Kai Saarnia

IPv6-käyttöönotto palveluntarjoajan konesaliverkossa

Sähkotekniikan korkeakoulu

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-insinöörin tutkintoa varten Espoossa 20.1.2014.

Työn valvoja:

Prof. Jukka Manner

Työn ohjaaja:

DI Pauli Niemi

Aalto-yliopisto
Sähkötekniikan korkeakoulu

AALTO-YLIOPISTO
SÄHKOTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Kai Saarnia

Työn nimi: IPv6-käyttöönotto palveluntarjoajan konesaliverkossa

Päivämäärä: 20.1.2014 Kieli: Suomi Sivumäärä:9+112

Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoverkkotekniikka Koodi: S-38

Valvoja: Prof. Jukka Manner

Ohjaaja: DI Pauli Niemi

Tämä diplomityö on tehty toimeksiantona Capgemini Finland Oy:lle (myöh. Capgemini). Sen tavoitteena on ottaa IPv6-protokolla käyttöön Capgeminin konesaliverkossa niin, että se on saavutettavissa Internetistä IPv4-protokollan lisäksi myös IPv6-protokollalla.

Työn ensimmäisessä luvussa kerrotaan lyhyesti siitä, mitkä tämän työn taustat ja tavoitteet ovat sekä minkä ongelman ja osaongelmat se ratkaisee. Toisessa luvussa kerrotaan, mitkä IPv4-protokollan ongelmat ovat ja miksi IPv6-protokolla lopulta korvaa sen. Kolmannessa luvussa esitellään IPv6-protokollaa ja sen tukiprotokollia IETF:n (*Internet Engineering Task Force*) RFC-dokumenttien (*Request For Comments*) ja kirjallisuuden pohjalta. Neljännessä luvussa perehdytään lyhyesti IPv6-protokollan tietoturvaan IPv6-käyttöönottoon liittyen ja kerrotaan, millaisia IPv6-transitiomekanismeja on olemassa. Viidennessä luvussa näytetään ensin tyypillinen palvelinkeskuksen konesaliverkon verkkotopologia ja esitellään sen jälkeen Capgeminin konesaliverkon rakenne. Kuudennessa luvussa yhdistetään Capgeminin konesaliverkko Internetiin IPv6-protokollalla ja rakennetaan Capeminin laboratorioon IPv6-testiverkko. Luvussa kehitetään myös konsepti, jolla voidaan provisioida IPv6-protokollalla toimiva www-palvelu Capgeminin konesaliverkossa mahdollisimman helposti ja kustannustehokkaasti. Lopuksi seitsemännessä luvussa käydään läpi IPv6-käyttöönoton tulokset, seuraukset ja siinä esiintyneet haasteet sekä tehdään suunnitelma siitä, mitkä ovat seuraavat askeleet IPv6-protokollan laajemmalle käyttöönotolle Capgeminin konesaliverkossa.

Avainsanat: IPv6, Internet-protokolla, käyttöönotto, palveluntarjoaja, konesaliverkko, palvelinkeskus

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Kai Saarnia

Title: IPv6 Deployment in a Service Provider's Data Center Network

Date: 20.1.2014 Language: Finnish Number of pages:9+112

Department of Communications and Networking

Professorship: Networking Technology Code: S-38

Supervisor: Prof. Jukka Manner

Advisor: M.Sc. (Tech.) Pauli Niemi

This Master's thesis was done for Capgemini Finland Oy (later referred to as Capgemini). The objective of the thesis is to deploy the IPv6 protocol in Capgemini's data center network so that it is reachable from the Internet also via IPv6 in addition to IPv4.

In the first chapter of the thesis the background and objectives of the thesis in addition to the problem it solves are discussed. In the second chapter the inadequacy of the IPv4 protocol and the reasons why IPv6 will eventually replace it are explained. In the third chapter the IPv6 base protocol and its supporting protocols are presented based on RFC (*Request For Comments*) documents published by the IETF (*Internet Engineering Task Force*) and literature. In the fourth chapter IPv6 security with respect to the IPv6 deployment and IPv6 transition mechanisms are introduced. In the fifth chapter, a typical data center network topology is first shown after which the Capgemini data center network is showcased. In the sixth chapter the Capgemini data center network is connected to the Internet via IPv6 and an IPv6 test network is set up in the Capgemini laboratory. A proof of concept to provision an IPv6 web service in the Capgemini data center network with minimal capital and operational expenditure is also developed. Finally, in the seventh chapter the results, consequences and challenges of the IPv6 deployment are reviewed and a plan is made as to what the next steps for a more comprehensive IPv6 deployment in the Capgemini data center network are.

Keywords: IPv6, Internet Protocol, deployment, service provider, data center network

Esipuhe

Haluan kiittää valvojaani, professori Jukka Manneria ja ohjaajaani, DI Pauli Niemeä hyvistä vinkeistä tämän diplomityön kirjoitusprosessin aikana. Haluan kiittää myös isääni, DI Raimo Saarniaa työn oikolukemisesta useaan kertaan ja kollegojani, DI Lauri Turusta ja ins. Juska Kettusta avusta IPv6-käyttöönotossa sekä lopuksi Capgemini Finland Oy:tä mahdollisuudesta tehdä tämä diplomityö muun työn ohessa.

Otaniemi, 20.1.2014

Kai Saarnia

Sisältö

Lyhenteet

6over4	Transmission of IPv6 over IPv4 Domains without Explicit Tunnels
6rd	IPv6 Rapid Deployment on IPv4 Infrastructures
6to4	Connection of IPv6 Domains via IPv4 Clouds
AAAA	Quad-A, IPv6-DNS-tietue
ACE	Application Control Engine
ACL	Access Control List
APN	Access Point Name
ARP	Address Resolution Protocol
ASIC	Application Specific Integrated Circuit
BGP	Border Gateway Protocol
BIH	Bump-in-the-Host
CAPEX	Capital Expenditure
CATNIP	Common Architecture for Next-generation Internet Protocol
CE	Customer Edge
CEF	Cisco Express Forwarding
CIDR	Clasless Inter-Domain Routing
CLNP	Connectionless Network Protocol
CoS	Class of Service
CWDM	Coarse Wavelength Division Multiplexing
DAD	Duplicate Address Detection
DDNS	Dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zone
DNS	Domain Name Service
eBGP	Exterior BGP
EGP	Exterior Gateway Protocol
ESP	Encapsulated Security Payload
ESXi	Elastic Sky X Integrated
FDDI	Fiber Distributed Data Interface
FEX	Fabric Extender
GUA	Global Unicast address
HSRP	Hot Standby Routing Protocol
IAB	Internet Architecture Board
IANA	Internet Assigned Numbers Authority
iBGP	Internal BGP
ICANN	Internet Corporation for Assigned Names and Numbers
ICE	Interactive Connectivity Establishment
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol

IHL	Internet Header Length
IMS	IP Multimedia Subsystem
IOS	Internetwork Operating System
IP	Internet Protocol
IPAM	IP Address Management
IPng	IP The Next Generation
IPS	Intrusion Prevention System
ISATAP	Intra-Site Automatic Tunnel Addressing Protocol
ISC	Internet Systems Consortium
ISG	Integrated Services Gateway
IS-IS	Intermediate System to Intermediate System
ISO	International Organization for Standardization
ISOC	Internet Society
ISP	Internet Service Provider
LAN	Local Area Network
LIR	Local Internet Registry
LNP	Local Network Protection
LSA	Link-State Advertisement
LSP	Link-State Protocol Data Unit
LTE	Long Term Evolution
MAC	Media Access Control
MAN	Metropolitan Area Network
MBGP	Multiprotocol BGP
MLD	Multicast Listener Discovery
MTU	Maximum Transmission Unit
NA	Neighbor Advertisement
NA(P)T-PT	Network Address (Port) Translation - Protocol Translation
NAT	Network Address Translation
NDP	Neighbor Discovery Protocol
NLPID	Network Layer Protocol Identifier
NLRI	Network Layer Reachability Information
NS	Neighbor Solicitation
NSSA	Not-So-Stubby Area
NTP	Network Time Protocol
NUD	Neighbor Unreachability Detection
OPEX	Operational Expenditure
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OUI	Organizationally Unique Identifier
PA	Provider Assigned
PACL	Port ACL
PE	Provider Edge
PI	Provider Independent
PIP	'P' Internet Protocol
PMTUD	Path MTU Discovery

QoS	Quality of Service
RA	Router Advertisement
RD	Router Discovery
RFC	Request For Comments
RIP	Routing Information Protocol
RIPE	Réseaux IP Européens
RIR	Regional Internet Registry
RS	Router Solicitation
SIIT	Stateless IP/ICMP Translation
SIP	Simple IP
SIPP	Simple IP Plus
SLAAC	Stateless Address Autoconfiguration
SOCKS	Socket Secure
SSG	Secure Services Gateway
STUN	Session Traversal Utilities for NAT
TCAM	Ternary Content Addressable Memory
Teredo	Tunneling IPv6 over UDP through Network Address Translations (NATs)
TLV	Type-Length-Value
ToS	Type of Service
TRT	Transport Relay Translation
TSP	Tunnel Setup Protocol
TTL	Time to Live
TUBA	TCP and UDP over Bigger Addresses
TURN	Traversal Using Relays around NAT
UE	User Equipment
ULA	Unique Local Address
URPF	Unicast Reverse Path Forwarding
WAN	Wide Area Network
VEM	Virtual Ethernet Module
vGW	Virtual Gateway
VLAN	Virtual LAN
WLAN	Wireless LAN
VoIP	Voice Over IP
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding
VSM	Virtual Supervisor Module
VTY	Virtual Terminal Line

1 Johdanto

IETF:n (*Internet Engineering Task Force*) RFC-dokumentti (*Request For Comments*) numero 791 *Internet Protocol* julkaistiin syyskuussa vuonna 1981 [1]. Silloin kenelläkään ei käynyt mielessä, että 32-bittisten lähde- ja kohdeosoitteiden valinta osoittautuisi virhearvioksi jo 15 vuotta myöhemmin. Osoitteet eivät toki loppuneet vielä vuonna 1996, mutta jo silloin oli selvää, että reilut neljä miljardia osoitetta eivät riitä kovin pitkälle 2000-luvulle. Globaalisti IP-osoitteita hallinnoiva IANA (*Internet Assigned Numbers Authority*) jakoi viimeiset vapaat IPv4-osoitealueet alueellisille Internet-registraateille (RIR, *Regional Internet Registry*) AfriNIC:lle, APNIC:lle, ARIN:lle, LACNIC:lle ja RIPE NCC:lle 3.2.2011 [2]. Euroopan IP-osoitteita hallinnoiva RIPE NCC puolestaan alkoi jakaa paikallisille Internet-registraateille (LIR, *Local Internet Registry*) eli esim. Internet-operaattoreille osoitteita viimeisestä vapaana olevasta /8-osoiteavaruudesta 14.9.2012 [3]. Yhteen /8-avaruuteen (verkkomaski 255.0.0.0) mahtuu 16777216 IP-osoitetta, ja yksi LIR voi saada tästä avaruudesta suurimmillaan /22-kokoisen alueen, joka käsittää 1024 IPv4-osoitetta. LIR ei ole edes oikeutettu hakemaan tätä /22-aluetta, jos sille ei vielä ole allokoitu IPv6-osoitealuetta. [4] ICANN (*The Internet Corporation for Assigned Names and Numbers*) otsikoikin 3.2.2011 julkistamansa tiedotteen *The Future Rests with IPv6* [5].

1.1 Tavoitteet

Miksi ongelmaan herätään vasta nyt, kun IPv4-osoitteet uhkaavat loppua kokonaan? IETF julkaisi ensimmäisen IPv6-määrittelyn jo joulukuussa 1995, ja päivitetty versio julkaistiin kolme vuotta myöhemmin [6, 7]. Tämän diplomityön tavoitteena onkin ensin selvittää miksi nyt, 18 vuotta alkuperäisen ja 15 vuotta päivitetyn version julkaisemisen jälkeen IPv6-protokollaa ei vieläkään ole otettu laajalti käyttöön Internetissä yritysten ja muiden organisaatioiden omista verkoista puhumattakaan. Työn varsinainen tavoite on kuitenkin sen jälkeen tutkia, kuinka Capgemini voi ottaa IPv6-protokollan käyttöön omassa konesaliverkossaan. Capgeminin tavoitteena on olla valmis tarjoamaan IPv6-palveluita vuoden 2013 loppuun mennessä. Monet konesalipalveluista kiinnostuneet asiakkaat vaativat nykyään palveluilleen myös IPv6-tukea ja tavoitteena onkin, että tämän työn seurauksena Capgeminilla on valmiudet tarjota nykyisille ja potentiaalisille asiakkailleen IPv6-palveluita konesaliverkossaan.

1.2 Ongelma ja osaongelmat

Työn varsinainen ongelma, jonka tämä diplomityö ratkaisee on se, kuinka IPv6 saadaan otettua käyttöön Capgeminin konesaliverkossa häiriöttömästi ja niin, että käyttöönotosta ei aiheudu haittaa Capgeminin omille eikä sen asiakkaiden palveluille. Osaongelmia aiheuttaa se, että Capgemini on mm. jatkuvia ulkoistuspalveluita tarjoava yritys ja kaikki muutokset sen tuotantoinfrastruktuuriin on tehtävä ajaltaan rajattujen huoltoikkunoiden aikana. Tästä seuraa se, että kovinkaan laajoja muutoksia ei yhden huoltoikkunan aikana voi tehdä ja toisaalta myös se, että muutokset on oltava hyvin tarkkaan suunniteltuja ja testattuja.

1.3 Tulokset

Tässä diplomityössä osoitetaan, miksi IPv6-protokolla korvaa IPv4-protokollan ja otetaan se käyttöön Capgeminin konesaliverkossa hallitusti ja häiriöttömästi siten, että käyttöönotosta ei aiheudu haittaa Capgeminin omille eikä sen asiakkaiden palveluille. Työssä yhdistetään palveluntarjoajan konesaliverkko Internetiin IPv6-protokollalla parhaiden käytäntöjen mukaisesti ja rakennetaan IPv6-testiverkko palveluntarjoajan laboratorioon. Lopuksi työssä kehitetään konsepti IPv6-www-palvelun provisiointiin Capgeminin konesaliverkossa mahdollisimman kustannustehokkaasti.

1.4 Rajaus ja rakenne

Tämä diplomityö on rajattu Capgeminin konesaliverkon verkkokomponenttien osalta niin, että työssä keskitytään itse konesali(lähi)verkkoon, palomuurialustoihin, Internet-reitittimiin ja nimipalveluun (DNS, *Domain Name Service*). Muut verkon komponentit, kuten välityspalvelimet, kuormanjakajat, langaton lähiverkko (WLAN, *Wireless Local Area Network*) ja etäyhteyspalvelu (VPN, *Virtual Private Network*) ovat tämän työn tarkastelun ulkopuolella.

Työn toisessa luvussa kerrotaan, mitkä IPv4-protokollan ongelmat ovat ja miksi IPv6-protokolla lopulta korvaa sen. Kolmannessa luvussa esitellään IPv6-protokollaa ja sen tukiprotokollia kirjallisuuden, RFC-dokumenttien ja muiden artikkelien pohjalta ja verrataan sitä IPv4-protokollaan. Neljännessä luvussa kerrotaan lyhyesti IPv6:n tietoturvasta IPv6-käyttöönottoon liittyen ja näytetään, millaisia IPv6-transitiomekanismeja on olemassa. Viidennessä luvussa keskitytään Capgeminin verkkoinfrastruktuuriin ja mietitään tarkemmin, kuinka IPv6 voidaan ottaa käyttöön em. verkkokomponenteissa. Itse IPv6-käyttöönotto tehdään kuudennessa luvussa ja siinä yhdistetään Capgeminin konesaliverkko Internetiin IPv6-protokollalla sekä rakennetaan IPv6-testiverkko Capgeminin laboratorioon. Viimeisessä, seitsemännessä luvussa tarkastellaan IPv6-käyttöönottoa, siinä ilmenneitä ongelmia ja haasteita sekä kerrotaan, mitkä ovat seuraavat askeleet IPv6-protokollan laajemmalle käyttöönotolle Capgeminin konesaliverkossa.

Koska alan kirjallisuus ja termistö on pääosin englanniksi, kaikki työssä käytetyt termit on esitetty myös englanniksi, jotta lukijalle ei jäisi epäselväksi, mitä missäkin kohtaa tarkoitetaan. Näitä termejä on niin paljon, että sana *engl.* on tarkoituksella jätetty pois alkuperäisen termin edestä tekstin luettavuuden kannalta. Englanninkielinen termi seuraa suomeksi käännettyä termiä heti sen jälkeen suluissa kursiivilla kirjoitettuna, esim. otsikko (*header*). Tekstin luettavuuden parantamiseksi myös siinä ensimmäistä kertaa esiintyvät lyhenteet on avattu. Työn aiheesta johtuen siinä on käytetty lähteinä paljon IETF:n julkaisemia RFC-dokumentteja, ja niissä käytetyt avainsanat on tässä työssä käännetty seuraavasti: [8]

MUST, *REQUIRED, SHALL*	täytyy
MUST NOT, *SHALL NOT*	ei saa
SHOULD, *RECOMMENDED*	pitäisi
SHOULD NOT, *NOT RECOMMENDED*	ei pitäisi
MAY, *OPTIONAL*	saattaa

2 Tarve IPv6-protokollalle

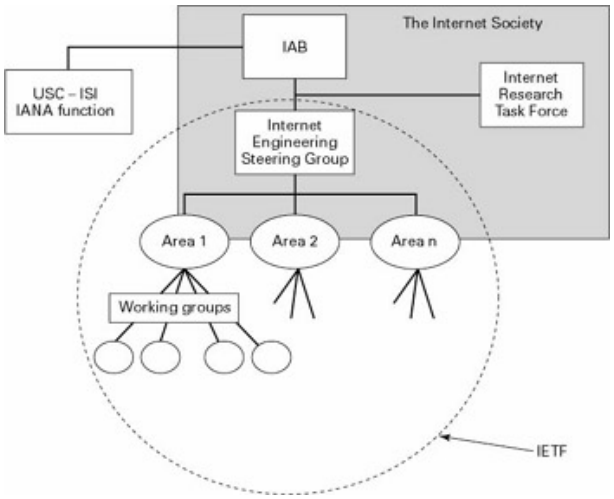
Tässä luvussa kerrotaan ensin IPv6-protokollan historiasta ja siitä, miksi se korvaa tulevaisuudessa IPv4-protokollan. Sen jälkeen mietitään, miksi IPv6-käyttöönotto on ollut niin hidasta kuin se on ollut ja kerrotaan *World IPv6* - ja *World IPv6 Launch* -päivistä, joiden tarkoituksena on ollut vauhdittaa IPv6-käyttöönottoa.

2.1 Historiaa

Vuonna 1978 julkaistussa dokumentissa *The Catenet Model for Internetworking* pohditaan Internet-protokollassa käytettävien osoitteiden pituutta. Vaihtoehtona olivat olleet jopa 120-bittiset osoitteet, mutta tuolloin tultiin siihen tulokseen, että niiden käyttäminen olisi tehnyt IP-otsikoista liian pitkiä. [9] Vuonna 1980 julkaistussa IP-standardissa osoitteet olivatkin 32-bittisiä. Kahdeksaa bittiä käytettiin ilmaisemaan verkon tunnistetta, eli yksilöiviä verkkoja ei voinut olla enempää kuin 256 kappaletta. [10] Seuraavana vuonna esiteltiin A-, B- ja C-luokan osoitteet, joissa verkko ilmaistiin vastaavasti joko 7, 14 tai 21 bitillä. A-luokan ensimmäinen, B-luokan kaksi ensimmäistä ja C-luokan kolme ensimmäistä bittiä olivat varattuja. Nämä luokat olivatkin käytössä vuonna 1981 julkaistussa IPv4-standardissa. Luokat olivat kuitenkin staattisia, eli organisaatiolle tai yhteisölle oli annettava A-, B- tai C-luokan osoite sen koosta riippuen. [1, 11] Jakamalla puolet 126 (0.0.0.0 ja 127.0.0.0 ovat varattuja) mahdollisesta A-luokan verkosta ollaan käytetty jo 25% koko IPv4:n osoiteavaruudesta. Lisäksi kovin moni organisaatio tai yhteisö tuskin tarvitsee 16777216 osoitetta, jotka A-luokan verkko pitää sisällään. [12]

1990-luvun alussa huomattiin, että IPv4:n 32-bittiset osoitekentät ovat liian lyhyitä ja että Internet-reitittimien reititystaulut olivat kasvaneet liian isoiksi. Vuonna 1990 ennustettiin, että B-luokan osoitteet loppuisivat maaliskuuhun 1994 mennessä. [13] Vuonna 1993 julkaistu CIDR (*Classless Inter-Domain Routing*) mahdollisti osoitteiden jakamisen vapaammin kuin A-, B- ja C-luokkien perusteella, mutta IPv4-osoiteavaruuden ennustettiin silti loppuvan joskus vuosien 2005 ja 2011 välillä [12, 14].

IETF alkoi tutkia ongelmaa heinäkuussa 1991, ja uusi tutkimusalue nimeltä IPng (*Internet Protocol next generation*) perustettin [15, 16, 17]. IAB:n (*The Internet Architecture Board*) kokouksessa Kobessa, Japanissa kesäkuussa 1992 uudeksi Internet-protokollaksi oli esillä kolme vaihtoehtoa: tammikuussa 1992 perustetun ISOC:n (*Internet Society*) ehdottama TUBA (*TCP and UDP over Bigger Addresses*), IPv7 ja *IP in IP* [18, 19]. TUBA pohjautui ISO:n (*International Organization for Standardization* OSI-mallin (*Open Systems Interconnection*) CLNP-protokollaan (*Connectionless Network Protocol*), mikä nosti esiin kysymyksen siitä, oltiinko Internetiä myymässä ISO:lle. IAB:n TUBA-suunnitelma vedettiinkin takaisin IETF:n kokouksen aikaan heinäkuussa 1992, mikä puolestaan aiheutti keskustelua siitä, kenen tehtävä on tehdä päätöksiä ISOC:ssa. Tämä tapahtuma tunnetaan *Koben tapauksena* ja se aiheutti merkittäviä muutoksia IETF:n, IESG:n (*Internet Engineering Steering Group*) ja IAB:n nimityksissä ja päätöksenteossa. Kuva 1 havainnollistaa ISOC:n, IAB:n, IESG:n ja IETF:n suhtautumista toisiinsa vuoden 1993 aikoihin. [20, 21]



Kuva 1: ISOC, IAB, IESG & IETF vuonna 1993. [22]

R. Ullmannin IPv7:sta tuli vuonna 1993 TP/IX ja myöhemmin vuonna 1994 CATNIP (*Common Architecture for Next Generation Internet Protocol*) [19, 23]. Robert Hindenin *IP in IP:stä* (Encaps) puolestaan tuli vuonna 1993 IPAE (*IP Address Encapsulation*), jota oli tarkoitus käyttää siirtymävaiheessa kohti Steve Deeringin marraskuussa 1992 ehdottamaa SIP:iä (*Simple IP*) [24, 25]. Lopulta SIP yhdistettiin Paul Francisin PIP:iin (*'P' Internet Protocol*) ja niistä syntyi SIPP (*Simple IP Plus*) [26]. [27] IETF julkaisi joulukuussa 1994 dokumentin, joka määritteli 17 arvostelukriteeriä uudelle Internet-protokollalle. Nämä kriteerit olivat seuraavat: [28]

- skaalautuvuus
  - IPng-protokollan täytyy mahdollistaa 10^{12} päätelaitteen ja 10^9 verkon tunnistus ja osoitteistus.
- topologinen joustavuus
  - IPng:n reititysprotokollien täytyy mahdollistaa monenlaiset verkkotopologiat.
- suorituskyky
  - IPng:llä saavutettavien yhteysnopeuksien täytyy olla vastaavia kuin IPv4:llä.
- vakaa toiminta
  - Verkon ja sen reititys- ja valvontaprotokollien täytyy olla vakaasti toimivia.
- IPv4-siirtymä
  - IPng-protokollalla täytyy olla suoraviivainen siirtymäsuunnitelma IPv4:stä.
- mediariippumattomuus

  - IPng-protokollan täytyy olla yhteensopiva eri LAN- (*Local Area Network*), MAN- (*Metropolitan Area Network*) ja WAN-verkoissa (*Wide Area Network*) käytettävien siirtoteiden kanssa ja tukea linkkinopeuksia kertaluokassa 1bps-100Gbps.

- epäluotettava datagrammipalvelu
  - IPng-protokollan täytyy tukea epäluotettavaa datagrammipalvelua.
- konfigurointi, hallinnointi ja operointi
  - IPng-protokollan konfiguroinnin, hallinnoinnin ja operoinnin täytyy olla helppoa ja hajautettua, päätelaitteet ja reitittimet on oltava automaattisesti konfiguroitavissa.
- turvallinen toiminta
  - IPng:n täytyy tarjota turvallinen verkkokerros.
- yksilöllinen nimeäminen
  - IPng:n täytyy nimetä kaikki IP-kerroksen objektit yksilöivästi. Nimillä saattaa tai ei saata olla merkitystä sijainnin, topologian tai reitityksen suhteen.
- saatavuus ja dokumentointi
  - IPng:n määrittelevät, siihen liittyvät ja sen reititysprotokollat täytyy julkaista RFC-dokumentteina, ja niihin perustuvista ohjelmistoratkaisuista ei saa periä lisensöintimaksuja.
- monilähetys
  - IPng:n täytyy tukea sekä yksi- (*unicast*) että monilähetystä (*multicast*).
- laajennettavuus
  - IPng:n täytyy olla laajennettavissa, varmistaa Internetin palveluiden saatavuus myös tulevaisuudessa ja olla taaksepäin yhteensopiva.
- verkkopalvelu, QoS (*Quality of Service*)
  - IPng:n täytyy tukea QoS:ää.
- liikkuvuus
  - IPng:n täytyy tukea liikkuvuutta.
- valvontaprotokolla (*control protocol*)
  - IPng:n täytyy sisältää tuki verkkojen testaamiseen ja ongelmanratkaisuun.

- yksityisverkot
  - IPng:n täytyy tukea sekä IP-pohjaisia että ei-IP-pohjaisia yksityisverkkoja.

Toisaalta on mielenkiintoista, mitä uudelta protokollalta ei vaadittu: [28]

- fragmentointi
- IP-otsikon tarkistussumma
- palomuurituki
- verkonhallinta
- seuranta (*accounting*)
- reititys
  - skaalautuvuus
  - politiikat
  - QoS
  - palaute
  - vakaus
  - monilähetys

Esillä olleet ehdotukset arvioitiin näiden kriteerien perusteella, ja tammikuussa 1995 julkaistiin suositus uuden sukupolven Internet-protokollaksi [13]. Tässä vaiheessa jäljellä olivat enää CATNIP, SIPP ja TUBA. Kaikissa näissä oli omat ongelmansa, mutta CATNIP:ssa niitä oli eniten. IPng-työryhmä suosittelikin lopulta pienin muutoksin SIPP-protokollaa uudeksi Internet-protokollaksi. SIPP:n 64-bittiset osoitekentät vaihdettiin 128-bittisiksi, TUBA:sta ja CIDR:sta otettiin parhaita paloja ja joitain reititysotsikkoparannuksia tehtiin. Nimeksi valittiin IPv6, koska numero 5 oli varattu kokeelliselle suoratoistoprotokollalle. [21]

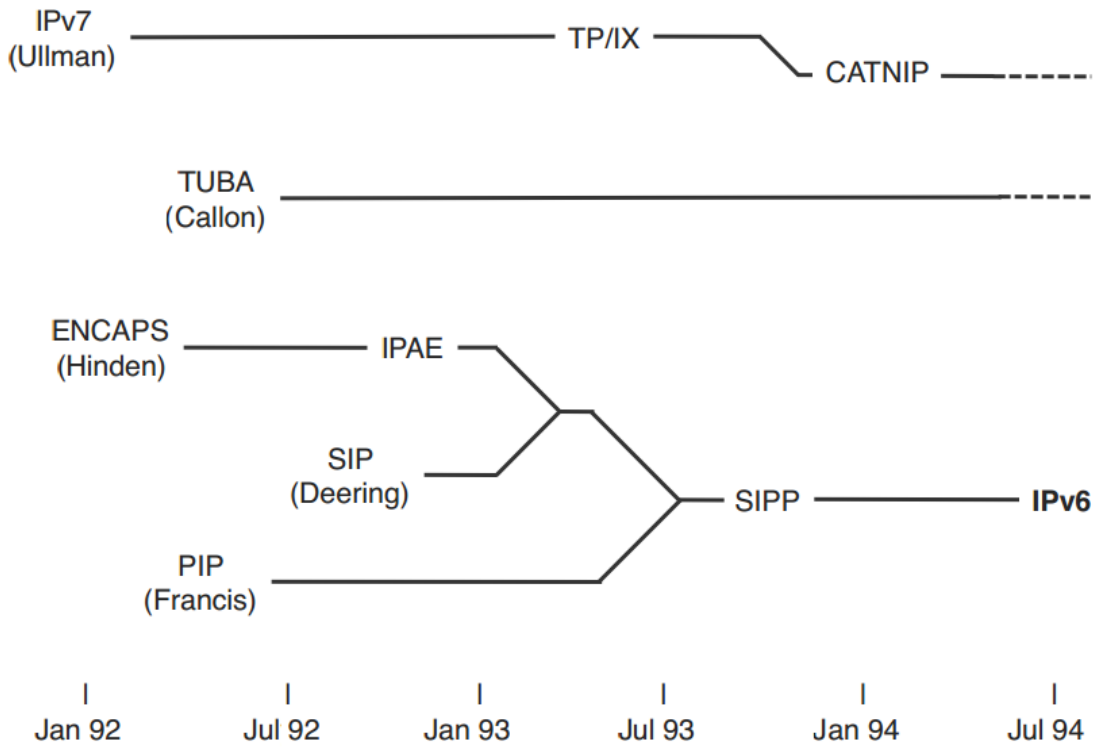
Taulukko 2: IP-versionumerot. [29]

Versio	**Nimi**
0-3	-
4	Internet Protocol (IPv4)
5	Stream Protocol (SP)
6	SIP $\rightarrow$ SIPP $\rightarrow$ IPv6
7	IPv7 $\rightarrow$ TP/IX $\rightarrow$ CATNIP
8	PIP
9	TUBA
10-15	-

Samassa suosituksessa listataan myös IPv6:n tärkeimmät ominaisuudet: [13]

- laajennettu osoitteistus ja reititys
- yksinkertaistettu otsikko
- laajennusotsikot ja optiot
- autentikointi ja yksityisyys
- automaattinen konfigurointi
- lähdereititys
- yksinkertainen ja joustava siirtymä IPv4:stä
  - vaiheittainen päivitys
  - vaiheittainen käyttöönotto
  - helppo osoitteistus
  - alhaiset käyttöönottokustannukset
- QoS-ominaisuudet

Kuva 2 havainnollistaa, kuinka esillä olleista ehdotuksista lopulta päädyttiin IPv6:een.



Kuva 2: IPv6:n kehityspolku. [15]

2.2 Tietoliikenneverkon arvo

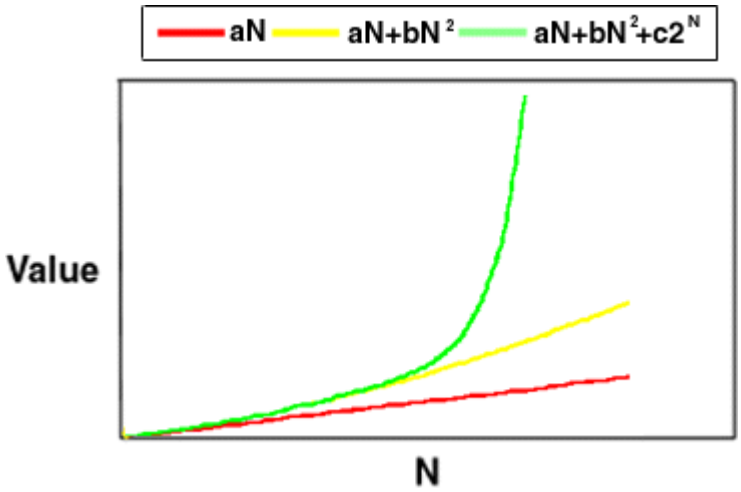
Tietoliikenneverkon arvoa siihen liittyvälle tilaajalle tai käyttäjälle kuvaamaan on kehitetty erilaisia lakeja. Ensimmäinen näistä oli Sarnoffin laki, jonka mukaan verkon arvo kasvaa lineaarisesti siihen liittyneiden käyttäjien mukaan: 100 käyttäjän verkko on 10 kertaa arvokkaampi kuin 10 käyttäjän verkko. Tämä laki kehitettiin aikanaan yksisuuntaisiin yleislähetysverkkoihin (*broadcast*). Seuraava, Metcalfen laki kehitettiin jo kaksisuuntaisia verkkoja varten ja sen mukaan verkon arvo kasvaa neliössä suhteessa käyttäjien määrään. Reedin laki puolestaan väittää, että verkon arvo kasvaa eksponentiaalisesti suhteessa käyttäjien määrään. Nämä lait on esitetty taulukossa 3. [30, 31]

Taulukko 3: Tietoliikenneverkon arvo käyttäjämäärillä N ja M. [30]

Laki	**Sarnoff**	**Metcalfe**	**Reed**
Verkon arvo, N käyttäjää	N	N^2	2^N
Yhdistetty arvo, N + M käyttäjää	$N + M$	$N^2 + M^2 + 2NM$	$2^N + 2^M$

Kuten kuvasta 3 nähdään, käyttäjämäärän N kasvaessa verkon arvo kasvaa Reedin lain mukaan huimasti verrattuna Sarnoffin ja Metcalfen lakeihin. Jos mietitään tarkemmin, mikä näistä kolmesta laista pätee tarkimmin nykyisiin verkkoihin ja yhteisöihin, Sarnoffin laki voidaan heti aluksi unohtaa, koska se kehitettiin lähinnä yksisuuntaisia radio- ja televisiolähetyksiä varten. Metcalfen ja Reedin laeissa puolestaan on yksi perustavanlaatuinen virhe: ne olettavat, että jokainen verkon yhteys on yhtä arvokas. Sekä Metcalfen että Reedin lakien mukaan kaksi verkkoa ovat myös arvokkaampia yhdessä kuin erikseen. Vieläpä niin, että verkkojen keskinäisellä koolla toistensa suhteen ei ole merkitystä. On selvää, että tämä ei todellisuudessa pidä paikkaansa, kun mietitään esim. Internetin naapurusperiaatteita. Näiden lakien mukaan isompi operaattori saisi naapuruudesta pienemmän operaattorin kanssa yhtä suuren hyödyn kuin pienempi operaattori. [30, 31]

Oikea vastaus on jossain Sarnoffin ja Metcalfen lakien välissä, ja Zipfin lain on todettu mallintavan reaalimailman ns. pitkän hännän *(long tail)* ilmiötä hämmästyttävän hyvin. Pitkän hännän tarkempi analyysi on tämän työn laajuuden ulkopuolella, mutta mainittakoon, että sillä tarkoitetaan ilmiötä, jossa järjestettäessä jokin suuri otos koon tai suosion mukaan k:nneksi järjestetyn alkion arvo vastaa arvoa $1/k$ ensimmmäisestä alkiosta. Zipfin lain mukaan verkon arvo on $Nlog(N)$, joka johdetaan siitä, että jokainen uusi verkon jäsen lisää sen arvoa arvolla $1/k$, jossa k on uuden jäsenen järjestysnumero. Tästä syntyvä logaritminen sarja $1+1/2+1/3+1/(N-1)$ lähestyy arvoa $log(N)$ ja verkon arvo saadaan kertomalla tämä käyttäjien määrällä N. Tätä arvoa voidaan käyttää arviona laskettaessa, kannattaako verkon rakentaminen, kun tiedetään yhden käyttäjän verkkoon liittämisen hinta. [30, 31]



Kuva 3: Tietoliikenneverkon arvo käyttäjämäärällä N. [30]

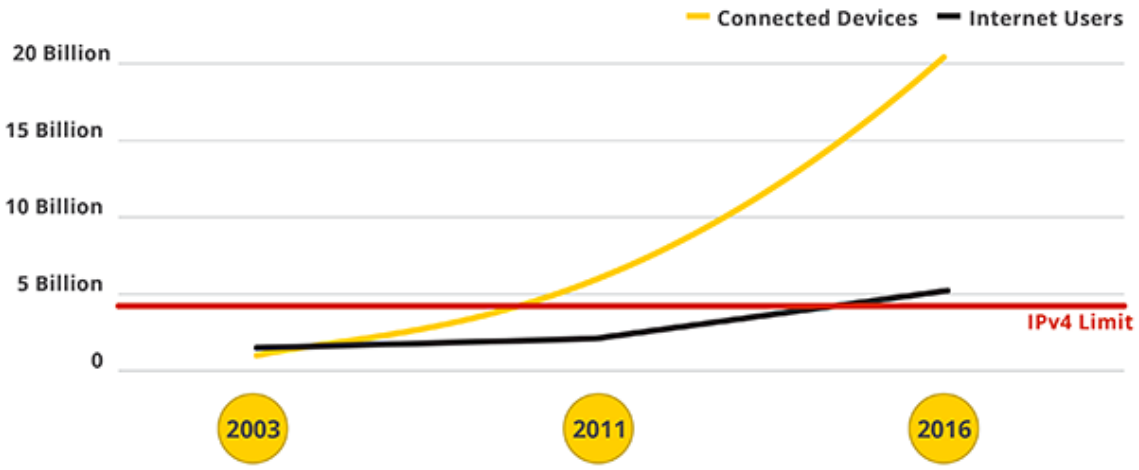
2.3 IPv4-protokollan ongelmat

Kuten taulukosta 4 nähdään, melko tarkkaan kolmasosa maailman ihmisistä käytti Internetiä vuoden 2011 lopussa. Käyttäjien määrä puolestaan oli tuolloin melko tarkkaan puolet IPv4-osoitteiden osoiteavaruudesta, joka sisältää 2^{32} eli 4294967296 osoitetta. Tämä tarkoittaa siis sitä, että jos jokaisella näistä käyttäjistä olisi kaksi yksilöivän IPv4-osoitteen tarvitsevaa päätelaitetta, esimerkiksi tietokone ja puhelin, olisi IPv4-osoiteavaruus täynnä.

Taulukko 4: Internetin käyttäjät maailmassa 31.12.2011. [32]

Maanosa	Väkiluku 2011	Käyttäjiä 31.12.2000	Käyttäjiä 31.12.2011	Penetraatio	Kasvu 2000-2011	Käyttäjiä
Afrikka	1037524058	4514400	139875242	13,5 %	2988,4 %	6,2 %
Aasia	3879740877	114304000	1016799076	26,2 %	789,6 %	44,8 %
Eurooppa	816426346	105096093	500723686	61,3 %	376,4 %	22,1 %
Lähi-Itä	216258843	3284800	77020995	35,6 %	2244,8 %	3,4 %
Pohj. Amerikka	347394870	108096800	273067546	78,6 %	152,6 %	12,0 %
Lat. Amerikka	597283165	18068919	235819740	39,5 %	1205,1 %	10,4 %
Oseania/Australia	35426995	7620480	23927457	67,5 %	214,0 %	1,1 %
Yhteensä	6930055154	360985492	2267233742	32,7 %	528,1 %	100,0 %

Kuva 4 havainnollistaa IPv4-osoitteiden, Internetin käyttäjien ja Internetiin yhteydessä olevien laitteiden määrän suhdetta toisiinsa vuosina 2003–2011 ja ennustaa tulevan trendin vuosille 2011–2016. Kuten nähdään, Internetiin yhteydessä olevien laitteiden määrä on jo ylittänyt IPv4-osoitteiden määrän ja myös Internetin käyttäjien määrän ennustetaan ylittävän sen joskus vuosina 2014–2015. IPv4-osoitteiden loppuminen onkin tärkein syy siihen, miksi tarvitaan uusi Internet-protokolla. [12]

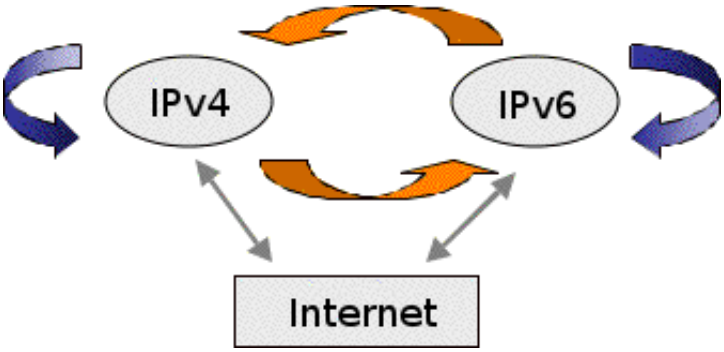


Kuva 4: IPv4-osoitteet ja Internetin käyttäjät sekä siihen liitetyt laitteet 2003–2016. [33]

Internet-reitittimien reititystaulujen kasvu on toinen syy siihen, miksi uutta Internet-protokollaa alettiin kehittää. Tätä kirjoitettaessa Capgeminin aktiivisen Internet-reitittimen reititystaulussa on 431715 riviä eli yksilöivää IPv4-reittiä. Kolmantena syynä IPv4:n elinkaaren päättymiseen voidaan nähdä sen riippuvaisuus yksityisistä verkko-osoitteista ja osoitteenmuunnoksista. Näitä tekniikoita käyttämällä rikotaan Internetin *end-to-end*-periaatetta eli sitä, että kaikki kommunikointi olisi puhtaasti lähettäjän ja vastaanottajan välistä. [12]

2.4 IPv6-käyttöönotto

IPv4:n ja IPv6:n välistä keskinäistä suhdetta voidaan tällä hetkellä kuvata kaksipuolisena markkinana, jossa alusta eli Internet-yhteisö yrittää maksimoida hyödyn IPv4:n ja IPv6:n yhteiselosta. Tämä on esitetty kuvassa 5.

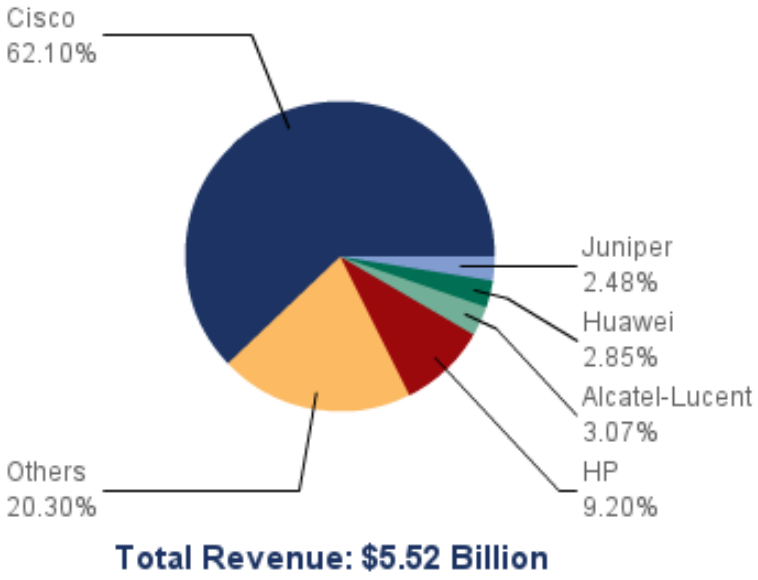


Kuva 5: IPv4- ja IPv6-protokollan kaksipuolinen markkina. [34]

Kuvassa 5 sinisillä nuolilla on kuvattu kummankin puolen sisäisiä verkostoilmiöitä (*network effect*) ja oransseilla nuolilla puolten välisiä verkostoilmiöitä. Verkostoilmiöllä tarkoitetaan sitä, että verkko on sen käyttäjälle sitä arvokkaampi, mitä

enemmän kyseisellä verkolla on käyttäjiä. Kaksipuoliset verkostoilmiöt tai markkinat puolestaan ovat sellaisia, joissa toisen ryhmän teot vaikuttavat toisen ryhmän tekoihin. IPv6:n tapauksesssa tämä tarkoittaa sitä, että mitä enemmän Internet-yhteisö vaatii IPv6-palveluita ja -tukea, sitä enemmän palvelun- ja sisällöntuottajat sekä laitevalmistajat niitä tarjoavat. Kääntäen, mitä enemmän palvelun- ja sisällöntuottajat sekä laitevalmistajat IPv6-palveluita ja tukea tarjoavat, sitä enemmän Internet-yhteisö IPv6-palveluita käyttää. [34]

Kuten missä tahansa muussakin teollisuudessa, myös tietoliikenneteollisuudessa johtavilla laitevalmistajilla on suuri rooli uuden teknologian kehittämisessä ja siinä, kuinka se otetaan markkinoilla vastaan. Cisco on markkinaosuudeltaan selvästi suurin L2/L3-Ethernet-kytkimien valmistaja, kuten kuvasta 6 nähdään. Se onkin tunnistanut neljä päätekijää, jotka vauhdittavat IPv6-protokollan käyttöönottoa: IPv4-osoitteet ja niiden loppuminen, valtioiden IT-strategia, käyttöjärjestelmätuki ja infrastruktuurin evoluutio. Seuraavaksi käydään nämä käsitteet läpi. [35]



Kuva 6: Liikevaihdoltaan suurimmat L2/L3-Ethernet-kytkinvalmistajat Q2/2012. [36]

- IPv4-osoitteet
  - Internetiin liitettyjen laitteiden määrän kasvu on aiheuttanut sen, että IPv4-osoitteet uhkaavat loppua.
  - Globalisaatio ja yritysten kasvu vaativat yhä enemmän IP-osoitteita.
  - Älypuhelimet ja muut mobiililaitteet ovat entistä useammin yhteydessä Internetiin.
  - IPv4-osoitteita jaettiin tehottomasti 1980- ja 1990-luvuilla.
  - Virtualisoinnin takia sama fyysinen laite saattaa vaatia usean IP-osoitteen.

  - Yritysostojen yhteydessä kahden eri yrityksen yksityiset IP-osoitealueet ovat yleensä päällekkäisiä, ja joudutaan turvautumaan osoitteenmuunnoksiin.

- Valtioiden IT-strategia
  - Valtioiden IT-strategia ohjaa vahvasti ko. maassa toimivien yritysten toimintaa. Yhdysvalloissa valtio asetti liittovaltion virastoiden IPv6-takarajaksi 30.6.2008, mistä johtuen useat virastoiden kanssa yhteistyössä toimivat urakoitsijat joutuivat myös päivittämään verkkonsa IPv6-yhteensopiviksi.

- Käyttöjärjestelmätuki
  - IPv6 on nykyään tuettuna kaikissa laajasti käytössä olevissa käyttöjärjestelmissä ja ne usein suosivat IPv6-protokollaa IPv4:n sijaan.

- Infrastruktuurin evoluutio
  - Yhä useampi teknologia, jonka ei alunperin ajateltu käyttävän IP-protokollaa on nyt tai tulevaisuudessa siitä riippuvainen. Esimerkkejä ovat sensori- ja älykkäät sähköverkot, kaapeliverkon laajakaistayhteydet ja jo mainitut mobiiliyhteydet.

