KYMENLAAKSON AMMATTIKORKEAKOULU
Tietotekniikka / Tietoverkkotekniikka

Erno Tolonen

VPN-RATKAISUT OPERAATTORIN SIIRTYESSÄ IPV6-YHTEYSKÄYTÄNTÖÖN

Opinnäytetyö 2011

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU
Tietotekniikka

TOLONEN, ERNO	VPN-ratkaisut operaattorin siirtyessä IPv6-yhteyskäytäntöön
Opinnäytetyö	54 sivua + 19 liitesivua
Työn ohjaaja	yliopettaja Martti Kettunen
Toimeksiantaja	SimuNet-hanke/KYMP OY
Tammikuu 2011	
Avainsanat	ipv6, mpls, vpn, migraatio

IPv4-osoitteiden loppumisen myötä on siirtyminen IPv6-protokollaan erittäin ajankohtainen aihe. Palveluntarjoajien palveluista VPN-yhteydet ovat yksi tärkeimmistä ja erittäin oleellinen askel migraatiossa. Oleellista on kuitenkin nykyisten palveluiden saatavuuden ja luotettavuuden säilyttäminen.

Tässä opinnäytetyössä perehdyttiin MPLS-teknologian toimintaan ja sen tarjoamiin mahdollisuuksiin IPv6 VPN -yhteyksien toteutuksessa. Myös muita, ei MPLS-kohtaisia IPv6 VPN -ratkaisuja tutkittiin ja niiden soveltuvuutta arvioitiin. Pääpaino työssä oli kuitenkin MPLS-pohjainen 6VPE-ratkaisu, joka mahdollistaa IPv6 VPN -yhteyksien toteuttamisen jo olemassa olevassa IPv4-pohjaisessa MPLS-verkossa.

Työssä toteutettiin 6VPE-pohjainen IPv6 VPN -ratkaisu SimuNet-verkkoon, joka on Kymenlaakson ammattikorkeakoulun ja alueen verkko-operaattorien yhteistyössä toteuttama laboratorioympäristössä toimiva, palveluntarjoajan verkkoa simuloiva verkkoympäristö.

Työn tuloksena saatiin aikaan toimiva 6VPE-ratkaisu SimuNet-verkkoon ja todettiin, että 6VPE-toteutus on erittäin hyvä ratkaisu, varsinkin jos käytössä on jo MPLS-verkko. Todettiin myös, että muut IPv6 VPN -ratkaisut ovat myös mahdollisia ja erityisesti tilanteissa, joissa ei haluta MPLS-teknologiaa käyttää, erittäin varteenotettavia.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU
University of Applied Sciences
Information Technology

TOLONEN, ERNO	VPN Solutions for Service Providers Migrating to IPv6
Bachelor's Thesis	54 pages + 19 pages of appendices
Supervisor	Martti Kettunen, Principal Lecturer
Commissioned by	SimuNet/KYMP OY
January 2011	
Keywords	ipv6, mpls, vpn, migration

The exhaustion of IPv4 address space has made migration to IPv6 a very hot topic. One of the most important services that internet service providers offer are VPN connections, which are also a crucial part of the IPv6 migration process. Maintaining reliability and availability of IPv4 services during the migration is also an extremely important matter.

This thesis focuses heavily on MPLS technology and the solutions it offers for implementing IPv6 VPN connections. In addition, other possible IPv6 VPN solutions, which do not use MPLS, are studied and their viability is considered. The main focus, however, is on 6VPE, a solution which enables IPv6 VPNs to run on an IPv4-based MPLS network.

The MPLS-based 6VPE solution was created in the SimuNet network, which is a network that aims to simulate a real Internet service provider's network operating in a laboratory environment. The SimuNet project is a result of co-operation between Kymenlaakso University of Applied Sciences and local service providers.

As the result of the thesis, a working 6VPE solution was accomplished, and it can be stated that it is the most convenient solution, if one is already running MPLS in one's network. In addition, it was found out that other IPv6 VPN solutions can be very viable, especially if for some reason MPLS is not desirable in one's network.

SISÄLLYS

LYHENNELUETTELO

6VPE	MPLS IPv6 VPN -teknologia
ATM	Asynchronous Transfer Mode, siirtotapa
AToM	Any Transport over MPLS
BGP	Border Gateway Protocol
CE	Customer Edge, asiakasverkon reunalaite
GRE	Generic Routing Encapsulation
IGP	Interior Gateway Protocol
IPsec	Internet Protocol Security
IPv4	Internet Protocol versio 4
IPv6	Internet Protocol versio 6
L2	Layer 2, OSI-mallin 2. kerros
L2TPv3	Layer 2 Tunneling Protocol versio 3
L3	Layer 3, OSI-mallin 3. kerros
LDP	Label Distribution Protocol
LFIB	Label Forwarding Information Base
LIB	Label Information Base
LSR	Label Switching Router, MPLS-reititin
MAC	Media Access Control, L2-osoite
mGRE	Multipoint GRE, monipistetunneli
MPLS	Multiprotocol Label Switching, lippuihin perustuva kytkentäteknologia
P	Provider Router, runkoverkon core-reititin
PE	Provider Edge, runkoverkon reunalaite
PPP	Point-to-point Protocol

VC	Virtual Circuit, virtuaalinen linkki
VPLS	Virtual Private LAN Service, MPLS-verkossa käytettävä teknologia lähiverkkojen emulointiin
VPN	Virtual Private Network
vpnv6	IPv6 osoite+BGP:n levittämät lipputiedot
VRF	Virtual Routing/Forwarding

1 JOHDANTO

Työn tavoitteena oli tutkia ja käytännössä testata MPLS-verkon päivittämistä IPv6 VPN -yhteensopivaksi. Työ toteutettiin SimuNet-hankkeelle, joka on Kymenlaakson ammattikorkeakoulun ja alueen verkko-operaattoreiden yhteinen hanke, jota EAKR (Euroopan Aluekehitysrahasto) tukee. SimuNet-hankkeessa pyritään mallintamaan verkko-operaattorin runkoverkkoa laboratorio-olosuhteissa.

SimuNet-hankkeeseen liittyviä opinnäytetöitä tehtiin samanaikaisesti muitakin. Riku Oinosen opinnäytetyö käsittelee L2-tason palveluita operaattoriverkossa, Riku Leinonen käsittelee työssään IPv6-palomuureja ja tietoturvaa, Lauri Sulevan työ käsittelee multicast-asioita ja Pasi Vanhalan ja Joni Hakkaraisen työssä käsitellään reduntanttisuutta kontrolleripohjaisessa langattomassa lähiverkossa.

Keskeisenä lähtökohtana työlle oli siirtyminen IP-protokollan versio neljästä versioon kuusi. IPv6-protokollaan siirtyminen on erittäin ajankohtaista, sillä vapaat IPv4-osoitteet ovat loppumassa lähitulevaisuudessa. Internet Assigned Numbers Authority (IANA) -järjestö, joka vastaa IP-osoitteiden jakamisen koordinoimisesta, jakoi helmikuussa 2011 viimeiset /8-prefixin osoitealueet alueellisille organisaatioille (Regional Internet Registry, RIR). Toistaiseksi RIR-organisaatioilla riittää vielä osoitteita, mutta nekin tulevat pian loppumaan.

Työssä käsitellään MPLS-ympäristön IPv6-migraatiota, jossa tarkemmin keskitytään IPv6 MPLS VPN -yhteyksien toteuttamiseen olemassa olevassa IPv4-verkossa. Tämän lisäksi tutkitaan muitakin tapoja toteuttaa IPv6 VPN -yhteyksiä. Tarve tällaiselle tutkimukselle syntyy palveluntarjoajien tarpeesta säilyttää nykyinen käyttövarmuus ja käytettävyys IPv4-asiakkaille, kun samanaikaisesti on kasvava tarve tarjota IPv6-asiakkaille samoja palveluja kuin vanhoille IPv4-asiakkaillekin. Tämän työn aiheen vuoksi on oleellista ymmärtää MPLS-teknologian perusajatukset, jonka vuoksi MPLS-teoriaa käsitellään hieman tarkemmin.

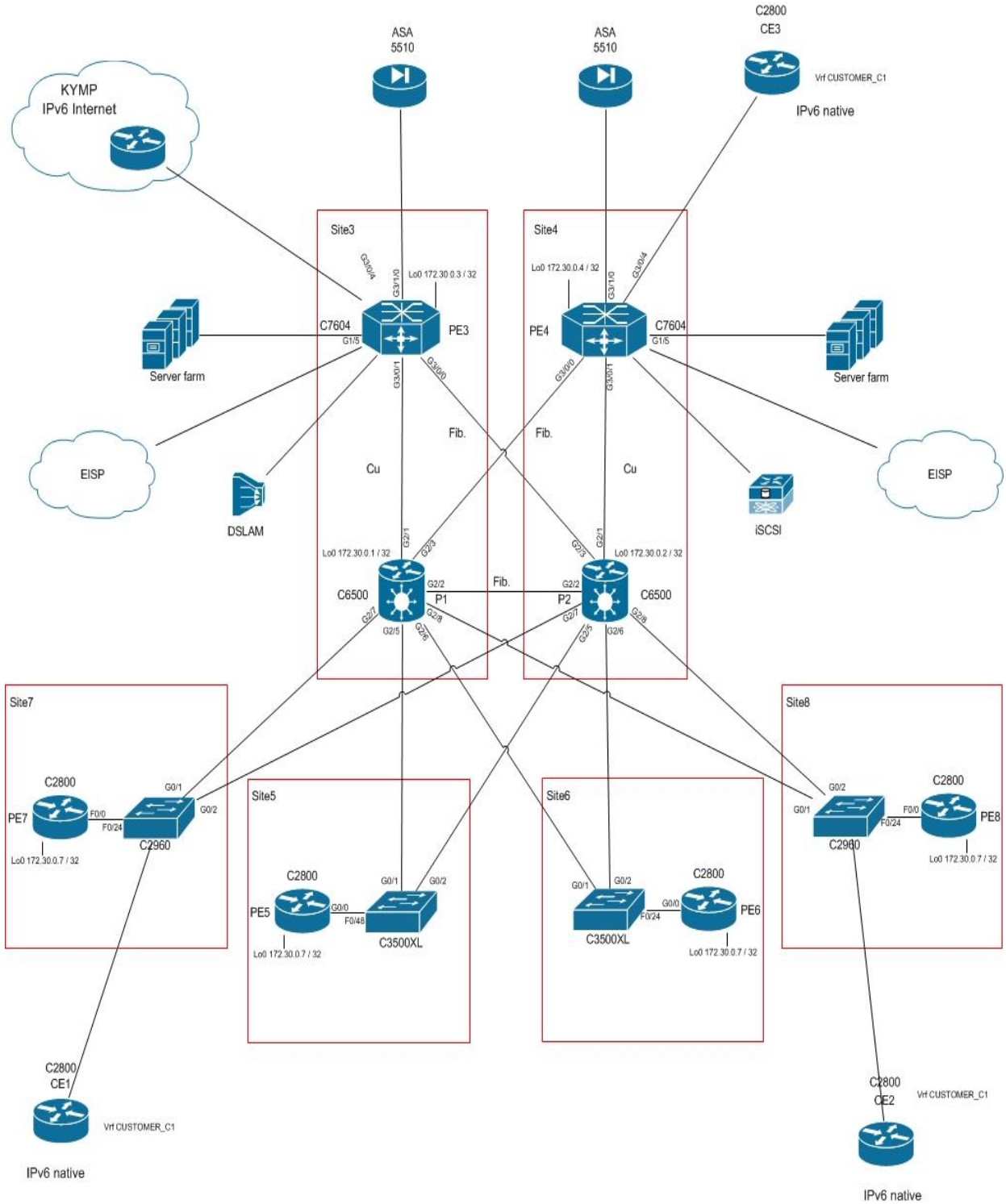
Työssä käytettiin Cisco Systemsin laitteita, joten konfigurointiesimerkit ja -huomiot keskittyvät Ciscon järjestelmiin. Työn ulkopuolelle jätettiin MPLS-teknologian traffic

engineering -ominaisuudet sekä verkon optimointi. Riku Oinosen opinnäytetyö käsittelee VPLS-teknologiaa, joten sen tarkempi käsittely ei kuulu tähän työhön. Lisäksi puhtaat toisen kerroksen teknologiat ovat tämän työn ulkopuolella, kuten Provider Backbone Bridging –teknologia.

2 SIMUNET-HANKE

SimuNet-hankkeen tarkoituksena on toteuttaa laboratorioympäristössä verkko-operaattorin tuotantoverkkoa vastaava tutkimus- ja kehitysympäristö. SimuNet on toteutettu yhteistyössä Kymenlaakson ammattikorkeakoulun ja alueen yritysten kanssa. Hankkeessa on mukana KYMP Oy, Optimiratkaisut Oy, Loviisan Puhelin Oy, Haminan Energia Oy ja Cursor Oy. SimuNet on EAKR:n (Euroopan aluekehitysrahasto) rahoittama hanke. (Kettunen 2009.)

SimuNetin tavoitteena on tarjota ajantasalla oleva T&K-ympäristö sekä ammattikorkeakoulun opiskelijoille että alueen yrityksille. Fyysisesti SimuNet sijaitsee Ky-AMK:n tiloissa, mutta yrityksiä varten on etäkäyttömahdollisuus. Alla (kuva 1) SimuNet-verkon topologia.



Kuva 1. SimuNetin topologia.

3 IPV6

IPv6 (Internet Protocol version 6) on nykyisen IP-protokolla version 4 seuraaja. Suurimpana erona versiossa kuusi on mahdollisten osoitteiden lukumäärä, kun IPv4-osoitteiden määrä on noin 4,3 miljardia, on vastaava luku IPv6-protokollalla 3,4 x 10^{38}. IPv6-osoitteet ovat mallia x:x:x:x:x:x:x:x, jossa jokainen x vastaa 16-bittistä heksadesimaaliarvoa (0000 – FFFF).

Syynä uudempaan versioon siirtymiseen on nykyisten IPv4-osoitteiden loppuminen maailmanlaajuisesti. Arvioiden mukaan IPv4-osoitteet tulevat loppumaan kesken vuoden 2011 loppuun mennessä (RIPE 2011).

Ongelmana siirtymävaiheessa on operaattoreilla palveluiden saatavuuden takaaminen niin vanhoille IPv4-asiakkaille kuin uusille IPv6-asiakkaillekin. Nykyinen saatavuus- ja luotettavuustaso palveluntarjoajien IPv4-verkoissa on huippuluokkaa, ja sitä ei palveluntarjoajilla ole varaa menettää. Siirtymävaiheessa palveluntarjoajien onkin käytännössä pakko etsiä ratkaisuja, joissa tarjotaan IPv6-palveluita olemassa olevassa IPv4-verkossa.

4 MPLS-TEKNIIKKA

MPLS (Multiprotocol Label Switching) on teknologia, joka käyttää pakettien ohjaamiseen verkossa lippuja (label), jotka liitetään esimerkiksi IP-paketteihin. Vastaavanlaista tekniikkaa on käytetty aiemmin sekä frame relay että ATM -verkoissa. MPLS-teknologian sijoittaminen OSI-malliin tietylle kerrokselle on haastavaa, koska osa sen ominaisuuksista sijoittuu 2. kerrokselle ja osa 3. kerrokselle, minkä vuoksi sanotaan, että MPLS on 2,5 kerroksen teknologia (De Ghein 2007, 5-28).

4.1 MPLS-tekniikan hyödyt

Seuraavissa kappaleissa tutustutaan etuihin, joita MPLS-teknologia tarjoaa. Asiat käsitellään yleisesti ja joitakin pääkohtia pohditaan tarkemmin.

