Riku-Matti Ahonen

Mobile IPv6 yhteentoimivuus eri laiteympäristöissä

Tietotekniikan
pro gradu -tutkielma
12. huhtikuuta 2007

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Tekijä: Riku-Matti Ahonen
Yhteystiedot: riahonen@cc.jyu.fi
Työn nimi: Mobile IPv6 yhteentoimivuus eri laiteympäristöissä
Title in English: Mobile IPv6 multivendor interoperability
Työ: Tietotekniikan pro gradu -tutkielma
Sivumäärä: 113
Tiivistelmä: Internetin kasvaneiden vaatimusten myötä ja nykyisen IPv4-protokollan rajoitteiden takia on ilmestynyt ongelmia, joita pyritään ehkäisemään kehittämällä uutta IPv6-protokollaa vastaamaan tulevaisuuden kehityssuuntia. IPv6-protokolla tuo ratkaisun muun muassa IP-osoitteiden riittävyyteen 128 bittisellä osoiteavaruudella, parannetuilla tietoturvaominaisuuksilla ja liikkuvuuden tuella. Tulevaisuuden kannalta yksi IPv6-protokollan tärkeimpiä ominaisuuksia tulee olemaankin juuri liikkuvuuden tuki. IPv6-protokollan liikkuvuuden mahdollistava Mobile IPv6 -laajennus tarjoaa mahdollisuuden liikkuvalle päätelaitteelle siirtyä verkosta toiseen yhteyden katkeamatta. Tässä tutkielmassa keskityttiin tutkimaan erilaisia Mobile IPv6 -toteutuksia ja niiden suorituskykyä. Tutkielman lopussa kerrotaan myös IPv6-protokollan verifiointiin liittyvästä toiminnasta, jotta IPv6-protokollaa kehitettäisiin mahdollisimman standardinmukaisesti ja yhteensopivaksi erilaisten järjestelmien kesken.
English abstract: The Internet of future is going towards ubiquitous access of data and services. These increased demands and limiting recourses of Internet Protocol version 4 are trying to prevent developing the new IP-protocol. This new protocol is called IPv6 and it solves a problem of IP-address needs, provides improved security and mobility. From the point of view of future ubiquitous networks the mobility of IPv6 will be one of the most important features. IPv6 provides mobility support with its extension called Mobile IPv6. This makes possible continuous move between the networks for a mobile node. There are several Mobile IPv6 implementations available these days. Some of most interesting implementations tested in this thesis. The aim of the IPv6 is high quality interoperable IP-protocol so that the verification system is important part of development. IPv6 verification testing are introduced in the end of thesis.
Avainsanat: IPv6, Mobile IPv6, MIPL, KAME, Cisco
Keywords: IPv6, Mobile IPv6, MIPL, KAME, Cisco

Sanasto

AH	Authenticating Header. IPsec-protokolla, joka tarjoaa todennuksen ja takaa viestien eheyden, mutta ei luottamuksellisuutta.
AR	Access Router. Liityntäreititin, jonka välityksellä liikkuva päätelaite liittyy verkkoon.
BAck	Binding Acknowledgement. Kuittaus BU-viestiin.
BU	Binding Update. Kiinnityksen rekisteröintipyyntö, jonka avulla liikkuva päätelaite ilmoittaa vaihtuneen vierasosoitteensa kotiagentille ja liikennöintikumppaneille.
CN	Correspondent Node (suom. liikennöintikumppani). Päätelaite, jonka kanssa liikkuva päätelaite kommunikoi.
CoA	Care-of-Address, vierasosoite. Osoite, jonka liikkuva päätelaite saa siirtyessään vieraaseen verkkoon.
DAD	Duplicate Address Detection. Osoitteen ainutlaatuisuuden tarkistaminen.
ESSID	Extended Service Set Identifier. Tarkoittaa WLAN-verkon verkkotunnusta. (E)SSID:n avulla voidaan erottaa samalla alueella olevat WLAN-verkot toisistaan ja kytkeytyä haluttuun verkkoon.
ESP	Encapsulating Security Payload. IPsec-protokolla jota käytetään pakettivirtojen salaamiseen.
FFHMIPv6	Flow-based Fast Handover for MIPv6. Vuotietojen hyödyntämiseen perustuva MIPv6-protokollan optimointiehdotus.
FHMIPv6	Fast Handovers for MIPv6. Siirtoyhteyskerroksen laukaisimiin perustuva MIPv6-protokollan optimointiehdotus.
HA	Home Agent, kotiagentti. Reititin, joka pitää kirjaa liikkuvien päätelaitteiden sijainnista.
Handover	Yhteysvastuun vaihto. Tilanne, jossa reititin luovuttaa päätelaitteen yhteyden toiselle reitittimelle.

HoA	Home Address, kotiosoite. IP-osoite, jolla liikkuva päätelaite tunnistetaan riippumatta sen fyysisestä sijainnista.
Hop-by-Hop -otsikko	Otsikko, jonka kaikki reitittimet tarkistavat. Sisältää tietoa pakettien käsittelystä reitittimillä.
IETF	Internet Engineering Task Force. Internetiä kehittävä ja standardoiva yhteisö. Julkaisee esimerkiksi RFC-dokumentteja.
IPng	Internet Protocol next generation. Tarkoittaa IPv6-protokollaa.
IPv4	Internet Protocol version 4. Laitteiden väliseen tiedonsiirtoon käytettävä protokolla.
IPv6	Internet Protocol version 6. IPv4-protokollan seuraaja.
L2	Layer 2. OSI-mallin siirtoyhteyskerros.
MIPv6	Mobile IPv6. IPv6-protokollan laajennus, joka mahdollistaa päätelaitteiden liikkumisen verkkojen välillä.
MN	Mobile Node (suom. liikkuva päätelaite). Päätelaite, joka sisältää tuen Mobile IPv6 -protokollalle.
NAT	Network Address Translation. Menetelmä, jolla yhdessä verkossa käytetty IP-osoite muunnetaan toisessa verkossa tunnetuksi IP-osoitteeksi.
NUD	Neighbor Unreachability Detection. Menetelmä, jota käytetään naapurilaitteiden saavuttamattomuuden havaitsemiseen.
RA	Router Advertisement. Viesti, joita reitittimet lähettävät verkkoon. RA-viestien avulla liikkuvat päätelaitteet muodostavat uuden IP-osoitteen vaihtaessaan verkkoa.
RFC	RFC:t ovat sisällöltään ja asemaltaan joukko asiakirjoja, jotka kuvaavat Internetin erilaisia käytäntöjä, lähinnä teknisten menettelyjen järjestelmiä eli protokollia.
RO	Route Optimization. Reitin optimointi liikkuvan päätelaitteen ja liikennöintikumppanin välillä siten, ettei liikennettä välitetä kotiagentin kautta.
RR	Return Routability. Menetelmä, jolla todennetaan BU-viestin lähettäjä.
RS	Router Solicitation. Viesti, jolla voidaan pyytää reititintä lähettämään RA-viesti.

Sisältö

1 Johdanto

Internetin käytöstä on tullut maailman laajuisesti ihmisten jokapäiväinen asia. Sen käytön suosio on kasvanut räjähdysmäisesti viimeisen kymmenen viidentoista vuoden aikana johtuen pääasiassa WWW:n (*World Wide Web*) tulosta markkinoille 90-luvun alussa. WWW:n ansiosta ihmisten tarjolle on tullut valtavasti erilaisia palveluita ja samalla myös tietokoneiden hinnat ovat laskeneet jokaisen kukkarolle sopiviksi. Tämä on edesauttanut Internetin laajenemista maailman joka kolkkiin.

Myös Internetin luonne on alkanut muuttua kiinteistä langallisista verkoista entistä enemmän langattomaan suuntaan ja jatkuvasti kehitetään uusia sovelluskohteita, joissa IP-protokollaa voitaisiin käyttää tiedonsiirtoon. Oman osansa tähän Internetin laajenemiseen ovat tarjonneet olemassa olevat matkapuhelinverkot, joihin puheominaisuuksien lisäksi on alettu kehittämään uudenlaisia sisältöä ja palveluita 2-3G -verkkojen myötä. Nykyiset matkapuhelimet tarjoavat jo nyt mahdollisuuden selailla Internet-sivuja matkapuhelinverkkojen kautta missä tahansa, mutta tulevaisuuden 4G-verkoissa ollaan entistä enemmän sulauttamassa näitä verkkoja muiden langattomien verkkoratkaisujen, kuten langattomien lähiverkkojen (*WLAN*) ja Bluetoothin, kanssa. [1][2][3][4][5][6][7][8][9]

3G:n ja 4G:n myötä on tullut uusia käsitteitä. Yksi tällainen on ABC (engl. *Always Best Connected*) eli vapaasti suomennettuna ”aina parhaiten kytketty”. Tulevaisuuden tietoverkoissa keskeisessä osassa on langattomuus ja myös päätelaitteiden liikkuvuuden mahdollistaminen. Tietojenkäsittelyyn kykenevistä päätelaitteista, esimerkiksi multimediapuhelimet, on tulossa entistä pienempiä ja monipuolisempia toiminnoiltaan. Lisäksi Internetistä tulee käyttäjilleen entistä tärkeämpi osa elämää ja henkilökohtaisia asioita hoidetaan sen kautta. Tulevaisuuden verkoissa päätelaitteiden liikkuvuutta tavoitellaan yhdistelemällä tiedonsiirtoa parhaiten tietyllä hetkellä käytettävissä olevien tiedonsiirtokanavien kautta. Esimerkiksi, kun WLAN:n kantama loppuu, päätelaite osaa siirtää yhteyden automaattisesti 3G-puhelinverkon kautta kulkevaksi. [1]

Myöskään IP-protokolla ei ole lähitulevaisuudessa poistumassa käytöstä, vaan sen käyttöä tullaan tukemaan entistä useammissa sovelluskohteissa. Tulevaisuuden suuntauksena onkin tietoverkkojen kehityksessä kokonaisvaltainen IP-pohjaisuus (engl. *all-IP*), jolla tarkoitetaan juuri sitä, että IP-protokollaa tullaan käyttämään tiedonsiirtoon niin kiinteissä kuin erilaisissa langattomissakin verkoissa. IP-pohjaisuu-

teen siirryttäessä voidaan verkoissa palvelujen tarjoamiseen käyttää olemassa olevaa verkkoinfrastruktuuria, jossa toimivat niin uudet kuin vanhatkin palvelut käytössä olevasta liikennöintirajapinnasta riippumatta. IP-protokolla tulee olemaan merkittävässä asemassa tulevaisuuden tiedonsiirtoverkoissa. [3]

IP-pohjaisuus ja tätä kautta IP-protokollan soveltaminen uusiin kohteisiin tarkoittaa valtavasti uusia päätelaitteita. Käyttäjämäärän kasvu on jo tuonut mukanaan ongelmia, joita Internetin alkulähteillä 1970-luvulla ei osattu käsittää. Suurin ongelma tällä hetkellä on nykyisen IPv4-protokollan tarjoaman 32-bittisen osoiteavaruuden, joka tarjoaa noin 4 miljardia osoitetta, loppuminen. Havaittujen ongelmien johdosta alettiin 1990-luvun alussa suunnitella uutta protokollaa korvaamaan IPv4-protokollaa [20]. Tämä uusi protokolla sai nimekseen IPv6 (Internet Protocol version 6). IPv6-protokollan suurin muutos edeltäjäänsä on 128-bittiseksi laajennettu osoiteavaruus, joka tarjoaa varmasti riittävän määrän osoitteita laajenevalle Internetille. IPv6-protokollan suunnittelussa otettiin huomioon myös sen laajennettavuus, tietoturva ja liikkuvuus. Protokollan kehysrakenne pyrittiin myös suunnittelemaan mahdollisimman yksinkertaiseksi ja mahdollisien laajennuksien toteuttaminen helpoksi.

IPv6-protokollaan on määritelty liikkuvuuden mahdollistava tuki, joka on toteutettu protokollan laajennettavuusominaisuuksia hyödyntäen. Tämän nimeksi tuli luonnollisesti Mobile IPv6 -protokolla ja sen määrittely valmistui IETF:n (*Internet Engineering Task Force*) Mobile IP -työryhmässä vuonna 2004. Mobile IPv6-protokollan avulla liikkuva päätelaite voi liikkua verkosta toiseen samalla osoitteelle yhteyksien katkeamatta. [32]

IPv6-protokollan suunnittelu on nykyaikana myös helpompaa ja tehokkaampaa verrattuna IPv4-protokollan suunnitteluun. IPv4:n käytöstä Internetissä on jo paljon kokemusta ja on alettu hahmottamaan paremmin sen ongelma-alueita. Internet on myös verraton apu yhdistettäessä ympäri maailmaa olevien kehittäjien ja tutkimusprojektien innovaatioita. Ympäri maailmaa on synnytetty useita tutkimusprojekteja ja foorumeita, joissa tutkitaan ja testataan IPv6:n kehitystä. IPv6:n pioneerina toimi 6bone-projekti, joka oli IPv6-pakettien reititystä tukeva virtuaaliverkko. Kehittäjät saivat sieltä IPv6-osoitteita, jonka avulla voitiin testata omia IPv6-toteutuksia 6bone:n palvelimien kautta. [10]

6bone-hanke on kuitenkin sittemmin kuopattu ja työtä jatkaa Hexago-niminen projekti [11]. Hexago tarjoaa muun muassa IPv6-yhdyskäytävän IPv4-verkon päällä. Sivuilta saa myös ohjelman [12] ja IPv6-osoitteen, jonka avulla voidaan luoda IPv4/IPv6-tunneli *Freenet6* IPv6-palvelimelle. On olemassa myös paljon muita IPv6-protokollan kehitykseen kannustavia hankkeita. Suurimpia näistä ovat IPv6 Forum

[13], IPv6 Portal [14], IPv6 Promotion Council [15], IPv6 Style [16] ja 6net [17] jonka tutkimus on siirtynyt 6DISS-projektin [18] alaisuuteen. Edellä mainituista foorumeista ja portaaleista saa kaikenlaista tietoa IPv6:sta ja tutkimuksissa tehdyistä tuloksista.

1.1 Tutkimusongelma

Tässä tutkielmassa keskityttiin tutkimaan nykytilannetta Mobile IPv6 -kehityksen osalta. Erilaisille käyttöjärjestelmille on kehitetty Mobile IPv6 -toteutuksia, mutta vasta harva niistä tukee kaikkia Mobile IPv6:n osa-alueita. Pisimmällä kehityksessä tällä hetkellä ovat Helsingin Teknillisessä Korkeakoulussa kehitetty MIPL Linuxille sekä japanilaisten yliopistojen ja yritysten yhteistyössä kehittämä KAME/SHISA BSD-käyttöjärjestelmille. Kummatkin näistä tukevat jo lähes täysin Mobile IPv6 -toiminnallisuutta. Suuret reititinvalmistajat eivät vielä ole lähteneet kehittämään toiminnallisuutta MIPv6-kotiagentteja silmällä pitäen, mutta Ciscolta löytyy uusimpiin IOS-versioihin tuki kotiagentille IPsec-tukea lukuun ottamatta.

Tutkimuksen aikana suunniteltiin ja toteutettiin Mobile IPv6 -tutkimusverkko, jossa testattiin Mobile IPv6 toiminnallisuutta eri käyttöjärjestelmien ja niille kehitettyjen toteutusten kesken. Mukaan otettiin MIPL:n, KAME/SHISA:n ja Ciscon toteutukset. MIPL:ä ja KAME/SHISA:a käytettiin jokaisessa Mobile IPv6:ta tukevassa laitteessa liikkuvana päätelaitteena, liikennöintikumppanina tai kotiagenttina. Ciscoa vain käytettiin kotiagenttina. Testeissä mitattiin muun muassa viiveitä yhteysvastuun vaihtojen aikana sekä toteutusten yleistä toimivuutta keskenään.

1.2 Lukurakenne

Johdannon jälkeen kappaleessa 2 käydään läpi IPv6-protokollan toiminnallisuutta ja uusia ominaisuuksia verrattuna IPv4-protokollaan. Lisäksi Mobile IPv6 -protokollaa esitellään yksityiskohtaisemmin ja analysoidaan sen suorituskykyä aiemmin tehtyjen tutkimusten perusteella.

IPv6- sekä MIPv6 -tuesta eri käyttöjärjestelmissä selvitetään vielä kappaleen 2 lopuksi. Kappaleessa 3 selvitetään MIPv6 yhteentoimivuutta laboratorion tutkimusympäristössä tehdyssä tutkimuksessa. Kappaleessa esitellään tutkimusympäristö laitteistoineen, kerrotaan käytetyistä testausmenetelmistä ja saaduista tuloksista.

Kappale 4 sisältää tietoa IPv6-protokollan verifioinnista ja testauksesta sekä esitellään testausta varten kehitettyjä testausympäristöjä. Kappaleessa 5 käydään aihe

yhteenvetona läpi ja ehdotetaan jatkotutkimusaiheita. Dokumentin lopussa on nähtävillä liitteitä verkkoympäristön laitteiden esimerkki konfiguraatioista ja tehdyistä testeistä.

2 IPv6 ja liikkuvuus

Nykyisin käytössä oleva IPv4-protokolla on toimittanut tehtävänsä hyvin ja eikä sitä ole aikojen saatossa olennaisesti muutettu sen historian aikana 80-luvun alusta lähtien [19]. IPv4 on tarjonnut helposti toteutettavan ja hyvin skaalautuvan alustan tämän päivän Internetille.

IPv4-protokolla on tuonut tullessaan myös ongelmia internetin räjähdysmäisen kasvun myötä. Protokollaa suunniteltaessa ei otettu riittävästi huomioon seuraavia asioita:

- IP-osoitteiden riittämättömyyttä tulevaisuuden tarpeiden mukaan,
- tarvetta helpommalle konfiguraatiolle,
- IP-tason turvallisuusvaatimuksia,
- tarvetta tukea palvelunlaatua,
- reititystaulujen kasvua ja niiden käsittelyä runkoreitittimissä.

IPv4-protokollan tuoman kokemuksen mukaisesti IETF (Internet Engineering Task Forse) alkoi kehittää uutta protokollaa, joka aiemmin tunnettiin nimellä IPng ja nykyään nimellä IP version 6 (IPv6). [20][25]

2.1 IPv6 ja sen uudet ominaisuudet

IPv6 sisältää monia parannuksia verrattuna edeltäjäänsä IPv4:en. IPv6-protokollaa on pyritty myös yksinkertaistamaan vähentämällä otsikkokenttien määrää, jotta paketin käsittely erilaisissa verkkolaitteissa olisi nopeampaa ja tehokkaampaa. Tärkeimpiä muutoksia ovat [25]:

- **Uusi otsikkorakenne**. IPv6-protokollaan on kehitetty uusi otsikkorakenne, jolla pyritään pitämään otsikon koko kurissa. IPv6-protokollassa on jätetty pois tai tehty optioiksi tarpeettomaksi katsottuja IPv4-otsikkokenttiä ja siirtämällä niitä laajennusotsikkoihin. Lisäksi, IPv6-paketin perusotsikot ja optiokenttä on rajattu 64 bittiin. Tällä on pyritty parantamaan pakettien käsittelynopeutta paketin kulkeman reitin varrella.

- **Laajennettu osoiteavaruus**. Kun IPv4-protokollassa on tarjolla 32 bittinen osoiteavaruus, niin IPv6-protokollassa osoitteita on 128 bittinen määrä. Tämä tarkoittaa sitä, että IPv4:ssä osoitteita on 2^{32} kappaletta, kun taas IPv6:ssa osoitteita on huomattavasti enemmän, eli 2^{128} kappaletta. IPv6-osoiteavaruuden johdosta muun muassa ongelmia aiheuttava NAT ei ole enää tarpeellinen.

- **Tuki laajennoksille ja optioille**. IPv6 sallii joustavamman tavan käsitellä laajennoksia kuin IPv4. IPv6:ssä tätä kutsutaan otsikkoketjuksi. Tässä menetelmässä tehokasta on se, että solmut tarkastavat otsikoista vai ne, jotka niille ovat tärkeitä. Eli esimerkiksi, reitittimet tarkastavat paketista vain hyppykohtaiset optiot ja reititysotsikon.

- **Parempi tuki palvelunlaadulle**. IPv6-pakettiin on lisätty uusi kenttä, Vuotunniste (engl. Flow Label), jonka avulla on mahdollisuus merkitä paketteja sen mukaan millaiseen luokkaan ne kuuluvat. Tämän avulla reitittimet tietävät miten kyseisiä paketteja tulisi käsitellä reitin varrella. Tällä tavoin voidaan toteuttaa liikenteen palvelunlaatua.

- **Tietoturva**. IPv6-protokollaan on lisätty tuki tietoturvalaajennuksille. IPv4-protokollan yhtenä heikkouksena olikin tietoturvaominaisuuksien puute. IPv6-protokollassa käytetään IPsec-protokollaa, joka mahdollistaa tiedon eheyden ja lähettäjän oikeellisuuden varmistamisen. [21]

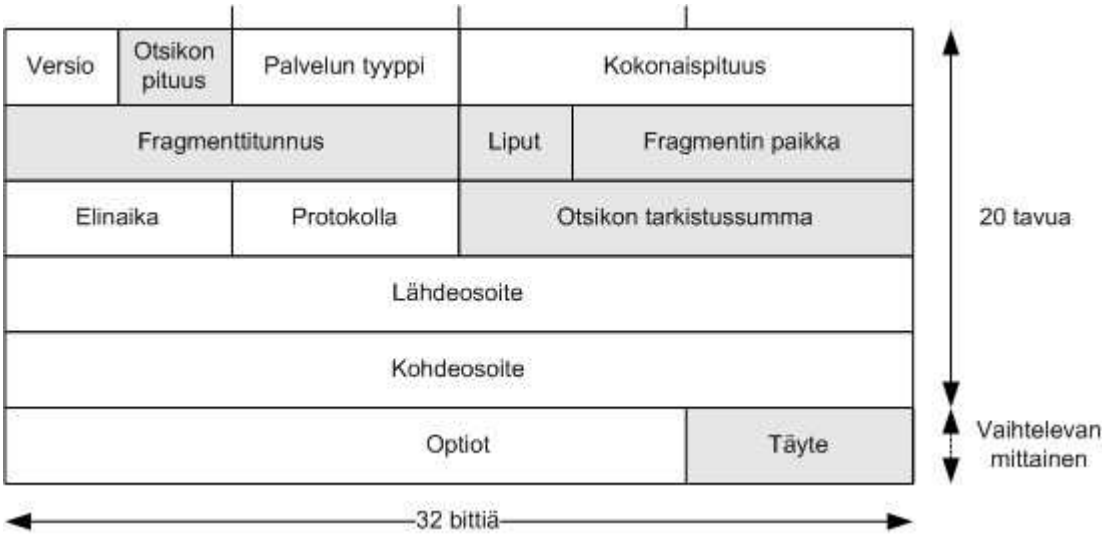
- **IP-osoitteen autokonfigurointi**. IPv6-protokolla tuo mukanaan parannuksia myös osoitteiden automaattiseen määrittelyyn. Tilaton IP-osoitteiden autokonfigurointi (engl. Stateless Address Autoconfiguration) mahdollistaa IP-osoitteiden konfiguroimisen automaattisesti.