2.4.1 Myytit

Tärkein yksittäinen syy IPv6-käyttöönottoon on sen valtava osoiteavaruus. IPv6-protokollaan liittyy paljon myyttejä ja harhaluuloja ja kaikki muut syyt IPv6:n paremmuudelle IPv4:ään verrattuna ovat yleensä puhdasta markkinointipuhetta. Kannattaakin suhtautua varauksella, jos IPv6:n väitetään olevan tehokkaampi tai turvallisempi protokolla kuin IPv4. Sen voisi ajatella olevan tehokkaampi protokolla kuin IPv4 yhtä pitkien IP-otsikoiden ja fragmentoinnin sekä otsikoiden tarkistussummien puuttumisen takia. Käytännössä kuitenkin myös IPv4-otsikot ovat aina yhtä pitkiä, koska optioita käytetään harvoin. Ciscon tekemät mittaukset osoittavatkin, että IPv4:llä ja IPv6:lla saavutetut siirtonopeudet ovat samaa tasoa, ja IPv4:llä saavutettiin joissain tilanteissa jopa IPv6-protokollaa suurempia siirtonopeuksia. [37] Lisäksi tulevaisuudessa siirtonopeuksien kasvaessa myös reitittimien prosessointiteho kasvaa ja em. asioiden merkitys vähenee. IPv6-protokollan käyttöönotto ei automaattisesti myöskään takaa pienempiä reititystauluja, jos reittien aggregointia ei tehdä asianmukaisella tavalla. Transitiovaiheessa reitittimillä on lisäksi yleensä omat, erilliset reititystaulunsa IPv4:lle ja IPv6:lle, ja yksi IPv6-reitti vie myös neljä kertaa IPv4-reittiä enemmän muistia neljä kertaa pidemmän osoitteen takia. [12, 38]

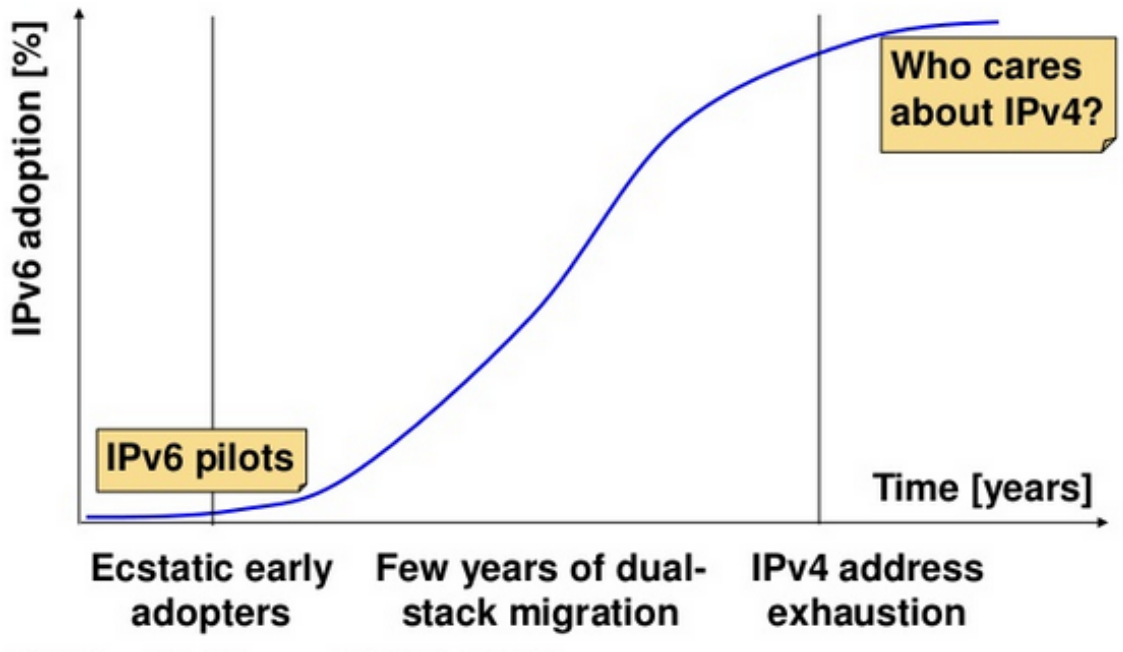
Tietoturva lienee yksi suurimmista IPv6:een liittyvistä harhaluuloista. IPv6-protokollan ajatellaan automaattisesti olevan yhtäältä turvallisempi protokolla kuin IPv4, koska aiemmin IPv6-solmun täytyi tukea IPsec-protokollaa [39]. Toisaalta IPv6-protokollan ajatellaan olevan turvattomampi protokolla kuin IPv4, koska siinä

ei lähtökohtaisesti käytetä osoitteenmuunnoksia. IPsec-vaatimus on kuitenkin poistunut, ja nyt IPv6-solmun pitäisi tukea IPsec-protokollaa [40]. Vaikka IPsec mahdollistaa kahden solmun tietoturvallisen kommunikoinnin, ei sitä voida kaikilla laitteilla ja kaikissa ympäristöissä käyttää ja vaikka käytettäisiinkin, ei se estä kaikkia IPv6-protokollaan kohdistuvia hyökkäyksiä. [38] Yleinen harhaluulo on myös se, että IPv6 ei mahdollistaisi toimipaikan kytkemistä Internetiin useamman eri palveluntarjoajan kautta (*multihoming*), eli että IPv6:ssa ei olisi PI-osoitteita (*Provider Independent*). Tämä piti paikkansa vuoteen 2007 asti, mutta nykyään myös IPv6:ssa on /32-/48-PI-osoiteavaruuksille varattu osoitealue 2001:678::/29, josta myös Capgeminin IPv6-osoitealue on allokoitu [41, 42].

IPv6-protokollan käyttöönotto ei suinkaan tarkoita sitä, että se korvaisi heti IPv4-protokollan kokonaan. Useat yritykset ja organisaatiot ovat investoineet paljon IPv4-protokollaan, ja se on todettu toimivaksi ratkaisuksi. Yritysten toimintatapaan kuuluu, että vanhan, toimivan ratkaisun vaihtamiseksi uuteen täytyy aina löytyä perusteltu, liiketoiminnallinen syy. Muutosvastarintaa esiintyy varmasti, mutta palveluntarjoajilla tämä syy on yksinkertaisesti se, että niiden asiakkaat vaativat ennen pitkää IPv6-tukea käyttämilleen palveluille ja sovelluksille. Yritysten omia, sisäisiä sovelluksia ja palveluita voidaan ja usein täytyykin käyttää IPv4:llä vielä pitkään, jos ne eivät ole IPv6-tuettuja. Kahden protokollapinon käyttäminen mahdollistaa tämän, ja IPv4 ja IPv6 tulevatkin elämään rinnakkaiseloa pitkään, ennen kuin IPv4:n käyttäminen voidaan joskus tulevaisuudessa kokonaan lopettaa. [38]

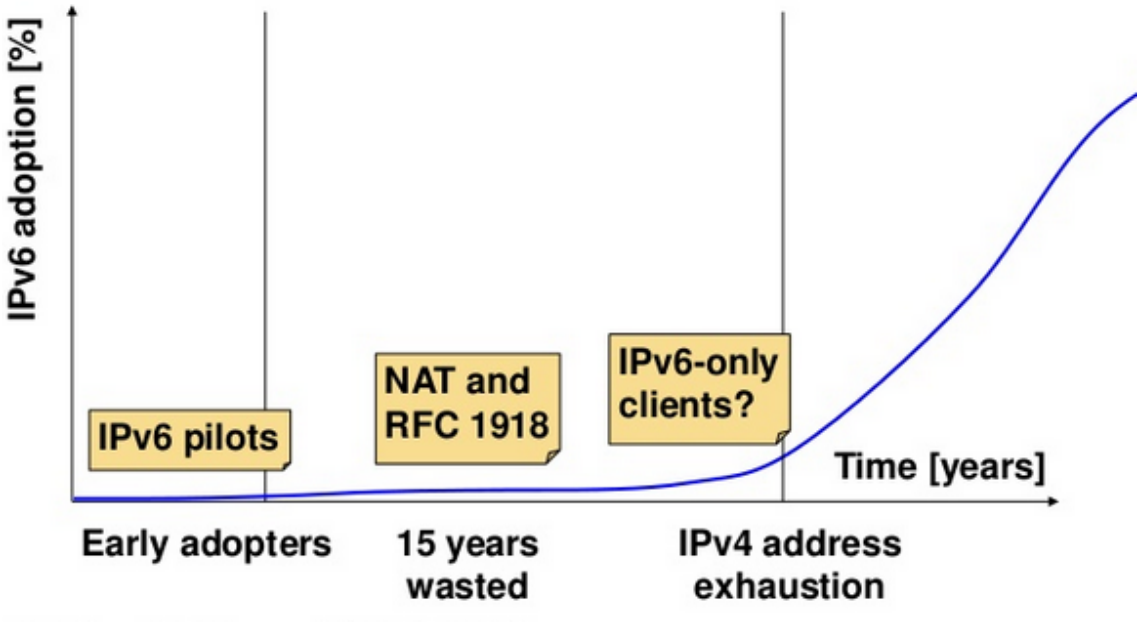
2.4.2 Ongelmat

Kuva 7 esittää perinteistä S-käyrää IPv6-käyttöönotosta ja IPv4-IPv6-migraatiosta.



Kuva 7: Kuinka IPv6-käyttöönoton piti mennä. [43]

Miksi nyt, kun ollaan siinä tilanteessa, että IPv4-osoitteet todellakin ovat loppuneet, tämä käyrä näyttää kuitenkin enemmän kuvan 8 kaltaiselta?

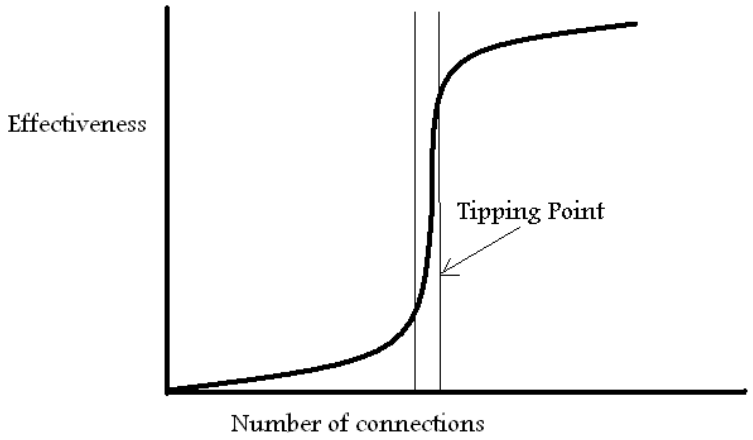


Kuva 8: Kuinka IPv6-käyttöönotto todellisuudessa meni. [43]

Tähän on monia syitä, kuten tekniikat, jotka kehitettiin lähinnä mahdollistamaan IPv4-protokollan elinkaaren pidentäminen 1990-luvulla. Näitä ovat esim. NAT (*Network Address Translation*), joka mahdollistaa usean käyttäjän saman julkisen IPv4-osoitteen käyttämisen, CIDR, joka kehitettiin ratkaisemaan IPv4-osoitteiden luokkahierarkiaa ja yksityiset IP-osoitealueet [14, 44, 45]. Varsinkin NAT hyväksyttiin laajalti ratkaisemaan IPv4-osoiteongelmaa, vaikka sen käyttöön liittyy paljon ongelmia ja kysymyksiä. Yksi näistä ongelmista on se, että NAT rikkoo Internetin *end-to-end*-periaatetta [46]. Tästä syystä liikennöinti NATin takana olevaan laitteeseen on usein vaikeaa tai jopa mahdotonta, ja sitä varten on kehitetty useita tekniikoita, kuten STUN (*Session Traversal Utilities for NAT*), TURN (*Traversal Using Relay NAT*) ja ICE (*Interactive Connectivity Establishment*) [47, 48, 49]. IETF on lisäksi julkaissut RFC-dokumentteja, jotka korostavat osoitteenmuunnosten käytöstä seuraavia ongelmia [50, 51]. Yleisesti kuvitellaan, että NAT lisää Internet-käyttäjän tietoturvaa. Vaikka tämä osaltaan pitääkin paikkansa, NAT:ia ei alunperin kehitetty tietoturvamekanismiksi eikä se täten ratkaise kaikkia tietoturvaongelmia. IPv6-protokolla kehitettiin siitä lähtökohdasta, että sitä käytettäessä osoitteenmunnoksia ei tarvita, ja LNP (*Local Network Protection*) pyrkii ratkaisemaan samat ongelmat kuin NAT ilman tarvetta osoitteenmuunnoksille [52]. [53]

IPv6:n suurin ongelma liittyy kuitenkin ehkä siihen, että päästäkseen markkinoille uuden teknologian täytyy olla joko markkinavetoinen (*market pull*) tai teknologiatyöntöinen (*technology push*). Se ei ole tähän asti ollut oikein kumpaakaan, ja siksi se ei ole vieläkään laajassa käytössä. Markkinan veto syntyy vasta siinä vaiheessa, kun teknologia saa taakseen kriittisen massan ja ylittää ns. notkahdus-

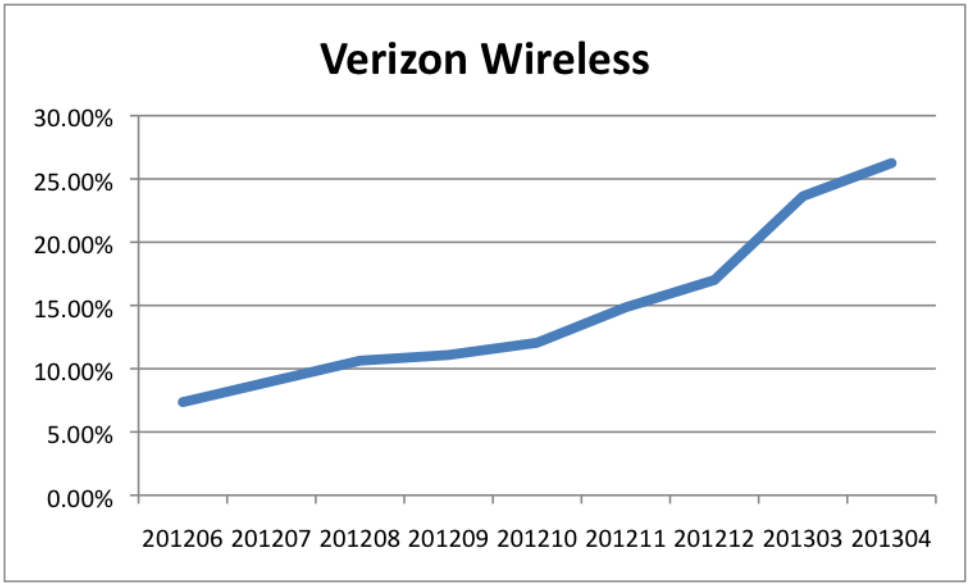
pisteen (*tipping point*), jonka jälkeen se lähtee yleensä eksponentiaaliseen kasvuun. Notkahduspisteen käsite on kuvattu kuvassa 9.



Kuva 9: Notkahduspiste (*tipping point*). [54]

Jokainen teknologia voidaan luokitella joko häiritseväksi (*disruptive*) tai kestäväksi (*sustaining*). Häiritsevä teknologia syntyy usein ns. vahingossa ja aivan eri markkinasegmenttiin kuin mihin se alunperin oli suunniteltu. Useimmat teknologiat ovat kestäviä, yrityksen ydinosaamista, joiden varaan se rakentaa toimintaansa. Kestäviä teknologioita kehittämällä voidaan saada aikaan muutosta ja kehitystä (*evolution*), kun taas häiritsevät teknologiat voivat aiheuttaa mullistuksia ja vallankumouksia (*revolution*). Häiritsevän teknologian ei kuitenkaan tarvitse ilmestyessään olla parempi kuin nykyiset teknologiat – se tekee jotakin eri tavalla, ja usein se tunnistetaankin häiritseväksi vasta jälkikäteen. Microsoftin Intelin x86-prosessoriarkkitehtuuriin pohjautuvat käyttöjärjestelmät häiritsivät Applen liiketoimintaa, ja vaikka Microsoftilla meni yli 10 vuotta kehittäessään käyttöjärjestelmänsä samalle tasolle Applen kanssa, niin sen jälkeen Microsoftista tuli markkinajohtaja. Vaikka on olemassa myös eriäviä mielipiteitä, Bill Gatesin 1996 kirjoittama artikkeli *Content Is King* pitää monen mielestä edelleen paikkaansa [55]. Internetin ja OSI-mallin näkökulmasta tämä tarkoittaa sitä, että sovelluskerros on kuningas ja sen alla olevat kerrokset mahdollistajia tai edesauttajia (*enabler*). Tarvitaan ns. killerisovelluksia, jotta IPv6 lähtisi kunnolla lentoon. Yhtäältä voidaankin todeta, että koska IPv6:ta hyödyntävää killerisovellusta ei vielä ole kehitetty tai sellaista ei ole tunnistettu, ei myöskään IPv6 ole vielä niin laajassa käytössä kuin se voisi olla ja todennäköisesti tulevaisuudessa on. Toisaalta, vaikka IPv6 olisi kuinka hyvä ja teknologisesti ylivertainen protokolla, ei se edesauttajan asemassa silti saa riittävästi *hypeä* taakseen levitäkseen markkinoilla yksinään, jos se ei tarjoa jotain selkeästi ylivertaista IPv4:ään verrattuna. Onkin kaksi eri vaihtoehtoa: joko IPv4-protokollan elinkaari päättyy ja markkinat ovat pakotettuja ottamaan IPv6:n käyttöön tai syntyy killerisovellus, joka vaatii IPv6:n toimiakseen. Tällä hetkellä ensimmäinen vaihtoehto näyttää todennäköisemmältä, koska miksi kukaan haluaisi kehittää sovellusta, jonka toiminnasta kaikilla alustoilla ja päätelaitteilla ei voi olla varma? [12, 56]

Mielenkiintoinen huomio on, että huhtikuussa 2013 tilaajamäärällä mitattuna Yhdysvaltojen suurimman mobiilioperaattori Verizon Wirelessin verkossa jo yli 25% liikenteestä oli IPv6-liikennettä, kuten kuvasta 10 nähdään [57]. Tämä johtuu pääosin Verizon Wirelessin LTE-käyttöönotosta (*Long Term Evolution*). Verizon Wirelessin LTE-verkko perustuu pitkälti IPv6:een, sen rungossa käytetään IPv6-osoitteita ja sen IMS APN (*Internet Multimedia Subsystem Access Point Name*) toimii puhtaasti IPv6-protokollalla. UE-laitteille (*User Equipment*) on kahden protokollapinon tuki. [58] Nähtäväksi jää, tuleeko LTE:stä IPv6-killerisovellus.



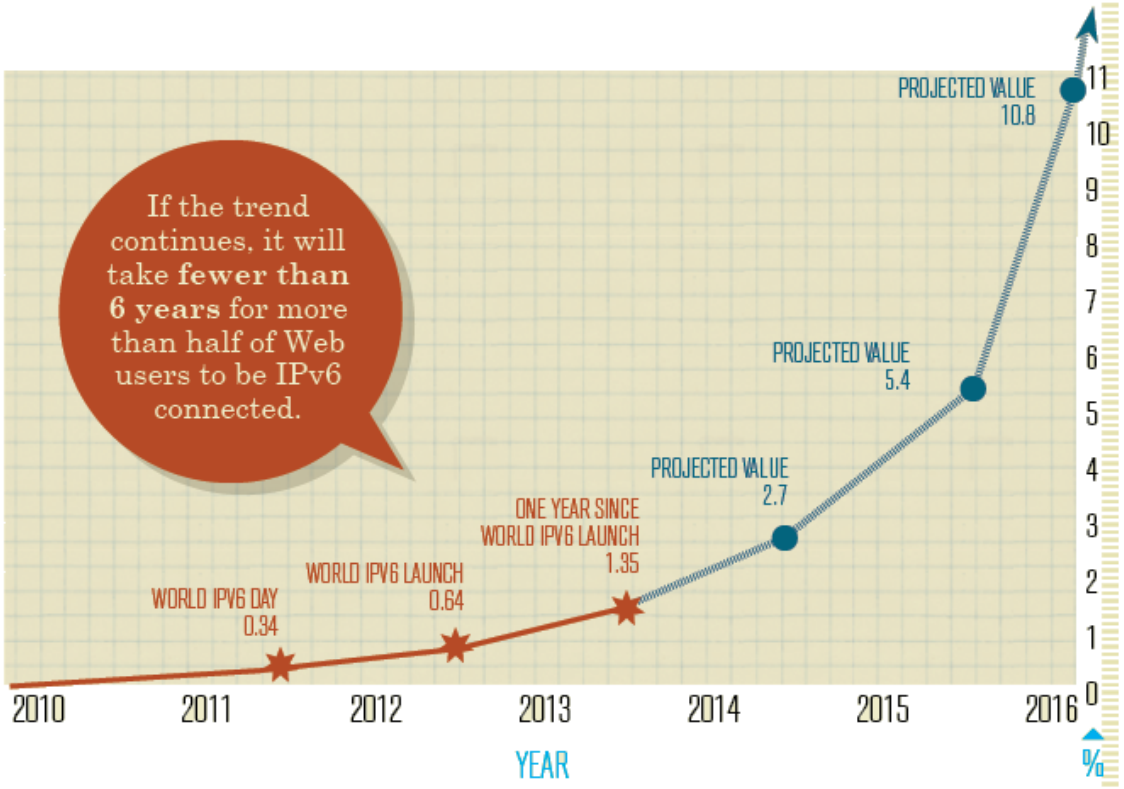
Kuva 10: IPv6-liikenne Verizon Wirelessin verkossa 06/2012 – 04/2013. [59]

Täytyy muistaa, että IPv4 on ollut menestystarina: se on mahdollistanut Internetin huikean suosion tähän päivään asti. Silti kysyttäessä tavalliselta Internetin käyttäjältä, mitkä hänen mielestään ovat tärkeimpiä Internetin sovelluksia ja ominaisuuksia, vastaus todennäköisesti on joko jokin tietty sovellus tai sivusto (esim. Google, Facebook tai Skype) tai se, että sisältö on aina ja helposti saatavilla. Tavallista Internetin käyttäjää ei kiinnosta, millä protokollalla tai sen versiolla sovelluksia tai sisältöä käytetään.

IPv6:n käyttöönotto on pitkälti myös peliteoreettinen dilemma. Peliteoriassa yhden tahon toiminta riippuu toisen tai toisten tahojen toiminnasta. Tavoitteena voi olla joko oman tai yhteisen hyödyn maksimoiminen. Koska IPv4 ja IPv6 eivät lähtökohtaisesti ole yhteensopivia keskenään, on todennäköisesti huono ratkaisu ottaa IPv6 käyttöön välittämättä muiden toiminnasta. Yritykset eivät ole ottaneet IPv6:ta käyttöön yhtäältä koska sitä tukevia laitteita ei ole ollut saatavilla, mutta toisaalta myös, koska Internet-palveluntarjoajat (ISP, *Internet Service Provider*) eivät ole tarjonneet IPv6-tukea. Toisinpäin ajateltuna Internet-palveluntarjoajat eivät ole tarjonneet IPv6-tukea, koska yritykset ja yhteisöt eivät ole sitä niiltä pyytäneet eivätkä halunneet. Sekä kaikkien tahojen oma että yhteinen hyöty maksimoituu, jos kaikki tekevät päätöksen ottaa IPv6-protokollan samanaikaisesti käyttöön.

Internet Society yhdessä suurimpien Internet-sivustojen ja -sisällöntarjoajien kuten Googlen, Facebookin, Yahoon, Akamain ja Limelight Networksin kanssa järjestivätkin *World IPv6* -päivän 8.6.2011. Yhteensä osallistujia oli 434 ja ne ottivat 24 tunnin ajaksi IPv6-protokollan käyttöön julkaisemalla IPv6-nimipalvelutietueet sivustoilleen [60]. Testin tuloksia ei tässä käydä tarkemmin läpi, mutta mainittakoon, että esim. Google ja Cisco eivät havainneet merkittäviä ongelmia testin aikana ja Facebook jätti kehittäjäsivustonsa IPv6-tuen päälle myös testin jälkeen [61, 62, 63]. Jotkin yritykset kuten Check Point jättivät IPv6-tuen päälle myös tuotantosivustolleen [64].

Vuosi *World IPv6* -päivän jälkeen 6.6.2012 järjestettiin *World IPv6 Launch* -päivä, jolloin IPv6 oli tarkoitus jättää pysyvästi päälle. Mukana olivat kaikki alkuperäisen *World IPv6* -päivän osallistujat ja monia muita. Vuosi *World IPv6 Launch* -päivän jälkeen IPv6-liikenne oli yli tuplaantunut Internetissä ja 2013 on kolmas vuosi peräkkäin, kun näin on tapahtunut. Jos nykyinen trendi jatkuu, yli puolet Internetin käyttäjistä käyttää IPv6-protokollaa alle kuuden vuoden päästä. [65]



Kuva 11: IPv6-liikenteen kasvu 2010–2013 ja ennuste 2013–2016. [66]

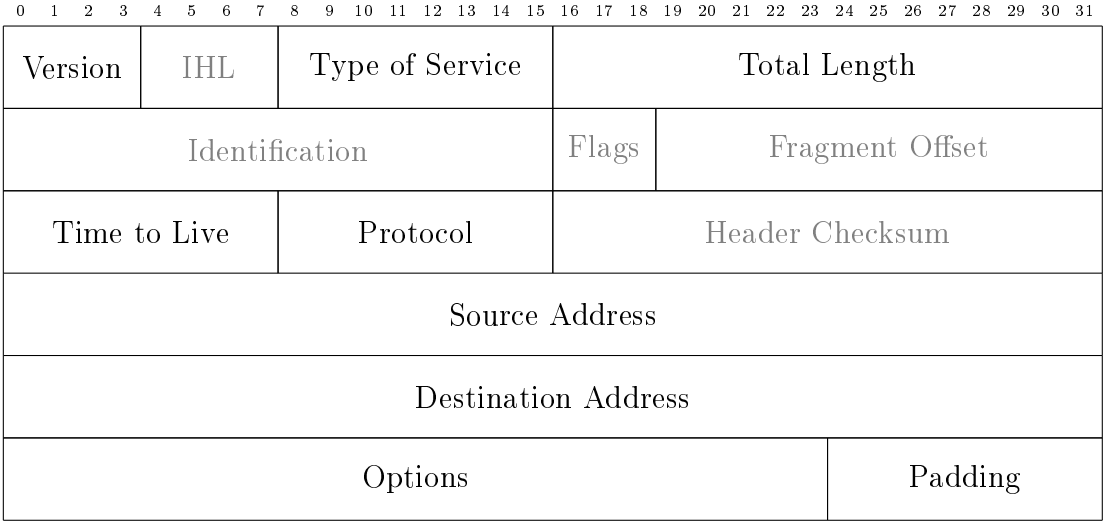
Tarve IPv6-protokollalle on tunnistettu jo pitkään, mutta IPv4-protokollan elinkaarta on pidennetty erilaisilla tässä luvussa kuvatuilla keinoilla. Viime vuosina IPv6-protokollaa on kuitenkin mm. *World IPv6* - ja *World IPv6 Launch* -päivien seurauksena otettu entistä enemmän käyttöön, ja jos kuvan 11 ennuste toteutuu, yli puolet Internetin käyttäjistä käyttää IPv6-protokollaa vuoteen 2019 mennessä. Seuraavassa luvussa esitellään itse IPv6-protokollaa ja sen tukiprotokollia.

3 Perus- ja tukiprotokollat

IPv6 suunniteltiin korvaamaan IPv4. Suurimmat erot IPv4:ään verrattuna liittyvät IPv6:n osoitteistukseen, otsikkoon, laajennuksiin ja optioihin sekä autentikointiin ja yksityisyyteen. Seuraavissa luvuissa käsitelläänkin näitä aiheita kirjallisuuden ja RFC-dokumenttien pohjalta. Lukijalla oletetaan olevan perustason tuntemus IPv4-protokollasta.

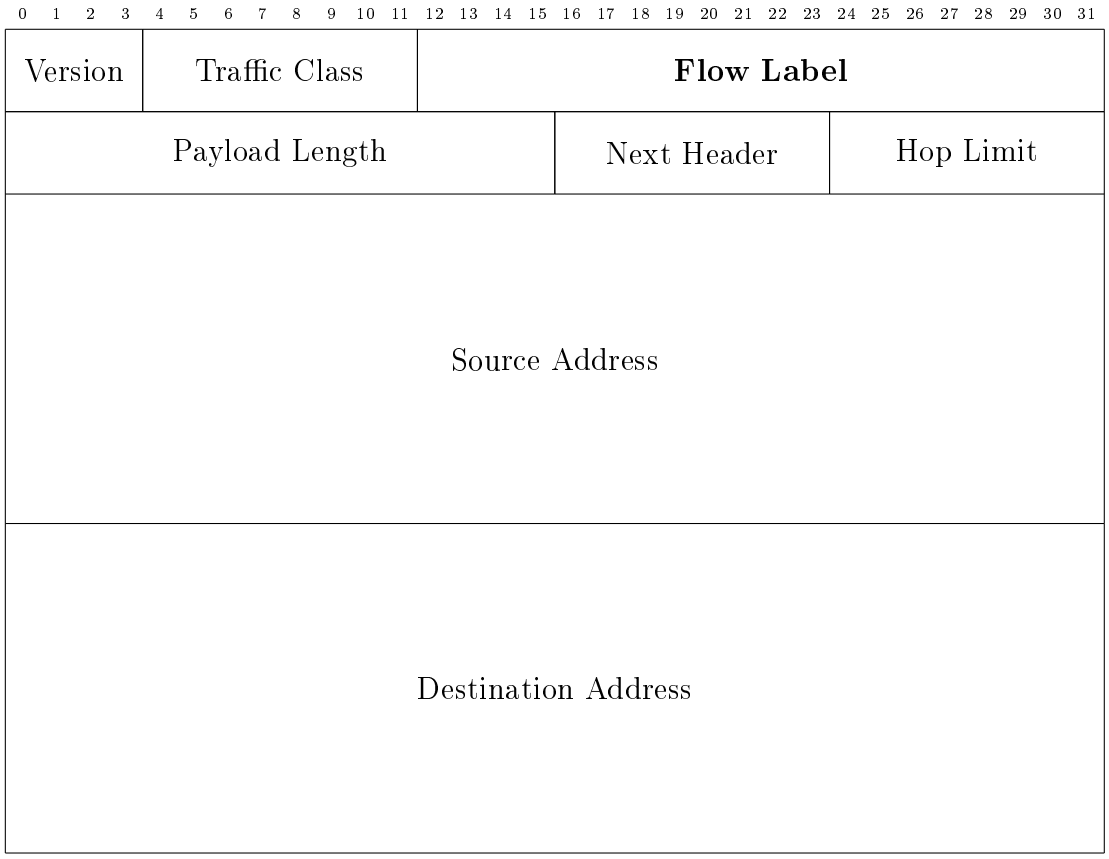
3.1 IPv4- vs. IPv6-otsikko

Jokaisen tietoliikenneprotokollan tärkein suunnittelun ja toiminnan lähtökohta on sen otsikon rakenne. Seuraavaksi kuvataankin IPv4- ja IPv6-otsikot ja esitellään niiden tärkeimmät erot. Selvyyden vuoksi kenttien nimet ovat englanniksi. Kuvassa 12 kuvattu IPv4-otsikko on 20-60-tavuinen ja se koostuu 13 tai 14 kentästä riippuen käytetäänkö optioita. Harmaalla on kuvattu kentät, joita ei ole IPv6-otsikossa. [1]



Kuva 12: IPv4-otsikko. [1]

IPv6-otsikko on 40-tavuinen ja se koostuu kahdeksasta kentästä. Sen lisäksi, että siinä on vähemmän kenttiä (8) kuin IPv4-otsikossa (13-14), siihen on tehty kolme merkittävää yksinkertaistusta IPv4-otsikkoon verrattuna: IPv6-otsikko on aina saman pituinen (40 tavua), otsikon tarkistussummasta luovuttiin ja reitittimet eivät enää voi fragmentoida IPv6-paketteja. Aina saman pituinen otsikko mahdollistaa sen, että IPv4:n IHL-kenttää (*Internet Header Length*) ei enää tarvita IPv6:ssa. IPv4:n otsikon tarkistussumma tarkoitti sitä, että reitittimen täytyi laskea se aina uudestaan, kun se esim. vähensi otsikon TTL-arvoa (*Time to Live*). Tarkistussumma lasketaan usein jo siirtokerroksella mm. Ethernet-protokollassa, joten IPv4:n tapauksessa se lasketaan usein kaksi kertaa ja IPv6:ssa siitä luovuttiin kokonaan. Fragmentoinnin puutteesta johtuen IPv4:n *identification*-, *flags*- ja *fragment offset* -kentät puuttuvat IPv6-otsikosta kokonaan. Kuvan 13 IPv6-otsikossa tummennettuna on kuvattu *flow label* -kenttä, jota ei ole IPv4-otsikossa. [7]



Kuva 13: IPv6-otsikko. [7]

Versiokenttä pidettiin otsikon ensimmäisenä ja saman pituisena, koska alkuperäinen ajatus oli, että paketin prosessointipäätös tehtäisiin tämän kentän avulla. Tästä ajatuksesta kuitenkin luovuttiin, ja IPv4- ja IPv6-paketit demultipleksoidaan jo OSI-mallin siirtokerroksella aina kun vain mahdollista. [67]

IPv4:n *total length* -kenttä on 16-bittinen, joten IPv4-paketin maksimikoko on 2^{16} tavua eli 64 kilotavua. IPv4-solmun täytyy pystyä vastaanottamaan 576-tavuinen tai pienempi paketti, ja jos paketin koko on yli 576 tavua, lähettäjän täytyy ensin varmistua siitä, että vastaanottaja kykenee vastaanottamaan paketin. [1] IPv6-otsikossa puolestaan on 16-bittinen *payload length* -kenttä, joten siinä hyötykuorman maksimipituus on 64 kilotavua. IPv6 vaatii jokaisen linkin tukevan vähintään 1280 tavun MTU-arvoa (*Maximum Transmission Unit*). Suositus on kuitenkin, että IPv6:ta käytettäessä MTU asetetaan vähintään 1500 tavuun. [7] IPv6 tukee myös jumbogrammeja, joita käyttämällä voidaan lähettää isompia kuin 64 kilotavun paketteja [68]. Reitittimien tekemästä IPv4-pakettien fragmentoinnista on osoittautunut olevan enemmä haittaa kuin hyötyä, ja IPv6-paketteja ei enää fragmentoida kuin mahdollisesti päätelaitteessa ennen niiden lähetystä [69]. Jos IPv4-paketin polulla on linkki, joka tukee vain pientä MTU-arvoa, paketti fragmentoidaan niin pieniin osiin, että jokainen pala alkuperäisestä paketista pääsee linkin yli. Jos yksikin näistä paloista häviää matkalla tai sen siirrossa kestää liian kauan, joudutaan koko paketti lähettämään uudestaan. IPv6 olettaa, että päätelaitteet oppivat polun tukeman

paketin maksimikoon *Path MTU Discovery* -mekanismin avulla. Siinä päätelaite pienentää lähettämiensä pakettien kokoa, kunnes se ei enää vastaanota ICMPv6-viestiä (*Internet Control Message Protocol*) *Destination Unreachable (Datagram Too Big)*, jolloin se tietää, että polku tukee sen viimeksi lähettämän paketin MTU-arvoa. [70]

IPv4:n TTL-kenttä on uudelleennimetty *hop limit* -kentäksi. Alunperin TTL-kentän oli tarkoitus esittää IPv4-paketin elinikää sekunneissa kuten esim. DNS-nimipalvelutietueessa, mutta käytännön syistä päädyttiin vain vähentämään kentän arvoa yhdellä joka IP-hypyllä. *Hop limit* onkin kuvaavampi nimi tälle toiminnallisuudelle. [1] IPv4:n *protocol*-kenttä on IPv6:ssa korvattu *next header* -kentällä. Myös sen toiminnallisuus on kuitenkin sama, eli kenttä kertoo minkätyyppinen otsikko seuraa IPv4- tai IPv6-otsikkoa. Usein se on joko 6 (TCP) tai 17 (UDP), mutta IPv6:n tapauksessa se voi olla myös toinen IPv6-laajennusotsikko, joista kerrotaan lisää seuraavassa luvussa. [7]

Vuotunniste- ja liikenneluokkakenttiä käytetään reaaliaikaliikenteen käsittelemiseen. Vuotunnisteella voidaan tunnistaa samaan liikennevuohon kuuluvat paketit, joilla on verkolle samanlaiset vaatimukset. Liikenneluokalla puolestaan voidaan priorisoida esim. VoIP-paketteja (*Voice over IP*) muiden pakettien yli. IPv6-lähde voi käyttää 20-bittistä vuotunnistekenttää erottelemaan liikennevoita toisistaan ja antamaan esim. reaaliaikaliikenteelle korkeampi prioriteetti. Liikenteen, jolla on sama lähde- ja kohdeosoite, lähde- ja kohdeportti ja joka käyttää samaa kuljetuskerroksen protokollaa on perinteisesti kategorisoitu kuuluvan samaan liikennevuohon. Liikennevuo ei kuitenkaan välttämättä ole rajattu vain yhteen kuljetuskerroksen protokollaan, ja fragmentoinnin ja salauksen takia jotkin näistä kentistä eivät välttämättä esiinny IPv6-paketissa tai niiden etsiminen voi olla tehotonta. IPv6-liikennevuo tunnistetaankin kolmen kentän, vuotunnisteen ja lähde- sekä kohdeosoitteen avulla. Nämä löytyvät IPv6-otsikossa aina määrätyiltä paikoilta, eikä niitä fragmentoida tai salata. QoS on ollut jatkuvan keskustelun kohteena jo IPv4:ään liittyen, ja myös IPv6:n vuotunnisteen käyttöön liittyy kysymyksiä. Kesäkuuhun 2011 mennessä kenttää ei ole juuri käytetty ja sen käyttöön on esitetty useita eri vaihtoehtoja: [7, 71, 72]

- pseudosatunnaiset vuotunnistearvot [73]
- QoS-vaatimukset vuotunnisteeseen sisällytettyjen parametrien mukaisesti
- pakettien kytkennän hallinta
- eriytettyjen palveluiden (*differentiated services*) QoS-arkkitehtuurin laajentaminen
- muut käytöt [74]

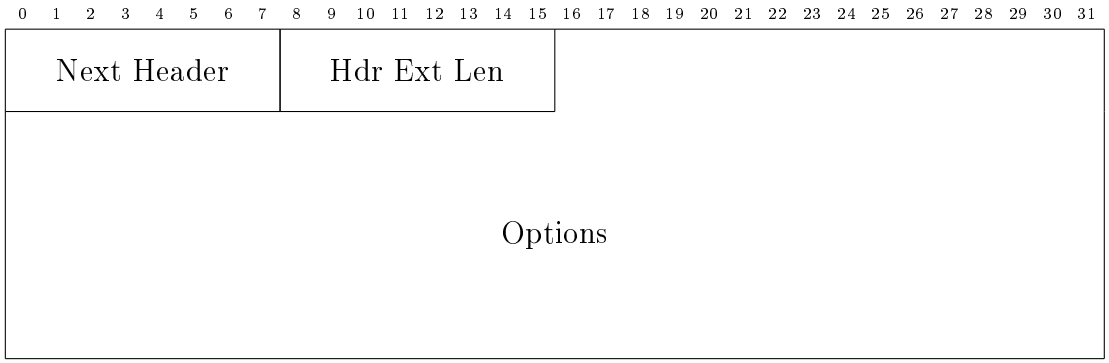
8-bittistä liikenneluokkakenttää taas voivat käyttää sekä lähde että liikennepolulla olevat reitittimet. Liikenneluokkakenttä vastaa IPv4:n ToS-kenttää (*Type of Service*). Senkään käyttöä ei ole määritelty tarkasti, mutta kuten IPv4, myös IPv6 tulkitsee kentän eriytettyjen palveluiden semantiikan mukaisesti, jos ollenkaan [75]. QoS ja CoS (*Class of Service*) ovat tämän työn laajuuden ulkopuolella. [76, 77]

3.2 Laajennusotsikot

IPv4-otsikon optiokenttä on poistettu kokonaan IPv6:sta. Tämä ei kuitenkaan tarkoita sitä, ettei IPv6-paketeille voisi antaa ns. erityiskohtelua, vaan IPv6:ssa optiokentän toiminnallisuutta vastaavat laajennusotsikot (*extension headers*). Niitä voi olla n kpl IPv6- ja ylemmän kerroksen otsikon välissä ja niitä on seitsemän erilaista: [7]

1. *Hop-by-hop*-optio-otsikko

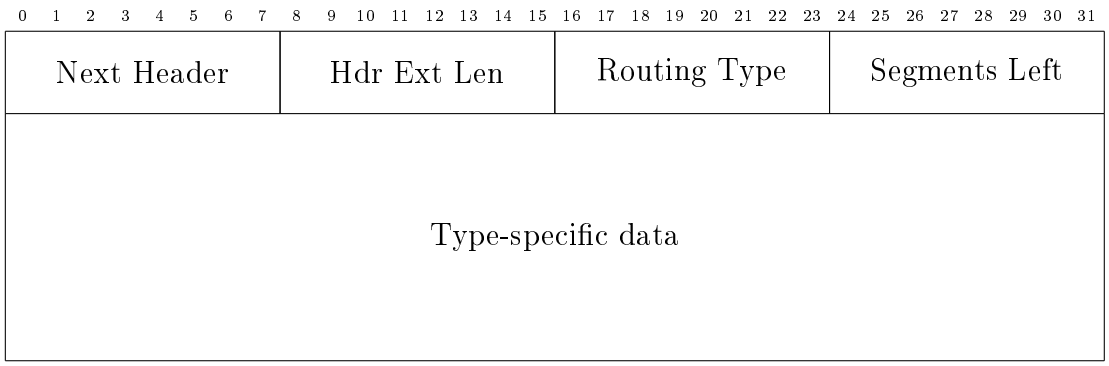
    - *Hop-by-hop*-optio-otsikolla välitetään tietoa, jonka jokaisen polun IPv6-solmun halutaan käsittelevän. Otsikko tunnistetaan IPv6-otsikon *next header* -kentän arvosta 0.



Kuva 14: Hop-by-hop-optio-otsikko. [7]

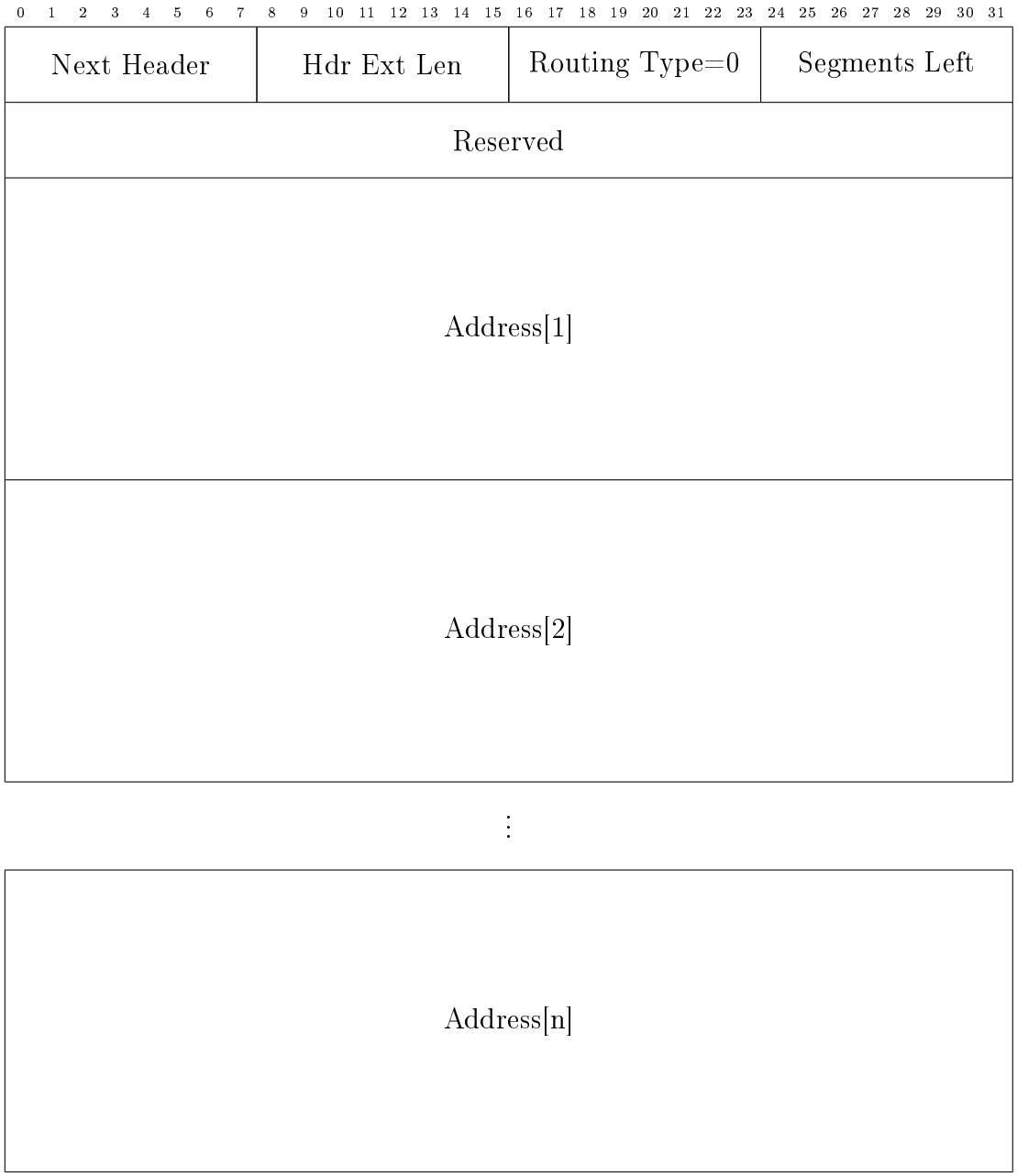
2. Reititysotsikko

    - Reititysotsikolla IPv6-lähde voi pakottaa paketin kulkemaan tiettyä reittiä. Otsikko tunnistetaan edeltävän otsikon *next header* -kentän arvosta 43.



Kuva 15: Reititysotsikko. [7]

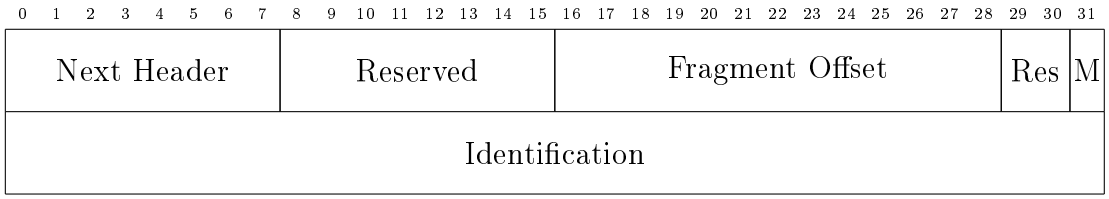
    - Tyypin 0 reititysotsikko oli ainoa alkuperäisessä IPv6-standardissa määritelty reititysotsikko, mutta sen käyttö on vanhentunut, ja päätelaitteiden ja reitittimien täytyy olla välittämättä tyypin 0 reititysotsikosta. [78]



Kuva 16: Tyypin 0 reititysotsikko. [7]

3. Fragmentointiotsikko

   - Fragmentointiotsikolla IPv6-lähde voi pilkkoa paketin pienempin osiin, jos polulla on linkki, jonka MTU-arvo on pienempi kuin paketin koko. Lähde saa polun MTU:n selville polun PMTUD-protokollan avulla, ja vain se voi fragmentoida IPv6-paketin, ei välissä olevat reitittimet [70]. Fragmentointiotsikko tunnistetaan edeltävän otsikon *next header* -kentän arvosta 44.