4.1.1 Yksi yhteinen verkko

MPLS-tekniikan ehdottomasti suurin etu on sen kyky siirtää lähes mitä tahansa dataa yhden ja saman runkoverkon läpi. MPLS-verkkoon tulevat kehykset (frame) liputetaan niiden kohteen tai jonkin muun periaatteen mukaan ja jokainen reitti on ennalta määritetty, jolloin pakettien ohjaus on nopeaa ja tehokasta. MPLS-pilven läpi voi IPv4-pakettien lisäksi kuljettaa IPv6-paketteja sekä Ethernet, PPP ja muita kerroksen 2 protokollia. Ominaisuutta, joka mahdollistaa 2. kerroksen teknologioiden kuljettamisen MPLS-pilven läpi kutsutaan AToM (Any Transport over MPLS) -tekniikaksi, jossa MPLS-pilveen kuuluvat reitittimet eivät tunne kuorman sisältöä tai protokollaa. (De Ghein 2007, 7.)

AToM-tekniikka mahdollistaa siis tason 2 palvelujen tarjoamisen yhden verkon yli, samalla tavoin kuin erillinen tason 2 protokollaan perustuva verkko, tarjoten samalla kaikki muut MPLS-teknologian edut.

4.1.2 ATM-integraatio

Tärkeä MPLS-teknologian tarjoama ominaisuus, joka johti sen käyttöönottoon laajassa mittakaavassa, on MPLS ja ATM –teknologioiden integraatio. IP-verkkojen tullessa valtaan aiheutti ATM-IP -migraatio suuria ongelmia. Ratkaisut, joita ongelmaan kehitettiin, olivat vaikeita toteuttaa ja hankalia käyttää ja ylläpitää. Teknologiat jotka MPLS:n lisäksi mahdollistavat IP-liikenteen ATM-verkossa ovat RFC 1483:n (myöhemmin RFC 2684) määrittelemä enkapsulointi AAL5:n (ATM adaption layer 5) kautta, LAN Emulointi (LANE) joka sai ATM-verkon näyttämään yhdeltä suurelta lähiverkolta, sekä Multiprotocol over ATM (MPOA) –teknologia, joka tarjosi erittäin tiukan integraation IP ja ATM -teknologioiden välillä. Edellä mainitut ratkaisut kaikki vaativat, että verkko on overlay-tyyppinen, eli jokaisen reitittimen välillä täytyy olla oma VC-piiri, joiden määrä nousee erittäin nopeasti verrattuna reitittimien määrään. (De Ghein 2007, 7-8.)

MPLS-verkko toimii vertaismallilla, jossa jokainen reititin muodostaa naapuruussuhteen vain lähimpiin reitittimiin. MPLS ATM -reitittimet käsittävät reititysprotokollan ja LDP (label distribution protocol) -protokollan kontrollitasolla (control plane), sekä

ATM-solujen siirtomahdollisuuden datatasolla (data plane). MPLS-teknologian avulla ATM-verkko ja muu MPLS-pilvi voidaan yhdistää, jolloin ATM-verkon reunalla oleva ATM LSR (label switch router) -reititin pilkkoo kehykset ATM:lle sopiviksi soluiksi ja päinvastoin. (De Ghein 2007, 7-8.)

4.1.3 Runkoverkon yksinkertaisuus

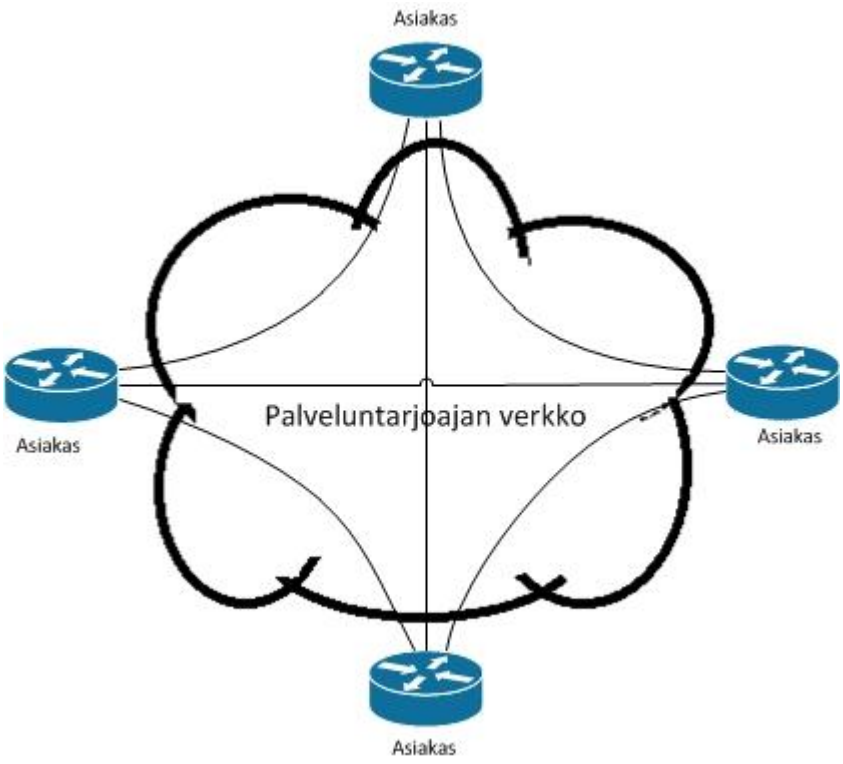
Tavallisessa IP-verkossa jokaisen reitittimen pitää olla tietoinen saapuvan paketin kohdeosoitteesta ja sen sijainnista, jotta reititin voi tehdä reitityspäätöksiä. Palveluntarjoajan runkoverkossa reitittimien tulee tuntea myös kaikki oman runkoverkon ulkopuoliset prefixit, jolloin reititystaulut kasvavat erittäin suuriksi. Ainoa reititysprotokolla joka pystyy vastamaan tällaisiin tarpeisiin on BGP (Border Gateway Protocol). IP-runkoverkossa jokaisessa reitittimessä täytyy olla BGP käytössä, ja jokaisen BGP-reititystaulun täytyy olla täydellinen. Tämä aiheuttaa huomattavasti kuormitusta palveluntarjoajan runkoverkolle. (De Ghein 2007, 8-10.)

MPLS-verkossa paketit kytketään kohdeosoitteen sijaan MPLS-lippujen perusteella, jolloin palveluntarjoajan runkoverkon sisäiset reitittimet eivät tarvitse täysiä IP-reititystauluja, vaan pakettien ohjaus tapahtuu pelkkien lippujen avulla. Ainoastaan MPLS-pilven reunalla olevat reitittimet tarvitsevat IP-reititystaulut, jotta ne osaavat reitittää liikenteen sisään ja ulos MPLS-pilvestä. Tämä tarkoittaa MPLS-verkossa sitä, että runkoverkon sisällä ei tarvitse käyttää BGP-protokollaa, joka vähentää kuormitusta ja nopeuttaa verkon toimintaa. (De Ghein 2007, 8-10.)

4.1.4 Peer-to-peer VPN -malli

VPN (Virtual Private Network) -teknologia emuloi yksityistä verkkoa yleisen verkkoinfrastruktuurin läpi. VPN-verkot mahdollistavat erillisten yksityisten verkkojen yhdistämisen siten, että vain samaan VPN-verkkoon kuuluvat pääsevät siihen kiinni. VPN-teknologia on erittäin tärkeä osa nykyajan verkkoinfrastruktuuria, koska yritysten ja yhteisöjen verkot on yhdistettävä siten, että erilliset verkot näyttävät yhdeltä samalta sisäverkolta. VPN-verkko voidaan toteuttaa kahdella tavalla, overlay- tai peer-to-peer -mallilla.

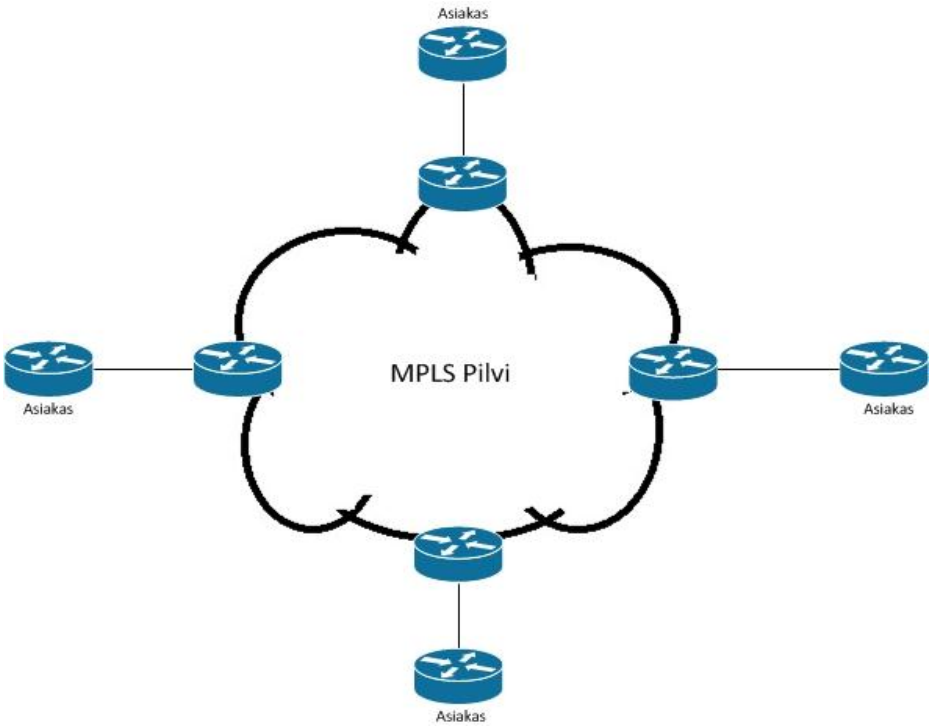
Overlay VPN -mallissa palveluntarjoaja tarjoaa asiakkaalle suoran linkin eri verkkojen välille. Tässä mallissa palveluntarjoaja tarjoaa vain point-to-point -linkin tai -tunnelin asiakkaan verkkojen välille, osallistumatta ollenkaan liikenteen reititykseen. Linkki voidaan toteuttaa OSI-mallin tasolla 1, 2 tai 3. Fyysisen kerroksen ratkaisuista esimerkkinä voidaan käyttää TDM (Time-division multiplexing), SONET tai SDH -linkkejä. Tason 2 ratkaisuna voi olla esimerkiksi ATM:n tai Frame Relayn avulla luotu virtuaalinen linkki. IP-protokollaa käyttäen virtuaalilinkki voidaan luoda käyttäen esimerkiksi GRE (generic routing encapsulation) -tunnelia. Overlay-mallissa jokainen asiakkaan vaatima VPN-yhteys pitää luoda erikseen jokaista sijaintia tai verkkoa varten, joka VPN-verkkoon halutaan liittää. Overlay-mallin ongelmana on linkkien määrän huomattava kasvu asiakkaan verkkojen määrän kasvaessa. (De Ghein 2007, 10-12.)



Kuva 2. Overlay-malli

Peer-to-peer VPN -mallissa palveluntarjoaja osallistuu asiakkaan liikenteen reititykseen, jolloin palveluntarjoajan reunareititin muodostaa naapuruussuhteen asiakkaan reunareitittimen kanssa. Erillisiä tunneleita ei tavallisessa IP-pohjaisessa peer-to-peer

VPN -mallissa luoda, vaan reititystä hallitaan pääsylistoilla (access list) ja reitityspro-tokollan reittifiltteröinnillä. Perinteisen peer-to-peer VPN -mallin ongelmana on, että konfiguraatiomuutoksia täytyy tehdä monessa paikassa samaan aikaan, jos asiakkaan tarpeissa tapahtuu muutoksia tai halutaan lisätä tai poistaa sijainteja VPN-verkosta. Ennen MPLS-teknologiaa peer-to-peer -mallia ei juurikaan käytetty, vaan suosittiin overlay VPN -mallia. (De Ghein 2007, 12-16.)



Kuva 3. Peer-to-peer MPLS -malli

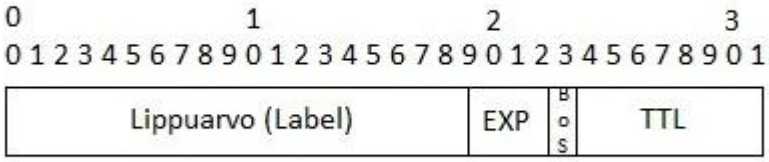
MPLS tarjoaa paremman tavan toteuttaa peer-to-peer VPN -verkkoja. Tason 3 (L3) MPLS VPN -verkot perustuvat VRF (virtual routing/forwarding) -instansseihin. VRF-instassiin kuuluva liikenne ohjataan MPLS-verkossa lippujen perusteella vain samaan instanssiin kuuluville, jolloin VPN-verkon yksityisyys on turvattu. MPLS L3 VPN -verkossa asiakkaan verkon lisääminen VPN-verkkoon vaatii vain uuden verkon lisää-misen VRF-instanssiin, eikä erillisiä virtuaalilinkkejä tai pääsylistoja tarvitse tehdä. (De Ghein 2007, 12-16.)

4.2 Toimintamalli

Tässä osiossa käsitellään MPLS-tekniikan toiminnan perusteita: kuinka pakettien ohjaus lippujen perusteella tapahtuu ja mitä ovat MPLS verkon elementit.

4.2.1 MPLS-liput

Kaikkiin MPLS-verkkoon saapuviin paketteihin lisätään MPLS-verkon reunalla lippu, tai lippuja, joiden perusteella pakettien kytkentä tapahtuu. MPLS-verkon sisällä kytkentä tapahtuu puhtaasti näiden lippujen avulla, riippumatta siitä minkä tyyppinen paketti oikeasti on. MPLS-lippu on 32-bittinen kenttä, joka lisätään kehykseen tason 2 ja 3 otsikkokenttien väliin. MPLS-lipun ensimmäiset 20 bittiä kertovat lipun arvon, seuraavat 3 bittiä on varattu QoS (Quality of Service) -palveluiden käyttöön, 23. bitti on BoS (Bottom of Stack) -bitti, joka kertoo onko lippu alimmaisena lippupinossa (label stack). Viimeiset 8 bittiä ovat TTL (Time To Live) -käytössä. (De Ghein 2007, 25-27.)



Kuva 4. MPLS-lipun rakenne

MPLS-verkossa kytkettävät paketit voivat vaatia useampia lippuja, kuten esimerkiksi MPLS VPN tai AToM -käytössä. Lippupino voi sisältää rajattoman määrän lippuja tarpeesta riippuen. Lippupinossa kaikkien paitsi alimmaisen lipun BoS-bitin arvo on 0 ja alimmaisella lipulla se on 1. (De Ghein 2007, 25-27.)

MPLS-verkon läpi tietystä pisteestä tiettyyn pisteeseen kulkevaa polkua kutsutaan FEC (Forwarding Equivalence Class) -poluksi. MPLS-verkon reitittimet liittävät kaikkiin tuntemiinsa FEC-polkuihin lippuarvon, joka on laitekohtainen ja jolla ei ole

verkonlaajuista merkitystä. Tämän vuoksi tarvitaan protokolla, joka kerää ja jakaa tiedon lipuista. LDP (Label Distribution Protocol) levittää laitekohtaiset lippuarvot naapurireitittimille. Tiedot lippusidoksista tallennetaan LIB (Label Information Base) -tietueeseen, josta LSR-reititin valitsee yhden polun, jonka se tallentaa LFIB (Label Forwarding Instance Base) -tietueeseen. Liputettujen pakettien kytkentä tapahtuu LFIB-tietueen perusteella. (De Ghein 2007, 29-37.)

