskih mehanizmov. Ob prehajanju iz omrežja IPv4 na omrežje IPv6 nastaja vedno več otokov IPv6, ki pa so obdani z omrežji IPv4. Da bi komunikacija med omrežji IPv6 potekala nemoteno, med njimi ustvarimo tunele, ki paket IPv6 enkapsulirajo v paket IPv4 in ga pošljejo na drugo stran.



Slika 3.3: Osnovni princip delovanja tunelov

3.2.1 SIIT

SIIT (Stateless IP/ICMP Translation) [5] je protokol, ki izvaja pretvarjanje glave paketov IP ter paketov ICMP iz protokola IPv4 v protokol IPv6 in obratno. Ta protokol je osnova mnogim drugin prehodnim mehanizmom, zato ga bom opisal nekoliko podrobneje.

Prevajalnik IP/ICMP deluje v dveh različnih načinih, s stanji in brez stanj. V obeh primerih predvidevamo, da sistem z naslovom IPv4 in brez naslova IPv6 komunicira s sistemom z naslovom IPv6 in brez naslova IPv4 oziroma da sistema nimata stične usmerjevalne povezljivosti in mora biti njuna komunikacija prevedena.

V načinu brez stanj sistem IPv4 predstavlja poseben nabor naslovov IPv6 (pretvorjeni naslovi IPv4), sistem IPv6 pa ima naslove (prevedljive naslove IPv4), ki so lahko algoritmično preslikani v podskupino naslovov IPv4 ponudnika storitev. Prevedljivi naslovi IPv4 so podskupina pretvorjenih naslovov IPv4. V tem primeru prevajalna tabela ni potrebna.

V načinu s stanji poseben nabor naslovov IPv6 predstavlja sistem IPv4 (pretvorjeni naslovi IPv4), sistem IPv6 pa lahko uporabi katerekoli naslove IPv6, razen naslovov iz tega območja. V tem primeru je potrebno imeti prevajalno tabelo za povezavo med sistemskimi naslovi IPv6 in naslovi IPv4, vzdrževanimi v prevajalniku.

Ko prevajalnik IP/ICMP prejme paket IPv4, naslovljen na destinacijo preko domene IPv6, prevede glavo paketa IPv4 v glavo paketa IPv6. Originalna glava paketa IPv4 je odstranjena iz paketa, nadomesti pa jo glava paketa IPv6, prav tako je po potrebi osvežena kontrolna vsota. Podatki ostanejo nespremenjeni. Mehanizem IP/ICMP nato paket v omrežje posreduje na podlagi ciljnega naslova IPv6.

3.3 Translacijski mehanizmi

Tunelski mehanizmi z ustvarjanjem navideznih povezav omogočajo medsebojno povezovanje bodisi otokov IPv4 bodisi okotov IPv6. Translacijski mehanizmi pa omogočajo neposredno komunikacijo med napravami, ki podpi-

rajo zgolj enega izmed protokolnih skladov. Ti mehanizmi bodo pomembni predvsem v času popolne izčrpanosti naslovnega prostora IPv4, ko bodo organizacije in končni uporabniki lahko dobili zgolj še IPv6 povezljivost, velik del preostalega interneta pa bo še vedno uporabljal protokol IPv4.

3.3.1 NAT64

To poglavje opisuje prehodni mehanizem za prevajanje naslovov NAT64, ki omogoča odjemalcem IPv6, da se povežejo s strežniki IPv4 s pomočjo protokolov TCP, UDP ali ICMP. Skupaj z mehanizmom DNS64 omogoča odjemalcem IPv6, da vzpostavijo komunikacijo s strežniki, ki imajo samo IPv4 povezljivost. Pod določenimi pogoji pa je mogoča tudi obratna komunikacija, kar pomeni, da lahko strežniki IPv4 inicializirajo komunikacijo do odjemalca IPv6. Protokol NAT64 je definiran v RFC 6146 (»Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers«), protokol DNS64 pa v RFC 6147 (»DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers«).

NAT64 brez stanj je mehanizem, ki translira pakete IPv6 v pakete IPv4 ter obratno. Translacija obsega preslikavo glave IPv6 v glavo IPv4 in obratno na podlagi algoritma, definiranega v RFC 6145 (»IP/ICMP Translation Algorithm«).

Kombinacija teh dveh mehanizmov v prihodnosti velja za zelo pomembno, saj bo zaradi pomanjkanja naslovov IPv4 vedno več odjemalcev IPv6, ki pa se bodo želeli povezati z obstoječimi strežniki, dosegljvimi preko starega protokola IPv4.

Terminologija, uporabljena pri mehanizmu NAT64:

- *3-tuple (trojček)*: je podatkovni objekt, ki vsebuje izvorni naslov IP, ponorni naslov IP ter identifikator ICMP in enolično določa ICMP zah-

tevo po seji. Ko se zahteve ICMP prenašajo preko mehanizma NAT64, vsaka seja vsebuje dva trojčka, enega za IPv4 in drugega za IPv6.

- *5-tuple (petorček)*: je podatkovni objekt, ki vsebuje izvorni naslov IP, izvorno številko vrat, ponorni naslov IP, ponorno številko vrat ter vrsto transportnega protokola (TCP, UDP) in enolično določa sejo TCP/UDP. Ko se seje TCP/UDP prenašajo preko mehanizma NAT64, vsaka seja vsebuje dva petorčka, enega za IPv4 in drugega za IPv6.

- *BIB (Binding Information Base)*: tabela preslikav, opravljenih pri mehanizmu NAT64. Za vsak protokol obstaja svoja tabela preslikav (TCP, UDP,ICMP).

- *Seja*: tok paketov, prenesenih med dvema odjemalcema.

- *Sejna tabela*: Tabela, ki hrani vse seje, ki se uporabljajo pri prokololu NAT64.

- *Transportni naslov*: Kombinacija naslova/vrat IPv6 ali naslova/vrat IPv4.

- *Terka*: Navezuje se bodisi na trojčka bodisi na perorčka.

- *WKP (Well Known Prefix)*: označuje predpono IPv6 64:ff9b::/96, ki se uporablja za algoritmično preslikovanje naslovov različnih družin (v našem primeru med naslovi IPv6 in naslovi IPv4). Predpona ni usmerljiva v globalnem IPv6 Internetu.

Značilnosti/prednosti mehanizma NAT64 brez stanj:

- Deljenje/souporaba naslovov IPv4: Mehanizem NAT64 omogoča več odjemalcem IPv6, da si delijo isti naslov IPv4 za dostop do IPv4 Interneta.

- Komunikacijo med nekim odjemalcem IPv6 in strežnikom IPv4 lahko prične tudi strežnik IPv4, če:

- od prej že obstaja BIB za omenjeni napravi (pomeni, da sta v preteklosti te dve napravi že komunicirali)
- je preslikava med njunimi trojčki statično navedena

Osnovne zahteve za delovanje mehanizma NAT64 so množica odjemalcev IPv6, množica strežnikov IPv4, naprave, na kateri je implementiran mehanizem NAT64 (in ima dostop tako do omrežja IPv6 kot do omrežja IPv4) ter strežnik DNS64, do katerega lahko dostopajo odjemalci IPv6.

Mehanizem NAT64 je implementiran na napravi, ki ima na voljo vsaj dva vmesnika (vmesnik IPv4, povezan v omrežje IPv4 ter vmesnik IPv6, povezan v omrežje IPv6). Paketi, ki jih generira odjemalec IPv6 in so namenjeni strežniku IPv4, se bodo po omrežju IPv6 najprej usmerjali do naprave, ki podpira mehanizem NAT64, ta naprava pa jih bo preslikala v pakete IPv4 in jih poslala naprej do ciljne naprave v omrežju IPv4. Med preslikavo paketa IPv6 v paket IPv4, si odvisno od protokola, uporabljenega pri prenosu (TCP, UDP, ICMP), naprava, ki uporablja mehanizem NAT64, v svojo BIB tabelo shrani preslikave naslovov/vrat IP, s čimer omogoči paketom, poslanih s strani strežnika IPv4, da uspešno dosežejo odjemalca IPv6.

Preslikave naslovov/vrat lahko opravimo statično ali pa se generirajo dinamično ob prvem paketu, ki doseže napravo, ki uporablja mehanizem NAT64. Dodatni mehanizmi, kot sta ICE (tip NAT traversal metode) in statično preslikovanje, pa omogočata to, da lahko tudi strežniki IPv4 inicializirajo komunikacijo z odjemalci IPv6.

Za preslikovanje naslovov IP mehanizem NAT64 uporablja dve množici naslovov, ki se jih lahko uporablja pri preslikovanju. V eni množici so naslovi IPv6, ki predstavljajo naslove IPv4 v omrežju IPv6, v drugi množici pa so naslovi IPv4, ki predstavljajo naslove IPv6 v omrežju IPv4.