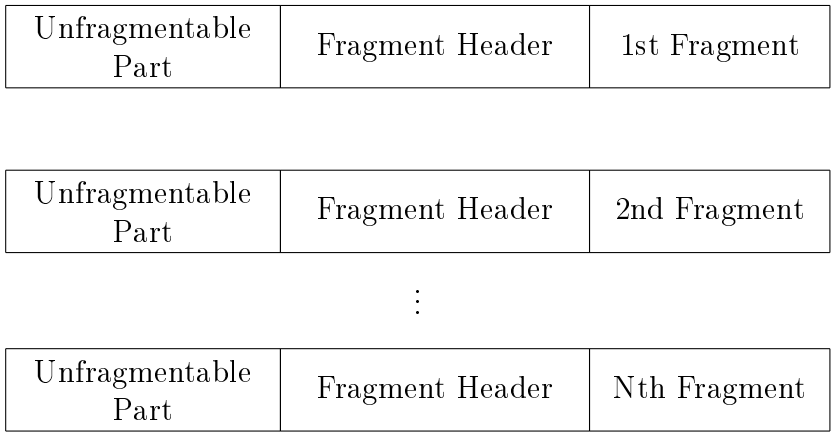
Kuva 17: Fragmentointiotsikko. [7]

- Lähde generoi tunnistekenttään (*Identification*) yksilöivän tunnisteen, jolla kohde erottaa osat toisistaan. Alkuperäisessä, pilkottavassa paketissa on kaksi osaa: pilkottavissa olematon (*unfragmentable*) ja oleva (*fragmentable*) osa:

Kuva 18: Pilkottavissa olematon ja oleva osa. [7]

- Pilkottavissa olemattomassa osassa on IPv6-otsikko ja laajennusotsikot, jotka kaikkien polun solmujen täytyy prosessoida. Pilkottavissa oleva osa koostuu laajennusotsikoista, jotka vain lopullisen kohteen täytyy prosessoida, ylemmän kerroksen otsikosta ja itse hyötykuormasta. Tämä osa pilkotaan osiin, joiden kaikkien viimeistä osaa lukuunottamatta täytyy olla pituudeltaan 64 bitin monikertoja. Nämä lähetetään erillisinä osapaketteina:

Kuva 19: Alkuperäinen paketti. [7]

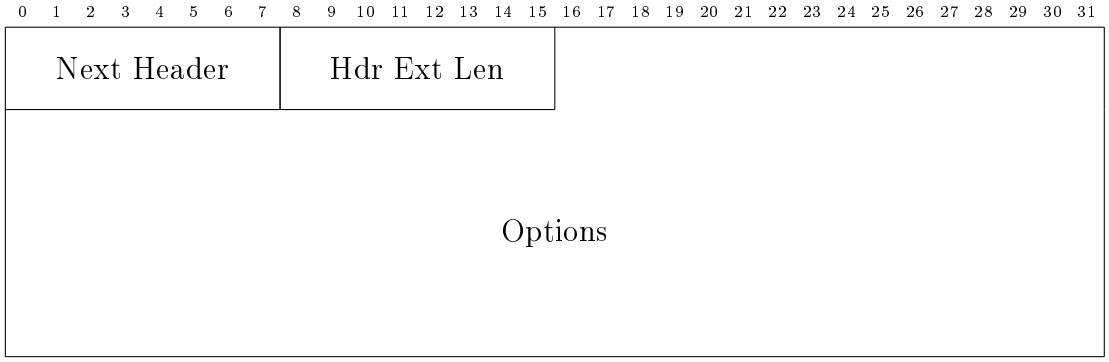


Kuva 20: Osapaketit. [7]

- Kohteessa osapaketit kootaan yhteen ja muodostetaan alkuperäinen paketti.

4. Kohdeoptio-otsikko (*Destination Options Header*)

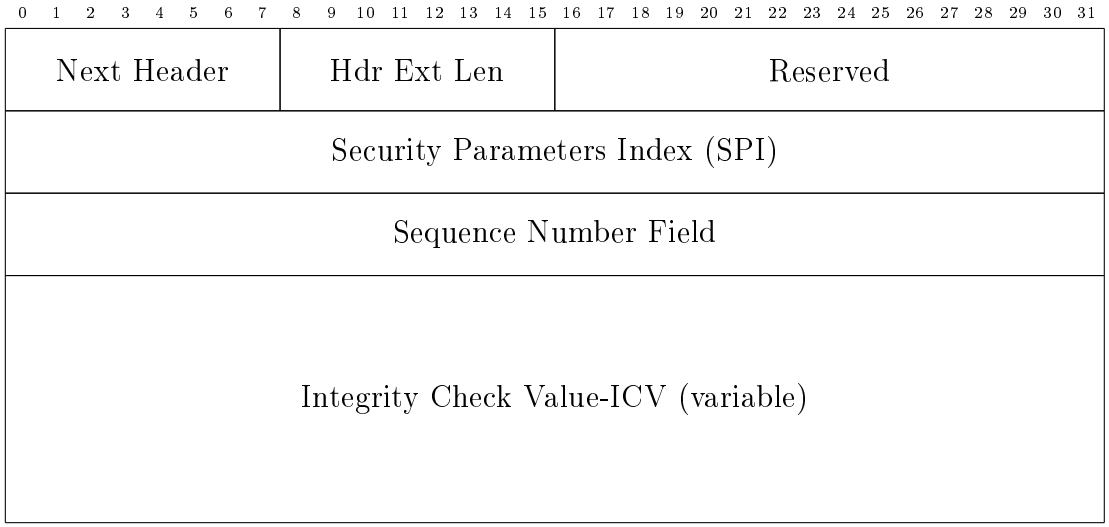
   - Kohdeoptio-otsikkoa käytetään silloin, kun halutaan välittää lisätietoa vain pelkälle kohteelle. Kohdeoptio-otsikko tunnistetaan edeltävän otsikon *next header* -kentän arvosta 60.



Kuva 21: Kohdeoptio-otsikko. [7]

5. Autentikointiotsikko

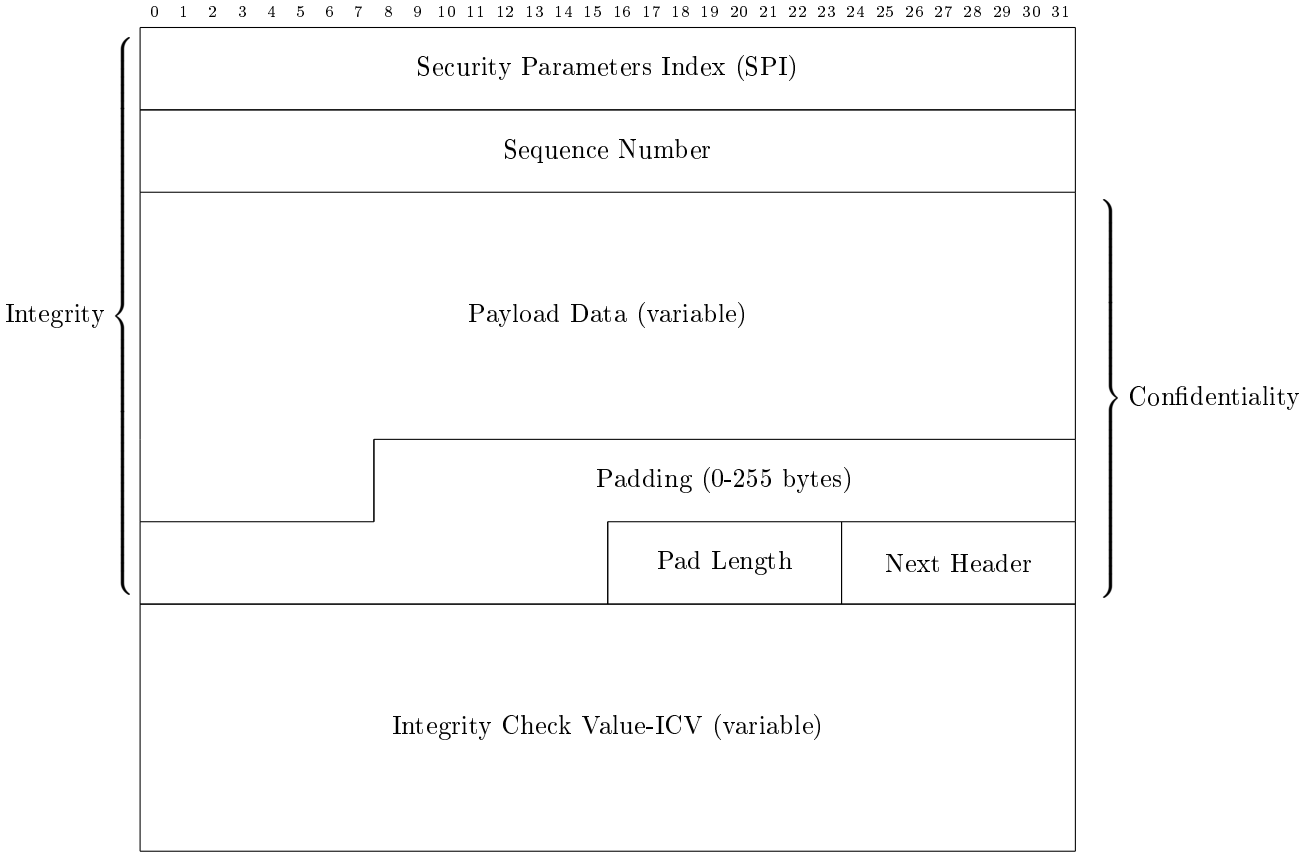
   - Autentikontiotsikko varmistaa tiedon eheyden (*integrity*) ja sen lähteen autentikoinnin sekä tarjoaa suojan uudelleenlähetyksiä (*replay*) vastaan. Sitä voidaan käyttää yksinään tai joko yhdessä tai sisäkkäin *Encapsulating Security Payload* -otsikon (ESP) kanssa. Autentikointiotsikko tunnistetaan edeltävän otsikon *next header* -kentän arvosta 51. [79]



Kuva 22: Autentikointiotsikko. [79]

6. ESP-otsikko (*Encapsulating Security Payload*)

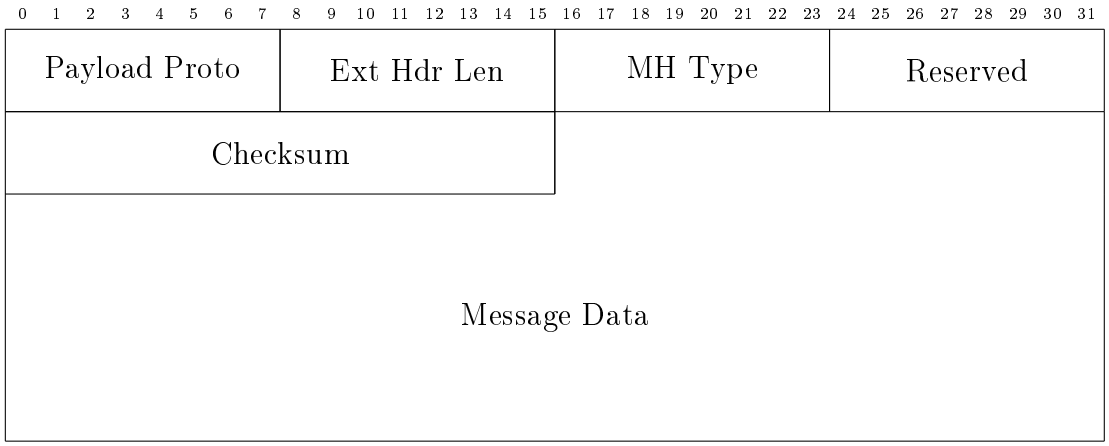
   - ESP-otsikko tarjoaa samat asiat kuin autentikointiotsikko, mutta lisäksi myös tiedon ja rajatun liikennevuon luottamuksellisuuden (*confidentiality*). Ero autentikointi- ja ESP-otsikoiden tarjoamassa eheydessä on sen laajuus – ESP-otsikko ei suojaa IPv6-otsikon kenttiä, jos sitä ei käytetä tunnelointitilassa (*tunnel mode*). Siinä ESP-otsikko sijoitetaan ennen IPv6-otsikkoa, kun taas siirtotilassa (*transport mode*) ESP-otsikko sijoitetaan IPv6-otsikon ja ylemmän kerroksen protokollan otsikon väliin. Samoin kun autentikointiotsikkoa, ESP-otsikkoa voidaan käyttää yksinään tai joko yhdessä tai sisäkkäin autentikointiotsikon kanssa. ESP-otsikko tunnistetaan edeltävän otsikon *next header* -kentän arvosta 50. [80]



Kuva 23: ESP-otsikko. [80]

7. Mobiliteettiotsikko

   - Mobiliteettitotsikko tarjoaa mobiili-IP-tuen IPv6-protokollalle, eli voidaan puhua mobiili-IPv6:sta. Mobiili-IPv6 on tämän työn laajuuden ulkopuolella, mutta mobiliteettiotsikon rakenne esitetään kuitenkin seuraavaksi. Mobiliteettiotsikko tunnistetaan edeltävän otsikon *next header* -kentän arvosta 135. [81]



Kuva 24: Mobiliteettiotsikko. [81]

- *Next header* -kentän arvolla 59 ilmaistaan, että seuraavaa otsikkoa ei ole olemassa. [7]

IPv6-paketissa voi siis olla enemmän kuin yksi laajennusotsikko. Ne voivat periaatteessa olla missä järjestyksessä tahansa, mutta suositeltava järjestys on seuraava: [7]

1. IPv6-otsikko
2. *Hop-by-Hop*-optio-otsikko
3. Kohdeoptio-otsikko (1)
4. Reititysotsikko
5. Fragmentointiotsikko
6. Autentikointiotsikko
7. ESP-otsikko
8. Kohdeoptio-otsikko (2)
9. Ylemmän kerroksen otsikko

Kohdeoptio-otsikko on listalla kahdesti. Jos halutaan määritellä optioita, jotka ensimmäisen ja muiden reititysotsikossa olevien kohteiden halutaan käsittelevän, laitetaan otsikko ennen reititysotsikkoa (1). Jos taas halutaan vain lopullisen kohteen käsittelevän optiot, laitetaan otsikko vasta viimeiseksi ennen ylemmän kerroksen protokollan otsikkoa (2). Ylempien kerrosten protokollien täytyy ottaa huomioon IPv6:ssa muuttuneet asiat siltä osin, kuin ne hyödyntävät verkkokerroksen tietoa toiminnassaan: [7]

- Jos ylemmän kerroksen protokolla laskee tarkistussummansa IPv6-otsikon lähde- ja kohdeosotteiden yli, täytyy laskenta laajentaa kattamaan 128-bittiset osoitteet IPv4:n 32-bittisten osoitteiden sijaan.

- IPv4:n TTL-kenttää ei ole IPv6:ssa, vaan se on nimetty uudelleen hyppyrajoituskentäksi. Käytännössä kaikki IPv4-toteutukset käsittelevät TTL-kenttää hyppyrajoituskentän tavoin, eli vähentävät TTL-arvoa yhdellä jokaisen hypyn jälkeen, joten tämä ei ole suuri ongelma.

- IPv6-otsikko on vähintään 20 tavua pidempi kuin IPv4-otsikko, mikä täytyy ottaa huomioon laskettaessa suurinta mahdollista hyötykuormaa ylemmän kerroksen protokollalle.

- Erityistä huomiota ylempien kerrosten protokollien toimintaan täytyy kiinnittää myös silloin, kun IPv6-paketti sisältää reititysotsikon tai -otsikoita.

3.3 Osoitteistus

On olemassa kolmentyyppisiä IPv6-osoitteita: *unicast*-, *anycast* ja *multicast*-osoitteita. IPv4:stä tuttuja *broadcast*-osoitteita ei IPv6:ssa ole. Kuten IPv4:ssä, myös IPv6:ssa osoitteet annetaan rajapinnoille (*interface*) eikä solmuille. Solmulla voi siten olla useita sen eri rajapinnoille määriteltyjä *unicast*-osoitteita, ja mitä vain näistä voidaan käyttää tunnistamaan kyseinen solmu. Jokaisella rajapinnalla täytyy olla vähintään yksi linkkilokaali (*link-local*) *unicast*-osoite, mutta sillä voi olla useita eri *unicast*-, *anycast*- tai *multicast*-osoitteita. [82]

IPv6-osoite on 128-bittinen osoite, joten yksilöiviä osoitteita on 2^{128} eli n. $3,40 * 10^{38}$ kpl. Tarkka luku on 340282366920938463463374607431768211456, eli n. 340 biljoonaa biljoonaa biljoonaa, n. 340 triljoonaa triljoonaa tai n. 340 sekstiljoonaa. Tarkistamatta kuinka monta nollaa kussakin lukuyksikössä olikaan, jokainen ymmärtää, että osoitteita on paljon. IPv6-osoite esitetään yleensä jollakin kolmella seuraavalla tavalla: [82]

- Heksadesimaalimuodossa x:x:x:x:x:x:x:x, jossa x on yhdestä neljään heksadesimaalilukua. Jokaisesta kahdeksasta 16-bittisestä kentästä voidaan etunollat jättää kirjoittamatta, mutta jokaisessa kentässä on oltava vähintään yksi heksadesimaaliluku.

- ::-notaatiota voidaan käyttää lyhentämään pitkiä, pelkkää nollaa sisältäviä osia IPv6-osoitteesta. Tätä notaatiota voidaan käyttää osoitteessa vain kerran.

- Jos IPv6-osoite sisältää IPv4-osoitteen, voidaan osoite kirjottaa myös muodossa x:x:x:x:x:x:d.d.d.d, jossa x:t ovat kuusi 16-bittistä eniten merkitsevää heksadesimaalilukua ja d:t neljä 8-bittistä vähiten merkitsevää desimaalilukua.

IPv6-osoitteita voidaan esittää myös CIDR-notaatiolla IPv4-osoitteiden tavoin [14]. IPv6-osoitteen tyypin määrää osoitteen eniten merkitsevät bitit taulukon 5 mukaisesti. *Anycast*-osoitteet otetaan *unicast*-osoiteavaruuksista. [82]

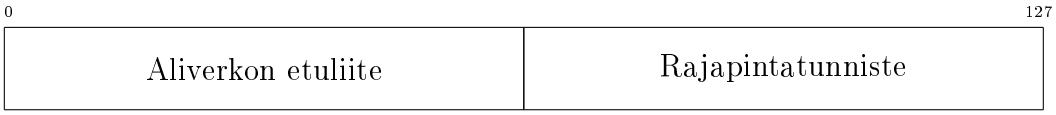
Taulukko 5: IPv6-osoitetyypit. [82]

Osoitetyyppi	**Binäärietuliite**	**IPv6-notaatio**
Määrittelemätön	00...0 (128 bittiä)	::/128
Loopback	00...1 (128 bittiä)	::1/128
Multicast	11111111	FF00::/8
Linkkilokaali unicast	1111111010	FE80::/10
Globaali unicast	kaikki muut	

Osoite 0:0:0:0:0:0:0:0 on määrittelemätön, ja se ilmaisee osoitteen puuttumista. 0:0:0:0:0:0:0:1 taas on loopback-osoite, jota voidaan käyttää haluttaessa lähettää IPv6-paketti itselle. [82]

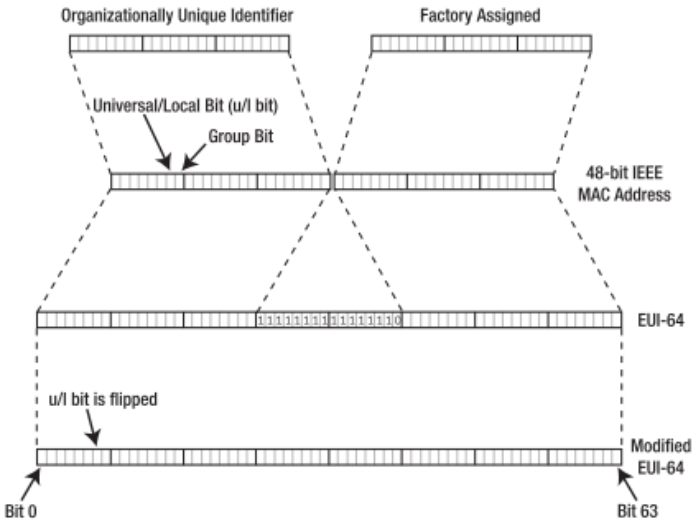
Unicast-osoitteet

Unicast-osoitteita on IPv6:ssa erityisosoitteiden lisäksi kolmenlaisia: globaaleja, paikkalokaaleja (*site-local*) ja linkkilokaaleja. Paikkalokaalit osoitteet ovat poistuneet käytöstä ja uusien IPv6-implementaatioiden tulee käsitellä paikkalokaaleja osoitteita globaaleina *unicast*-osoitteina [83]. Globaaleihin *unicast*-osoitteisiin kuuluvat mm. IPv6-osoitteet, joissa on sulautettu IPv4-osoite. *Unicast*-osoite on 128-bittinen osoite, jossa n bittiä kuuluu aliverkon etuliitteeseen (*subnet prefix*) ja jossa loput 128-n bittiä muodostavat rajapintatunnisteen (*interface identifier*):



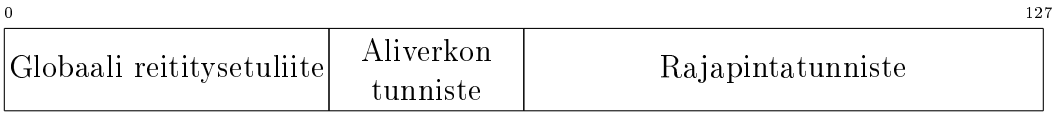
Kuva 25: Unicast-osoite. [82]

Rajapintatunnistetta käytetään tunnistamaan aliverkon rajapintoja ja saman aliverkon rajapintatunnisteiden täytyy olla yksilöiviä. Kaikilla muilla kuin binääriarvolla 000 alkavilla *unicast*-osoitteilla täytyy olla 64-bittinen rajapintatunniste, joka on muokatussa EUI-64-formaatissa. Tällainen tunniste muodostetaan siten, että ensin 48-bittisen MAC-osoitteen (*Media Access Identifier*) 24-bittisen OUI-tunnisteen (*Organizationally Unique Identifier*) ja lopun MAC-osoitteen 24 bitin väliin lisätään heksadesimaaliluku $FFFE$ eli binääriluku 1111111111111110. Näin saadaan IEEE (*Institute of Electrical and Electronics Engineers*) EUI-64-tunniste, jonka u-bitti ympäri kääntämällä saadaan lopulta muokattu EUI-64-tunniste. Sen u-bitti kertoo, onko rajapintatunniste globaali vai lokaali. Muokatussa EUI-64-tunnisteessa globaalin rajapintatunnisteen u-bitti on 1 ja lokaalin 0. Ryhmäbitti kertoo, onko MAC-osoite *unicast*- vai *multicast*-MAC-osoite. Kuva 26 havainnollistaa, kuinka MAC-osoitteesta saadaan muokattu EUI-64-tunniste. [84]



Kuva 26: MAC-osoite ja EUI-64- sekä muokattu EUI-64-tunniste. [84]

Globaalissa *unicast*-osoitteessa ensimmäiset n bittiä muodostavat globaalin reititysetuliitteen, seuraavat m bittiä aliverkon tunnisteen ja loput 128-n-m bittiä rajapintatunnisteen:



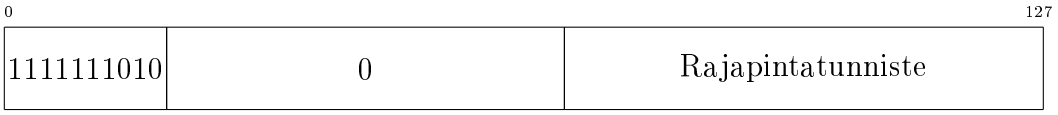
Kuva 27: Globaali unicast-osoite. [82]

Kaikilla muilla kuin binääriarvolla 000:lla alkavilla globaaleilla *unicast*-osoitteilla on 64-bittinen rajapintatunniste. 000:lla alkavilla globaaleilla *unicast*-osoitteilla ei ole tätä rajapintatunnisteen pituusrajoitusta. Esimerkkinä näistä ovat IPv6-osoitteet, joihin on sulautettu IPv4-osoite. Niitä on kahdentyyppisiä: IPv4-yhteensopivia ja *IPv4-mäpättyjä* IPv6-osoitteita. Niissä kummassakin IPv4-osoite sijoitetaan IPv6-osoitteen viimeiseen neljännekseen. IPv4-yhteensopivat IPv6-osoitteet kehitettiin IPv6-siirtymää varten, mutta nykyiset siirtymämekanismit eivät enää käytä niitä, joten niiden käyttö on vanhentunut. [82]

Kuva 28: IPv4-yhteensopiva IPv6-osoite. [82]

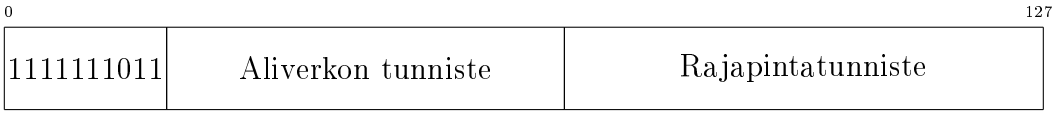
Kuva 29: IPv4-mäpätty IPv6-osoite. [82]

Linkkilokaalia *unicast*-osoitetta käytetään yksittäisellä linkillä esim. automaattiseen osoitteen konfigurointiin tai naapurin löytämiseen (*neighbor discovery*). Reitittimet eivät saa reitittää pakettia, jolla on linkkilokaali lähde- tai kohdeosoite. Vaikka linkkilokaaleille *unicast*-osoitteille on varattu osoiteavaruus FE80::/10, ovat ne käytännössä aina avaruudesta FE80::/64: [82]



Kuva 30: Linkkilokaali unicast-osoite. [82]

Paikkalokaalin *unicast*-osoitteen käyttö on vanhentunut, mutta koska olemassaolevat toteutukset voivat vielä käyttää sitä, esitellään se sen rakenne tässä.

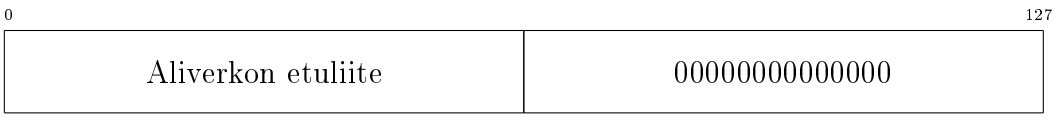


Kuva 31: Paikkalokaali unicast-osoite. [82]

Anycast-osoitteet

Kuten aiemmin kerrottiin, *anycast*-osoitteet allokoidaan *unicast*-osoiteavaruudesta. Siksi konfiguroitaessa rajapinnalle *anycast*-osoite täytyy sille kertoa kyseessä olevan *anycast*-osoite. Esimerkki *anycast*-osoitteen käytöstä voi olla Internet-palveluntarjoaja, joka konfiguroi PE-reitittimellensä (*Provider Edge*) *anycast*-osoitteen, jolloin reititysotsikkoa käyttämällä tämän palveluntarjoajan palveluita käyttävä asiakas voi valita käyttämänsä ISP:n tai yhden ISP:n tapauksessa varmistaa, että IPv6-paketti reitittyy aina reititysprotokollan mielestä lähimmälle PE-reitittimelle. [82]

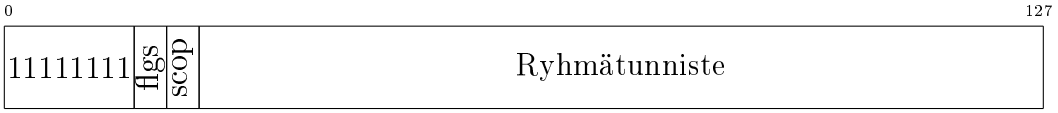
Aliverkkoreititin-*anycast*-osoite (*subnet-router*) on osoite, jossa vastaavan *unicast*-osoitteen rajapintatunniste on asetettu nollaksi. Reitittimen täytyy vastata aliverkkoreititinosoitteeseen jokaisesta aliverkosta, johon sillä on rajapinta. Esimerkki tällaisen osoitteen käytöstä ovat sovellukset, jotka eivät välitä siitä, mikä reititin pyyntöön vastaa. [82]



Kuva 32: Aliverkkoreititin-anycast-osoite. [82]

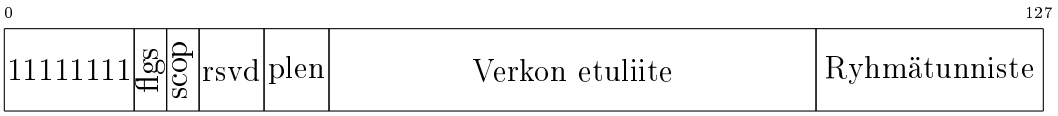
Multicast-osoitteet

Multicast-osoite tunnistetaan sen 11111111-alusta (FF):



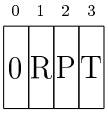
Kuva 33: Multicast-osoite. [82]

Unicast-etuliiteinformaation sisältävä *multicast*-osoite taas näyttää seuraavalta:



Kuva 34: Unicast-etuliiteinformaation sisältävä multicast-osoite. [85]

4-bittisen *flags*-kentän bittien merkitys on seuraava:



Kuva 35: Multicast-osoitteen flags-kenttä. [82]

- ensimmäinen bitti on aina 0
- T-bitti kertoo, onko kyseessä pysyvästi vai tilapäisesti allokoitu *multicast*-osoite:
  - T=0: IANA:n staattisesti allokoima (*well-known*) osoite
  - T=1: tilapäinen, dynaamisesti allokoitu osoite
- P-bitti kertoo, onko *multicast*-osoite allokoitu verkon etuliitteen mukaan
  - P=0: osoitetta ei ole allokoitu verkon etuliitteen mukaan
  - P=1: osoite on allokoitu verkon etuliitteen mukaan
    - Tällöin T=1
- R-bitti kertoo, onko kohtaamispisteen (*Rendezvous Point*) osoite sisällytetty *multicast*-osoitteeseen: [86]
  - R=0: RP:n osoitetta ei ole sisällytetty multicast-osoitteeseen
    - Tällöin osoitteen muoto on yllä esitetty
  - R=1: RP:n osoite on sisällytty multicast-osoitteeseen

∗ Tällöin sekä P=1 että T=1 eli syntyy etuliite FF70::/12. Tällöin 8-bittisen *reserved*-kentän neljä viimeistä bittiä sisältävät RP:n rajapintatunnisteen (RIID) eli osoitteen muoto on seuraava:

Kuva 36: Kohtaamispisteen rajapintatunnisteen sisältävä multicast-osoite. [86]

Scope-kenttää käytetään rajoittamaan *multicast*-ryhmän laajuutta. Kentän mahdolliset arvot on listattu taulukossa 6. *Unicast*-etuliitepohjaisen *multicast*-osoitteen laajuus ei saa ylittää *multicast*-osoitteeseen sisällytetyn *unicast*-etuliitteen laajuutta. [85]

Taulukko 6: Multicast-osoitteen scope-kentän mahdolliset arvot. [82]

0	varattu
1	rajapintalokaali laajuus
2	linkkilokaali laajuus
3	varattu
4	ylläpitäjälokaali laajuus
5	paikkalokaali laajuus
6	(määrittelemätön)
7	(määrittelemätön)
8	organisaatiolokaali laajuus
9	(määrittelemätön)
A	(määrittelemätön)
B	(määrittelemätön)
C	(määrittelemätön)
D	(määrittelemätön)
E	globaali laajuus
F	varattu

4- tai 8-bittisen *reserved*-kentän täytyy olla 0. *plen*-kenttä kertoo, kuinka monta bittiä verkon etuliite -kentästä on käytetty verkon tunnistamiseen, kun P=1. Verkon etuliite -kenttä ilmaisee *multicast*-osoitteen omistavan *unicast*-aliverkon etuliitteen. Jos P=1, kentässä on *multicast*-osoitteen omistavan tai allokoivan tahon *unicast*-verkkoetuliite. [85] Ryhmätunnisteen asettaminen on määritelty omassa RFC-dokumentissaan ja se on tämän työn laajuuden ulkopuolella [87]. Seuraavat *multicast*-osoitteet ovat ennalta määriteltyjä: [82]

Varatut multicast-osoitteet	FF00:0:0:0:0:0:0:0
	FF01:0:0:0:0:0:0:0
	FF02:0:0:0:0:0:0:0
	FF03:0:0:0:0:0:0:0
	FF04:0:0:0:0:0:0:0
	FF05:0:0:0:0:0:0:0
	FF06:0:0:0:0:0:0:0
	FF07:0:0:0:0:0:0:0
	FF08:0:0:0:0:0:0:0
	FF09:0:0:0:0:0:0:0
	FF0A:0:0:0:0:0:0:0
	FF0B:0:0:0:0:0:0:0
	FF0C:0:0:0:0:0:0:0
	FF0D:0:0:0:0:0:0:0
	FF0E:0:0:0:0:0:0:0
	FF0F:0:0:0:0:0:0:0

Kaikki solmut -osoitteet (*all nodes*) sisältävät kaikki IPv6-solmut tietyllä laajuudella: [82]

Laajuus	**Osoite**
Rajapintalokaali	FF01:0:0:0:0:0:0:1
Linkkilokaali	FF02:0:0:0:0:0:0:1

Kaikki reitittimet -osoitteet (*all routers*) sisältävät kaikki IPv6-reitittimet tietyllä laajuudella: [82]

Laajuus	**Osoite**
Rajapintalokaali	FF01:0:0:0:0:0:0:2
Linkkilokaali	FF02:0:0:0:0:0:0:2
Paikkalokaali	FF05:0:0:0:0:0:0:2

Tavoiteltu solmu -osoite (*solicited node*) on muotoa FF02:0:0:0:0:1:FFXX:XXXX, joka muodostetaan ottamalla *unicast*- tai *anycast*-osoitteen 24 vähiten merkitsevää bittiä ja liittämällä ne etuliitteeseen FF02:0:0:0:0:1:FF00::/104. Osoite voi olla siis väliltä FF02:0:0:0:0:1:FF00:0000 – FF02:0:0:0:0:1:FFFF:FFFF. [82]

Lopuksi mainittakoon, että IPv6-solmun täytyy tunnistaa itsensä seuraavilla osoitteilla: [39]

- jokaisen rajapinnan linkkilokaalilla osoitteella
- jokaisen rajapinnan jokaisella muulla *unicast*- tai *anycast*-osoitteella
- *loopback*-osoitteella
- kaikki solmut -*multicast*-osoitteella
- tavoiteltu solmu -*multicast*-osoitteella

- kaikkien muiden ryhmien *multicast*-osoitteilla, joihin IPv6-solmu kuuluu

Reitittimen vaaditaan tunnistavan itsensä ylläolevien lisäksi seuraavilla osoitteilla:

- aliverkkoreititin-*anycast*-osoitteella kaikissa niissä rajapinnoissa, joissa se reitittää liikennettä
- kaikilla muilla *anycast*-osoitteilla, jotka reitittimelle on konfiguroitu
- kaikki reitittimet -*multicast*-osoitteella

3.4 Reititys

IPv6-reititys vastaa hyvin pitkälti IPv4-reititystä. Staattisissa IPv6-reiteissä *next-hop* -osoitteen pitäisi olla linkkilokaali osoite [88]. Linkkilokaalien osoitteiden käytölle on monia perusteluita, mutta usein niitä ei käytetä, koska ollaan huolissaan niiden dynaamisesta luonteesta (verkkorajapinnan osoite vaihtuu sen MAC-osoitteen vaihtuessa), koska linkkilokaali osoite ei vastaa pingiin toisesta aliverkosta ja koska niitä ei yksinkertaisesti olla totuttu käyttämään. Erityisesti niitä ei voida käyttää, jos *next-hop* -osoite on useamman kuin yhden hypyn päässä tai silloin, jos staattinen reitti jaetaan eteenpäin (*redistribute*) jollekin dynaamiselle reititysprotokollalle. Tällöin on pakko käyttää GUA- (*Global Unicast Address*) tai ULA-osoitetta (*Unique Local Address*). ULA-osoitteet ovat IPv6:n vastine IPv4:n yksityisille IP-osoitealueille ja ne otetaan osoiteavaruudesta FC00::/7. [44, 89, 90, 91]

Yleisesti suositaan myös ns. kohtalon jakoa (*fate-sharing*) hallinta- ja välityskerrosten (*control & data plane*) välillä eli sitä, että IPv4-reittejä mainostetaan IPv4-siirtotien yli ja IPv6-reittejä IPv6-siirtotien yli. Tällöin ei pääse syntymään tilannetta, jossa esim. IPv4-siirtotietä käyttävä IPv6-reititysprotokolla ei huomaa ongelmaa IPv6-siirtotiessä vaan reitittää IPv6-liikennettä mustaan aukkoon. [90]

IPv6-liikennettä reitite tään samoilla dynaamisilla reititysprotokollilla kuin IPv4-liikennettäkin, mutta niistä on luonnollisesti tehty uudet versiot tai niihin on lisätty ominaisuuksia tukemaan IPv6-reititystä. Seuraavaksi esitellään lyhyesti uudet versiot RIP- (*Routing Information Protocol*), OSPF- (*Open Shortest Path First*), IS-IS- (*Intermediate System to Intermediate System*) ja BGP-protokollista (*Border Gateway Protocol*) ja listataan niiden tärkeimmät erot IPv4-versioihin verrattuna.

RIPng

RIPng perustuu RIPv2:een ja jakaa samat perusominaisuudet sen kanssa: [92, 93]

- Bellman-Ford-etäisyysvektorialgoritmi
- reittipäivitykset 30s välein
- 180s vanhenemisaika reitille
- kiinteät metriikat

- 15 hypyn maksimi (*counting to infinity* -ongelma)
- *split horizon* ja *poison reverse*
- reittitunnisteet (*route tags*)

RIPng käyttää linkkilokaalia lähde- ja *next-hop*-osoitetta sekä linkkilaajuista *multicast*-kohdeosoitetta ja hyppyrajoitusta, mikä suojaa protokollaa väärennetyiltä reitittimiltä ja protokollaviesteiltä. Tärkeimmät muutokset ja uudet ominaisuudet RIPv2:een verrattuna ovat seuraavat: [92, 93]

- *next-hop* -osoite on prefiksiä mainostavan reitittimen rajapinnan linkkilokaali IPv6-osoite
- reittipäivityksen lähdeosoite on sen lähettävän reitittimen rajapinnan linkkilokaali IPv6-osoite
- reittipäivityksen kohdeosoite on FF02::9, eli kaikki RIP-reitittimet -*multicast*-osoite
- reittipäivityksen hyppyrajoitus on 255
  - Näin reittipäivityksen vastaanottava reititin voi varmistua siitä, että se on peräisin samalla linkillä olevalta reitittimeltä. Jos reititin vastaanottaa reittipäivityksen, jonka hyppyrajoituskentän arvo on pienempi kuin 255, se hylkää sen.
- käytettävä UDP-portti on 521 eikä 520 kuten RIPv1:ssä ja RIPv2:ssa
- RIPng:n versionumero on 1
- IPv6-reititystaulu on täysin erillinen IPv4-reititystaulusta
- RIPng luottaa IPsec-autentikointiin, kaikki RIP-spesifinen autentikointi on poistettu

OSPFv3

OSPFv2 on IPv4-reititysprotokolla, eli se ei osaa reitittää mitään muuta protokollaa. OSPFv3 taas on protokollariippumaton reititysprotokolla, joten sitä voidaan käyttää IPv6-reititykseen. Tämä on mahdollista siksi, että itse OSPF-paketit ja tärkeimmät LSA-viestit (*Link State Advertisement*) eivät enää sisällä osoitteita vaan välittävät ainoastaan topologiatietoa reititintunnisteeseen perustuen. OSPFv3 perustuu OSPFv2:een ja sillä on sen kanssa samat perusominaisuudet: [93, 94]

- Dijkstran algoritmi
- tulvitus, *flooding*
- *designated router* -vaali

- OSPF-alueet
  - tyngät (*stub*) ja *NSSA*-alueet (*Not-So-Stubby Area*)
- *demand circuit* -tuki

OSPFv3:n tärkeimmät muutokset ja uudet ominaisuudet OSPFv2:een verrattuna ovat seuraavat: [93, 94]

- Osoitteistussemantiikka on poistettu OSPF-paketeista ja reititin- sekä verkko-LSA-viesteistä
- Uudet LSA:t *intra-area-prefix-LSA* ja *link-LSA* välittävät IPv6-osoiteinformaatiota
  - IPv6-osoitteet LSA-viesteissä ilmaistaan prefiksillä ja sen pituudella.
- Reititintunniste, aluetunniste ja *LSA Link State* -tunniste ovat yhä 32-bittisiä, joten niille ei voi antaa IPv6-osoitetta
- LSA-viestien tulvituslaajuus (*flooding scope*) on eksplisiittisesti ilmaistu LSA-viestissä
- *Next-hop* -osoite on prefiksiä mainostavan reitittimen rajapinnan linkkilokaali IPv6-osoite
- OSPFv3-protokollaa ajetaan linkki- eikä aliverkkokohtaisesti kuten OSPFv2:ssa
  - Yksittäisellä linkillä voidaan ajaa useita OSPFv3-instansseja.
- IPv6-otsikon *next header* -kentän arvo 89 kertoo kyseessä olevan OSPFv3-paketti
  - OSPFv3-paketin lähdeosoite on sen lähettävän reittimen rajapinnan linkkilokaali IPv6-osoite.
  - Kohdeosoite on joko FF02::5 (*AllSPFRouters*) tai FF02::6 (*AllDRouters*).
- OSPFv3-pakettien hyppyrajoitus on 1
  - 255 olisi ollut parempi valinta, sillä silloin kohdereititin voisi varmistua siitä, että paketti on tullut samalla linkillä olevalta reittimeltä. Vastaanottaessaan OSPFv3-paketin hyppyrajoituskentän arvolla 1 reititin ei voi olla varma siitä, että pakettia ei ole reititetty 254 kertaa ennen tätä tai siitä, että kentän arvoa ei olisi vähennetty tähän arvoon juuri ennen kuin se saapui ko. linkille.
- OSPFv3-pakettien versionumero on 3
- Autentikointi on poistettu ja OSPFv3 luottaa IPsec-autentikointiin

OSPFv3 käyttää reititintunnisteita linkkitilatietokannan rakentamiseen, joten jos samassa verkossa käytetään sekä IPv4- ja IPv6-protokollaa, niiden topologiat eivät voi erota toisistaan jos halutaan reitittää sekä IPv4- että IPv6-protokollaa samalla OSPFv3-instanssilla. [93, 94]

IS-IS

IS-IS on alunperin OSI:n kehittämä IGP-reititysprotokolla (*Interior Gateway Protocol*), joka kehitettiin alusta pitäen protokollariippumattomaksi reititysprotokollaksi [95]. Myöhemmin siihen lisättiin IPv4-tuki ja IPv6-tukea varten siihen jouduttiin lisäämään vain kaksi uutta TLV-kenttää (*type-length-value*) ja uusi IPv6-protokollatunniste: [93, 96, 97]

- Uusi IPv6-saavutettavuus-TLV (tyyppi 236), joka on IPv6-reittimainostus
- Uusi IPv6-rajapintaosoite-TLV (tyyppi 232), joka on *Hello*-paketeille reitittimen linkkilokaali ja LSP-paketeille (*Link State Protocol Data Unit*) reitittimen globaali IPv6-osoite
- Uusi IPv6-NLPID (*Network Layer Protocol ID*) (142) kertoo, että reititin tukee IPv6-reititystä IS-IS-protokollalla

BGP

BGP-protokolla on EGP-protokolla (*Exterior Gateway Protocol*), eli sitä käytetään reititykseen eri hallinnollisten alueiden (AS, *Autonomous System*) välillä. Kun BGP:tä käytetään eri alueiden välillä, kutsutaan sitä eBGP:ksi (*External BGP*) ja kun taas saman alueen sisällä, kutsutaan sitä iBGP:ksi (*Internal BGP*). BGP-protokollaan määriteltiin uusi osoiteperhe (*Address Family*) IPv6:ta varten ja MBGP-protokolla (*Multiprotocol Extensions for BGP-4*) tukee IPv6-reititystä. Siihen lisättiin myös kaksi uutta NLRI-attribuuttia (*Network Layer Reachability Information*), jotta sillä voidaan reitittää IPv4:n lisäksi myös muita protokollia. Näissä attribuuteissa käytetään ao. informaatiota reitittämään ei-IPv4-liikennettä: [93, 98, 99]

- *Address Family* -informaatiota
  - Kertoo, minkä protokollan liikennettä reititetään.
- *Next hop* -informaatiota
  - *Next-hop* -osoite NLRI-kentän sisältämälle reitille.
- *Network Layer Reachability* -informaatiota
  - Mainostetun reitin etuliite ja sen pituus.

Kahden protokollapinon toteutuksissa, joissa käytetään sekä IPv4- että IPv6-protokollaa, voidaan niin haluttaessa samalla MBGP-instanssilla reitittää molempia protokollia. Näin tehtäessä BGP-sessioiden määrä puolittuu, mutta toisaalta jos esim. BGP-sessio on muodostettu IPv4-siirtotien yli ja IPv6-siirtotie ei jostain syystä toimi, ollaan tilanteessa jossa reititys toimii, mutta IPv6-liikennettä ei voida verkossa välittää. Suositus onkin käyttää ns. kohtalon jakoa (*fate sharing*), eli mainostaa IPv4-reittejä IPv4-siirtotien yli ja IPv6-reittejä IPv6-siirtotien yli. [90]

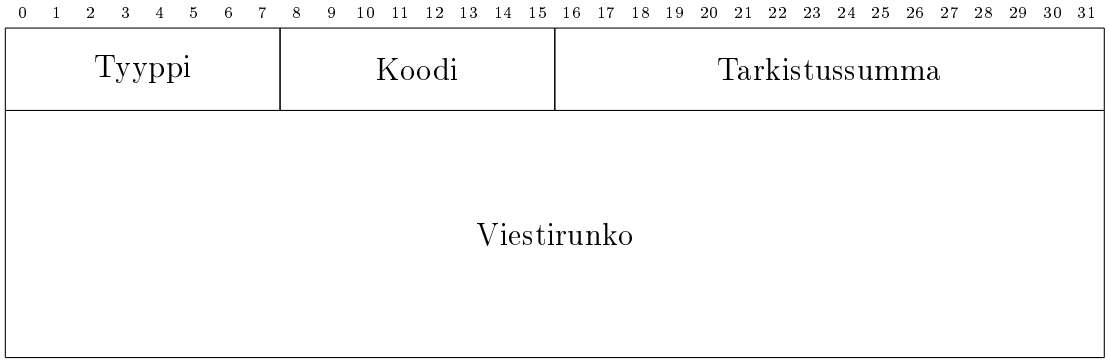
Toinen BGP:hen liittyvä kysymys on se, käytetäänkö BGP-sessiossa linkkilokaaleja vai globaaleja osoitteita. Linkkilokaalien osoitteiden käyttö tarjoaa tietoturvaa, mutta niitä ei voida käyttää monen hypyn BGP-sessiossa. MBGP-standardissa vaaditaankin, että *next-hop* -kenttä sisältää globaalin osoitteen. [99]

3.5 Tukiprotokollat

Kuten IPv4, myös IPv6 hyödyntää useita ns. tukiprotokollia toiminnassaan. Seuraavaksi esitellään näistä tärkeimmät, ICMPv6 (*Internet Control Message Protocol*) ja NDP (*Neighbor Discovery Protocol*) sekä kerrotaan, miten IPv6-osoitteita voidaan joko muodostaa automaattisesti SLAAC-mekanismilla (*Slateless Address Autoconfiguration*) tai jakaa DHCPv6-protokollalla (*Dynamic Host Configuration Protocol*).