4.2.2 MPLS-verkon laitteet

MPLS-verkon reitittimiä kutsutaan LSR (label switch router) -reitittimiksi. LSR-reitittimiä on kolmen tyyppisiä: ingress-, egress- ja intermediate LSR -reitittimiä. Ingress LSR on MPLS-verkon reunalla oleva reititin, joka ottaa vastaan MPLS-verkon ulkopuolelta tulevat paketit ja lisää niihin lipun tai lippupinon, ja ohjaa paketin eteenpäin MPLS-verkkoon. Egress LSR -reitittimet ovat reunareitittimiä, jotka ottavat vastaan liputettuja paketteja ja poistavat liput paketeista ja ohjaavat ne ulos MPLS-verkosta. Intermediate LSR -reitittimet ovat MPLS-verkon sisäisiä runkoreitittimiä, jotka kytkevät liputettuja paketteja. LSR-reitittimillä on niiden roolista riippuen toiset vakiintuneet termit: ingress ja egress LSR -reitittimiä kutsutaan PE (Provider Edge) -reitittimiksi, koska ne sijaitsevat palveluntarjoajan MPLS-verkon reunalla, kun taas intermediate LSR -reitittimiä kutsutaan P (Provider) -reitittimiksi, koska ne sijaitsevat palveluntarjoajan runkoverkossa. Tässä työssä jatkossa käytetään termejä P ja PE -reititin viitattaessa tietynlaisiin LSR-reitittimiin. (De Ghein 2007, 29.)

LSR-reitittimet voivat suorittaa paketeille kolme operaatiota: push, pop ja swap. Push-operaatiolla lisätään pakettiin lippu tai lippuja, pop-operaatiolla poistetaan lippuja ja swap-operaatiolla vaihdetaan lippupinon ylimmäinen lippu uuteen. (De Ghein 2007, 29.)

5 MPLS JA IPV6

Palveluntarjoajille, jotka käyttävät MPLS-runkoverkkoa, on tarjolla neljä erilaista lähestymistapaa IPv6-protokollan käyttöönottoon. Tässä työssä käsitellään lyhyesti kolme näistä tavoista ja tarkemmin neljäs ja tällä hetkellä paras tapa.

5.1 Natiivi IPv6 MPLS

Suunniteltaessa IPv6-protokollan käyttöönottoa MPLS-verkossa ensimmäisenä ajatuksena on usein koko MPLS-verkon päivittäminen moniprotokollaympäristöksi, jossa ajetaan rinnakkain IPv4 ja IPv6 -protokollia. Tällainen ratkaisu saattaa kuulostaa kaikkein yksinkertaisimmalta ja tehokkaimmalta ratkaisulta. Ratkaisussa on kaksi tärkeää ongelmakohtaa. Ensinnäkin, jotta voitaisiin toteuttaa täysi IPv6 MPLS -verkko, on kaikki verkon laitteet päivitettävä dual-stack -toimintaan. IPv4-toiminnallisuus on myös taattava vielä pitkään tulevaisuudessa, koska hyvin suuri osa asiakkaista käyttää edelleen IPv4-protokollaa. Toinen ongelma on LDP-protokollan puuttuva IPv6 tuki laitevalmistajien puolelta, vaikka LDPv6-teknologia onkin jo määritelty (Hogg & Vyncke 2009, 110). Näiden syiden takia ei kannata toistaiseksi edes harkita puhtaan IPv6 MPLS -verkon suunnittelua. (De Ghein 2007, 352-353.)

5.2 IPv6 ja AToM

AToM-ratkaisussa IPv6-liikenne kuljetetaan MPLS-verkon läpi pseudowire tai virtual circuit -yhteyksien yli 2. tason liikenteenä. Tason 2 liikenne voi olla mitä tahansa AToM-tekniikan tukemaa tyyppiä, esimerkiksi Ethernet, ATM tai Frame Relay -liikennettä. Etuna verrattuna natiivi IPv6 -ratkaisuun on, että runkoverkossa ei tarvita IPv6-protokollaa. Tässäkin ratkaisussa on kuitenkin ongelmansa. AToM-kehykset ovat 2. tason, eivätkä IPv6-kehyksiä, joten kehyksiin lisätään ylimääräinen 2. tason otsikkokenttä, joka lisää turhaa kuormaa verkkoon. Toinen ongelma on toteutuksen kömpelyys, koska virtual circuit ja pseudowire -yhteydet ovat yleensä point-to-point -tyyppisiä. Tähän ongelmaan löytyy kuitenkin ratkaisu VPLS-teknologiasta, jota tutkitaan tarkemmin kappaleessa 6.2.3: VPLS. (De Ghein 2007, 352-353.)

5.3 IPv6 ja MPLS VPNv4

IPv6-yhteyksiä on mahdollista toteuttaa IPv4 MPLS VPN -yhteyksien läpi. Tämän ratkaisun vaatimuksena on, että CE-reitittimet ovat dual-stack -tilassa, ja IPv6-liikenne tunneloidaan CE-CE välillä VPN-yhteyden yli. Hyvänä puolena tässä on se, että palveluntarjoajan MPLS-verkossa ei tarvitse tehdä muutoksia, koska MPLS VPN -yhteydet ovat jo valmiiksi olemassa. Ainoastaan CE-reitittimet pitää päivittää, ja tun-

nelit luoda CE-reitittimien välille. Tunnelointiin voidaan käyttää IPv6 over IPv4 GRE-, manuaalisia-, 6to4-, IPv4 yhteensopivia IPv6- tai ISATAP -tunneleita. Ongelmaksi muodostuu ylimääräinen IP-otsikko paketeissa, joka aiheuttaa verkkoon turhaa kuormaa. Toinen, ja vakavampi ongelma on tunneleiden point-to-point -luonne, joka aiheuttaa huomattavasti lisätyötä. (De Ghein 2007, 353.)

5.4 6PE-malli

6PE-malli on tällä hetkellä ainoa ratkaisu, joka tarjoaa MPLS-verkossa toimivan peer-to-peer -tyyppisen toteutuksen. 6PE-mallissa palveluntarjoajan reitittimistä vain PE-reitittimiä käytetään dual-stack -tilassa, eikä P-reitittimiin tarvitse tehdä muutoksia. 6PE-reitittimiin liittyvät asiakkaat voivat käyttää joko IPv4- tai IPv6 -protokollaa. Vaatimukset 6PE-ratkaisuun ovat seuraavat:

- PE-laitteet dual-stack -tilassa (IPv4 ja IPv6)
- CE-laitteet joko IPv4 tai IPv6 tai molempia
- P-laitteet IPv4
- PE-laitteiden välillä täydellinen MP-iBGP -verkosto
- Reititysprotokolla tai staattinen reititys PE-CE välillä

MP-iBGP -protokolla hoitaa IPv6-prefixien ja niihin liittyvien lippujen (IPv6+label) levityksen PE-laitteiden välillä. IPv6-paketteihin tulee kaksi lippua, alimmaiseksi BGP-protokollan jakama lippu, ja ylimmäiseksi IGP-protokollan lippu. Ylimmäistä lippua käytetään paketin kuljettamiseen MPLS-rungon läpi, ja alimmaista lippua käytetään egress PE -reitittimellä IPv6-prefixin selvittämiseen. (De Ghein 2007, 354-358.)

6PE-ratkaisun hyödy ovat seuraavat:

- MPLS-verkossa vain PE-laitteet dual-stack tai IPv6 -käytössä
- Paketit liputettuja IPv6 paketteja → ei ylimääräistä otsikkoa
- Peer-to-peer -malli

- PE-CE reititysprotokollaksi kelpaa mikä tahansa IPv6-yhteensopiva protokolla
- Yksinkertainen toteuttaa

Tällä hetkellä 6PE on järkevin tapa tarjota IPv6-yhteyksiä MPLS-verkossa.

5.4.1 6PE:n konfigurointi

Tässä käsitellään lyhyesti 6PE-ratkaisun konfigurointi Ciscon laitteilla. Konfiguraatioesimerkit eivät liity itse työn toteutukseen, vaan ovat erillisiä esimerkkejä. 6PE lisää vain yhden täysin oman komennon konfiguraatioon:

neighbor *ip-osoite* ***send-label***

Komento ottaa käyttöön MPLS-lippujen levityksen MP-iBGP -protokollan kautta.

Vanhemmissa IOS-versioissa saatetaan lisäksi joutua käyttämään komentoa **mpls ipv6 source-interface** *tyyppi numero*, joka kertoo mitä lähdeosoitetta liikenteelle käytetään.

Seuraavassa lyhyt konfiguraatioesimerkki PE ja CE -laitteista. Esimerkistä on poistettu kaikki konfiguraatiot, jotka eivät ole oleellisia. Esimerkissä käytetään naapuri PE-laitteen osoitteena osoitetta 10.100.1.1. Tarkoituksella on jätetty pois runkoverkon suuntaan menevät yhteydet. Tarkoituksena on esittää mahdollisimman yksinkertaisesti komennot, joita tarvitaan 6PE-ratkaisun käyttöönotossa.

CE-laite:

```
!
ipv6 unicast-routing
ipv6 cef
!
interface loopback0
           ip address 10.10.1.1 255.255.255.255
           ipv6 address 2000:1::1/128
           ipv6 rip RIP enable
!
interface serial 0/0
           description CE-PE linkki
           no ip address
```

```
            ipv6 address 2000:1:1::1/64
            ipv6 enable
            ipv6 rip RIP enable
!
ipv6 router rip RIP
!
```

PE-laite:

```
!
ipv6 unicast-routing
ipv6 cef
!
interface serial 0/0
            description PE-CE linkki
            no ip address
            ipv6 address 2000:1:1::2/64
            ipv6 enable
            ipv6 rip RIP enable
router bgp 1
            neighbor 10.100.1.1 remote-as 1
            neighbor 10.100.1.1 update-source loopback0
            no auto-summary
            !
            address-family ipv6
                        neighbor 10.100.1.1 activate
                        neighbor 10.100.1.1 send-community both
                        neighbor 10.100.1.1 send-label
                        redistribute connected
                        redistribute rip RIP
                        exit-address-family
!
ipv6 router rip RIP
            redistribute bgp 1
!
```

6PE-toteutukseen liittyen on SimuNet-hankkeelle tehty projektityö syksyllä 2010 (Suurnäkki 2010).

6 IPV6 VPN -TOTEUTUKSET

VPN-yhteyksien kehittämiseen ja toteuttamiseen oli alussa kaksi pääsyytä, tietoturvalähtökohta sekä IPv4-osoitteiden säästäminen. VPN-yhteyksillä voitiin yhdistää useampi eri paikassa sijaitseva verkko yhdeksi verkoksi, jolloin voitiin käyttää privaattiosoitteita, vaikka tietoa siirrettiin palveluntarjoajan yhteyksien yli. Tietoturvanäkökulmasta VPN-yhteydet olivat erinomainen ratkaisu, koska niiden liikenne ei päädy muualle kuin VPN-yhteyden toiseen päähän.

IPv6-maailmassa ei ole enää ongelmaa osoitteiden riittävyydessä, eikä sellaista ongelmaa ole tulevaisuudessakaan näkyvissä. Tietoturvanäkökulma on kuitenkin edelleen validi VPN-ratkaisuja mietittäessä, ja siihen pätevät samat perustelut kuin IPv4 VPN -toteutuksissa. Tässä osiossa käsitellään erilaisia tapoja toteuttaa IPv6 VPN -yhteyksiä, lähinnä palveluntarjoajan näkökulmasta, ja pohditaan hieman niiden hyviä ja huonoja puolia, lähinnä toteutus- ja ylläpitonäkökulmasta. Ensinnä tarkistellaan fundamentaalisia eroja kahden mahdollisen toteutusmallin välillä, jotka ovat CE- ja PE-pohjaiset VPN-ratkaisut.

6.1 CE-pohjaiset VPN-ratkaisut

CE-pohjaiset ratkaisut perustuvat CE-laitteiden välisiin tunneleihin, jolloin vastuu VPN-yhteyksistä on täysin CE-laitteilla, eivätkä PE-laitteet ota osaa reititykseen tai tunnelointiin. VPN-yhteydet aloitetaan ja päätetään CE-laitteilta. Tällainen ratkaisumalli on pakosti overlay-tyyppinen, eli jokaisesta CE-laitteesta, joka kuuluu VPN-verkkoon, täytyy olla yhteys jokaiseen muuhun verkkoon kuuluvaan CE-laitteeseen. Hyötypuolena CE-CE -ratkaisussa ei ole merkitystä kulkeeko tieto useamman eri palveluntarjoajan verkon läpi, kunhan CE-CE IP-yhteys on kunnossa. (Popoviciu, Levy-Abegnoli & Grossetete 2006.)

CE-CE -tunnelit voidaan toteuttaa käytännössä millä tahansa tunnelointimetodilla, OSI-mallin kerroksilla 1, 2 tai 3. Alla on esimerkkejä eri kerrosten tunnelointimahdollisuuksista.

Taulukko 1: CE-CE -tunnelit OSI-mallissa

Kerros 1	TDM
Kerros 2	L2TPv3
Kerros 3	IPsec GRE-tunnelointi 6to4-tunnelointi

Tässä työssä tarkastellaan tarkemmin kerrosten 2 ja 3 VPN-yhteyksien toimintaa. Kerroksen 3 tunneleista GRE- ja 6to4-tunnelointia ei käsitellä sen tarkemmin, koska niistä puuttuvat tietoturvaominaisuudet, jollei niissä käytä IPsec-suojausta, jolloin voi mieluummin käyttää suoraan IPsec-tunnelia. Täten käsiteltäväksi CE-CE -ratkaisuista jää L2TPv3 ja IPsec -ratkaisut.

6.1.1 L2TPv3

L2TPv3 (Layer 2 Tunneling Protocol version 3) -protokolla ei vaadi palveluntarjoajalta muuta kuin toimivan IP-verkon, eli MPLS ei ole vaatimus L2TPv3-yhteyksien toteuttamiseen. L2TPv3-yhteyksien läpi voidaan kuljettaa monia kerroksen 2 ja 3 protokollia, kuten IPv6. Ominaisuus, joka mahdollistaa IPv6-pakettien kuljetuksen IP-verkon välityksellä on *IPv6 protocol demultiplexing*. Jos palveluntarjoaja tarjoaa jo valmiiksi L2TPv3-palveluita, mahdollistaa tämä yksinkertaisen tavan yhdistää IPv6-saarekkeita IP-verkon kautta, ilman että runkoverkkoon tarvitsee tehdä muutoksia. (Lewis 2006, 114-118.)

L2TPv3 on harkinnan arvoinen ratkaisu palveluntarjoajille, jotka eivät jostakin syystä halua käyttää MPLS-teknologiaa runkoverkoissaan. L2TPv3 mahdollistaa siis IPv6-yhteyksien luomisen IP-verkon kautta, mutta myös esimerkiksi MPLS VPN -yhteyksien kuljettamiseen IP-rungon lävitse.

6.1.2 IPsec

IPsec VPN -ratkaisun lähtökohtana on tietoturvallisuus. Nykyään IPsec on laajimmalti käytetty CE-CE VPN -ratkaisu. IPsec nojaa tietoturva ratkaisuissaan kahteen protokollaan: Authentication Header (AH) -protokolla vastaa tiedon eheydestä ja oikeellisudeesta, ja Encapsulating Security Payload (ESP) -otsikot vastaavat tiedon kryptaamisesta ja luottamuksellisuudesta. Luottamuksellisuuden varmistus tapahtuu kryptausalgoritmeillä ja eheyden varmistus hash-algoritmeilla. (Popoviciu, Levy-Abegnoli & Grossetete 2006.)