Naslovi IPv6 so določeni z eno ali več predponami. Označimo množico pred-

pon IPv6 s Pref64::/n. Te predpone bo mehanizem NAT64 uporabil za generiranje naslovov IPv6 s pomočjo podanega naslova IPv4. Naslovi IPv4 pa so tudi določeni z neko predpono, ki jo običajno določi lokalni administrator. Ker je običajno na voljo mnogo manj naslovov IPv4 kot pa imamo na voljo naslovov IPv6, preslikave niso stalne, ampak se po določenemu času nekativnosti pobrišejo.

Predpostavimo, da imamo na voljo dve vozlišči:

- Odjemalec H1, ki se nahaja v omrežju IPv6 in ima naslov 2001:db8::1
- Odjemalec H2, ki se nahaja v omrežju IPv4 in ima naslov 192.0.2.1 ter domeno www.example.com

Da bi lahko odjemalec H1 uspešno komuniciral z odjemalcem H2, moramo njuni omrežji povezati z napravo, ki podpira mehanizem NAT64 ter postaviti strežnik DNS64, do katerega lahko dostopajo odjemalci IPv6. Predpostavimo še, da ima mehanizem NAT64 na voljo samo en naslov IPv4 (203.0.113.1) ter množico naslovov IPv6 (2001:db8::/48). Ko imamo postavljeno potrebno infrastrukturo, so za uspešno komunikacijo med odjemalcema H1 in H2 potrebni naslednji koraki:

- Odjemalec H1 izvede zahtevo DNS za www.example.com in od strežnika DNS64 prejme AAAA RR (resource record). AAAA odgovor vsebuje naslov IPv6, sestavljen iz WKP ter naslova IPv4 odjemalca H2 (npr. 64:ff9b:192.0.2.1).
- Odjemalec H1 pošlje paket TCP SYN odjemalcu H2. Izvorni naslov v obliki (IP naslov, vrata) je (2001:db8::1, 1500), ponorni pa (64:ff9b::192.0.2.1, 80). Številke vrat določi odjemalec H1.
- Paket je po omrežju IPv6 usmerjan do vmesnika IPv6 na napravi, ki podpira mehanizem NAT64.
- Naprava, ki podpira mehanizem NAT64, prejme paket in naredi naslednje:

  - Izbere vrata, ki pri naslovu IPv4 203.0.113.1 še ni bil izbran (denimo, da so to vrata 2000) ter naredi preslikavo iz (2001:db8::1, 1500) v (203.0.113.1, 2000)
  - S pomočjo algoritma, definiranega v RFC 6145 ("IP/ICMP Translation Algorithm"), prevede glavo paketa IPv6 v glavo paketa IPv4.
  - (203.0.113.1, 2000) postane izvorni naslov, (192.0.2.1, 80) pa ponorni naslov paketa IPv4.

- Naprava, ki uporablja mehanizem NAT64 pošlje paket preko vmesnika IPv4 do ciljne naprave.

- Odjemalec H2 odgovori s paketom TCP SYN+ACK z izvornim naslovom ter (192.0.2.1, 80) ponornim naslovom (203.0.113.1, 2000).

- Paket se preko omrežja IPv4 usmerja do vmesnika IPv4 na napravi, ki uporablja mehanizem NAT64. Ko paket prispe na vmesnik, naprava NAT64 preveri, če obstaja kakšna preslikava za (203.0.113.1, 2000). Ker obstaja preslikava ((2001:db8::1, 1500), (203.0.113.1, 2000)), naprava, ki uporablja mehanizem NAT64 naredi naslednje:
  - S pomočjo algoritma, definiranega v RFC 6145 ("IP/ICMP Translation Algorithm"), prevede glavo paketa IPv4 v glavo paketa IPv6.
  - Izvorni naslov paketa IPv6 postane (64:ff9b::192.0.2.1, 80), ponorni naslova pa (2001:db8::1, 1500)

- Translirani paket se pošlje preko vmesnika IPv6 do odjemalca H1

3.3.2 464XLAT

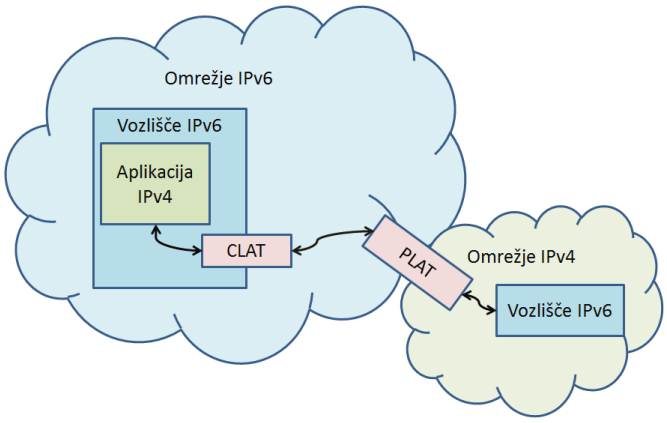
464XLAT [9] je translacijski mehanizem, ki omogoča povezljivost IPv4 preko omrežja IPv6 s pomočjo že znanih mehanizmov:

- NAT64 brez stanj [8]
- SIIT [7]

Terminologija, uporabljena pri mehanizmu 464XLAT:

- PLAT (Provider side translator): označuje translator na strani ponudnika storitev, ki deluje v skladu z RFC 6146 [8] in ki opravlja N:1 preslikave naslovov IPv6 v globalni naslov IPv4 in obratno
- CLAT (Customer side translator): označuje translator na strani končnega uporabnika, ki deluje v skladu z RFC 6145 [7] in ki opravlja 1:1 preslikave zasebnega naslova IPv4 v globalni naslov IPv6 in obratno. Nahaja se lahko tako na usmerjevalniku kot na končni napravi (npr. mobilnem telefonu)

Mehanizem je uporaben v primeru, ko imamo neko napravo, ki ima direkten dostop samo do omrežja IPv6 in na kateri je nameščena aplikacija, ki podpira le protkolni sklad IPv4. Primer take aplikacije je Skype. Če je Skype nameščen na napravi, ki ima dostop le do omrežja IPv6, in kjer ne uporabljamo mehanizma 464XLAT, aplikacija ne deluje. Primer sheme omrežja je prikazan na spodnji sliki.

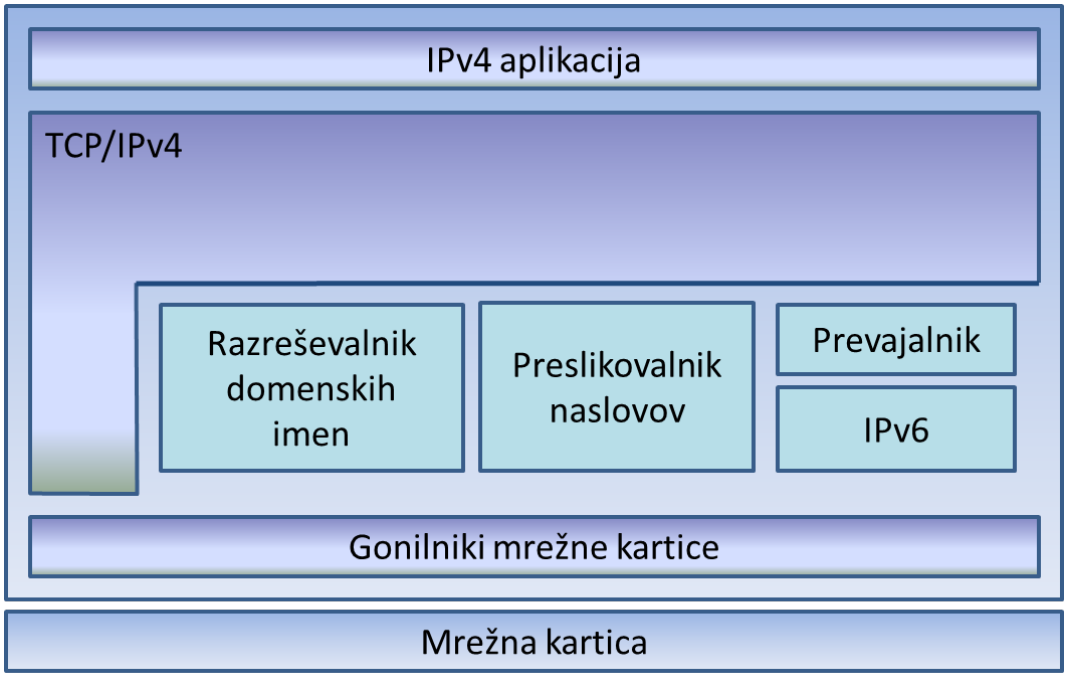


Slika 3.4: Shema omrežja pri uporabi mehanizma 464XLAT

3.3.3 Bump-in-the-Stack (BIS)

Na samem začetku prehoda iz omrežij IPv4 na omrežja IPv6, kjer imamo v mislih predvsem usmerjevalnike in ostalo omrežno opremo, je obenem skoraj nemogoče nadgraditi tudi vse aplikacije, tako da bi podpirale nov protokolni sklad IPv6. V ta namen je bil razvit mehanizem Bump-in-the-Stack [10], ki omogoča aplikacijam, ki podpirajo le protokolni sklad IPv4 in ki so nameščene na napravah, ki imajo lahko omogočen tako protokolni sklad IPv6 kot protokolni sklad IPv4, da dosopajo do vsebin IPv6.