3.5.1 ICMPv6

ICMPv6-protokollan toiminta perustuu ICMPv4-protokollan toiminnallisuuteen tietyin muutoksin. Kuten ICMPv4-protokollaa, myös ICMPv6:ta käytetään ilmaisemaan IPv6-pakettien prosessoinnissa tapahtuneita virheitä ja toteuttamaan muuta verkkokerroksen toiminnallisuutta kuten diagnostiikkaa. IPv6-otsikon *next header* -kentän arvo ICMPv6-otsikolle on 58. ICMPv6-otsikko on kuvattu kuvassa 37. [100, 101]



Kuva 37: ICMPv6-otsikko. [101]

Tyyppikenttä kertoo viestin tyypin, koodikenttä riippuu tyyppikentän arvosta ja tarkistussummaa käytetään ICMPv6-viestin ja osittain myös IPv6-otsikon korruptoitumisen havaitsemiseen. ICMPv6-viestit on jaettu kahteen luokkaan: virhe- (*error*) ja tiedotuksellisiin (*informational*) viesteihin. Virheviesteissä ICMPv6-otsikon tyyppikentän eniten merkitsevä bitti on 0 ja tiedotuksellisissa viestissä 1 – virheviestien tyypit ovat väliltä 0–127 ja tiedotuksellisien viestien tyypit väliltä 128–255, kuten taulukoista 7 ja 8 nähdään. [101]

Taulukko 7: ICMPv6-protokollan virheviestit. [101]

Tyyppi	**Viesti**
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem

Taulukko 8: ICMPv6-protokollan tiedotukselliset viestit. [101]

Tyyppi	**Viesti**
128	Echo Request
129	Echo Reply

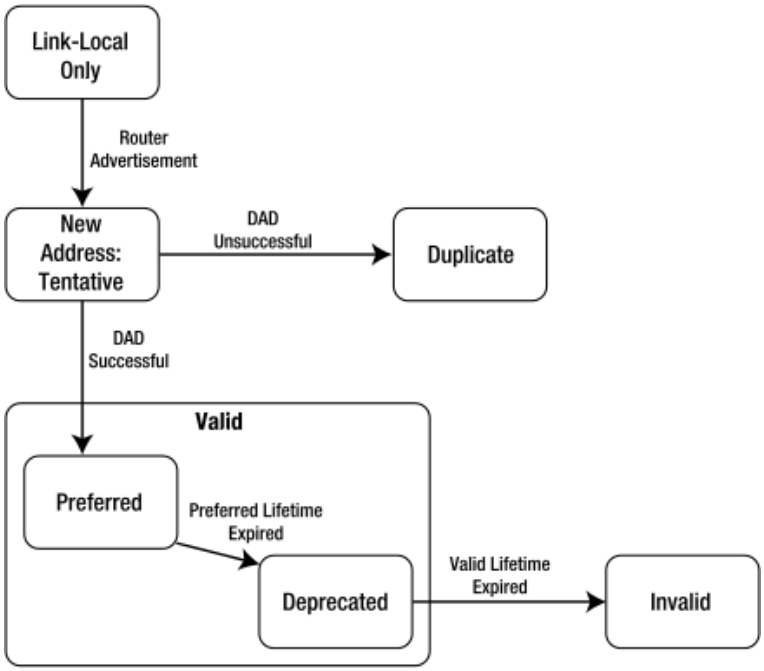
3.5.2 NDP

IPv6-solmut käyttävät naapurinetsintäprotokollaa määrittääkseen niiden IPv6-naapurien linkkikerroksen osoitteet, löytääkseen omia paketteja eteenpäin välittävät naapurireitittimet ja ylläpitääkseen saavutettavuustietoa naapureista. ND-protokolla määrittelee ratkaisumekanismin seuraaviin kysymyksiin, jotka liittyvät kahden samalla linkillä olevan IPv6-solmun keskinäiseen kommunikointiin: [88]

- *Router Discovery (RD)*
  - Kuinka päätelaitteet löytävät samalla linkillä olevat reitittimet?
- *Prefix Discovery*
  - Kuinka päätelaitteet saavat selville, mitkä prefiksit ovat saavutettavissa ilman reititystä?
- *Parameter Discovery*
  - Kuinka solmu oppii L2- tai L3-parametrit (esim. MTU, *hop limit*), jotka se asettaa lähettämiinsä paketteihin?
- *Address Autoconfiguration*
  - Kuinka solmu voi konfiguroida rajapinnalleen IPv6-osoitteen automaattisesti ja tilattomasti?
- *Address Resolution*
  - Kuinka solmut saavat selville naapurin linkkikerroksen osoitteen?
- *Next-hop Determination*
  - Mihin *next-hop* -osoitteeseen tietylle vastaanottajalle lähetettävä liikenne tulee lähettää?
- *Neighbor Unreachability Detection (NUD)*
  - Kuinka solmut päättelevät, että naapuri ei ole enää tavoitettavissa?
- *Duplicate Address Detection (DAD)*

– Kuinka solmu päättelee, onko sen IPv6-osoite yksilöllinen?

DAD-algoritmi suoritetaan kaikille *unicast*-osoitteille riippumatta siitä, ovatko ne manuaalisesti konfiguroituja tai esim. SLAAC:lla tai DHCPv6:lla saatuja. [102] Algoritmin toiminta on esitetty kuvassa 38.



Kuva 38: DAD-algoritmi. [84]

DAD-algoritmi toimii seuraavasti: [84, 102]

1. Aluksi päätelaitteella on ainoastaan linkkilokaali osoite (jolle on myös suoritettu DAD-algoritmi)
2. Vastaanottaessaan RA-viestin, jossa on *autonomous address configuration* -lippu päällä, se muodostaa IPv6-osoitteen perustuen rajapintatunnisteeseen ja RA-viestissä saatuun prefiksiin
3. Päätelaite merkitsee osoitteen tilaksi *tentative*, liittyy sen tavoiteltu solmu - ja kaikki solmut -*multicast*-ryhmiin ja lähettää NS-viestin. Mikäli –
   – päätelaite vastaanottaa NA-viestin toiselta päätelaitteelta, jolla on sama osoite
     → se merkitsee osoitteen tilaksi *duplicate*
   – päätelaite vastaanottaa NS-viestin toiselta päätelaitteelta, joka on juuri suorittamassa DAD-algoritmia
     → se merkitsee osoitteen tilaksi *duplicate*
   – päätelaite ei saa vastausta NS-viestiin
     → se merkitsee osoitteen tilaksi *preferred* ja ottaa osoitteen käyttöönsä, kunnes se muuttuu tilaan *deprecated* tai *invalid*

- Redirect
  - Kuinka reititin kertoo päätelaitteelle paremmasta *first-hop*-osoitteesta saavuttaakseen tietyn kohteen?

Näiden kysymysten ratkaisemiseksi ND-protokolla määrittelee viisi ICMPv6-viestiä: [88]

- *Router Solicitation (RS)*
  - Kun IPv6-rajapinta otetaan käyttöön, päätelaitteet voivat lähettää RS-viestin, joka laukaisee reitittimet lähettämään RA-viestin heti.
- *Router Advertisement (RA)*
  - Reitittimet lähettävät RA-viestejä joko jaksottaisesti tai vastauksena RS-viestiin. RA-viestit sisältävät prefiksejä, joita käytetään esim. määrittelemään, onko prefiksi samalla linkillä, osoitteiden konfigurointiin tai ehdottamaan hyppyrajoitus tietylle prefiksille.
- *Neighbor Solicitation (NS)*
  - Solmu lähettää NS-viestin saadakseen selville naapurin linkkikerroksen osoitteen tai varmistaakseen, että naapuri on vielä saavutettavissa välimuistiin tallentamallaan linkkikerroksen osoitteella. NS-viestejä käytetään myös DAD-algoritmissa.
- *Neighbor Advertisement (NA)*
  - NA-viesti on vastaus NS-viestiin. Solmu voi lähettää NA-viestin myös, jos sen linkkikerroksen osoite vaihtuu.
- *Redirect*
  - Reitittimet voivat lähettää uudelleenohjausviestin kertoakseen päätelaitteille paremman *first-hop* -osoitteen saavuttaakseen tietyn kohteen.

Em. kysymysten lisäksi ND-protokolla ratkaisee myös seuraavat ongelmat: [88]

- Linkkikerroksen osoitteen vaihtuminen
  - Vaikka NUD-algoritmi huolehtii siitä, että linkkikerroksen osoitteen vaihtuessa kaikki naapurit saavat selville uuden osoitteen, solmu voi myös lähettää naapureilleen NA-viestin, jolloin ne saavat tietää muutoksesta nopeammin.
- Kuorman jakaminen

  - ND-protokolla mahdollistaa reitittimelle tulevan liikenteen kuorman jakamisen useammalle samalla linkillä olevalle rajapinnalle. Reititin voi jättää lähettämistään RA-viesteistä linkkikerroksen lähdeosoitteen pois, jolloin naapurien on lähetettävä NS-viesti oppiakseen reitittimen linkkikerroksen osoitteen. Reititin voi sen jälkeen lähettää NA-viestissä eri linkkikerroksen osoitteen esim. sen perusteella, kuka NS-viestin lähetti.

- *Anycast*-osoitteet

  - *Anycast*-osoitteet aiheuttavat sen, että solmu voi vastaanottaa usean NA-viestin samalla kohdeosoitteella. ND-protokolla käsittelee *anycast*-osoitteen sisältäviä NA-viestejä ei-ylikirjoittavina, eli ne eivät ylikirjoita toisen NA-viestin samaan *anycast*-osoitteeseen lähettämän viestin informaatiota.

- *Proxy*-mainostukset

  - Solmu, joka haluaa vastaanottaa IPv6-paketteja sellaisen solmun puolesta, joka ei pysty vastaamaan NS-viesteihin voi lähettää ei-ylikirjoittavan NA-viestin. *Mobile IPv6 Home Agentit* käyttävät *proxy*-mainostuksia puolustaakseen mobiilin solmun osoitetta, kun se poistuu verkosta.

ND-protokolla yhdistää siis ARP- (*Address Resolution Protocol*), *ICMP Router Discovery* - ja *ICMP Redirect* -protokollien toiminnallisuuden. Siinä on myös useita parannuksia ja lisäyksiä verrattuna IPv4-protokolliin, joissa ei ole mm. yhteisesti sovittua tapaa tavoittamattoman naapurin havaitsemiseen. Muita ND-protokollan etuja on listattu alla: [88]

- RD on osa perusprotokollaa.

- RA-viestit sisältävät linkkikerroksen osoitteet ja verkon prefiksin sekä mahdollistavat IPv6-osoitteen automaattisen konfiguroinnin.

- Reititin voi mainostaa MTU:ta päätelaitteille varmistaakseen, että ne käyttävät samaa MTU:ta linkeillä, joilla sitä ei ole eksplisiittisesti määritelty.

- Osoitteenselvitykseen käytettävät *multicast*-viestit lähetetään 2^{24} eri multicast-osoitteeseen, eikä niitä tulviteta kuten ARP-protokollassa.

- Uudelleenohjausviestit sisältävät uuden ensimmäisen hypyn linkkikerroksen osoitteen, eikä erillistä osoitteenselvitystä tarvita.

- Samalle linkille voidaan yhdistää useita prefiksejä. Päätelaitteet oppivat RA-viesteistä kaikki linkillä olevat prefiksit. Reititin voi kuitenkin tietoisesti jättää tietyn prefiksin pois RA-viestistä, jolloin päätelaite olettaa sen olevan eri aliverkossa ja lähettää ko. prefiksin liikenteen reitittimelle, joka voi lähettää päätelaitteelle uudelleenohjausviestin.

- Toisin kuin IPv4:ssä, IPv6:ssa uudelleenohjausviestin vastaanottaja olettaa, että uusi ensimmäinen hyppy on samalla linkillä.
- NUD-algoritmi on osa perusprotokollaa ja se parantaa pakettien välityksen luotettavuutta. Liikkuvat solmut voivat poistua verkosta, eikä niille enää lähetetä paketteja vanhentuneen ARP-tietueen takia.
- Toisin kuin ARP, NUD-algoritmi tunnistaa myös tilanteet, joissa linkki toimii vain toiseen suuntaan.
- Toisin kuin *ICMP Router Discovery* -protokollassa, RA-viesteissä ei ole preferenssikenttää. NUD-algoritmi tunnistaa välittömästi toimimattoman reitittimen ja vaihtaa käyttämään toimivaa reititintä.
- Linkkilokaalien osoitteiden käyttö reitittimien tunnistamiseen mahdollistaa päätelaitteiden ja reittittimien välisen assosiaation säilyttämisen, vaikka globaali prefiksi vaihtuisikin.
- Asettamalla hyppyrajoituksen arvoksi 255 ND havaitsee toiselta linkiltä tulleet viestit, koska niiden hyppyrajoituskentän arvo on tällöin pienempi kuin 255. IPv4:ssä voidaan lähettää sekä ICMP-uudelleenohjauksia että RA-viestejä toiselta linkiltä.
- Osoitteenselvityksen tekeminen ICMP-kerroksella tekee siitä mediariippumattomamman kuin mitä ARP on ja mahdollistaa esim. IP-autentikointiotsikon tai ESP-protokollan käytön.

3.5.3 SLAAC

Päätelaite voi konfiguroida verkkorajapinnalleen automaattisesti IPv6-osoitteen. Tätä toiminnallisuutta kutsutaan tilattomaksi osoitteen automaattiseksi konfiguroinniksi, ja se sisältää linkkilokaalin ja globaalien osoitteiden luomisen sekä duplikaattiosoitteiden tunnistamisen. On tärkeää huomata, että osoitteen automaattinen konfigurointi ei vaadi päätelaitteen eikä yleensä myöskään reitittimen manuaalista konfiguroimista, eikä esim. DHCP-palvelinta. Päätelaite muodostaa osoitteensa sillä olevan paikallisen ja reitittimen mainostaman tiedon perusteella. Kuten aiemmin kerrottiin, IPv6-osoite muodostuu reitittimen mainostamasta verkkoprefiksistä ja päätelaitteen muodostamasta rajapintatunnisteesta. Nämä yhdistämällä päätelaite saa käyttöönsä globaalin IPv6-osoitteen. Ilman reititintäkin päätelaite voi muodostaa itselleen linkkilokaalin osoitteen, joka riittää kommunikointiin samalla linkillä olevien IPv6-solmujen kanssa. SLAAC voi muodostaa IPv6-osoitteen myös käyttämällä pseudosatunnaista 64-bittistä lukua rajapintatunnisteen sijaan. [102, 103]

Edellä kuvattu mekanismi on täysin tilaton, eikä osoitteistukseen voida manuaalisesti vaikuttaa. Jos halutaan esim. pakottaa tietty IPv6-osoite tietylle päätelaitteelle, voidaan käyttää DHCPv6-protokollaa, josta kerrotaan seuraavassa luvussa. Jotta voidaan varmistua siitä, että automaattisella konfiguroinnilla luodut osoitteet ovat yksilöllisiä, mekanismiin kuuluu myös duplikaattiosoitteiden tunnistus. Tämä

algoritmi suoritetaan jokaiselle osoitteelle riippumatta siitä, onko osoite muodostettu automaattisella konfiguroinnilla, onko se saatu DHCPv6-protokollalla vai onko se manuaalisesti konfiguroitu. [102]

SLAAC saattaa kuulostaa helpolta tavalta tehdä IPv6-käyttöönotto. Sitä ei kuitenkaan pitäisi käyttää palvelinverkkosegmentissä, jos verkon ylläpitäjä ei ole varma siitä, että IPv6-tuki on otettu pois päältä palvelimilta, joilla sen ei pitäisi olla päällä. Aiemmin esitellyn *Router Advertisement* -viestin *autonomous address configuration* -lippu kertoo päätelaitteelle, voiko se käyttää RA-viestin verkkoprefiksiä konfiguroidakseen itselleen IPv6-osoitteen. Vaikka tämä lippu ei olisi päällä eikä päätelaite täten konfiguroi itselleen IPv6-osoitetta, oletusyhdyskäytävän sisältävän RA-viestin saadessaan UNIX-pohjainen palvelin voi silti luulla olevansa *dual-stack* -verkossa, *dual stack* tietoiset sovellukset alkavat muodostaa IPv6-sessioita ja palvelin lähettää TCP SYN -paketteja käyttäen linkkilokaalia IPv6-osoitettaan. Windows Server 2008 -palvelin ei tarvitse edes RA-viestiä, vaan IPv6 on oletuksena päällä ja IPv6-pinoa preferoidaan IPv4-pinon yli [104]. Saadessaan paketin linkkilokaalilla lähdeosoitteella oletusyhdyskäytävä pudottaa sen ja lähettää palvelimelle vastauksena *destination unreachable (beyond scope of source address)* -ICMPv6-viestin. Palvelimesta riippuen se joko reagoi tähän ICMP-viestiin, jolloin sovellus toipuu nopeahkosti, tai se ei reagoi lainkaan, jolloin sovellus jää odottamaan TCP SYN -aikakatkaisua. [105]

3.5.4 DHCPv6

DHCPv6 on tilallinen vastine SLAAC:lle. Sitä voidaan käyttää sekä yhdessä SLAAC:n kanssa että erikseen, jos halutaan kontroilloida tiukemmin sitä, mikä päätelaite saa käyttöönsä minkäkin IPv6-osoitteen. DHCPv6-protokollan asiakas-palvelin-arkkitehtuuri on vastaava kuin DHCPv4:ssä: asiakaspäätelaite lähettää DHCPv6 *multicast* -osoitteeseen *solicit*-viestin, johon DHCPv6-palvelin vastaa *advertise*-viestillä. Seuraavaksi päätelaite lähettää palvelimelle *request*-viestin, johon palvelin vastaa *reply*-viestillä, joka sisältää pyydetyt konfiguraatioparametrit. DHCPv6-*multicast*-osoitteita on kaksi: [106]

- FF02::1:2, kaikki DHCPv6-välitysagentit ja -palvelimet
- FF05::1:3, kaikki DHCPv6-palvelimet

Router Advertisement -viestillä on tärkeä rooli myös DHCPv6:n toiminnassa: Se kertoo, tuleeko päätelaitteen käyttää DHCPv6:sta osoitteen, muiden konfiguraatoparametrien vai sekä osoitteen että muiden konfiguraatioparametrien noutamiseen DHCPv6-palvelimelta. Jos RA-viestin O-bitti on päällä, päätelaite voi käyttää DHCPv6:sta muiden konfiguraatioparametrien (esim. DNS-nimipalvelimien) noutamiseen. Jos taas RA-viestin M-bitti on päällä, voi päätelaite noutaa IPv6-osoitteen DHCPv6-palvelimelta. Jos RA-viestin O-bitti on päällä mutta M-bitti ei, päätelaite voi käyttää SLAAC:ia IPv6-osoitteen konfiguroimiseen, mutta käyttää DHCPv6:sta esim. saadakseen tiedon käytettävistä DNS-nimipalvelimista. Tällaiseen toiminnallisuuteen viitataan usein tilattomana DHCPv6:na, koska DHCP-palvelimen ei tarvitse ylläpitää tietoa jaetuista osoitteista. [106, 107]

DHCPv6 olettaa, että DHCPv6-palvelin on samassa aliverkossa kuin päätelaite. Näin ei suinkaan aina ole, jolloin DHCPv6-palvelin korvataan DHCPv6-välitysagentilla (*relay agent*). Välitysagentti enkapsuloi päätelaitteen lähettämät viestit *Relay-forward* -viestiin ja lähettää ne eteenpäin DHCPv6-palvelimelle. DHCPv6-palvelin enkapsuloi vastausviestinsä *Relay-repl* -viestiin, jonka välitysagentti dekapsuloi ja lähettää edelleen päätelaitteelle. DHCPv6:ssa on kolme uutta viestityyppiä verrattuna DHCPv4:ään: [106, 107]

- CONFIRM (4)
  - Päätelaite voi lähettää *Confirm*-viestin DHCPv6-palvelimelle varmistaakseen, että sen osoite on vielä voimassa.
- RELAY-FORW (12)
  - Välitysagentti lähettää *Relay-forward*-viestin välittääkseen DHCPv6-viestin joko toiselle välitysagentille tai DHCPv6-palvelimelle.
- RELAY-REPL (13)
  - DHCPv6-palvelin lähettää *Relay-reply*-viestissä välitysagentille viestin, jonka välitysagentti välittää päätelaitteelle.

DHCPv4-viestejä vastaavat DHCPv6-viestit on listattu taulukossa 9.

Taulukko 9: DHCPv4- ja DHCPv6-protokollien viestityypit. [107]

DHCPv4	**DHCPv6**
SOLICIT (1)	DHCPDISCOVER
ADVERTISE (2)	DHCPOFFER
REQUEST (3), RENEW (5), REBIND (6)	DHCPREQUEST
REPLY (7)	DHCPACK/DHCPNAK
RELEASE (8)	DHCPRELEASE
INFORMATION-REQUEST (11)	DHCPINFORM
DECLINE (9)	DHCPDECLINE
CONFIRM (4)	-
RECONFIGURE (10)	DHCPFORCERENEW
RELAY-FORW (12),RELAY-REPL (13)	-

DHCPv6:ta voidaan käyttää myös välittämään reitittimelle IPv6-verkon prefiksi. Internet-operaattori voi esikonfiguroida tämän toiminnallisuuden päälle asiakkailleen lähettämiinsä reitittimiin ja varmistua siitä, että he saavat näin automaattisesti päätelaitteilleen IPv6-osoitteen joko SLAAC:lla tai DHCPv6:lla. DHCPv6-päätelaitteet kuuntelevat DHCPv6-viestejä UDP-portissa 546 ja DHCPv6-palvelimet sekä -välitysagentit UDP-portissa 547. [107, 108, 109, 110]

3.5.5 DHCPv6 vs. DHCPv4

DHCPv6 ei tue oletusyhdyskäytävän välittämistä päätelaitteelle. SLAAC ei puolestaan tue DNS-nimipalvelimien eikä DNS-hakulistan välittämistä päätelaitteelle. RA-viesteihin julkaistiin vuonna 2010 laajennukset, joilla tämä tosin onnistuu [111]. Myös oletusyhdyskäytävän välittämisestä DHCPv6-optiolla on olemassa Internet-drafti, mutta kumpikaan ominaisuus ei vielä ole tuettu useimmissa käyttöjärjestelmissä [112]. Tämä tarkoittaa sitä, että tarvitaan välttämättä siis kahta eri mekanismia välittämään päätelaitteelle sama informaatio kuin DHCPv4:lla.

Taulukko 10: DHCPv4, DHCPv6 & SLAAC/RA. [113]

Parametri	**DHCPv4**	**DHCPv6**	**SLAAC/RA**
IPv6-osoite	x	x	x
Oletusyhdyskäytävä	x		x
Nimipalvelin	x	x	
DNS-hakulista	x	x	

Koska oletusyhdyskäytävän välittäminen DHCPv6-optiolla on vasta drafti ja RA-viestien DNS-tuki puuttuu useimmista käyttöjärjestelmistä, DHCPv4-toiminnallisuutta voidaan jäljitellä käyttämällä joko ns. tilallista tai tilatonta DHCPv6:ta riippuen siitä, jaetaanko osoitteet DHCPv6:lla vai muodostetaanko ne SLAAC:lla. Kummallakin vaihtoehdolla on hyvät ja huonot puolensa. Osoitteiden jakamisessa DHCPv6:lla on seuraavia etuja ja haittoja: [114]

- Palveluntarjoajalla on täysi kontrolli osoitteista ja siitä, miten niitä jaetaan.
- DDNS-päivitykset (*Dynamic DNS*) tehdään keskitetystä pisteestä (DHCP-palvelimelta).
- Tuntemattomat päätelaitteet voidaan eristää verkosta kokonaan tai niiden pääsyä voidaan rajoittaa.
- DHCPv6-tukea ei välttämättä ole vielä kaikissa käyttöjärjestelmissä.

Osoitteiden muodostamisessa SLAAC:lla on seuraavia etuja ja haittoja: [114]

- SLAAC-tuki on kaikissa käyttöjärjestelmissä, joissa on IPv6-tuki.
- Ne päätelaitteet, jotka eivät tue DHCPv6:ta saavat silti toimivan yhteyden DNS-nimipalvelimia lukuunottamatta.
- Palveluntarjoalla ei ole kontrollia siitä, miten osoitteita jaetaan.
- Kuten aiemmin mainittiin, SLAAC muodostaa osoitteen verkkorajapinnan MAC-osoitteeseen perustuen, mikä voi olla turvallisuusriski.
- DDNS-päivitykset täytyy erikseen sallia jokaiselta verkkoon liitetyltä päätelaitteelta.

4 Tietoturva & transitiomekanismit

Tietoturvasta IPv6-protokollaan liittyen voisi kirjoittaa kokonaisen diplomityön, joten tässä on tarkoitus käydä lähinnä vain IPv6-käyttöönoton kannalta oleelliset asiat läpi. Toisaalta, koska käyttöönotto tehdään Capgeminin tuotantoverkossa, ei tässä voida myöskään esittää tarkkoja konfiguraatiorivejä, joilla käyttöönotto tehdään. Siksi seuraavaksi esitelläänkin parhaita käytäntöjä, joilla IPv6-verkon eri komponentteja kannattaa suojata. Kuten aiemmin kävi ilmi, IPv6-osoitteita on olemassa moneen eri käyttötarkoitukseen. Tämä tarkoittaa sitä, että Internetin tai minkä tahansa muun verkon reunareitittimillä kannattaa estää tiettyjen reittien vastaanottaminen ja myös tietyntyyppinen IPv6-liikenne. Verkon pääsykerroksella taas puolestaan kannattaa oletusarvoisesti estää esim. ND-protokollan RA- ja DHCPv6-viestit. [115]

4.1 Pääsykerros

Cisco on implementoinut tuotteisiinsa useita pääsykerroksella käytettäviä suojausmekanismeja. IPv4-suojausmekanismeja kutsuttiin CISF-mekanismeiksi (*Catalyst Integrated Security Features*) ja IPv6-suojausmekanismit ovat nimeltään FHS-mekanismeja (*First Hop Security*) [116]. Näitä ovat esim. seuraavat: [117]

- RA Guard
- DHCP Guard
- IPv6 Snooping
- Source/Prefix Guard
- Destination Guard

RA- ja DHCP Guard -mekanismeilla voidaan suojautua väärennetyiltä RA-viesteiltä ja DHCP-palvelimilta. Sekä RA- että DHCP Guard -mekanismien konfiguroimiseen on olemassa omat komentonsa, mutta ne saadaan konfiguroitua myös yksinkertaisella porttipääsynhallintalistalla. Jos verkossa on laite, joka lähettää väärennettyjä RA-viestejä, voi se saada verkossa olevan päätelaitteen konfiguroimaan itselleen automattisesti IPv6-osoitteen. Puolestaan, jos verkossa on laite joka esittää olevansa DHCPv6-palvelin, voi se vastata päätelaitteen lähettämään *solicit*-viestiin ja antaa sille haluamansa IPv6-osoitteen. Tämä vaatii vielä sen, että päätelaite on konfiguroitu hakemaan IPv6-osoite DHCPv6-palvelimelta, mutta jos verkossa on sekä väärennetty reititin että DHCPv6-palvelin, hyökkääjä voi käyttää reitittimen lähettämän RA-viestin O- ja M-bittejä aiemmin esitetyllä tavalla ja saada verkossa olevan päätelaitteen hakemaan itselleen IPv6-osoitteen tai muita konfiguraatioparametreja väärennetyltä DHCPv6-palvelimelta. Mainittakoon, että näitä ongelmia ratkaisemaan on kehitetty SEND-protokolla (*SEcure Neighbor Discovery*), mutta se ei ole tuettu kaikissa laitteissa ja on muutenkin hankala implementoida [118, 119]. Sekä DHCP-liikenne että RA-viestit kannattaakin estää mahdollisuuksien mukaan

heti pääsykytkimen portissa, jos siinä ei ole kiinni DHCP-palvelinta tai reititintä. Esim. Ciscon 4500- ja 6500-sarjan kytkimissä voidaan käyttää *ipv6 nd raguard* -komentoa, jolla saadaan aikaan sama lopputulos kuin ao. pääsylistan *deny icmp any any router-advertisement* -rivillä. Tämä pääsylista estää kuitenkin myös DHCPv6-liikenteen kytkinportista. [120]

```
ipv6 access-list IPV6_ACCESS_PORT
 remark Block all traffic DHCP server -> client
 deny udp any eq 547 any eq 546
 remark Block Router Advertisements
 deny icmp any any router-advertisement
 permit any any
!
interface gigabitethernet 1/0/1
 switchport
 ipv6 traffic-filter IPV6_ACCESS_PORT in
!
```

4.2 Reunareititin

Ciscon tunnistamat parhaat käytännöt IPv6-reittien vastaanottamiseen Internetin – tai minkä tahansa muun verkon reunalla olevalla reitittimellä ovat seuraavat: [121]

- Kielletään oletusreitti:

  ```
  ipv6 prefix-list v6in-filter deny ::/0
  ```

- Kielletään *loopback*-osoite ja määrittelemätön sekä IPv4-yhteensopivat ja -*mäpätyt* osoitteet:

  ```
  ipv6 prefix-list v6in-filter deny ::/8 le 128
  ```

- Sallitaan Teredo-osoitteet:

  ```
  ipv6 prefix-list v6in-filter permit 2001::/32
  ipv6 prefix-list v6in-filter deny 2001::/32 le 128
  ```

- Kielletään dokumentointiin käytettävät osoitteet:

  ```
  ipv6 prefix-list v6in-filter deny 2001:db8::/32 le 128
  ```

- Sallitaan 6to4-osoitteet:

  ```
  ipv6 prefix-list v6in-filter permit 2002::/16
  ipv6 prefix-list v6in-filter deny 2002::/16 le 128
  ```

- Kielletään omat osoitteet:

  ```
  ipv6 prefix-list v6in-filter deny YOUR_CIDR_BLOCK_IPV6 le 128
  ```

- Kielletään vanhat 6bone-osoitteet:

```
ipv6 prefix-list v6in-filter deny 3ffe::/16 le 128
```

- Kielletään ULA-osoitteet:

```
ipv6 prefix-list v6in-filter deny fc00::/7 le 128
```

- Kielletään linkkilokaalit osoitteet:

```
ipv6 prefix-list v6in-filter deny fe80::/10 le 128
```

- Kielletäään *multicast*-osoitteet:

```
ipv6 prefix-list v6in-filter deny ff00::/8 le 128
```

- Sallitaan globaalit *unicast*-osoitteet:

```
ipv6 prefix-list v6in-filter permit 2000::/3 le 48
```

- Kielletään muut osoitteet:

```
ipv6 prefix-list v6in-filter deny ::/0 le 128
```

Myöhemmin tässä työssä tehtävässä IPv6-käyttöönotossa noudatetaan luonnollisesti näitä parhaita käytäntöjä. Yllä esiintyvistä osoitteista originoituvan liikenteen estämisen lisäksi on muutamia muita parhaita käytäntöjä, jotka kannattaa ottaa huomioon reunareitittimen pääsylistoissa: [122]

- Pudotetaan fragmentoidut paketit:

```
deny ipv6 any YOUR_CIDR_BLOCK_IPV6 fragments
```

- Pudotetaan omasta IPv6-avaruudesta tulevat paketit:

```
deny ipv6 YOUR_CIDR_BLOCK_IPV6 any
```

- Sallitaan naapuruus BGP-naapurin kanssa:

```
permit tcp host bgp_peer_ipv6 host router_ipv6 eq bgp
permit tcp host bgp_peer eq bgp host router_ipv6
```

- Pudotetaan IPv6-liikenne linkkiverkkoihin ym. verkkoinfrastruktuuriin:

```
deny ipv6 any INTERNAL_INFRASTUCTURE_ADDRESSES
```

- Sallitaan IPv6-liikenne omaan IPv6-avaruuteen:

```
permit ipv6 any YOUR_CIDR_BLOCK_IPV6
```

Lopuksi reunareitittimellä kannattaa pudottaa *hop-by-hop*-optio-otsikot ja tyypin 0 reititysotsikot, ottaa URPF-toiminnallisuus (*unicast reverse path forwarding*) käyttöön sekä ICMPv6-uudelleenohjausviestit pois käytöstä. [123, 124]

```
deny hbh any any
deny ipv6 any any routing-type 0
ipv6 verify unicast reverse-path
no ipv6 redirects
```

4.3 NDP

Ehkä tärkein kysymys IPv6-käyttöönoton tietoturvaan liittyen on kuitenkin siinä käytettävien aliverkkojen koko. IPv4-reititin ylläpitää ARP-taulua, ja kuten aiemmin todettiin, IPv6:ssa ND-protokolla korvaa ARP-protokollan. IPv6-reititin ylläpitää siis NDP-taulua IPv6-naapureistaan. Ciscon reitittimillä NDP-taulun sisällön näkee *sh ipv6 neighbor* -komennolla:

```
cap-lab-ipv6-fw01# sh ipv6 neighbor
IPv6 Address                              Age Link-layer Addr State Interface
fe80::250:56ff:fea5:7180                    0 0050.56a5.7180  REACH inside
fe80::222:55ff:fee4:8b1a                   79 0022.55e4.8b1a  STALE outside
fe80::222:55ff:fee4:7f1a                   79 0022.55e4.7f1a  STALE outside
...                                        15 0050.56a5.7180  STALE inside
fe80::5:73ff:fea0:3b                       79 0005.73a0.003b  STALE outside
```

Kuten aiemmin näytettiin, lähes kaikki IPv6-RFC-dokumentit perustuvat siihen olettamukseen, että IPv6-osoitteen 64 ensimmäistä bittiä käytetään verkon prefiksin ja 64 jälkimmäistä rajapinnan tunnistamiseen. Yleisesti ollaankin sitä mieltä, että /64-prefiksien konfiguroiminen on ainoa oikea ratkaisu ja että pidempien prefiksien käyttäminen rikkoo IPv6:n tukiprotokollien kuten ND-protokollan toiminnan [125]. Tässä työssä näytetään kuitenkin myöhemmin, että ND-protokolla toimii myös pidemmillä prefikseillä. /64-prefiksin ongelma liittyy em. ND-tauluun: /64-prefiksin verkossa on 2^{64} eli 18446744073709551616 IPv6-osoitetta. Esim. Cisco Nexus 5500 -kytkimen ND-tauluun puolestaan mahtuu 6500 riviä eli /64-prefiksi on 2837960626724546 kertaa isompi kuin sen NDP-taulun koko. Tämä tarkoittaa sitä, että mikä tahansa /64-prefiksin verkossa oleva laite voi tulvittaa reitittimen ND-taulun, jolloin ND-protokollan toiminta reitittimellä estyy. [124, 126]

Ratkaisu ND-tulvitusongelmaan voisivatkin olla pidemmät, esim. /120-prefiksit. Ne vastaavat /24-IPv4-aliverkkoja, joten niiden pitäisi usein olla konesaliympäristössä riittävän kokoisia. Yksi ratkaisu on allokoida /64-prefiksi, mutta konfiguroida /120-prefiksi ja muuttaa se isommaksi, kun ND-tulvitukseen liittyvät ongelmat on ratkaistu em. FHS-mekanismeilla. /120-prefiksi rikkoo mm. SLAAC:n toiminnan, mutta tämä ei ole ongelma, koska sitä ei haluta Capgeminin konesaliverkossa käyttää. Tämä ratkaisu saattaa kuitenkin osoittautua käyttökelvottomaksi reitittimien TCAM-rajoitusten (*Ternary Content Addressable Memory*) vuoksi. Esim. Cisco Nexus 5500 -kytkin tukee 16384 /64-reittiä, mutta vain 128 muuta kuin /64-reittiä. [126, 127]

Toinen ratkaisu onkin pelkän allokoimisen lisäksi myös konfiguroida /64-prefiksi, mutta käyttää verkossa osoitteita vain ensimmäisestä /120-prefiksistä ja rajoittaa reitittimellä tai palomuurilla liikenne tähän ensimmäiseen /120-prefiksiin. Koska /64-prefiksin tiedetään toimivan ilman ongelmia, konfiguroidaan tässä työssä IPv6-testiverkko ensin /120- ja sitten /121-prefiksillä nähdeksemme, saadaanko niitä käyttämällä aikaan toimiva IPv6-verkko. Nämä valinnat tehtiin siitä syystä, että /120-prefiksi vastaa /24-IPv4-aliverkkoa ja /121-prefiksi vastaavasti /25-IPv4-aliverkkoa, jotka ovat kummatkin yleisiä konesaliympäristössä käytettäviä IPv4-aliverkkoja. [126]

4.4 IPv6-transitiomekanismit

IPv6-transitioon on määritelty kolmenlaisia mekanismeja: kahden protokollapinon käyttö (*dual stack*), tunnelointi ja pakettimuunnokset. Yleisesti voidaan todeta, että ainakin konesaliympäristössä kahden protokollapinon käyttö on näistä suositeltavin ratkaisu ja tunnelointiin sekä varsinkin pakettimuunnoksiin kannattaa turvautua vain, jos päätelaitteet eivät tue IPv6-protokollaa. Seuraavaksi esitellään kukin mekanismi ja niihin kuuluvia protokollia lyhyesti.

4.4.1 Kaksi protokollapinoa (*dual stack*)

Kahden protokollapinon ratkaisussa joko solmun samalle tai sen kahdelle eri verkkorajapinnalle konfiguroidaan sekä IPv4- että IPv6-osoitteet, jolloin se käyttää IPv4- tai IPv6-pinoa sen mukaan, kummalla protokollalla siihen otetaan yhteyttä. Kahden protokollapinon käytön suurimpia etuja ovat, että se on helppo ottaa käyttöön ja että IPv4-tuki säilyy muuttumattomana. Haittoja ovat kaksi reititysprosessia ja -taulua, jotka kuormittavat prosessoria ja lisäävät muistinkäyttöä. Esimerkki Ciscon reitittimen kahden protokollapinon konfiguraatiosta, jossa rajapinnalle on asetettu sekä IPv4- että IPv6-osoitteet, on esitetty alla. [128]

```
interface Ethernet 0
 ip address 192.168.100.1 255.255.255.0
 ipv6 address 2001:db8:1:1::1/64
!
```

4.4.2 Tunnelointi (enkapsulointi)

Tunneloinnilla tarkoitetaan yleensä saman tai alemman protokollakerroksen protokollan enkapsuloimista toiseen, ylemmän tason protokollaan. Seuraavissa esimerkeissä IPv6-paketti enkapsuloidaan joko saman, verkkokerroksen IPv4-paketin hyötykuormaksi tai ylemmän, esim. kuljetuskerroksen UDP-datagrammin hyötykuormaksi. Tunneloinnilla on se hyvä puoli, että uusi protokolla voidaan ottaa käyttöön häiritsemättä vanhan protokollan toimintaa. [129]

- 6in4
  - *6in4*-tunneli muodostetaan kahden julkisen Internetissä reitittyvän IPv4-osoitteen välille, ja tunnelin päät enkapsuloivat ja dekapsuloivat IPv6-paketin IPv4-pakettiin/paketista suunnasta riippuen. IPv6-paketti näkee tunnelin vain yhtenä hyppynä, vaikka IPv4-paketti saattaa kulkea monenkin hypyn yli. [129]
- 6to4 (*Connection of IPv6 Domains via IPv4 Clouds*)
  - 6to4 mahdollistaa IPv6-päätelatteiden tai -verkkojen kommunikoinnin IPv4-verkon yli. 6to4 muodostaa /48-verkkoprefiksin IPv4-osoitteen perusteella siten, että 2002::/16-prefiksiin lisätään heksadesimaalilukuna 32-bittinen IPv4-osoite. IPv6-paketti enkapsuloidaan IPv4-pakettiin 6to4-reitittimellä ja lähetetään IPv4-verkkoon. [130]

- 6rd (*IPv6 Rapid Deployment on IPv4 Infrastructures*)
  - 6rd on muokattu versio 6to4-tunnelointiprotokollasta. Siinä palveluntarjoaja käyttää omaa IPv6-prefiksiään 6to4:n 2002::/16-prefiksin sijaan. Asiakkaan näkökulmasta 6rd-protokollalla toteutettu yhteys ei eroa natiivista IPv6-toteutuksesta. Vuonna 2007 ranskalainen Internet-operaattori Free of the Iliad group teki viidessä viikossa 6rd-toteutuksen, jonka jälkeen 1500000 operaattorin asiakkaalla oli käytössään IPv6-yhteys heidän niin halutessaan. [131, 132]
- 6over4
  - 6over4 vastaa hyvin pitkälti 6to4:ää, mutta se on tarkoitettu vain päätelaitteille. 6over4 käyttää IPv4-monilähetystä virtuaalisen linkin luomiseen kahden IPv6-päätelaitteen välille, joten päätelaitteiden täytyy tukea sekä IPv4:ää että IPv6:ta. 6over4 muodostaa IPv6-osoitteen niin, että /64-prefiksiin lisätään 4 tavua nollia ja 4-tavuinen IPv4-osoite. [133]
- ISATAP (*Intra-Site Automatic Tunnel Addressing Protocol*)
  - Myös ISATAP on tarkoitettu tunneloimaan kahta protokollapinoa tukevien solmujen IPv6-paketit IPv4-verkon yli, mutta toisin kuin 6over4:ssä, ISATAP ei vaadi IPv4-verkolta monilähetystukea. Se muodostaa rajapintatunnisteensa yhdistämällä 24-bittisen IANA OUI:n (00-00-5E), 8-bittisen heksadesimaaliarvon 0xFE ja 32-bittisen IPv4-osoitteen. OUI:n ensimmäisen tavun *u*- ja *g*-bittien vuoksi ensimmäinen tavu ei aina ole 0 (ks. luku 3.3). ISATAP muodostaa myös linkkilokaalin osoitteen, joka saadaan lisäämällä FE80::/64-prefiksiin em. rajapintatunniste. [134]
- Tunnelinvälittäjät (*tunnel broker*)
  - Internetissä toimii monta ns. tunnelinvälittäjää, joita voidaan kutsua virtuaalisiksi IPv6-operaattoreiksi. Ne tarjoavat yleensä maksua vastaan IPv6-tunnelin IPv4-verkon yli kahta protokollapinoa tukevalle reitittimelle tai päätelaitteelle. TSP-protokollaa (*Tunnel Setup Protocol*) voidaan käyttää tunnelin automaattiseen neuvottelemiseen ja jossain määrin myös NAT:n aiheuttamien ongelmien ratkaisemiseen. [135, 136] Ilmaisia tunnelinvälittäjiä ovat esim. Hurricane Electric ja SixXS, joilta saa oman /48- tai /64-aliverkon, jota voi käyttää IPv6-tunnelointiin. [137, 138] HE:ltä saa kaksi /64-aliverkkoa, joista toinen on linkkiverkko HE:n PoP:n (*Point of Presence*) ja käyttäjän reitittimen tai päätelaitteen välillä ja toinen on tarkoitettu aliverkotukseen. Lähin HE:n PoP on Tukholmassa, Ruotsissa. Seuraavassa on esitetty tiedot omasta IPv6-tunnelistani:

IPv6 Tunnel Endpoints	
Server IPv4 Address:	216.66.80.90
Server IPv6 Address:	2001:470:27:4eb::1/64
Client IPv4 Address:	88.192.72.205
Client IPv6 Address:	2001:470:27:4eb::2/64
Routed IPv6 Prefixes	
Routed /64:	2001:470:28:4eb::/64

IPv6-tunneli on siis muodostettu IPv4-osoitteiden 216.66.80.90 ja 88.192.72.205 välille, ja käytettävät IPv6-osoitteet ovat 2001:470:27:4eb::1/64 ja 2001:470:27:4eb::2/64. Linkkiverkko on 2001:470:27:4eb::/64 ja minulle varattu IPv6-verkko 2001:470:28:4eb::/64. Komennot, joilla otin IPv6-tunnelin käyttöön omalla Windows 7 -työasemallani olivat seuraavat:

```
netsh interface teredo set state disabled
netsh interface ipv6 add v6v4tunnel IP6Tunnel 88.192.72.205 216.66.80.90
netsh interface ipv6 add address IP6Tunnel 2001:470:27:4eb::2
netsh interface ipv6 add route ::/0 IP6Tunnel 2001:470:27:4eb::1
```

SixXs:ltä saa tunnelin DNA:n PoP:sta Helsingistä, ja myös se tarjoaa ilmaisen /64-aliverkon ja -linkkiverkon:

Inner Us	2001:14b8:100:336::1
Inner Them	2001:14b8:100:336::2
Outer Us	62.78.96.38
Outer Them	88.192.72.205
IPv6 Them	2001:14b8:100:336::2/64
Prefix	2001:14b8:100:8336::/64

- Teredo *(Tunneling IPv6 over UDP through Network Address Translations (NATs))*
  - Kaikki em. mekanismit enkapsuloivat IPv6-paketin IPv4-pakettiin ja asettavat IPv4-otsikon protokollakentän arvoksi 41 (IPv6). Useimmat NAT-ratkaisut päästävät lävitseen ainoastaan TCP- tai UDP-liikennettä, joten IPv4-enkapsuloitu IPv6-liikenne ei mene niistä läpi [139]. Vaikka NAT päästäisikin liikenteen lävitseen, esim. 6to4 ei toimi, jos osoitteenmuunnos tehdään eri laitteessa kuin 6to4. Teredo on Microsoftin kehittämä IPv6-tunnelointimekanismi, joka enkapsuloi IPv6-paketit UDP/IPv4-paketteihin. Myös TSP-protokollalla voidaan enkapsuloida IPv6-paketit UDPv4-pakettihin, mutta NAT:n aikakatkaisu on silti ongelma. Teredo-protokolla toimii STUN-protokollan tavoin ja selvittää UDPv4-liikenteen avulla, millainen NAT-mekanismi liikenteen polulla on. Se jakaa globaalisti reitittyvät osoitteet jokaiselle sitä käyttävälle päätelaitteelle, enkapsuloi IPv6-paketit UDPv4-datagrammeihin ja reitittää liikennettä Teredo- ja natiivien IPv6-päätelaitteiden välillä. Teredo toimii vain *full cone* -, *restricted*- ja *port-restricted*-tyyppisten NAT-ratkaisujen kanssa, ei symmetrisellä NAT:lla ja liikenteen täytyy originoitua NAT:n takaa. [140, 141]

4.4.3 Pakettimuunnokset

Tunneloinnilla on kaksi isoa rajoitusta: uuden protokollan käyttäjät eivät voi hyödyntää vanhan protokollan tarjoamia palveluilta ja he eivät voi kommunikoida vanhan protokollan käyttäjien kanssa ilman kahta protokollapinoa. Kolmas siirtymämekanismi ovatkin pakettimuunnokset, jotka mahdollistavat molemmat em. tunneloinnin rajoituksista. [129]

- SIIT (*Stateless IP/ICMP Translation Algorithm*)
  - SIIT käyttää IP/ICMP-muunninta, joka muuntaa verkkojen rajalla IPv4/ICMPv4-paketteja IPv6/ICMPv6-paketeiksi ja toisinpäin. [142]
- NA(P)T-PT (*Network Address (Port) Translator - Protocol Translator*)
  - NAT-PT muuntaa IPv4-osoitteet IPv6-osoitteiksi ja toisinpäin ja NAPT-PT myös kuljetuskerroksen tunnisteet. SIIT-algoritmia käytetään itse protokollamuunnokseen. NA(P)T-PT:n käyttö on vanhentunut, joten sitä ei esitellä tässä enempää. [143, 144, 145]
- TRT (*Transport Relay Translator*)
  - TRT toimii vastaavasti kuin SIIT, mutta kuljetuskerroksella. Se muuntaa TCP,UDP/IPv6-paketteja TCP,UDP/IPv4-paketeiksi ja toisinpäin. [146]
- NAT64 & DNS64
  - Myös NAT64 käyttää SIIT-algoritmia IPv4-IPv6-otsikkomuunnoksiin. IPv4-osoitteet muunnetaan IPv6-osoitteiksi ja toisinpäin käyttäen algoritmia ja erityistä NAT64:lle varattua IPv6-prefiksiä. DNS64 on mekanismi, jolla voidaan samaa algoritmia käyttäen generoida A-tietueista AAAA-nimitietueita. NAT64:ää ja DNS64:ää yhdessä käyttämällä IPv6-päätelaite voi ottaa yhteyttä IPv4-palvelimeen tekemättä itse palvelimelle mitään muutoksia. Toiminta vaatii ainoastaan NAT64-toiminnallisuuden käyttöönoton IPv4- ja IPv6-verkot yhdistävissä laitteissa ja muutaman DNS64-nimipalvelimen, joita IPv6-päätelaitteet voivat käyttää nimenselvitykseen. [145, 147, 148]

Tässä luvussa on esitelty IPv6-protokollan tietoturvaa ja IPv4-IPv6-transitiomekanismeja. IPv6-protokollaan liittyy useita tietoturvakysymyksiä, jotka tulee ottaa huomioon heti verkon reunalla sekä pääsykerroksella että reunareitittimellä. Jos IPv6-verkkoa ei suojata asianmukaisesti pääsykerroksella, ARP-protokollan korvaavaan ND-protokollaan liittyy merkittävä tietoturvariski. Ennen kuin ND-protokollan tulvitusongelmat on saatu ratkaistua pääsykerroksella esim. Ciscon FHS-mekanismeilla, voidaan käyttää vain pientä osaa koko /64-IPv6-verkosta. Transitiomekanismit IPv4-protokollasta IPv6-protokollaan voidaan jakaa kolmeen kategoriaan: kahden protokollapinon käyttämiseen, tunnelointiin ja pakettimuunnoksiin. Seuraavassa luvussa esitellään IPv6-protokollaa tyypillisessä palvelinkeskuksen konesaliverkossa.