IPsecin yleinen toimintaperiaate koostuu IPsec-keskittimestä, joka yleensä sijaitsee yrityksen verkossa, ja IPsec-asiakkaista, jotka ottavat yhteyden keskittimeen. Asiakas voi olla IPsec VPN -asiakasohjelma yksittäisellä tietokoneella tai CPE (Customer Premises Equipment) -reitittimellä oleva IPsec VPN -ohjelma, joka mahdollistaa kokonaisen verkon pääsyn VPN-verkkoon. IPsec VPN -yhteydet ovat pääsääntöisesti point-to-point -tyyppisiä, mutta on ratkaisuja, joilla voidaan helpottaa useiden sijantien liittämistä VPN-verkkoon, esimerkiksi BGP ja 6to4-tunnelointi yhdistelmällä riittää, että BGP-naapuruussuhteet ovat kunnossa. Tällaisessa ratkaisussa tietoturva ei tosin ole samaa tasoa kuin puhtailla IPsec-sovelluksilla. (Popoviciu, Levy-Abegnoli & Grossetete 2006.)

IPv6 VPN -yhteyksien toteuttamiseen IPsec:illä liittyy pieni reititysongelma, sillä IPsec ei tue IPv6-reititysprotokollia, joten reititysprotokollaa varten täytyy luoda erillinen IPsec-tunneli, joka tunneloidaan primäärisen IPsec-tunnelin sisällä. (Popoviciu, Levy-Abegnoli & Grossetete 2006)

IPsec IPv6 VPN -yhteyksien toteutus vastaa IPv4-toteutusta, ja jo olemassa olevien IPv4-tunnelien läpi voidaan tunneloida myös IPv6-liikenne, joten ratkaisu on helppo, jos käytössä on ennestään IPsec-ratkaisu.

6.2 PE-pohjaiset VPN-ratkaisut

PE-pohjaisessa L3 VPN -ratkaisussa PE-laite osallistuu asiakkaan reititykseen. PE ja CE -laitteiden välillä käytetään reititysprotokollaa tai staattista reititystä, jolloin riittää että PE-laitteet keskustelevat keskenään, ja jokainen CE-laite keskustelee vain oman

PE-naapurinsa kanssa. CE-laitteiden ei siis tarvitse keskustella toistensa kanssa, mikä helpottaa toteutusta, koska yksi PE-laite voi hoitaa monta asiakasta.

Tarjolla on kaksi erilaista lähestymistapaa toteuttaa PE-pohjaisia MPLS L3 VPN -ratkaisuja. Virtuaalinen reititin -ratkaisussa PE-reititin jaetaan useampaan loogiseen reitittimeen, joista jokainen vastaa yhdestä VPN-yhteydestä. Toinen tapa on käyttää yhtä reititintä hoitamaan kaikki VPN-yhteydet. Tämä on BGP-MPLS VPN -ratkaisuissa käytetty tapa. Ciscon reitittimissä IPv6-tuki on vain BGP-MPLS VPN -ratkaisulle, joten tässä työssä keskitytään siihen. Peer-to-peer MPLS IPv6 VPN -ratkaisusta käytetään termiä 6VPE. (Popoviciu, Levy-Abegnoli & Grossetete 2006)

MPLS VPN -toteutuksessa ei ole erillistä tietoturvaprotokollaa, vaan tietoturvasta huolehditaan sillä, että VPN-yhteyden tietovirta kulkee alkupisteestä loppuun MPLS-kytkettynä, jolloin itse pakettien sisältöön ei puututa. Tämä tarkoittaa sitä, että tietoturva on palveluntarjoajan vastuulla. (Popoviciu, Levy-Abegnoli & Grossetete 2006)

PE-pohjainen VPN-ratkaisu MPLS-verkossa on mahdollista toteuttaa myös 2. kerroksella. VPLS-teknologia mahdollistaa multipoint-to-multipoint -tyyppisten L2 VPN -yhteyksien toteutuksen.

Palveluntarjoajat, jotka eivät käytä MPLS-teknologiaa, voivat toteuttaa PE-pohjaisia L3 VPN -ratkaisuja dynaamisten mGRE (Multipoint GRE) -tunneleiden avulla.

6.2.1 6VPE perusteet

6VPE on tähän työhön liittyvistä teknologioista tärkein, joten sitä käsitellään vaaditulla tarkkuudella. 6VPE vastaa toteutukseltaan melko paljon perinteistä IPv4 MPLS VPN -ratkaisua. Jotta voitaisiin ymmärtää 6VPE-tekniikan ajatusta ja toimintaa, käsitellään ensin sen vaatimukset ja pääkohdat.

- MPLS IPv4 -runkoverkko, jossa ovat IPv4 IGP ja LDP -protokollat

6VPE-toteutus perustuu toimivaan MPLS-runkoverkkoon, jossa on käytössä IPv4 IGP -protokolla, esimerkiksi OSPF, IS-IS tai EIGRP. Lippujen jakoon käytetään LDP-protokollaa tai traffic engineeringiä varten RSVP (Resource Reservation Protocol) -protokollaa.

- PE-reitittimien on oltava IPv6 kykyisiä, P-reitittimien ei tarvitse olla IPv6-kykyisiä

MPLS VPN -mallissa PE-reitittimet osallistuvat asiakkaan reititykseen, joten IPv6 VPN -yhteyksiä varten PE-laitteiden täytyy osata IPv6-protokolla. P-reitittimet toimivat edelleen samalla tavoin kuin IPv4 MPLS -verkossa, joten niihin ei tehdä muutoksia, eikä niiden tule osata IPv6-protokollaa.

- PE-laitteella on VRF-instanssi CE-laitteen suuntaan

6VPE, kuten MPLS VPN -toteutuksetkin perustuvat VRF (Virtual routing/forwarding) -instansseihin. VRF-instanssiin kuuluva liikenne ei pääse muualle kuin siihen VPN-verkkoon johon se kuuluu.

- PE-CE välillä on VRF-instansseja tukeva IPv6-reititysprotokolla

Koska PE-laite osallistuu asiakkaan reititykseen, on PE ja CE -laitteiden välillä oltava IPv6 reititysprotokolla. VPN-yhteyksiä varten tämän protokollan on tuettava VRF-instansseja. Työn kirjoituksen hetkellä vain yksi protokolla täyttää nämä vaatimukset: eBGP. Mahdollista on myös toteuttaa PE-CE -reititys staattisilla IPv6-reiteillä.

- Täydellinen MP-iBGP -verkosto (mesh) PE-laitteiden välillä

PE-laitteiden välillä tulee olla täydelliset MP-iBGP -naapuruussuhteet, jotta IPv6 VPN -prefixien ja niihin liittyvien lippujen levitys onnistuu. IPv6 VPN -prefixiä kutsutaan myös vpnv6-prefixiksi, ja sen sekä siihen liittyvän lipun yhdistelmästä käytetään termiä vpnv6+lippu.

- IPv6-paketit kuljetetaan MPLS-verkon läpi kahdella lipulla

(De Ghein 2007, 364-366.)

Ymmärtääksemme miksi IPv6 VPN -paketit tulee merkitä kahdella lipulla, on ensin ymmärrettävä miten PHP (Penultimate Hop Popping) toimii. PHP-tekniikan idea on, että vähentääkseen PE-laitteen kuormitusta, sitä edeltävä P-laite poistaa (pop) paketista lipun ennenkuin ohjaa paketin PE-laitteelle, jotta PE-laitteen tarvitsee tehdä yksi operaatio vähemmän. Tämä on oletuskäytäntö MPLS-verkossa. IPv6 VPN -sovelluksessa kuitenkin paketti on IPv6-paketti, jolloin jos paketissa olisi vain yksi

lippu, ei P-laite osaisi sitä enää reitittää, koska P-laitteet MPLS-verkossa eivät osaa IPv6-protokollaa. Tämän vuoksi IPv6-paketit merkitään kahdella lipulla, joista ylempi on IGP-protokollan lippu, ja alempi BGP-protokollan vpnv6-lippu. (De Ghein 2007, 52, 379.)

6VPE eroaa IPv4 MPLS VPN -toteutuksesta käytännössä melko vähän. BGP-protokollassa käytetään MPLS VPN -toteutuksessa vpnv4-osoiteperhettä (address family), kun 6VPE taas käyttää vpnv6-osoiteperhettä. Viimeisen hypyn (last hop) osoite 6VPE-toteutuksessa on viimeisen PE-laitteen IPv4-kartoitettu IPv6-osoite. 6VPE ja MPLS VPN -toteutuksia voidaan käyttää samanaikaisesti, jopa samassa liityntäportissa, jolloin porttiin konfiguroidaan VRF-instanssit erikseen IPv4 ja IPv6 -liikenteelle. (De Ghein 2007, 365-366.)

6.2.2 6VPE-totetuksen konfigurointi

Tässä osiossa käsitellään teoriassa 6VPE-toteutuksen konfigurointi Ciscon laitteilla. Esimerkit ovat täysin teoreettisia, jotta voitaisiin keskittyä tarkastelemaan vaadittuja komentoja ja niiden merkitystä.

Kuten aiemmin on mainittu, on 6VPE-konfiguraatio melko samankaltainen kuin MPLS VPN -konfiguraatio. Tässä käsitellään siis lähinnä 6VPE-kohtaisia komentoja. Alla esimerkeissä käytetty topologia. Esimerkissä käytetään yhtä VRF-instanssia, *asiakas*, IGP-protokollana OSPF-protokollaa, IP-osoitteet ovat yksityisestä 10 verkosta ja IPv6-osoitteet 2000:: verkosta.

I. MPLS runkoverkon konfigurointi

MPLS-runkoverkko konfiguroidaan normaalisti, kuten IPv4 MPLS -verkko.

II. IPv6 VRF -instanssien konfigurointi

VRF-instanssit konfiguroidaan muuten samoin kuin IPv4:n vastaavat, pienillä muutoksilla itse komentoihin.

IPv4: ip vrf *asiakas*

IPv6: vrf definition *asiakas*

Esimerkki vrf-instanssin luomisesta:

```
vrf definition asiakas
           rd 1:1
           !
           address-family ipv6
                      route-target export 1:1
                      route-target import 1:1
                      exit-address-family
           !
```

Tästä esimerkistä tulee huomioida vrf definition -komennon lisäksi rd -komento, joka määrittelee VRF-instanssin reittierottimen (route distinguisher), jota käytetään vpnv6-osoitteessa, sekä route-target -määrittelyissä kertomaan mihin VPN-verkkoihin reittiä jaetaan ja mitä reittejä tähän VPN-yhteyteen otetaan.

III. Liityntäporttien liittäminen VRF-instanssiin

Kuten edellisessä, tässäkin on pieni muutos komentoihin verrattuna IPv4 MPLS VPN -komentoihin.

IPv4: ip vrf forwarding *asiakas*

IPv6: vrf forwarding *asiakas*

Esimerkki:

```
interface serial 0/0
           description PE-CE linkki
           vrf forwarding asiakas
           ipv6 address 2000::1/64
           ipv6 enable
!
```

Vrf forwarding asiakas kertoo, että liityntäportti kuuluu VRF-instanssiin ”asiakas”, ja kaikki IPv6-liikenne tästä portista liitetään tuohon VPN-yhteyteen.

IV. Osoiteperheiden konfigurointi BGP-protokollaan

BGP-protokollaan lisätään IPv6 VPN-yhteyksiä varten uusi osoiteperhe vpnv6, sen lisäksi konfiguroidaan osoiteperhe ipv6 vrf *asiakas*. Komennot ovat vastaavia kuin MPLS VPN -toteutuksessa.

```
router bgp 1
           no synchronization
           bgp log-neighbor-changes
           neighbor 10.10.10.2 remote-as 1
           neighbor 10.10.10.2 update-source loopback0
           !
           address-family ipv4
                      no neighbor 10.10.10.2 activate
                      exit-address-family
           !
           address-family vpnv6
                      neighbor 10.10.10.2 activate
                      neighbor 10.10.10.2 send-community both
                      exit-address-family
           !
           address-family ipv6 vrf asiakas
                      neighbor 2000::2 remote-as 60001
                      neighbor 2000::2 activate
                      neighbor 2000::2 as-override
                      redistribute connected
                      no synchronization
                      exit-address-family
           !
```

Osoiteperheiden ipv4 ja vpnv6 neighbor-lauseissa käytetään naapuri PE-laitteen loopback-osoitetta, ja ipv6 vrf -osoiteperheen neighbor-lauseissa käytetään CE-laitteen CE-PE -liityntäportin IPv6-osoitetta. Toinen huomioitava komento, vaikka sitä käytetään sekä IPv4 että IPv6 VPN -yhteyksillä, on neighbor 2000::2 as-override. Normaalisti BGP hylkäisi reitin, joka tulee oman as-alueen (autonomous system) ulkopuolelta, mutta jolla on sama ASN (autonomous system number) kuin sillä itsellään. Tätä käytetään suojaamaan BGP-verkkoa silmukoilta. As-override poistaa tämän käyttäytymi-

sen, jolloin VPN-reitit levittyvät oikein, mutta silmukkasuojaus ei ole toiminnassa. VPN-yhteyksille tämä on järkevin tapa ratkaista ASN-ongelmat, eikä silmukkaongelma ole merkittävä VPN-yhteyksillä. (De Ghein 2007, 231.)

V. IPv6-reitityksen konfigurointi PE-CE välille

Tämän työn kirjoituksen hetkellä ainoa reititysprotokolla joka tukee IPv6 VRF -instanseja on MP-eBGP, jonka lisäksi voi käyttää staattista reititystä.

CE-laitteen BGP-konfiguraatio:

```
router bgp 60001
           bgp log-neighbor-changes
           neighbor 2000::1 remote-as 1
           !
           address-family ipv4
                      no neighbor 2000::1 activate
                      exit-address-family
           !
           address-family ipv6
                      neighbor 2000::1 activate
                      network 2000:100::/64
                      exit-address-family
           !
```

PE-laitteen BGP-konfiguraatio (vain PE-CE -kohtainen):

```
router bgp 1
           address-family ipv6 vrf asiakas
                      neighbor 2000::2 remote-as 60001
                      neighbor 2000::2 activate
                      neighbor 2000::2 as-override
                      redistribute connected
                      no synchronization
                      exit-address-family
           !
```

Tämäkin konfiguraatio, kuten nähdään, on hyvin samankaltainen kuin IPv4-verkoissa, vain pienillä eroilla komennoissa.

6.2.3 VPLS

Virtual Private LAN Service (VPLS) -teknologiaa käytetään L2 VPN -palveluiden toteuttamiseen IP/MPLS -verkossa (Oinonen 2011, 14). VPLS-verkkoon liitetyt lähiverkkosegmentit näkevät MPLS-runkoverkon loogisena kytkimenä, jossa samaan VPLS-instanssiin kuuluvat segmentit kuuluvat samaan loogiseen lähiverkkoon (Xu 2010, 464). Asiakkaiden kannalta VPLS-ratkaisu on houkutteleva, koska kaikki VPN-yhteyksiin liittyvä konfigurointi on palveluntarjoajan vastuulla.