Mehanizem deluje tako, da prestreza podatke med modulom TCP/IPv4 ter gonilniki mrežnih kartic na napravah, na katerih je nameščen. Na tem mestu prestreza pakete IPv4 ter jih pretvarja v pakete IPv6 in obratno. Strukturo naprave, na kateri je nameščen ta mehanizem, prikazuje spodnja slika.



Slika 3.5: Shema naprave, na kateri je nameščen mehanizem Bump-in-the-Stack

Mehanizem je sestavljen iz treh komponent, ki so predstavljene v nadaljevanju.

Prevajalnik

Naloga prevajalnika je pretvarjanje paketov IPv4 v pakete IPv6 in obratno v skladu z mehanizmom SIIT [7]. Ko prevajalnik prejme paket IPv4, ki ga pošlje aplikacija IPv4, ta pretvori glavo paketa IPv4 v glavo paketa IPv6 in po potrebi fragmentira paket IPv6, saj so pretvorjeni paketi večji za 20B ter ga pošlje v omrežje IPv6. Ko pa prevajalnik prejme paket IPv6 iz omrežja, pa pretvori glavo paketa IPv6 v glavo paketa IPv4 ter ga pošlje aplikaciji IPv4.

Razreševalnik domenskih imen

Razreševalnik domenskih imen prestreza zahteve DNS, ki jih generira aplikacija. Aplikacija vedno generira zahteve za vnose A (IPv4) na strežniku DNS. Razreševalnik domenskih imen take pakete prestreže ter naredi novo poizvedbo, tako za naslov IPv4 (A) kot tudi za naslov IPv6 (AAAA). V primeru, da strežnik DNS vrne naslov IPv4, razreševalnik domenskih imen vrne aplikaciji dobljeni naslov. Če pa naslov IPv4 za dano zahtevo DNS ne obstaja (obstaja le naslov IPv6), pa razreševalnik domenskih imen dobi naslov IPv4 danega naslova IPv6 preko preslikovalnika naslovov, predstavljenega v nadaljevanju. Ta naslov se nato posreduje aplikaciji IPv4.

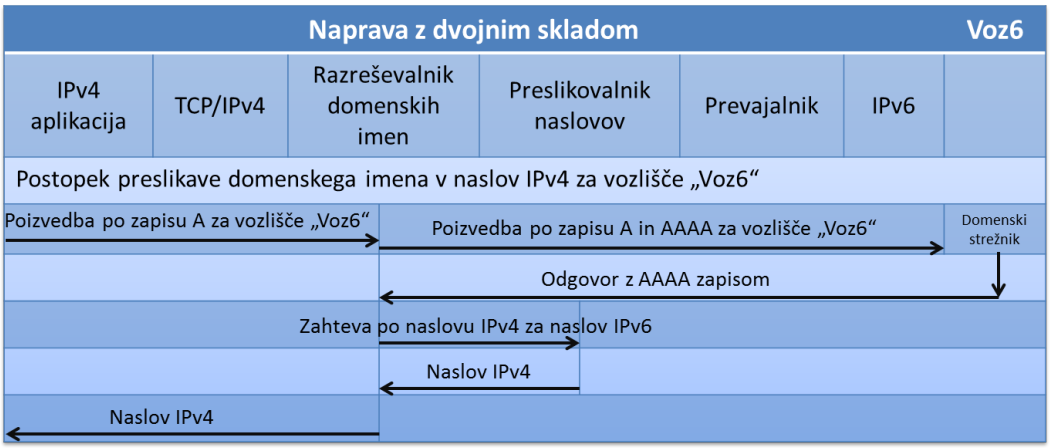
Preslikovalnik naslovov

Preslikovalnik naslovov skrbi za dodeljevanje zasebnih naslovov IPv4, ki jih zahteva razreševalnik domenskih imen. Za vsak naslov IPv4, ki ga posreduje razreševalniku domenskih imen, hrani po eno relacijo naslov IPv4-IPv6. Preslikovalnik naslovov se uporablja v dveh primerih:

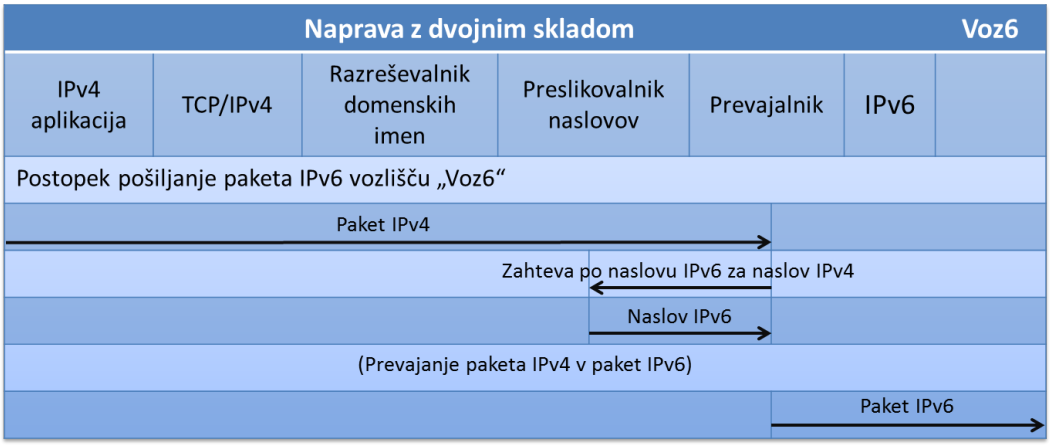
- ko razreševalnik domenskih imen od strežnika DNS dobi le vnos AAAA ali

- ko prevajalnik prejme paket IPv6 in zanj še ne obstaja preslikava naslov IPv6-IPv4

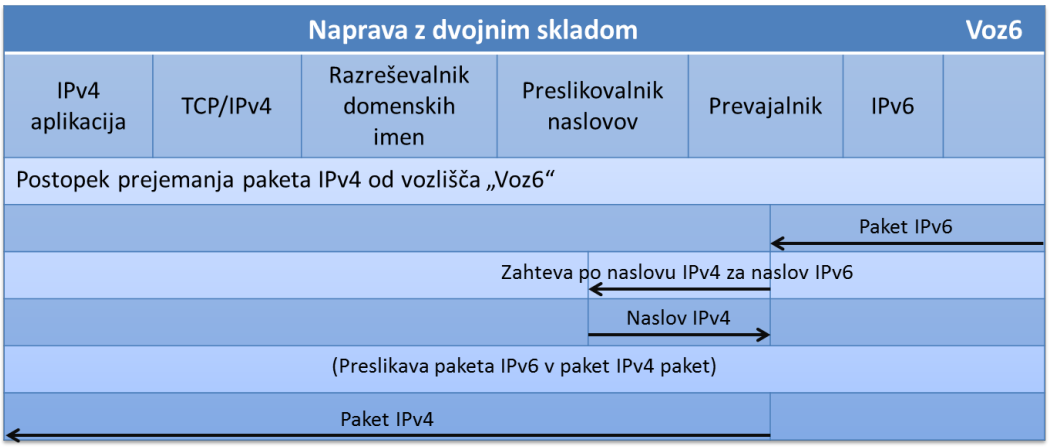
Na spodnjih slikah so prikazani primeri poizvedbe DNS, pošiljanje in prejemanje paketa IPv4 pri uporabi mehanizma Bump-in-the-Stack.