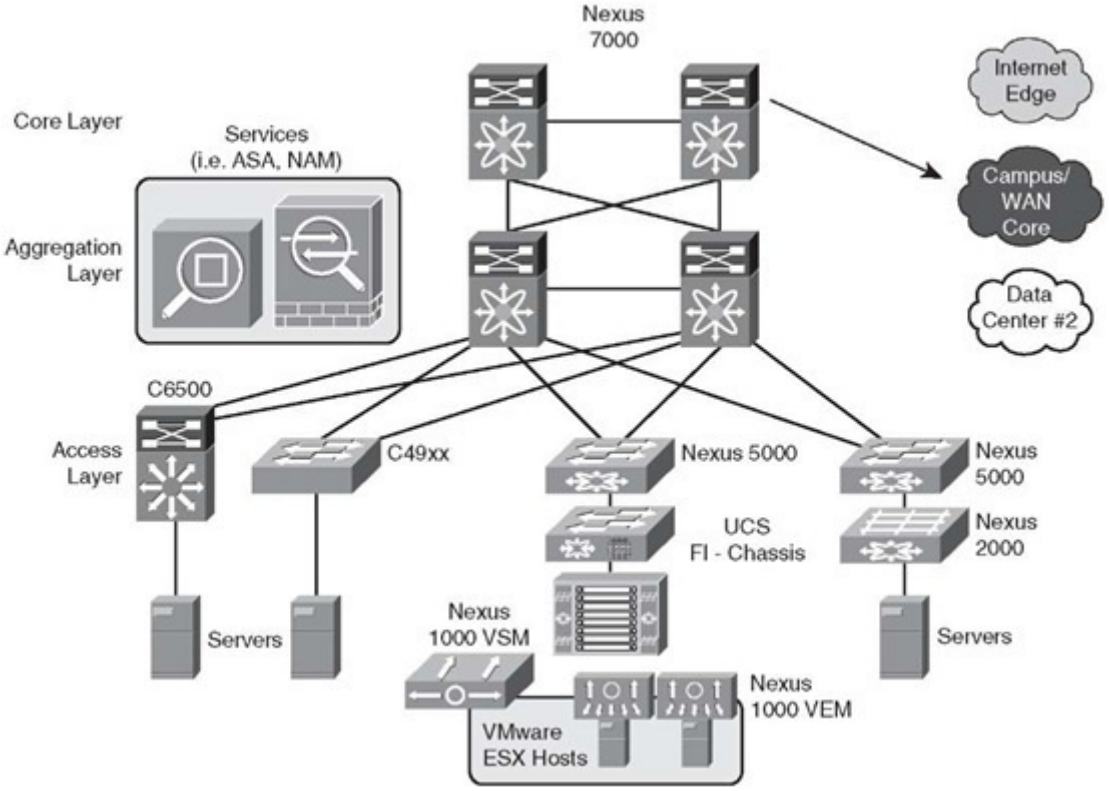
5 IPv6 palvelinkeskuksen konesaliverkossa

Kuten edellisessä luvussa todettiin, kahden protokollapinon käyttäminen on konesaliympäristössä ainoa järkevä ratkaisu IPv6-käyttöönottoon. Sen vaihtoehto on rakentaa IPv4-verkon rinnalle täysin erillinen IPv6-verkko, jossa käytetään pelkkää IPv6-protokollaa. Tämä on luonnollisesti kallis vaihtoehto sekä OPEX- että CAPEX-kustannuksiltaan, koska verkkolaitteiden, käyttöjärjestelmälisenssien, palvelimien ja hallinnoimisen kustannukset täytyy kertoa kahdella kahden erillisen verkon takia, eikä se ole siksi houkutteleva eikä järkevä vaihtoehto. Liiketoimintakriittiset sovellukset voivat kuitenkin usein olla hyvinkin vanhoja, eikä niitä haluta päivittää, jos ei ole aivan pakko. Näin ollen päädytään melkein aina siihen tilanteeseen, että konesalissa on vielä joitain palvelimia, joiden käyttöjärjestelmä tai itse sovellukset eivät tue IPv6:ta vaikka konesaliverkko tukisi täysin kahta protokollapinoa. Käytännössä tämä tarkoittaa sitä, että tarvitaan jokin tapa, jolla IPv4-päätelaitteet voivat kommunikoida IPv6-päätelaitteiden kanssa. Tällaisia tapoja ovat esim. edellisessä luvussa kuvatut tunnelointi- ja pakettimuunnosmekanismit. [35]

Tässä luvussa kerrotaan ensin, kuinka tyypillisen palvelinkeskuksen konesaliverkko on rakennettu ja mitä verkon eri kerroksilla täytyy kahden protokollapinon käyttämistä varten ottaa huomioon. Sen jälkeen esitellään Capgeminin konesaliverkon rakenne sen reitittimien, kytkimien ja palomuurien osalta sekä näytetään lukua 6 silmällä pitäen lyhyesti tärkeimmät IPv6-komennot sekä Juniperin Junos- että Ciscon IOS-käyttöjärjestelmissä. Capgeminilla on konesaliverkossaan Ciscon Catalyst-kytkimien lisäksi myös Nexus-kytkimiä, joissa on NX-OS-käyttöjärjestelmä, mutta sen konfigurointi vastaa hyvin pitkälti IOS-käyttöjärjestelmän konfiguroimista. Lopuksi esitellään kaksi kollegani tekemää IPv6-osoitteistussuunnitelmaa Capgeminin konesaliverkkoon. Aloitetaan kuitenkin esittelemällä tyypillinen palvelinkeskuksen verkkotopologia.

5.1 Palvelinkeskuksen konesaliverkko

Cisco jakaa verkkotopologiansa usein kolmeen eri kerrokseen: pääsy- (*access*), aggregointi- (*aggregation*) ja runkokerrokseen (*core*). Tämä on havainnollistettu kuvassa 39. Pääsykerroksen tehtävänä on nimensä mukaisesti tarjota päätelaitteille kuten palvelimille pääsy verkkoon joko fyysisesti vai virtuaalisesti. Aggregointikerros puolestaan on kerros, johon kaikki pääsykerroksen *uplink*-liitännät terminoidaan. Aggregointikerroksella tarjotaan usein myös erilaisia verkkopalveluita kuten IDS/IPS- (*Intrusion Detection/Prevention System*), VPN-, kuormanjako- sekä verkonvalvonta- ja -hallintapalveluita. Runkokerros puolestaan yhdistää aggregointikerroksen muuhun verkkoon kuten Internetiin tai muihin palvelinkeskuksiin. [35]



Kuva 39: Konesaliverkon pääsy-, aggregointi- ja runkokerrokset. [35]

5.1.1 Pääsykerros

Pääsykerros toimii puhtaasti ISO/OSI-mallin toisella kerroksella, joten nopeasti voisi ajatella, että sitä ei tarvitsisi IPv6-käyttöönotossa ajatella ollenkaan. Näin ei kuitenkaan välttämättä ole: jos pääsykerroksen kytkimeen kytketyn päätelaitteen täytyy vastaanottaa IPv6-multicast-liikennettä, täytyy kytkimen tukea IPv6 MLD-toiminnallisuutta (*Multicast Listener Discovery*). Pääsykerroksella saatetaan lisäksi toteuttaa IPv6 QoS-luokittelua, jolloin IPv6-pakettien luokittelu ja merkitseminen, uudelleenmerkitseminen ja päätelaitteen tekemä luokittelu täytyy sallia pääsykytkimellä. *Multicast* ja QoS ovat kummatkin tämän työn laajuuden ulkopuolella, joten niihin ei tässä paneuduta tämän tarkemmin. Pääsykerroksella saatetaan toteuttaa myös esim. *control plane* -valvontaa tai aiemmin esiteltyjä ensimmäisen hypyn tietoturvamekanismeja kuten suojautumista väärennetyiltä RA- ja DHCPv6-viesteiltä porttipääsynhallintalistalla (PACL, *Port Access Control List*). [35]

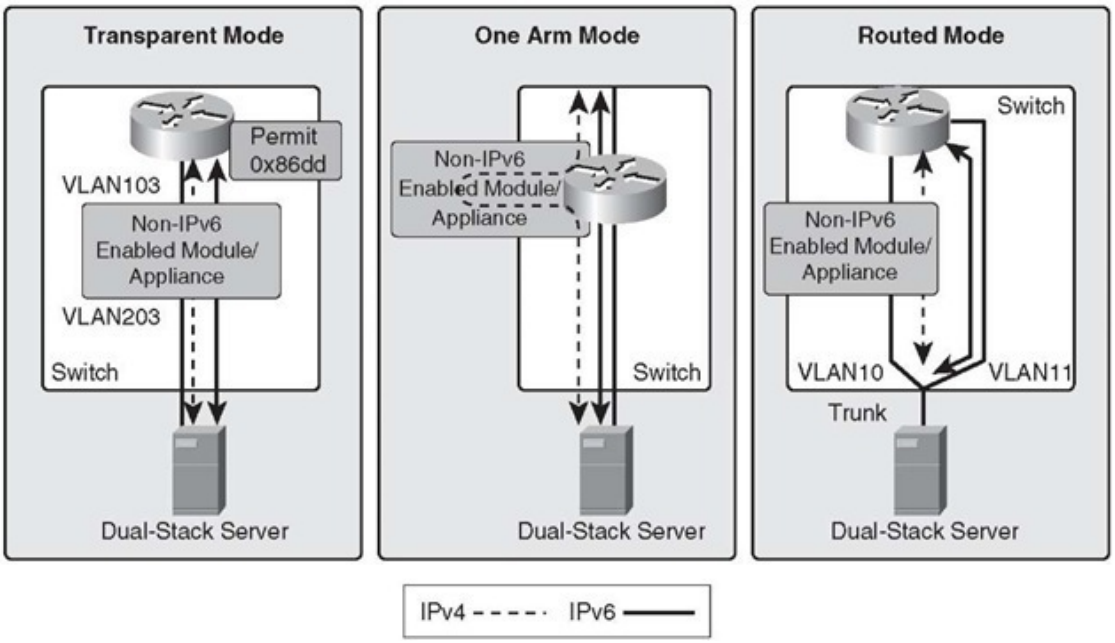
Tiimattaessa eli yhdistettäessä useita palvelimien fyysisiä verkkoliitäntöjä yhdeksi loogiseksi liitännäksi täytyy varmistua siitä, että palvelimen verkkokortti todella tukee IPv6:ta. Jos näin ei ole ja tiimatulle liitännälle annetaan IPv6-osoite, voi käydä niin, että osoite annetaankin loogisen liitännän sijaan kaikille fyysisille liitännöille eikä palvelin ole enää saavutettavissa IPv6:lla, koska DAD-algoritmi asettaa osoitteen *duplicate*-tilaan. [35]

5.1.2 Aggregointi- ja runkokerrokset

Aggregointikerroksella yhdistetään pääsykerroksen kytkimien fyysiset *uplink*-liitännät ja loogiset VLAN:t (*Virtual Local Area Network*) sekä tarjotaan useita em. verkkopalveluita. Palomuuripalvelu on yksi esimerkki verkkopalvelusta, joka tarjotaan myös Capgeminin verkossa tältä kerrokselta. Aggregointikerroksella on siis jo monia ISO/OSI-mallin kerroksilla 3-7 toimivia laitteita, joten lukuisia asioita täytyy ottaa huomioon suunniteltaessa IPv6-käyttöönottoa. [35]

Aggregointikerroksella voi olla vain IPv4:ää tukevia moduuleita tai laitteita, mutta aggregointikerroksen täytyy silti välittää IPv6-liikennettä lävitseen. Kuinka IPv6-liikenne saadaan aggregointikerroksen läpi riippuu siitä, onko laite kytketty verkkoon sillatulla (*transparent*), reititetyllä (*routed*) vai yksikätisellä (*one-armed*) tavalla. Näistä ensimmäisessä laite toimii L2-kerroksella ja IPv6-liikenne voidaan sillata sen läpi *ethertype*-pääsynhallintalistalla *0x86DD* (IPv6). Jos kyseessä on liikennepolulla oleva reitittävä laite, täytyy IPv6-liikenne reitittää sen ohi joko omaan fyysiseen tai loogiseen L3-porttiin. Jälkimmäisessä vaihtoehdossa palvelimen verkkoliitäntä konfiguroidaan *trunk*-liitännäksi ja IPv4- ja IPv6-liikenne välitetään sille saman fyysisen verkkoliitännän kautta omilla VLAN-tunnisteillaan. Tämä tehdään tässä työssä myöhemmin, kun IPv6-testiverkkoon asennetaan testipalvelin. Jos taas kyseessä on yksikätinen liikennepolun ulkopuolinen laite, ei IPv6-liikenteen sallimiseksi sen läpi tarvitse tehdä mitään. Eri vaihtoehdot on esitetty kuvassa 40. [35]

Konesaliverkon runkokerroksen läpi kulkee eniten liikennettä, joten se halutaan yleensä pitää niin yksinkertaisena kuin vain mahdollista. Runkokerroksella IPv6-käyttöönotto rajoittuukin usein vain tarvittavien portti- ja reititysprotokollakonfiguraatioiden tekemiseen. [35]



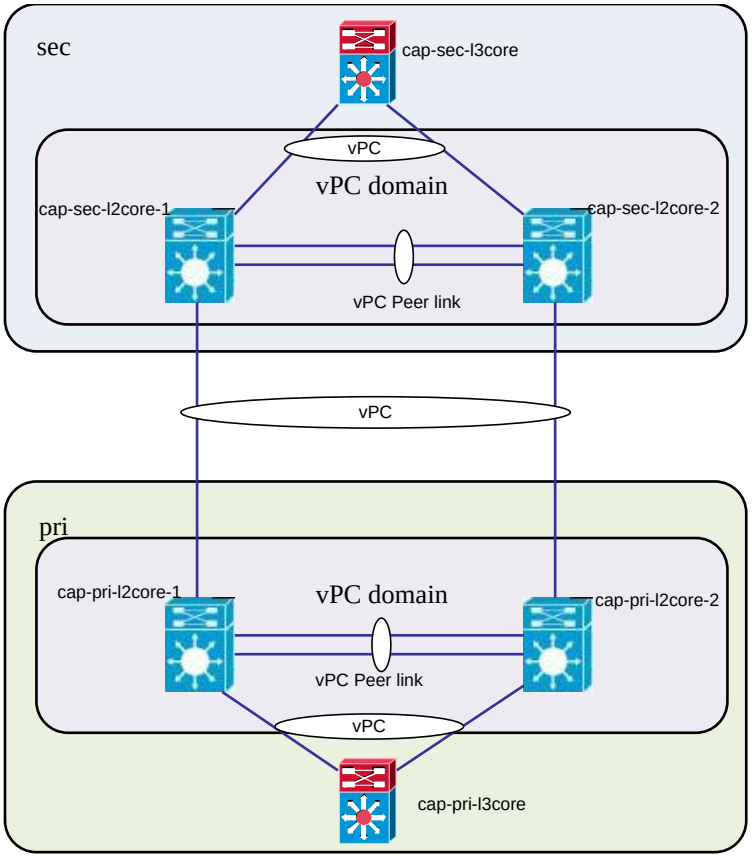
Kuva 40: IPv6-liikenteen salliminen aggregointikerroksen läpi.

5.2 Capgeminin konesaliverkko

Capgeminin konesaliverkko on toteutettu pääosin Ciscon reitittimillä, kytkimillä ja reitittävillä kytkimillä sekä Juniperin palomuureilla. Konesaliverkko rakentuu kahdesta Ciscon L3-runkokytkimestä (yksi kummassakin konesalissa), neljästä Ciscon L2-runkokytkimestä (kaksi kummassakin konesalissa), kuudesta Ciscon pääsykytkimestä (neljä aktiivisessa ja kaksi passiivisessa konesalissa) ja 20 Ciscon FEX-laitteesta (*Fabric Extender*). Ciscon pääsykytkimet ja FEX-laitteet muodostavat konesaliverkon pääsykerroksen. VMwaren ESX/ESXi-palvelimien virtuaalikytkimen tapauksessa pääsykerroksen raja hämärtyy, mutta useimmiten pääsykerroksen muodostavat Ciscon pääsykytkimet ja niihin kytketyt FEX-laitteet. Capgeminin konesaliverkossa aggregointi- ja runkokerrokset on yhdistetty eli pääsykerroksen kytkimien *uplink*-liitännät on kytketty L2-runkokytkimiin, joista puolestaan on musta kuituyhteys toiseen konesaliin. Reititys voidaan ajatella konesaliverkon palveluna, ja sen tarjoavat reitittävät L3-runkokytkimet on liitetty *router on a stick* -tyyppisesti L2-runkokytkimiin.

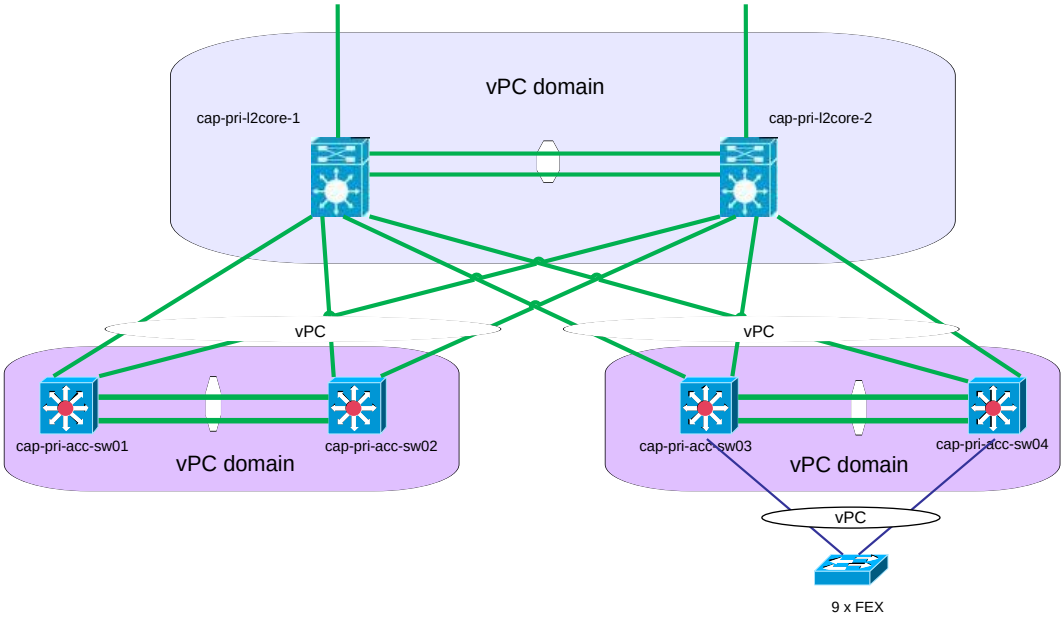
5.2.1 L2

Yleiskuva Capgeminin konesaliverkosta L2-kerroksella on esitetty kuvassa 41. Se koostuu yhteensä neljästä L2-runkokytkimestä ja kahdesta L3-runkokytkimestä.

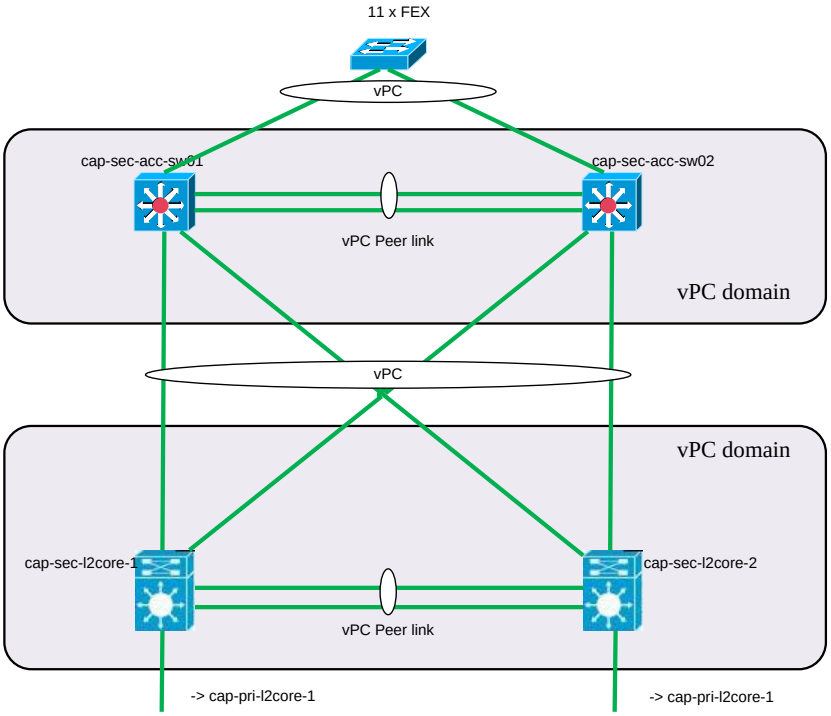


Kuva 41: Capgeminin konesaliverkko, yleiskuva.

Aktiivinen konesali on esitetty kuvassa 42 ja passiivinen konesali kuvassa 43. Aktiivisessa konesalissa on neljä pääsykytkintä, joihin on kytketty yhdeksän FEX-laitetta ja passiivisessa konesalissa kaksi pääsykytkintä, joihin on kytketty 11 FEX-laitetta.



Kuva 42: Capgeminin konesaliverkko, aktiivinen konesali.

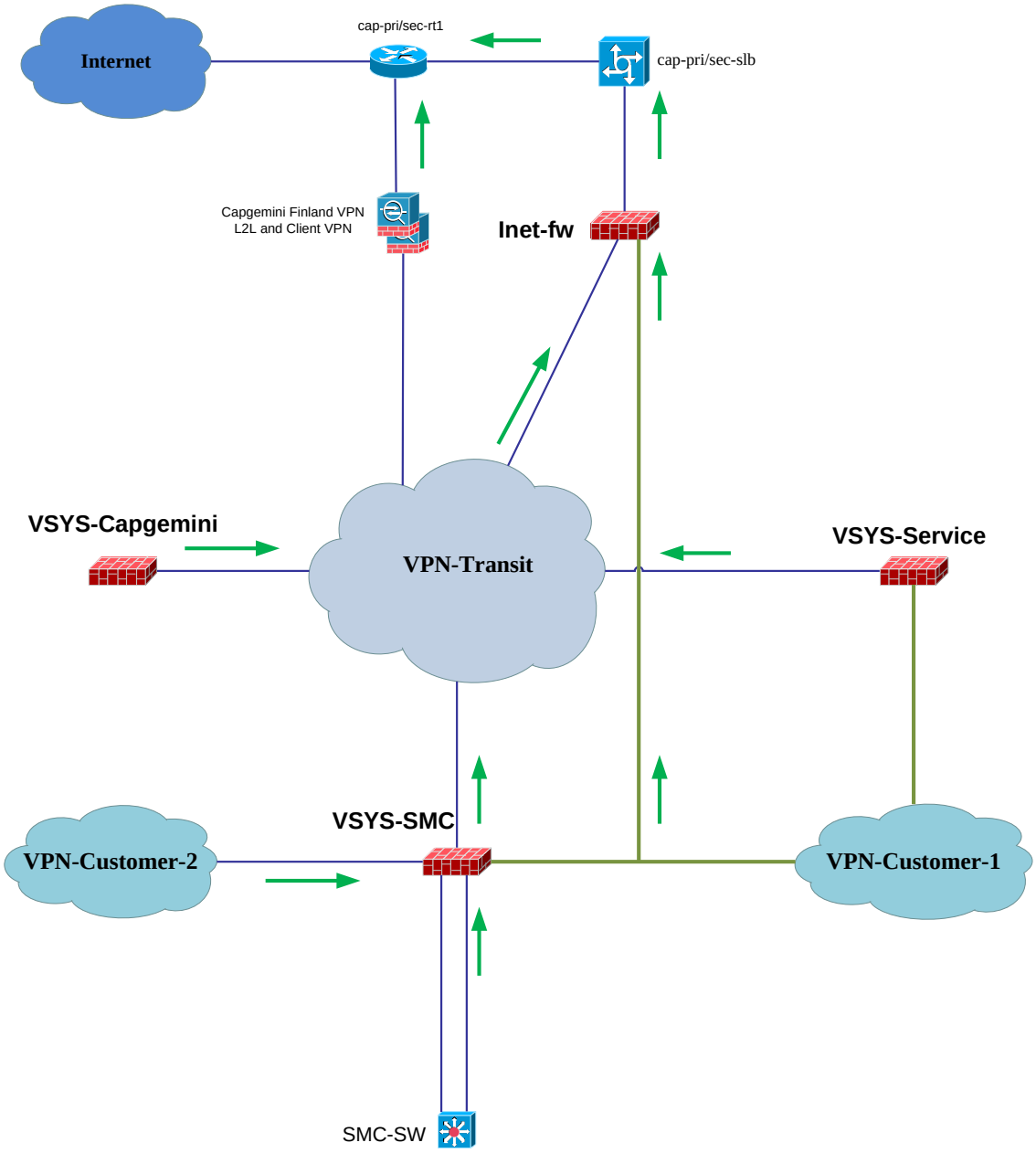


Kuva 43: Capgeminin konesaliverkko, passiivinen konesali.

Capgeminin konesaliverkko on siis kahteen konesaliin kahdennettu verkko, jossa toinen konesali on reitityskerroksella aktiivinen ja toinen passiivinen. Konesalien lyhenteet esiintyvät laitenimissä, ja tietoturvasyistä aktiivinen konesali on tässä työssä lyhennetty *pri* ja passiivinen konesali *sec*. Konesalit on yhdistetty toisiinsa kahdella mustalla kuidulla, joista on lohkottu 1290-1610 nm aallonpituuksia CWDM-tekniikalla (*Coarse Wavelength Division Multiplexing*).

5.2.2 L3

Yleiskuva Capgeminin konesaliverkosta reitityskerroksella on esitetty kuvassa 44.

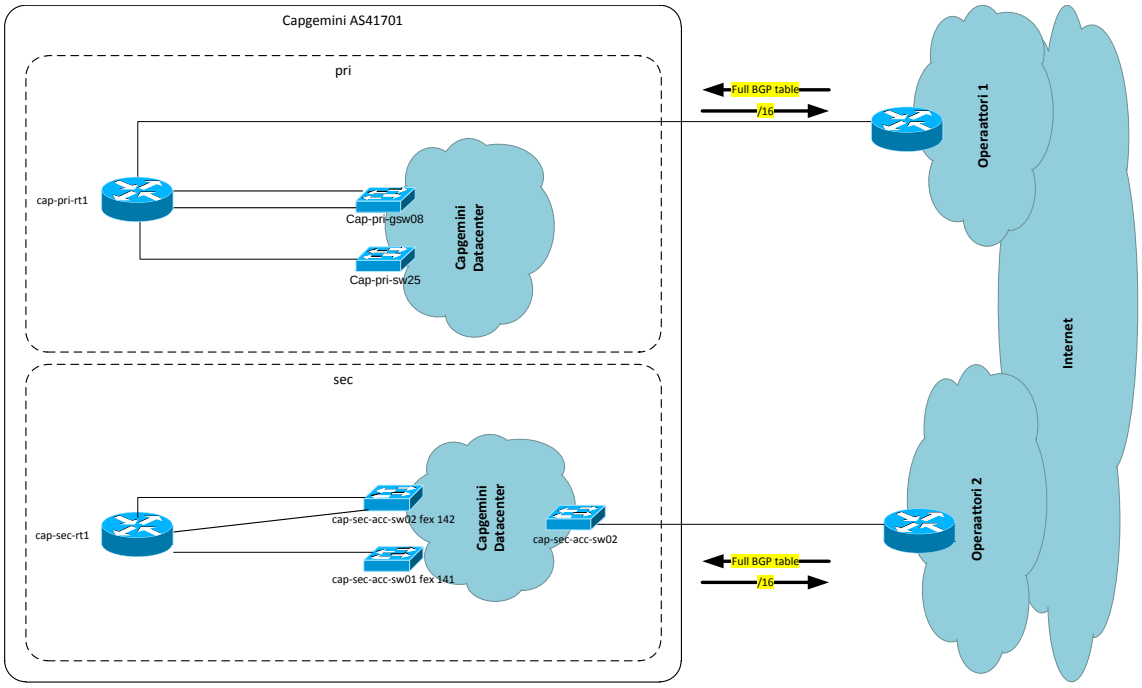


Kuva 44: Capgeminin konesaliverkko, L3.

Kuvan 44 reitittimet cap-pri/sec-rt1 ovat Capgeminin Internet-reitittimet ja kuvassa näkyvät VRF:t (*Virtual Routing and Forwarding*) VPN-Transit, VPN-Customer-1 ja VPN-Customer-2 ovat kaikki cap-pri/sec-l3core-kytkimien virtuaalireitittimiä. Kuten johdannossa kerrottiin, sekä VPN-etäyhteyspalvelu että kuormanjakaja (cap-pri/sec-slb) ovat tämän työn laajuuden ulkopuolella, mutta ne on merkitty kuvaan 44 täydellisyyden vuoksi. DNS-nimipalvelimet on jaettu kahteen eri segmenttiin: *autoratiiviset* nimipalvelimet ovat DMZ-alueella (*Demilitarized Zone*) Internet-palomuurin takana ja nimikyselyitä tekevät *resolver*-palvelimet VSYS-Service-palomuurin takana. Palomuurit VSYS-Capgemini, VSYS-Service ja VSYS-SMC ovat kaikki virtuaalipalomuureja (*Virtual System*) Juniperin ISG-alustalla (*Integrated Security Gateway*). Internet-palomuuri Inet-fw sen sijaan on oma fyysinen Juniperin SSG-palomuurinsa (*Secure Services Gateway*). Molemmat SceenOS-käyttöjärjestelmällä varustetut ISG/SSG-palomuurit on tulevaisuudessa tarkoitus korvata Junos-käyttöjärjestelmällisillä Juniperin SRX-palomuureilla, joten tässä työssä keskitytään SRX-palomuurien ominaisuuksiin ja sen Junos-käyttöjärjestelmän konfiguroimiseen.

5.2.3 Internet-liityntä

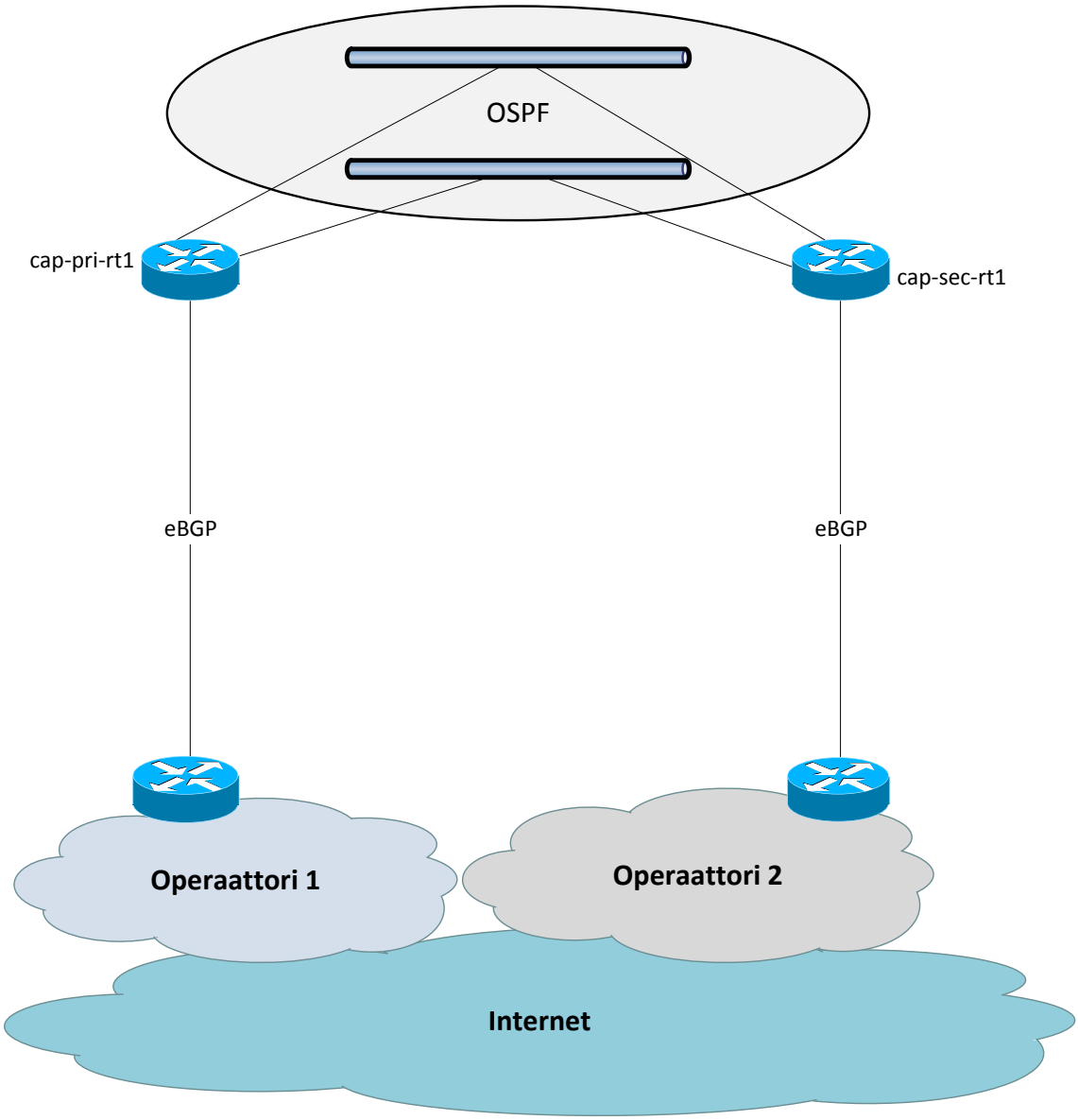
Capgemini on liittynyt IPv4-protokollalla Internetiin kahden palveluntarjojan kautta. Toinen liityntä on toteutettu aktiivisesta konesalista ja toinen passiivisesta konesalista. Internet-reitittimet on kytketty pääsykerroksen kytkimiin kuvan 45 mukaisesti.



Kuva 45: Capgeminin Internet-liityntä, L2.

Reitityskerroksella Capgeminilla on IPv4-BGP-naapuruus kahden eri Internet-palveluntarjoajan kanssa ja se vastaanottaa näiltä BGP-protokollalla koko Inter-

netin reititystaulun. Vastaavasti Capgemini mainostaa palveluntarjoajien suuntaan koko RIPE:n sille allokoimaa julkista /16-osoiteavaruutta. Internet-reitittimillä on Capgeminin eri asiakkaille määriteltyjä QoS- ja CoS-parametreja, mutta niiden esittely ja käsittely on tämän työn laajuuden ulkopuolella. Internet-liityntä reitityskerroksella on havainnollistettu kuvassa 46.



Kuva 46: Capgeminin Internet-liityntä, L3.

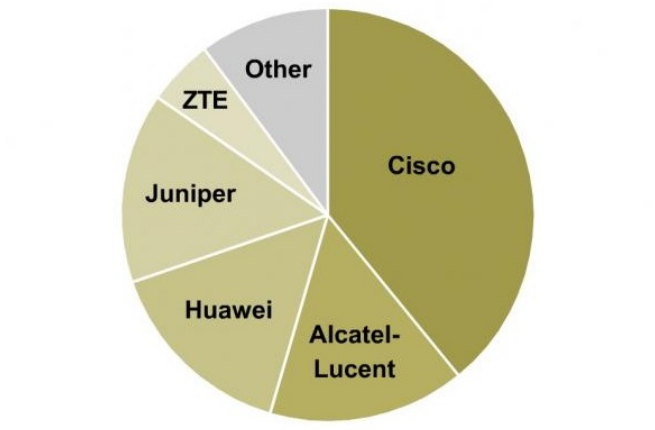
5.2.4 DNS-nimipalvelu

Capgeminin tarjoama nimipalvelualusta perustuu ISC:n (*Internet Systems Consortium*) BIND-ohjelmistoon tietyin poikkeuksin. Aiemmin nimipalvelua tarjottiin puhtaasti neljällä BIND-ohjelmistoon perustuvalla nimipalvelimella. Sittemmin syntyi tarve jakaa Capgeminin verkkoinfrastruktuuri kahtia. Infrastruktuurit on joudut-

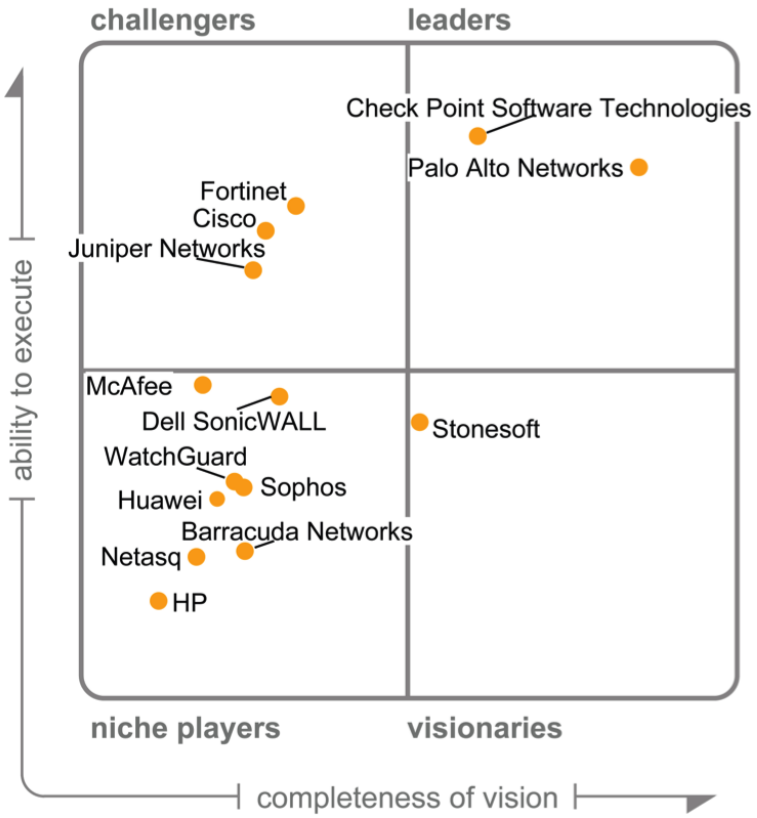
tu eriyttämään toisistaan asiakastarpeiden perusteella joko fyysisesti tai loogisesti, mutta ne toimivat tämän työn kannalta toisiaan vastaavina palvelualustoina, eikä niitä tässä työssä siksi sen enempää erotella tai käydä läpi. Mainittakoon kuitenkin, että samalla kun verkkoinfrastruktuurit eriytettiin, otettiin nimipalvelun automatisoinnin vuoksi käyttöön EfficientIP:n SOLIDserver-ohjelmisto [149]. Samalla rakennettiin kumpaankin ympäristöön ns. *stealth*-nimipalveluarkkitehtuuri, jossa nimipalvelun tarjoamiseen käytetään kolmea palvelinta. Yksi näistä on ns. *hidden master* -palvelin, toinen *pseudo master* -palvelin ja kolmas *slave*-palvelin. BIND-konfiguraatiossa *hidden master* -palvelin on isäntäpalvelin ja sekä *pseudo master*- että *slave*-palvelimet orjapalvelimia, joten *pseudo master*- ja *slave*-palvelimien DNS-tietueet päivittyvät *hidden master* -palvelimelta alueen siirtotoiminnallisuudella (*zone transfer*) [150]. *Hidden master* -palvelin taas päivittyy automaattisesti SOLIDserver-palvelimelta, jota käytetään myös IP-osoitteiden hallinnointiin. SOLIDserver-tuotteen IPAM- ja DNS-moduulit (*IP Address Management*) on liitetty toisiinsa, ja IP-osoitetta varatessa se saadaan automaattisesti suoraan nimipalveluun. Tämä poistaa manuaalisen nimipalvelinten konfiguroinnin tarpeen, mutta sen perusasiat käydään seuraavassa luvussa joka tapauksessa läpi.

5.3 IPv6-konfigurointi

Cisco on selvä markkinajohtaja sekä L2/L3-kytkin- että reititinmarkkinoilla, kuten kuvat 6 ja 47 osoittavat. Juniper Networks puolestaan on viiden johtavan palomuurien valmistajan joukossa, kuten kuva 48 osoittaa. Kuten aiemmin todettiin, Capgeminin konesaliverkon reitittimet ja kytkimet ovat pääosin Ciscon ja palomuurit Juniperin laitteita, joten tässä luvussa käydään läpi Ciscon IOS- ja NX-OS- sekä Juniperin Junos-käyttöjärjestelmien tärkeimmät IPv6-protokollaan liittyvät komennot. NX-OS on Ciscon Nexus-kytkimissä käytettävä käyttöjärjestelmä, ja sen konfigurointi vastaa pitkälti Catalyst-kytkimien IOS-käyttöjärjestelmän konfiguroimista.



Kuva 47: Liikevaihdoltaan suurimmat reititin- ja kytkinvalmistajat Q1/2013. [151]



Kuva 48: Gartnerin nelikenttä palomuurivalmistajista. [152]

5.3.1 Cisco IOS & NX-OS

Ciscon reitittimen IPv6-konfigurointi alkaa ottamalla IPv6-reititys käyttöön. RA-viestien lähetys on automaattisesti enabloitu Ethernet- ja FDDI-rajapinnoilla, jos globaali *ipv6 unicast-routing* -komento on annettu. Tällöin reititin lähettää RA-viestejä, jotka sisältävät rajapinnan globaalin IPv6-osoitteen verkkoprefiksin. [153]

```
Router(config)# ipv6 unicast-routing
```

Seuraavaksi otetaan IPv6-CEF-toiminnallisuus (*Cisco Express Forwarding*) käyttöön. Se mahdollistaa liikenteen kytkemisen ja reitityksen ASIC-moduulissa (*Application Specific Integrated Circuit*) reitittimen prosessorin sijaan: [154]

```
Router(config)# ipv6 cef
```

Tyypin 0 reititysotsikot voidaan ottaa pois käytöstä antamalla ao. komento: [154]

```
Router(config)# no ipv6 source-route
```

Staattisen IPv6-reitin konfiguroinnissa voidaan käyttää CIDR-notaatiota: [154]

```
Router(config)# ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address |
interface-type interface-number [ipv6-address]} [nexthop-vrf [vrf-name1 | default]]
[ administrative-distance ] [administrative-multicast-distance | unicast | multicast]
[ next-hop-address ] [tag tag] [name name]
```

CIDR-notaatiota voidaan käyttää myös IPv6-pääsy ja *prefix*-listoissa. *Standard*-tyyppiset pääsylistat eivät ole enää tuettuja, vaan IPv6-pääsylistan täytyy määrittää sekä lähde- että kohdeosoitteet. [154] IOS-käyttöjärjestelmän *IPv6 ACL Extensions for IPsec Authentication Header* -toiminnallisuus mahdollistaa ylempien kerrosten protokollien suodattamisen pääsylistoissa myös silloin, kun käytetään IPsec-autentikointiotsikkoa [155].

```
Router(config)# ipv6 access-list access-list-name
Router(config-ipv6-acl)# deny | permit protocol {source-ipv6-prefix/prefix-length |
any | host source-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/
prefix-length | any | host destination-ipv6-address} [operator [port-number]]
[dscp value] [fragments] [log] [log-input] [sequence value] [time-range name]

Router(config)# ipv6 prefix-list list-name [seq seq-number] {deny ipv6-prefix/
prefix-length | permit ipv6-prefix/prefix-length | description text} [ge ge-value]
[le le-value]
```

IPv6-pääsylistassa on yksi perustavanlaatuinen ero verrattuna IPv4-pääsylistaan, joka liittyy pääsylistan lopussa oleviin implisiittisiin riveihin: jokaisessa Ciscon IPv4-pääsylistassa on lopussa implisiittinen kaiken liikenteen kieltävävä rivi *deny ip any any*, joten on luonnollista, että myös IPv6-pääsylistan lopussa on vastaava implisiittinen rivi. IPv6-pääsylistassa on kuitenkin sen lisäksi kaksi muutakin implisiittistä riviä: [156]

```
permit icmp any any nd-na
permit icmp any any nd-ns
deny ipv6 any any
```

Nämä rivit sallivat ND-protokollan toiminnan, joten jos IPv6-pääsylistaan konfiguroidaan viimeiseksi riviksi eksplisiittinen *deny ipv6 any any* -rivi, täytyy pääsylistaan konfiguroida myös eksplisiittiset ND-protokollan NA- ja NS-viestit sallivat rivit. Näiden implisiittisten rivien olemassaolo huomattiin IPv6-käyttöönotossa myös käytännössä: IPv6-pääsylistaan konfiguroitiin viimeiseksi eksplisiittinen kaiken kieltävä rivi, joka esti täten myös NA- ja NS-viestit eikä ND-protokolla toiminut oikein. Pääsylistan asettamiseen rajapinnalle ei enää käytetä *access-group*-komentoa, vaan IPv6-osoite ja pääsylista konfiguroidaan rajapinnalle seuraavasti: [154]

```
Router(config)# interface GigabitEthernet0/1
Router(config-if)# ipv6 enable
Router(config-if)# ipv6 address {ipv6-prefix/prefix-length | prefix-name sub-bits/
prefix-length}
Router(config-if)# ipv6 traffic-filter access-list-name {in | out}
```

Etähallintaan käytettäville VTY-linjoille (*Virtual Terminal Line*) pääsylista asetetaan *access-class* -komennon sijaan *ipv6 access-class* -komennolla. BGP-konfiguraatio on muuten vastaava kuin IPv4:llä, mutta käytetään IPv6-osoiteperhettä: [154]

```
Router(config)# router bgp as-number
Router(config-router)# address-family ipv6 [unicast | multicast | vpnv6]
[vrf vrf-name]
```

5.3.2 Juniper Junos

Junos on Juniperin käyttöjärjestelmä, joka on suunniteltu yhteiseksi käyttöjärjestelmäksi niin Juniperin palomuureihin, reitittimiin kuin kytkimiinkin. Forrester Consultingin tekemän tutkimuksen mukaan yritys voi vähentää verkon ylläpitoon kuluvia operatiivisia kustannuksia jopa 42% ottamalla käyttöön Junos-käyttöjärjestelmän. [157] Tässä työssä ei oteta kantaa siihen, pitääkö luku paikkaansa, mutta vaikka tässä työssä keskitytäänkin ensisijaisesti Juniperin palomuureihin, on hyvä pitää mielessä, että seuraavat komennot pätevät ainakin suurimmilta osin myös Junos-käyttöjärjestelmällisiin reitittimiin ja reitittäviin kytkimiin. IPv6-konfigurointi aloitetaan yleensä asettamalla palomuurin *loopback*-osoite, jota käytetään mm. monessa dynaamisessa reititysprotokollassa. [158]

1. Mennään konfiguraatiotilaan:

```
admin@fw> configure
Entering configuration mode
```

2. Mennään konfiguroitavan rajapinnan alle:

```
[edit]
admin@fw# edit interfaces lo0 unit 0
```

3. Asetetaan palomuurille 128-bittinen IPv6-osoite:

```
[edit interfaces lo0 unit 0]
admin@fw# set family inet6 address 2001:db8::1/128
```

Palomuurilla on nyt IPv6-loopback-osoite. Seuraavaksi asetetaan osoite tuotantorajapinnalle. [158]

1. Mennään konfiguroitavan rajapinnan alle:

```
[edit]
admin@fw# edit interfaces ge-1/0/1 unit 0
```

2. Asetetaan rajapinnalle IPv6-osoite:

```
[edit interfaces ge-1/0/1 unit 0]
admin@fw# set family inet6 address 2001:db8:0:1::/64 eui-64
```

Staattinen IPv6-reitti asetetaan seuraavasti: [158]

```
[edit routing-options rib inet6.0]
admin@fw# set static route 0::0/0 next-hop 2001:db8:0:1::1
```

Lisäksi täytyy vielä ottaa käyttöön liikennevuohon perustuva IPv6-liikenteen välittäminen, jotta palomuuri välittää IPv6-liikennettä: [159]

```
[edit security forwarding-options family inet6]
admin@fw# set mode flow-based
```

Tallennetaan lopuksi tehdyt muutokset: [158]

```
[edit security forwarding-options family inet6]
admin@fw# commit
commit complete
```

5.3.3 ISC BIND

Tässä luvussa oletetaan, että lukijalla on perustietämys Internetin nimipalvelusta eli DNS:stä ja ISC:n BIND-nimipalveluohjelmistosta, joka on avoimeen lähdekoodiin perustuva, Internetissä laajimmin käytössä oleva nimipalvelusovellus [160]. Vapaasti saatavilla oleva kirja *DNS for Rocket Scientists* kannattaa ensin ainakin silmäillä läpi, jos näin ei ole [150].