Perinteinen VPLS-ratkaisu perustuu pseudowire-linkeillä luotuun full mesh -tyyppiseen topologiaan, jossa jokainen VPLS-instassiin kuuluva PE-laite yhdistetään pseudowire-linkillä kaikkiin muihin instassiin kuuluviin PE-laitteisiin (Minei & Lucek 2005, 317-321). Full mesh -toteutukset ovat työläitä toteuttaa, etenkin jos laitteita on useita, jonka vuoksi on kehitetty uusia VPLS-toteutuksia. Hierarkinen VPLS (H-VPLS) -toteutuksessa VPLS-verkko jaetaan runko- ja liityntätasoon. VPLS-rungossa on edelleen oltava full mesh -topologia, mutta liityntätasolta voidaan tarttua VPLS-verkkoon yhdellä pseudowire-linkillä (Oinonen 2011, 15-16). Viimeisin askel VPLS-palveluiden kehityksessä on A-VPLS (Advanced VPLS), jossa on uusi konfigurointitila *virtual-ethernet* VPLS-instanssien konfigurointiin, sekä VPLS-liikenteen kuormanjakoa parantava flow label -kenttä (Singh 2011).

VPLS-teknologian käyttöön suurien lähiverkkojen emuloimiseen liittyy kuitenkin ongelma. VPLS perustuu L2 ethernet -teknologiaan, jolloin liikenteen ohjauksessa käytetään MAC-osoitteita. VPLS-verkoissa, joissa CE-laitteena käytetään kytkintä ja joissa on useita LAN-segmenttejä yhdistettynä, saattaa MAC-osoitteiden määrä nousta huomattavan suureksi, ja toisin kuin IP-osoiteavaruudet, MAC-osoitteita ei voida yhdistää prefixien avulla, vaan jokainen osoite täytyy löytyä erikseen. Tämä aiheuttaa lisäkuormaa VPLS-instanssiin kuuluville palveluntarjoajan laitteille. (Minei & Lucek 2005, 320-321.)

6.2.4 L3 VPN -yhteydet mGRE-tunneleilla

PE-pohjaisia IPv6 VPN -yhteyksiä voidaan toteuttaa myös IP-verkossa, jossa ei ole käytössä MPLS-teknologiaa. mGRE -tunnelointi mahdollistaa MPLS VPN -toteutusta

vastaavan toteutuksen, jossa tunnelit luodaan dynaamisesti BGP-protokollan levittämien vpnv6-tietojen perusteella. mGRE-tunneleiden avulla voidaan luoda myös multipoint L2TPv3 -tunneleita. (Cisco 2010.)

7 MIGRAATION TOTEUTUS

Työ toteutettiin SimuNet-verkkoon (topologia, ks. kuva 1), jossa on käytössä MPLS. Tarkoituksena oli ottaa käyttöön IPv6 VPN -yhteyksiä ja tarkkailla, vaikuttaako muutosten teko alla olevaan IPv4-toimintaan. Työn aikana oli myös tarkoitus arvioida, kuinka hyvin 6VPE-toteutus käytännössä toimii ja kuinka helppoa se on ottaa käyttöön.

7.1 Vaatimukset

Vaatimuksina 6VPE-teknologian käyttöönotolle on MPLS-teknologian käyttö, sekä PE ja CE -laitteissa IPv6-valmius sekä tuki 6VPE-teknologialle. Muutama laite vaati ohjelmistopäivityksen, jotta vaadittava 6VPE-tuki onnistui. Käytetty Cisco IOS -versio oli 12.4(T), josta löytyy tuki kaikille tarvittaville teknologiolle. MPLS oli jo valmiiksi käytössä SimuNetissä, joten sekin vaatimus toteutui.

7.2 Suunnittelu

Alkuperäinen suunnitelma oli luoda useita 6VPE-asiakkaita ja liittää nämä eri reititysprotokollilla VPN-verkkoon. Kuitenkin työn edetessä huomattiin, että vaihtoehtoiset reititysprotokollat ovat vähissä, ainoastaan MP-eBGP osasi työn tekemisen hetkellä IPv6 VRF -tekniikan käytön. Lopulta päädyttiin siihen, että kaksi VPN-asiakasta käyttää MP-eBGP -protokollaa, ja yksi asiakas liitetään VPN-verkkoon staattisella reitityksellä. Työssä ajateltiin että VPN-asiakkaat ovat saman yrityksen eri toimipisteitä. Asiakkaiden suhteen päädyttiin siihen, että nämä käyttävät puhtaasti IPv6-protokollaa, eli asiakkaat eivät käytä IPv4-protokollaa ollenkaan.

Asiakkaan reitittiminä käytettiin kolmea Ciscon 2800-sarjan reititintä, jotka nimettiin C1_R1, C1_R2 ja C1_R3. Käytännön syistä CE-reitittimet ovat laboratorion tiloissa, eivätkä kuulu SimuNetin runkoverkkoon. Halusin myös asiakkaat kiinni erilaisiin lait-

teisiin, joten asiakkaan reitittimet R1 ja R2 tulivat kiinni PE7 ja PE8 -laitteisiin, ja asiakkaan R3-reititin yhdistettiin PE4-reitittimeen.

Asiakkaiden liittäminen SimuNet-verkkoon tuotti aluksi hieman päänvaivaa, koska SimuNetissä ei aluksi oltu varauduttu tarpeeksi hyvin asiakkaiden liittämiseen verkkoon, ja liityntäportit loppuivat nopeasti kesken. Ratkaisuna tähän liitettiin SimuNetin PE-reitittimiin edustakytkimiä, joiden avulla saatiin lisää portteja käyttöön.

7.2.1 IPv6-osoitteet

Aivan työn loppumetreillä SimuNet-verkon IPv6-osoitekäytäntöön tehtiin täysi uudistus, jonka ehdin juuri ja juuri saada mukaan tähän työhön. Tämän työn asiakkaat saivat itselleen 2A00:1DD0:100:10::/56 lohkon, joka jaettiin eri sijainneille siten, että R1-R3 -reitittimien takana olevat verkot ovat:

2A00:1DD0:100:10C1::/64
2A00:1DD0:100:10C2::/64
2A00:1DD0:100:10C3::/64

Vianhaun ja testauksen helpottamiseksi myös linkkiverkot otettiin globaalista IPv6-avaruudesta, ja PE-CE -linkkien verkot ovat:

2A00:1DD0:100:F110::/64
2A00:1DD0:100:F210::/64
2A00:1DD0:100:F310::/64

Osoitteiden valinta liityntäporteille ja loopback osoitteille tehtiin mahdollisimman yksinkertaisesti. Linkkiverkkojen päätepisteiden osoitteet ovat ::1 ja ::2, loopback-osoitteet ovat ::1, vastaavissa verkoissa.

7.3 Laitteiden konfigurointi

Tässä osiossa käsitellään yksitäisten laitteiden konfiguraatiot. PE-laitteina käytettiin kahta Ciscon 2800-sarjan reititintä (PE7 ja PE8) sekä yhtä 7600-sarjan reitintä (PE4). CE-laitteina käytettiin kolmea 2800-sarjan reititintä (C1_R1, C1_R2 ja C1_R3). Laitteiden konfiguraatioita on tässä osiossa karsittu reilusti, koska varsinkin PE4 sisältää paljon muihin töihin liittyviä konfiguraatioita, joilla ei ole tähän työhön merkitystä.

Tästä osiosta löytyvät siis vain tähän työhön liittyvät konfiguraatiot. Kokonaiset konfiguraatiotiedostot löytyvät liitteistä. Korostetut rivit ovat erityisen huomionarvoisia.

7.3.1 PE-laitteet

1. PE7-reitittimen konfiguraatio

```
hostname PE7
!
vrf definition CUSTOMER_C1
 rd 6:6
 !
 address-family ipv6
 route-target export 6:6
 route-target import 6:6
 exit-address-family
!
ip cef
!
ip multicast-routing
ipv6 unicast-routing
ipv6 cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address 172.30.0.7 255.255.255.255
!
interface FastEthernet0/0.3
 description FE to IPv6 VPN Customer C1 R1
 vrf forwarding CUSTOMER_C1
 mtu 1600
 encapsulation dot1Q 3
 ipv6 address 2A00:1DD0:100:F110::1/64
```

```
 ipv6 enable
!
router bgp 65001
bgp log-neighbor-changes
 neighbor SISAVERKKO peer-group
 neighbor SISAVERKKO remote-as 65001
 neighbor SISAVERKKO update-source Loopback0
 neighbor SISAVERKKO version 4
 neighbor 2A00:1DD0:100:F110::2 remote-as 65006
 neighbor 172.30.0.4 peer-group SISAVERKKO
 neighbor 172.30.0.8 remote-as 65001
 neighbor 172.30.0.8 update-source Loopback0
 !
 address-family ipv4
  no neighbor 2A00:1DD0:100:F110::2 activate
  neighbor 172.30.0.4 activate
  neighbor 172.30.0.8 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv6
  neighbor 172.30.0.4 activate
  neighbor 172.30.0.8 activate
  neighbor 172.30.0.8 send-community both
 exit-address-family
 !
 address-family ipv6 vrf CUSTOMER_C1
  neighbor 2A00:1DD0:100:F110::2 remote-as 65006
  neighbor 2A00:1DD0:100:F110::2 activate
  neighbor 2A00:1DD0:100:F110::2 as-override
  redistribute connected
  no synchronization
 exit-address-family
!
```

```
ip forward-protocol nd
!
mpls ldp router-id Loopback0 force
!
```

2. PE8-reitittimen konfiguraatio

```
hostname PE8
!
vrf definition CUSTOMER_C1
 rd 6:6
 !
 address-family ipv6
 route-target export 6:6
 route-target import 6:6
 exit-address-family
!
ip cef
ip multicast-routing
ipv6 unicast-routing
ipv6 cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address 172.30.0.8 255.255.255.255
!
interface FastEthernet0/0.3
 description FE to IPv6 VPN Customer C1 R2
 vrf forwarding CUSTOMER_C1
 mtu 1600
 encapsulation dot1Q 3
 ipv6 address 2A00:1DD0:100:F210::1/64
 ipv6 enable
!
router bgp 65001
```

```
 bgp log-neighbor-changes
 neighbor SISAVERKKO peer-group
 neighbor SISAVERKKO remote-as 65001
 neighbor SISAVERKKO update-source Loopback0
 neighbor SISAVERKKO version 4
 neighbor 2A00:1DD0:100:F210::2 remote-as 65006
 neighbor 172.30.0.4 peer-group SISAVERKKO
 neighbor 172.30.0.7 remote-as 65001
 neighbor 172.30.0.7 update-source Loopback0
 !
 address-family ipv4
  no neighbor 2A00:1DD0:100:F210::2 activate
  neighbor 172.30.0.4 activate
  neighbor 172.30.0.7 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv6
  neighbor 172.30.0.4 activate
  neighbor 172.30.0.7 activate
  neighbor 172.30.0.7 send-community both
 exit-address-family
 !
 address-family ipv6 vrf CUSTOMER_C1
  neighbor 2A00:1DD0:100:F210::2 remote-as 65006
  neighbor 2A00:1DD0:100:F210::2 activate
  neighbor 2A00:1DD0:100:F210::2 as-override
  redistribute connected
  no synchronization
 exit-address-family
!
mpls ldp router-id Loopback0 force
!
```

3. PE4-reitittimen konfiguraatio

PE4-laitteella huomioitavaa on, että ipv6 cef on oletuksena käytössä, kun ipv6 unicast-routing -komennon antaa. Tämä on erilainen käyttäytyminen kuin 2800-sarjan laitteilla.

```
!
hostname PE4
!
mls ipv6 vrf
!
vrf definition CUSTOMER_C1
 rd 6:6
 !
 address-family ipv6
  route-target export 6:6
  route-target import 6:6
 exit-address-family
!
ipv6 unicast-routing
!
interface Loopback0
 ip address 172.30.0.4 255.255.255.255
!
interface Loopback6
 no ip address
 ipv6 address 2A00:1DD0:100::4/128
!
interface GigabitEthernet3/0/4
 description 6VPE C1 Site3
 vrf forwarding CUSTOMER_C1
 mtu 1600
 no ip address
 speed 100
 no negotiation auto
 ipv6 address 2A00:1DD0:100:F310::1/64
```

```
!
router bgp 65001
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor SISAVERKKO peer-group
 neighbor SISAVERKKO remote-as 65001
 neighbor SISAVERKKO update-source Loopback0
 neighbor SISAVERKKO version 4
 neighbor 172.30.0.7 remote-as 65001
 neighbor 172.30.0.7 update-source Loopback0
 neighbor 172.30.0.8 remote-as 65001
 neighbor 172.30.0.8 update-source Loopback0
 !
 address-family ipv4
  neighbor 172.30.0.7 activate
  neighbor 172.30.0.8 activate
 exit-address-family
 !
 address-family ipv6
  redistribute connected
  neighbor SISAVERKKO send-label
  neighbor 172.30.0.3 activate
 exit-address-family
 !
 address-family vpnv6
  neighbor 172.30.0.7 activate
  neighbor 172.30.0.7 send-community extended
  neighbor 172.30.0.8 activate
  neighbor 172.30.0.8 send-community extended
 exit-address-family
 !
 address-family ipv6 vrf CUSTOMER_C1
  redistribute connected
  redistribute static
 exit-address-family
```

```
 !
ipv6 route vrf CUSTOMER_C1 2A00:1DD0:100:10C3::/64
2A00:1DD0:100:F310::2
!
mpls ldp router-id Loopback0 force
!
```

7.3.2 CE-laitteet

Asiakasverkon reunalaitteet ovat puhtaasti IPv6-pohjaisia. Huomioitavaa konfiguroitaessa puhtaita IPv6-laitteita on, että ip cef pitää konfiguroida ennen ipv6 cef -komentoa tässäkin tapauksessa.

1. **C1_R1-reitittimen konfiguraatio**

```
!
hostname C1_R1
!
ip cef
!
no ip domain lookup
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
 ipv6 address 2A00:1DD0:100:10C1::1/64
!
interface FastEthernet0/0
 description FE to PE7
 no ip address
 duplex auto
 speed auto
 ipv6 address 2A00:1DD0:100:F110::2/64
```

```
 ipv6 enable
!
router bgp 65006
 bgp router-id 6.6.6.1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 2A00:1DD0:100:F110::1 remote-as 65001
 !
 address-family ipv6
  neighbor 2A00:1DD0:100:F110::1 activate
  network 2A00:1DD0:100:10C1::/64
 exit-address-family
```

2. C1_R2-reitittimen konfiguraatio

```
!
hostname C1_R2
!
ip cef
no ip domain lookup
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
 ipv6 address 2A00:1DD0:100:10C2::1/64
!
interface FastEthernet0/0
 description FE to PE8
 no ip address
 duplex auto
 speed auto
 ipv6 address 2A00:1DD0:100:F210::2/64
 ipv6 enable
!
router bgp 65006
```

```
 bgp router-id 6.6.6.2
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 2A00:1DD0:100:F210::1 remote-as 65001
 !
 address-family ipv6
  neighbor 2A00:1DD0:100:F210::1 activate
  network 2A00:1DD0:100:10C2::/64
 exit-address-family
!
```

3. C1_R3-reitittimen konfiguraatio

```
!
hostname C1_R3
!
ip cef
no ip domain lookup
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
 ipv6 address 2A00:1DD0:100:10C3::1/64
!
interface FastEthernet0/0
 description FE to PE4
 no ip address
 duplex auto
 speed auto
 ipv6 address 2A00:1DD0:100:F310::2/64
!
ipv6 route ::/0 2A00:1DD0:100:F310::1
!
```

7.4 Toteutuksen testaus

Testaukseen käytettiin Cisco IOS -käyttöjärjestelmän show-käskyjä sekä ping- ja traceroute -käskyjä ja näiden mpls-versioita selvittämään onko toiminta toivotunlaista. Testitulokset otetaan pääsääntöisesti vain yhdeltä laitteelta per testi, jos tulokset ovat vastaavanlaiset.