- **Uusi protokolla naapurin havaitsemiseen**. Neighbour Discovery - protokolla on sarja ICMPv6-viestejä (Internet Control Protocol for IPv6), jotka ylläpitävät vuorovaikutusta naapurisolmujen kanssa. Tämä korvaa IPv4-protokollan ARP ja ICMPv4-viestit (Router Discovery ja Redirect multicast ja unicast Neighbour Discovery -viesteillä.

2.1.1 Yksinkertaistettu IPv6-paketin otsikkorakenne

IPv6-otsikkorakenne on yksinkertaistunut verrattuna IPv4-protokollaan. Otsikkokenttien lukumäärää on vähennetty joko jättämällä otsikosta kenttiä kokonaan pois tai muuttamalla niitä optioiksi. IPv4-protokollassa kenttien yhteenlaskettu koko on 20 tavua (160 bittiä), kun taas IPv6-paketin otsikossa kenttien koko on yhteensä 40

tavua (320 bittiä). Tällä on pyritty vähentämään pakettien prosessointiin kuluvaa aikaa verkon solmuissa paketin kulkeman reitin varrella. Kuvassa 2.1 on IPv4-paketin otsikkorakennetta ja tummennetulla on kuvattu ne otsikkokentät, jotka on jätetty pois IPv6-paketin otsikkorakenteesta.



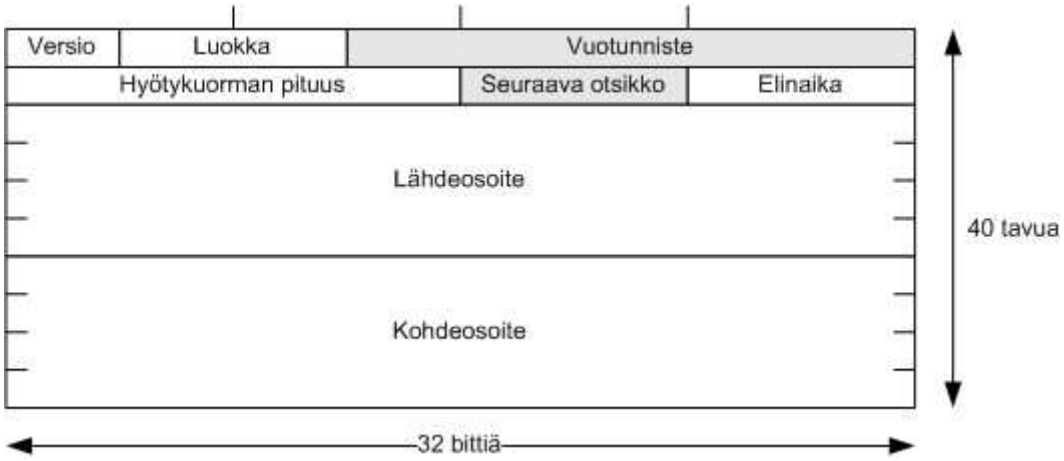
Kuva 2.1: IPv4-paketin rakenne.

IPv6-paketin perusotsikko sisältää kahdeksan kenttää ja on kooltaan 40 tavua. IPv6-paketista on jätetty pois fragmentaatioon liittyvät kentät. Yksi syy fragmenttikenttien jättämiseksi pois on se, että fragmentaatiota ei käsitellä reitittimissä ja tarkistussumma ei ole käytössä verkkokerroksella. Sen sijaan paketin lähettäjä käsittelee fragmentaatiota ja tarkistussummaa käytetään siirtoyhteys- ja kuljetuskerroksilla. IP-paketin prosessoinnin helpottamiseksi ja tehostamiseksi verkkolaitteissa IPv6-paketin perusotsikot ja optiokenttä on rajattu 64 bittiin. Kuvassa 2.2 on tummennettuna uudet kentät; vuotunniste ja seuraava otsikko.

IPv6-paketti koostuu kahdeksasta otsikkokentästä sekä mahdollisista optionaalisista laajennusotsikoista. Jokaisesta laajennusotsikosta löytyy *Seuraava otsikko* -kenttä, joka osoittaa seuraavan laajennusotsikon tyypin tai protokollapinossa ylemmältä kerrokselta saadun datan tyypin. Kuvassa 2.2 IPv6-perusotsikko, jonka koko on 40 tavua. Tämä otsikko on mukana jokaisessa IPv6-paketissa, joka annetaan alemmille kerroksille eli pienimmillään IP-paketissa kulkee vain pelkkä IPv6-otsikko.

2.1.2 IPv6-laajennusotsikot

IPv6-protokollasta löytyy laajennuksia ajatellen tärkeä kenttä, Seuraava otsikko (eng. Next Header). Tätä kenttää käytetään osoittamaan laajennusotsikon tyyppi. Laajen-



Kuva 2.2: IPv6-otsikon rakenne.

Taulukko 2.1: IPv6-otsikon kentät.

Kenttä	**Pituus (bittiä)**	**Kuvaus**
Versio	4	Protokollan versionumero = 6
Luokka	8	palvelun laadun määrittelyyn liittyvä kenttä.
Vuotunniste	20	Reitittimien käyttämä tunniste tietyn vuon tunnistamiseen.
Hyötykurman pituus	16	Ilmoittaa otsikkoa seuraavan hyötykuorman määrän tavuina. Hyötykuormaa ovat myös mahdolliset laajennusotsikot.
Seuraava otsikko	8	Kertoo IPv6-otsikkoa seuraavan otsikon tyypin.
Elinaika	8	Tätä lukua vähennetään reitittimissä yhdellä ja paketti tuhotaan elinajan ollessa nolla.
Lähdeosoite	128	Paketin lähettäjän osoite.
Kohdeosoite	128	Paketin vastaanottajan osoite.

nusotsikot sisältävät tyypillisesti optioita, joita voidaan tarvita esimerkiksi reititykseen tai tietoturvaa liittyvissä toiminnoissa. Laajennusotsikot antavat siis mahdollisuuden laajentaa IPv6-protokollan ominaisuuksia ja esimerkiksi myöhemmin tässä tutkielmassa esiteltävä Mobile IPv6 -protokolla käyttää hyväkseen laajennusotsikoita. Kuvassa 2.3 nähdään, että IPv6-paketti voi sisältää nolla tai enemmän laajennusotsikoita, jotka ovat merkitty edellisen otsikon Seuraava otsikko -kenttään. Tästä käytetään myös nimitystä otsikkoketju.

Kuva 2.3: IPv6-laajennusotsikot.

Laajennusotsikot käsitellään vastaanottajan toimesta, joten niiden käyttäminen vähentää reitittimien prosessointitarvetta ja parantaa niiden suorituskykyä optioita sisältäviä paketteja käsiteltäessä. Jos laajennusotsikot esiintyvät IPv6-paketissa, niiden järjestys on määritelty. Järjestys on seuraava:

1. IPv6-otsikko
2. Hop-by-Hop Options -otsikko
3. Destination Options -otsikko
4. Routing -otsikko
5. Fragment -otsikko
6. Authentication -otsikko
7. Encapsulation Security Payload -otsikko
8. Destination Options -otsikko
9. Ylemmän kerroksen otsikko

Listasta voidaan huomata, että Destination Options -otsikko on merkitty kahteen kertaan. Tähän on päädytty, koska IPv6-paketissa Routing -otsikon ollessa mukana lähettäjä voi haluta tiettyjä toimintoja suoritettavan jokaisessa vierailtavassa reitittimessä. Tässä tapauksessa pakettiin voidaan lisätä Kohdeoptiot-lisäkenttä myös Hop-by-Hop Options ja Routing -otsikon väliin. Jälkimmäisen Destination Options -otsikon määräämät toiminnot suoritetaan ainoastaan lopullisessa kohdeosoitteessa.

2.1.3 Osoitteistus

IPv6-protokollalle on määritelty osoitearkkitehtuuri RFC:ssä 4291 [35]. IPv6-osoitteet jaetaan kolmeen osoitetyyppiin: unicast, anycast ja multicast. *Broadcast-osoite* on poistettu kokonaan IPv6:sta ja toiminnallisuus on korvattu multicast-osoitteilla.

IPv6-osoitteen ulkomuoto eroaa huomattavasti IPv4-osoitteen muodosta. Suurin syy osoitteen esitysmuodon muuttamiselle on ollut 128 bittiseksi muuttunut osoiteavaruus, jolloin IPv4-muotoinen osoite olisi muuttunut liian epäselväksi. IPv6-osoite muodostuu kahdeksasta heksadesimaalisessa muodossa olevasta osasta, joissa jokainen 16 bitin osa esitetään numeraalisesti 0-9 tai merkein a-f (10-15). Nämä neljän heksaluvun osat on jaoteltu toisistaan ":"-merkkien avulla. IPv6-osoite voidaan myös lyhentää, jos jokin osa osoitteessa sisältää vain nolla-arvoja. Lyhennetty osoite esitetään "::"-merkinnän avulla. Tätä merkintää voidaan käyttää vain yhden kerran osoitteessa.

IPv6-osoite voisi olla esimerkiksi `3ffe:abcd:0:0:0:263d:1f2a:1` ja lyhennetyssä muodossa `3ffe:abcd::263d:1f2a:1`.

Unicast-osoite on päätelaitteen tavallinen IP-osoite, joka asetetaan verkkoon kytketyn laitteen yhdelle verkkoliitännälle. Tähän osoitteeseen lähetetyt paketit toimitetaan juuri kyseiselle liitännälle. Unicast-osoiteet on jaoteltu eri tyyppeihin riippuen siitä, minkälaisessa ympäristössä osoitetta on tarkoitus käyttää. *Global unicast* -osoitteet on tarkoitettu tiedonsiirtoon, joka tapahtuu maailmanlaajuisesti, joten osoite vastaa tavallista IPv4-osoitetta. *Link-local* -osoitteita käytetään tiedonsiirtoon saman verkkoalueen sisällä. *Site-local* -osoitteet voidaan reitittää ainoastaan tietyn hallinnollisen verkkoalueen sisällä.

Toisin kuin unicast, *anycast-osoite* voidaan asettaa joukolle liityntöjä. Paketit, jotka lähetetään anycast-osoitteeseen, toimitetaan vain yhdelle anycast-osoitteen määrittämistä liitynnöistä, yleensä lähimmälle liitynnälle. Anycast-osoitetta ei voida asettaa kuin reitittimille eikä sitä ei voi käyttää paketin lähdeosoitteena.

Multicast-osoite voidaan anycastin tapaan myös asettaa joukolle liityntöjä. Multi-

Taulukko 2.2: Laajennusotsikoiden tyypit

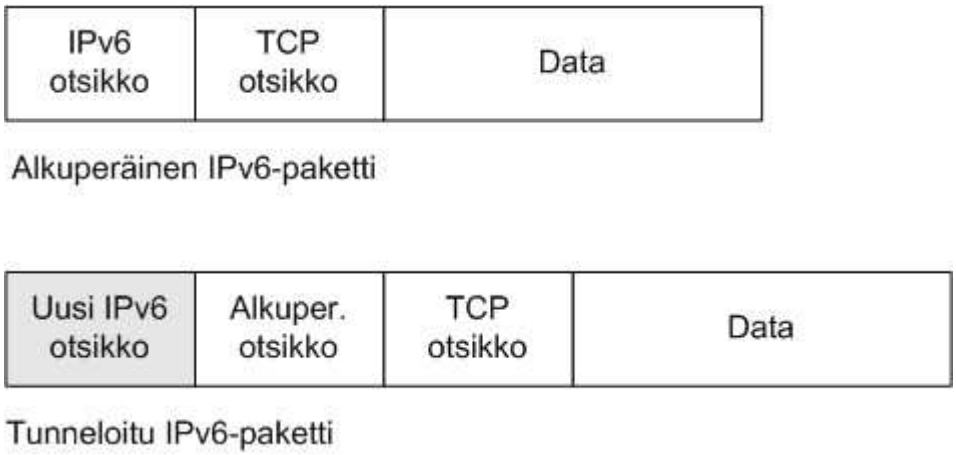
Otsikon tyyppi	**Seuraavan otsikon arvo**	**Kuvaus**
Ei laajennusotsikkoa	59	Kun IPv6-otsikon tai laajennusotsikon jälkeen ei tule enää uutta laajennusotsikkoa.
Hop-by-Hop Options	0	Otsikko käsitellään jokaisella reitittimellä, jonka kautta paketti kulkee. Jos on käytössä, niin oltava ensimmäinen otsikko IPv6 perusotsikon jälkeen.
Destination Options	60	Sisältää optioita, jotka käsitellään paketin kohdeosoitteessa. Voi seurata hop-by-hop -otsikkoa, jossa tapauksessa otsikko käsitellään myös jokaisella reitittimellä. Otsikko voi myös esiintyä ESP-otsikon jälkeen, jolloin se käsitellään vasta kohteessa.
Routing	43	Mahdollistaa tietyn reitin käyttämisen. Otsikkoon voidaan määritellä reitittimet, joiden kautta paketin kulkee kohteeseen.
Fragment	44	Otsikkoa käytetään jos paketin koko ylittää MTU:n. Käytetään paketin pilkkomiseen lähteessä ja uudelleenkokoamiseen kohteessa.
Authentication Encapsulation Security Payload	51 52	Authentication- ja ESP-otsikoita käytetään IPsec-protokollassa varmistamaan paketin eheys ja todentaa sen alkuperä.
Upper Layer	6 (TCP), 17 (UDP)	Ylemmän kerroksen (kuljetus) otsikot ovat tyypillisesti otsikoita, joita käytetään paketin sisällä kuljettamaan dataa. Yleensä nämä protokollat ovat TCP ja UDP.
Mobility	135	Käytetään liikkuvissa päätelaitteissa, liikennöintikumppaneissa ja kotiagenteissa viesteissä, jotka liittyvät sidoksien luomiseen ja hallintaan.

cast-osoitteeseen lähetetyt paketit lähetetään kaikille osoitteen määrittämille liityntöille, toisin kuin anycast-osoitteessa vain yhdelle. Myöskään multicast-osoitetta ei voi käyttää paketin lähdeosoitteena.

IPv6-osoitemallissa jokaisella liitynnällä on ainakin yksi unicast-osoite. Yksittäinen liityntä voi sisältää myös muita IPv6-osoitteita, jotka voivat olla mitä tyyppiä hyvänsä (unicast, anycast, multicast). Tarvittaessa osoitteessa ilmaistaan myös verkon prefiksi siten, että osoitteeseen liitetään "/"-merkin jälkeen lukuna se, kuinka monta bittiä vasemmalta luettuna kuuluu prefiksin määrittämään osaan. [35]

2.1.4 Tunnelointi

Tunneloinnilla tarkoitetaan prosessia, jossa päätelaite kapseloi IPv6-paketin toiseen IPv6-otsikkoon. Tätä kutsutaan IPv6-kapsuloinniksi (engl. *IPv6-in-IPv6 encapsulation*). [31] Tunneloinnin voi suorittaa paketin alkuperäinen lähettäjä tai muu jokin laite paketin kulkeman reitin varrella. Tunneloinnin suorittava laite lisää alkuperäisen paketin eteen uuden IPv6-otsikon sekä mahdollisia laajennusotsikoita kuvan 2.4 mukaisesti. Tunnelin päätepiste asetetaan uuden otsikon kohdeosoitteeksi. Kun



Kuva 2.4: IPv6 tunnelointi.

tunnelin päätepisteessä oleva päälaite vastaanottaa paketin ja havaitsee, että paketissa on kaksi IPv6-otsikkoa peräkkäin, se poistaa niistä ulommaisen. Tämän jälkeen paketti reititetään sisimpänä olevan otsikon osoittamaan kohdeosoitteeseen.

2.1.5 ICMPv6

ICMPv6-protokollaa (engl. Internet Control Message Protocol) käytetään kuljettamaan IP valvontaviestejä erilaisiin tarkoituksiin. ICMPv6 sisältää samoja toiminto-

ja kuin IPv4-protokollan vastaava ICMP. Näitä toimintoja ovat kuljetus- ja reititysvirheistä raportointi ja echo-palvelu vian etsintään. IPv6-paketin seuraava otsikko -kentän arvo 58 osoittaa ICMPv6-pakettiin.

ICMPv6-protokollaa käytetään *Neighbor Discovery* (ND) ja *Multicast Listener Discovery* (MLD) -protokollissa. ICMPv6 on määritelty dokumentissa RFC 2463 ja päivitetty dokumentissa RFC 4443. [30]

2.1.6 Neighbor Discovery

Naapurin havaitseminen (Neighbour Discovery) on protokolla, jonka avulla samassa verkossa olevat laitteet mainostavat olemassaolostaan naapureilleen ja samalla saavat tietoa naapureidensa olemassaolosta. Tämä protokolla kuuluu olennaisena osana IPv6-protokollan toimintaan. Naapurin havaitseminen IPv6:ssa vastaa seuraavia IPv4-protokollia: Router Discovery (RDISC), Address Resolution Protocol (ARP) ja ICMPv4 redirect.

Naapurin havaitseminen on määritelty seuraavissa dokumenteissa: RFC 2461 Neighbor Discovery for IP Version 6 (päivitetty RFC 4311) [26][27], RFC 2462 IPv6 Stateless Address Autoconfiguration [28] ja RCF 4443 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 [30]. Näiden protokollien avulla IPv6-laite havaitsee samassa verkossa olevat IPv6-laitteet, niin päätelaitteet kuin reitittimetkin. Reitittimien lähettämillä viesteillä päätelaitteet voivat automaattisesti konfiguroida osoitteen itselleen. Naapurin havaitsemis protokollien avulla voidaan selvittää osoitteen ainutlaatuisuutta (Duplicate Address Detection), jolla estetään päätelaitteiden konfiguroida itselleen sellaista osoitetta, joka verkosta jo löytyy.

Naapurin havaitseminen käyttää seuraavia ICMPv6-protokollan viestejä:

- **Router Advertisement (RA)**. RA-viesti voidaan lähettää vasteena RS-viestiin tai ajoittain pyytämättä verkon multicast-osoitteisiin. Kun reititintä kutsutaan RS-viestillä, RA-viesti lähetetään kutsuvan päätelaitteen unicast-osoitteeseen. Ajoittain lähetetty viesti lähetetään multicast-osoitteeseen kaikille päätelaitteille verkossa.

- **Router Solicitation (RS)**. RS-viestejä lähettävät päätelaitteet etsiessään verkosta reitittimiä. RS-viestit lähetetään kaikkien reitittimien multicast-osoitteisiin. Tämä multicast-osoite on reitittimen link-local multicast-osoite. Kaikki RS-viestin vastaanottaneet reitittimet vastaavat RA-viestillä.

- **Neighbor Advertisement (NA)**. NA-viesti voidaan lähettää vasteena pyyntöihin tai ajoittain. NA-viestiä voidaan käyttää myös apuna osoitteen muodosta-

misessa ja sen ainutlaatuisuuden tarkistamisessa.

- **Neighbor Solicitation (NS)**. NS-viestiä voidaan käyttää monenlaisissa tarkoituksissa, kuten osoitteen muodostamisessa, NUD- ja DAD-prosesseissa. NS-viesti voi olla multicast-viesti osoitteen muodostamisessa ja unicast-viesti naapurin havaitsemisessa.
- **Redirect**. Redirect-viestiä käyttää IPv6-reititin, kun se informoi päätelaitetta paremmasta reitistä, kuin johon paketti alun perin oli reititetty. Näitä viestejä lähetetään vain unicast-liikenteelle, -osoitteisiin ja vain päätelaitteet käsittelevät nämä viestit.

Neighbor Discovery -viestien avulla voidaan suorittaa erilaisia toimintoja, jotka ovat seuraavanlaisia:

Address Resolution

Osoitteen ratkaisemisessa käytetään *Address Resolution*-menetelmää. Tämä vastaa IPv4:n ARP-protokollaa. Jotta päätelaite voisi liikennöidä toisen laitteen kanssa, sen täytyy tietää toisen laitteen MAC-osoite. IPv6:n osoitteen ratkaisu sisältää NS- ja NA-viestien vaihdon, joilla pyritään selvittämään tämä halutun kohteen MAC-osoite.

Prosessin aloittava laite lähettää kohteen IP-osoitteen sisältämän NS-viestin kohteena olevan laitteen solicited-node multicast -osoitteeseen. Kun kohde vastaanottaa tämän NS-viestin, se päivittää omaan naapurivälimuistiin lähettäjän MAC- ja IP-osoitteen. Seuraavaksi kohde vastaa lähettäjälle unicast NA-viestin, joka sisältää *Target Link-Layer Address* -optiossa kohteen kohteen MAC-osoitteen. Kun lähettäjä vastaanottaa NA-viestin, se päivittää oman naapurivälimuistin viestin sisältämällä kohteen MAC-osoitteella.

Duplicate Address Detection

Osoitteen ainutlaatuisuuden tarkistaminen, eli DAD (Duplicate Address Detection), on menetelmä joka suoritetaan osoitteen konfiguroimisen jälkeen. DAD:n avulla tarkistetaan, ettei samassa verkossa ole jo muita päätelaitteita, joilla on sama osoite. IPv6-laitteet käyttävät *Neighbor Solicitation* -viestejä suorittaessaan DAD:a. Tällöin NS-viestin *Target Address* -kenttä sisältää tilapäisen (engl. tentative) IPv6-osoitteen jolle DAD suoritetaan, eli NS-viestin lähettäjän konfiguroima tilapäinen IP-osoite.

Jos toinen samanlainen IP-osoite on jo verkossa käytössä, tämän osoitteen käyttäjä huomaa sen DAD:n suorittajan NS-viestistä ja kuittaa NA-viestillä. Tämä NA-

viesti sisältää *Target Address* -kentässä IP-osoitteen. DAD:n suorittajan vastaanottaessa tämän NA-viestin, se lopettaa tämän duplikaatin IP-osoitteen käytön. Jos se ei kuitenkaan vastaanota NA-viestiä, ottaa se IP-osoitteen käyttöön. [26]

Router Discovery

Reitittimien havaitseminen (Router Discovery) on menetelmä, jolla pyritään etsimään reitittimiä verkosta. IPv6:n RA-viestissä on *Router Lifetime* -kenttä, joka ilmaisee elinajan jona reititin oletetaan olevan oletusreititin. Jos reitittimeen ei saada enää yhteyttä, tilanne huomataan *Neighbor Unreachability Detection* -menetelmän avulla RA-viestin Router Lifetime -kentän sijaan. Tämän jälkeen uusi reititin valitaan oletusreititin -listasta.

Oletusreitittimien konfiguroinnin lisäksi Router Discoveryn avulla voidaan konfiguroida myös IPv6-otsikon Elinaika-kentän arvoa, määrittää laitteen käyttämään tilallista autokonfiguraatiota, ajastimia joita käytetään saavuttamattomuuden havaitsemisessa sekä NS-viestien uudelleenohjauksessa, listaa verkon prefixeistä ja verkon MTU:ta (Maximum Transmission Unit). [25]

Router Discovery -prosessi toimii seuraavasti:

- IPv6 reitittimet lähettävät ajoittain RA-viestejä ilmoittamalla olemassolostaan. Ne myös tarjoavat konfiguraatioparametreja, kuten elinaika, MTU ja prefixit.
- IPv6 päätelaitteet vastaanottavat RA-viestejä ja käyttävät viestin sisältöä ylläpitäessään oletusreititin- ja prefix-listaa sekä muita konfiguraatioparametreja.
- Päätelaite alkaa lähettää RS-viestejä verkkoon all-routers multicast -osoitteeseen (FF02::2). RS-viestin saaneet reitittimet vastaavat viestiin lähettämällä RA-viestin kyseiselle päätelaitteelle.