Slika 3.6: Postopek poizvedbe DNS pri mehanizmu BIS



Slika 3.7: Postopek pošiljanja paketa IPv6 pri mehanizmu BIS



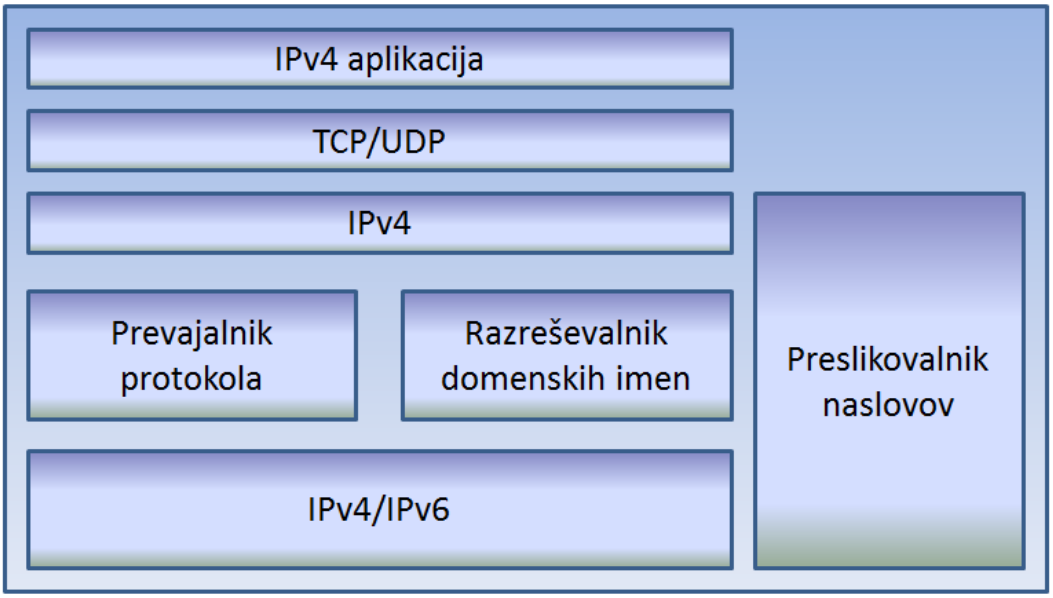
Slika 3.8: Postopek prejemanje paketa IPv6 pri mehanizmu BIS

3.3.4 Bump-in-the-API (BIA)

Mehanizem Bump-in-the-API [11] je izboljšana verzija mehanizma Bump-in-the-Stack, predstavljenega v prejšnjem poglavju. Mehanizem BIS prestreza pakete med modulom TCP/IP ter gonilniki mrežnih kartic, metdem ko mehanizem Bump-in-the-API uporablja prevajalnik API, ki se nahaja med vtičniškim API ter modulom TCP/IP. Mehanizem BIA nadomesti vtićniški API IPv4 klic z vtičniškim API IPv6 klicem in sestoji iz treh modulov.

Preslikovalnik nalog

Preslikovalnik nalog skrbi za preslikovanje vtičniških API funckij IPv4 v vtičniške API funkcije IPv6 in obratno. Ko preslikovalnik zazna klic IPv4, sprožen s strani aplikacije IPv4, le tega pretvori v klic IPv6, ki se nato uporablja za komunikacijo z napravo IPv6. Ko pa na drugi strani prestreže klic IPv6, namenjem aplikaciji IPv4, pa ga prevori v klic IPv4, ki pa ga nato uporablja za komunikacijo z aplikacijo IPv4.



Slika 3.9: Shema naprave, na kateri je nameščen mehanizem Bump-in-the-API

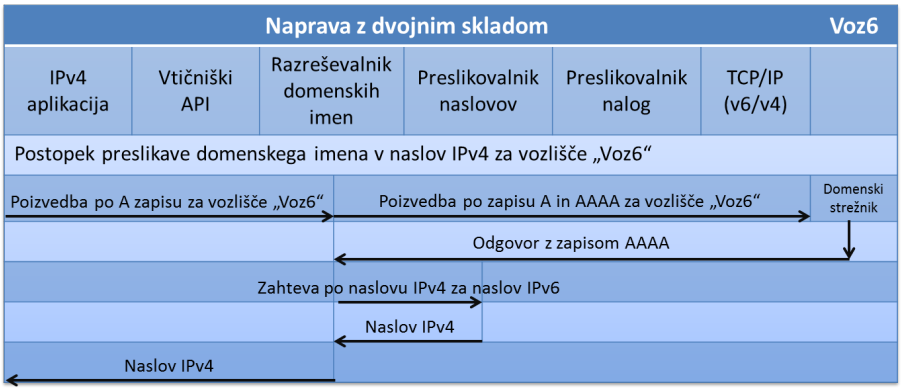
Razreševalnik domenskih imen

Razreševalnik domenskih imen skrbi za poizvedbe DNS aplikacije IPv4. Ko aplikacija poskuša pridobiti nek naslov IPv4 (npr. preko klica getbyhostname()), razreševalnik domenskih imen prestreže klic in ga nadomesti s klicem, ki zahteva tako naslov IPv4 kot naslov IPv6. V primeru, da je za dano zahtevo naslov IPv4 na voljo, aplikaciji vrne naslov IPv4, v nasprotnem primeru pa aplikaciji vrne sintetizirani naslov IPv4, ki ga je ustvaril preslikovanik naslovov, predstavljen v nadaljevanju. Sintetizirani naslov označuje virtualni naslov, ki ga zgenerira preslikovalnik naslovov za potrebe delovanje aplikacije IPv4 in se uporablja le za komunikacijo med aplikacijo IPv4 ter preslikovalnikom naslovov, ki pa ga nato prevede v naslov IPv6 strežnika/odjemalca, s katerim aplikacija IPv4 želi komunicirati.

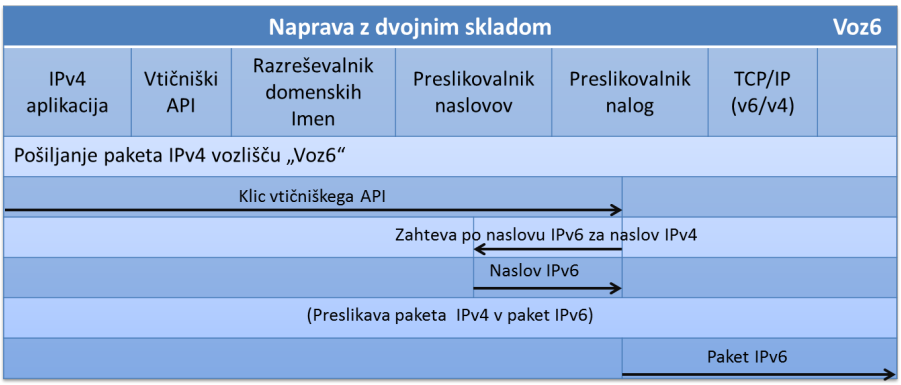
Preslikovalnik naslovov

Preslikovalnik naslovov vsebuje tabelo preslikav naslovov IPv4-IPv6. Naslove IPv4 dodeljuje iz vnaprej določenega bazena naslovov. Preslikovalnik naslovov, uporabljen pri mehanizmu BIA, je identičen tistemu, ki se uporablja pri mehanizmu BIS, predstavljenim v prejšnjem poglavju.

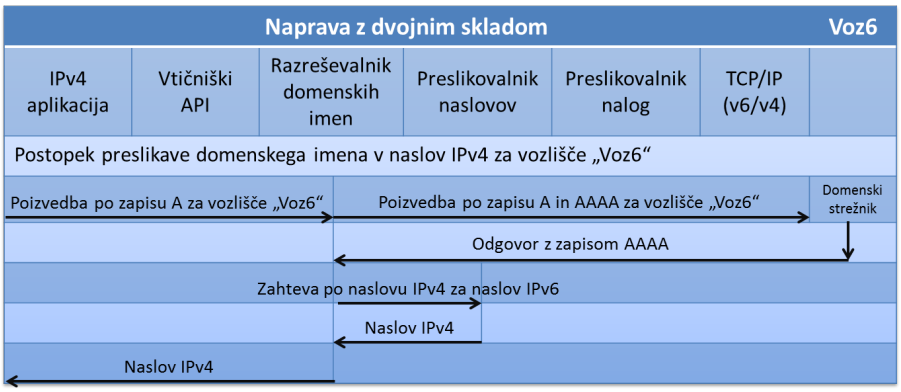
Na spodnjih slikah so prikazani primeri poizvedbe DNS, pošiljanje in prejemanje paketa IPv4 pri uporabi mehanizma Bump-in-the-API.