A vs. AAAA

IETF julkaisi alkuperäiset DNS:n laajennukset IPv6:ta varten vuonna 1995 ja päivitetyn version vuonna 2003. Niissä määriteltiin uusi AAAA-tietue IPv6-osoitteita varten ja ip6.int-verkkotunnus käänteisnimihakuja varten. Mainittakoon, että myös uudet A6- ja DNAME-tietueet esitellyt ratkaisu oli olemassa, mutta koska se olisi vaatinut tuohon aikaan laajalti käytetyn BIND4-nimipalveluohjelmiston uudistamista, ovat AAAA-tietueet edelleen käytössä tänä päivänä. ip6.int-verkkotunnus on kuitenkin korvattu ip6.arpa-verkkotunnuksella. AAAA-nimi ei ole sattumaa: 128-bittinen IPv6-osoite on neljä kertaa 32-bittistä IPv4-osoitetta pidempi, joten lienee loogista, että myös DNS-tietueen nimi on neljä kertaa pidempi. Suomen Googlen (`http://www.google.fi/`) IPv6-osoite on ainakin tätä kirjoitettaessa 2a00:1450:4010:c03::5e, joka määriteltäisiin google.fi-zonetiedostossa näin: [161, 162, 163]

```
www.google.fi. IN  AAAA 2a00:1450:4010:c03::5e
```

AAAA-tietueita voidaan lisätä A-tietueiden rinnalle, kuten Google on tehnyt:

```
>nslookup www.google.fi
...
Non-authoritative answer:
Name:    www.google.fi
Addresses:  2a00:1450:4010:c03::5e
          74.125.143.94

www.google.fi. IN  A    74.125.143.94
www.google.fi. IN  AAAA 2a00:1450:4010:c03::5e
```

Näin tehtäessä täytyy tosin olla varovainen, sillä monet käyttöjärjestelmät yrittävät ottaa ensin yhteyttä IPv6-osoitteeseen. Jos päätelaitteesta puuttuu IPv6-tuki tai sillä on esim. vain linkkilokaali IPv6-osoite, täytyy sen ensin odottaa IPv6-yhteyden aikakatkaisua, ennen kuin se ottaa yhteyttä IPv4-osoitteeseen. Vaihtoehtoisina ratkaisuina ehdotetaan esim. IPv6-osoitteeseen liitettyyn nimeen -v6- tai .v6-päätteen lisäämistä. [163]

in-addr.arpa vs. ip6.arpa

AAAA-tietueissa on siis sallittua lyhentää osoitteita, mutta PTR-tietueissa näin ei ole. Googlen IPv6-osoite 2a00:1450:4010:c03::5e on auki kirjoitettuna 2a00:1450:4010:c03:0:0:0:5e (tai 2a00:1450:4010:0c03:0000:0000:0000:005e), joten Googlen PTR-tietue kirjoitettaisiin siis näin: [163]

```
e.5.0.0.0.0.0.0.0.0.0.0.0.0.0.3.0.c.0.0.1.0.4.0.5.4.1.0.0.a.2.ip6.arpa. IN PTR
www.google.fi.
```

Tätä PTR-tietuetta ei luonnollisestikaan voida lisätä nykyiseen 94.143.125.74-in-addr.arpan sisältävään alueeseen, vaan sitä varten on luotava uusi, ip6.arpa-päätteinen alue, esim. 3.0.c.0.0.1.0.4.0.5.4.1.0.0.a.2.ip6.arpa tai kuten Google on tehnyt, 0.5.4.1.0.0.a.2.ip6.arpa. Googlen PTR-tietue lisättäisiin ensimmäiseen, 3.0.c.0.0.1.0.4.0.5.4.1.0.0.a.2.ip6.arpa-alueeseen seuraavasti: [163]

```
e.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0 PTR www.google.fi.
```

Vastaavasti 0.5.4.1.0.0.a.2.ip6.arpa-alueeseen se lisättäisiin seuraavasti: [163]

```
e.5.0.0.0.0.0.0.0.0.0.0.0.0.0.3.0.c.0.0.1.0.4 PTR www.google.fi.
```

$ORIGIN-lausetta voidaan käyttää samoin kuin IPv4-tietueissa, eli voitaisiin kirjoittaa myös: [163]

```
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.3.0.c.0.0.1.0.4.0.5.4.1.0.0.a.2.ip6.arpa.
e.5 PTR www.google.fi.
```

Alueet lisätään lopuksi nimipalvelimien *named.conf*-tiedostoon *zone*-lauseella: [163]

```
zone "3.0.c.0.0.1.0.4.0.5.4.1.0.0.a.2.ip6.arpa"
    type master; // primääri nimipalvelin
    file "db.2a00:1450:4010:c03";
;
```

tai

```
zone "0.5.4.1.0.0.a.2.ip6.arpa"
    type master; // primääri nimipalvelin
    file "db.2a00:1450";
;
```

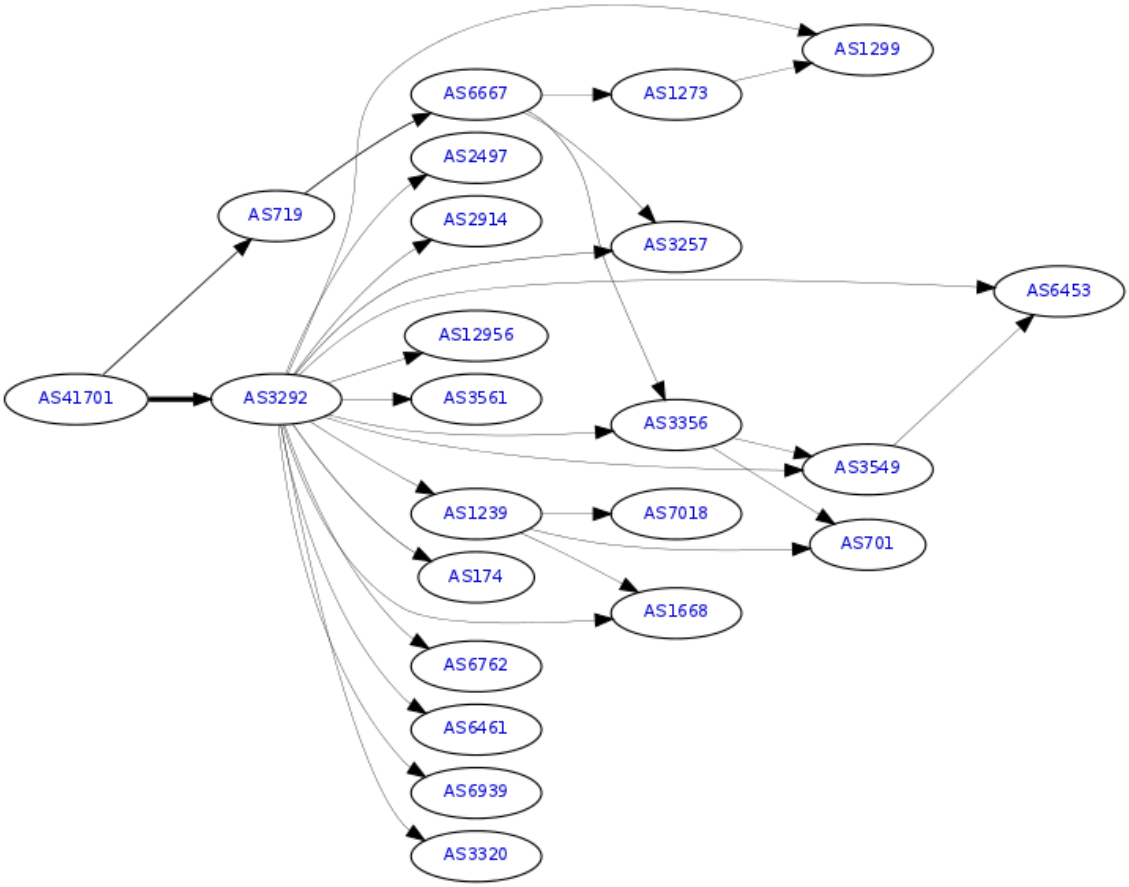
Tässä luvussa näytettiin ensin, kuinka tyypillinen palvelinkeskuksen konesaliverkko rakentuu ja kuinka IPv6-protokolla sinne sijoittuu. Sen jälkeen esiteltiin Capgeminin konesaliverkko tämän diplomityön kannalta oleellisilta osin ja näytettiin IPv6-konfigurointiesimerkkejä Capgeminin konesaliverkossa käytössä olevissa Ciscon IOS- ja NX-OS- ja Juniperin Junos-käyttöjärjestelmissä sekä ISC:n BIND-ohjelmistossa. Seuraavassa luvussa tehdään näiden esimerkkien, aiempien lukujen ja parhaiden käytäntöjen pohjalta IPv6-käyttöönotto Capgeminin konesaliverkossa.

6 IPv6-käyttöönotto

Tässä luvussa tehdään IPv6-käyttöönotto Capgeminin konesaliverkossa. IPv6-käyttöönottoon on oikeastaan kaksi eri vaihtoehtoa: ns. *inside-out*- ja *outside-in*-mallit. Ensimmäisessä IPv6-protokolla otetaan ensin käyttöön yrityksen sisäverkossa, mutta Internetiin ja muihin ulkoisiin kohteisiin liikennöidään yhä IPv4-protokollalla. Jälkimmäisessä vaihtoehdossa IPv6-käyttöönotto aloitetaan Internetin reunalta, eli yrityksen verkko yhdistetään Internetiin IPv6-protokollalla. Tämä on tyypillinen valinta mille tahansa palveluntarjoajalle, joka haluaa tarjota palveluitaan IPv6-protokollalla ja se oli myös Capgeminin valinta IPv6-käyttöönottoon. Seuraavaksi yhdistetäänkin Capgeminin konesaliverkko Internetiin IPv6-protokollalla ja rakennetaan sen jälkeen IPv6-testiverkko Capgeminin laboratorioon.

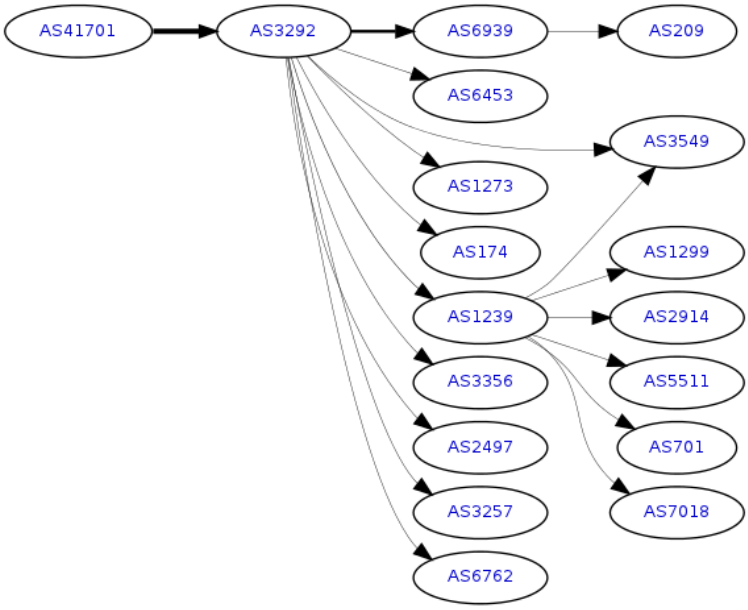
6.1 Internet-liityntä

Kuten aiemmin kerrottiin, Capgemini (AS41701) on liittynyt Internetiin IPv4-protokollalla kahden Internet-palveluntarjoajan kautta. Tämä on havainnollistettu kuvassa 49, jossa on kuvattu myös Capgeminin käyttämien tier 3 -operaattoreiden (AS719 ja AS3292) IPv4-naapuruuksia edelleen tier 2 - ja tier 1 -operaattoreiden kanssa.



Kuva 49: Capgeminin (AS41701) Internet-liityntä, IPv4. [164]

Luonnollinen valinta oli tehdä myös IPv6-liityntä Internetiin näiden palveluntarjoajien kautta. IPv6-käyttöönotto oli tarkoitus tehdä huhtikuun kuukausittaisella huoltokatkolla, mutta Internet-reitittimien ohjelmistoversio oli liian vanha eikä siinä ollut tukea BGP:n IPv6-osoiteperheelle. Reitittimet päivitettiin toukokuun huoltokatkolla, ja 22.5.2013 saatiin otettua käyttöön ensimmäisen operaattorin BGP-naapuruudessa myös IPv6-osoitteet. Capgeminin (AS41701) konesaliverkko on saavutettavissa Internetistä nyt myös IPv6-protokollalla, kuten kuva 50 osoittaa.



Kuva 50: Capgeminin (AS41701) Internet-liityntä, IPv6. [165]

Toisen operaattorin BGP-naapuruudessa IPv6-osoitteet on tarkoitus ottaa käyttöön myöhemmin. Alla on esitettynä Internet-reitittimien komennot, joilla IPv6-osoitteet otettiin käyttöön ensimmäisen operaattorin BGP-naapuruudessa.

```
cap-pri-rt1
no ipv6 source-route
ipv6 unicast-routing
ipv6 cef

ipv6 route .../48 Null0
ipv6 prefix-list cap-ipv6-in
...
ipv6 prefix-list cap-ipv6-out
...
ipv6 access-list IPV6VTY
...
line vty 0 4
 ipv6 access-class IPV6VTY in
line vty 5 15
 ipv6 access-class IPV6VTY in
```

```
interface GigabitEthernet0/0.478
 ipv6 enable
 no ipv6 redirects
 ipv6 address ...
!

router bgp 41701
 neighbor ... remote-as 41701
 !
 address-family ipv6
  redistribute static
  neighbor ... activate
  neighbor ... next-hop-self
 exit-address-family
!
```

cap-sec-rt1

```
ipv6 unicast-routing
no ipv6 source-route
ipv6 cef

ipv6 route .../48 Null0
ipv6 prefix-list cap-ipv6-in
...
ipv6 prefix-list cap-ipv6-out
...
ipv6 access-list IPV6VTY
...

line vty 0 4
 ipv6 access-class IPV6VTY in
line vty 5 15
 ipv6 access-class IPV6VTY in

ipv6 access-list CAP_IPV6_IN
...
ipv6 access-list CAP_IPV6_OUT
...

interface GigabitEthernet0/0.478
 ipv6 address ...
 ipv6 enable
 no ipv6 redirects
!
interface GigabitEthernet0/0.9
 ipv6 address ...
 ipv6 enable
 no ipv6 redirects
 ipv6 verify unicast reverse-path
 ipv6 traffic-filter CAP_IPV6_IN in
 ipv6 traffic-filter CAP_IPV6_OUT out
!
router bgp 41701
 neighbor ... remote-as 41701
```

```
 neighbor ... remote-as 3292
 neighbor ... password ...

address-family ipv6
  redistribute static
  neighbor ... activate
  neighbor ... next-hop-self
  neighbor ... activate
  neighbor ... prefix-list cap-ipv6-in in
  neighbor ... prefix-list cap-ipv6-out out
```

Varmistetaan Capgeminin passiiviselta Internet-reitittimeltä, että sekä operaattorin PE-reitittimen että Capgeminin aktiivisen Internet-reitittimen IPv6-osoitteet vastaavat pingiin:

```
cap-sec-rt1#ping ...
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to ..., timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
cap-sec-rt1#ping ...
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to ..., timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

IPv6-BGP-naapuruuksien toiminta voidaan todentaa *sh ip bgp ipv6 unicast summary* -komennolla:

```
cap-pri-rt1:
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
...
                4        41701  628396  183706   524676    0    0 2w5d        12728
cap-sec-rt1:
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
...
                4        41701  183695  628354   720033    0    0 2w5d            2
...
                4         3292  665337  145161   720029    0    0 2w1d        12728
```

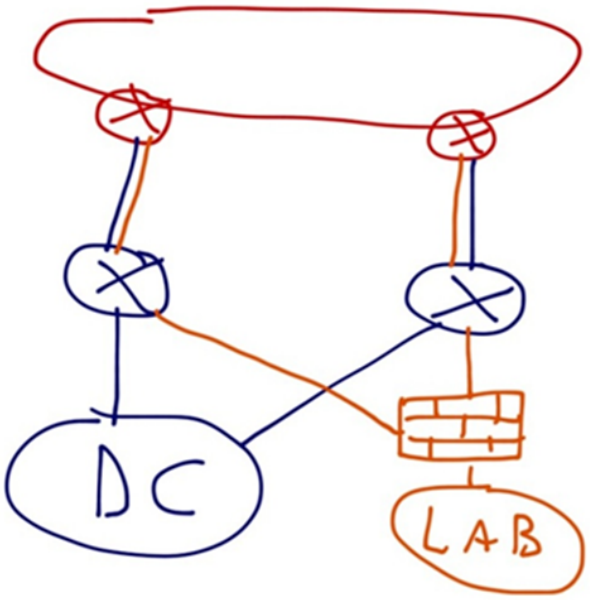
Aktiivisen konesalin reitittimellä näkyy ainoastaan iBGP-sessio passiivisen konesalin reitittimen kanssa, mutta passiivisen konesalin reitittimellä lisäksi eBGP-sessio operaattorin PE-reitittimen kanssa. Nähdään myös, että operaattori mainostaa 12728 IPv6-reittiä eli koko Internetin IPv6-reititystaulua [166]. Operaattorille puolestaan mainostetaan koko Capgeminin julkista /48-osoitealuetta:

```
sh ip bgp ipv6 unicast neighbors ... advertised-routes
     Network          Next Hop            Metric LocPrf Weight Path
 *>  .../48
                       ::                       0         32768 ?
```

Capgeminin Internet-reitittimet on saatu onnistuneesti vastaanottamaan Internetin IPv6-reititystaulu ja mainostamaan Capgeminin IPv6-avaruus Internetiin.

6.2 IPv6-testiverkko

Seuraavaksi Capgeminin laboratorioon rakennettiin IPv6-testiverkko, jotta IPv6-Internet-yhteyttä voitiin testata. Testiverkon tarkoituksena oli myös tutkia, kuinka verkko toimii ja käyttäytyy muilla kuin /64-prefiksillä. Laboratorioverkko (LAB) on kuvattu yleisellä tasolla kuvassa 51. Se sijaitsee omien, dedikoitujen laboratoriopalomuurien takana, jotta se ei vaaranna konesaliverkkoa (DC) millään tavalla.



Kuva 51: Laboratorioverkon sijoittuminen Capgeminin konesaliverkkoon.

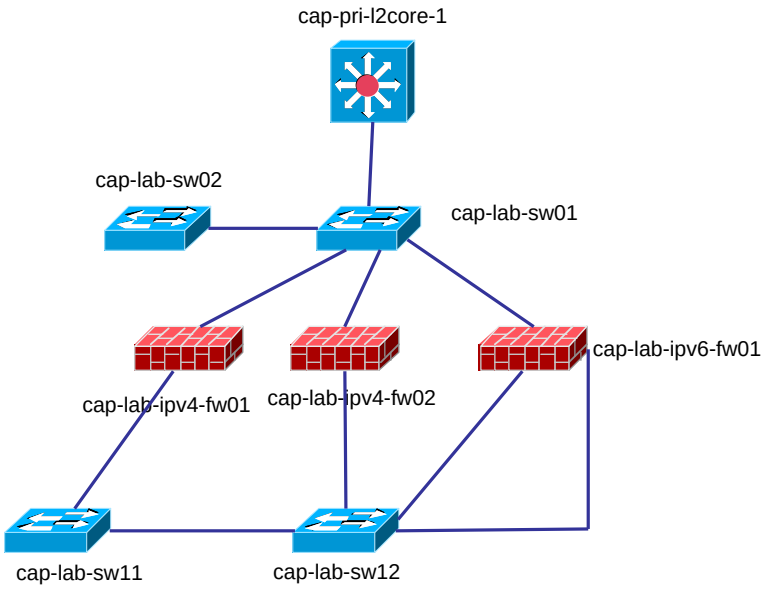
Kuvassa 51 punaisella on kuvattu molempien operaattorien PE-reitittimet, sinisellä nykyinen IPv4-liityntä ja oranssilla uusi, rakennettava IPv6-liityntä. Kun IPv4-verkossa otetaan käyttöön IPv6-protokolla, ensimmäinen kysymys joka nousee esiin on se, sekoitetaanko IPv4- ja IPv6-liikennettä verkossa vai kuljetetaanko ne omilla fyysisillä tai loogisilla linkeillään. Ensimmäinen vaihtoehto on yleensä parempi, koska tällöin tarvitaan vain puolet verkkokerroksen rajapinnoista ja mahdollisesti myös vähemmän fyysisiä portteja kuin jälkimmäisessä vaihtoehdossa. Capgeminin Internet-reitittimien rajapinnoille tehtiinkin kahden protokollapinon ratkaisu, eli samoille alirajapinnoille sekä Internetin että konesaliverkon suuntaan lisättiin IPv4-osoitteiden rinnalle IPv6-osoitteet. Ajatuksena oli ensin eriyttää IPv6 kokonaan IPv4:stä ja tehdä Internet-reitittimille omat virtuaaliset reititysinstanssit IPv6:ta varten, mutta tästä ajatuksesta luovuttiin yksinkertaisuuden vuoksi.

6.2.1 L2

Capgeminin laboratorion nykyistä IPv4-testiverkkoa ovat suojaamassa palomuurit cap-lab-ipv4-fw01 ja cap-lab-ipv4-fw02 kuvan 52 mukaisesti. Laboratorioverkon liitäntä konesaliverkkoon on tehty kytkimen cap-lab-sw01 kautta. Siihen on kytketty

sekä konesaliverkon runkokytkin cap-pri-l2core-1 että laboratoriopalomuurit cap-lab-ipv4-fw01 ja cap-lab-ipv4-fw02.

IPv6-laboratoriopalomuuriksi (cap-lab-ipv6-fw01) valittiin Ciscon palomuuri. Tämä valinta tehtiin lähinnä kahdesta syystä: kyseinen laite oli jäänyt asiakkaaltamme tarpeettomaksi ja sillä voidaan myöhemmin testata myös sekä LAN-to-LAN- että asiakas-VPN-yhteyksiä IPv6-protokollalla. Se kytkettiin IPv4-palomuurien rinnalle niin, että IPv6-liikenne kulkee ainoastaan sen läpi ja vastaavasti IPv4-liikenne ainoastaan sille tarkoitettujen palomuurien läpi. Tämä ratkaisu on kuvattu kuvassa 52.

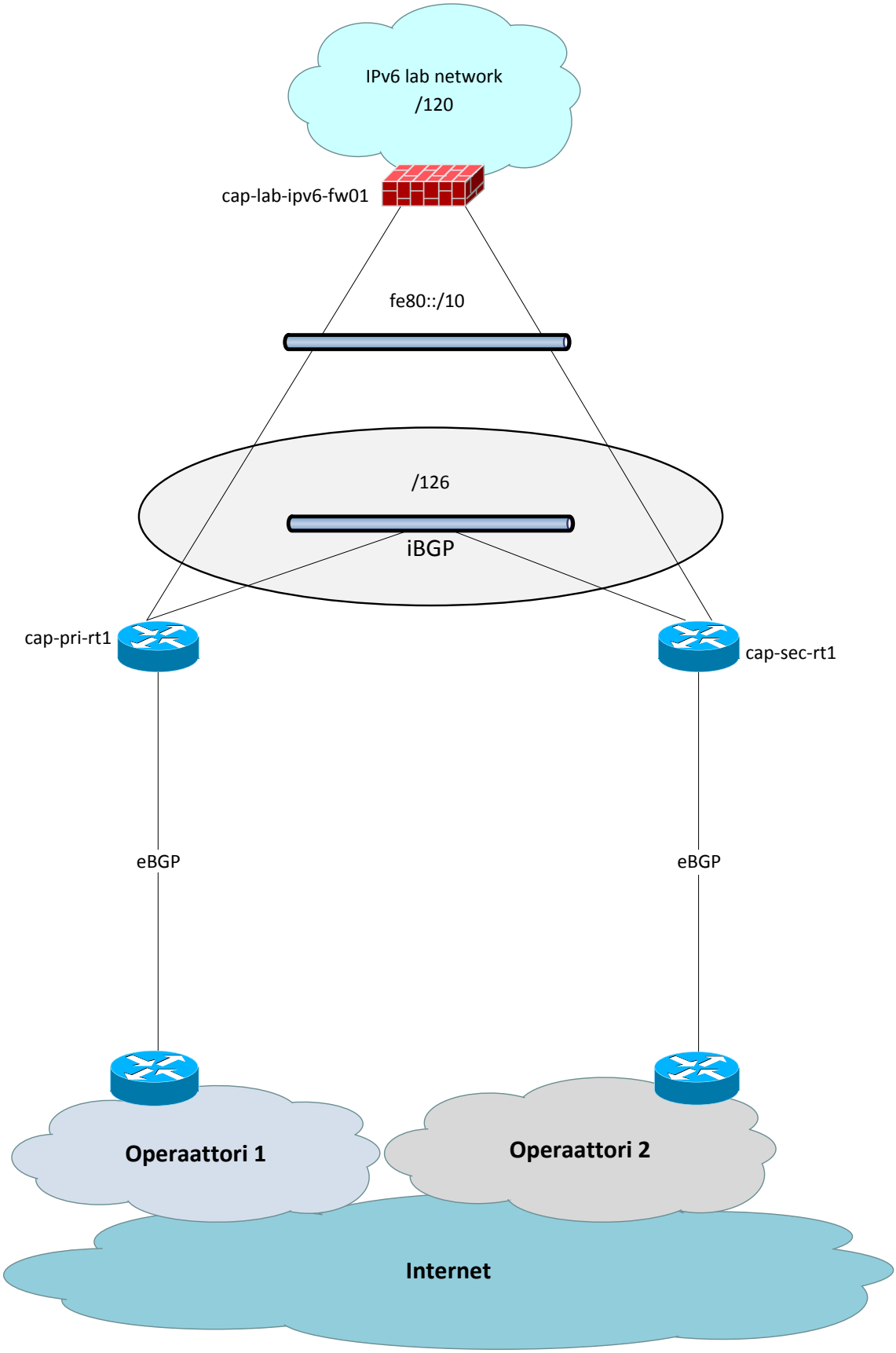


Kuva 52: Capgeminin laboratorioverkko, L2.

IPv6-laboratoriopalomuuri ei siis suinkaan ole fyysisesti kytketty Internet-reitittimiin, vaan kytkentä on tehty loogisesti konesaliverkon ja sen runkokytkimen cap-pri-l2core-1 kautta. Sekä Internet-reitittimien ja palomuurin väliselle linkkiverkolle että itse IPv6-testiverkolle allokoitiin oma virtuaalinen lähiverkkonsa (*VLAN*), jotta IPv6-liikenteen kulku pystytään tarkoin rajaamaan konesali- ja laboratorioverkoissa.

6.2.2 L3

Kuvassa 53 on kuvattu IPv6-testiverkko L3-kerroksella. Kuten kuvassa 51 esitettiin, laboratoriopalomuuri kytkettiin L3-kerroksella loogisesti suoraan kiinni Internet-reitittimiin käyttäen Vlan-tunnistetta 859. Itse IPv6-testiverkolle allokoitiin Vlan-tunniste 860. Operaattorien PE-reitittimien ja Capgeminin CE-reitittimien välisten linkkiverkkojen (Vlan9 ja Vlan30) osoitteet on allokoitu operaattoreiden IPv6-osoiteavaruudesta ja linkkiverkko (Vlan859) toteutettiin käyttäen linkkilokaaleja osoitteita Ciscon IOS-käyttöjärjestelmän rajoituksien vuoksi [167].



Kuva 53: IPv6-testiverkko, L3.

Alla on näytetty kuvan 53 esittelemää topologiaa varten tarvitut konfiguraatiot Internet-reitittimillä ja laboratoriopalomuurilla.

cap-pri-rt1

```
interface GigabitEthernet0/1.859
 description LAB-ASA-INETRT-link
 encapsulation dot1Q 859
 standby version 2
 standby 59 ipv6 autoconfig
 standby 59 timers msec 500 msec 1500
 standby 59 preempt delay minimum 60
 standby 59 authentication md5 key-string ...
 ipv6 enable
 no ipv6 redirects
!
```

cap-sec-rt1

```
interface GigabitEthernet0/1.859
 description LAB-ASA-INETRT-link
 encapsulation dot1Q 859
 standby version 2
 standby 59 ipv6 autoconfig
 standby 59 timers msec 500 msec 1500
 standby 59 priority 110
 standby 59 preempt delay minimum 60
 standby 59 authentication md5 key-string ...
 ipv6 enable
 no ipv6 redirects
!
```

cap-lab-ipv6-fw01

```
interface GigabitEthernet0/0.859
 vlan 859
 nameif outside
 security-level 0
 no ip address
 ipv6 enable
!

interface GigabitEthernet0/1.860
 vlan 860
 nameif inside
 security-level 100
 no ip address
 ipv6 address ...
 ipv6 enable
!
```

Linkkiverkon (Vlan859) rajapinnoille ei konfiguroitu globaaleja IPv6-osoitteita lainkaan, vaan käytettiin linkkilokaaleja osoitteita. Internet-reitittimillä *standby 59 ipv6 autoconfig* -komennolla saatiin HSRPv2-protokollaa käyttäen aikaan virtuaalinen linkkilokaali osoite FE80::5:73FF:FEA0:3B, jota voidaan laboratoriopalomuurilla käyttää *next-hop*-osoitteena reititettäessä IPv6-liikenne sieltä ulos. Vastaavasti In-

ternet-reitittimillä voidaan käyttää laboratoriopalomuurin rajapinnan linkkilokaalia osoitetta *next-hop*-osoitteena IPv6-testiverkolle:

```
cap-pri-rt1#sh ipv6 int GigabitEthernet0/1.859
GigabitEthernet0/1.859 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::222:55FF:FEE4:8B1A [UNA]
  Virtual link-local address(es):
    FE80::5:73FF:FEA0:3B [OOD]
  Description: LAB-ASA-INETRT-link
  No global unicast address is configured

cap-sec-rt1# sh ipv6 int GigabitEthernet0/1.859
GigabitEthernet0/1.859 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::222:55FF:FEE4:7F1A [UNA]
  Virtual link-local address(es):
    FE80::5:73FF:FEA0:3B [UNA/OOD]
  Description: LAB-ASA-INETRT-link
  No global unicast address is configured

cap-lab-ipv6-fw01# sh ipv6 int outside
outside is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::218:b9ff:fea8:6182
  No global unicast address is configured
```

IPv6-Internet-yhteys otettiin itse asiassa käyttöön passiivisessa konesalissa, ja em. konfiguraatiolla aktiivisen konesalin reitittimestä cap-pri-rt1 tulee aktiivinen korkeamman HSRPv2-prioriteetin takia. Tämä nähdään cap-sec-rt1-reitittimen virtuaaliosoitteen perässä olevasta unactive-tunnisteesta (UNA). HSRPv2-konfiguraatiota muuttamalla voitaisiin cap-sec-rt1-reitittimestä tehdä aktiivinen, mutta yhden ylimääräisen IPv6-hypyn ei pitäisi konesaliympäristössä vaikuttaa merkittävästi viipeeseen Internetiin liikennöitäessä.

Laboratoriopalomuurin pääasiallinen tarkoitus on suojata laboratorioverkkoa ei-halutulta liikenteeltä. Ciscon palomuurin ominaisuuksiin kuuluu, että liikennöinti sen läpi onnistuu ainoastaan suuremman tietoturvatason rajapinnoista pienemmän tietoturvatason rajapintoihin. Tämän vuoksi ulkojalan rajapinnan (Vlan859) tietoturvatasoksi (*security-level*) asetettiin 100 ja sisäjalan rajapinnan (Vlan860) tietoturvatasoksi 0. Rajapinnoille konfiguroitiin lisäksi pääsylistat, jotka rajoittavat testiverkosta ja testiverkkoon sallittavaa liikennettä vielä tarkemmalla tasolla. Pääsylista ipv6_out rajoittaa liikennettä IPv6-testiverkosta ulos seuraavasti:

```
ipv6 access-list ipv6_out permit tcp .../120 any eq www
ipv6 access-list ipv6_out permit tcp .../120 any eq https
ipv6 access-list ipv6_out permit icmp6 .../120 any
ipv6 access-list ipv6_out permit udp .../120 host 2001:4860:4860::8888 eq domain
ipv6 access-list ipv6_out permit udp .../120 host 2001:4860:4860::8844 eq domain
```

Pääsylistan rivit 1, 2 ja 3 sallivat http-, https- ja ICMPv6-liikenteen IPv6-testiverkosta ulos ja rivit 4 ja 5 DNS-kyselyt Googlen julkisilta IPv6-nimipalvelimilta. Pääsylista ipv6_in puolestaan sallii ainoastaan ICMPv6-liikenteen testiverkkoon:

```
ipv6 access-list ipv6_in permit icmp6 any .../120
```

Pääsylistat asetettiin rajapinnoille seuraavasti:

```
access-group ipv6_in in interface outside
access-group ipv6_out in interface inside
```

Ei-haluttu liikennne sekä sisään- että ulospäin pudotetaan siis heti sillä rajapinnalla, jolta liikenne palomuurille tulee sisään. Kuten aiemmin kerrottiin, asetettiin lopuksi vielä ulkojalan rajapinnalle oletusreitti Internet-reitittimien HSRPv2-osoitteeseen:

```
ipv6 route outside ::/0 fe80::5:73ff:fea0:3b
```

Näin ollen laboratoriopalomuurin IPv6-reititystaulu näyttää seuraavalta:

```
cap-lab-ipv6-fw01# sh ipv6 route

IPv6 Routing Table - 5 entries
Codes: C - Connected, L - Local, S - Static
L   .../128 [0/0]
     via ::, inside
C   .../120 [0/0]
     via ::, inside
L   fe80::/10 [0/0]
     via ::, outside
     via ::, inside
L   ff00::/8 [0/0]
     via ::, outside
     via ::, inside
S   ::/0 [0/0]
     via fe80::5:73ff:fea0:3b, outside
```

Reititetään vielä Internet-reitittimillä IPv6-testiverkko kohti laboratoriopalomuuria, niin kaiken pitäisi olla valmista:

```
ipv6 route .../120 FE80::218:B9FF:FEA8:6182
```

Pingataan ensin sekä aktiiviselta että passiiviselta Internet-reitittimeltä palomuurin ulkojalkaa:

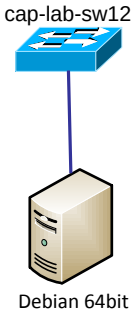
```
cap-pri-rt1#ping fe80::218:b9ff:fea8:6182
Output Interface: GigabitEthernet0/1.859
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::218:B9FF:FEA8:6182, timeout is 2 seconds:
Packet sent with a source address of FE80::222:55FF:FEE4:8B1A%GigabitEthernet0/1.859
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

cap-sec-rt1#ping fe80::218:b9ff:fea8:6182
Output Interface: GigabitEthernet0/1.859
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::218:B9FF:FEA8:6182, timeout is 2 seconds:
Packet sent with a source address of FE80::222:55FF:FEE4:7F1A%GigabitEthernet0/1.859
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Pingataan vielä palomuurin ulkojalalta sekä Internet-reitittimien virtuaalista HSRPv2-osoitetta että kummankin reitittimen omaa linkkilokaalia osoitetta:

```
cap-lab-ipv6-fw01# ping outside FE80::5:73FF:FEA0:3B
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to fe80::5:73ff:fea0:3b, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

cap-lab-ipv6-fw01# ping outside FE80::222:55FF:FEE4:8B1A
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to fe80::222:55ff:fee4:8b1a, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms

cap-lab-ipv6-fw01# ping outside FE80::222:55FF:FEE4:7F1A
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to fe80::222:55ff:fee4:7f1a, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

6.2.3 Debian-palvelin

Testiverkko ei ole mitään ilman verkossa olevaa testipalvelinta, joten verkkoon liitettiin seuraavaksi vanha työasemakäytössä ollut Debian-palvelin kuvan 54 mukaisesti.



Kuva 54: 64-bittinen Debian-testipalvelin, L2.

Palvelimessa ei ole kuin yksi verkkoliitäntä, joten sekä IPv4- että IPv6-liikenne täytyi viedä palvelimelle tätä samaa liitäntää pitkin. IPv4-rajapintaa käytetään puhtaasti palvelimen hallintaan ja itse testiliikenne kulkee IPv6-rajapintaa pitkin. Tämä onnistui seuraavalla `/etc/network/interfaces`-tiedoston konfiguraatiolla: [168]

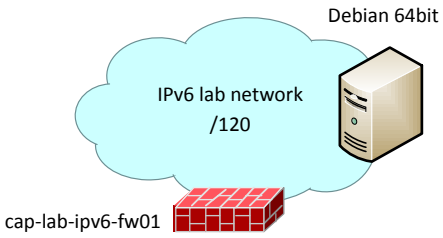
```
iface vlan30 inet static
       address ...
       netmask ...
       gateway ...
       vlan_raw_device eth0

iface vlan860 inet6 static
       address ...
```

```
        netmask 120
        gateway ...
        vlan_raw_device eth0
```

Konfiguraation voi tarkistaa *ifconfig*-komennolla:

```
vlan30    Link encap:Ethernet  HWaddr 00:08:74:a4:86:c7
          inet addr:...  Bcast:...  Mask:...
          inet6 addr: fe80::208:74ff:fea4:86c7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:253252 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9762 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:12020804 (11.4 MiB)  TX bytes:1153851 (1.1 MiB)

vlan860   Link encap:Ethernet  HWaddr 00:08:74:a4:86:c7
          inet6 addr: .../120 Scope:Global
          inet6 addr: fe80::208:74ff:fea4:86c7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8586 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3498 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1085357 (1.0 MiB)  TX bytes:324669 (317.0 KiB)
```

Palvelin saatiin näin IPv6-testiverkkoon kuvan 55 mukaisesti.



Kuva 55: 64-bittinen Debian-testipalvelin, L3.

Palvelimelle täytyi asettaa vielä nimipalvelimet. Tätä varten käytettiin Googlen julkisia nimipalvelimia 2001:4860:4860::8888 ja 2001:4860:4860::8844, jotka asetettiin `/etc/resolv.conf`-tiedostoon: [169]

```
nameserver 2001:4860:4860::8888
nameserver 2001:4860:4860::8844
```

Capgeminin omat nimipalvelimet osaavat tokin nekin selvittää IPv6-osoitteita, mutta niille liikennöinti ei vielä onnistu IPv6-protokollalla. Asettamalla Googlen nimipalvelimet saadaan sekä nimenselvitys että itse liikennöinti tapahtumaan kummatkin IPv6-protokollaa käyttäen. Kuten aiemmin näytettiin, Googlen nimipalvelimille tehtiin palomuuriavaukset IPv6-testiverkosta, joten nimenselvityksen ja itse IPv6-liikenteen pitäisi nyt toimia.

```
PING www.google.fi(lb-in-x5e.1e100.net) 56 data bytes
64 bytes from lb-in-x5e.1e100.net: icmp_seq=1 ttl=57 time=16.5 ms
64 bytes from lb-in-x5e.1e100.net: icmp_seq=2 ttl=57 time=16.2 ms
^C
--- www.google.fi ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 16.294/16.433/16.572/0.139 ms
```

Ainakin DNS- ja ICMPv6-liikenne siis toimivat. Kokeillaan vielä http-yhteyttä tekstipohjaisella w3m-selaimella osoitteeseen `http://www.whatismyv6.com/`:

```
This page shows your IPv6 and/or IPv4 address
You are connecting with an IPv6 Address of:
...
```

Myös http-yhteys toimii, joten näyttää siltä, että IPv6-käyttöönotto on tehty onnistuneesti. Pakettikaappaus IPv6-yhteydestä sivustolle `http://www.whatismyv6.com/` on liitteessä A. Palvelimelle asennettiin vielä Apache-http-palvelinohjelmisto, jolla testattiin, että palvelin on saavutettavissa myös Internetistä IPv6-protokollalla [170]. Tähän tarkoitukseen voidaan käyttää esim. aiemmin luvussa 4.4.2 esiteltyjä tunnelinvälittäjiä. Julkiseen capgemini.fi-verkkotunnukseen lisättiin myös AAAA-tietue `ipv6test.capgemini.fi`, joka osoittaa Debian-palvelimen IPv6-osoitteeseen:

```
$ORIGIN capgemini.fi.
ipv6test                AAAA    ...
```

Tässä kohtaa on huomattava, että capgemini.fi-verkkotunnus on määritelty löytyvän neljältä eri nimipalvelimelta:

```
[ksaarnia@cap-nis-02 ~]$ dig capgemini.fi soa

;; AUTHORITY SECTION:
capgemini.fi.           1582    IN      NS      ns1-swe.global.sonera.se.
capgemini.fi.           1582    IN      NS      ns2.capgemini.fi.
capgemini.fi.           1582    IN      NS      ns2-usa.global.sonera.net.
capgemini.fi.           1582    IN      NS      ns1.capgemini.fi.

;; ADDITIONAL SECTION:
ns1.capgemini.fi.       1221    IN      A       145.247.23.4
ns2.capgemini.fi.       1582    IN      A       145.247.23.20
ns1-swe.global.sonera.se. 1741  IN      A       80.64.9.124
ns1-swe.global.sonera.se. 1788  IN      AAAA    2001:6e8:400:ffff:1::53
ns2-usa.global.sonera.net. 1190 IN      A       46.38.190.56
```

Verkkotunnus löytyy siis sekä Capgeminin omilta että Soneran nimipalvelimilta. Capgeminin nimipalvelimilla ei vielä ole IPv6-osoitteita, mutta Soneran toisella nimipalvelimella on. Capgeminin nimipalvelimet on määritelty ilmoittamaan niille tehtävistä muutoksista Soneran nimipalvelimille, jotta nimipalvelukonfiguraatio pysyy synkronoituna, joten Capgeminin nimipalvelimille lisättävän tietueen pitäisi siirtyä alueen siirtotoiminnallisuudella (*zone transfer*) Soneran palvelimille ja olla näin haettavissa myös IPv6-protokollalla. Katsotaan seuraavaksi, tapahtuiko näin.

```
Pinging ipv6test.capgemini.fi [...] with 32 bytes of data:
Reply from ...: time=20ms
```

Jotta http-yhteyttä palvelimelle voidaan testata, täytyy sitä varten ensin tehdä http-yhteyden salliva palomuuriavaus laboratoriopalomuurille:

```
ipv6 access-list ipv6_in permit tcp any host ... eq www
```

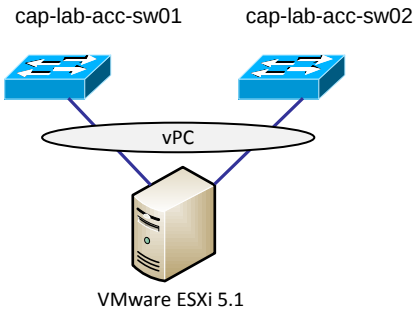
Kokeillaan sitten ottaa yhteys osoitteeseen `http://ipv6test.capgemini.fi/`:

```
It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.
```

Näyttää siltä, että AAAA-tietue siirtyi juuri niin kuin pitikin ja Capgeminin julkinen nimipalvelu on siis saavutettavissa myös IPv6-protokollalla. Tosin vain yhdellä neljästä nimipalvelimesta on IPv6-osoite, joten IPv6-käyttäjän näkökulmasta palvelu on kokonaan sen varassa. Lisätty AAAA-tietue toki ratkeaa myös IPv4-protokollalla kaikilta neljältä palvelimelta, mutta vähintään toiselle nimipalvelimelle täytyy saada IPv6-osoite ennen kuin IPv6-nimipalvelua voidaan viedä tuotantoon.

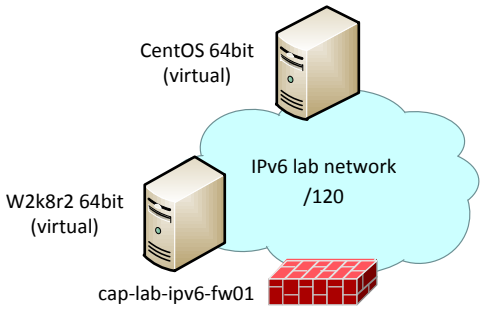
6.2.4 ESXi-palvelin

Capgeminin tavoitteena on olla valmis tarjoamaan IPv6-palveluita vuoden 2013 loppuun mennessä. Siksi haluttiin kehittää konsepti, jolla voidaan tarvittaessa hyvinkin nopeasti provisioida IPv6-pohjainen www-palvelu. Tätä varten IPv6-testiverkkoon asennettiin vielä VMware ESXi -palvelin, jolle puolestaan asennettiin 64-bittiset CentOS - ja Windows Server 2008 R2 -virtuaalipalvelimet. ESXi-palvelin kytkettiin laboratorioverkon pääsykytkimiin 2x1Gbps-porttikanavaliitännällä. Tämä on mahdollinen liitäntä tuotantopalvelimelle, joskin tuotantopalvelin kytketään yleensä 2x10Gbps-nopeudella. Saamassamme testipalvelimessa ei ollut 10Gbps-verkkoliitäntöjä, mutta tämän työn tavoitteena ei varsinaisesti olekaan tutkia suorituskykyyn liittyviä asioita. Palvelimen fyysinen liitäntä testiverkkoon on kuvattu kuvassa 56.



Kuva 56: ESXi-testipalvelin, L2.

Virtuaalipalvelimet konfiguroitiin IPv6-testiverkkoon kuvan 57 mukaisesti:

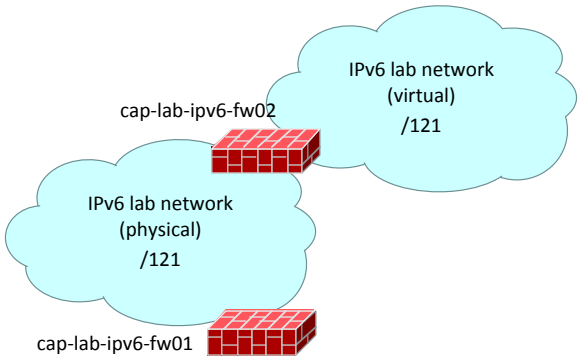


Kuva 57: Virtuaalipalvelimet IPv6-testiverkossa.

Kumpikin virtuaalipalvelin saatiin onnistuneesti testiverkkoon:

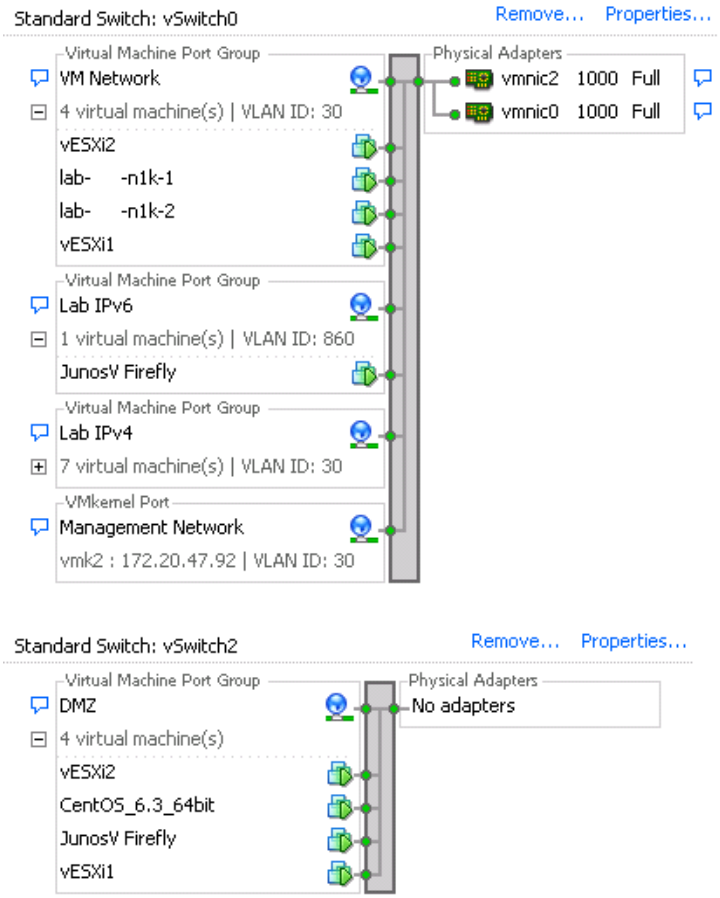
```
cap-lab-ipv6-fw01> ping ...
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to ..., timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms

cap-lab-ipv6-fw01> ping ...
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to ..., timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Tuotantoverkossa ei kuitenkaan vielä ole yhtäkään IPv6-palomuuria. Tämän vuoksi piti miettiä ratkaisuja, joilla IPv6-palvelu voidaan provisioida ilman uusia laitehankintoja tai vanhojen laitteiden konfiguroimista. Ciscon virtuaaliset ASA 1000V/VSG -palomuurit ei vieläkään tue IPv6-protokollaa, mutta Juniperin JunosV Firefly tukee [171, 172]. Se on virtuaalinen palomuuri, jossa on sama Junos-käyttöjärjestelmä kuin useissa muissa Capgeminin palomuureissa, joten se oli luonteva valinta virtuaaliseksi IPv6-testipalomuuriksi. Firefly-palomuuria (cap-lab-ipv6-fw02) varten IPv6-testiverkko jaettiin kahdeksi pienemmäksi /121-verkoksi kuvan 58 mukaisesti.