7.4.1 VRF-instanssien tarkistus

6VPE VRF -instansseihin liittyvät show-käskyt, joilla voi tarkistaa, onko VRF-instanssit luotu oikein ja ovatko ne oikeissa liityntäporteissa:

show vrf ipv6 *vrf:n nimi*

show vrf ipv6 detail

show vrf ipv6 interfaces

```
PE7#show vrf ipv6 CUSTOMER_C1
  Name                             Default RD          Protocols   Interfaces
  CUSTOMER_C1                      6:6                 ipv6        Fa0/0.3
PE7#show vrf ipv6 detail
VRF CUSTOMER_C1; default RD 6:6; default VPNID <not set>
  Interfaces:
    Fa0/0.3
Address family ipv6 (Table ID = 0x1E000001):
  Export VPN route-target communities
    RT:6:6
  Import VPN route-target communities
    RT:6:6
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix

PE7#show vrf ipv6 interfaces
Interface              VRF                              Protocol   Address
Fa0/0.3                CUSTOMER_C1                      up         2A00:1DD0:100:F110::1
PE7#
```

Kuva 6. VRF-instanssien show-käskyt.

Tuloksista nähdään, että VRF-instanssi on luotu ja se on oikeassa liityntäportissa.

7.4.2 Vpnv6 prefixien tarkistus

Komenolla **sh bgp vpnv6 unicast vrf** *vrf:n_nimi ipv6_osoite* voidaan tarkistaa onko vpnv6 prefix lisätty oikein BGP-protokollaan.

```
PE8#sh bgp vpnv6 unicast vrf CUSTOMER_C1 2a00:1dd0:100:10c1::1/64
BGP routing table entry for [6:6]2A00:1DD0:100:10C1::/64, version 75
Paths: (1 available, best #1, table CUSTOMER_C1)
  Advertised to update-groups:
     2
  65006
    ::FFFF:172.30.0.7 (metric 3) from 172.30.0.7 (172.30.0.7)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:6:6
      mpls labels in/out nolabel/46
PE8#
```

Tuloksista nähdään, että BGP-protokollan reititystaulusta löytyy oikea reitti.

7.4.3 Reitityksen tarkistus

PE-laitteilla komento, jota käytetään reittien tarkistukseen on:
show ipv6 route vrf *vrf_nimi ipv6_osoite*

CE-laitteilla vastaava komento on:
show ipv6 route *ipv6_osoite*

Erot johtuvat VPN-yhteyksien toiminnasta, PE-laitteiden liityntäportit, jotka osoittavat CE-laitteille kuuluvat VRF-instansseihin, jolloin ne eivät kuulu normaalin reitityksen piiriin.

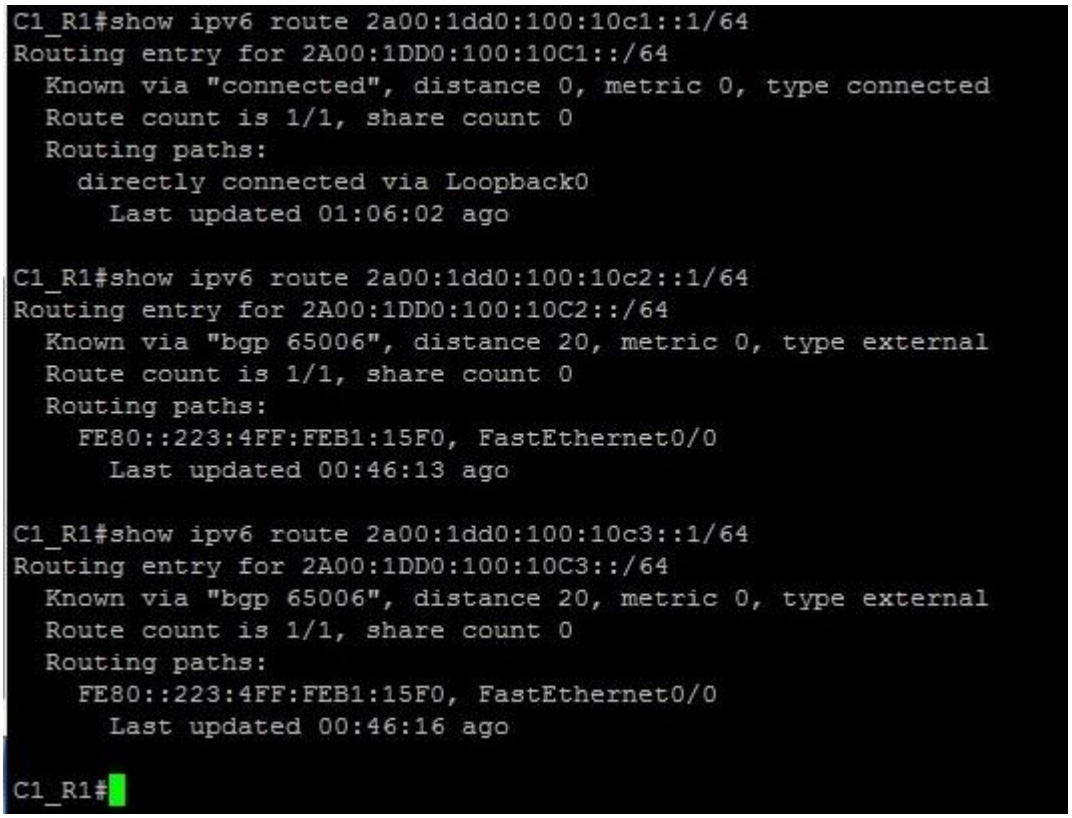
```
PE8#show ipv6 route vrf CUSTOMER_C1 2a00:1dd0:100:10c1::1/64
Routing entry for 2A00:1DD0:100:10C1::/64
  Known via "bgp 65001", distance 200, metric 0, type internal
  Route count is 1/1, share count 0
  Routing paths:
    172.30.0.7%Default-IP-Routing-Table indirectly connected
      MPLS Required
      Last updated 00:45:25 ago

PE8#show ipv6 route vrf CUSTOMER_C1 2a00:1dd0:100:10c2::1/64
Routing entry for 2A00:1DD0:100:10C2::/64
  Known via "bgp 65001", distance 20, metric 0, type external
  Route count is 1/1, share count 0
  Routing paths:
    FE80::21D:A1FF:FE4A:D9AC, FastEthernet0/0.3
      Last updated 00:48:35 ago

PE8#show ipv6 route vrf CUSTOMER_C1 2a00:1dd0:100:10c3::1/64
Routing entry for 2A00:1DD0:100:10C3::/64
  Known via "bgp 65001", distance 200, metric 0, type internal
  Route count is 1/1, share count 0
  Routing paths:
    172.30.0.4%Default-IP-Routing-Table indirectly connected
      MPLS Required
      Last updated 00:56:35 ago

PE8#
```

Kuva 7. Show ipv6 vrf route -käskyt.



Kuva 8. Show ipv6 route -käskyt.

Show ipv6 route -käskyjen tuloksista nähdään, että reititys toimii halutulla tavalla.

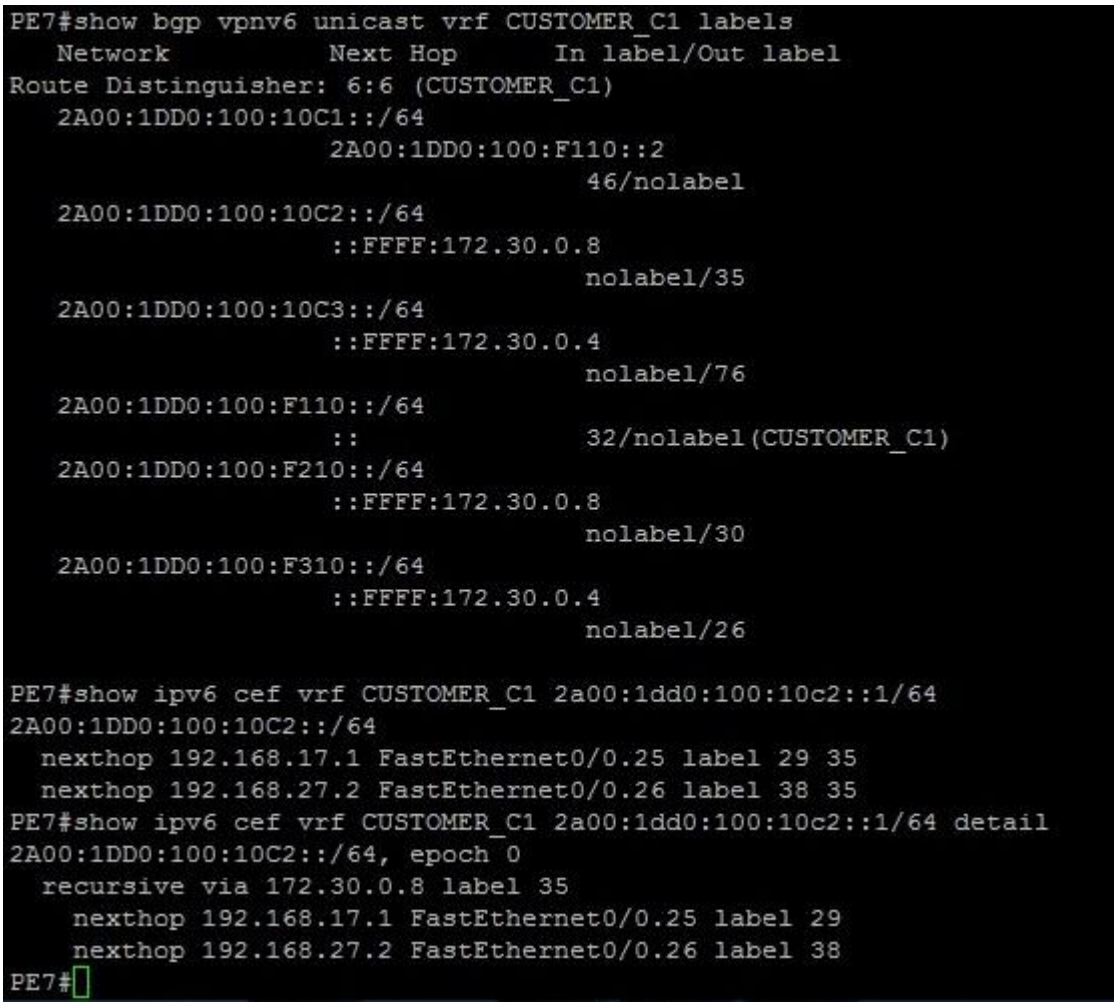
7.4.4 MPLS-lipputietojen tarkistus

Seuraavilla komennoilla voidaan tarkistaa, että MPLS on liputtanut reitit halutulla tavalla, ja että VPN-reiteillä on vaaditut kaksi lippua.

show bgp vpnv6 unicast vrf *vrf_nimi* **labels**

show ipv6 cef vrf *vrf_nimi ipv6_osoite*

show ipv6 cef vrf *vrf_nimi ipv6_osoite* **detail**



Kuva 9. MPLS-lipputietojen tarkistuskäskyt.

Kuvasta nähdään, että lipputiedot ovat oikean näköiset.

7.4.5 LFIB-taulun tarkistus

Komento **show mpls forwarding-table vrf** *vrf_nimi ipv6_osoite* **detail** kertoo LFIB-taulun sisällön halutulle osoitteelle.

PE8#**show mpls forwarding-table vrf CUSTOMER_C1 2a00:1dd0:100:10c1::1/64 detail**

Local Outgoing Prefix Bytes Label Outgoing Next Hop

Label Label or VC or Tunnel Id Switched interface

**None 46 2A00:1DD0:100:10C1::/64[V] **

0

Recursive paths, Label Stack{46}

0002E000

```
     VPN route: CUSTOMER_C1
     No output feature configured
PE8#show mpls forwarding-table vrf CUSTOMER_C1 2a00:1dd0:100:10c2::1/64 detail
Local  Outgoing     Prefix           Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id     Switched      interface
35     No Label     2A00:1DD0:100:10C2::/64[V]   \
                            986          Fa0/0.3    FE80::21D:A1FF:FE4A:D9AC
     MAC/Encaps=18/18, MRU=1604, Label Stack{}
     001DA14AD9AC001DA14ADBEE8100000386DD
     VPN route: CUSTOMER_C1
     No output feature configured
PE8#show mpls forwarding-table vrf CUSTOMER_C1 2a00:1dd0:100:10c3::1/64 detail
Local  Outgoing     Prefix           Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id     Switched      interface
None   76           2A00:1DD0:100:10C3::/64[V]   \
                            0            Fa0/0.26   192.168.28.2
     MAC/Encaps=18/26, MRU=1596, Label Stack{28 76}
     00D0016E780A001DA14ADBEE8100001A8847 0001C0000004C000
     VPN route: CUSTOMER_C1
     No output feature configured
PE8#
```

Tulosteesta saadaan selville paikalliset liput sekä ulosmenevät liput. Nähdään, että R1 ja R3 -reitittimien verkkoihin on merkitty ulosmenevät liput ja R2-reitittimen takana olevaan verkkoon on paikallinen lippu. Tämä on oikea tulos, koska R2 on PE8-reitittimessä kiinni oleva CE-reititin.

7.4.6 Toiminnan testaus ping- ja traceroute-komennoilla

Ping- ja traceroute-komennoilla saadaan selville toimivatko reitit.

```
C1_R1#ping 2a00:1dd0:100:10c3::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2A00:1DD0:100:10C3::1, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/4 ms
C1_R1#
```

Ping-komennon tulos on täydellinen.

```
C1_R1#traceroute 2a00:1dd0:100:10c2::1

Type escape sequence to abort.
Tracing the route to 2A00:1DD0:100:10C2::1

  1 2A00:1DD0:100:F110::1 [AS 65001] 4 msec 0 msec 0 msec
  2  *  *  *
  3 2A00:1DD0:100:F210::1 [AS 65001] [MPLS: Label 35 Exp 0] 0 msec 4 msec 0 msec
  4 2A00:1DD0:100:F210::2 [AS 65001] 4 msec 0 msec 4 msec
C1_R1#
```

Tracerouten tulosteesta CE-reitittimeltä nähdään, mitä reittiä paketit kulkevat, joka on tässä tapauksessa C1_R1 → PE7 → ? → PE8 → C1_R2. Tulosteen * * * -askel on runkoverkon P-reititin, joka ei ole IPv6-kykyinen, tämän takia siitä ei saada tietoa. Tulosteesta nähdään myös MPLS-lippu.

7.5 Vaikutukset IPv4 MPLS -verkkoon

6VPE-teknologian käyttöönotto olemassa olevassa verkossa ei aiheuta juurikaan ongelmia. Ainoa vaikutus, joka saatiin työn aikana selville, oli BGP-protokollan naapuruussuhteen resetoituminen, kun aktivoitiin IPv6 naapuruussuhde laitteeseen, johon oli jo ennestään IPv4 naapuruussuhde. SimuNetissä kesti noin 20 sekuntia, että naapuruussuhde palautui, mutta oikeassa tuotantoverkossa tässä voi kestää useita minuutteja.