Slika 3.10: Postopek poizvedbe DNS pri mehanizmu BIA



Slika 3.11: Postopek pošiljanja paketa IPv6 pri mehanizmu BIA



Slika 3.12: Postopek prejemanje paketa IPv6 pri mehanizmu BIA

3.3.5 Bump-in-the-Host (BIH)

Bump-in-the-Host [12] je mehanizem, ki omogoča aplikacijam, ki podpirajo samo protokolni sklad IPv4, dostopati do naprav, ki imajo omogočen samo protokolni sklad IPv6. Naprava, na kateri je nameščena omenjena aplikacija, ima direkten dostop bodisi samo do omrežja IPv6 bodisi tako do omrežja IPv6 kot omrežja IPv4. Mehanizem BIH aplikaciji IPv4 skrije možnost direktnega dostopa do omrežja IPv6 in ustvari navidezni vmesnik IPv4, preko katerega aplikacija IPv4 komunicira z zunanjim svetom. Mehanizem je zasnovan tako, da omogoča komunikacijo z napravami IPv6 tudi preko NAT-a.

Mehanizem BIH je kombinacija mehanizmov Bump-in-the-Stack ter Bump-in-the-API, predstavljenih v prejšnjih dveh poglavjih. Mehanizem BIH je primeren za tiste aplikacije IPv4 ki za pridobivanje naslovov IPv4 uporabljajo strežnike DNS ter v aplikacijskem nivoju ISO modela ne navajajo naslovov IPv4. Omogoča dve alternativi delovanja:

- implementacija prevajalnika med protokolnim skladom IPv4 in protokolnim skladom IPv6
- implementacija prevajalnika vtičniških API klicev med modulom TCP ter modulom IPv4

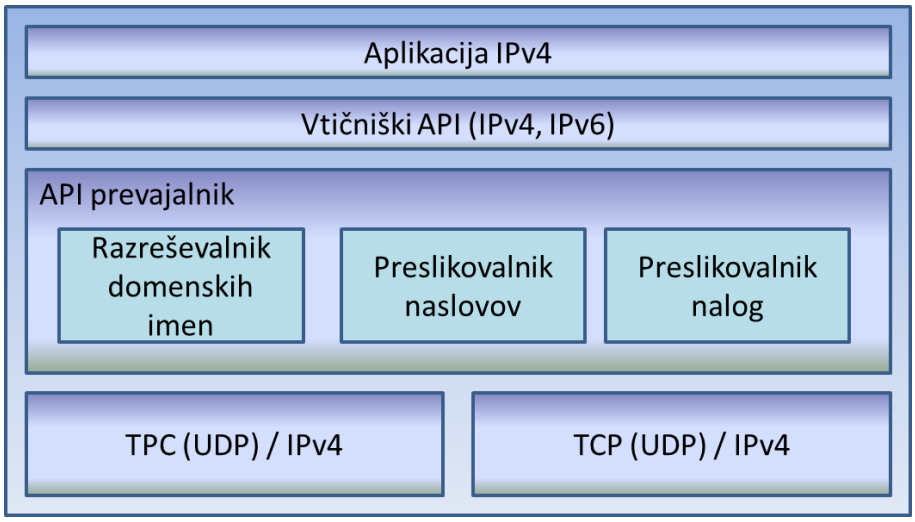
V primeru, da je mehanizem implementiran kot prevajalnik vtičniških API klicev, prevajalnik prestreže vtičniške API klice IPv4 ter jih pretvori v vtičniške API klice IPv6 in obratno. Delovanje te implementacije mehanizma je podobno delovanju mehanizmu Bump-in-the-API. V primeru, da pa je mehanizem implementiran na omrežnem nivoju, pa le ta prestreže pakete IPv4 ter jih preslika v pakete IPv6 ter obratno. Delovanje te aplikacije pa je podobno delovanju mehanizma Bump-in-the-Stack.

Terminologija, uporabljena pri mehanizmu BIH:

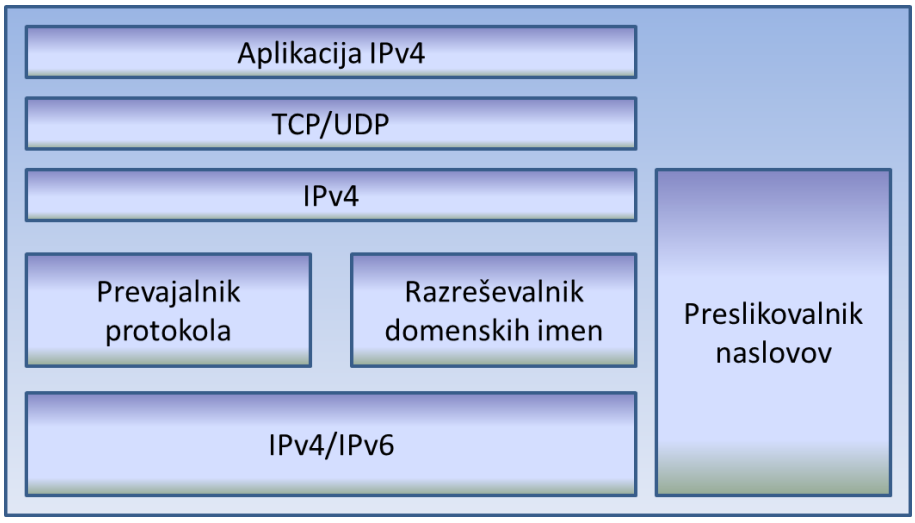
- *DNS sinteza*: Proces generiranja zapisa A za ustvarjen naslovi IPv4.
- *Pravi naslov IPv4*: Naslov IPv4 oddaljenega vozlišča, dobljenega preko zapisa A poizvedbe DNS
- *Pravi naslov IPv6*: Naslov IPv6 oddaljenega vozlišča, dobljenega preko zapisa AAAA poizvedbe DNS
- *Generirani naslov IPv4*: Naslov IPv4, ki se uporablja le znotraj naprave, ki ima implementiran mehanizem BIH in se uporablja le kot IPv4 predstavitev vozlišča IPv6 aplikaciji, ki podpira samo protokolni sklad IPv4.

Preslikovalnik nalog

Naloga preslikovalnika nalog je preslikava vtičniškega API IPv4 klica v vtičniški API IPv6 klic in obratno. Ko komponenta zazna klic aplikacije IPv4, ki podpira le protokolni sklad IPv4, ta klic prestreže in generira ustrezen klic IPv6 ter obratno. Potrebno je poudariti še, da v ta komponenta ne sme prestreći vseh klicev, ki jih generira aplikacija IPv4. Če aplikacija npr. genera klic za naslov IPv4 ki je sicer znotraj bazena naslovov IPv4, ki se uporabljajo za preslikavo, vendar preslikave za ta naslov še ni, klica ne sme pretvoriti v klic IPv6, saj v tem primeru aplikacija želi komunicirati z napravo IPv4.



Slika 3.13: Arhitektura naprave, ki ima implementiran mehanizem BIH kot vtičniški API prevajalnik



Slika 3.14: Arhitektura naprave, ki ima implementiran mehanizem BIH kot prevajalnik na mrežnem nivoju

Prevajalnik protokola

Naloga prevajalnika protokola je preslikava paketa IPv4 v paket IPv6 in obratno s pomočjo algoritma SIIT [7]. Pravtako ta prevajalnik ne sme pretvoriti

vseh paketov IPv4, ki prihajajo od aplikacije IPv4. V primeru, da aplikacija IPv4 pošlje paket na naslov IPv4, katerega ni v lokalni preslikovalni tabeli, takega paketa ne sme pretvoriti v paket IPv6, vendar ga mora posredovati do vmesnika IPv4 naprave.

Razreševalnik domenskih imen

Razreševalnik domenskih imen skrbi za poizvedbe DNS aplikacije IPv4. Če aplikacija izvede poizvedbo A (torej poizvedbo za naslov IPv4), komponenta to poizvedbo prestreže ter generira novo poizvedbo, ki zahteva tako naslove IPv4 kot naslove IPv6. Če je na voljo naslov IPv4, komponenta ne naredi ničesar, ampak le vrne rezultat aplikaciji. Če pa naslov IPv4 ni na voljo (na voljo je le naslov IPv6), pa za dobljeni naslov IPv6 generira nov naslov IPv4, ki ga nato vrne aplikaciji. V spodnji tabeli je prikazano delovanje razreševalnika domenskih imen glede na poizvedbo aplikacije ter odgovor strežnika DNS.