RA-viestejä reitittimet voivat siis lähettää ajoittain kaikille verkon laitteille multicast osoitteeseen tai vastauksena päätelaitteen lähettämään RS-viestiin.

Neighbor Unreachability Detection

Neighbor Unreachability Detection -prosessissa tarkastetaan, että ensimmäinen hyppy paketin reitin varrella on saavutettavissa. Hyppy voi olla joko päätelaite–päätelaite tai päätelaite-reititin. Naapurin saavutettavuutta voidaan testata lähettämällä unicast NS-viesti ja saada kuittaus pyydettynä NA-viestinä (solicited Neighbor Advertisement), jolloin *Solicited* -lippu on asetettu arvoon 1. [26]

2.1.7 Osoitteen automaattinen konfigurointi

Yksi IPv6-protokollan käytännöllisimmistä ominaisuuksista on kyky konfiguroida automaattisesti IP-osoite ilman DHCP-protokollaakin. IPv6-päätelaite voi oletuksena konfiguroida link-local -osoitteen jokaiselle liitynnälleen. Reitittimen havaitsemisen (Router Discovery) avulla päätelaite voi myös määrittää reitittimien osoitteita, verkon prefixejä ja muita konfiguraatioon liittyviä parametreja. Automaattinen osoitteen konfigurointi voidaan suorittaa tilattomalla tai tilallisella menetelmällä. Se kumpaa menetelmää käytetään, määräytyy *Router Advertisement* -viestiin merkityillä lipuilla. Kumpaakin menetelmää voidaan käyttää myös yhtä aikaa.

Tilaton osoitteen konfigurointi

Tilatonta osoitteen muodostusta käytettäessä osoitteen alkuosa vaihtuu uuden verkon prefiksin mukaiseksi. Osoitteen loppuosa ei yleensä muutu, koska se muodostetaan staattisesta ja mahdollisimman yksilöivästä tekijästä, kuten verkkokortin MAC-osoitteesta. Tilaton osoitteen konfigurointi perustuu kuittauksesta *Router Advertisement* -viesteihin, joissa kenttien *Managed Address Configuration* ja *Other Stateful Configuration* -liput on asetettu arvoon 0 ja yhteen tai useampaan *Prefix Information* -optioon. [28]

Tilallinen osoitteen konfigurointi.

Tilallinen konfigurointi perustuu DHCPv6-protokollaan. Jos reitittimellä on käytössä tämä menetelmä, liikkuvalle päätelaitteelle annetaan DHCP-palvelimelta osoite, joka ei ole vielä kenenkään käytössä. Päätelaite käyttää tilallista osoitteen konfigurointia, kun se vastaanottaa Router Advertisement -viestin ilman prefix-optioita, jossa *Managed Address Configuration* tai *Other Stateful Configuration* -lippu on asetettu arvoon 1.

2.1.8 Tietoturva

Tietoturva on noussut tärkeäksi osa-alueeksi Internetin kasvun myötä. IPv4-protokollaa suunniteltaessa ei tietoturvaa otettu huomioon, joten nykyinen IPv4-liikenne on ollut helposti kaapattavissa, analysoitavissa ja muutettavissa. Ilman tietoturvaa myöskään paketin lähettäjän oikeellisuutta ja laillisuutta ei pystytä todentamaan.

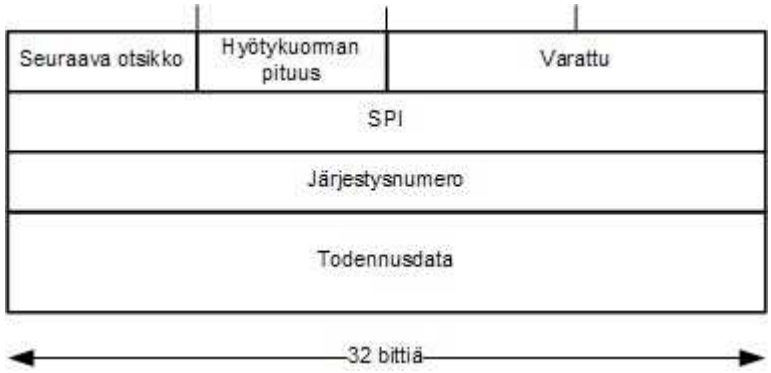
IPv6-protokollaa suunniteltaessa Internetin tietoturvauhkat ovat tulleet selviksi nykyisen protokollan käytön myötä. IPv6-protokollaan onkin otettu tietoturva osak-

si protokollaa. Tietoturva-arkkitehtuurina käytetään IPsec:ä (IP Security Architecture), joka on joukko protokollia yhteyksien turvaamiseen. Nämä yhdessä tarjoavat osapuolten todennuksen, salauksen ja tiedon eheyden varmistamisen.

IPsec tarjoaa kaksi protokollaa tietoturvapalveluihin, nämä ovat: AH (Authentication Header) ja ESP (Encapsulating Security Payload). AH tarjoaa todennuksen ja takaa viestien eheyden, mutta ei luottamuksellisuutta. ESP:tä käytetään puolestaan liikenteen salaamiseen. [21]

Käyttäjien todennus AH-protokollan avulla

Authentication-laajennusotsikon avulla voidaan todentaa paketin eheys tietoturvaryhmän (Security Association SA) luonnin jälkeen. Paketin eheys voidaan todentaa tarkistamalla, että paketti on peräisin oikealta lähettäjältä. Authentication-otsikko suojaa suurimman osan paketista, joten mahdolliset muutokset paketin siirtyessä osapuolten välillä huomataan vastaanottopäässä. Tämän laajennusotsikon tunniste edellisen laajennusotsikon Seuraava otsikko -kentässä on 51. Seuraavassa 2.5 kuvassa on esitetty Authentication-otsikon rakenne. [21]



Kuva 2.5: Authentication-otsikon rakenne.

Authentication-otsikon rakenne koostuu seuraavista kentistä:

Authentication-otsikkoa käytettäessä lähettäjä laskee tarkistussumman paketin sisällöstä. Tätä kutsutaan myös eheyden tarkistus arvoksi (engl. Integrity Check Value ICV), joka lasketaan paketin lähettämisen aikana tietyistä muuttumattomista (engl. immutable) kentistä.

Paketin vastaanottaja laskee tarkistussumman paketin sisällön, SPI:n ja salaisen avaimen perusteella. Tämän jälkeen se vertaa laskemaansa lukua Todennusdata-kentän lukuun. Jos luku on sama, varmistuu lähettäjän aitous. Vastaanottaja olla varma siitä, että paketin lähettäjä tietää salaisen avaimen ja näin ollen kuuluu sa-

Taulukko 2.3: Authentication-otsikon kentät

Kenttä	**Pituus (bittiä)**	**Kuvaus**
Seuraava otsikko	8	Ilmoittaa paketissa seuraavaksi tulevan laajennusotsikon.
Pituus	8	Kertoo Todennusdata-kentän pituuden 32-bitin yksiköissä.
Varattu	16	Varattu tulevaisuuden varalle. Alustetaan arvolla 0 ja otetaan huomioon Todennusdata-kentästä laskettavaan tarkistussummaan, mutta hylätään vastaanottopäässä.
SPI	32	Mielivaltainen 32 bittinen arvo, joka on yhdistelmä vastaanottajan IP-osoitetta ja AH-protokollaa. SPI-arvot 1-255 on varattuja tulevaisuutta varten. Arvo 0 tarkoittaa, ettei turvallisuusryhmää ole käytössä.
Järjestysnumero	32	Järjestysnumero, joka toimii laskurina.
Todennusdata	Vaihtuvamittainen	Kenttä johon sijoitetaan tarkistussumma, joka lasketaan tietyistä IPv6-otsikon kentistä, tietyistä lisäkentistä, ylemmän kerrosten datasta sekä salaisesta avaimesta, joka on tiedossa samaan turvallisuusryhmään kuuluvilla jäsenillä.

maan turvallisuusryhmään. Authentication-otsikko ei muuta eikä salaa sen jälkeen tulevaa dataa mitenkään, eli data liikkuu selväkielisenä verkossa. [21]

Security Association ja Security Policy Database

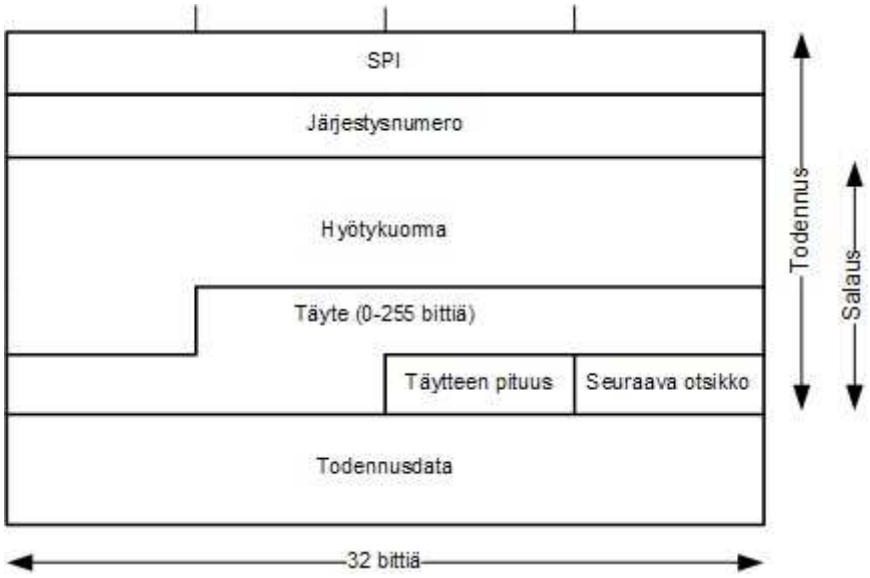
Päätös siitä mitä paketteja halutaan turvata Authentication-otsikolla, riippuu siitä millainen tietoturvaryhmä (Security Association SA) on luotu tai konfiguroitu lähettäjälle ja vastaanottajalle. Tietoturvamenetelmä määritellään tietoturvamenettely tietokannan mukaan (engl. Security Policy Database SPD) ja se konfiguroidaan sekä lähettäjälle että vastaanottajalle. SPD sisältää menettelytavat siitä mitä paketteja tulee suojata. SPD:en on määritelty valitsimia, jotka sisältävät lähettäjän ja vastaanottajan IP-osoitteita ja protokollanumeroita. SPD voi esimerkiksi sisältää tiedon,

että kaikki TCP-liikenne kahden laitteen välillä suojataan Authentication-otsikolla. Menettely voi myös viitata SA:han jonka avulla tiedetään, mitä avaimia, algoritmeja ja SPI-arvoja pitää käyttää paketin suojaamiseen. Täten kaikki laitteiden välinen liikenne menee SPD-tarkistuksen läpi jolloin tiedetään mitkä paketit pitäisi suojata. Tällainen menettely luodaan SA:ta luotaessa. [21]

Liikenteen salaaminen ESP-protokollalla

Authentication-otsikkoa käytettäessä ongelmana on datan siirtyminen selväkielisenä. IPsec tarjoaa ESP-protokollan (Encapsulated Security Protocol), joka mahdollistaa pakettien luottamuksellisuuden käyttäen salausalgoritmeja. ESP-laajennusotsikko sijoitetaan IPv6-pakettiin viimeiseksi laajennusotsikoksi, joka vielä halutaan lähettää selväkielisenä. ESP-laajennusotsikkokin on itse asiassa kuitenkin salattu osittain.

ESP-otsikko luodaan salausprosessissa paketin lähetyspäässä. Otsikko sisältää tiedot jotka vaaditaan, jotta vastaanottopää kykenee purkamaan paketin salauksen ja todentaa sen sisältö. Otsikko sisältää SPI- ja Järjestysnumero -kentät, joilla on samanlainen merkitys kuin AH-protokollassa edellä. Kuvassa 2.6 esitetään ESP-otsikon kentät. [21]



Kuva 2.6: ESP-laajennusotsikko.

Hyötykuorma-kenttä sisältää hyötykuorman, joka halutaan salata ESP:n avulla. Sitä seuraa tarvittava määrä täytettä (0-255 bittiä) ja tieto seuraavasta laajennusotsikosta, joka ilmaisee protokollan tyypin salattavalle hyötykuormalle. *Todennusda-*

ta-kentällä on sama merkitys kuin AH-protokollassa, mutta ESP:ssä todennus on valinnainen toimenpide. ESP:n ja AH:n erona on myös se, että ESP:tä käytettäessä todennus ja salaus eivät käsitä koko IP-pakettia, vaan ainoastaan otsikot jotka seuraavat ESP-otsikkoa. Täten IPv6-otsikko ei ole todennettu ESP:tä käytettäessä. [21]

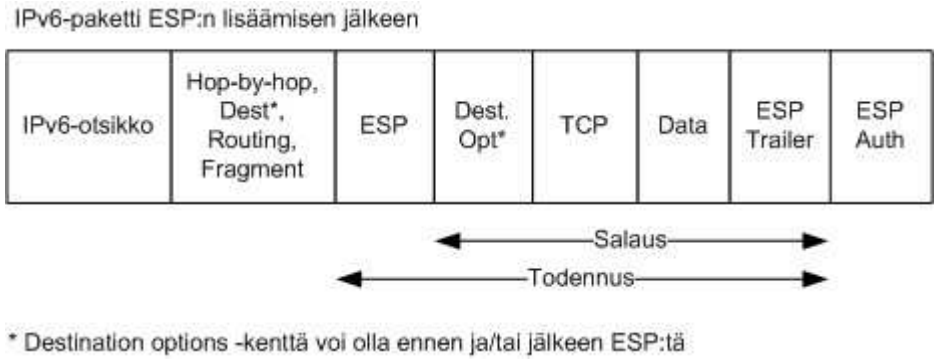
Kuljetus ja tunnelointi moodit

Sekä AH- että ESP-protokollaa voidaan käyttää kahdessa moodissa: kuljetus- ja tunnelointimoodissa. Kuljetusmoodia voidaan käyttää päätelaitteiden välillä ja se tarjoaa suojan ylemmän kerrosten protokollille ja valituille IP-otsikon kentille.

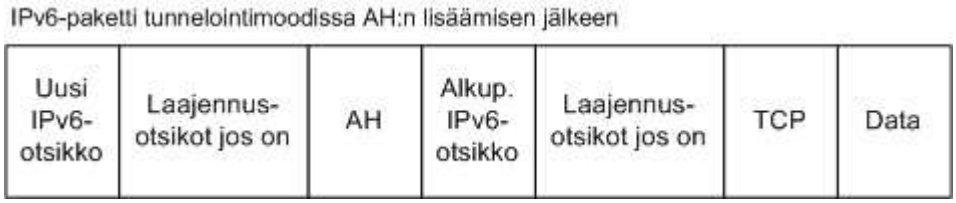
Kuljetusmoodissa AH- sekä ESP-otsikko asetetaan IPv6-otsikon ja ylemmän tason protokollien väliin tai ennen muita aikaisemmin asetettuja IPsec-otsikoita. IPv6-protokollassa AH ja ESP voidaan käsittää olevan päästä päähän hyötykuormaa, joten hop-by-hop-, reititys- ja fragmentaatio -laajennusotsikot tulee olla ennen sitä IPv6-paketissa. Destination Options -laajennusotsikko voi sijaita ennen tai jälkeen AH- tai ESP-otsikkoa. [21]

Kuva 2.7: IPv6-paketti kuljetusmoodissa AH-otsikon lisäämistä ennen ja jälkeen.

Tunnelointimoodia voidaan käyttää päätelaiteiden välillä sekä turvallisiin yhdyskäytäviin. Tunnelointimoodissa sisempi IP-otsikko kuljettaa alkuperäisiä lähde- ja kohdeosoitteita, kun taas ulompi IP-otsikko sisältää esimerkiksi tunnelin päätepisteiden osoitteet. Tässä moodissa AH- sekä ESP-suojaavat kokonaan sisemmän IP-paketin. Kuva 2.9.



Kuva 2.8: IPv6-paketti kuljetusmoodissa ESP-otsikon lisäämisen jälkeen.



Kuva 2.9: IPv6-paketti tunnelimoodissa AH:n lisäämisen jälkeen.

Kuva 2.10: IPv6-paketti tunnelimoodissa ESP:n lisäämisen jälkeen.

2.2 Mobile IPv6 (MIPv6)

Kuten jo johdannossa kerrottiin, tulee Internet muuttumaan nykyisestä muodostaan valtavasti tulevaisuudessa. IP-protokollalla tullaan toteuttamaan mitä erilaisimpia laitteita mitä erilaisimmissa sovelluskohteissa. Osoitetarpeen kasvun myötä uusi IPv6-protokolla olisi ratkaisu tähän ongelmaan. Yksi sovelluskohde on liikkuvat päätelaitteen, kuten matkapuhelimet ja kannettavat tietokoneet. Tähän asti Internet on ollut staattisesti paikallaan pysyvä, mutta tulevaisuuden skenaarioissa toivotaan liikkuvien päätelaitteiden kykenevän liikkumaan eri verkkojen alueilla yhteyksien katkeamatta.

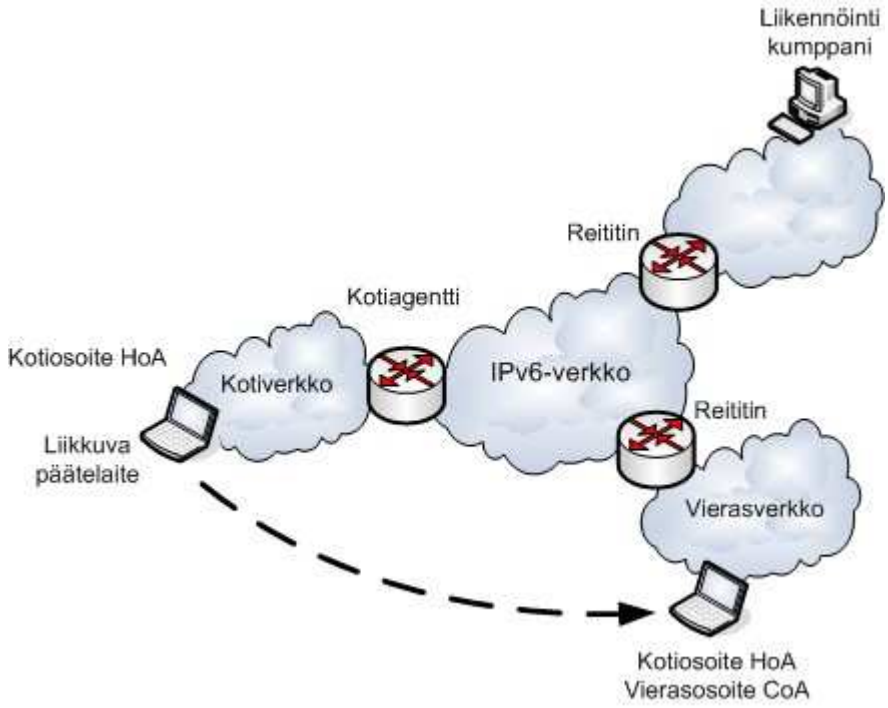
IPv6-verkossa päätelaite tarvitsee IP-osoitteen, jonka avulla se kykenee kommunikoimaan toisten laitteiden kanssa. Jos päätelaite siirtyy kotiverkon kantaman ulkopuolelle, saattaa se silloin siirtyä pois kotiverkon alueelta vieraaseen verkkoon. Vieraassa verkossa päätelaite tarvitsee uuden osoitteen, jolla se kykenee kommunikoimaan. Jos päätelaitteella on ollut kotiverkosta vieraaseen verkkoon siirryttäessä yhteyksiä muihin laitteisiin niin ne katkeavat, koska muut laitteet lähettävät paketteja päätelaitteen vanhaan osoitteeseen.

Ratkaisuksi tähän liikkuvuuden ongelmaan Internetin standardointiorganisaatio IETF on kehittänyt Mobile IP -työryhmässään Mobile IPv6 -protokollan. Se mahdollistaa päätelaitteiden liikkumisen IPv6-verkkojen välillä yhteyksien katkeamatta. [32] Seuraavissa kappaleissa kerrotaan yleisesti Mobile IPv6 -protokollan toiminnallisuudesta.

2.2.1 Toiminnallisuus

Mobile IPv6 -verkon toiminnallisuuteen liittyy komponentteja, joilla on erilaisia tehtäviä. Kuvassa 2.11 on kuvattu yksinkertainen MIPv6-verkko, jossa liikkuva päätelaite suorittaa siirtymän kotiverkosta vierasverkkoon. Verkon toiminnallisuutta voidaan kuvata seuraavasti: **Liikkuva päätelaite** (Mobile Node (MN)) on IPv6-verkossa toimiva laite, joka voi vaihtaa verkkoa ja tätä kautta osoitteita, ja ylläpitää yhteyksiä kotiosoitteensa avulla. Liikkuva päätelaite on tietoinen omasta kotiosoitteestaan ja vieraan verkon, johon se on liittynyt, osoitteesta (vierasosoite).

Kotiverkko (Home Network (HN)) on verkko, josta liikkuva päätelaite saa kotiosoitteensa. Täten myös kotiagentti sijaitsee samassa verkossa. **Kotiosoite** (Home Address (HoA)) Liikkuvan päätelaitteen liittyessä kotiverkkoon, saa se kotiosoitteensa kotiagentilta. Tämän osoitteen perusteella liikkuva päätelaite on aina saavutettavissa huolimatta sijaintipaikastaan. Kun liikkuva päätelaite liikkuu verkosta toiseen, paketit ohjataan sen kotiosoitteeseen kotiagentin luoman tunnelin kautta.



Kuva 2.11: Mobile IPv6 -verkko ja sen komponentit.

Kotiagentti (Home Agent (HA)) on reititin, joka ylläpitää rekisteriä liikkuvista päätelaitteista ja niiden osoitteista (koti- ja vierasosoitteet). Kun liikkuva päätelaite siirtyy vieraaseen verkkoon, kotiagentti reitittää paketit liikkuvan päätelaitteen sen hetkiseen osoitteeseen (CoA) IP-tunnelin kautta.

Vierasverkko (Foreign Network (FN)) on verkko, johon liikkuva päätelaite siirtyy kotiverkostaan. **Vierasosoite** (Care-of Address (CoA)) muodostetaan liikkuvalle päätelaitteelle sen liittyessä vierasverkkoon. Tilattomassa osoitteen muodostuksessa vierasosoite muodostetaan vieraan verkon verkko-osoitteesta ja liikkuvan päätelaitteen MAC-osoitteesta. Liikkuva päätelaite voi saada useitakin vierasosoitteita, mutta vain yksi vierasosoite voi olla kerrallaan rekisteröityneenä kotiagentilla. Liikkuvan päätelaitteen koti- ja vierasosoitteen yhdistämistä sanotaan sidonnaksi (engl. *binding*). Kotiagentit ja liikennöintikumppanit pitävät rekisteriä sidoksista sidosvälimuistissa (engl *binding cache*).