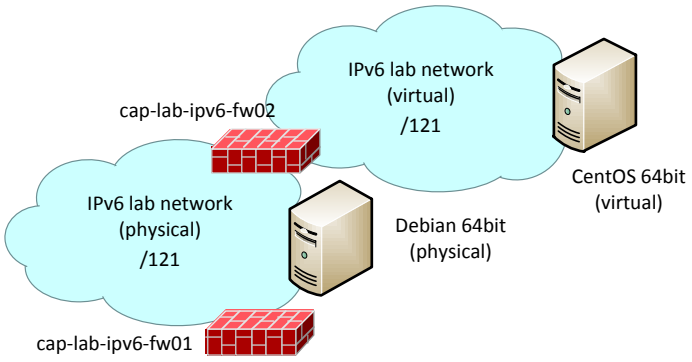
Kuva 58: Fyysinen ja virtuaalinen IPv6-testiverkko.

Virtuaalinen IPv6-verkko eristettiin Firefly-palomuurin taakse konfiguroimalla se omaksi porttiryhmäkseen ESXi-palvelimen toiselle virtuaalikytkimelle vSwitch2:



Kuva 59: ESXi-palvelimen verkkokonfiguraatio.

Firefly-palomuurin sisäjalan IPv6-osoite konfiguroitiin virtuaaliseen ja ulkojalan IPv6-osoite fyysiseen IPv6-testiverkkoon. Samalla CentOS-palvelin siirrettiin virtuaaliseen verkkoon kuvan 60 mukaisesti. Windows Server 2008 R2 -palvelin täytyi tässä vaiheessa jättää IPv6-testiverkosta pois, koska se ei ymmärtänyt /121-verkkoprefiksiä.

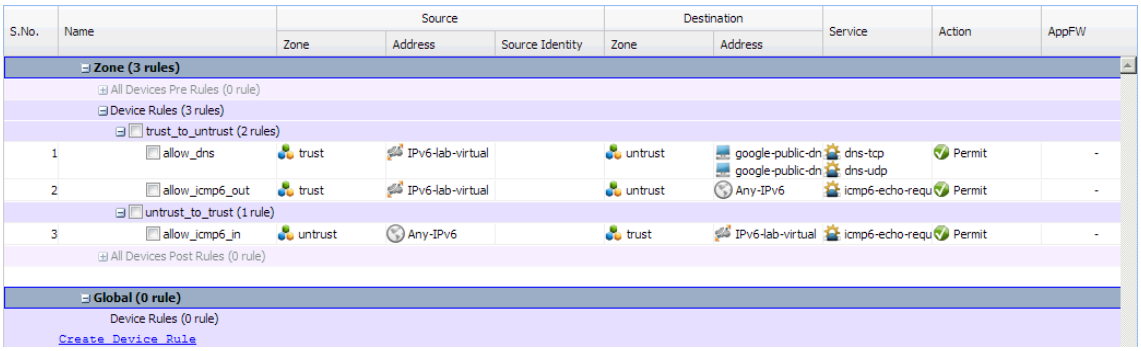


Kuva 60: Palvelimet fyysisessä ja virtuaalisessa IPv6-testiverkossa.

CentOS-palvelin saatiin onnistuneesti /121-verkkosegmenttiin:

```
cap-lab-ipv6-fw01> ping ...
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to ..., timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

Virtuaalinen Firefly-palomuuri lisättiin myös Capgeminin Junos Space -palomuurihallintanäkymään, jotta sääntökannan muokkaaminen on käyttäjäystävällisempää kuin suoraan komentoriviltä. Kuvakaappaus Junos Spacesta on esitetty kuvassa 61. Firefly-palomuurille tehtiin fyysisen IPv6-palomuurin tavoin DNS-avaukset Googlen julkisille nimipalvelimille (sääntö 1) ja lisäksi sallitiin ICMPv6 kumpaankin suuntaan (säännöt 2 ja 3).



Kuva 61: Firefly-palomuurin sääntökanta Junos Spacessa.

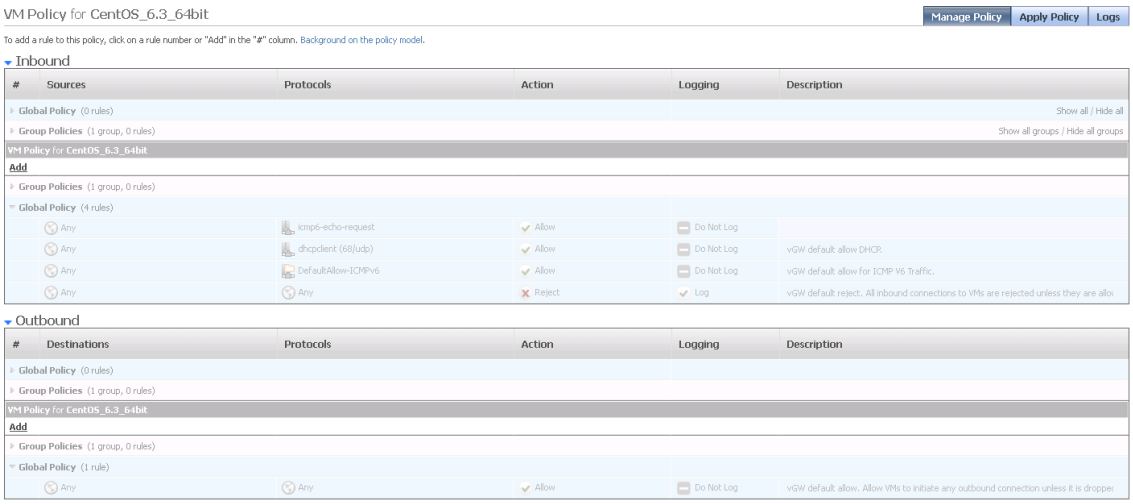
Firefly-palomuurin konfiguraatio sen tärkeimmiltä osin on esitetty alla.

```
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet6 {
                address ...;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address ...;
            }
        }
    }
}
routing-options {
    rib inet6.0 {
        static {
            route 0::0/0 next-hop ...;
        }
    }
}
security {
    address-book {
        global {
```

```
            address IPv6-lab-virtual ...;
            address google-public-dns-a.google.com 2001:4860:4860::8888/128;
            address google-public-dns-b.google.com 2001:4860:4860::8844/128;
        }
    }
    forwarding-options {
        family {
            inet6 {
                mode flow-based;
            }
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy allow_dns {
                match {
                    source-address IPv6-lab-virtual;
                    destination-address [ google-public-dns-a.google.com google-public-dns-b.google.com ];
                    application [ junos-dns-tcp junos-dns-udp ];
                }
                then {
                    permit;
                }
            }
            policy allow_icmp6_out {
                match {
                    source-address IPv6-lab-virtual;
                    destination-address any-ipv6;
                    application junos-icmp6-echo-request;
                }
                then {
                    permit;
                }
            }
        }
        from-zone untrust to-zone trust {
            policy allow_icmp6_in {
                match {
                    source-address any-ipv6;
                    destination-address IPv6-lab-virtual;
                    application junos-icmp6-echo-request;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone untrust {
            interfaces {
                ge-0/0/0.0;
            }
        }
        security-zone trust {
            interfaces {
                ge-0/0/1.0;
            }
        }
    }
}
```

Juniperilla on myös toinen virtuaalinen tietoturvaratkaisu, vGW Virtual Gateway [173]. Se koostuu kahdesta virtuaalipalvelimesta: Security Design vGW -palvelimesta, jolla luodaan palomuurisääntöjä vGW Security VM -virtuaalipalvelimelle, joka puo-

lestaan asettaa ne vGW-kernelmoduuliin [174, 175]. Vaikka vGW tukee täysin IPv6-protokollaa, se ei ole reitittävä tuote joten Firefly sopii siksi paremmin virtuaaliseksi palomuuriratkaisuksi Capgeminille. Virtual Gatewaylla voidaan kuitenkin haluttaessa tiukentaa virtuaalisen palvelinympäristön tietoturvaa entisestään rajoittamalla samassa verkossa olevien palvelimien liikennöintiä keskenään. Sitä varten täytyi testiympäristöön asentaa vielä VMware vCenter-ohjelmisto, joka asennettiin Windows Server 2008 R2 -virtuaalipalvelimelle [176]. Tuotantoympäristössä vCenter-ohjelmistoa ei kuitenkaan luonnollisesti kannata asentaa samalle ESXi-palvemelle jota sillä hallitaan. CentOS-virtuaalipalvelin suojattiin vielä erikseen vGW-palomuurilla. Kuvakaappaus vGW-palomuurilta on esitetty kuvassa 62.

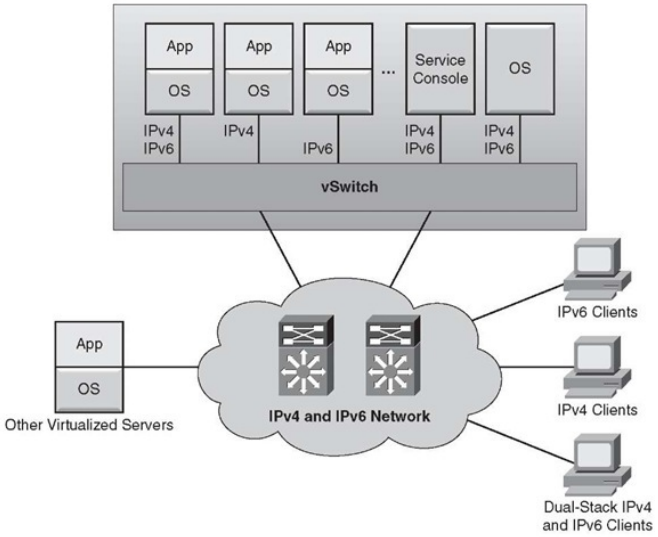


Kuva 62: vGW-palomuurilla suojattu CentOS-palvelin.

6.2.5 Cisco Nexus 1000V

ESXi-palvelimelle asennettiin vielä virtuaalinen Cisco Nexus 1000V -kytkin, joka korvaa ESXi-palvelimen dvSwitch-kytkimen (*vSphere Distributed Switch*) [177]. Capgeminin palvelin- ja VMware-tiimeillä on hiljattain ollut haasteita tiettyjen ympäristöjen VLAN-konfiguraatioiden kanssa. Lisähaasteen aiheuttaa se, että verkkotiimillä ei ole näkyvyyttä ESXi-palvelimen virtuaalikytkimen konfiguraatioon. Nexus 1000V -kytkimellä saadaan näkyvyys ESXi-palvelimen virtuaalikytkimeen myös Capgeminin verkkotiimille, jolloin ongelmanratkaisu helpottuu ja nopeutuu huomattavasti ja siitä tulee entistä läpinäkyvämpää.

ESXi-palvelimen dvSwitch-kytkin eroaa normaalista vSwitch-kytkimestä (*vSphere Standard Switch*) siinä, että se on hierarkiassa astetta vSwitch-kytkintä ylempänä: vSwitch-kytkin kytkee liikennettä yhden ESXi-palvelimen sisällä, mutta dvSwitch-kytkin voi kytkeä liikennettä usean eri ESXi-palvelimen välillä [178]. Tämän vuoksi fyysiselle ESXi-palvelimelle virtualisoitiin kaksi virtuaalista ESXi-palvelinta vESXi1 ja vESXi2, jotka lisättiin testiympäristöön [179]. Kuvassa 63 on havainnollistettu ESXi-palvelimen vSwitch-virtuaalikytkimen toimintaa.

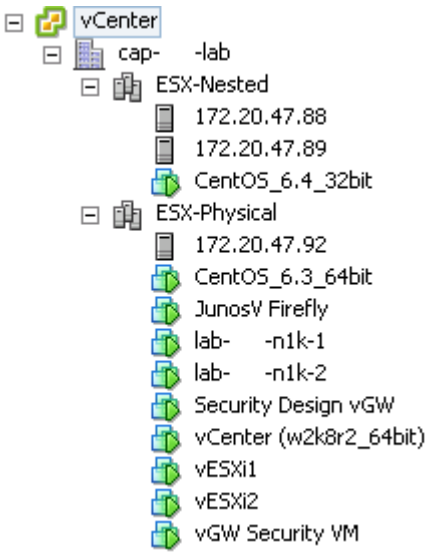


Kuva 63: ESXi-palvelimen vSwitch-virtuaalikytkin. [35]

Ciscon fyysisien kytkimien tapaan myös virtuaalinen Nexus 1000V -kytkin koostuu kahdesta moduulista: VEM-moduulista (*Virtual Ethernet Module*) ja VSM-moduulista (*Virtual Supervisor Module*). Kytkin konfiguroidaan VSM-moduulin kautta ja se välittää muutokset eteenpäin VEM-moduuleille. Yhdellä VSM-moduulilla voidaan hallita 64 VEM-moduulia. [177] Kummallekin virtualisoidulle ESXi-palvelimelle asennettiin VEM-moduulit ja VSM HA-pari (*High Availability*) asennettiin fyysiselle ESXi-palvelimelle. VSM-moduulit nimettiin cap-pri-n1k-1 ja cap-pri-n1k-2. 64-bittisten käyttöjärjestelmien virtualisoinnissa virtualisoiduilla ESXi-palvelimilla kohdatuista ongelmista johtuen vESXi2-palvelimelle asennettiin vielä yksi 32-bittinen CentOS-virtuaalipalvelin, joten testiympäristön lopullinen virtuaalipalvelimien lukumäärä oli 10:

1. virtualisoitu ESXi-palvelin (172.20.47.88 = vESXi1)
2. virtualisoitu ESXi-palvelin (172.20.47.89 = vESXi2)
3. 32-bittinen CentOS 6.4 -palvelin
4. 64-bittinen CentOS 6.3 -palvelin
5. Junos Firefly -palomuuri
6. Cisco Nexus 1000V VSM 1 (cap-pri-n1k-1)
7. Cisco Nexus 1000V VSM 2 (cap-pri-n1k-2)
8. Juniper Security Design vGW
9. 64-bittinen Windows Server 2008 R2 -palvelin (vCenter)
10. Juniper vGW Security VM

Nämä virtuaalipalvelimet on havainnollistettu kuvassa 64. Palvelimet 172.20.47.88 ja vESXi1 sekä 172.20.47.89 ja vESXi2 ovat samoja, virtualisoituja ESXi-palvelimia. Toisen tason virtuaaliympäristö on selkeyden vuoksi siirretty omaan klusteriinsa (ESX-Nested) ja ensimmäisen tason virtuaaliympäristö omaansa (ESX-Physical).



Kuva 64: Virtuaalinen testiympäristö, fyysinen ja kaksi virtuaalista ESXi-palvelinta.

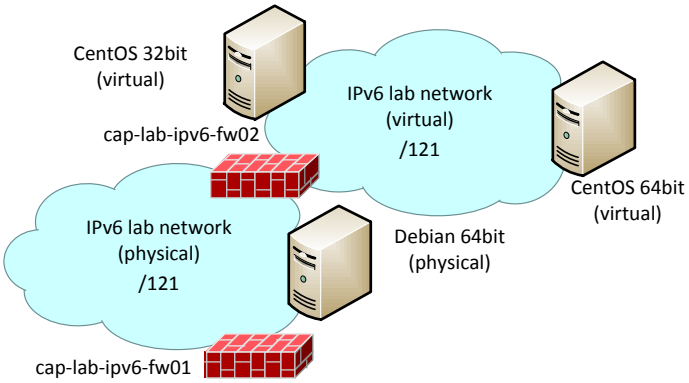
Nexus 1000V -kytkimeltä nähdään, että VEM-moduulit on asennettu virtuaalisille ESXi-palvelimille ja että kummatkin VSM-moduulit ovat fyysisellä ESXi-palvelimella:

```
lab-pri-n1k# sh mod
Mod  Ports  Module-Type                      Model              Status
---  -----  -------------------------------  -----------------  ------------
1    0      Virtual Supervisor Module        Nexus1000V         active *
2    0      Virtual Supervisor Module        Nexus1000V         ha-standby
3    332    Virtual Ethernet Module          NA                 ok
4    332    Virtual Ethernet Module          NA                 ok

Mod  Sw                Hw
---  ----------------  ------------------------------------------------
1    4.2(1)SV2(2.1)    0.0
2    4.2(1)SV2(2.1)    0.0
3    4.2(1)SV2(2.1)    VMware ESXi 5.1.0 Releasebuild-1065491 (3.1)
4    4.2(1)SV2(2.1)    VMware ESXi 5.1.0 Releasebuild-1065491 (3.1)

Mod  Server-IP        Server-UUID                           Server-Name
---  ---------------  ------------------------------------  --------------------
1    172.20.47.90     NA                                    NA
2    172.20.47.90     NA                                    NA
3    172.20.47.88     42252b85-70cb-8367-ce91-72a345ef3d9a  172.20.47.88
4    172.20.47.89     4225562c-066c-8b5d-91e0-5daa23087a3c  172.20.47.89

* this terminal session
```

Nexus 1000V -kytkin on 1.10.2012 alkaen ollut saatavilla ilmaiseksi [180]. Sen tarkempi analysointi ja konfigurointi on tämän työn laajuuden ulkopuolella, mutta todettakoon, että 32-bittinen CentOS-testipalvelin toimi ilman ongelmia IPv6:lla, kun se asetettiin Nexus 1000V -kytkimelle kuuluvaan porttiryhmään vESXi2-palvelimella:

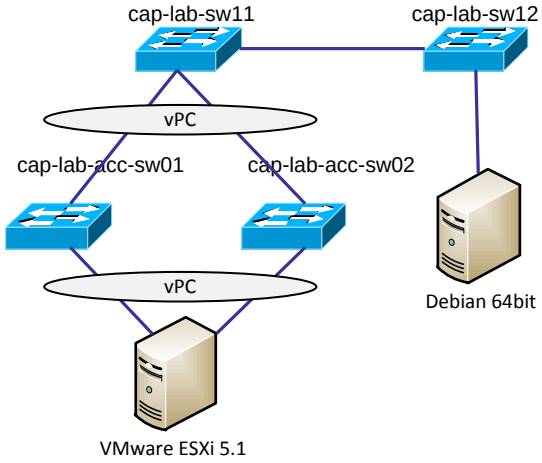
```
cap-lab-ipv6-fw01> ping ...
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to ..., timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

Lopullinen IPv6-testiverkko näyttää L3-kerroksella näin kuvan 65 mukaiselta.



Kuva 65: Lopullinen IPv6-testiverkko, L3.

L2-kerroksella testiverkon palvelimet kytkettiin kuvan 66 mukaisesti.



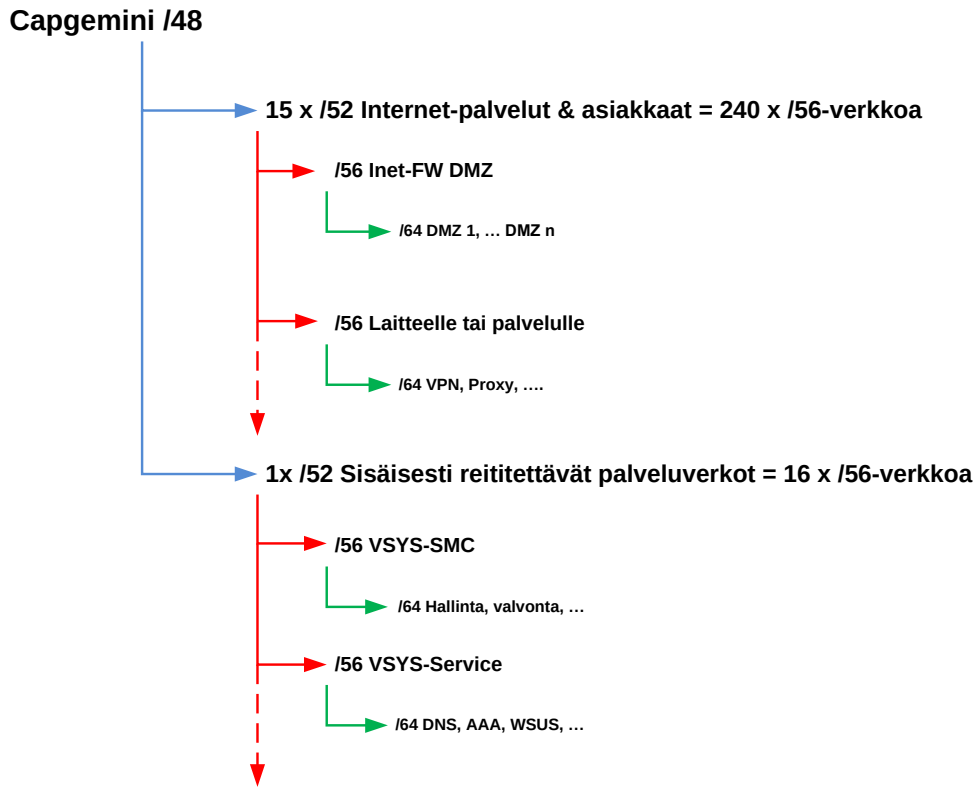
Kuva 66: Lopullinen IPv6-testiverkko, L2.

6.3 IPv6-osoitteistussuunnitelmat

RIPE on allokoinut Capgeminille julkisen /48-IPv6-osoitealueen. Kuten aiemmin todettiin, /64-palvelinverkkosegmenttien käyttäminen ilman asianmukaista suojautumista NDP-tulvitushyökkäykseltä voi olla ongelmallista, mutta /120-palvelinverkkosegmenttien käyttöön puolestaan liittyy omat ongelmansa. Seuraavaksi esitelläänkin IPv6-osoitteistussuunnitelmat sekä /64- että /120-palvelinverkkosegmenteille.

6.3.1 /64-palvelinverkkosegmentit

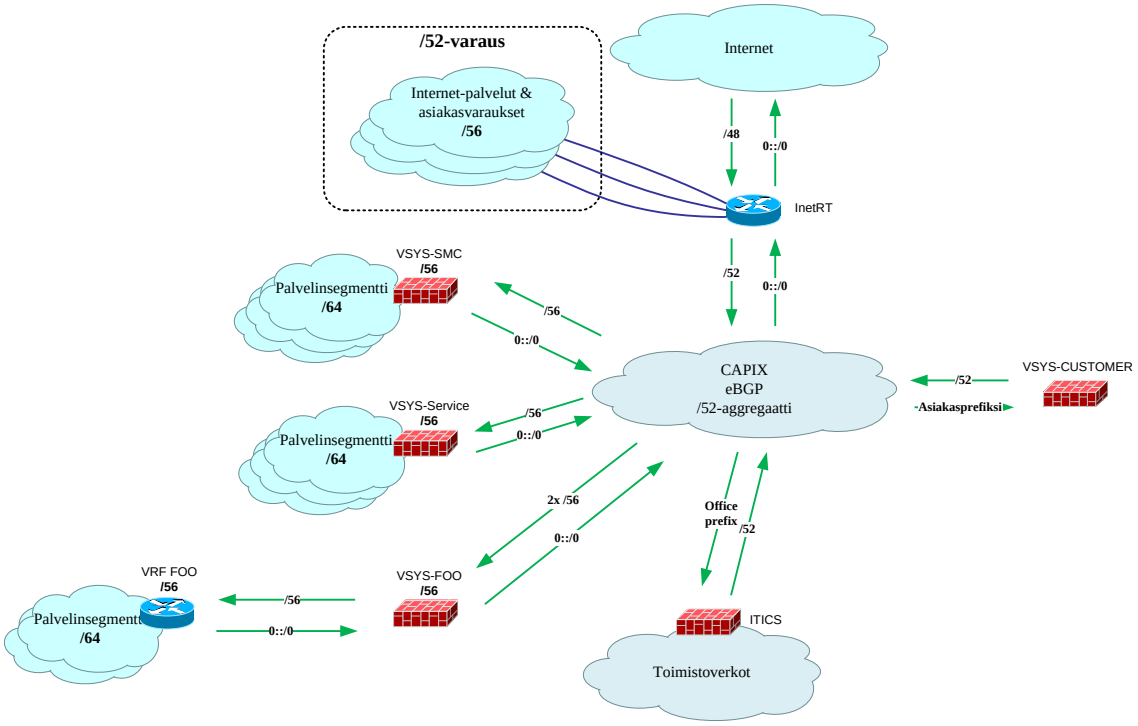
Kuvassa 67 on ylätason suunnitelma /64-palvelinverkkosegmenttejä varten. Siinä Capgeminin /48-prefiksi on jaettu 15 Capgeminin tarjoamien Internet-palveluiden ja sen asiakkaiden käyttöön jaettavaan /52-prefiksiin ja yhteen sisäiseen /52-prefiksiin.



Kuva 67: Capgeminin IPv6-osoitteistus, /64-palvelinverkkosegmentit.

Internetistä reitititetään koko Capgeminin julkinen /48-prefiksi Capgeminin Internet-reitittimille kuvan 68 mukaisesti. Siitä eteenpäin reitititetään vain sisäiseen käyttöön varattu /52-prefiksi, josta edelleen Capgeminin eri palomuureille n kpl /56-prefiksejä. Capgeminin toimistoverkot ovat ITICS:n palomuurin takana, joten sieltä reitititetään Capgeminin sisäinen /52-prefiksi kohti Capgeminin konesalin CAPIX-vrf:ää. Myös asiakkaiden palomuureille mainostetaan tätä sisäistä /52-prefiksiä ja CAPIX:n asiakasrajapinnoissa sallitaan pääsy ainoastaan Capgeminin sisäiseen /52-prefiksiin. Liikennöinti toisten asiakkaiden verkkoihin estetään pääsynhallintalistal-

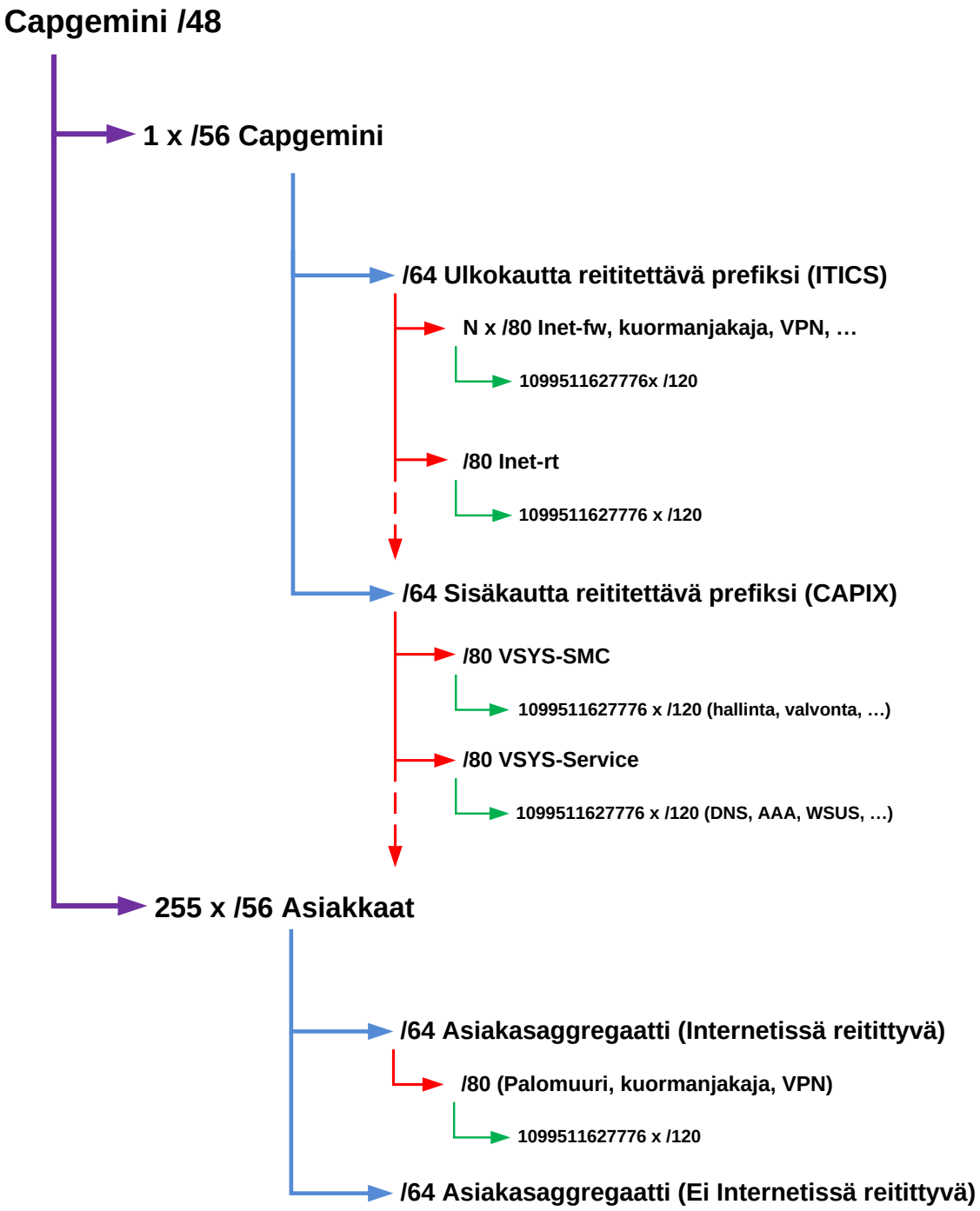
la. Tässä ratkaisussa palvelinverkkosegmenteille konfiguroidaan /64-prefiksit, mutta palomuurilla tai reitittimellä estetään liikenne muuhun verkkoon kuin esim. sen ensimmäisen /120-prefiksin rajaamaan segmenttiin. Näin voidaan suojautua ulkoa tulevalta NDP-tulvitushyökkäykseltä ennen kuin IPv6-FHS-mekanismit ovat tuettuja kaikissa Capgeminin käytössä olevissa verkkolaitteissa. Tämän ratkaisun etu on myös se, että palvelimien verkkokonfiguraatioon ei tarvitse myöhemmin koskea, vaan Capgeminin verkkotiimin poistettua palomuurisäännön tai pääsynhallintalistan verkon ensimmäiseen /120-segmenttiin on koko /64-segmentti käytettävissä.



Kuva 68: Capgeminin IPv6-osoitteistus, /64-palvelinverkkosegmentit.

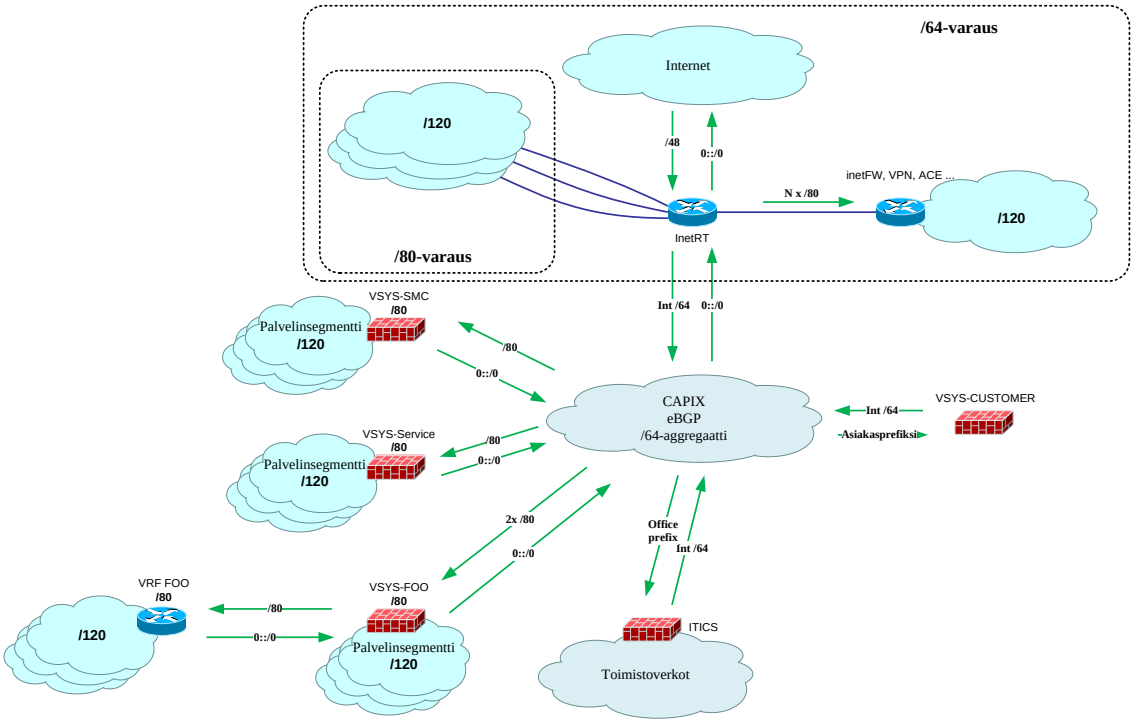
6.3.2 /120-palvelinverkkosegmentit

Kuvan 69 suunnitelmassa Capgeminin /48-osoitealueesta on lohkaistu sen omaan käyttöön /56-prefiksi ja kaikille sen asiakkaille jaettavaksi samankokoinen /56-prefiksi. Edelleen, Capgeminin omasta /56-prefiksistä yhden /64-prefiksin on ajateltu reitittyvän Internetissä ja toisen /64-prefiksin olevan Capgeminin sisäisesti reititettävä prefiksi. Myös tässä vaihtoehdossa Internetistä reititetään luonnollisesti koko Capgeminin julkinen /48-prefiksi Capgeminin Internet-reitittimille. Siitä eteenpäin reititetään vain sisäiseen käyttöön varattu /64-prefiksi (vrt. /52), josta edelleen Capgeminin eri palomuureille n kpl /80-prefiksejä (vrt. /56). Tässä ratkaisussa ei tarvita pääsylistoja rajaamaan liikennettä palvelinverkkosegmentteihin, mutta se ei välttämättä skaalaudu yhtä hyvin kuin edellinen reitittimien TCAM:n LPM-reittirajoitusten (*Longest Prefix Match*) vuoksi. Joissain laitteissa voi lisäksi olla rajoituksia muiden kuin /64-prefiksien vertaamiseen pääsynhallintalistoissa. [127]



Kuva 69: Capgeminin IPv6-osoitteistus, /120-palvelinverkkosegmentit.

Tässä suunnitelmassa on se huono puoli, että jos /120-segmenteissä havaitaan ennalta-arvaamattomia ongelmia, on niiden muuttaminen /64-segmenteiksi haastavaa. Yksi vaihtoehto olisi käyttää muuten kuvan 68 suunnitelmaa, mutta /120-palvelinverkkosegmenttejä. Tällöin niiden muuttaminen myöhemmin /64-segmenteiksi onnistuu helposti, mutta jokaisen verkossa olevan palvelimen prefiksi täytyy silti muuttaa erikseen. Kuvan 68 suunnitelmassa on se hyvä puoli, että kun NDP-



Kuva 70: Capgeminin IPv6-osoitteistus, /120-palvelinverkkosegmentit.

tulvitusongelmaan liittyvät ongelmat on ratkaistu esim. Ciscon FHS-mekanismeilla, voi Capgeminin verkkotiimi poistaa palomuurisäännön tai pääsynhallintalistan verkon ensimmäiseen /120-segmenttiin, jolloin koko /64-segmentti on käytettävissä.

6.4 IPv6-osoitteiden hallinta

Kuten jo aiemmin todettiin, Capgeminilla on käytössään EfficientIP SOLIDserver-ohjelmisto, jonka IPAM-moduuli (*IP Address Management*) on tarkoitettu IP-osoitteiden hallintaan [149]. Sillä voidaan hallita IPv4-osoitteiden lisäksi myös IPv6-osoitteita, joten se on luonnollinen valinta IPv6-hallintatyökaluksi. Ohjelmistoon kuuluu myös DNS-moduuli, jolla DNS-palvelimia voidaan integroida suoraan SOLIDserverin ja sen IPAM-moduulin kanssa. Tulevaisuudessa mm. Capgeminin julkinen verkkotunnus capgemini.fi on tarkoitus siirtää SOLIDserver-ohjelmiston hallitsemille nimipalvelimille, jolloin AAAA-DNS-tietue voidaan lisätä nimipalveluun automaattisesti samalla kun IPv6-osoite varataan SOLIDserverin IPAM-moduulista.

Tässä luvussa tehtiin IPv6-käyttöönotto Capgeminin konesaliverkossa. Se aloitettiin Internetin reunalta yhdistämällä Capgeminin konesaliverkko Internetiin IPv4-protokollan lisäksi myös IPv6-protokollalla. Sen jälkeen Capgeminin laboratorioon rakennettiin IPv6-testiverkko, johon asennettiin lukuisia palvelimia ja jossa IPv6-protokollan toimintaa voidaan testata. Lopuksi tehtiin osoitteistussuunnitelmat laajemmalle IPv6-käyttöönotolle Capgeminin konesaliverkossa. Viimeisessä luvussa tarkastellaan tässä diplomityössä aikaansaatuja tuloksia ja tehdään siitä yhteenveto.

7 Tulosten tarkastelu ja yhteenveto

Tämän diplomityön tavoitteena oli ottaa IPv6-protokolla käyttöön Capgeminin konesaliverkossa ja liittää se Internetiin IPv4-protokollan lisäksi myös IPv6-protokollalla. Toisena tavoitteena oli rakentaa Capgeminin laboratorioon IPv6-testiverkko, jossa IPv6-protokollaa ja sen käyttäytymistä voidaan testata ja jota voidaan jatkossa käyttää myös Capgeminin tarjoamien palveluiden toiminnan testaamiseen IPv6-protokollalla. Voidaan todeta, että tämän diplomityön ansiosta Capgeminilla on valmiudet tarjota palveluita asiakkailleen IPv4-protokollan lisäksi IPv6-protokollalla.

7.1 IPv6-käyttöönotto

IPv6-käyttöönotto aloitettiin liittämällä Capgeminin konesaliverkko Internetiin IPv6-protokollalla. Tämä saatiin tehtyä onnistuneesti ja Capgeminin konesaliverkko on nyt saavutettavissa Internetistä myös IPv6-protokollalla. IPv4- ja IPv6-protokollat eivät lähtökohtaisesti ole yhteensopivia keskenään, joten IPv6-protokollalla toimivien palveluiden tarjoaminen on palveluntarjoajille entistä tärkeämpää, kun niiden käyttäjät siirtyvät vähitellen käyttämään IPv6-protokollaa. Capgeminin tavoitteena onkin olla valmis tarjoamaan IPv6-palveluita vuoden 2013 loppuun mennessä. Tässä työssä on esitetty ratkaisu, jolla voidaan provisioida IPv6-www-palvelu Capgeminin konesaliverkossa helposti ja kustannustehokkaasti. Tämä ratkaisu testattiin myös käytännössä toimivaksi Capgeminin laboratoriossa IPv6-testiverkossa. Siitä tehtiin tarkoituksella mahdollisimman laaja ja kattava, jotta sitä voidaan hyödyntää myös jatkossa, kun testataan Capgeminin tarjoamien palveluiden toimintaa IPv6-protokollalla. Lopuksi tässä työssä esitettiin kaksi IPv6-osoitteistussuunnitelmaa Capgeminin konesaliverkkoon. Toisessa niistä käytettiin /64-palvelinverkkosegmenttejä ja toisessa /120-palvelinverkkosegmenttejä. Kummankin käyttämiseen liittyy kysymyksiä, mutta tässä työssä on esitetty ratkaisu, jolla /64-segmenttejä voidaan käyttää suojautuen samalla ulkoa tulevalta NDP-tulvitushyökkäykseltä ennen kuin FHS-mekanismit ovat tuettuja kaikissa Capgeminin käyttämissä verkkolaitteissa.

7.2 Parhaat käytännöt

Tässä työssä on tunnistettu muutamia parhaita käytäntöjä liittyen lähinnä IPv6-protokollan tietoturvaan. Seuraavaksi käydään ne vielä lyhyesti läpi.

7.2.1 Pääsykerros

Ciscon pääsykerroksella totetutettavia suojautumismekanismeja IPv4-verkossa kutsutaan CISF-mekanismeiksi ja IPv6-verkossa FHS-mekanismeiksi. Näillä mekanismeilla voidaan jo heti verkon pääsykerroksella suojautua useilta hyökkäyksiltä, ja niitä kannattaakin hyödyntää mahdollisuuksien mukaan. Kuten aiemmin kerrottiin, sekä DHCPv6- että RA-viestit kannattaakin oletusarvoisesti estää heti pääsykytkimellä. Jos *ipv6 nd raguard* -komento ei kuulu L2-kytkimen ominaisuuksiin, sama lopputulos saadaan aikaan IPv6-porttipääsynhallintalistalla. Tämä porttipääsynhallintalista estää RA-viestien lisäksi myös DHCPv6-liikenteen kytkinportista: [120]

```
ipv6 access-list IPV6_ACCESS_PORT
 remark Block all traffic DHCP server -> client
 deny udp any eq 547 any eq 546
 remark Block Router Advertisements
 deny icmp any any router-advertisement
 permit any any
!
interface gigabitethernet 1/0/1
 switchport
 ipv6 traffic-filter IPV6_ACCESS_PORT in
!
```

7.2.2 Reunareititin

Ciscon tunnistamat parhaat käytännöt IPv6-reittien vastaanottamiseen Internetin – tai minkä tahansa muun verkon reunalla olevalla reitittimellä ovat seuraavat: [121]

```
ipv6 prefix-list v6in-filter deny ::/0
ipv6 prefix-list v6in-filter deny ::/8 le 128
ipv6 prefix-list v6in-filter permit 2001::/32
ipv6 prefix-list v6in-filter deny 2001::/32 le 128
ipv6 prefix-list v6in-filter deny 2001:db8::/32 le 128
ipv6 prefix-list v6in-filter permit 2002::/16
ipv6 prefix-list v6in-filter deny 2002::/16 le 128
ipv6 prefix-list v6in-filter deny YOUR_CIDR_BLOCK_IPV6 le 128
ipv6 prefix-list v6in-filter deny 3ffe::/16 le 128
ipv6 prefix-list v6in-filter deny fc00::/7 le 128
ipv6 prefix-list v6in-filter deny fe80::/10 le 128
ipv6 prefix-list v6in-filter deny ff00::/8 le 128
ipv6 prefix-list v6in-filter permit 2000::/3 le 48
ipv6 prefix-list v6in-filter deny ::/0 le 128
```

Yllä esiintyvistä osoitteista originoituvan liikenteen estämisen lisäksi on muutamia muita parhaita käytäntöjä, jotka kannattaa ottaa huomioon reunareitittimen pääsylistoissa: [122]

```
deny ipv6 any YOUR_CIDR_BLOCK_IPV6 fragments
deny ipv6 YOUR_CIDR_BLOCK_IPV6 any
permit tcp host bgp_peer_ipv6 host router_ipv6 eq bgp
permit tcp host bgp_peer eq bgp host router_ipv6
deny ipv6 any INTERNAL_INFRASTUCTURE_ADDRESSES
permit ipv6 any YOUR_CIDR_BLOCK_IPV6
```

Reunareitittimellä kannattaa pudottaa myös *hop-by-hop*-optio-otsikot ja tyypin 0 reititysotsikot, ottaa URPF-toiminnallisuus (*unicast reverse path forwarding*) käyttöön sekä ICMPv6-uudelleenohjausviestit pois käytöstä: [123, 124]

```
deny hbh any any
deny ipv6 any any routing-type 0
ipv6 verify unicast reverse-path
no ipv6 redirects
```

Lopuksi RA-viestien lähettäminen kannattaa ottaa palvelinverkkosegmenteissä pois käytöstä tässä työssä aiemmin esiteltyjen syiden takia: [154]

```
ipv6 nd ra suppress all
```

7.2.3 Palvelinsegmentit ja linkkiverkot

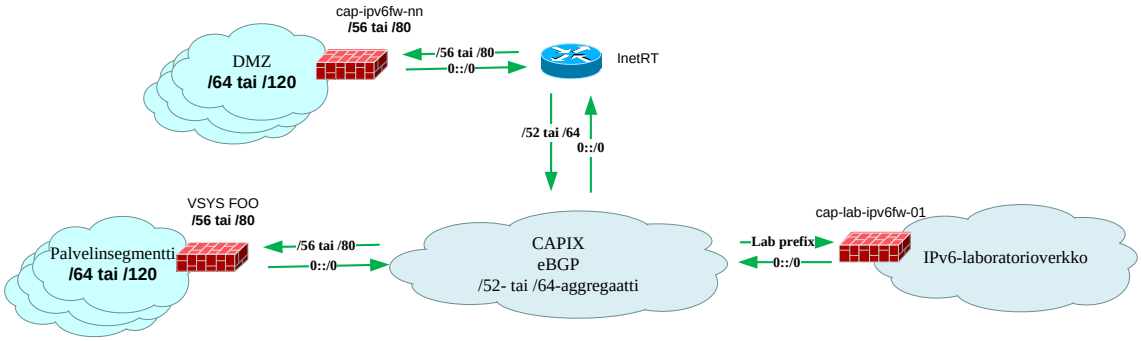
Tässä työssä on käyty läpi /64- ja /120-prefiksien käyttämistä palvelinverkkosegmenteissä. Kummankin käyttämiseen liittyy omat kysymyksensä, mutta /120-palvelinverkkosegmenteissä on se ongelma, että niiden muuttaminen myöhemmin /64-segmenteiksi voi olla haasteellista. Tässä työssä suositeltava ratkaisu onkin käyttää /64-palvelinverkkosegmenttejä, mutta rajata liikenne reitittimellä tai palomuurilla vain esim. niiden ensimmäiseen /120-segementtiin.

/126- ja /128-prefikseistä ja siitä, että niitä voidaan käyttää linkkiverkoissa ja *loopback*-osoitteina ollaan oltu samaa mieltä jo pitkään, mutta /127-prefiksin käyttöä linkkiverkoissa pidettiin pitkään huonona käytäntönä [125, 181]. Nykyään sen käyttöä kuitenkin suositellaan, ja parhaat käytännöt ovat seuraavat: [182, 183, 184]

- /64: Voidaan käyttää palvelinverkkosegmenteissä, jos suojaudutaan asianmukaisesti esim. NDP-tulvitushyökkäyksiltä.
- /120: Voidaan käyttää palvelinverkkosegmenteissä, jos reitittimien TCAM-rajoitukset eivät tule vastaan.
- /124: Voidaan käyttää linkkiverkoissa, jos tarvitaan useampaa kuin neljää osoitetta
- /126: Voidaan käyttää linkkiverkoissa, jos neljä osoitetta riittää
- /127: Voidaan käyttää linkkiverkoissa, jos kaksi osoitetta riittää
- /128: Voidaan käyttää *loopback*-osoitteille.

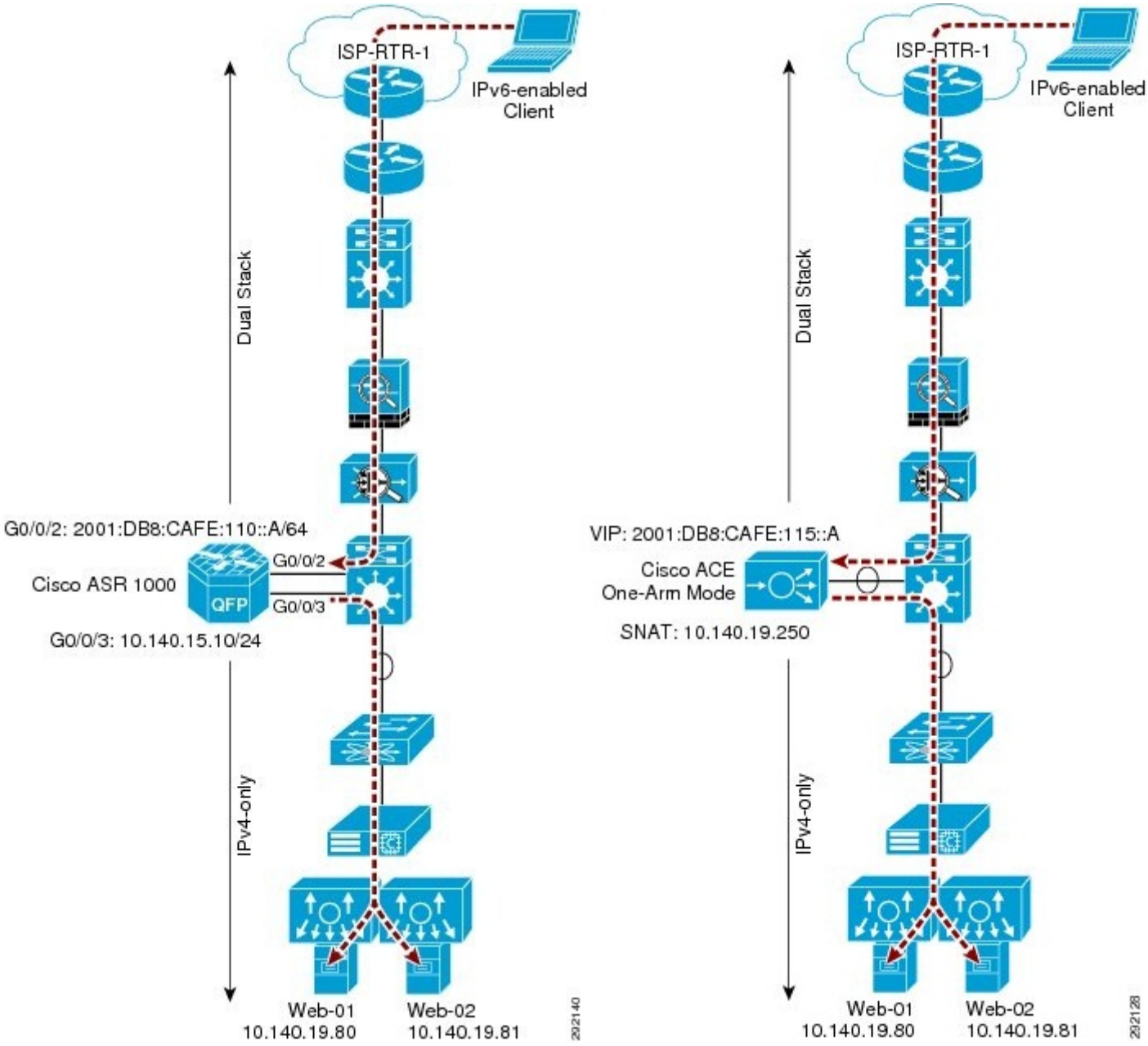
7.3 Mitä seuraavaksi?