Tabela 3.1: Delovanje razreševalnika domenskih imen

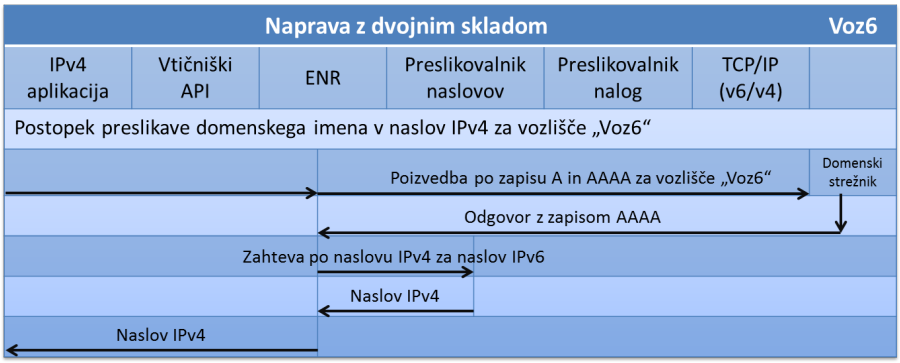
Poizvedba aplikacije IPv4	Odgovor strežnika DNS	Akcija
Naslov IPv4	Naslov IPv4	vrne pravi naslov IPv4
Naslov IPv4	Naslov IPv6	vrne generirani naslov IPv4
Naslov IPv4	Naslov IPv4/IPv6	vrne pravi naslov IPv4

Preslikovalnik naslovov

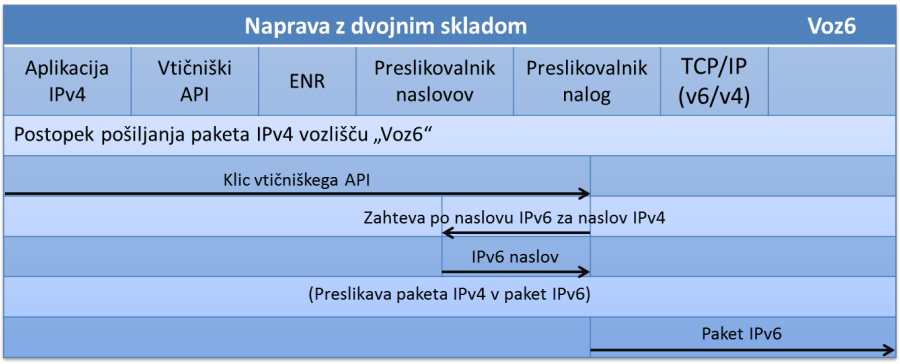
Naloga preslikovalnika naslovov je vzdrževanje tabele preslikav naslovov IPv4-IPv6, generiranje novih naslovov IPv4 ter brisanje obstoječih. Nov generirani naslov IPv4 se ustvari v primeru, ko:

- Razreševalnik domenskih imen dobi le naslov IPv6 oddaljenega vozlišča in preslikava za ta naslov še ne obstaja
- Preslikovalnik nalog prejme paket IPv6 in še ne obstaja preslikava za ponorni naslov prejetega paketa IPv6

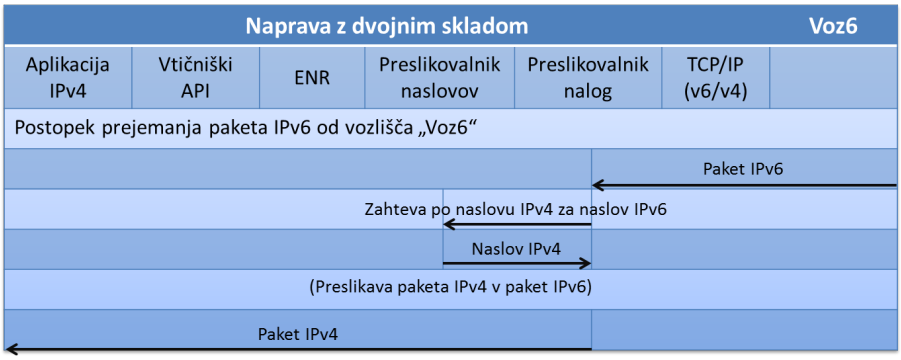
Na spodnjih slikah so primeri delovanja mehanizma BIH, implementiranega kot vtičniški API prevajalnik oziroma kot prevajalnik na omrežnem nivoju.



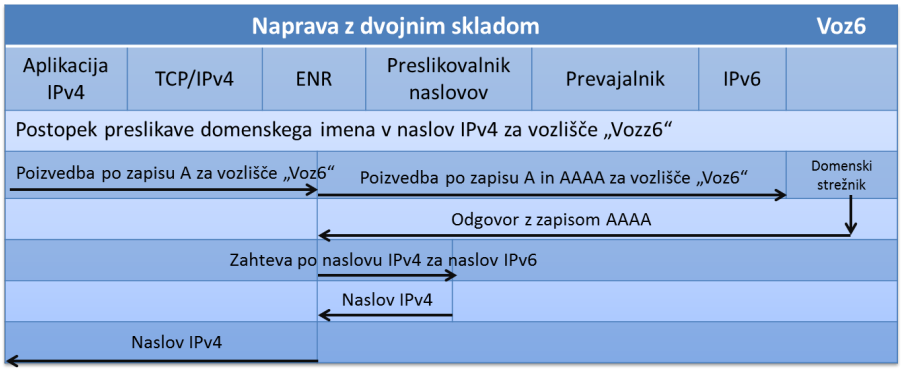
Slika 3.15: Postopek poizvedbe DNS pri mehanizmu BIH (BIA)



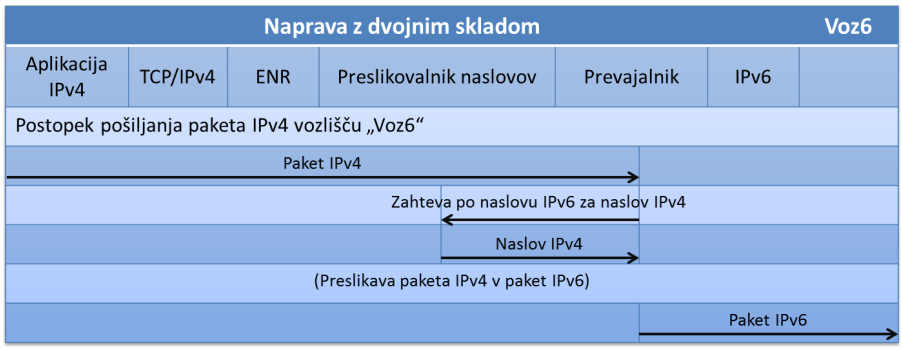
Slika 3.16: Postopek pošiljanja paketa IPv6 pri mehanizmu BIH (BIA)



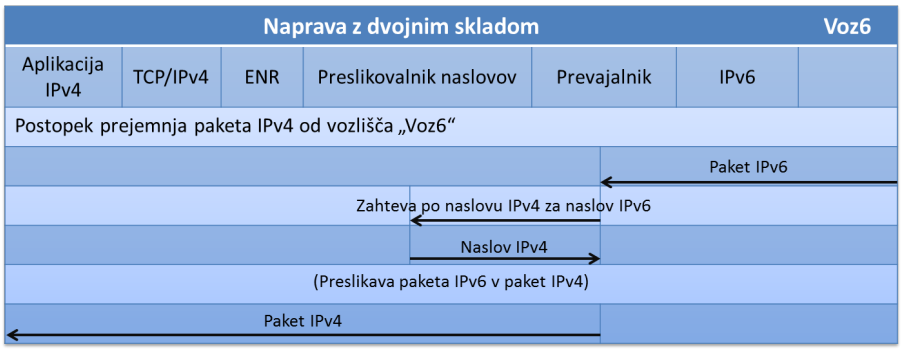
Slika 3.17: Postopek prejemanje paketa IPv6 pri mehanizmu BIH (BIA)



Slika 3.18: Postopek poivedbe DNS pri mehanizmu BIH (BIS)



Slika 3.19: Postopek pošiljanja paketa IPv6 pri mehanizmu BIH (BIS)



Slika 3.20: Postopek prejemanje paketa IPv6 pri mehanizmu BIH (BIS)

Poglavje 4

Namestitev ter testiranje mehanizma Bump-in-the-Host

Naslednje poglavje opisuje postopek namestitve ter konfiguracijo mehanizma Bump-in-the-Host na Linux operacijskem sistemu. V nadaljevanju pa sledi poskus uporabe omenjnega algoritma preko demonstracije delovanja aplikacije Skype. Skype je aplikacija, ki podpira le protokolni sklad IPv4, zato v čistem okolju IPv6 ne deluje.

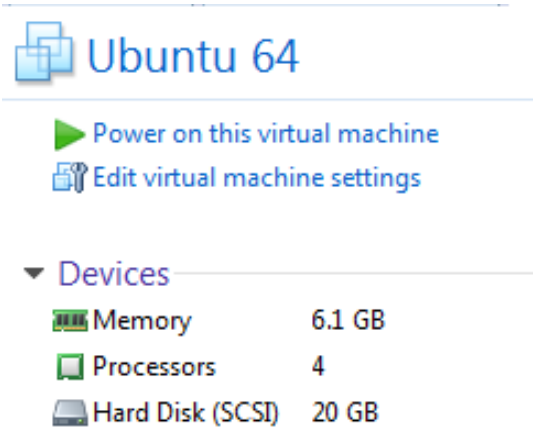
4.1 Priprava delovnega okolja

Operacijski sistem, ki sem ga uporabil pri preizkusu je bil Ubuntu 13.4 Desktop Edition AMD64. Postavljen je bil v virtualnem okolju s pomočjo aplikacije VmWare Workstation 9, na voljo pa je imel 2 procesorja s po dvema jedroma, 6.1 GB delovnega pomnilnika ter 20 GB prostora na trdem disku.