Liikennöintikumppani (Correspondent Node (CN)) on IPv6-verkossa toimiva laite, joka kommunikoi liikkuvan päätelaitteen kanssa. Liikennöintikumppanin ei kuitenkaan tarvitse välttämättä tarvitse tukea MIPv6-protokollaa.

Lisäksi Mobile IPv6 -protokollaan liittyy seuraavat käsitteet ja tietorakenteet:

Sidonta (Binding) on tietyn ajan kestävä yhteys liikkuvan päätelaitteen kotiosoitteen ja vierasosoitteen välillä. Tämän avulla kotiagentti osaa reitittää paketit liikkuvan päätelaitteen vierasosoitteeseen. Sidonta päivitetään, jos liikkuva päätelaite siirtyy uuteen vierasverkkoon tai sidonnan ajastin erääntyy.

Sidosvälimuisti (Binding Cache) on välimuisti RAM:ssa, johon on tallennettu rekisteri yhden tai useamman liikkuvan päätelaitteen sidoksista. Sidosvälimuistia ylläpidetään kotiagentilla sekä liikennöintikumppanilla joka tukee MIPv6-protokollaa. Sidosvälimuisti koostuu liikkuvan päätelaitteen koti- ja vierasosoitteesta sekä elinajasta. Liikennöintikumppanin sidosvälimuisti sisältää lisäksi autentikointi parametreja.

Sidospäivitys lista (Binding Update List (BUL)) on välimuisti liikkuvan päätelaitteen RAM:ssa. Lista koostuu sidoksista, joita on lähetetty kotiagentille sekä liikennöintikumppanille. Listaa voidaan käyttää myös valittaessa oikea vierasosoite suoraan liikennöintikumppanin kanssa kommunikoidessa.

2.2.2 Erot Mobile IPv4 -protokollaan

Myös nykyisin käytössä olevalle IPv4-protokollalle on määritelty liikkuvuuden tuki RFC-dokumentissa 3344 [36]. Liikkuvuuden toteuttaminen IPv4-protokollalle on kuitenkin ongelmallisempaa kuin tulevalle IPv6-protokollalle varsinkin riittämättömän IP-osoitavaruudensa takia. Ongelmallisuutensa takia liikkuvuusominaisuuksia IPv4-protollan ominaisuudessa ei kovin laajasti ole toteutettu eikä tuskin tulla toteuttamaankaan.

Dokumentti [36] määrittelee MIPv6:sta tuttujen kotiagentin, liikkuvan päätelaitteen ja liikennöintikumppanin lisäksi vierasagentin (engl. *Foreign Agent* FA). Vierasagentti ja kotiagentti yhdessä toteuttavat liikkuvan päätelaitteen liikkuvuutta. Erona kotiagenttiin, joka toimii kuten MIPv6-verkossakin, on se, että vierasagentteja tarvitaan jokaisessa verkossa johon liikkuvan päätelaitteen on mahdollista siirtyä. Vierasagentin tehtävänä on informoida liikkuvaa päätelaitetta sen nykyisestä sijainnista. Liikkuvan päätelaitteen ja liikennöintikumppanin kommunikoidessa keskenään kaikki näiden väliset paketit ohjataan vierasagentin ja kotiagentin kautta, eikä MIPv4:ssä voida käyttää MIPv6:ssa mahdollista reitin optimointia. [36]

MIPv4-protokollan kolmioreititys aiheuttaa ongelmia. Ensinnäkin vierasagentti on oltava toiminnassa jokaisessa verkossa, joihin liikkuva päätelaite kykenisi siirtymään. Tämä tuskin tulisi onnistumaan jokaisen palveluntarjoajan verkoissa ja ai-

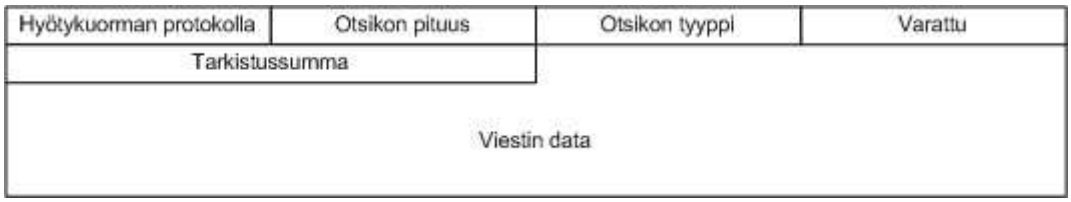
heuttaisi myös lisätöitä verkon ylläpitoon. Myös liikenteen tunneloiminen vieras- ja kotiagentin kautta kuormittaa verkkoja enemmän ja lisää viiveitä liikkuvan päätelaitteen ja liikennöintikumppanin välisessä kommunikoinnissa. MIPv4-verkkojen käyttöönotto saattaisi olla kannattavampaa, jos nämä ongelmat saataisiin ratkaistua.

2.2.3 Mobile IPv6 -laajennusotsikot

IPv6-protokollaan voi kehittää uusia toiminnallisuuksia laajennusotsikoiden avulla. Mobile IPv6 -protokolla on toteutettu näitä ominaisuuksia hyödyntäen. MIPv6-protokollassa laajennusotsikoita käytetään viestien välittämiseen ja liikkuvan päätelaitteen osoitteen käsittelyyn.

Mobility Header

Tärkein Mobile IPv6- protokollan laajennusotsikko on nimeltään *Mobility Header*. Mobility Header -laajennusotsikko osoitetaan arvolla 135 edellisen laajennusotsikon seuraava otsikko -kentässä. Sen tehtävänä on välittää liikkuvuuteen liittyviä viestejä. Kuvassa 2.4 on esitetty Mobility Header -laajennusotsikon rakennetta ja taulukossa 2.4 Mobility Header -laajennusotsikon kentät.



Kuva 2.12: Mobility Header -laajennusotsikko.

Laajennusotsikon kolmas kenttä (Otsikon tyyppi) ilmaisee Mobility Header:n kuljettaman viestin tyypin. Seuraavat viestin tyypit on määritelty Mobile IPv6 -protokollaan [32]:

- **Binding Refresh Request (BRR)**. Viestin lähettää kotiagentti tai liikennöintikumppani pyytäessään nykyistä sidosta liikkuvalta päätelaitteelta. Kun liikkuva päätelaita vastaanottaa BRR:n, lähettää se vastauksena sidospäivityksen (BU). Kotiagentti lähettää BRR:n kun sen sidosvälimuistin ajastin lähestyy umpeutumista. Liikennöintikumppani lähettää BRR:n kun sidosvälimuisti on käytössä ja sen ajastin lähestyy umpeutumista.

Taulukko 2.4: Mobility Header -laajennusotsikon kentät.

Kenttä	**Pituus (bittiä)**	**Kuvaus**
Hyötykuorman protokolla (Payload protocol)	8	Vastaa seuraava otsikko -kenttää IPv6-otsikossa. Arvo aina 59 osoittaen, että MH-otsikko on viimeinen otsikko paketissa.
Otsikon pituus (Header length)	8	Ilmoittaa MH-otsikon pituuden.
Otsikon tyyppi (Header Type)	8	Ilmoittaa kyseessä olevan viestin.
Varattu (Reserved)	8	Varattu tulevaisuutta varten.
Tarkistussumma (Checksum)	16	Otsikon tarkistussumma.
Viestin data (Message data)	Vaihtuvan mittainen	Sisältää viestin datan.

- **Binding Update (BU)**. MIPv6-laitteen lähettämä viesti toiselle laitteelle ilmoittaessaan uudesta vierasosoitteestaan. BU-viestiä voidaan käyttää seuraavissa tapauksissa:

  Päivittää kotiagentti uudella vierasosoitteella. Tätä kutsutaan kotirekisteröinniksi. Kotiagentti käyttää kotiosoitetta *Kotiosoite* -optiossa ja vierasosoitetta *Vaihteleva vierasosoite* -liikkuvuusoptiossa päivittäessään liikkuvan päätelaitteen koti- ja vierasosoitetta sidosvälimuistiinsa.

  Päivittää MIPv6-tuen omaava liikennöintikumppani, jonka kanssa liikkuva päätelaite kommunikoi. Tätä kutsutaan liikennöintikumppanin rekisteröimiseksi. Liikennöintikumppani käyttää kotiosoitetta *Kotiosoite* -optiossa ja paketin lähdeosoitetta päivittäessään päätelaitteen koti- ja vierasosoitetta sidosvälimuistiinsa.

- **Binding Acknowledgement (BAck)**. Viestin lähettää kotiagentti tai liikennöintikumppani kuitatessaan liikkuvan päätelaitteen lähettämän BU-viestin. Kuittaus sisältää elinajan sidoksen pitämisestä HA:n tai CN:n sidosvälimuistissa. Kotiagentin kuittausviestissä elianika viittaa siihen kuinka kauan kotiagentti palvelee liikkuvaa päätelaitetta. Sidoksen voi päivittää HA:n tai CN:n lähettämällä BRR-viestillä. BAck-viesti sisältää myös ilmoituksen kuinka usein MN:n tulisi lähettää BU-viestejä.

- **Binding Error (BE)**. Viestin lähettää liikennöintikumppani, jos se raportoi virheistä sidospäivityksessä
- **Home Test Init (HoTI)**. Viestin lähettää liikkuva päätelaite suorittaessaan *Return Routability* -prosessia. Tällöin se tutkii epäsuoran polun liikennöintikumppanille kotiagentin kautta.
- **Care-of Test Init (CoTI)**. Viestin lähettää liikkuva päätelaite suorittaessaan *Return Routability* -prosessia. Tällöin se tutkii suoran polun liikennöintikumppanille.
- **Home Test (HoT)**. Viestin lähettää liikennöintikumppani *Return Routability* -prosessin aikana vastatessaan HoTI-viestiin.
- **Care-of Test (CoT)**. Viestin lähettää liikennöintikumppani *Return Routability* -prosessin aikana vastatessaan CoTI-viestiin.

Type 2 Routing Header

Mobile IPv6 määrittelee uuden reititysotsikon, *Type 2 Routing Header*, jonka avulla Mobile IPv6- tuen omaava liikennöintikumppani voi lähettää paketteja suoraan kotiverkostaan poissa olevalle liikkuvalle päätelaitteelle sen vierasosoitteeseen. Tällöin liikennöintikumppanit asettavat IPv6-otsikon kohdeosoitteeksi liikkuvan päätelaitteen sen hetkisen vierasosoitteen ja reititysotsikkoon sen kotiosoitteen. Paketin saapuessa liikkuvan päätelaitteen vierasosoitteeseen, päätelaite havaitsee reititysotsikosta kotiosoitteensa, jolloin tietää paketin kuuluvan itselleen. [32]

Tyypin 2 reititysotsikko eroaa IPv6-spesifikaation vastaavasta (Type 0 Routing Header) [25] siinä, että siihen voi tallettaa vain yhden osoitteen Mobile IPv6 käyttöön. Tyypin 0 reititysotsikko voi tallentaa monta osoitetta erilaisiin reititys tarpeisiin. Erilaisen reititystyypin ansiosta esimerkiksi palomuurit voivat käsitellä lähdereititettyjä pakettaja erilailla kuin liikennöintikumppanilta suoraan liikkuvalle päätelaitteelle lähetettyjä paketteja. [32]

2.2.4 Tietorakenteet

Mobile IPv6 -protokollassa tietorakenteita käytetään protokollan toimintaan liittyvien tilatietojen varastoimiseen. Näiden tilatietojen avulla hallitaan viestien lähettämiseen, sijainnin päivittämiseen ja kotiagenttien hallintaan liittyviä toimenpiteitä.

Seuraavat tietorakenteet on määritelty MIPv6-protokollaan: sidosvälimuisti, sidospäivitys lista, kotiagenttien lista. [32]

Sidosvälimuisti (Binding cache)

Sidosvälimuisti sijaitsee jokaisessa kotiagentissa ja MIPv6:tta tukevassa liikennöintikumppanissa sisältäen tiedon sen hetkisistä sidoksista liikkuvan päätelaitteen koti- ja vierasosoitteiden välillä. Sidosvälimuisti pitää sisällään seuraavanlaista tietoa [32]:

- Liikkuvan päätelaitteen koti- ja vierasosoite,
- muistissa olevan sidoksen elinaika,
- lippu, joka ilmaisee kotirekisteröimisestä,
- aika jolloin edellinen sidospyyntö on lähetetty.

Laitteen vastaanottaessa BU-viestin tarkastetaan löytyykö lähettäjän osoite jo sidosvälimuistista. Jos löytyy, niin päivitetään sidoksen aikaleima ja järjestetään lista ajan mukaiseen järjestykseen. Jos osoitetta ei löydy, niin uusi sidos lisätään listaan.

Kun laite, joko HA tai CN, lähettää paketteja liikkuvalle päätelaitteelle, tarkistaa se sidosvälimuistista löytyykö paketin kohdeosoitetta vastaava sidos. Jos sidos löytyy, niin paketit lähetetään kohdeosoitteen sijaan sidosvälimuistiin merkityn vierasosoitteen mukaiseen osoitteeseen.

Kotiagentti käyttää tätä välimuistia liikkuvan päätelaitteen kotisoitteeseen lähetettyjen pakettien ohjaamisen liikkuvan päätelaitteen sen hetkiseen vierasosoitteeseen. MIPv6:tta tukeva liikennöintikumppani käyttää välimuistia lähettääkseen paketit suoraan liikkuvalle päätelaitteelle.

Sidospäivitys lista (Binding Update List)

Sidospäivitys lista sijaitsee liikkuvalla päätelaitteella ja pitää sisällään viimeisimmät kotiagentille ja liikennöintikumppaneille lähetetyt sidospäivitykset. Sidospäivitys lista sisältää [32]:

- MN:n osoite, josta BU lähetettiin,
- kotiosoite, josta BU lähetettiin,
- lippu, joka ilmaisee kotirekisteröimisestä,

- edellisessä BU:ssa lähetetty CoA,
- BU:n elinaika-kentässä oleva arvo,
- sidoksen jäljellä oleva elinaika,
- Sekvenssi numero -kentän maksimiarvo edellisestä BU:sta,
- aikaleima edellisestä BU-viestin lähetyksestä,
- ilmoitus BU:n uudelleenlähettämisen tarpeesta,
- lippu joka ilmoittaa, ettei BU-viestejä tarvitse enää lähettää.

Liikkuvan päätelaitteen on verkossa liikuttaessa ilmoitettava uusista sijainneista BU-viestejä kotiagentille ja liikennöintikumppaneille, jolloin ne osaavat päivittää sidosvälimuistiaan ja ohjata paketit liikkuvan päätelaitteen uusiin osoitteisiin. Jotta liikkuva päätelaite tietää missä kotiagentti ja liikennöintikumppani sijaitsevat, tallettaa se näiden osoitteet sidospäivitys listaan. Listaan on talletettu tieto myös aiemmin lähetetyistä BU-viesteistä ja listassa olevien ajastimien avulla voidaan pyytää sidosten päivitystä niiden elinajan erääntyessä.

Kotiagenttien lista (Home Agents List)

Kotiagentit ylläpitävät kotiagenttien listaa tallentamalla siihen tietoa muista samalla verkkoalueella olevista kotiagenteista. Kotiagentti havaitaan RA-viestistä, jossa *Home Agent (H)* -bitti on asetettu 1. Kotiagentit voivat lähettää kotiagenttien listan liikkuvalle päätelaitteelle, joka on vierasverkossa ja lähettää kotiagentin etsintäviestin. Kotiagenttien lista sisältää seuraavia tietoja [32]:

- Verkossa olevan reitittimen link-local osoite, joka saadaan *Router Advertisement* -viestin lähdeosoitteesta,
- kotiagentin globaali osoite, joka saadaan RA-viestin *Prefix*-kentästä *Prefix Information* -optiossa *Router Address (R)* -lipun ollessa 1,
- jäljellä oleva elinaika, joka saadaan joko *Home Agent Lifetime* -kentästä *Home Agent information* -optiossa tai *Home Agent Lifetime* -kentästä RA-viestissä. Kun elinaika erääntyy, merkintä poistetaan kotiagenttien listalta,
- kotiagenttien prioriteettijärjestys, joka saadaan *Home Agent Preference* -kentästä *Home Agent Information* -optiosta. Jos RA-viesti ei sisällä *Home Agent information* -optiota preference-arvoksi asetetaan 0, joka on määritelty medium-tason

prioriteetiksi. Liikkuva päätelaite käyttää preference-arvoa valitessaan kotiagenttia. Kotiagentti järjestää kotiagenttien listan käyttäen preference-arvoa. Kun liikkuva päätelaite vastaanottaa listan kotiagentin etsinnän aikana, se valitsee ensimmäisen kotiagentin listasta.

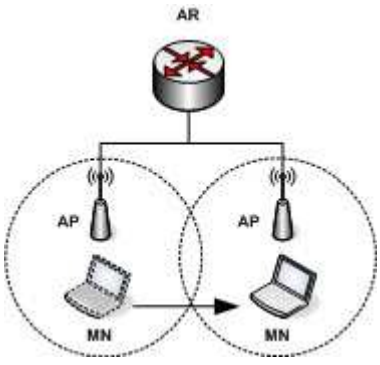
Liikkuva päätelaite käyttää listaa siis uuden kotiagentin etsintään esimerkiksi vikatilanteissa entisen kotiagentin lakatessa vastaamasta viesteihin. Tällöin liikkuva päätelaite lähettää verkkoon kotiagentin etsintäviestin, johon yksi toiminnassa olevista kotiagenteista vastaa lähettämällä sille kotiagenttilistan.

2.2.5 Yhteysvastuun vaihto

Yhteysvastuun vaihto, eli *handover* tai *handoff*, on tilanne, jossa reititin luovuttaa päätelaitteen yhteyden toiselle reitittimelle. Handover voi tapahtua eri tavoin riippuen millä OSI-kerroksen tasolla liikutaan. Kun handover vaikuttaa vain siirtoyhteyskerroksella, kutsutaan sitä tason 2 handoveriksi (L2 handover). Jos päätelaite liikkuu samalla toisen verkon alueelle ja saa uuden osoitteen, kutsutaan sitä tason 3 handoveriksi (L3 handover).

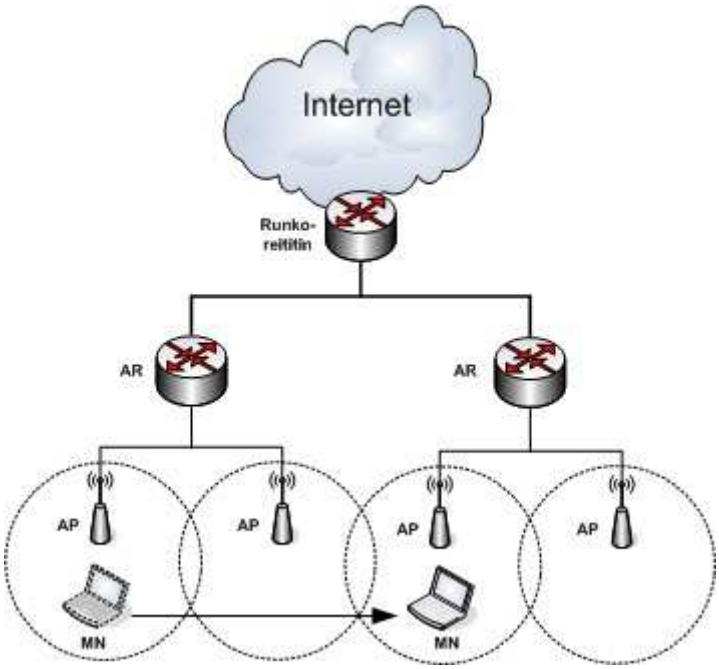
Yhteysvastuun vaihto voidaan jaotella myös *horisontaaliseen* ja *vertikaaliseen* handovereihin. Horisontaalista handoveria tapahtuu, kun liikkuva päätelaite vaihtaa verkkoa samantyyppisten tiedonsiirtorajapintojen välillä (esim. WLAN -> WLAN). Vertikaaliseksi handoveriksi taas kutsutaan tapahtumaa, jossa päätelaite vaihtaa tiedonsiirtorajapinnan erilaisten liityntöjen välillä (esim. WLAN -> GPRS). [37]

Kuva 2.13 esittää handoveria joka tapahtuu saman verkon alueella eri tukiasemien välillä. Tässä kyseessä on siten L2 handover, joka tapahtuu vain siirtoyhteyskerroksella. Kuva 2.14 esittää handoveria joka tapahtuu eri verkkojen alueella



Kuva 2.13: Handover tukiasemien välillä.

ja täten eri tukiasemien välillä. Tässä kyseessä on siten tason 2 ja 3 handover, joka tapahtuu sekä siirtoyhteys- että verkkokerroksella.



Kuva 2.14: Handover eri verkkojen välillä.

2.2.6 Mobile IPv6 yhteysvastuun vaihto

Kun liikkuva päätelaite vaihtaa liityntää IPv6-verkosta toiseen, suorittaa se MIPv6 yhteysvastuun vaihdon. Tämän prosessin aikana liikkuvan päätelaitteen on havaittava siirtyminen uuteen verkkoon, konfiguroitava siellä uusi vierasosoite ja informoitava kotiagenttia ja mahdollista liikennöintikumppania uudesta sijainnistaan.

Liikkeen havaitseminen

Liikkuvan päätelaitteen siirtyessään vieraaseen verkkoon on sen vaihdettava vierasosoitteensa uuden verkon mukaiseksi. Ensin päätelaitteen on havaittava verkon vaihtuminen ja tähän Mobile IPv6 -protokollassa voidaan käyttää apuna RA-viestejä sekä NUD-menetelmää. [26]

Liikkuva päätelaite huomaa siis yhteysvastuun vaihdon tapahtuneen tarkastellessaan liityntäreitittimien lähettämiä RA-viestejä (*Router Advertisement*). [26] Liityntäreitittimissä on erityinen palvelu lähettämässä tietyin väliajoin näitä viestejä. Alustasta riippuen se voi esimerkiksi RADVD (Linux) tai RTADVD (*BSD).

Liikkuva päätelaite huomaa liityntäreitittimen vaihtuneen RA-viestin sisältämästä reitittimen osoitteen prefixistä tai jos edellisestä liityntäreitittimestä RA-viestiä ei saavukaan tietyn ajan kuluessa, joka on merkitty RA-viestin *Advertisement Interval*

-optioon. [32] Jälkimmäisessä tapauksessa liikkuva päätelaite lähettää verkkoon RS-viestin (*Router Solicitation*) pyytääkseen reitittimeltä RA-viestiä.

NUD-menetelmää käytettäessä liikkuva päätelaite käyttää apuna naapurin havaitsemisen NS-viestiä, jolla voi tarkastaa verkon laitteen saavutettavuus. Päätelaitteet lähettävät tätä viestiä verkkoon tietyin väliajoin tai menetettäessä yhteyden naapuriin. Jos yhteys oletusreitittimeen kadotetaan, lähettää liikkuva päätelaite sille NS-viestin. Jos reititin ei vastaa kyselyyn, liikkuva päätelaite käynnistää *Router Discovery* -prosessi uuden oletusreitittimen löytämiseksi verkosta. RD-prosessin suoritettuaan onnistuneesti liikkuva päätelaite havaitsee uuden oletusreitittimen ja havaitsee siitä liikkuneensa uuteen verkkoon.