8 YHTEENVETO

Työn tavoitteena oli suunnitella ja toteuttaa IPv6 VPN -yhteyksiä SimuNet-verkossa. Toisena tavoitteena oli tutustua yleisesti ratkaisuihin, joilla voidaan IPv6 VPN -yhteyksiä toteuttaa, sekä pohtia mitkä seikat vaikuttavat ratkaisumallin valintaan.

Itse työn toteutuksen suunnittelussa esiin nousi vain yksi varteenotettava vaihtoehto IPv6 VPN -ratkaisun toteutukseen. Koska SimuNet-verkossa oli jo valmiiksi käytössä MPLS-teknologia, oli 6VPE-ratkaisu luonnollinen valinta. 6VPE perustuu jo olemassa olevaan IPv4-pohjaiseen MPLS-verkkoon, joten sen käyttöönotto on yksinkertaista, eikä aiheuta ongelmia olemassa olevaan verkkoon.

SimuNet-verkon migraatio 6VPE-järjestelmään onnistui ilman suurempia ongelmia. Ainoa ongelma, mihin törmäsin toteutuksessa, oli vanhat käyttöjärjestelmäversiot laitteistossa, mutta tämäkin korjaantui päivityksillä ja käyttöön otettu versio 12.4 tarjosi kaikki kaivatut ominaisuudet ja toteutus onnistui hyvin.

8.1 Jatkotutkimus

Työn ulkopuolelle jäi dynaamisten mGRE-tunneleiden ja multipoint L2TPv3 -tunneleiden avulla toteutettujen VPN-ratkaisujen testaus ja toteutus, jotka ovat IP-pohjaisille verkko-operaattoreille harkinnan arvoisia vaihtoehtoja. Tässä olisi mahdollisuus uuteen projektiin tai mahdollisesti opinnäytetyöhön.

8.2 IPv6 VPN pohdintaa

Työssä käytetty 6VPE-teknologia on erittäin hyvä valinta palveluntarjoajille, jotka käyttävät MPLS-teknologiaa verkossaan. 6VPE-ratkaisun konfigurointi on hyvin samankaltaista kuin olemassa olevan MPLS VPN -ratkaisun, joka on erittäin laajasti käytetty ja toimivaksi todettu, joten kynnys 6VPE-teknologian käyttöönottoon on matala. 6VPE-ratkaisun etuna on myös se, että jos käytössä on jo MPLS-verkko, on siihen luultavasti jo konfiguroitu palvelunlaatu (QoS) ja TE (traffic engineering) asetukset, jotka pätevät yhtä lailla 6VPE-yhteyksiin. 6VPE on myös erittäin hyvin skaalautuva, kuten MPLS yleisestikin. Peer-to-peer -malli vähentää VPN-yhteyksien hallintaan ja ylläpitoon liittyvää työmäärää. MPLS-teknologia mahdollistaa myös peer-to-peer -mallilla tarjottavat L2 VPN -palvelut VPLS-teknologian avulla.

Täytyy kuitenkin muistaa, että kaikki palveluntarjoajat eivät käytä MPLS-teknologiaa. Tällaisiin tilanteisiin löytyy vaihtoehtoisia tapoja toteuttaa IPv6 VPN -yhteyksiä: L2TPv3, IPsec, sekä PE-pohjaisista ratkaisuista mGRE-tunnelit. On olemassa myös muita ratkaisuja, mutta niiden käsittely jäi tämän työn ulkopuolelle. Huomattava ero

L2TPv3- ja IPsec-ratkaisujen ja 6VPE-ratkaisun välillä on niiden lähtökohta: ensinmainitut ovat CE-pohjaisia ratkaisuja, jolloin ne ovat lähtökohtaisesti overlay-tyyppisiä, kun taas 6VPE on PE-pohjainen ja perustuu peer-to-peer -malliin, joten 6VPE on ylläpidollisesti kevyempi ratkaisu. Dynaamiset mGRE-tunnelit ovat lähinnä 6VPE-toteutusta, jos MPLS-teknologiaa ei käytetä.

L2TPv3-protokolla on tunnelointiprotokolla, jolla voidaan kuljettaa monenlaista liikennettä IP-verkon yli. L2TPv3-protokollan *IPv6 Prototocol Demultiplexing* -ominaisuuden avulla voidaan sen läpi kuljettaa myös IPv6-liikennettä. L2TPv3-protokollan etuna on sen monipuolisuus, sillä voidaan kuljettaa IPv6-liikenteen lisäksi lähes mitä tahansa 2. kerroksen liikennettä. Jos siis käytössä on IP-runkoverkko, jossa pitää lisäksi kuljettaa IPv6 VPN -liikennettä sekä muita protokollia, on L2TPv3 varteenotettava vaihtoehto toteutusratkaisuksi. Lisäksi olemassa olevaan L2TPv3-arkkitehtuuriin on helppo lisätä IPv6-tunnelointi, joten jos käytössä on jo ennestään L2TPv3, on se käytännöllinen ratkaisu myös IPv6 VPN -yhteyksien tarjoamiseen.

IPsec on laajalti käytetty tapa toteuttaa IPv4 VPN -yhteyksiä, joka tarjoaa vahvan reunalta-reunalle tyyppisen tietosuojan. IPsec tukee myös IPv6 VPN -tunneleita, joten jos käytössä on jo IPsec-arkkitehtuuri, on se luonnollinen valinta myös IPv6 VPN -ratkaisuksi.

Dynaamisten mGRE-tunneleiden avulla voidaan toteuttaa PE-pohjaisia, peer-to-peer -tyyppisiä VPN-yhteyksiä. mGRE-tunneloinnin etuna on sen monipuolisuus, sillä sen läpi voidaan lähettää IPv6-liikenteen lisäksi esimerkiksi L2TPv3 tai IPsec -liikennettä.

Mikä siis on juuri teille sopiva ratkaisu? Tutkimuksen ja työn toteutuksen jälkeen mieleni tekisi vastata kaikille 6VPE. Mutta kysymykseen liittyy vahvasti jo käytössä olevat teknologiat. Aiheuttaa turhaa työtä ja ongelmia siirtyä teknologisesta ratkaisusta toiseen, varsinkin jos käytössä olevasta ratkaisusta on useiden vuosien kokemus. Siispä oikeana ensimmäisenä IPv6 VPN -ratkaisuna voidaan pitää sellaista ratkaisua, joka on rakennettu teknologian päälle joka jo on käytössä. Myös vaadittavien palveluiden määrä ja laajuus täytyy ottaa huomioon. Jos VPN-yhteyksiä tarvitsee toteuttaa vain rajallisesti, ovat CE-pohjaiset ratkaisut erittäin varteenotettavia. Uudelle toimijalle, joka yrittää päättää mitä pohjaratkaisua ryhtyy käyttämään, on vastaus mielestäni selvä:

MPLS-pohjainen ratkaisu, tässä siis 6VPE, mahdollisesti yhdessä VPLS-teknologian kanssa.

LÄHTEET

Cisco 2010. Dynamic layer-3 VPNs with multipoint GRE tunnels. Saatavissa: http://www.cisco.com/en/US/docs/ios/interface/configuration/guide/ir_greL3vpn.html. [viitattu 5.4.2011]

Lewis, Mark 2006. Comparing, Designing, and Deploying VPNs. Cisco Press.

Hogg, S. & Vyncke, E. 2009. IPv6 Security. Cisco Press.

De Ghein, Luc 2007. MPLS Fundamentals. Cisco Press.

Minei, I. & Lucek, J. 2005. MPLS-Enabled Applications. Chichester; John Wiley & Sons.

Oinonen, Riku 2011. MPLS L2VPN ja operaattoriverkon kahdennetut palvelut. Kymenlaakson ammattikorkeakoulu: Opinnäytetyö.

Popoviciu, Ciprian, Levy-Abegnoli, Eric & Grossetete, Patrick 2006. Deploying IPv6 Networks. Cisco Press.

RIPE 2011. RIPE NCC – FAQ: IPv4 Exhaustion. Saatavissa: http://www.ripe.net/internet-coordination/ipv4-exhaustion/faq [viitattu 25.3.2011]

Kettunen, Martti 2009. SimuNet-hanke. Saatavissa: http://www.ictlab.kyamk.fi/index.php?option=com_content&view=article&id=47&Itemid=54 [viitattu 5.4.2011]

Singh, A. 2011. Design and Deployment of Data Center Interconnects using Advanced VPLS (A-VPLS). Cisco Live 2011 -tapahtuman kalvosarja.

Suurnäkki, Simo 2010. 6PE SimuNet. Saatavissa: http://papaya.ictlab.kyamk.fi/~amake/SimuNet/SimuNet_6PE.pdf [viitattu 7.4.2011]

Xu, Z. 2010. Design and Implementing IP/MPLS-Based Ethernet Layer 2 VPN Services. Indianapolis: Wiley Publishing, Inc.

LIITE 1: PE4 konfiguraatio

```
PE4#sh run
Building configuration...
Current configuration : 9148 bytes
!
! Last configuration change at
10:01:53 UTC Tue Mar 29 2011
!
version 15.1
service timestamps debug datetime
msec
service timestamps log datetime msec
service counters max age 10
service unsupported-transceiver
!
hostname PE4
!
boot-start-marker
boot-end-marker
!
mls ipv6 vrf
!
vrf definition CUSTOMER_C1
 rd 6:6
 !
 address-family ipv6
  route-target export 6:6
  route-target import 6:6
 exit-address-family
!
!
no aaa new-model
!
!
!
ip source-route
!
ip vrf INTERNET
 rd 1:80
 route-target export 1:80
 route-target import 1:80
!
no ip domain lookup
ip multicast-routing
!
!
ipv6 unicast-routing
!
!
vtp mode transparent
clns routing
mls flow ip interface-full
no mls flow ipv6
mls cef error action reset
multilink bundle-name authenticated
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
system flowcontrol bus auto
diagnostic bootup level minimal
no errdisable detect cause gbic-
invalid
!
redundancy
 main-cpu
  auto-sync running-config
 mode sso
!
vlan internal allocation policy as-
cending
vlan dot1q tag native
vlan access-log ratelimit 2000
!
vlan 10
 name Area1FWout
!
vlan 20
 name Area2FWout
!
vlan 81
 name INTERNET_C1
!
```

```
vlan 90
 name WLC&APs
!
vlan 91
 name WLC_C1
!
vlan 100
 name Servers1
!
vlan 101
 name Palomuurin_ohitus
!
vlan 200
 name Servers2
!
vlan 300
 name Failover
!
vlan 400
 name iSCSI&ClusterVLAN
!
!
l2 vfi FW_OUT_10 manual
 vpn id 10
 neighbor 172.30.0.3 encapsulation
mpls
!
l2 vfi FW_OUT_20 manual
 vpn id 20
 neighbor 172.30.0.3 encapsulation
mpls
!
l2 vfi INTERNET_C1 manual
 vpn id 81
 neighbor 172.30.0.3 encapsulation
mpls
!
l2 vfi WLC manual
 vpn id 90
 neighbor 172.30.0.7 encapsulation
mpls no-split-horizon
 neighbor 172.30.0.3 encapsulation
mpls
!
l2 vfi WLC_C1 manual
 vpn id 91
 neighbor 172.30.0.3 encapsulation
mpls
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 172.30.0.4
255.255.255.255
!
interface Loopback6
 no ip address
 ipv6 address 2A00:1DD0:100::4/128
!
interface GigabitEthernet1/1
 switchport
 switchport access vlan 400
 switchport mode access
 mtu 9216
!
interface GigabitEthernet1/2
 switchport
 switchport access vlan 400
 switchport mode access
 mtu 9216
!
interface GigabitEthernet1/3
 switchport
 switchport access vlan 400
 switchport mode access
 mtu 9216
!
interface GigabitEthernet1/4
 switchport
 switchport access vlan 400
```

```
 switchport mode access
!
interface GigabitEthernet1/5
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan
100,101,200
 switchport mode trunk
 no keepalive
!
interface GigabitEthernet1/6
 no ip address
 shutdown
!
interface GigabitEthernet1/7
 ip address 172.16.50.1
255.255.255.252
 ip pim sparse-mode
 ip igmp version 3
!
interface GigabitEthernet1/8
 description Firewall failover
 mtu 1600
 no ip address
 xconnect 172.30.0.3 300 encapsula-
tion mpls
!
interface GigabitEthernet1/9
 switchport
 switchport access vlan 1100
 switchport mode access
!
interface GigabitEthernet3/0/0
 description P1-PE4 fiber
 mtu 1600
 ip address 192.168.14.4
255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 negotiation auto
 mpls ip
!
interface GigabitEthernet3/0/1
 description P2-PE4 copper
 mtu 1600
 ip address 192.168.24.4
255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 speed 1000
 no negotiation auto
 mpls ip
!
interface GigabitEthernet3/0/2
 no ip address
 speed 1000
 no negotiation auto
!
interface GigabitEthernet3/0/2.110
 encapsulation dot1Q 110
 ip address 150.100.1.6
255.255.255.252
!
interface GigabitEthernet3/0/2.130
 encapsulation dot1Q 130
 ip address 150.100.1.2
255.255.255.252
!
interface GigabitEthernet3/0/3
 no ip address
 shutdown
 speed 1000
 negotiation auto
!
interface GigabitEthernet3/0/4
 description 6VPE C1 Site3
 vrf forwarding CUSTOMER_C1
 mtu 1600
 no ip address
 speed 100
 no negotiation auto
 ipv6 address
2A00:1DD0:100:F310::1/64
!
interface GigabitEthernet3/1/0
 description Firewall EVC
```

```
 mtu 1600
 no ip address
 speed 1000
 no negotiation auto
 service instance 1 ethernet
  encapsulation dot1q 10
  rewrite ingress tag pop 1 symmetric
  bridge-domain 10
 !
 service instance 2 ethernet
  encapsulation dot1q 20
  rewrite ingress tag pop 1 symmetric
  bridge-domain 20
 !
 service instance 5 ethernet
  encapsulation dot1q 100
  rewrite ingress tag pop 1 symmetric
  bridge-domain 100
 !
 service instance 6 ethernet
  encapsulation dot1q 200
  rewrite ingress tag pop 1 symmetric
  bridge-domain 200
 !
!
interface GigabitEthernet3/1/1
 no ip address
 speed 1000
 negotiation auto
!
interface GigabitEthernet3/1/2
 no ip address
 shutdown
 speed 1000
 negotiation auto
!
interface GigabitEthernet3/1/3
 mtu 1600
 no ip address
 speed 1000
 negotiation auto
 service instance 1 ethernet
  encapsulation dot1q 10
  rewrite ingress tag pop 1 symmetric
  bridge-domain 90
 !
 service instance 2 ethernet
  encapsulation dot1q 50
  rewrite ingress tag pop 1 symmetric
  bridge-domain 91
 !
!
interface GigabitEthernet3/1/4
 no ip address
 shutdown
 speed 1000
 negotiation auto
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan10
 description Firewall context 1 out-
side
 mtu 1600
 ip address 172.30.1.4
255.255.255.248
 no ip redirects
 standby version 2
 standby 10 ip 172.30.1.5
 standby 10 preempt
 standby 110 ipv6 autoconfig
 standby 110 preempt
 ipv6 address
2A00:1DD0:100:A5A1::4/64
 xconnect vfi FW_OUT_10
!
interface Vlan20
 description Firewall context 2 out-
side
 mtu 1600
 ip address 172.31.1.4
255.255.255.248
 no ip redirects
 standby version 2
```

```
 standby 20 ip 172.31.1.5
 standby 20 priority 150
 standby 20 preempt
 standby 120 ipv6 autoconfig
 standby 120 priority 150
 standby 120 preempt
 ipv6 address
2A00:1DD0:100:A5A2::4/64
 xconnect vfi FW_OUT_20
!
interface Vlan81
 ip vrf forwarding INTERNET
 no ip address
 shutdown
 xconnect vfi INTERNET_C1
!
interface Vlan90
 description WLC
 mtu 1600
 no ip address
 xconnect vfi WLC
!
interface Vlan91
 mtu 1600
 ip dhcp relay information trusted
 ip address 172.30.91.4 255.255.255.0
 ip helper-address 172.30.2.90
 standby 91 ip 172.30.91.5
 standby 91 preempt
 xconnect vfi WLC_C1
!
interface Vlan100
 description Firewall context 1 in-
side
 mtu 1600
 no ip address
 xconnect 172.30.0.3 100 encapsula-
tion mpls
!
interface Vlan101
 description Palomuurin_ohitus
 ip address 172.30.101.4
255.255.255.0
 standby version 2
 standby 101 ipv6 autoconfig
 standby 101 preempt
 ipv6 address 2A00:1DD0:100:101::4/64
 xconnect 172.30.0.3 101 encapsula-
tion mpls
!
interface Vlan200
 description Firewall context 2 in-
side
 mtu 1600
 no ip address
 xconnect 172.30.0.3 200 encapsula-
tion mpls
!
interface Vlan400
 description iSCSI&ClusterVLAN
 mtu 1600
 no ip address
 no ip redirects
 xconnect 172.30.0.3 400 encapsula-
tion mpls
!
router ospf 1
 redistribute static subnets
 network 172.16.50.0 0.0.0.3 area 0
 network 172.30.0.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.255.255 area
0
!
router bgp 65001
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor SISAVERKKO peer-group
 neighbor SISAVERKKO remote-as 65001
 neighbor SISAVERKKO update-source
Loopback0
 neighbor SISAVERKKO version 4
 neighbor 150.100.1.1 remote-as 65300
 neighbor 150.100.1.1 version 4
 neighbor 150.100.1.5 remote-as 65100
 neighbor 150.100.1.5 version 4
 neighbor 172.30.0.3 peer-group SI-
SAVERKKO
```

```
 neighbor 172.30.0.5 peer-group SI-
SAVERKKO

 neighbor 172.30.0.6 peer-group SI-
SAVERKKO

 neighbor 172.30.0.7 remote-as 65001

 neighbor 172.30.0.7 update-source
Loopback0

 neighbor 172.30.0.8 remote-as 65001

 neighbor 172.30.0.8 update-source
Loopback0

 !

 address-family ipv4

  network 20.20.0.0 backdoor

  network 172.30.0.0 mask 255.254.0.0

  neighbor 150.100.1.1 activate

  neighbor 150.100.1.1 route-map PRED
in

  neighbor 150.100.1.1 filter-list 1
in

  neighbor 150.100.1.5 activate

  neighbor 172.30.0.3 activate

  neighbor 172.30.0.5 activate

  neighbor 172.30.0.6 activate

  neighbor 172.30.0.7 activate

  neighbor 172.30.0.8 activate

 exit-address-family

 !

 address-family ipv6

  redistribute connected

  neighbor SISAVERKKO send-label

  neighbor 172.30.0.3 activate

 exit-address-family

 !

 address-family vpnv6

  neighbor 172.30.0.7 activate

  neighbor 172.30.0.7 send-community
extended

  neighbor 172.30.0.8 activate

  neighbor 172.30.0.8 send-community
extended

 exit-address-family

 !

 address-family ipv6 vrf CUSTOMER_C1

  redistribute connected

  redistribute static

 exit-address-family

 !

 address-family ipv4 vrf INTERNET

  redistribute static

  redistribute connected

 exit-address-family

!

!

ip as-path access-list 1 deny _400?

ip as-path access-list 1 permit .*

no ip http server

no ip http secure-server

ip route 172.30.0.0 255.255.255.0
Null0

ip route 172.30.2.0 255.255.255.0
172.30.1.1

ip route 172.31.2.0 255.255.255.0
172.31.1.1

!

logging esm config

ipv6 route vrf CUSTOMER_C1
2A00:1DD0:100:10C3::/64
2A00:1DD0:100:F310::2

ipv6 route 2A00:1DD0:100:A1FA::/64
2A00:1DD0:100:A5A1::1

ipv6 route 2A00:1DD0:100:BE7A::/64
2A00:1DD0:100:A5A2::1

ipv6 router ospf 6

!

!

route-map PRED permit 10

 set as-path prepend 1

!

mpls ldp router-id Loopback0 force

!

!

control-plane

!

!

line con 0

 logging synchronous

line vty 0 4

 login

 transport input telnet

line vty 5
```

```
 no login
!
!
!
end
```