Privzeta verzija jedra je bila 3.8.0-31-generic, za potrebe testiranja mehanizma Bump-in-the-Host pa sem namestil verzijo 2.6.32-61.

Za potrebe testiranja mehanizma Bump-in-the-Host pa so bile nameščene še naslednje aplikacije:

- Subversion 1.7.5



Slika 4.1: Strojna oprema, namenjena testiranju

- Aptitude 0.6.8.1
- Wireshark 1.8.2
- Skype 4.2.0.11-1

4.2 Prednamestitvene zahteve mehanizma BIH

Za uspešno namestitev in delovanje mehanizma Bump-in-the-Host je potrebno zagotoviti naslednje prednamestitvene zahteve.

1. Potreben je Linux operacijski sistem z jedrom verzije vsaj 2.6.32-24, omogočen pa mora biti tudi jedrni modul, ki podpira protokolni sklad IPv6.

2. Glede procesorja in pomnilnika ni posebnih zahtev. Zmogljivost oziroma učinkovitost mehanizma je pogojena le s hitrostjo oziroma latenco preklopa med uporabniškim in jedrnim kontekstom, kar pomeni, da je edini dejavnnik, ki vpiva na zmogljivost, hitrost procesorske enote.

4.3 Postopek namestitve mehanizma BIH

Namestitev mehanizma BIH poteka v naslednjih korakih.

1. Prenos projekta na sistem, na katerem hočemo mehanizem omogočiti. Projekt je javno dostopen na spletnem naslovu *http://code.google.com/p/bump-in-the-host/*, najlažje pa ga prenesemo s pomočjo enega od programov za nadzor različic (npr. TortoiseSVN)

2. Projekt razširimo s pomočjo ukaza *tar zxvpf bih.tar.gz*

3. Premaknemo se v mapo, kamor smo paket razpakirali ter z ukazom *make* prevedemo jedrne module BIH algoritma

4. Premaknemo se še en nivo globje v mapo dn6, kjer ponovno z ukazom *make* prevedemo ustrezne pakete, potrebne pri uporabi IPv6

5. Premaknemo se en nivo višje ter nato še v mapo kernel-module, kjer pa z ukazom *make pack* zgeneriramo zgoščeno mapo BIH.tzg, ki vsebuje naslednje datoteke:

   - BIH.ko
   - dns6
   - config.sh

4.4 Konfiguriranje mehanizma BIH

Konfiguracija mehanizma BIH se nahaja v datoteki config.sh, nastavimo pa lahko naslednje parametre:

1. Virtualni naslov IPv4 (v4addr), ki bo viden aplikaciji IPv4

   - v4addr=192.168.1.34

2. Naslov IPv6 (v6addr), skonfiguriran na fizičnem vmesniku

- v6addr=2001:da8:bf:1010::c0a8:122

3. Gateway naslov IPv6 (v6gw)

   - v6gw=2001:da8:bf:1010::c0a8:101

4. Naslov IPv6 domenskega strežnika (nameserver)

   - nameserver=2001:da8:bf:1010::1

5. Naslovni prostor virtualnega omrežja IPv4 (pool)

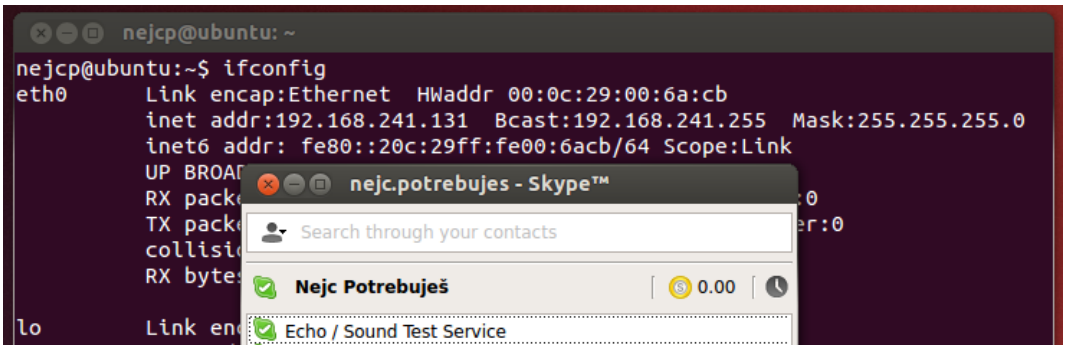
   - pool=192.168.1.30-192.168.1.40

Preslikovanje med naslovom IPv4 in naslovom IPv6 je lahko bodisi avtomatično bodisi statično. Statično preslikovanje izvedemo z ukazom:

- map="192.168.1.35 2001:da8:bf:1010::c0a8:123"

4.5 Testiranje aplikacije Skype z omogočenima protokolnima skladoma IPv4 in IPv6

Prvi primer testiranja aplikacije, ki podpira le protokolni sklad IPv4, sem izvedel tako, da sta bila omogočena tako protokolni sklad IPv4 kot protokolni sklad IPv6, mezanizem Bump-in-the-Host pa ni bil aktiven. Aplikacija je delovala pravilno.



Slika 4.2: Delovanje aplikacije Skype pri omogočenem protokolnem skladu IPv4 in protokolnem skladu IPv6

4.6 Testiranje aplikacije Skype z onemogočenim protkolnim skladom IPv4 in aktivnim mehanizmom BIH

Drugi primer testiranja aplikacije Skype pa je potekal tako, da sem najprej z ukazom *ip address del 192.168.241.131/255.255.255.0 dev eth0* na vmesniku eth0 onemogočil protokolni sklad IPv4, nato pa z ukazom *insmod bih.ko* omogočil delovanje mehanizma Bump-in-the-Host.

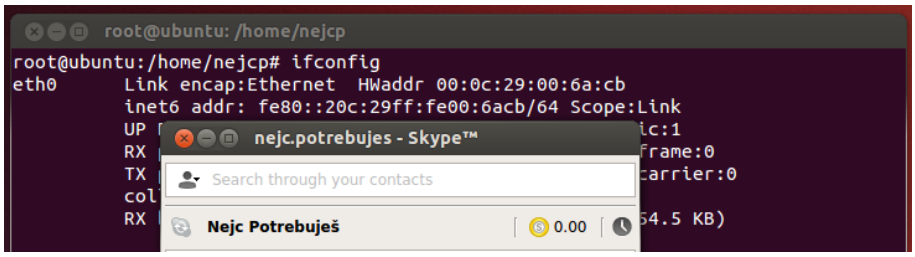
Ker lahko mehanizem BIH deluje v dveh načinih delovanja (kot mehanizem BIS oziroma kot mehanizem BIA), sem preizkusil oba. Način delovanja mehanizma BIH določimo z naslednjim ukazi:

- echo "BIS» /proc/bih/mode ali
- echo "BIA» /proc/bih/mode

Najprej sem izvedel testiranje delovanja mehanizma BIH v načinu BIS. V tem primeru mehanizem BIH prestreže pakete, ki se prenašajo med slojem TCP/IP in gonilniki mrežnih kartic. Paket IPv4, ki ga mehanizem prestreže,

pretvori v paket IPv6 in ga pošlje naprej gonilnikom mrežnih kartic ter obratno. Rezutat testiranja mehanizma v tem načinu delovanja je bil neuspešen. Povezava aplikacije Skype s strežnikom je bila neuspešna. Vzrok za napako sem najprej želel izločiti v aplikaciji Skype s pregledom datotek, ki jih aplikacija zgenerira tekom delovanja in v katerih so zapisane podrobnosti o delovanju aplikacije (angl. log files). Ker pa je vsebina teh datotek kriptirana in posledično nisem dobil podatkov o vzroku za napako, sem nato z aplikacijo Wireshark poskušal zajeti morebitne pakete, ki bi jih aplikacija Skype želela poslati v omrežje. Ob zajemu prometa na vmesniku sem glede na izvorni naslov paketov ugotovil, da nobeden od paketov ne ustreza paketom, ki bi jih aplikacija Skype želela poslati v omrežje. Zadnja možnost, s katero bi lahko odkril vzrok za napako pri povezovanju aplikacije Skype v omrežje je pregled delovanja mehanizma BIH. Pri pregledu slednjega sem ugotovil, da kljub pravilnim nastavitvam preslikave naslova IPv4 v naslov IPv6, mehanizem BIH ni uspel vrniti pravega naslova IPv6, ki bi ga uporabili za izvorni naslov paketov IPv6, poslanih v omrežje.