Osoitteen muodostaminen

Kun yhteysvastuun vaihto on suoritettu onnistuneesti, niin seuraavaksi liikkuvan päätelaitteen on saatava uusi vieras- eli CoA-osoite (*Care-of Address*). Liikkuvan päätelaitteen on ensin tarkistettava link-local -osoitteensa ainutlaatuisuus uudessa verkossa. Liikkuva päätelaite suorittaa DAD:n (*Duplicate Address Detection*) link-local -osoitteelleen. Sen jälkeen se voi muodostaa uuden CoA-osoitteen joko tilattomalla tai tilallisella autokonfiguraatiolla.[28][29] Myös tälle uudelle CoA-osoitteelle suoritetaan DAD sen ainutlaatuisuuden tarkistamiseksi.

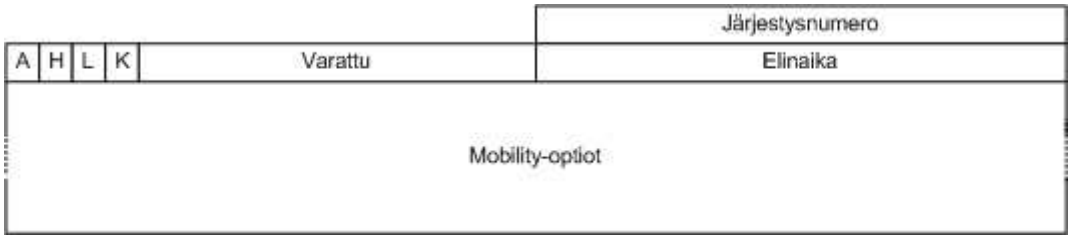
DAD:n suorittaminen aiheuttaa viiveitä, mutta ainutlaatuisuuden tarkistaminen on suoritettava uudessa verkossa, koska IP-osoite on voitu luoda sattumanvaraisesti yksityisyyssyistä tai laitteen MAC-osoitetta ei ole saatu. Laitteelta ei välttämättä löydy komponenttia, josta yksilöllisen MAC-osoitteen voisi luoda tai se on muokattavissa. Laitteen verkkokortti ei ole IEEE:n hyväksymä eikä tällöin MAC-osoitetta ole rekisteröity tai verkkokorttiin on saattanut tulla valmistusvaiheessa virhe.

Kun uuden CoA-osoitteen muodostaminen on suoritettu onnistuneesti, liikkuvan päätelaitteen on ilmoitettava siitä myös kotiagentilleen ja mahdolliselle liikennöintikumppanille. Liikkuva päätelaite lähettää BU-viestin (*Binding Update*) ilmoittaessaan uudesta koti- ja vierasosoitteen sidoksesta johon kotiagentti ja mahdollisesti liikennöintikumppani vastaavat BAck-viestillä (*Binding Acknowledgement*).

2.2.7 Rekisteröinnit

Kotirekisteröinti

Kun liikkuvalle päätelaitteelle on muodostettu vierasosoite ja todettu se ainutlaatuiseksi, voidaan se ottaa käyttöön. Liikkuvan päätelaitteen on suoritettava kotirekisteröinti, jotta kotiagentti osaa ohjata liikkuvalle päätelaitteelle kuuluvat paketit sen vierasosoitteeseen. Rekisteröinti suoritetaan lähettämällä vierasosoite kotiagentille, jotta sidos koti- ja vierasosoitteen välillä voidaan päivittää. Osoitteen ilmoittaminen tapahtuu *Binding Update* -viestin avulla. Kuvassa 2.15 BU-viestin rakenne. [32]



Kuva 2.15: Binding Update -viestin rakenne.

Binding Update -viesti lähetetään Mobility Header -laajennusotsikossa. Taulukossa 2.5 on kuvattu BU-viestin rakennetta:

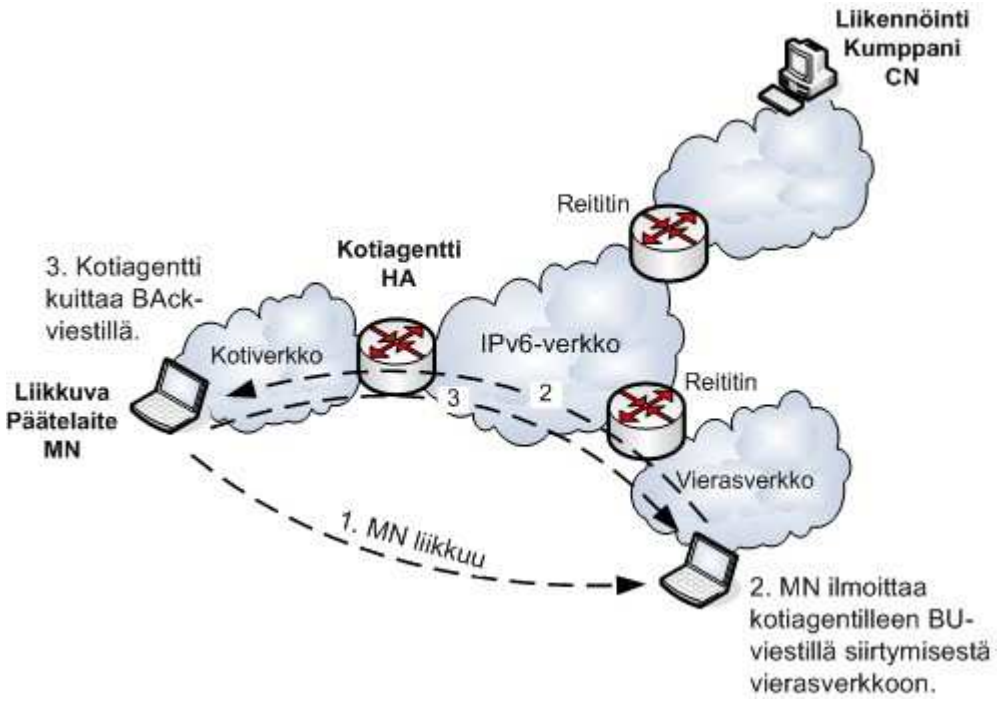
Kotiagentille lähetettävässä BU-viestissä on oltava päällä A- ja H-bitit, sekä tarvittaessa L-bitti. Lisäksi viestissä on oltava *Destination Options* -tyyppinen laajennusotsikko, joka sisältää liikkuvan päätelaitteen kotiosoitteen. Jos käytetään IPsec-todennusta, kotiagentin vastaanottaessa BU-viestin, todetaan se ensin IPsec-tekniikan avulla validiksi. Viestin ollessa validi, sidosvälimuistiin lisätään uusi rivi viestistä saaduilla tiedoilla.

Kun kotiagentti on päivittänyt sidoksen sidosvälimuistiin, se lähettää kaikille saman aliverkon päätelaitteille NA-viestin (Neighbor Advertisement), jonka avulla muodostetaan sidos liikkuvan päätelaitteen kotiosoitteen ja kotiagentin linkkitason osoitteen välille. Tällöin liikkuvalle päätelaitteelle lähetetyt paketit menevät kotiagentille.

Liikkuvalle päätelaitteelle kulkevista paketeista kotiagentti käsittelee itse hallintaviestit ja välittää muut paketit liikkuvalle päätelaitteelle sidoksen mukaiseen vierasosoitteeseen. Kotiagentti voi vastata hallintaviesteihin esimerkiksi estäessään muita laitteita konfiguroimasta itselleen samaa osoitetta. Näiden toimenpiteiden jälkeen kotiagentti lähettää liikkuvalle päätelaitteelle kuittauksen Binding Acknow-

Taulukko 2.5: Binding Update -laajennusotsikon kentät.

Kenttä	**Pituus (bittiä)**	**Kuvaus**
Järjestysnumero	16	Tämän avulla vastaanottaja tietää, missä järjestyksessä BU-viestit ovat saapuneet ja lähettäjä, mihin viestiin on saapunut kuittaus.
Acknowledge (A)	1	Tällä ilmaistaan, halutaanko BU-viestiin kuittaus.
Home Registration (H)	1	Tämä bitti on oltava päällä kaikissa kotiagentille menevissä BU-viesteissä. Tämän avulla MN voi myös pyytää reititintä kotiagentikseen.
Link-Local Address Compatibility (L)	1	Asetetaan päälle, jos MN:n koti- ja link-local -osoitteen loppuosat ovat samat.
Key Management Mobility Capability (K)	1	Jos bitti ei ole päällä, IPsec-tietoturvaominaisuudet eivät säily liikkumisen jälkeen. Kun BU-viesti ei ole tarkoitettu kotiagentille, bitti on asetettava nollaksi.
Varattu	12	Nämä bitit on varattu ja ne on asetettava nolliksi.
Elinaika	16	Ilmaisee, kauanko viestin sidos on voimassa. Yksi aikayksikkö vastaa neljää sekuntia. Sidos on poistettava sidosvälimuistista, jos sitä vastaava kentän arvo on 0.
Mobility-optiot	Vaihtuvan mittainen	Tässä kentässä on BU-viestiin liittyvää lisätietoa. Kentän avulla Mobility Header -laajennusotsikon pituudesta saadaan 8 tavun kerrannainen.



Kuva 2.16: Kotirekisteröinti.

ledgement -viestillä. Kuvassa 2.16 on havainnollistettu liikkuvan päätelaitteen kotirekisteröitymistä.

Kotirekisteröintiprosessissa menetellään seuraavasti:

1. Liikkuva päätelaite liikkuu kotiverkosta vierasverkkoon. Vierasverkossa liikkuva päätelaite ottaa yhteyden uuteen liityntäreitittimeen ja muodostaa itselleen vierasosoitteen.

2. Liikkuva päätelaite ilmoittaa BU-viestin välityksellä kotiagentilleen siirtymisestä vierasverkkoon.

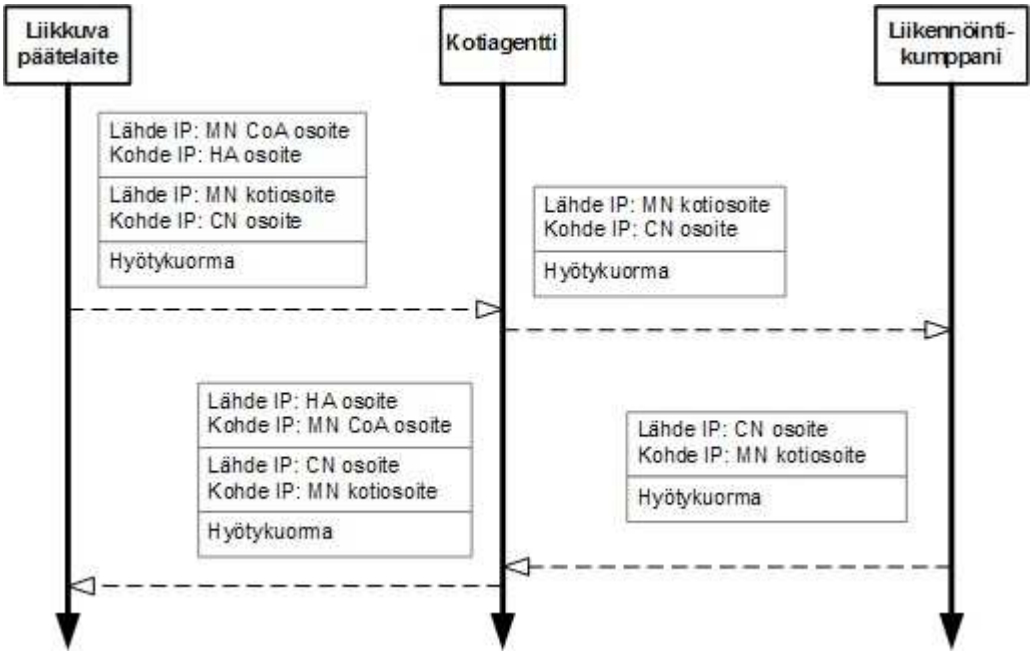
3. Kotiagentti vastaanottaa viestin ja kuittaa sen BAck-viestillä.

Kun kotirekisteröinti on suoritettu, voidaan alkaa kommunikoida mahdollisten liikennöintikumppanien kanssa.

Liikennöintikumppanin rekisteröinti

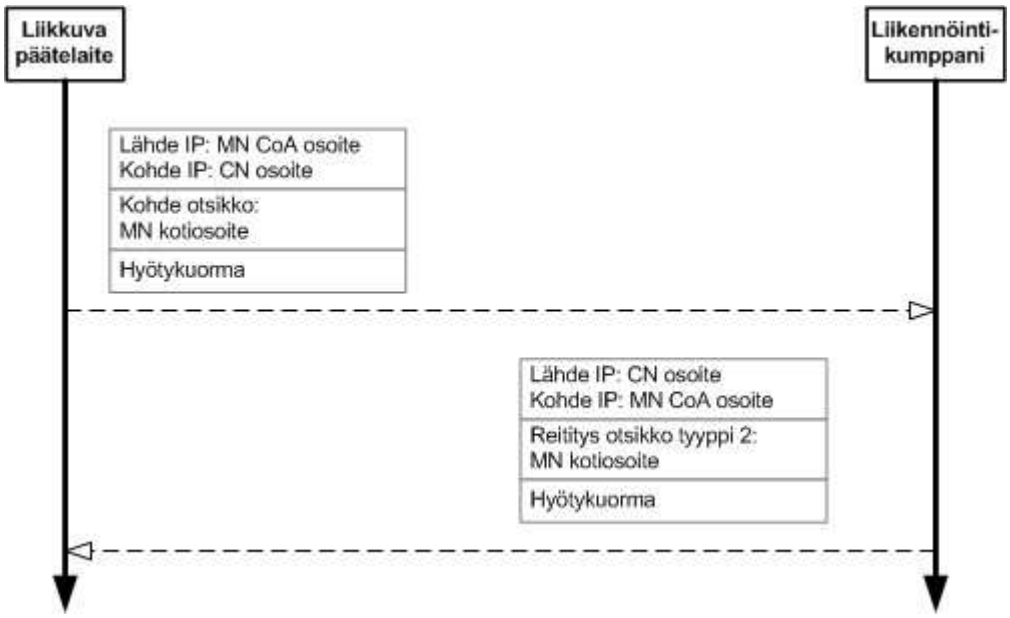
Liikkuvan päätelaitteen ja liikennöintikumppanin väliseen kommunikointiin on Mobile IPv6 -protokollassa olemassa kaksi tapaa:

Epäsuorasti. Jos liikennöintikumppani ei ole MIPv6-tuen omaava tai liikkuvan päätelaitteen rekisteröintiä ei ole vielä suoritettu reititetään näiden väliset paketit kotiagentin kautta. Kotiagentin ja liikkuvan päätelaitteen välinen liikenne tunneloidaan. Liikkuva päätelaite kapseloi IPv6-paketin CoA-osoitteellaan ja kotiagentin osoitteella, alkuperäisen otsikon sisältäen liikkuvan päätelaitteen kotiosoitteen sekä liikennöintikumppanin osoitteen. Liikennöintikumppanin lähettäessä paketteja liikkuvalle päätelaitteelle käyttää se liikkuvan päätelaitteen kotiosoitetta. Kotiagentti kapsuloi sidosmuistissaan olevan sidoksen mukaisesti liikkuvan päätelaiteen vierasosoitteeseen. Tätä menetelmää kutsutaan *kaksisuuntaiseksi tunneloinniksi* (engl. Bidirectional Tunneling). Kuva 2.17. [32]



Kuva 2.17: Kaksisuuntainen tunnelointi.

Suoraan. Jos liikennöintikumppani on MIPv6-tuen omaava, voi liikkuva päätelaite kommunikoida sen kanssa suoraan ilman pakettien reitittämistä kotiagentin kautta. Liikkuva päätelaite lähettää paketteja suoraan liikennöintikumppanille käyttäen liikkuvan päätelaitteen IP-osoitetta ja sisällyttäen IP-osoitteensa *Home Address Destination* -optioon. Liikennöintikumppani lähettää paketteja liikkuvan päätelaitteen CoA-osoitteeseen sisällyttäen tyypin 2 reititysotsikkoon liikkuvan päätelaitteen kotiosoitteen. Tätä suoraa menetelmää kutsutaan *reitin optimoimiseksi* (engl. Route Optimization). Kuva 2.18. [32]



Kuva 2.18: Reitin optimointi.

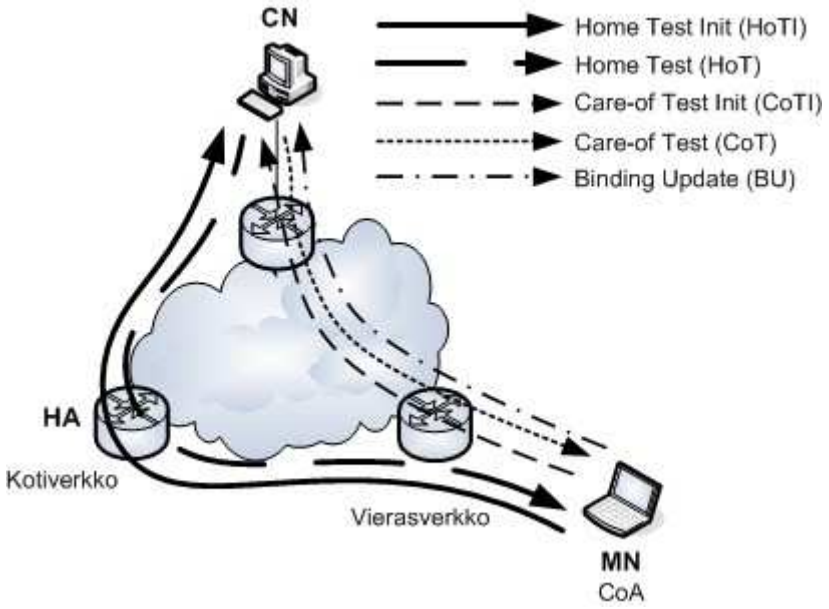
Reitin optimointi (Route optimization)

Jos sekä liikkuva päätelaite, että liikennöintikumppani tukevat MIPv6-protokollaa, voidaan kommunikointi suorittaa näiden välillä optimaalisesti. Reitin optimointi tarkoittaa, että liikkuva päätelaite ja liikennöintikumppani voivat siirtää dataa suoraan toistensa välillä, ilman että paketteja reititetään kotiagentin kautta. Optimoinnissa MIPv6-protokollaa tukeva liikennöintikumppani rekisteröi liikkuvan päätelaitteen vierasosoitteen. Ennen reitin optimointia on suoritettava kaksi menettelyä toiminnan varmistamiseksi. Ensimmäiseksi suoritetaan *Return Routability* -testi, jolla varmistetaan, että liikkuva päätelaite on todella saavutettavissa vieras- ja kotiosoitteestaan. Toiseksi suoritetaan varsinainen vierasosoitteen rekisteröinti BU-viestillä liikennöintikumppanille.

Return Routability

Menettely käynnistetään kun liikkuva päätelaite siirtyy toiseen verkkoon ja rekisteröi vierasosoitteen kotiagentilleen. Menettely voidaan myös aloittaa kun liikkuva päätelaite saa liikennöintikumppanin lähettämän tunneloidun paketin kotiagentilta. Tunneloinnin perusteella liikkuva päätelaite havaitsee, että paketti on tullut kotiagentin kautta, eikä reitti tällöin ole liikennöintikumppanille optimaalinen. [32]

Kuvassa 2.19 on esitetty Return Routability -menettelyä viesteineen.



Kuva 2.19: Return Routability -menettely

Reitin optimointi aloitetaan kun liikkuva päätelaite suorittaa Return Routability -menettelyn lähettämällä liikennöintikumppaneille samanaikaisesti kaksi viestiä: *Home Test Init (HoTI) ja Care-of Test Init (CoTI)*. HoTI-viesti reititetään kotiagentin kautta ja CoTI-viesti lähetetään liikennöintikumppanille suoraan.

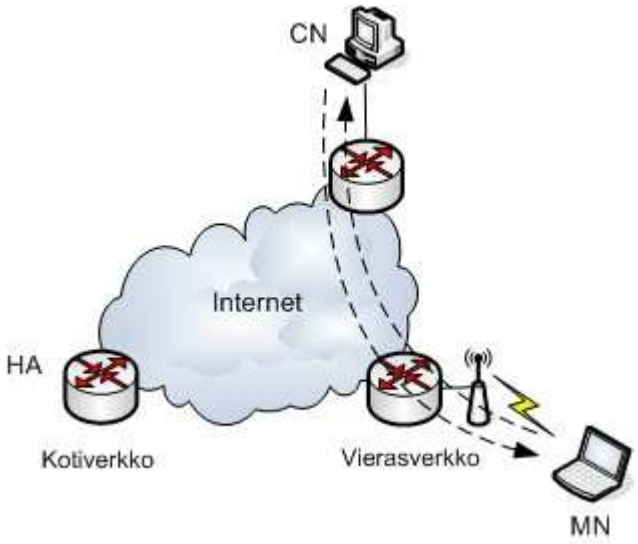
Sekä HoTI- että CoTI-viesteillä pyydetään liikennöintikumppanilta valtuusavainta (token). Liikkuvan päätelaitteen ja kotiagentin välillä viesti suojataan IPsec-menetelmän avulla, jos kotiagentti tukee sitä tai menetelmä on kytketty päälle.

HoTI- ja CoTI -viestit sisältävät satunnaisesti määritetyn 64-bittisen tunnisteen (engl. nonce). Tunnisteet myös indeksoidaan (engl. nonce index), joka mahdollistaa oikean vastausviestin yhdistämisen tiettyyn HoTI- tai CoTI-viestiin. Vastausviesteinä käytetään HoT (Home Test) ja CoT (Care-of Test) -viestejä. Vastaukset lähetetään samaa reittiä kuin valtuusavainten pyyntöviestitkin.

Prosessi on suoritettu onnistuneesti, kun liikkuva päätelaite on vastaanottanut HoT- ja CoT-viestit. Liikkuva päätelaite määrittää saamistaan valtuusavaimista yhden sidosten hallinta-avaimen (engl. Binding Management Key). Tämän jälkeen liikkuva päätelaite voi lähettää BU-viestin liikennöintikumppanille.

Return Routability -menettelyn jälkeen liikkuva päätelaite lähettää BU-viestin mukana sidoksen hallinta-avaimen ja liikennöintikumppanin lähettämän indeksi. Liikennöintikumppani puolestaan laskee saamastaan BU-viestistä aikaindeksin avulla

arvon sidoksen hallinta-avaimelle ja vertaa laskemaansa avainta viestin sisältämään avaimeen. Rekisteröinti hyväksytään, jos avaimet ovat samat ja tämän jälkeen sidos lisätään liikennöintikumppanin sidosvälimuistiin. Kuvassa 2.20 reitin optimoinnin jälkeinen tilanne, jossa paketit reititetään suorinta reittiä liikennöintikumppanin ja liikkuvan päätelaitteen välillä.