LIITE 2: PE7 konfiguraatio

```
PE7#sh run
Building configuration...
Current configuration : 3573 bytes
!
version 12.4
service timestamps debug datetime
msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE7
!
boot-start-marker
boot-end-marker
!
vrf definition CUSTOMER_C1
 rd 6:6
 !
 address-family ipv6
 route-target export 6:6
 route-target import 6:6
 exit-address-family
!
logging message-counter syslog
!
no aaa new-model
!
dot11 syslog
ip source-route
!
!
ip cef
!
!
ip multicast-routing
ipv6 unicast-routing
ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
mpls label protocol ldp
!
!
!
!
voice-card 0
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 172.30.0.7
255.255.255.255
!
interface FastEthernet0/0
 mtu 1600
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.2
 mtu 1600
 encapsulation dot1Q 2
 ip address 172.16.60.1 255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
!
interface FastEthernet0/0.3
```

```
 description FE to IPv6 VPN Customer
C1 R1
 vrf forwarding CUSTOMER_C1
 mtu 1600
 encapsulation dot1Q 3
 ipv6 address
2A00:1DD0:100:F110::1/64
 ipv6 enable
!
interface FastEthernet0/0.13
 description WLAN AP
 mtu 1600
 encapsulation dot1Q 13
 xconnect 172.30.0.3 90 encapsulation
mpls
  backup peer 172.30.0.4 90
!
interface FastEthernet0/0.25
 description P1-PE7
 mtu 1600
 encapsulation dot1Q 25
 ip address 192.168.17.7
255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 mpls ip
!
interface FastEthernet0/0.26
 description P2-PE7
 mtu 1600
 encapsulation dot1Q 26
 ip address 192.168.27.7
255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 mpls ip
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 no fair-queue
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/1/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/1/1
 no ip address
 shutdown
 clock rate 2000000
!
router ospf 1
 log-adjacency-changes
 network 172.16.60.0 0.0.0.255 area 0
 network 172.30.0.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.255.255 area
0
!
router bgp 65001
bgp log-neighbor-changes
 neighbor SISAVERKKO peer-group
 neighbor SISAVERKKO remote-as 65001
 neighbor SISAVERKKO update-source
Loopback0
 neighbor SISAVERKKO version 4
 neighbor 2A00:1DD0:100:F110::2 re-
mote-as 65006
 neighbor 172.30.0.3 peer-group SI-
SAVERKKO
 neighbor 172.30.0.4 peer-group SI-
SAVERKKO
 neighbor 172.30.0.5 peer-group SI-
SAVERKKO
 neighbor 172.30.0.6 peer-group SI-
SAVERKKO
 neighbor 172.30.0.8 remote-as 65001
 neighbor 172.30.0.8 update-source
Loopback0
 !
 address-family ipv4
```

```
  no neighbor 2A00:1DD0:100:F110::2
activate
  neighbor 172.30.0.3 activate
  neighbor 172.30.0.4 activate
  neighbor 172.30.0.5 activate
  neighbor 172.30.0.6 activate
  neighbor 172.30.0.8 activate
  no auto-summary
  no synchronization
  network 172.30.0.0 mask 255.254.0.0
 exit-address-family
 !
 address-family vpnv6
  neighbor 172.30.0.4 activate
  neighbor 172.30.0.8 activate
  neighbor 172.30.0.8 send-community
both
 exit-address-family
 !
 address-family ipv6 vrf CUSTOMER_C1
  neighbor 2A00:1DD0:100:F110::2 re-
mote-as 65006
  neighbor 2A00:1DD0:100:F110::2 ac-
tivate
  neighbor 2A00:1DD0:100:F110::2 as-
override
  redistribute connected
  no synchronization
 exit-address-family
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
ip pim ssm default
!
!
!
!
!
mpls ldp router-id Loopback0 force
!
control-plane
!
!
!
!
!
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

LIITE 3: PE8 konfiguraatio

```
PE8#sh run
Building configuration...
Current configuration : 3246 bytes
!
version 12.4
service timestamps debug datetime
msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE8
!
boot-start-marker
boot-end-marker
!
vrf definition CUSTOMER_C1
 rd 6:6
 !
 address-family ipv6
 route-target export 6:6
 route-target import 6:6
 exit-address-family
!
logging message-counter syslog
!
no aaa new-model
dot11 syslog
ip source-route
!
!
!
!
ip cef
ip multicast-routing
ipv6 unicast-routing
ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
mpls label protocol ldp
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
voice-card 0
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 172.30.0.8
255.255.255.255
!
interface FastEthernet0/0
 mtu 1600
 no ip address
```

```
 duplex auto
 speed auto
!
interface FastEthernet0/0.2
 mtu 1600
 encapsulation dot1Q 2
 ip address 172.16.70.1 255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
!
interface FastEthernet0/0.3
 description FE to IPv6 VPN Customer
C1 R2
 vrf forwarding CUSTOMER_C1
 mtu 1600
 encapsulation dot1Q 3
 ipv6 address
2A00:1DD0:100:F210::1/64
 ipv6 enable
!
interface FastEthernet0/0.25
 description P1-PE8
 mtu 1600
 encapsulation dot1Q 25
 ip address 192.168.18.8
255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 mpls ip
!
interface FastEthernet0/0.26
 description P2-PE8
 mtu 1600
 encapsulation dot1Q 26
 ip address 192.168.28.8
255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 mpls ip
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/1/0
 no ip address
 shutdown
 no fair-queue
 clock rate 2000000
!
interface Serial0/1/1
 no ip address
 shutdown
 clock rate 2000000
!
router ospf 1
 log-adjacency-changes
 network 172.16.70.0 0.0.0.255 area 0
 network 172.30.0.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.255.255 area
0
!
router bgp 65001
 bgp log-neighbor-changes
 neighbor SISAVERKKO peer-group
 neighbor SISAVERKKO remote-as 65001
 neighbor SISAVERKKO update-source
Loopback0
 neighbor SISAVERKKO version 4
 neighbor 2A00:1DD0:100:F210::2 re-
mote-as 65006
 neighbor 172.30.0.3 peer-group SI-
SAVERKKO
 neighbor 172.30.0.4 peer-group SI-
SAVERKKO
 neighbor 172.30.0.5 peer-group SI-
SAVERKKO
 neighbor 172.30.0.6 peer-group SI-
SAVERKKO
 neighbor 172.30.0.7 remote-as 65001
 neighbor 172.30.0.7 update-source
Loopback0
 !
 address-family ipv4
  no neighbor 2A00:1DD0:100:F210::2
activate
  neighbor 172.30.0.3 activate
  neighbor 172.30.0.4 activate
  neighbor 172.30.0.5 activate
  neighbor 172.30.0.6 activate
```

```
  neighbor 172.30.0.7 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv6
  neighbor 172.30.0.4 activate
  neighbor 172.30.0.7 activate
  neighbor 172.30.0.7 send-community
both
 exit-address-family
 !
 address-family ipv6 vrf CUSTOMER_C1
  neighbor 2A00:1DD0:100:F210::2 re-
mote-as 65006
  neighbor 2A00:1DD0:100:F210::2 ac-
tivate
  neighbor 2A00:1DD0:100:F210::2 as-
override
  redistribute connected
  no synchronization
 exit-address-family
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
ip pim ssm default
!
!
!
!
!
!
!
!
!
!
!
!
mpls ldp router-id Loopback0 force
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

LIITE 4: C1_R1 konfiguraatio

```
C1_R1#sh run
Building configuration...


Current configuration : 1769 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname C1_R1
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
memory-size iomem 10
!
dot11 syslog
ip source-route
!
!
ip cef
!
!
no ip domain lookup
ipv6 unicast-routing
ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
voice-card 0
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
!
!
!
!
!
interface Loopback0
 no ip address
 ipv6 address 2A00:1DD0:100:10C1::1/64
!
interface FastEthernet0/0
 description FE to PE7
 no ip address
 duplex auto
 speed auto
```

```
 ipv6 address
2A00:1DD0:100:F110::2/64
 ipv6 enable
!
interface FastEthernet0/1
 no ip address
 ip ospf 1 area 0
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 no fair-queue
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
interface ATM0/3/0
 no ip address
 shutdown
 no atm ilmi-keepalive
!
interface wlan-controller1/0
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
!
router bgp 65006
 bgp router-id 6.6.6.1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 2A00:1DD0:100:F110::1 re-
mote-as 65001
 !
 address-family ipv6
  neighbor 2A00:1DD0:100:F110::1 ac-
tivate
  network 2A00:1DD0:100:10C1::/64
 exit-address-family
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
line con 0
 logging synchronous
line aux 0
line 66
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin
lapb-ta mop udptn v120 ssh
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

LIITE 5: C1_R2 konfiguraatio

```
C1_R2#sh run
Building configuration...


Current configuration : 1410 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname C1_R2
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
ipv6 unicast-routing
ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
voice-card 0
!
!
!
!
!
vtp mode transparent
archive
 log config
  hidekeys
!
!
!
!
!
!
!
!
!
interface Loopback0
 no ip address
 ipv6 address 2A00:1DD0:100:10C2::1/64
!
interface FastEthernet0/0
 description FE to PE8
 no ip address
 duplex auto
```

```
 speed auto
 ipv6 address
2A00:1DD0:100:F210::2/64
 ipv6 enable
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/1/0
 no ip address
 shutdown
 no fair-queue
 clock rate 2000000
!
interface Serial0/1/1
 no ip address
 shutdown
 clock rate 2000000
!
router bgp 65006
 bgp router-id 6.6.6.2
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 2A00:1DD0:100:F210::1 re-
mote-as 65001
 !
 address-family ipv6
  neighbor 2A00:1DD0:100:F210::1 ac-
tivate
  network 2A00:1DD0:100:10C2::/64
 exit-address-family
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
```

LIITE 6: C1_R3 konfiguraatio

```
C1_R3#sh run

Building configuration...



Current configuration : 1399 bytes

!

version 12.4

service timestamps debug datetime
msec

service timestamps log datetime msec

no service password-encryption

!

hostname C1_R3

!

boot-start-marker

boot-end-marker

!

logging message-counter syslog

!

no aaa new-model

dot11 syslog

ip source-route

!

!

!

!

ip cef

no ip domain lookup

ipv6 unicast-routing

ipv6 cef

!

multilink bundle-name authenticated

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

voice-card 0

!

!

!

!

!

vtp domain CISCO

vtp mode transparent

archive

 log config

  hidekeys

!

!

!

!

!

!

!

!

!

interface Loopback0

 no ip address

 ipv6 address
2A00:1DD0:100:10C3::1/64

!

interface FastEthernet0/0

 description FE to PE4

 no ip address
```

```
 duplex auto
 speed auto
 ipv6 address
2A00:1DD0:100:F310::2/64
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/1/0
 no ip address
 shutdown
 clock rate 2000000
!
!
interface Serial0/1/1
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/2/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/2/1
 no ip address
 shutdown
 clock rate 2000000
!
interface ATM0/3/0
 no ip address
 shutdown
 no atm ilmi-keepalive
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
ipv6 route ::/0 2A00:1DD0:100:F310::1
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```