Drugo testiranje delovanja mehanizma BIH sem izvedel tako, da sem mehanizem prekonfiguriral tako, da je deloval v načinu BIA. V tem primeru mehanizem BIH prestreže klic IPv4 vtičniškega API ter ga nadomesti s klicem IPv6 vtičniškega API in obratno. Tudi rezultat tega testiranja je bil negativen. Povezava aplikacije Skype s strežnikom je bila neuspešna. Vzroke za napako sem iskal na podoben način kot pri prvem testiranju. Tudi v tem primeru aplikacija Wireshark ni zajela nobenega paketa, ki bi jih aplikacija Skype želela poslati v omrežje. Po pregledu podatkov o delovanju mehanizma BIH med poskusom komuniciranja aplikacije Skype s strežnikom sem ugotovil, da je vzrok za težavo enak kot pri prvem testiranju. Tudi tukaj mehanizem BIH na podlagi podanega naslova IPv4 ni uspel zgenerirati naslova IPv6, ki bi ga uporabili za izvorni naslov paketov IPv6, poslanih v omrežje.



Slika 4.3: Delovanje aplikacije Skype pri onemogočenem protokolnem skladu IPv4 in aktivnim mehanizmom BIH

Na spletu je možno zaslediti veliko predlogov za implementacijo različnih translacijskih mehanizmov. Dejanskih implementacij je bolj malo, edina javno dostopna implementacija, ki sem jo našel, pa je implementacija mehanizma BIH, ki sem jo testiral. Na spletu je dostopno poročilo, ki so ga objavili avtorji mehanizma, iz katerega je razvidno, da mehanizem BIH deluje. Rezultati mojega testiranja pa so bili negativni, mogoče tudi zaradi slabe dokumentacije mehanizma oziroma slabih navodil za konfiguracijo mehanizma. Iz zgornjih rezultatov je razvidno, da translacijski mehanizem BIH še ni zrel za širšo uporabo. Potrebno bo še veliko dodelav in razvoja, da bo mehanizem postal uporaben.

Poglavje 5

Zaključek

Prehod iz starih omrežij IPv4 na nova omrežja IPv6 je v današnjem času mnogo bolj zapleten kot je bil predviden ob pojavitvi protokola IPv6. Ob pojavitvi protokola IPv6 je bilo predvideno, da bi postopoma na vseh napravah v omrežju omogočili protokol IPv6. Ko bi bil protokol IPv6 omgočen na vseh napravah, protokol IPv4 ne bi bil več potreben in bi ga lahko izklopili. Vendar je stanje v realnosti drugačno od teorije. Ob posameznem vklapljanju protokola IPv6 so nastali manjši otoki omrežij IPv6 znotraj omrežja IPv4. Eden od problemov, ki se pojavi v tem primeru je, kako omogočiti povezavo dveh IPv6 omrežij, obdanih z omrežjem IPv4. V ta namen je razvitih mnogo tunelskih prehodnih mehanizmov. Nekateri od njih se ne uporabljajo več, drugi pa so osnova za razvoj novih. Drugi problem, ki pa se pojavi v zgornjem primeru, pa je omogočiti povezovanje naprav, ki so v omrežju IPv4, z napravami, ki se nahajajo v omrežju IPv6. V ta namen pa je razvitih mnogo translacijskih prehodnih mehanizmov. Namen teh mehanizmov je preslikati naslov IPv6 v naslov IPv4 na meji med omrežjema IPv4 in IPv6. Problem, ki se pojavi ob preslikovanju naslovov IPv4 v naslove IPv6 je število naslovov, ki jih imamo pri posameznem protokolu na razpolago. Ker je število naslovov IPv4 mnogo manj od števila naslovov IPv6, se pogosto en naslov IPv6 preslika v več različnih naslovov IPv4. Tudi če v bližnji prihodnosti na večini omrežnih povezav omogočimo protokol IPv6 pa se bodo lahko na

samih napravah izvajale aplikacije, ki pa podpirajo le protokolni sklad IPv4. V tem primeru je potrebno zagotoviti preslikovanje naslovov IPv4 v naslove IPv6 in obratno že na nivoju operacijskega sistema.
Namen tega dokumenta je predstaviti delovanje enega od translacijskih prehodnih mehanizmov, ki omogoča aplikacijam, ki podpirajo samo protokolni sklad IPv4, dostopati do vsebin, ki so dostopne le preko omrežja IPv6. Na začetku sta najprej predstavljena oba protokola, protokol IPv4 in protokol IPv6, njune slabosti oziroma izboljšave, nato pa sledi pregled prehodnih mehanizmov IPv4/v6. Na kratko je predstavljen mehanizem z dvojnim skladom ter nekaj tunelskih mehanizmov. Kasneje se osredotočimo na translacijske mehanizme in sicer na tiste mehanizme, ki omogočajo aplikacijam, ki podpirajo le protokolni sklad IPv6, dostopati do vsebin, dostopnih le preko omrežja IPv6. Podrobneje je predstavljen mehanizem Bump-in-the-Host, katerega implementacijo sem kasneje tudi testiral s pomočjo aplikacije Skype. Testiranje mehanizma BIH nam sicer ni prineslo željenega rezultata, tj. uspešne povezave aplikacije, ki podpira le protokolni sklad IPv4, v omrežje IPv6, smo pa prikazali koncept delovanja, ki tovrstnim aplikacijam omogoča povezovanje v omrežje IPv6.

Literatura

[1] TCP/IP Overview and History, The TCP/IP Guide, Accessed 01-October-2013, [Online]. Available: http://www.tcpipguide.com/free/t_TCPIPOverviewandHistory.htm

[2] Variable Length Subnet Mask (VLSM), Orbit Computer Solutions, Accessed 07-October-2013, [Online]. Available: http://www.orbit-computer-solutions.com/VLSM.php.

[3] Classless Inter-Domain Routing, WikiPedia, Accessed 03-October-2013, [Online]. Available: http://en.wikipedia.org/wiki/Classless-Inter-Domain-Routing.

[4] Dynamic Host Configuration Protocol,WikiPedia, Accessed 02-October-2013, [Online]. http://en.wikipedia.org/wiki/Dynamic-Host-Configuration-Protocol.

[5] P. Srisuresh, and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations,"RFC 2663 (Proposed Standard), Internet Engineering Task Force, Aug. 1999 [Online]. Avaliable: http://www.ietf.org/rfc/rfc2663.

[6] Comparison of IPv6 application support, WikiPedia, Accessed 01-October-2013, [Online]. Available: http://en.wikipedia.org/wiki/Comparison-of-IPv6-application-support.

[7] E. Nordmark, Štateless IP/ICMP Translation Algorithm (SIIT),"RFC 2765 (Proposed Standard), Internet Engineering Task Force, Feb. 2000 [Online]. Avaliable: http://www.ietf.org/rfc/rfc2765.

[8] M. Bagnuloi, P. Matthews, and I. van Beijnum, Štateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers,"RFC 6146 (Proposed Standard), Internet Engineering Task Force, Apr. 2011 [Online]. Avaliable: http://www.ietf.org/rfc/rfc6146.

[9] M. Mawatari, M. Mawashima, and C. Bryne, "464XLAT: Combination of Stateful and Stateless Translation,"Internet-Draft (work in progress), draft-ietf-v6ops-464xlat-08, Internet Engineering Task Force, Sep. 2012 [Online]. Avaliable: http://tools.ietf.org/html/draft-ietf-v6ops-464xlat-08.

[10] K. Tsuchiya, H. Higuchi and Y. Atarashi, "Dual-Stack Hosts Using "Bump-in-the-Stack"Technique (BIH),"RFC 2767 (Proposed Standard), Internet Engineering Task Force, Feb. 2000 [Online]. Avaliable: http://www.ietf.org/rfc/rfc2767.

[11] S. Lee, M-K. Shin and Y-J. Kim, "Dual-Stack Hosts Using "Bump-in-the-API"(BIA),"RFC 3338 (Proposed Standard), Internet Engineering Task Force, Oct. 2002 [Online]. Avaliable: http://www.ietf.org/rfc/rfc3338.

[12] B. Huang, H. Deng, and T. Savolainen, "Dual-Stack Hosts Using "Bump-in-the-Host"(BIH),"RFC 6535 (Proposed Standard), Internet Engineering Task Force, Feb. 2012 [Online]. Avaliable: http://www.ietf.org/rfc/rfc6535.