Kuva 2.20: Reitin optimointi

Palaaminen kotiverkkoon

Kun liikkuva päätelaite palaa kotiverkkoonsa, on sen informoitava kotiagenttia, ettei sen tarvitse enää käsitellä liikkuvalle päätelaitteelle kuuluvia paketteja. Tässä rekisteröinnissä lähetetään BU-viesti A- ja H-bitti päällä, elinaika arvolla 0 ja vierasosoite sidottuna kotiosoitteeseen. Lähdeosoitteena on kotiosoite.

Kotiverkkoon tullessaan liikkuvan päätelaitteen on myös opittava kotiagentin MAC-osoite. Tähän käytetään NS-viestiä jonka *Target Address* -kenttään on asetettu kotiagentin globaali IP-osoite. Kohdeosoitteeksi puolestaan asetetaan kotiagentin *solicited node multicast* -osoite [35] ja lähdeosoiteeksi määrittelemätön osoite (::). Koska viesti on samanmuotoinen kuin DAD-prosessissa kotiagentti vastaa tähän NA-viestillä *all-nodes multicast* -osoitteeseen [35] sisällyttäen viestiin oman uniikin ja linkkitason osoitteen.

Tämän jälkeen liikkuva päätelaite lähettää BU-viestin jossa lähdeosoitteeksi on asetettu sen kotiosoite ja kohdeosoitteeksi kotiagentin osoite. Tämä tehdään ilman DAD:n suorittamista kotiosoitteelle. Tämän jälkeen kotiagentti lähettää BAck-viestin

ja lopettaa liikkuvan päätelaitteen kotiosoitteen ylläpidon.

Ennen BAck-viestin lähettämistä saattaa tulla tilanne, että kotiagentti lähettää NS-viestin liikkuvalle päätelaitteelle, johon se vastaa NA-viestillä. Tällä kotiagentti tarkistaa päätelaitteen saavutettavuuden. Liikkuva päätelaite ei kuitenkaan suorita DAD-prosessia.

Kun liikkuva päätelaite on vastaanottanut BAck-viestin, se lähettää NA-vietin *all-nodes multicast* -osoitteeseen O-lippu asetettuna. Tämä viesti informoi kaikkia laitteita lähettämään paketit suoraan liikkuvalle päätelaitteelle. Tämän jälkeen kaikki liikenne reititetään suoraan liikkuvalle päätelaitteelle, kuten normaalissa IPv6-verkossa. [32][38]

2.2.8 MIPv6:n tietoturva

IPv6-protokollan käyttöön on vahvasti liitetty tietoturvaominaisuuksien tuki IPsec-protokollaa käyttäen [25]. Tästä kerrottiin jo IPv6-osan kappaleessa 2.1.8. Mobile IPv6:ssa näitä palveluita käytetään rekisteröintiviestien turvaamiseen liikkuvan päätelaitteen ja kotiagentin välillä. Mobile IPv6:n tietoturva on määritelty RFC-dokumentissa 3775 [33].

Kuten kappaleessa 2.1.8 kerrottiin, liikkuvalle päätelaitteelle ja kotiagentille on määriteltävä sopimukset tietoturvaryhmän mukaan (engl. *Security Association (SA)*), jonka avulla osaavat kummatkin osapuolet luoda välilleen turvatun tiedonsiirtokanavan. SA sisältää tietoa kummankin laitteen kotiosoitteista, salausalgoritmeista ja -avaimista ja muista turvaparametreista. SA:n lisäksi liikkuvalle päätelaitteelle ja kotiagentille on määriteltävä myös tietoturvapolitiikka (engl. *Security Policy (SP)*). Liikkuvan päätelaitteen mahdollisuudesta siirtyä vierasverkkoihin ja saada vierasosoitteita aiheuttaa sen, ettei IPv6:n IPsec-toiminnallisuutta voida käyttää suoraan. Mobile IPv6:n on määritelty IPsec:n laajennus, jotta liikkuvan päätelaitteen mahdolliset uudet vierasosoitteet osataan päivittää IPsec:n vaatimaan tietoturvapolitiikan tietokantaan. [32][33]

RFC:ssä 3775 on määritelty IPsecin käyttö MIPv6:ssa [32]. Dokumentissa on ehdotettu IPsec-suojauksessa käytetyt protokollat ja moodit eri rekisteröintiviesteille ja hyötykuormalle taulukon 2.6 mukaan. IPsec-protokollaksi on määritelty ESP kaikille viesteille. BU/BA (Binding Update/Binding Acknowledgement) ja MPS/MPA (Mobile Prefix Solicitation/Mobile Prefix Advertisement) on määritelty kuljetusmoodilla toteutettavaksi. HoTI/HoT (Home Test Init/Home test) ja hyötykuorma on puolestaan määritelty tunnelimoodilla toteutettavaksi. Dokumentissa on myös ehdotettu IPsec-toteutuksiin pakolliseksi BU/BA-viestin salauksen, suositeltavaksi

MPS/MPA ja HoTI/HoT-viestien salauksen ja valinnaiseksi hyötykuorman salausmahdollisuuden.

Taulukko 2.6: IPsec-suojaus MIPv6:ssa.

Tyyppi	**IPsec protokolla/moodi**	**Ehdotus**
BU/BA	ESP/Kuljetus	Pakollinen
MPS/MPA	ESP/Kuljetus	Suositeltu
HoTI/HoT	ESP/Tunneli	Suositeltu
Hyötykuorma	ESP/Tunneli	Valinnainen

Myös automaattista avainten vaihtoa (*Internet Key Exchange*, IKE) [22] ollaan kehittelemässä MIPv6-protokollassa toimivaksi [33]. Kuitenkin se on vielä sen verran kehitysvaiheessa, ettei sitä ole vielä toteutettu tämän hetken MIPv6-toteutuksiin, joissa IPsec ESP-salauksella on jo varsin hyvällä mallilla.

IPsec-tuki vaihtelee tällä hetkellä saatavilla olevissa MIPv6-toteutuksissa. MIPv6-toteutusten tuesta ja asennuksesta saa parhaan käsityksen ottamalla selvää esimerkiksi seuraavassa kappaleessa 2.3 esitetyistä järjestelmistä.

2.3 IPv6 ja MIPv6 tuki

IPv6-tekniikkaa on jo kehitetty kymmenisen vuotta ja se löytyy tällä hetkellä jo melkein kaikista käytetyimmistä käyttöjärjestelmistä. IPv6-toiminnallisuus on näissä joko ytimeen käännettynä ja aktivoituna tai mahdollisuutena kääntää ja aktivoida se jälkikäteen tarpeen tullen.

Linuxissa IPv6-tuki on käännetty jo uusimpiin 2.4 ja 2.6 sarjan ytimiin ja on myös oletuksena päällä yleisimmissä jakeluversioissa. Samoin BSD-käyttöjärjestelmissä (FreeBSD, NetBSD ja OpenBSD) IPv6-tuki on käännetty ytimeen ja on aktivoitavissa asetustiedostoja käyttäen.

Tässä tutkielmassa käytettiin myös Mobile IPv6-tuen tarjoavaa IPv6-toteutuksia. Linux- ja BSD-järjestelmiin tarjolla olevat MIPv6-toteutukset ovat olleet jo jonkin aikaa saatavilla suhteellisen toimivina kehitysversioina. Tässä tutkimuksessa keskityttiin Linux-, BSD- ja Cisco IOS-käyttöjärjestelmien (M)IPv6-toteutuksiin, koska Linuxille ja BSD:lle on tarjolla tällä hetkellä kehittyneimmät MIPv6-toteutukset ja paljon käytettyihin Cisco-reitittimiin on uusimpiin IOS-versioihin myös saatavilla kotiagentin MIPv6-toiminnallisuus.

2.3.1 Linux

Linuxille on jo pitkään ollut saatavilla IPv6-tuki. Tuki lisättiin ytimeen (versio 2.1.8) vuoden 1996 lopulla. Tämän jälkeen sitä kehitettiin vähemmän aktiivisesti, joten Linuxin IPv6-tuki ei pysynyt IETF:n IPv6-työryhmien luomien luonnosten ja standardien kehitystahdin mukana. Linuxin IPv6-toteutuksen kehnouden 90-luvun lopulla huomasi ainakin japanilainen USAGI-projekti, joka otti ohjat käsiin ja alkoi kehittää ajantasaista IPv6-pinoa Linuxille vuoden 2000 lopulla [55]. Siihen otettiin vaikutteita japanilaisen KAME-projektin toteutuksesta, jossa on kehitetty IPv6/IPsec-pinoa BSD-käyttöjärjestelmille. Tämän jälkeen USAGI:n IPv6-pino kasvoi välillä liian suureksi kooltaan, jotta se olisi voitu liittää ytimen julkaisuversioon, joten jotain ominaisuuksia jouduttiinkin karsimaan julkaisuversiosta. Vielä nykyäänkin Linuxin jakeluversiot sisältävät USAGI-koodia IPv6-toteutuksissaan. [53]

Linuxille on kehitetty myös useita Mobile IPv6 -toteutuksia. Tällä hetkellä varmasti puhutuin ja kehittynein toteutus on MIPL (*Mobile IPv6 for Linux*), jota kehitetään Helsingin Teknillisessä korkeakoulussa. Versioon 1.1 asti kehitettiin GO-Core projektin alaisuudessa Helsingin Teknillisessä korkeakoulussa. Tämä versio oli implementoitu kerneliin 2.4.x Mobile IPv6 -määrittelyn mukaisesti (draft 24), mutta ei tukenut IPsec-protokollaa.

Versiosta 2.0 asti sitä on kehitetty yhteistyössä GO-Core:n ja USAGI/WIDE-projektien kanssa. Go-Core ja USAGI ovat yhteistyössä kehittäneet MIPv6-protokollaa tukevat laajennukset Linuxin IPv6-pinoon. Projektin tavoitteena on saada tulevaisuudessa toimiva MIPv6-tuki Linux-kernelin jakeluversioon. Tällä hetkellä MIPL on saatavilla versiossa 2.0.2 Linux-kernelille 2.6.16. Se tarjoaa täyden toiminnallisuuden MIPv6-ympäristöön (MN, HA ja CN) sekä tukee IPsec-protokollaa.

MIPL on ladattavissa ilmaiseksi projektin kotisivuilta [54]. Versio 2.0.2 sisältää lisäyksen Linux ytimeen 2.6.16 ja mip6d-sovelluksen käyttöympäristöön.

Myös edellä mainitulla USAGI-projektilla on kotisivuillaan tarjolla MIPv6-toteutus, joka kantaa nimeä UMIP. Itse asiassa se on MIPL 2.0 version (20051214) pohjalta USAGI:n tarpeiden mukaan muokattu toteutus. Kuten MIPL, se tarjoaa täyden toiminnallisuuden MIPv6-ympäristöön (MN, HA ja CN) sekä tukee IPsec-protokollaa. [56]

2.3.2 BSD

KAME IPv6 -protokollapino on IPv6-toteutus BSD-käyttöjärjestelmille. KAME-projekti aloitettiin vuonna 2000 yhteistyönä useiden japanilaisten yliopistojen ja yritysten kesken. Projektin tavoitteena on luoda ilmainen IPv6/IPsec-ohjelmistopaketti

yleisimmille BSD-käyttöjärjestelmille (FreeBSD, NetBSD ja OpenBSD). KAME IPv6 -pino tukee IPv6:en liittyviä spesifikaatioita RFC 2460, 2461, 2462, 2463. [57] BSD-käyttöjärjestelmien uusimmissa jakeluversioissa IPv6-toiminnallisuus on valmiiksi käännetty ytimeen, mutta se ei ole oletuksena päällä. IPv6-tuen voi asettaa päälle määrittelemällä se asetustiedostoon (*rc.conf*), josta se luetaan ja asetetaan päälle käynnistysvaiheessa.

KAME IPv6 -pinoon on toteutettu MIPv6-laajennus japanilaiselta SHISA-projektilta. SHISA syntyi vuonna 2004, kun WIDE-projektin alaisuudessa toimineet kaksi MIPv6-kehitykseen erikoistunutta projektia, KAME (KAMEMIP) ja InternetCAR (SFCMIP), yhdistettiin. SHISA on MIPv6/NEMO-protokollapino, joka sisältää laajennuksen kerneliin sekä ohjelmia käyttöympäristöön (engl. *userland*). SHISA tukee MN, HA ja CN sekä NEMO (*Network Mobility*) -toiminnallisuutta. Tällä hetkellä protokollapino tukee spesifikaatioita RFC 3775, 3776 ja 3963. [57][59]

KAME/SHISA-ohjelmistopaketin voi ladata ilmaiseksi KAME:n kotisivuilta [57]. Tätä pakettia ei tarjota julkaisuversiona (engl. *release*) kuten MIPL:ä, vaan kehitysversiona (engl. *snapshot*) josta julkaistaan uusi versio joka maanantai. KAME/SHISA-toeutus on saatavilla tällä hetkellä BSD-käyttöjärjestelmien versioille: FreeBSD 5.4, NetBSD 2.0 ja OpenBSD 3.6. Ohjelmistopaketti sisältää muokatun ytimen kyseiselle käyttöjärjestelmälle sekä MIPv6-toiminnallisuuteen liittyviä ohjelmia käyttöympäristöön. [57]

2.3.3 Cisco

Maailman suurimpiin reititinvalmistajiin kuuluva Cisco on luonnollisesti mukana IPv6:n kehitystyössä. Cisco toi IPv6-tuen sen omaan IOS-käyttöjärjestelmäänsä jo vuonna 2000. IPv6 on tuettu IOS-käyttöjärjestelmissä alkaen versiosta 12.0(22)S. [49]

Cisco aloitti myös Mobile IPv6 -protokolla kehittelyn yhdessä Lancasterin yliopiston MIPv6-projektin kanssa [40]. Cisco tukee MIPv6-toiminnallisuutta uusimmissa IOS-versioissaan kotiagentin osalta. MIPv6 on tuettu alkaen IOS-versiosta 12.3(14)T, 12.4, 12.4(2)T ja IPv6-pääsylistojen osalta versiosta 12.4(2)T. Ciscon kotiagentti on toiminnallisuudeltaan jo varsin vakaa, mutta siitä puuttuu tuki Mobile IPv6:n vaatimasta IPsec-toteutuksesta. [49]

2.3.4 Windows

Microsoft Windows tukee tällä hetkellä IPv6-protokollaa uusimmissa käyttöjärjestelmissään. Näitä ovat XP (SP1 lähtien), 2003 Server, CE ja Vista. Vistaa lukuun ottamatta Windowsin asennuksen mukana tulee IPv6-ohjelmisto, jonka voi kytkeä toi-

mintaan verkkoasetuksista. Tämä versio on perua Windowsin tutkimusryhmän tuotoksista, joka taas oli Lancasterin yliopiston projektin pohjalta toteutettu. Tämä aikaisempi projekti oli nimeltään *Lancaster and Microsoft Active Research Collaboration (LandMARC)*. Jälkeenpäin tämä IPv6-toteutus siirrettiin XP:n julkaisuversioon [51]. Windowsin Vistassa puolestaan IPv6 on jo käyttövalmiina IPv4-protokollan rinnalla sisältäen myös täyden tuen IPsec:lle. [50][48]

Windowsille on kehitelty myös Mobile IPv6-toteutuksia niin tutkimus, kuin kaupallisissa tarkoituksissa. Edellä mainittu XP:n sisältämä IPv6-toteutus käsittää vain CN-tuen ja tämänkin suojattuna IPsecin AH-protokollalla [51]. Tällöin se ei ole yhteensopiva olemassa olevien Linux- ja BSD-toteutusten kanssa. Windowsille on kehitelty myös kaupallisia toteutuksia, kuten Treck Incorporationin lähinnä sulautettuihin Windows-järjestelmiin suunnattu MIPv6-toteutus, joka sisältää tuen liikkuvalle päätelaitteelle ja liikennöintikumppanille. [52][48]

2.3.5 Muut valmistajat

Edellä mainittujen järjestelmien lisäksi maailmalla on olemassa paljon muitakin valmistajia, jotka ovat toteuttaneet IPv6 ja MIPv6-toiminnallisuutta järjestelmiinsä. Sivulla [48] on listattu tällä hetkellä olemassa olevia eri järjestelmien IPv6-toteutuksia.

IPv6-kehityksessä ollaan jo pitkällä, joten kaikista käytetyimmistä työasema- ja reititin-järjestelmistä löytyy tuki IPv6:lle. Edellä mainitun Windowsin ja Linuxin lisäksi Applen Macintosh-tietokoneissa on ollut IPv6-tuki OS X 10.2 Jaguar käyttöjärjestelmästä lähtien. Myös edellä mainitun BSD-käyttöjärjestelmän lisäksi UNIX-tyylisten käyttöjärjestelmien IPv6-tuki on laajentunut käsittämään useimmat muun muassa yrityskäytössä paljon käytetyt käyttöjärjestelmät, kuten IBM:n AIX ja Hewlett-Packardin HP-UX. [48]

Reititinpuolella IPv6-tuki on jo arkipäivää. Edellä mainitun Ciscon muita tunnettuja reititinvalmistajia on muun muassa Nortel Networks, Juniper Networks ja Hitachi. Luonnollisesti nämä kaikki tukevat IPv6-protokollaa. Ciscon lisäksi ainakin Hitachilla on myös Mobile IPv6 -toteutuksia, koska Hitachin IPv6 perustuu KAME:n protokollapinoon. Hitachi on myös mukana japanilaisessa WIDE-projektissa, joka kehittää tulevaisuuden tietoverkkoja. [48][60]

Myös mobiilipuolella on ollut kiinnostusta IPv6:n ja MIPv6:n kehittelemiseksi, johtuen tulevaisuuden visioista kehittää 3/4G-verkoista entistä enemmän Internetiin sidoksissa olevia järjestelmiä. Mobiililaitteissa paljon käytettyyn Symbian käyttöjärjestelmään IPv6-tuki tuli version 7 mukana. Luonnollisesti johtavat matkapuhelinvalmistajat kilvan kehittävät IPv6 ja Mobile IPv6 -toteutuksiaan. Maailman suu-

rin matkapuhelinvalmistaja Nokia on myös kehitellyt IPv6-protokollaa tukevia puhelimia Symbia-käyttöjärjestelmän avustuksella. Vuonna 2004 Nokia suoritti myös esittelyluontoisesti maailman ensimmäisen Mobile IPv6 -puhelun *3G World Congress Convention and Exhibition* -kongressissa Hong Kong:ssa. [46][47]

3 Mobile IPv6 yhteentoimivuus

Mobile IPv6 on protokollana vielä suhteellisen nuori ja sen standardointikin valmistui vasta vuonna 2004. Tälle protokollalle, tai IPv6:n laajennukselle, on kuitenkin tehty kehitystyötä erilaisten käyttöjärjestelmien ja laitealustojen parissa. Tulevaisuudessa näitä toteutuksia tulee myös olemaan mitä erilaisimmissa laitteissa ja niiden yhteentoimivuus keskenään olisi suotavaa. Kehittäjät tuskin ovat kovinkaan paljoa testanneet tuotoksiaan toisten projektien toteutusten kesken, joten ongelmia yhteentoimivuudessa varmastikin on. Yhteentoimivuus myös riippuu paljon siitäkin, kuinka standardinmukainen kunkin kehittäjän toteutus on ja myös siitä, että eri käyttöjärjestelmät saattavat toimia muutenkin hieman erilailla eri tilanteissa.

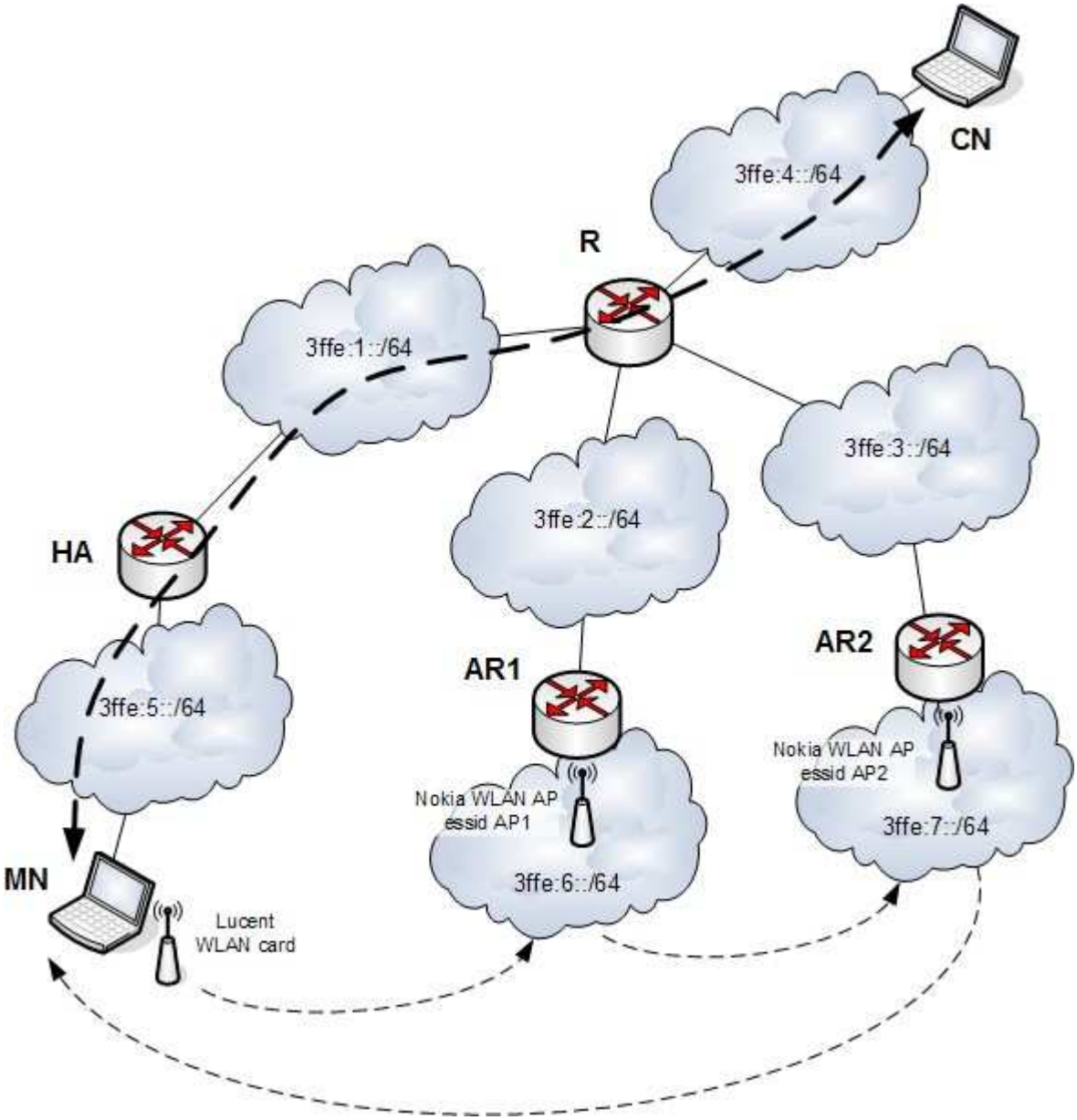
Tämän tutkielman aikana pyrittiin tutustumaan kolmen tällä hetkellä kehittyneimmän Mobile IPv6 -toteutuksen toimintaan ja selvittämään niiden yhteentoimivuutta myös keskenään Mobile IPv6 -verkon eri laitteiden osalta. Tutkimusympäristö koostui MIPL:n, KAME:n ja Ciscon MIPv6-toteutuksista. Seuraavissa kappaleissa kerrotaan tutkimusympäristöstä ja tehdyistä testeistä. Testeissä tutkittiin eri toteutusten toimintaa keskenään ja mahdollisia eroavaisuuksia yhteysvastuun vaihdon viiveissä.