Seuraava looginen askel laajemmalle IPv6- käyttöönotolle Capgeminin konesaliverkossa on CAPIX-vrf:n tekeminen. Tällöin fyysinen laboratoriopalomuuri voidaan terminoida siihen Internet-reitittimien sijaan ja Capgeminin eri palomuureja voidaan alkaa liittää siihen aiemmin esitettyjen suunnitelmien mukaisesti. DMZ-palveluita voidaan tarjota IPv6-protokollalla virtuaalista Firefly-palomuuria hyödyntäen, kuten kuvassa 71 on esitetty.



Kuva 71: CAPIX-vrf.

Jos jokin Capgeminin asiakas haluaa käyttää IPv6-palveluita Internetistä ennen kuin sen palvelimet ovat IPv6-palvelinsegmentissä, voidaan Capgeminin välityspalvelimet konfiguroida tekemään IPv4-IPv6-osoitteenmuunnosta. Jos taas jonkin asiakkaan Capgeminin konesalissa olevia palveluita halutaan käyttää IPv6:lla ennen kuin palvelimet ovat IPv6-verkossa, voidaan käyttää NAT64- tai SLB64-mekanismeja. SLB64 on sinänsä parempi ja yksinkertaisempi vaihtoehto, että siinä Cisco ACE -kuormanjakaja tekee sekä IPv6-IPv4-osoitteenmuunnoksen että kuorman jakamisen palvelimille. NAT64 tekee pelkän osoitteenmuunnoksen ja kuormanjakaja on konfiguroitava erikseen. Nämä mekanismit on kuvattu kuvassa 72. [123]



Kuva 72: NAT64 & SLB64. [123]

7.4 Yhteenveto

Tämän diplomityön tavoitteena oli ottaa IPv6-protokolla käyttöön Capgeminin konesaliverkossa. Syyt Capgeminin IPv6-käyttöönottoon olivat puhtaasti liiketoiminnalliset: IPv4- ja IPv6-protokollat eivät lähtökohtaisesti ole yhteensopivia keskenään, joten IPv6-protokollaa käyttävälle käyttäjälle on pystyttävä tarjoamaan IPv6-palveluita. Kuten tässä työssä on kerrottu, IPv6-protokollan käyttö on parin viime vuoden aikana yleistynyt Internetissä merkittävästi, joten palveluntarjoajien kuten Capgeminin on oltava valmiina tarjoamaan palveluitaan myös IPv6-protokollalla. IPv6-käyttöönotto aloitettiin yhdistämällä ensin Capgeminin konesaliverkko Internetiin IPv6-protokollalla ja rakentamalla sen jälkeen IPv6-testiverkko Capgeminin laboratorioon. Lopuksi tässä työssä kehitettiin konsepti, jolla voidaan provisioida IPv6-protokollalla toimiva www-palvelu Capgeminin konesaliverkossa mahdollisimman helposti ja kustannustehokkaasti.

Tämän työn toisessa luvussa kerrottiin ensin siitä, mitkä IPv4-protokollan ongelmat ovat ja miksi IPv6-protokollaa käytetään ensin pitkään IPv4-protokollan rinnalla ennen kuin se lopulta syrjäyttää sen. Tämän jälkeen kolmannessa luvussa itse IPv6- ja sen tukiprotokollat kuten ICMPv6 ja NDP esiteltiin IETF:n RFC-dokumenttien ja kirjallisuuden pohjalta. Neljännessä luvussa sivuttiin IPv6-protokollan tietoturvaa lähinnä Capgeminin IPv6-käyttöönottoon liittyen ja kerrotiin, millaisia mekanismeja IPv4-IPv6-transitioon on olemassa. Näitä transitiomekanismeja käyttämällä voidaan IPv4- ja IPv6-protokollaa käyttää verkossa rinnakkain niin, että IPv6-protokollaa ei tarvitse ottaa heti käyttöön koko verkossa. Viidennessä luvussa esiteltiin ensin tyypillinen palvelinkeskuksen konesaliverkon verkkotopologia ja näytettiin sen jälkeen, kuinka Capgeminin konesaliverkko on rakennettu itse konesali(lähi)verkon, palomuurien, Internet-liitynnän ja nimipalvelimien osalta. Luvussa käytiin lyhyesti läpi ISC:n BIND-nimipalveluohjelmiston, Juniper Networksin Junos- ja Ciscon IOS- sekä NX-OS-käyttöjärjestelmien konfiguroimista seuraavassa luvussa tehtävää IPv6-käyttöönottoa varten. Kuudennessa luvussa tehtiin itse IPv6-käyttöönotto Capgeminin konesaliverkossa. Käyttöönotto aloitettiin liittämällä Capgeminin konesaliverkko Internetiin IPv6-protokollalla. Tämä oli tarkoitus tehdä huhtikuun kuukausittaisella huoltokatkolla, mutta Capgeminin Internet-reitittimien ohjelmistoversio oli liian vanha, eikä siinä ollut tukea BGP:n IPv6-osoiteperheelle. Reitittimet päivitettiin toukokuun huoltokatkolla, ja 22.5.2013 lähtien Capgeminin konesaliverkko on ollut saavutettavissa Internetistä IPv4-protokollan lisäksi myös IPv6-protokollalla. Internetiin liittämisen jälkeen Capgeminin laboratorioon rakennettiin IPv6-testiverkko, johon liitettiin myös kaksi testipalvelinta. Toinen näistä oli VMware ESXi -palvelin, jolle asennettiin lukuisia virtuaalipalvelimia. Lopuksi luvussa esiteltiin kaksi eri suunnitelmaa Capgeminin IPv6-osoitteistukselle. Viimeisessä, seitsemännessä luvussa käytiin läpi IPv6-käyttöönoton ja -testiverkon rakentamisen tulokset, haasteet ja ongelmat sekä vedettiin yhteen parhaat käytännöt, joita palvelinkeskuksen konesaliverkon IPv6-käyttöönotossa kannattaa noudattaa. Lopuksi kerrottiin, mitkä ovat seuraavat askeleet IPv6-protokollan laajemmalle käyttöönotolle Capgeminin konesaliverkossa. Tämän diplomityön tekemisen ansiosta Capgeminilla on hyvät valmiudet tarjota asiakkailleen IPv6-palveluita.

Viitteet

[1] Postel, J. *Internet Protocol.* Internet Engineering Task Force, RFC 791, September 1981.

[2] *Free Pool of IPv4 Address Space Depleted.* Verkkodokumentti. Päivitetty 3.2.2011. Viitattu 12.10.2012. Saatavissa: http://www.nro.net/news/ipv4-free-pool-depleted.

[3] *RIPE NCC Begins to Allocate IPv4 Address Space From the Last /8.* Verkkodokumentti. Päivitetty 14.9.2012. Viitattu 12.10.2012. Saatavissa: http://www.ripe.net/internet-coordination/news/ripe-ncc-begins-to-allocate-ipv4-address-space-from-the-last-8.

[4] *IPv4 Address Allocation and Assignment Policies for the RIPE NCC Service Region.* Verkkodokumentti. Päivitetty 21.6.2013. Viitattu 22.7.2013. Saatavissa: http://www.ripe.net/ripe/docs/ripe-592.

[5] *Available Pool of Unallocated IPv4 Internet Addresses Now Completely Emptied.* Verkkodokumentti. Päivitetty 3.2.2011. Viitattu 12.10.2012. Saatavissa: http://www.icann.org/en/news/releases/release-03feb11-en.pdf.

[6] Deering, S. ja Hinden, R. *Internet Protocol, Version 6 (IPv6) Specification.* Internet Engineering Task Force, RFC 1883, December 1995.

[7] Deering, S. ja Hinden, R. *Internet Protocol, Version 6 (IPv6) Specification.* Internet Engineering Task Force, RFC 2460, December 1998.

[8] Bradner, S. *Key words for use in RFCs to Indicate Requirement Levels.* Internet Engineering Task Force, RFC 2119, March 1997.

[9] Cerf, V. *The Catenet Model for Internetworking.* DARPA/IPTO, IEN 48, July 1978.

[10] Postel, J. *Internet Protocol.* Internet Engineering Task Force, RFC 760, January 1980.

[11] Postel, J. *Assigned Numbers.* Internet Engineering Task Force, RFC 790, September 1981.

[12] Loshin, P. *IPv6: Theory, Protocol, and Practice.* 2. painos. Morgan Kaufmann, 2003.

[13] Bradner, S. ja Mankin, A. *The Recommendation for the IP Next Generation Protocol.* Internet Engineering Task Force, RFC 1752, January 1995.

[14] Fuller, V., Li, T., Yu, J. ja Varadhan, K. *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy.* Internet Engineering Task Force, RFC 1519, September 1993.

[15] Miller, M. A. *Implementing IPv6: Migrating to the Next Generation Internet Protocols.* M&T Books, 1998.

[16] Clark, D., Chapin, L., Cerf, V., Braden, R. ja Hobby, R. *Towards the Future Internet Architecture.* Internet Engineering Task Force, RFC 1287, December 1991.

[17] Gross, P. ja Almquist, P. *IESG Deliberations on Routing and Addressing.* Internet Engineering Task Force, RFC 1380, November 1992.

[18] Callon, R. *TCP and UDP with Bigger Addresses (TUBA), A Simple Proposal for Internet Addressing and Routing.* Internet Engineering Task Force, RFC 1347, June 1992.

[19] Ullmann, R. *TP/IX: The Next Internet.* Internet Engineering Task Force, RFC 1475, June 1993.

[20] Crocker, S. *The Process for Organization of Internet Standards Working Group (POISED).* Internet Engineering Task Force, RFC 1396, January 1993.

[21] Huitema, C. *IPv6: The New Internet Protocol.* 2. painos. Prentice Hall PTR, 1998.

[22] Meuller, M. L. *Ruling the Root: Internet Governance and the Taming of Cyberspace.* The MIT Press, 2002.

[23] McGovern, M. ja Ullmann, R. *CATNIP: Common Architecture for the Internet.* Internet Engineering Task Force, RFC 1707, October 1994.

[24] Hinden, R., Deering, S. ja Crocker, D. *IPv7 Criteria Analysis for IP Address Encapsulation (IPAE) and the Simple Internet Protocol (SIP).* Internet Engineering Task Force, Internet-Draft, November 1992.

[25] Gilligan, R. E., Nordmark, E. ja Hinden, B. *IPAE: The SIPP Interoperability and Transition Mechanism.* Internet Engineering Task Force, Internet-Draft, November 1993.

[26] Hinden, R. *Simple Internet Protocol Plus White Paper.* Internet Engineering Task Force, RFC 1710, October 1994.

[27] Loshin, P. *IPv6 Clearly Explained.* AP Professional, 1999.

[28] Partridge, C. ja Kastenholz , F. *Technical Criteria for Choosing IP The Next Generation (IPng).* Internet Engineering Task Force, RFC 1726, December 1994.

[29] *Version Numbers.* Verkkodokumentti. Viitattu 25.7.2013. Saatavissa: http://www.iana.org/assignments/version-numbers/version-numbers.xhtml.

[30] Reed, D. P. *That Sneaky Exponential - Beyond Metcalfe's Law to the Power of Community Building.* Saatavissa: http://reed.com/dpr/locus/gfn/reedslaw.html.

[31] Briscoe, B., Odlyzko, A. ja Tilly, B. *Metcalfe's Law is Wrong.* IEEE Spectrum, 2006. Saatavissa: http://spectrum.ieee.org/computing/networks/metcalfes-law-is-wrong/0.

[32] *Internet usage statistics.* Verkkodokumentti. Päivitetty 6.10.2012. Viitattu 18.10.2012. Saatavissa: http://internetworldstats.com/stats.htm.

[33] *Google IPv6.* Verkkodokumentti. Viitattu 21.7.2013. Saatavissa: http://www.google.com/intl/en/ipv6/.

[34] Eisenmann, T., Parker, G. ja Alstyne, M. V. *Platform Networks - Core Concepts Executive Summary.* The MIT Center for Digital Business, 2007. Saatavissa: http://ebusiness.mit.edu/research/papers/2007.06_Eisenmann_Parker_Van%20Alstyne_Platform%20Networks_232.pdf.

[35] McFarland, S., Sambi, M., Sharma, N. ja Hooda, S. *IPv6 for Enterprise Networks.* Cisco Press, 2011.

[36] *Enterprise and Service Provider Router and Switch Markets Post Gains.* Viitattu 10.7.2013. Saatavissa: http://www.enterprisenetworkingplanet.com/netsysm/enterprise-and-service-provider-router-and-switch-markets-post-gains.html.

[37] *Performance-Comparison Testing of IPv4 and IPv6 Throughput and Latency on Key Cisco Router Platforms.* Verkkodokumentti. Viitattu 22.7.2013. Saatavissa: http://www.cisco.com/web/strategy/docs/gov/IPv6perf_wp1f.pdf.

[38] *IPv6 Myths.* Verkkodokumentti. Päivitetty 18.2.2011. Viitattu 26.7.2013. Saatavissa: http://blogs.cisco.com/security/ipv6-myths/.

[39] Loughney, J. *IPv6 Node Requirements.* Internet Engineering Task Force, RFC 4294, April 2006.

[40] Jankiewicz, E., Loughney, J. ja Narten, T. *IPv6 Node Requirements.* Internet Engineering Task Force, RFC 6434, December 2011.

[41] *IPv6 Address Allocation and Assignment Policy.* Verkkodokumentti. Viitattu 10.7.2013. Saatavissa: https://www.ripe.net/ripe/docs/ripe-589.

[42] *Address Space Managed by the RIPE NCC.* Verkkodokumentti. Viitattu 10.7.2013. Saatavissa: https://www.ripe.net/ripe/docs/ripe-555.

[43] *NAT64 and DNS64 in 30 minutes.* Verkkodokumentti. Viitattu 21.7.2013. Saatavissa: http://www.slideshare.net/IOSHints/nat64-and-dns64-in-30-minutes.

[44] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J. ja Lear, E. *Address Allocation for Private Internets.* Internet Engineering Task Force, RFC 1918, February 1996.

[45] Srisuresh, B. ja Egevang, K. *Traditional IP Network Address Translator (Traditional NAT).* Internet Engineering Task Force, RFC 3022, January 2001.

[46] Saltzer, J. H. et al. *End-to-End Arguments in System Design.* M.I.T. Laboratory for Computer Science, 1984. Saatavissa: http://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf.

[47] Rosenberg, J., Mahy, R., Matthews, P. ja Wing, D. *Session Traversal Utilities for NAT (STUN).* Internet Engineering Task Force, RFC 5389, October 2008.

[48] Mahy, R., Matthews, P. ja Rosenberg, J. *Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN).* Internet Engineering Task Force, RFC 5766, April 2010.

[49] Rosenberg, J. *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols.* Internet Engineering Task Force, RFC 5245, April 2010.

[50] Hain, T. *Architectural Implications of NAT.* Internet Engineering Task Force, RFC 2993, November 2000.

[51] Holdrege, M. ja Srisuresh, P. *Protocol Complications with the IP Network Address Translator.* Internet Engineering Task Force, RFC 3027, January 2001.

[52] Van de Velde, G., Hain, T., Droms, R., Carpenter, B. ja Klein, E. *Local Network Protection for IPv6.* Internet Engineering Task Force, RFC 4864, May 2007.

[53] Grossetete, P., Popoviciu, C. P. ja Wettling, F. *Global IPv6 Strategies: From Business Analysis to Operational Planning.* Cisco Press, 2008.

[54] *Are We at a Tipping Point?* Verkkodokumentti. Viitattu 19.10.2012. Saatavissa: http://www.gregorybalestrero.com/are-we-at-a-tipping-point/.

[55] Gates, B. *Content Is King.* Microsoft Corporation, 1996. Saatavissa: http://web.archive.org/web/20010126005200/http://www.microsoft.com/billgates/columns/1996essay/essay960103.asp.

[56] Christensen, C. *The Innovator's Dilemma.* Harvard Business School Press, 1997.

[57] *Grading the top 10 U.S. carriers in the first quarter of 2013.* Verkkodokumentti. Viitattu 7.7.2013. Saatavissa: http://www.fiercewireless.com/special-reports/grading-top-10-us-carriers-first-quarter-2013.

[58] *IPv6 at Verizon Wireless.* Verkkodokumentti. Viitattu 7.7.2013. Saatavissa: http://conference.apnic.net/__data/assets/pdf_file/0017/50813/vzw_apnic_13462152832-2.pdf.

[59] *Over 25% of Verizon Wireless Traffic Is Now Over IPv6.* Verkkodokumentti. Viitattu 7.7.2013. Saatavissa: http://www.internetsociety.org/deploy360/blog/2013/04/over-25-of-verizon-wireless-traffic-is-now-over-ipv6/.

[60] *Archive: 2011 World IPV6 Day.* Verkkodokumentti. Viitattu 19.10.2012. Saatavissa: http://www.internetsociety.org/ipv6/archive-2011-world-ipv6-day.

[61] *World IPv6 Day begins 24 hours from now. Websites, start your engines.* Verkkodokumentti. Viitattu 6.7.2013. Saatavissa: http://googleblog.blogspot.fi/2011/06/world-ipv6-day-begins-24-hours-from-now.html.

[62] *Exciting Results from World IPv6 Day.* Verkkodokumentti. Viitattu 6.7.2013. Saatavissa: https://www.facebook.com/note.php?note_id=10150198443513920.

[63] *World IPv6 Day: A Watershed Moment Towards a New Internet Protocol.* Verkkodokumentti. Viitattu 6.7.2013. Saatavissa: http://blogs.cisco.com/news/world-ipv6-day-a-watershed-moment-towards-a-new-internet-protocol/.

[64] *Check Point's World IPv6 Day Experience.* Verkkodokumentti. Viitattu 6.7.2013. Saatavissa: http://www.InternetEngineeringTaskForce.org/proceedings/81/slides/v6ops-2.pdf.

[65] *World IPv6 Lanch.* Verkkodokumentti. Viitattu 19.10.2012. Saatavissa: http://www.worldipv6launch.org/.

[66] *IPv6 momentum infographic.* Verkkodokumentti. Viitattu 7.7.2013. Saatavissa: http://www.worldipv6launch.org/infographic/.

[67] Aboba, B., Davies, E. ja Thaler, D. *Multiple Encapsulation Methods Considered Harmful.* Internet Engineering Task Force, RFC 4840, April 2007.

[68] Borman, D., Deering, S. ja Hinden, R. *IPv6 Jumbograms.* Internet Engineering Task Force, RFC 2675, August 1999.

[69] Mathis, M., Heffner, J. ja Chandler, B. *Fragmentation Considered Very Harmful.* Internet Engineering Task Force, Internet-Draft, July 2004.

[70] McCann, J., Deering, S. ja Mogul, J. *Path MTU Discovery for IP version 6.* Internet Engineering Task Force, RFC 1981, August 1996.

[71] Hu, Q. ja Carpenter, B. *Survey of Proposed Use Cases for the IPv6 Flow Label.* Internet Engineering Task Force, RFC 6294, June 2011.

[72] Amante, S., Carpenter, B., Jiang, S. ja Rajahalme, J. *IPv6 Flow Label Specification.* Internet Engineering Task Force, RFC 6437, November 2011.

[73] Partridge, C. *Using the Flow Label Field in IPv6.* Internet Engineering Task Force, RFC 1809, June 1995.

[74] Blake, S. *Use of the IPv6 Flow Label as a Transport-Layer Nonce to Defend Against Off-Path Spoofing Attacks.* Internet Engineering Task Force, Internet-Draft, October 2009.

[75] Nichols, K., Blake, S., Baker, F. ja Black, D. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.* Internet Engineering Task Force, RFC 2474, December 1998.

[76] Grossman, D. *New Terminology and Clarifications for Diffserv.* Internet Engineering Task Force, RFC 3260, April 2002.

[77] Ramakrishnan, K., Floyd, S. ja Black, D. *The Addition of Explicit Congestion Notification (ECN) to IP.* Internet Engineering Task Force, RFC 3168, September 2001.

[78] Abley, J., Savola, P. ja Neville-Neil, G. *Deprecation of Type 0 Routing Headers in IPv6.* Internet Engineering Task Force, RFC 5095, December 2007.

[79] Kent, S. *IP Authentication Header.* Internet Engineering Task Force, RFC 4302, December 2005.

[80] Kent, S. *IP Encapsulating Security Payload (ESP).* Internet Engineering Task Force, RFC 4303, December 2005.

[81] Perkins, C., Johnson, D. ja Arkko, J. *Mobility Support in IPv6.* Internet Engineering Task Force, RFC 6275, July 2011.

[82] Hinden, R. ja Deering, S. *IPv6 Addressing Architecture.* Internet Engineering Task Force, RFC 4291, February 2006.

[83] Huitema, C. ja Carpenter, B. *Deprecating Site Local Addresses.* Internet Engineering Task Force, RFC 3879, September 2004.

[84] Beijnum, I. v. *Running IPv6.* Apress, 2006.

[85] Haberman, B. ja Thaler, D. *Unicast-Prefix-based IPv6 Multicast Addresses.* Internet Engineering Task Force, RFC 3306, August 2002.

[86] Savola, P. ja Haberman, B. *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address.* Internet Engineering Task Force, RFC 3956, November 2004.

[87] Haberman, B. *Allocation Guidelines for IPv6 Multicast Addresses.* Internet Engineering Task Force, RFC 3307, August 2002.

[88] Narten, T., Nordmark, E., Simpson, W. ja Soliman, H. *Neighbor Discovery for IP version 6 (IPv6).* Internet Engineering Task Force, RFC 4861, September 2007.

[89] Hinden, R. ja Haberman, B. *Unique Local IPv6 Unicast Addresses.* Internet Engineering Task Force, RFC 4193, October 2005.

[90] Matthews, P. *Design Choices for IPv6 Networks.* Internet Engineering Task Force, Internet-Draft, February 2013.

[91] Behringer, M. ja Vyncke, E. *Using Only Link-Local Addressing Inside an IPv6 Network.* Internet Engineering Task Force, Internet-Draft, December 2013.

[92] Malkin, G. ja Minnear, R. *RIPng for IPv6.* Internet Engineering Task Force, RFC 2080, January 1997.

[93] Blanchet, M. *Migrating to IPv6: A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks.* Wiley, 2006.

[94] Coltun, R., Ferguson, D., Moy, J. ja Lindem, A. *OSPF for IPv6.* Internet Engineering Task Force, RFC 5340, July 2008.

[95] ISO/IEC 10589. *Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473).* ISO/IEC, 2002.

[96] Callon, R. *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments.* Internet Engineering Task Force, RFC 1195, December 1990.

[97] Hopps, C. *Routing IPv6 with IS-IS.* Internet Engineering Task Force, RFC 5308, October 2008.

[98] Bates, T., Chandra, R., Katz, D. ja Rekhter Y. *Multiprotocol Extensions for BGP-4.* Internet Engineering Task Force, RFC 4760, January 2007.

[99] Marques, P. ja Dupont, F. *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing.* Internet Engineering Task Force, RFC 2545, March 1999.

[100] Postel, J. *Internet Control Message Protocol.* Internet Engineering Task Force, RFC 792, September 1981.

[101] Conta, A., Deering, S. ja Gupta, M. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.* Internet Engineering Task Force, RFC 4443, March 2006.

[102] Thomson, S., Narten, T. ja Jinmei, T. *IPv6 Stateless Address Autoconfiguration.* Internet Engineering Task Force, RFC 4862, September 2007.

[103] Narten, T., Draves, R. ja Krishnan, S. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6.* Internet Engineering Task Force, RFC 4941, September 2007.

[104] *Changes to IPv6 in Windows Vista and Windows Server 2008.* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://technet.microsoft.com/en-us/library/bb878121.aspx.

[105] *Don't use IPv6 RA on server LANs.* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://blog.ioshints.info/2012/10/dont-use-ipv6-ra-on-server-lans.html.

[106] Droms, R. *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6.* Internet Engineering Task Force, RFC 3736, April 2004.

[107] *DHCP for IPv6.* Verkkodokumentti. Viitattu 21.7.2013. Saatavissa: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6554/ps6600/ps6641/aag_C45-456070_v2.pdf.

[108] Troan, O. ja Droms, R. *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6.* Internet Engineering Task Force, RFC 3633, December 2003.

[109] Miyakawa, S. ja Droms, R. *Requirements for IPv6 Prefix Delegation.* Internet Engineering Task Force, RFC 3769, June 2004.

[110] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C. ja Carney, M. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6).* Internet Engineering Task Force, RFC 3315, July 2003.

[111] Jeong, J., Park, S., Beloeil, L. ja Madanapalli, S. *IPv6 Router Advertisement Options for DNS Configuration.* Internet Engineering Task Force, RFC 6106, November 2010.

[112] Dec, W., Mrugalski, T., Sun, T., Sarikaya, B. ja Matsumoto, A. *DHCPv6 Route Options.* Internet Engineering Task Force, Internet-Draft, August 2012.

[113] *DHCPV6+SLAAC+RA = DHCPV4.* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://blog.ioshints.info/2011/02/dhcpv6slaacra-dhcpv4.html.

[114] *IPv6 and DHCPv6: Does SLAAC Have A Future?* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://blog.geoff.co.uk/2011/08/02/ipv6-automated-network-configuration/.

[115] Chown, T. ja Venaas, S. *Rogue IPv6 Router Advertisement Problem Statement.* Internet Engineering Task Force, RFC 6104, February 2011.

[116] *Catalyst Integrated Security Features (CISF).* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://www.cisco.com/web/strategy/docs/gov/turniton_cisf.pdf.

[117] *IPv6 First Hop Security (FHS).* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6553/aag_c45-707354.pdf.

[118] Arkko, J., Kempf, J., Zill, B. ja Nikander, P. *SEcure Neighbor Discovery (SEND).* Internet Engineering Task Force, RFC 3971, March 2005.

[119] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C. ja Mohacsi, J. *IPv6 Router Advertisement Guard.* Internet Engineering Task Force, RFC 6105, February 2011.

[120] *IPv6 First Hop Security – Protecting Your IPv6 Access Network.* Verkkodokumentti. Viitattu 10.7.2013. Saatavissa: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6553/whitepaper_c11-602135.html.

[121] *BGP Best Current Practices.* Verkkodokumentti. Viitattu 10.7.2013. Saatavissa: ftp://ftp-eng.cisco.com/pfs/isp-workshops/BGP_Presentations/3-best-practices.pdf.

[122] *Protecting Your Core: Infrastructure Protection Access Control Lists.* Verkkodokumentti. Viitattu 10.7.2013. Saatavissa: http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801a1a55.shtml.

[123] *Deploying IPv6 in the Internet Edge.* Verkkodokumentti. Viitattu 17.7.2013. Saatavissa: http://www.cisco.com/en/US/docs/solutions/Enterprise/Borderless_Networks/Internet_Edge/InternetEdgeIPv6.html.

[124] Nikander, P., Kempf, J. ja Nordmark, E. *IPv6 Neighbor Discovery (ND) Trust Models and Threats.* Internet Engineering Task Force, RFC 3756, May 2004.

[125] Van de Velde, G., Popoviciu, C., Chown, T., Bonness, O. ja Hahn, C. *IPv6 Unicast Address Assignment Considerations.* Internet Engineering Task Force, RFC 5375, December 2008.

[126] *IPv6 NDP Table Exhaustion Attack.* Verkkodokumentti. Viitattu 11.7.2013. Saatavissa: http://inconcepts.biz/~jsw/IPv6_NDP_Exhaustion.pdf.

[127] *Cisco Nexus 5000 Series Configuration Limits for Cisco NX-OS Release 5.2(1)N1(1).* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://www.cisco.com/en/US/docs/switches/datacenter/nexus5000/sw/configuration_limits/limits_521/nexus_5000_config_limits_521.html.

[128] Nordmark, E. ja Gilligan, R. *Basic Transition Mechanisms for IPv6 Hosts and Routers.* Internet Engineering Task Force, RFC 4213, October 2005.

[129] Baker, F., Li, X., Bao, C. ja Yin, K. *Framework for IPv4/IPv6 Translation.* Internet Engineering Task Force, RFC 6144, April 2011.

[130] Carpenter, B. ja Moore, K. *Connection of IPv6 Domains via IPv4 Clouds.* Internet Engineering Task Force, RFC 3056, February 2001.

[131] Despres, R. *IPv6 Rapid Deployment on IPv4 Infrastructures (6rd).* Internet Engineering Task Force, RFC 5569, January 2010.

[132] Townsley, W. ja Troan, O. *IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – Protocol Specification.* Internet Engineering Task Force, RFC 5969, August 2010.

[133] Carpenter, B. ja Jung, C. *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels.* Internet Engineering Task Force, RFC 2529, March 1999.

[134] Templin, F., Gleeson, T. ja Thaler, D. *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP).* Internet Engineering Task Force, RFC 5214, March 2008.

[135] Durand, A., Fasano, P., Guardini, I. ja Lento, D. *IPv6 Tunnel Broker.* Internet Engineering Task Force, RFC 3053, January 2001.

[136] Blanchet, M. ja Parent, F. *IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP).* Internet Engineering Task Force, RFC 5572, February 2010.

[137] *Hurricane Electric Free IPv6 Tunnel Broker.* Verkkodokumentti. Viitattu 26.7.2013. Saatavissa: http://tunnelbroker.net/.

[138] *IPv6 Deployment & Tunnel Broker.* Verkkodokumentti. Viitattu 26.7.2013. Saatavissa: http://www.sixxs.net/.

[139] *Teredo Overview.* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://technet.microsoft.com/en-us/library/bb457011.aspx.

[140] Rosenberg, J., Weinberger, J., Huitema, C. ja Mahy, R. *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs).* Internet Engineering Task Force, RFC 3489, March 2003.

[141] Huitema, C. *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs).* Internet Engineering Task Force, RFC 4380, February 2006.

[142] Li, X., Bao, C. ja Baker, F. *IP/ICMP Translation Algorithm.* Internet Engineering Task Force, RFC 6145, April 2011.

[143] Tsirtsis, G. ja Srisuresh, P. *Network Address Translation - Protocol Translation (NAT-PT).* Internet Engineering Task Force, RFC 2766, February 2000.

[144] Aoun C. ja Davies, E. *Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status.* Internet Engineering Task Force, RFC 4966, July 2007.

[145] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M. ja Li, X. *IPv6 Addressing of IPv4/IPv6 Translators.* Internet Engineering Task Force, RFC 6052, October 2010.

[146] Hagino, J. ja Yamamoto, K. *An IPv6-to-IPv4 Transport Relay Translator.* Internet Engineering Task Force, RFC 3142, June 2001.

[147] Fujisawa, K. ja Onoe, A. *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers.* Internet Engineering Task Force, RFC 6146, October 2011.

[148] Bagnulo, M., Sullivan, A., Matthews, P., Beijnum, I. v. *DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers.* Internet Engineering Task Force, RFC 6147, April 2011.

[149] *SOLIDserver - IPAM-DNS-DHCP Appliances - EfficientIP.* Verkkodokumentti. Viitattu 1.7.2013. Saatavissa: http://efficientip.com/solidserver/solidserver.

[150] *DNS for Rocket Scientists.* Verkkodokumentti. Päivitetty 21.9.2011. Viitattu 12.10.2012. Saatavissa: http://www.zytrax.com/books/dns/.

[151] *Infonetics & IDC: Worldwide Carrier Ethernet Switch & Router Results and Market Forecasts.* Verkkodokumentti. Viitattu 7.7.2013. Saatavissa: http://community.comsoc.org/blogs/alanweissberger/infonetics-idc-worldwide-carrier-ethernet-switch-router-results-and-market-for.

[152] *2013 Gartner Magic Quadrant for Enterprise Network Firewalls.* Verkkodokumentti. Viitattu 7.7.2013. Saatavissa: http://connect.paloaltonetworks.com/gartner-mq-2013.

[153] Dunmore, M. *An IPv6 Deployment Guide.* The 6NET Consortium, 2005.

[154] *Cisco IOS IPv6 Command Reference.* Verkkodokumentti. Viitattu 14.7.2013. Saatavissa: http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/command/ipv6-cr-book.pdf.

[155] *IPv6 Configuration Guide, Cisco IOS Release 12.4T.* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/configuration/12-4t/ipv6-12-4t-book.pdf.

[156] *Securing IPv6.* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://blogs.cisco.com/security/securing-ipv6/.

[157] *The Total Economic Impact Of Juniper Networks' Junos Network Software.* Forrester Consulting, 2011. Saatavissa: http://www.juniper.net/us/en/local/pdf/analyst-reports/2000455-en.pdf.

[158] Grundemann, C. *Day One: Exploring IPv6.* Junos Networking Technologies Series, 2010. Saatavissa: http://forums.juniper.net/jnet/attachments/jnet/Day1Books/12/10/Exploring%20IPv6.pdf (vaatii rekisteröinnin).

[159] *Enabling Flow-Based Processing for IPv6 Traffic.* Verkkodokumentti. Viitattu 24.7.2013. Saatavissa: http://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/topic-45426.html.

[160] *DNS server survey.* Verkkodokumentti. Päivitetty 23.5.2004. Viitattu 12.10.2012. Saatavissa: http://mydns.bboy.net/survey/.

[161] Thomson, S. ja Huitema, C. *DNS Extensions to support IP version 6.* Internet Engineering Task Force, RFC 1886, December 1995.

[162] Thomson, S., Huitema, C., Ksinant, V. ja Souissi, M. *DNS Extensions to support IP version 6.* Internet Engineering Task Force, RFC 3596, October 2003.

[163] Liu, C. *DNS and BIND on IPv6.* O'Reilly Media, 2011.

[164] *AS41701 IPv4 Route Propagation.* Verkkodokumentti. Viitattu 7.6.2013. Saatavissa: http://bgp.he.net/AS41701#_graph4.

[165] *AS41701 IPv6 Route Propagation.* Verkkodokumentti. Viitattu 7.6.2013. Saatavissa: http://bgp.he.net/AS41701#_graph6.

[166] *IPv6: IPv6 / IPv4 Comparative Statistics.* Verkkodokumentti. Viitattu 7.6.2013. Saatavissa: http://bgp.potaroo.net/v6/v6rpt.html.

[167] *HSRP: Global IPv6 Address.* Verkkodokumentti. Viitattu 7.6.2013. Saatavissa: http://www.cisco.com/en/US/docs/ios-xml/ios/ipapp_fhrp/configuration/xe-3se/3850/ip6-fhrp-hsrp-global.pdf.

[168] *VLAN configuration on Ubuntu (Debian).* Verkkodokumentti. Viitattu 7.6.2013. Saatavissa: http://www.mysidenotes.com/2007/08/17/vlan-configuration-on-ubuntu-debian/.

[169] *Using Google Public DNS.* Verkkodokumentti. Viitattu 7.6.2013. Saatavissa: https://developers.google.com/speed/public-dns/docs/using.

[170] *The Apache HTTP Server Project.* Verkkodokumentti. Viitattu 1.7.2013. Saatavissa: http://httpd.apache.org/.

[171] *JunosV Firefly.* Verkkodokumentti. Viitattu 26.6.2013. Saatavissa: http://www.juniper.net/support/downloads/?p=firefly.

[172] *Introduction to the ASA 1000V.* Verkkodokumentti. Viitattu 26.6.2013. Saatavissa: http://www.cisco.com/en/US/docs/security/asa/asa87/configuration/guide/intro_intro.html.

[173] *vGW Series.* Verkkodokumentti. Viitattu 26.6.2013. Saatavissa: http://www.juniper.net/us/en/products-services/security/vgw-series/.

[174] *Understanding the vGW Security VM.* Verkkodokumentti. Viitattu 29.6.2013. Saatavissa: http://www.juniper.net/techpubs/en_US/vgw5.5/topics/concept/security-vgw-security-vm.html.

[175] *Understanding the vGW Series Kernel Module.* Verkkodokumentti. Viitattu 29.6.2013. Saatavissa: http://www.juniper.net/techpubs/en_US/vgw5.5/topics/concept/security-vgw-kernel-module.html.

[176] *VMware vCenter Server Virtualization & Server Management Software.* Verkkodokumentti. Viitattu 1.7.2013. Saatavissa: http://www.vmware.com/products/vcenter-server/overview.html.

[177] *Cisco Nexus 1000V Series Switches Data Sheet.* Verkkodokumentti. Viitattu 3.7.2013. Saatavissa: http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9902/data_sheet_c78-492971.html.

[178] *VMware Virtual Networking Concepts.* Verkkodokumentti. Viitattu 1.7.2013. Saatavissa: http://www.vmware.com/files/pdf/virtual_networking_concepts.pdf.

[179] *How to Deploy a Nexus 1000v lab with a single ESX host.* Verkkodokumentti. Viitattu 29.6.2013. Saatavissa: https://learningnetwork.cisco.com/servlet/JiveServlet/previewBody/17233-102-1-67123/How%20to%20Deploy%20a%20Nexus%201000v%20lab%20with%20a%20single%20ESX%20host.pdf.

[180] *New Nexus 1000V Free-mium Pricing Model.* Verkkodokumentti. Viitattu 3.7.2013. Saatavissa: http://blogs.cisco.com/datacenter/new-nexus-1000v-free-mium-pricing-model/.

[181] Savola, P. *Use of /127 Prefix Length Between Routers Considered Harmful.* Internet Engineering Task Force, RFC 3627, Semptember 2003.

[182] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L. ja Narten, T. *Using 127-Bit IPv6 Prefixes on Inter-Router Links.* Internet Engineering Task Force, RFC 6164, April 2011.

[183] *BEST CURRENT OPERATIONAL PRACTICES – IPv6 Subnetting (v1).* Verkkodokumentti. Viitattu 22.7.2013. Saatavissa: http://www.ipbcop.org/wp-content/uploads/2012/02/BCOP-IPv6_Subnetting.pdf.

[184] *IPv6 Addressing Guide.* Verkkodokumentti. Viitattu 22.7.2013. Saatavissa: http://www.cisco.com/en/US/docs/solutions/SBA/February2013/Cisco_SBA_BN_IPv6AddressingGuide-Feb2013.pdf.

IPv6-pakettikaappaus (http://www.whatismyv6.com/)

```
No.     Time        Source                Destination           Protocol Length Info
     10 14.426311   ...                   2001:4860:4860::8888  DNS      98     Standard query AAAA
www.whatismyv6.com

Frame 10: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4860:4860::8888 (2001:4860:4860::8888)
User Datagram Protocol, Src Port: 45229 (45229), Dst Port: domain (53)
Domain Name System (query)

No.     Time        Source                Destination           Protocol Length Info
     11 14.443489   2001:4860:4860::8888  ...                   DNS      126    Standard query response AAAA
2001:4810::110

Frame 11: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4860:4860::8888 (2001:4860:4860::8888), Dst: ... (...)
User Datagram Protocol, Src Port: domain (53), Dst Port: 45229 (45229)
Domain Name System (response)

No.     Time        Source                Destination           Protocol Length Info
     12 14.443676   ...                   2001:4810::110        TCP      94     60628 > http [SYN]
Seq=0 Win=5760 Len=0 MSS=1440 SACK_PERM=1 TSval=109257457 TSecr=0 WS=64

Frame 12: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60628 (60628), Dst Port: http (80), Seq: 0, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     13 14.611891   2001:4810::110        ...                   TCP      98     http > 60628 [SYN, ACK]
Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 WS=2 TSval=3662395863 TSecr=109257457 SACK_PERM=1

Frame 13: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4810::110 (2001:4810::110), Dst: ... (...)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60628 (60628), Seq: 0, Ack: 1, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     14 14.611912   ...                   2001:4810::110        TCP      86     60628 > http [ACK]
Seq=1 Ack=1 Win=5760 Len=0 TSval=109257499 TSecr=3662395863

Frame 14: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60628 (60628), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     15 14.612108   ...                   2001:4810::110        HTTP     301    GET / HTTP/1.0

Frame 15: 301 bytes on wire (2408 bits), 301 bytes captured (2408 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60628 (60628), Dst Port: http (80), Seq: 1, Ack: 1, Len: 215
Hypertext Transfer Protocol

No.     Time        Source                Destination           Protocol Length Info
     16 14.779290   2001:4810::110        ...                   HTTP     684    HTTP/1.1 301 Moved Permanently
(text/html)

Frame 16: 684 bytes on wire (5472 bits), 684 bytes captured (5472 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4810::110 (2001:4810::110), Dst: ... (...)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60628 (60628), Seq: 1, Ack: 216, Len: 598
Hypertext Transfer Protocol
Line-based text data: text/html
```

```
No.     Time        Source                Destination           Protocol Length Info
     17 14.779314   2001:4810::110        ...                   TCP      86     http > 60628 [FIN, ACK]
Seq=599 Ack=216 Win=65664 Len=0 TSval=3662396031 TSecr=109257499

Frame 17: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4810::110 (2001:4810::110), Dst: ... (...)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60628 (60628), Seq: 599, Ack: 216, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     18 14.779367   ...                   2001:4810::110        TCP      86     60628 > http [ACK]
Seq=216 Ack=599 Win=6976 Len=0 TSval=109257541 TSecr=3662396031

Frame 18: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60628 (60628), Dst Port: http (80), Seq: 216, Ack: 599, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     19 14.779666   ...                   2001:4810::110        TCP      86     60628 > http [FIN, ACK]
Seq=216 Ack=600 Win=6976 Len=0 TSval=109257541 TSecr=3662396031

Frame 19: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60628 (60628), Dst Port: http (80), Seq: 216, Ack: 600, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     20 14.780079   ...                   2001:4860:4860::8888  DNS      94     Standard query AAAA whatismyv6.com

Frame 20: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4860:4860::8888 (2001:4860:4860::8888)
User Datagram Protocol, Src Port: 58144 (58144), Dst Port: domain (53)
Domain Name System (query)

No.     Time        Source                Destination           Protocol Length Info
     21 14.796781   2001:4860:4860::8888  ...                   DNS      122    Standard query response AAAA
2001:4810::110

Frame 21: 122 bytes on wire (976 bits), 122 bytes captured (976 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4860:4860::8888 (2001:4860:4860::8888), Dst: ... (...)
User Datagram Protocol, Src Port: domain (53), Dst Port: 58144 (58144)
Domain Name System (response)

No.     Time        Source                Destination           Protocol Length Info
     22 14.796924   ...                   2001:4810::110        TCP      94     60629 > http [SYN]
Seq=0 Win=5760 Len=0 MSS=1440 SACK_PERM=1 TSval=109257545 TSecr=0 WS=64

Frame 22: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60629 (60629), Dst Port: http (80), Seq: 0, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     23 14.944694   2001:4810::110        ...                   TCP      86     http > 60628 [ACK]
Seq=600 Ack=217 Win=65662 Len=0 TSval=3662396196 TSecr=109257541

Frame 23: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4810::110 (2001:4810::110), Dst: ... (...)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60628 (60628), Seq: 600, Ack: 217, Len: 0
```

```
No.     Time        Source                Destination           Protocol Length Info
     24 14.959935   2001:4810::110        ...                   TCP      98     http > 60629 [SYN, ACK]
Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 WS=2 TSval=3662396212 TSecr=109257545 SACK_PERM=1

Frame 24: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4810::110 (2001:4810::110), Dst: ... (...)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60629 (60629), Seq: 0, Ack: 1, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     25 14.959950   ...                   2001:4810::110        TCP      86     60629 > http [ACK]
Seq=1 Ack=1 Win=5760 Len=0 TSval=109257586 TSecr=3662396212

Frame 25: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60629 (60629), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     26 14.960038   ...                   2001:4810::110        HTTP     297    GET / HTTP/1.0

Frame 26: 297 bytes on wire (2376 bits), 297 bytes captured (2376 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60629 (60629), Dst Port: http (80), Seq: 1, Ack: 1, Len: 211
Hypertext Transfer Protocol

No.     Time        Source                Destination           Protocol Length Info
     27 15.137829   2001:4810::110        ...                   TCP      1454   [TCP segment of a reassembled PDU]

Frame 27: 1454 bytes on wire (11632 bits), 1454 bytes captured (11632 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4810::110 (2001:4810::110), Dst: ... (...)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60629 (60629), Seq: 1, Ack: 212, Len: 1368

No.     Time        Source                Destination           Protocol Length Info
     28 15.137849   2001:4810::110        ...                   HTTP     387    HTTP/1.1 200 OK  (text/html)

Frame 28: 387 bytes on wire (3096 bits), 387 bytes captured (3096 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4810::110 (2001:4810::110), Dst: ... (...)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60629 (60629), Seq: 1369, Ack: 212, Len: 301
[2 Reassembled TCP Segments (1669 bytes): #27(1368), #28(301)]
Hypertext Transfer Protocol
Line-based text data: text/html

No.     Time        Source                Destination           Protocol Length Info
     29 15.137854   2001:4810::110        ...                   TCP      86     http > 60629 [FIN, ACK]
Seq=1670 Ack=212 Win=65664 Len=0 TSval=3662396390 TSecr=109257586

Frame 29: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4810::110 (2001:4810::110), Dst: ... (...)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60629 (60629), Seq: 1670, Ack: 212, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     30 15.138002   ...                   2001:4810::110        TCP      86     60629 > http [ACK]
Seq=212 Ack=1369 Win=8640 Len=0 TSval=109257631 TSecr=3662396390

Frame 30: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60629 (60629), Dst Port: http (80), Seq: 212, Ack: 1369, Len: 0
```

```
No.     Time        Source                Destination           Protocol Length Info
     31 15.138025   ...                   2001:4810::110        TCP      86     60629 > http [ACK]
Seq=212 Ack=1670 Win=11392 Len=0 TSval=109257631 TSecr=3662396390

Frame 31: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60629 (60629), Dst Port: http (80), Seq: 212, Ack: 1670, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     32 15.140494   ...                   2001:4810::110        TCP      86     60629 > http [FIN, ACK]
Seq=212 Ack=1671 Win=11392 Len=0 TSval=109257631 TSecr=3662396390

Frame 32: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: DellComp_a4:86:c7 (00:08:74:a4:86:c7), Dst: Cisco_a8:61:83 (00:18:b9:a8:61:83)
Internet Protocol Version 6, Src: ... (...), Dst: 2001:4810::110 (2001:4810::110)
Transmission Control Protocol, Src Port: 60629 (60629), Dst Port: http (80), Seq: 212, Ack: 1671, Len: 0

No.     Time        Source                Destination           Protocol Length Info
     33 15.308479   2001:4810::110        ...                   TCP      86     http > 60629 [ACK]
Seq=1671 Ack=213 Win=65662 Len=0 TSval=3662396561 TSecr=109257631

Frame 33: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: Cisco_a8:61:83 (00:18:b9:a8:61:83), Dst: DellComp_a4:86:c7 (00:08:74:a4:86:c7)
Internet Protocol Version 6, Src: 2001:4810::110 (2001:4810::110), Dst: ... (...)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60629 (60629), Seq: 1671, Ack: 213, Len: 0
```