3.1 Tutkimusympäristö

Tutkimusta varten laboratorioon asennettiin verkko, joka muodostui kuvan 3.1 osoittamasta topologiasta. Verkkoon oli liitettynä runkoreititin R, liityntäreitittimet AR1 ja AR2, kotiagentti HA, liikkuva päätelaite MN sekä liikennöintikumppani CN. Tähtitopologia runkoreitittimen, liityntäreitittimien, kotiagentin ja liikennöintikumppanin kesken oli kaapeloitu toisiinsa Ethernet-liitynnöillä. Liityntäreitittimien liitynnät ulkoverkkoihin päin oli toteutettu WLAN-liitynnöillä. Liikkuvassa päätelaitteessa oli kaksi liityntää: Ethernet-liityntä kotiverkkoon kytkeytymistä varten sekä WLAN-liityntä liikkuvan päätelaitteen suoritettaessa yhteysvastuun vaihto vierasverkkoihin 3ffe:6::/64 ja 3ffe:7::/64.

Tutkimusympäristössä runkoreititin R sekä liityntäreitittimet AR1 ja AR2 pysyivät muuttumattomina koko testauksen ajan. Nämä olivat Linux-reitittimiä, joihin verkkoasetukset oli konfiguroitu kokonaan IPv6-pohjaisesti. HA, MN ja CN olivat Mobile IPv6 -toiminnallisuuden omaavia laitteita, joihin oli asennettu MIPL ja

KAME -toteutukset vastaavasti Linux- ja FreeBSD-käyttöjärjestelmien päälle. Lisäksi Cisco-reititintä HA-toiminnallisuudella varustetulla IOS-käyttöjärjestelmällä käytettiin testeissä kotiagenttina. Näitä Mobile IPv6 -toteutuksia vaihdeltiin testien kuluessa eri laitteissa ja testattiin kaikkien kombinaatioiden toimivuutta keskenään.



Kuva 3.1: Tutkimusympäristön testiverkko.

3.1.1 Laitteisto

Tutkimusverkko rakennettiin kuvan 3.1 mukaisesti ja laitteistona käytettiin taulukon 3.1 mukaisia päätelaitteita. Kuten taulukosta 3.1 voi huomata, käytetty laitteisto oli varsin kirjavaa. Laitteet jouduttiin valitsemaan laboratoriossa joutilaina olleista PC-laitteista, eikä resurssien puutteista johtuen voitu ajatella täysin tasaveroisen laitteiston hankkimista tämän tutkimuksen käyttöön. Toisaalta käytetty laitteisto kuvaa myös tilannetta, joka aidossa tilanteessa saattaisi esiintyä. Erilaisia päätelaitteita erilaisine resursseineen kommunikoi keskenään Mobile IPv6 -verkossa.

Taulukko 3.1: Tutkimusympäristön laitteisto.

Laite	**CPU**	**RAM (MB)**	**OS**
R	Athlon XP1700+ 1110 MHz	512	1
AR1	PIII 670 MHz	160	1
AR2	PIII 600 MHz	128	1
HA	Celeron 700 MHz	512	2,3,4
MN	PIII 900 MHz	128	2,3
CN	PIII M 1000 MHz	512	2,3
Käyttöjärjestelmät: 1) Ubuntu Linux 6.06, kernel 2.6.15 Ubuntu 2) Ubuntu Linux 6.06, kernel 2.6.16, MILP 2.0.2 3) FreeBSD 5.4, kame-20060710-freebsd54-snap 4) Cisco 3600, IOS 12.4(7)			

Runko- ja liityntäreitittimien käyttöjärjestelmäksi valittiin Linux jakeluversionaan Ubuntu 6.06. Tämän jakeluversion ydin oli versiota 2.6.15. Linux valittiin reititinkäyttöön helpon asennuksen takia ja se on todettu olevan vakaa myös reititin käytössä. Lisäksi uusimmissa Linux-ytimissä on vakiona IPv6-protokolla päällä, joten ylimääräisiä asennuksia tämän takia ei tarvinnut tehdä perusasennuksen lisäksi. Runko- ja liityntäreitittimien verkkoasetukset tehtiin täysin IPv6-pohjaisiksi ja kuvan 3.1 mukaisine verkkoineen.

Kotiagentin käyttöjärjestelmää ja Mobile IPv6 -toteutusta vaihdeltiin kesken testien. Kotiagentin MIPv6-toteutukset olivat Linuxille MIPL 2.0.2, FreeBSD:lle kame-20060710-freebsd54-snap ja Ciscolle IOS-versio 12.4(7). MIPL versio 2.0.2 on tällä hetkellä sen uusin julkaisuversio ja tämä versio vaatii Linux-ytimen 2.6.16. FreeBSD käyttöjärjestelmäksi valittiin versio 5.4, joka on jo parin vuoden takaa. Tähän on syynä, että KAME-projektin tarjoamat ”snapshot”-julkaisut lähdekoodeineen muo-

kattuine ytimineen ja käyttöympäristön (engl. *userland*) ohjelmistoineen on tarjolla ainoastaan tälle versiolle. Laboratoriossa tarjolla ollut Cisco-reititin oli mallia 3640 ja siihen asennettiin IOS-versio 12.4(7), joka oli sillä hetkellä viimeisin versio kyseiselle reititinmallille.

Liikkuvan päätelaitteen MIPv6-toteutusta vaihdettiin myös testien välillä, joten tähän laitteeseen asennettiin sekä Linux että FreeBSD liikkuvan päätelaitteen MIPv6 -toiminnallisuudella. Kuten kotiagentissa, tähänkin asennettiin Linux MIPL-versiolla 2.0.2 ja FreeBSD KAME-kehitysversiolla kame-20060710-freebsd54-snap.

Myös liikennöintikumppanin MIPv6-toteutusta vaihdettiin myös testien välillä, joten myös tähän laitteeseen asennettiin sekä Linux että FreeBSD liikennöintikumppanin MIPv6 -toiminnallisuudella. Kuten kotiagentissa ja liikkuvassa päätelaitteessa, tähänkin asennettiin Linux MIPL-versiolla 2.0.2 ja FreeBSD KAME-kehitysversiolla kame-20060710-freebsd54-snap. Myös Windows XP:tä kokeiltiin, koska siinä pitäisi olla CN-tuki reitin optimoinnilla. IPv6 ja CN-asetukset asetettiin päälle, mutta testien mukaan XP ei kuitenkaan toiminut reitin optimoinnilla Linuxin ja FreeBSD:n kanssa. XP:n reitin optimointi on suojattu AH-todennuksella, mikä ei kuitenkaan ole tuettu MIPL:ssä tai KAME:ssa. XP:n salaustoiminnon kuitenkin saa poistettua päältä, mutta tällä ei ollut vaikutusta toimivuuteen. Tämän takia XP jätettiin pois liikennöintikumppanin roolista.

3.1.2 Verkko

Tutkimusympäristön verkko rakennettiin kuvan 3.1 mukaisesti ja se muodostui seitsemästä eri IPv6-verkosta. Liikkuva päätelaite liittyi Ethernet-liitännällä kotiverkkoon 3ffe:5::/64. Liikkuvalla päätelaitteella oli myös toinen liitäntä, joka oli WLAN-liitäntä vierasverkkoihin 3ffe:6::/64 ja 3ffe:7::/64 siirtymistä varten.

Keskusreitittimeen R oli kytketty kaikki muut laitteet, joten se sisälsi neljä Ethernet-liitäntää verkkoja 3ffe:1—4::/64 varten. AR1 kytkettiin verkkoon 3ffe:2::/64 ja AR2 verkkoon 3ffe:3::/64. Lisäksi näillä liityntäreitittimillä oli toiset liitännät ulkoverkkoihin 3ffe:6::/64 ja 3ffe:7::/64 päin ja nämä olivat WLAN-liitäntöjä, joiden kautta liikkuva päätelaite suoritti siirtymänsä. WLAN-tukiasemina käytettiin kahta vanhaa Nokian A020 ja A021 tukiasemaa sekä WLAN-sovittimena liikkuvassa päätelaitteessa Lucent PCMCIA-korttia.

Liikennöintikumppani oli liitetty suoraan keskusreitittimeen Ethernet-liitännällä ja sen verkkoon 3ffe:4::/64. Kaikki verkot ja reititykset tehtiin staattisilla asetuksilla liitteen A mukaisesti. Lisäksi kotiagentin kotiverkon 3ffe:5::/64 puoleiseen liityntään sekä liityntäreitittimien WLAN-liityntöihin oli asetettu *radvd*-palvelu (engl.

Router Advertisement Daemon) lähettämään reititinmainostuksia kyseisiin verkkoihin. Kotiagentti asetettiin standardin [32] mukaisesti lähettämään RA-viestejä liityntnällään 3ffe:5::1/64, johon kotiagentti oli asetettu ja liityntäreitittimet lähettämään RA-viestejä verkko-osoitteidensa mukaisilla prefixeillä. LIITE A.

Kuten edellä jo mainittiin, niin liitteestä A löytyy tutkimusverkon konfiguraatioita. Nämä ovat tutkimusverkossa MIPL-järjestelmässä käytettyjä esimerkkejä tärkeimmistä asetuksista. Asetusten tekoon löytyy ohjeita myös esimerkiksi MIPL:n ja KAME:n asennuspaketeista sekä kehittäjien kotisivuilta. [54][57] KAME-sivustolta [57] löytyy myös *Newsletter*-osio [58] ja aktiivinen sähköpostilista, joista löytyy apua ja ohjeita uusimpien versioiden asentamiseen. Ciscon konfigurointiin löytyy myös apua yrityksen kotisivuilta [49] ja eri IOS-versioissa samojen asetusten teko voi olla hyvinkin erilaista. Muita ohjeita (M)IPv6-protokollan asettamiseksi tiettyyn järjestelmään voi etsiä järjestelmien kehittäjien kotisivuilta ja muilta aiheeseen liittyviltä Internet-sivuilta.

3.2 Suorituskykytestit

Tutkimusympäristössä oli sen rakentamisen jälkeen tarkoitus ajaa erilaisia testejä, jotta saataisiin kuva käytössä olevien Mobile IPv6 -toteutusten toiminnasta keskenään. Testeissä tutkittiin etupäässä yhteysvastuun vaihdon aiheittamia viiveitä eri järjestelmien kesken. Laitteet konfiguroitiin käyttämään reitin optimointia liikkuvan päätelaitteen ja liikennöintikumppanin välillä. Laitteissä käytettiin myös kahta erilaista liityntätapaa verkkoon, eli Ethernet- ja WLAN-liityntää, joiden perusteella liikkuvan päätelaitteen moniliitäntäisyyden (engl. *multihoming*) toimivuutta voitiin tutkia.

Ajatuksissa oli myös käyttää IPsec-suojausta signalointiviesteille, mutta KAME-järjestelmän ongelmat yhteentoimivuudessa MIPL-järjestelmän kanssa kaatoivat tämän vaihtoehdon. IPsec-suojaus toimi kyllä täydellisesti, jos järjestelmä oli kokonaan MIPL tai KAME. Myös liikkuvan päätelaitteen ollessa MIPL ja kotiagentin sekä liikennöintikumppanin ollessa MIPL tai KAME, IPsec-suojaus toimi. Ongelma IPsec:n kanssa esiintyi liikkuvan päätelaitteen ollessa KAME ja kotiagentin MIPL siirryttäessä vierasverkosta 1 vierasverkkoon 2. Tällöin KAME MN:n lähettäessä HoTI-viesti, kotiagentti hylkäsi aina liikkuvalta päätelaitteelta saapuneen HoT-viestin. Toisaalta Cisco saatiin viivetesteihin mukaan, koska siinä ei ole vielä MIPv6:n IPsec-tukea.

Liikkuvan päätelaitteen liikettä verkossa simuloitiin shell-skriptillä (ks. esimerkki liitteestä B), jossa liikkuvan päätelaitteen Mobile IPv6 -toiminnallisuus käynnis-

tettiin, asetettiin liityntöjä päälle tai pois ja WLAN-liityntää yhdistettiin eri tukiasemiin liikkumisen eri vaiheissa. Liikkuva päätelaite viipyi viisitoista sekuntia jokaisessa verkossa (3ffe:5->6->7->5::/64) liikkumisen aikana ja tämän todettiin olevan riittävä aika yhteysvastuun vaihdon suorittamiseksi.

3.2.1 Testausmenetelmät

Tutkimusympäristön laitteisiin asennettiin siis erilaisia käyttöjärjestelmiä ja niiden MIPv6-toteutuksia taulukon 3.1 mukaisesti. Samat testit tehtiin eri käyttöjärjestelmien kombinaatioilla ja näistä yritettiin selvittää yhteysvastuun vaihdon aikana tapahtuvia viiveitä ja toiminnan erilaisuutta eri järjestelmien välillä. Mobile IPv6 -protokollan ominaisuuksista käytettiin reitin optimointi kytkettynä, mutta IPsec-toiminnallisuutta ei otettu mukaan KAME-järjestelmässä ilmenneiden ongelmien takia.

Testejä tehtiin kolmella erilaisella liikenteellä lähettäen paketteja liikennöintikumppanilta liikkuvalle päätelaitteelle päin. Kaikki testidata puolestaan otettiin talteen liikkuvalla päätelaitteella *Ethereal*-verkkoanalysaattorin [66] avulla. Testit tehtiin käyttämällä ICMPv6- (Ping6) , UDP- ja TCP -liikennettä seuraavasti:

Ping – 100 x 128 tavua / s ja 1000 x 128 tavua / s
UDP – 100 x 128 tavua / s ja 1000 x 128 tavua / s
TCP – Tiedoston lataaminen HTTP-palvelimelta

Ping-liikennettä muodostettiin käyttämällä Linuxin ja FreeBSD:n *ping6*-sovellusta lähettämällä 128 tavun kokoisia paketteja intervalleilla 100 ja 1000 pakettia sekunnissa. Tosin FreeBSD ei pystynyt lähettämään pingiä intervallilla 1000/s, joten nämä testit jätettiin suorittamatta. UDP-liikennettä muodostettiin käyttämällä *MGEN*-liikennegeneraattoria [65] lähettämällä 128 tavun kokoisia paketteja intervalleilla 100 ja 1000 pakettia sekunnissa. TCP-liikennettä simuloitiin asentamalla liikennöintikumppanille HTTP-palvelimet, jonka kautta liikkuva päätelaite latasi isoa tiedostoa itselleen verkossa liikkumisen aikana. Tässä tapauksessa päädyttiin käyttämään HTTP-palvelinta liikennegeneraattorin sijaan, koska IPv6-protokollaa tukevaa TCP-liikennettä tuottavaa generaattoria, joka olisi toiminut sekä Linuxissa että FreeBSD:ssä, ei löytynyt.

Nämä edellä mainitut testit suoritettiin jokaisella MIPv6-toteutuksien kombinaatioilla. Tuloksissa ja taulukoissa näitä on merkitty seuraavan esimerkin mukaan:

MN	HA	CN
MIPL –	Cisco	– KAME

Eli käytetyssä merkintätyylissä ensimmäinen laite on liikkuva päätelaite, toinen laite on kotiagentti ja kolmas laite on liikennöintikumppani. Vastaavasti laitteen paikalla oleva järjestelmä (MIPL/KAME/Cisco) ilmoittaa kombinaatiossa testatun järjestelmän.

Taulukoissa (Liite C ja D) on merkitty verkkoja myös seuraavalla tavalla:

HN	Home Network eli kotiverkko (3ffe:5::/64)
FN1	Foreign Network 1 eli vierasverkko 1 (3ffe:6::/64)
FN2	Foreign Network 2 eli vierasverkko 2 (3ffe:7::/64)

Liikkuvalta päätelaitteelta suoritettiin edellämainitut testit ja otettiin testidata talteen. Testit suoritettiin skriptin avulla, joissa eri verkkoihin siirtyminen tapahtui hieman erilailla seuraavalla tavalla:

HN=>FN1	Sekä Ethernet- että WLAN-liitäntä (essid AP_1) ylhäällä alusta asti. Handover-vaiheessa Ethernet-liitäntä alas.
FN1=>FN2	WLAN-liitäntä ylhäällä ja essid:n vaihdolla siirtyminen toiseen vierasverkkoon (essid AP_2).
FN2=>HN	Ethernet liitäntä ylös ja välittömästi WLAN-liitäntä alas.

Testi sisälsi neljä vaihetta ja jokainen vaihe kesti 15 sekuntia. Ensimmäisessä vaiheessa skriptin käynnistyessä nostettiin liikkuvalla päätelaitteella Ethernet- ja WLAN-liitännät ylös ja muodostettiin yhteys kotiagenttiin. MIPL:ssä edellä mainitut suoritettiin skriptillä laitteen käynnistyksen jälkeen. KAME:ssa sen oma MIPv6-palveluita hallitseva skripti käynnisti MN-toiminnallisuuden laitteen käynnistysvaiheessa ja testausskriptillä asetettiin alussa vain WLAN-liitäntä ylös.

Toisessa vaiheessa, kun siirryttiin kotiverkosta ensimmäiseen vierasverkkoon, asetettiin Ethernet-liityntä alas, jolloin MN-sovellus alkoi käyttää automaattisesti WLAN-liitäntää. WLAN-liitäntähän oli alussa käynnistetty samaan aikaan kuin Ethernet-liitäntäkin ja liitetty se tukiasemaan AP_1. Tässä vaiheessa voitiin testeissä

havainnoida eri järjestelmien kykyä käsitellä moniliitäntäjärjestelmää (engl. *multihoming*). Tässä tapauksessa Ethernet-liitäntä oli prioriteetiltaan korkeampi eli ensisijainen liitäntä.

Kolmannessa vaiheessa siirryttiin ensimmäisestä vierasverkosta toiseen, jolloin ainoastaan WLAN-liitännän essid-arvo vaihdettiin osoittamaan tukiasemaa AP_2. Neljännessä vaiheessa siirryttiin takaisin kotiverkoon, jolloin Ethernet-liitäntä nostettiin ylös ja WLAN-liitäntä kytkettiin pois päältä. WLAN-liitäntää ei tarvitsisi periaatteessa asettaa alas, jos sitä ei tarvitse, koska Ethernet-liitäntä oli prioriteetiltaan ensisijainen liitäntä. WLAN-liitännän alasajo tehtiin kuitenkin varmuuden vuoksi, vaikka moniliitännän toimivuus todettiin kummassakin järjestelmässä (MIPL ja KAME) hyväksi.

Tällä tavalla testattiin kaikki erilaiset järjestelmäkombinaatiot ja käytettiin edellä mainittuja liikennetyyppejä. Seuraavassa kappaleessa analysoidaan testien tuloksia ja testeistä muodostetut löytyvät liitteistä C ja D.

3.2.2 Tulokset ja johtopäätökset

Testejä tehtiin lähettämällä liikennöintikumppanilta ping-, UDP- ja TCP-liikennettä liikkuvalle päätelaitteelle kappaleen 3.2.1 periaatteen mukaisesti. Nämä testit tehtiin kaikkien kolmen käytössä olleen MIPv6-järjestelmän kombinaatioilla; MIPL:ä ja KAME:a kotiagenttina, liikkuvana päätelaitteena ja liikennöintikumppanina, Ciscoa ainostaan kotiagenttina. Taulukot tehdyistä testeistä ja yhteysvastuun vaihdon aikana tapahtuneista viiveistä löytyy liitteistä C ja D.

Viiveet onositeltu liitteiden taulukoissa testin eri vaiheisiin. Vaiheet koostuu yhteysvastuun vaihdon osista joissa liikkuva päätelaite siirtyy toiseen verkkoon. Viiteaika on otettu viimeisestä paketista vanhassa verkossa. Tästä eteenpäin taulukoiden viiveet ovat sekuntia edellisestä arvosta ja taulukon alareunasta löytyy lisäksi kokonaisviiveet eri testin vaiheista. Kokonaisviiveet eri testeistä on esitetty myös kuvissa 3.2 ja 3.3.

Testeissä MIPL 2.0.2 osoittautui suoriutuvan huomattavasti nopeammin yhteysvastuun vaihdosta verrattuna FreeBSD:n KAME-MIPv6 -sovellukseen. Viiveisiin voi vaikuttaa monikin asia, eikä testien tuloksia voi täten suoraan verrata keskenään. Ensinnäkin reitittimien RA-viestejä on vaikea tällaisessa testissä saada osumaan kohdalle juuri silloin, kun liikkuva päätelaite siirtyy vieraaseen verkkoon. Jos tällöin liikkuvalla päätelaitteella ei ole tietoa uudesta reitittimestä, on sen etsittävä NS-viestillä uusia reitittimiä lähistöltä. RA-viestien lähetys on standardissa [32] suositeltu 3-10 sekunnin välille. Kuitenkin RA-viestien minimi- ja maksimiarvoille on

määritelty standardissa myös minimiarvot, jotka ovat 0.03 ja 0.07 sekuntia [32]. Liian nopeaa intervallia ei kuitenkaan tule verkoissa käyttää, koska tämä lisää jo niin paljon liikennettä, että siitä saattaa tulla ongelmia [41]. Näissä testeissä käytettiin MIPL:ssä ja KAME:ssa 0.05 ja 3 sekunnin välistä lähetettäviä RA-viestejä. Ciscon IOS-versiossa oli mahdollisuus asettaa vain yksi arvo RA-viestien intervallille, joten siinä käytettiin 1 sekunnin intervallia.

Myös KAME:n ja MIPL:n toteutuksissa on muitakin eroavaisuuksia. KAME on pyritty kehittämään konservatiivisella tavalla standardin [32] mukaiseksi, kun taas MIPL:ssä tiettyjä toimintoja on pyritty optimoimaan mahdollisesti yhteysvastuun vaihdon nopeuttamiseksi. KAME eroaa MIPv6-rekisteröintiviestien välityksen suhteen niin, että siinä yhteysvastuun vaihdon tapahtuessa liikkuva päätelaite lähettää ensin BU:n kotiagentille ja odottaa BAck-vastausta. Vasta BAck-viestin saapumisen jälkeen se aloittaa RR-prosessin ja lähettää HoTI- ja CoTI-viestit [41]. KAME myös lähettää nämä HoTI- ja CoTI-viestit aina kun liikkuva päätelaite siirtyy vieraaseen verkkoon, paitsi jos siirrytään kotiverkkoon, lähetetään ainoastaan HoTI-viesti. Näiden kaikkien viestien lähettämisen tarpeellisuudesta voi olla montaa mieltä ja MIPL:ssä tätä rekisteröintivaihetta onkin pyritty optimoimaan.

MIPL:ssä MIPv6-rekisteröinnit suoritetaan seuraavan kaavan mukaan. Liikkuvan päätelaitteen siirtyessä vieraaseen verkkoon lähettää se ensimmäiseksi BU-viestin kotiagentille. KAME:sta poiketen se voi lähettää samaan aikaan myös CoTI-viestin suoraan liikennöintikumppanille. Testien mukaan MIPL myös jättää seuraavat HoTI-viestit lähettämättä ensimmäisen yhteysvastuun vaihdon jälkeen, joten MIPL-toteu-tuksessa tietoturvan kannalta ajatellaan ensimmäisen HoTI-rekisteröinnin riittävän identifioimaan MN. CoTI-viestit puolestaan lähetetään jokaisen yhteysvastuun vaihdon yhteydessä.

Seuraavassa yhteenveto tehtyjen testien tuloksista eriteltynä liikkuvan päätelaitteen ollessa MIPL tai KAME. Kuvissa 3.2 ja 3.3 on esitetty kokonaisviiveet liitteiden C ja D tulosten pohjalta.

Liikkuvana päätelaitteena MIPL

Kuvassa 3.2 nähdään siis kokonaisviiveet liikkuvan päätelaitteen ollessa MIPL. Kuten aiemmin tuli ilmi, viiveisiin vaikuttaa moni asia ja testien tuloksissa onkin heittoa suuntaan jos toiseen. Reitittimien mainostusviestien ajoituksen lisäksi erilaisen liikenteen käyttäytymisessä on eroja johtuen käytetyistä protokollista (esim. UDP ja TCP) ja myös lähetetyn datan määrä vaikuttaa viiveisiin yhdessä tiedonsiirtolinkkien läpäisykyvyn kanssa.

Kaiken kaikkiaan liikkuvan päätelaitteen näkökulmasta katsottuna MIPL suoriutui tehtävästään huomattavasti paremmin kuin KAME. Vertaamalla kokonaisviiveitä eri testeistä kuvissa 3.2 ja 3.3 voidaan todeta, että MIPL suoriutui kaikista testeistä seitsemän sekunnin sisällä, kun taas KAME:lla suoritetuissa vastaavissa testeissä aikaa kului pahimmillaan kaksinkertainen määrä.

Kotiagentin ollessa MIPL ja liikennöintikumppanin KAME havaittiin testien sujuneen hieman nopeammin kuin CN:n ollessa MIPL. Varsinkin tämä esiintyi UDP- ja TCP-liikenteellä MN:n palatessa vierasverkosta FN2 kotiverkkoon. PING-liikenteellä tulokset olivat taas lähes yhteneväiset. Suurimmat viiveet esiintyivät siirryttäessä kotiverkosta vierasverkkoon FN1. Tässähän sekä Ethernet- että WLAN-liitäntä on ylhäällä alusta asti ja yhteysvastuun vaihto laukaistaan Ethernet-liitännän alasajolla. Tästä voisi ajatella, että tämän skenaarion pitäisi olla nopein, koska liitännät ovat alusta asti ylhäällä. Kuitenkin liikenteen nopeuden nosto aiheutti muun muassa BU-viestien katoamista, jolloin MN joutui lähettämään niitä uudestaan HA:lle.

Kotiagentin ollessa KAME ja myös CN:n KAME näissä testeissä saatiin hieman parempi tulos kuin CN:n ollessa MIPL. KAME:n paremmuudesta ei voi kuitenkaan olla varma erojen pienuuden ja mainostusviestien ajoituksien takia. MIPL ja KAME ovat suhteellisen tasavertaisia, kun verrataan HA:n ja CN:n toimintaa näiden eri järjestelmien kesken.

Kotiagentin ollessa Cisco saatiin yllättäen parhaimmat ja tasaisimmat tulokset. Lisäksi ero CN:n ollessa KAME tuli näissä testeissä esiin paremmin kuin edellisissä. PING-liikenteessä MIPL oli hieman parempi, mutta UDP-liikenteessä varsinkin yhteysvastuun vaihto vierasverkkojen FN1 ja FN2 välillä kesti kauemmin CN:n ollessa MIPL. Sama ilmiö toistui TCP-liikenteellä. Kuitenkin yhteysvastuun tapahduttua tilanteessa, jossa siirrytään pois tai palataan kotiverkkoon, oli suhteellisen tasaista CN:n kaikissa tapauksissa. Kotiverkkoon liittyvistä viiveistä voisi päätellä, että Ciscon reititinominaisuudet olisi paremmat Linux-pohjaiseen reitittimiin verrattuna, joita käytettiin vierasverkoissa.

Yleisesti ottaen kokonaisviiveet mobiiliverkossa, jossa MN on MIPL, ovat suhteellisen pieniä ja hyväksyttäviä MN:n kommunikoidessa CN:n kanssa. Tässä testissä kokonaisviiveet reitin optimointia käyttäen, eli viimeisen paketin vanhassa verkossa ja uuden paketin uudessa verkossa vierasosoitetta käyttäen välinen aika, oli pahimmillaankin alle seitsemän sekuntia. Vastaavasti parhaimmissa tapauksissa päästiin noin puolen sekunnin viiveisiin. Paketteja siirtyi myös vanhalla HoA-osoitteella liikuvan päätelaitteen ja kotiagentin välisen tunnelin kautta ennen RR-prosessin suorittamista testin ensimmäisen osan aikana, jossa siirryttiin kotiverkosta vierasverkkoon. Tässä tapauksessa sovelluksen tiedonsiirron katkos oli BU/Back-

viestien vaihdon jälkeen keskimäärin noin yksi sekunti. Liitteestä C voi ottaa tarkemmin selvää pakettien välisistä viiveistä.

IPsec-suojausta ei KAME:n ongelmien takia voitu käyttää, joten se jätettiin pois käytöstä. Vertailun vuoksi otettiin kuitenkin MIPL-järjestelmän kesken testiajo IPsec päällä. Tuloksen löytyvät liitteestä C. Testin perusteella ei IPsec-suorituskyvystä voi vetää suura johtopäätöksiä, mutta IPsec: käyttö (salaus/purku) ei näyttäisi kovinkaan paljoa vaikuttavan viiveisiin.

Liikkuvana päätelaitteena KAME

Kuvassa 3.3 nähdään kokonaisviiveet liikkuvan päätelaitteen ollessa KAME. Vertaamalla MIPL:llä suoritettuihin vastaaviin testeihin KAME:n viiveet olivat lähes kaksinkertaisia MIPL:in verrattuna. Suurilta osin tämä johtuu KAME:n konservatiivisen standardinmukaisesta tavasta lähettää rekisteröintiviestejä yhteysvastuun vaihdon aikana. KAME:n MIPv6-pinoa ei ole optimoitu kuten MIPL:n vastaavaa.

Viiveitä KAME-järjestelmässä lisää se, että vasta BAck-viestin saapumisen jälkeen lähetetään HoTI/CoTI-viestit. Kuten aikaisemmin jo kerrottiin, MIPL MN puolestaan lähettää samaan aikaan BU- ja CoTI-viestit, jolloin viiveissä säästetään BAck-viestin saapumiseen kestävä aika. Lisäksi KAME MN lähettää aina HoTI-viestin vaihtaessaan verkkoa ja myös CoTI-viestin aina, paitsi kotiin palattaessa. MIPL:ssä lähetetään HoTI kun siirrytään kotoa ensimmäisen kerran vieraaseen verkkoon, mutta ei muulloin. CoTI lähetetään aina lukuun ottamatta takaisin kotiverkkoon siirtymistä. Nämä eroavaisuudet näyttävät olevankin suurin syy KAME:n kangerteluun yhteysvastuun vaihdoissa.

Kuten kuvasta 3.3 nähdään, KAME-testeissä suurimmat viiveet kohdistuivat vierasverkkojen välillä tapahtuneeseen yhteysvastuun vaihtoon. Jopa lähes 15 sekunnin viiveitä havaittiin testeissä erityisesti TCP-liikenteen testeissä. Tämä saattaa johtua KAME:n IPv6-pinon TCP-protokollan mukautuvuudesta yhteyden katkeamiseen yhteysvastuu vaihdon aikana. Toisaalta KAME-laitteessa käytetty HTTP-palvelin oli vanhempaa versiota kuin MIPL-laitteessa käytetty. Kotiagentin ollessa KAME ei suuria muutoksia liikennöintikumppanin vaihdellessa ollut. Viiveet vaihtelivat keskimäärin noin kolmen ja kahdeksan sekunnin väleillä riippuen liikenteestä.

Kotiagentin ollessa MIPL ei suuria muutoksia liikennöintikumppanien kesken taaskaan ollut. Viiveet liikkuivat tässäkin tapauksessa noin kolmen ja kahdeksan sekunnin väleillä lukuun ottamatta TCP-liikennettä. MIPL liikennöintikumppanina tuotti ehkä kuitenkin hieman tasaisempia tuloksia, mistä voisi päätellä FreeBSD:n

KAME:n (M)IPv6-viestien käsittelyjen vasteen olevan hieman hitaampaa kuin Linuxin MIPL:n.

Cisco kotiagenttina tuotti taas verrattain hyviä ja tasaisia tuloksia. Viiveet pysyttelivät alla kymmenen sekunnin, lukuun ottamatta taas TCP-liikennettä. Suurimmat viiveet tapahtuivat taas liikkumisessa vierasverkkojen välillä. Tästä voi päätellä, että KAME:n (M)IPv6-toteutuksen liikkeen havaitsemisen aikana tapahtuva viestien vaihto (MLD/NS/NA/RA) ja käsittely on hitaampaa, kuin MIPL:n toteutuksessa.

KAME:n ollessa MN havaittiin myös hitautta yhteysvastuun vaihdossa, kun siirryttiin kotiverkosta vierasverkkoon. Tämä voi osoittaa, että KAME:n moniliitännän tuki ei vielä ole niin kehittynyt kuin MIPL:n vastaava. Liitteestä D voi analysoida tuloksia ja huomata useiden sekuntien viiveitä liitäntää vaihdettaessa ja BU-viestiä lähetettäessä. Aika ajoin myös kotiinpaluussa havaittiin ongelmia. Vaikka Ethernet-liitäntä oli jo nostettu ylös, niin KAME MN lähetti HoTI-viestin WLAN-liitännän kautta. WLAN-liitäntä kytkettiin kuitenkin pois jolloin tiedonsiirto katkesi epäonnistuneen HoT-kuittauksen johdosta.

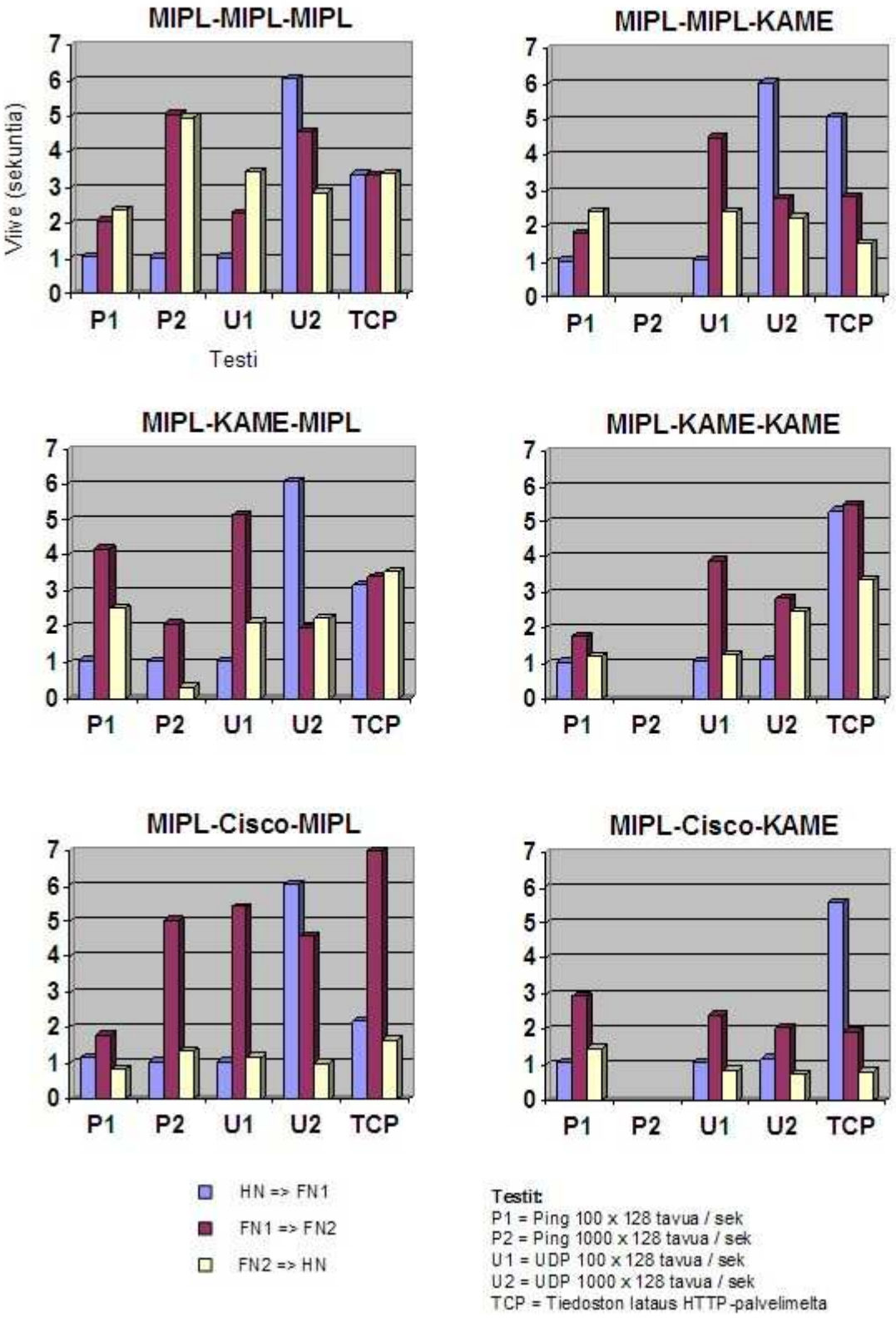
Kokonaisuudessaan KAME toimi kuitenkin vakaasti, mutta viiveet sen standardinmukaisen MIPv6-toteutuksen viestienvaihdossa olivat pääsyy tuottamaan heikomman tuloksen MIPL:n kanssa näissä testeissä. Liitteestä D voi ottaa tarkemmin selvää pakettien välisistä viiveistä.

Yhteenvetona näistä MIPL oli selvästi nopein näissä testeissä liikkuvan päätelaitteen näkökulmasta katsottuna, johtuen optimoidusta rekisteröintiviestien vaihdosta. Cisco toimi vakaasti kotiagentin roolissa ja jopa nopeammin kuin MIPL ja KAME. Ciscon suurin puute tällä hetkellä onkin, ettei se tue vielä IPsec:ä MIPv6-ympäristössä. Reitittimenä se on kuitenkin tehokkaampi Linuxiin ja FreeBSD:hen verrattuna, onhan se ainoastaan reitittimeksi suunniteltukin. Kaikissa järjestelmissä ei-reaaliaikainen tiedonsiirto onnistuu viiveiden puolesta lähes ongelmitta, mutta reaaliaikasovellusten käyttöä ajatellen viiveet ovat vielä liian suuria.

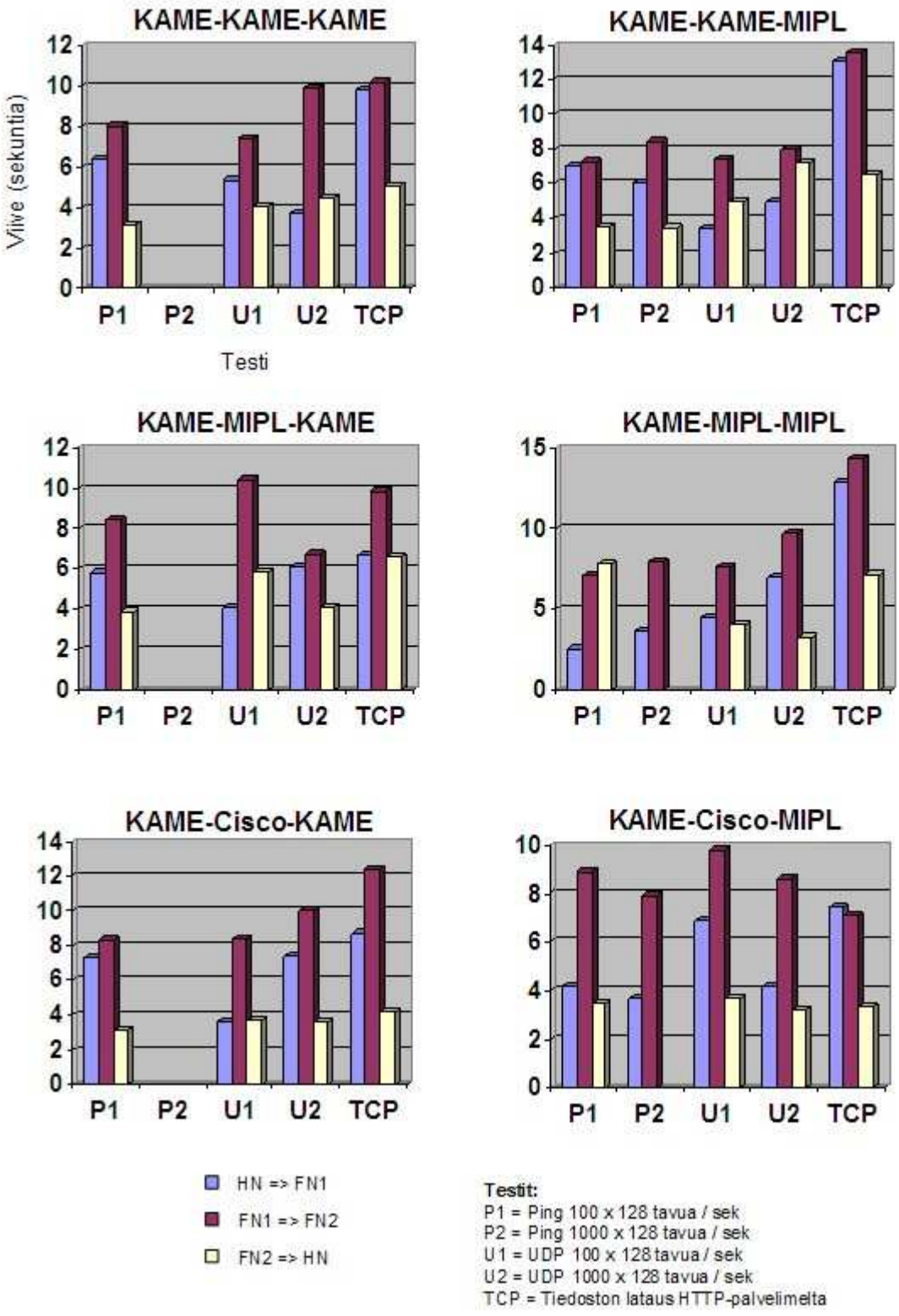
3.3 MIPv6-suorituskyky tutkimuksissa

Mobile IPv6 -protokollaa on tutkittu jo runsaasti yliopistojen ja yritysten erilaisissa tutkimuksissa. Myös eri foorumeilla ja isoilla hankkeilla, joista jo kappaleessa 1 kerrottiin, on paljon tutkimustuloksia tarjottavana. Yksi tuottavimmista IPv6-tutkimuksen saralla on eurooppalainen yhteistyöhanke 6net [17]. Seuraavassa kerrotaan tutkimuksista, joissa on käsitelty MIPv6:n suorituskykyä.

6net:n julkaisussa [40] kerrotaan yleisesti IPv6 ja MIPv6:sta. Julkaisussa [39] on tutkittu yhteysvastuun vaihdon suorituskykyä. Tutkimuksessa tutkittiin miten Mo-



Kuva 3.2: Testien tuloksia. Liikkuvan päätelaitteen ollessa MIPL.



Kuva 3.3: Testien tuloksia. Liikkuvan päätelaitteen ollessa KAME.

bile IPv6:n yhteysvastuun vaihdon aikana tapahtuva viestien vaihto vaikuttaa viiveisiin. Viiveet todettiin olevan vielä tällä hetkellä liian suuria reaaliaikasovellusten ajamiseen sujuvasti MIPv6-verkossa. Tutkimuksessa tutkittiin myös FMIPv6-protokollan (MIPv6:n nopeat yhteysvastuun vaihdot) [34] ominaisuuksia ja sen tarjoamaa suorituskyvyn parannusta yhteysvastuun vaihdon suorittamiseen perinteiseen MIPv6-protokollaan verrattuna.

Julkaisussa [41] tutkittiin MIPv6-protokollaa, reitin optimointia ja yhteysvastuun vaihdon aikana tapahtuvia viiveitä ja niiden syitä. Julkaisussa oli selvitetty yhteysvastuun vaihdon kulkua varsin syvällisesti ja kerrottu syitä nykyisen protokollan ongelmakohdista. Tutkimuksessa oli kehitelty kolme yhdistettyä menetelmää yhteysvastuun vaihdon nopeuttamiseksi: reagoivat yhteysvastuun vaihdot modifoimattomissa reitittimissä ja reitittimien tuella, sekä liikkeen ennakointi ja ennakoivan yhteysvastuun vaihdon hallinta.

Julkaisussa [42] analysoitiin MIPv6 yhteysvastuun vaihtoja langattomissa lähiverkoissa. Tässäkin tutkittiin edellä mainittua FMIPv6-protokollaa ja keskityttiin vertaamaan yhteysvastuun vaihdon viiveitä neljässä eri tapauksessa: perinteisessä MIPv6-protokollassa, MIPv6:n edelleenohjaus menetelmässä, odotuksenmukaisessa menetelmässä ja tunnelointiin perustuvassa menetelmässä. Yhteysvastuun vaihdon viiveitä laskettiin L2-tason ominaisuuksien perusteella. Tutkimuksessa todettiin L2-tason yhteysvastuun vaihdon viiveen olevan tärkeässä asemassa. MIPv6 ja FMIPv6:n välisissä testeissä FMIPv6:n todettiin olevan suorituskykyisempi, mutta optimaalisissa oloissa MIPv6 pääsee lähelle. Tunnelointiin perustuva menetelmä suoriutui nopeammin kuin odotuksenmukainen menetelmä, mutta tunnelointiin perustuva joutuu käyttämään vanhaa osoitetta kauemmin kuin odotuksenmukaisessa menetelmässä. Tässäkin tutkimuksessa todettiin MIPv6 ja FMIPv6:n tarjoavan liian pitkiä viiveitä reaaliaikaisia sovelluksia ajatellen.

Julkaisussa [43] tutkittiin vertikaalisen yhteysvastuun vaihdon ominaisuuksista heterogeenisissä verkoissa. Tässä keskityttiin nimenomaan GPRS solukkoverkkojen ja WLAN-verkkojen vertailemiseen. Vertikaalisessa yhteysvastuun vaihdossahan on kyse eri tiedonsiirtorajapintojen vaihtamisesta parhaimman yhteyden mukaan. Moniliitäntäisyys tulee olemaan tärkeä asia tulevaisuuden liikkuvissa päätelaitteissa. Tutkimuksessa löydettiin ongelmia GPRS- ja WLAN-liitäntöjen erilaisuudesta siirtoyhteystasolla. Myös GPRS:n tapa puskuroida liikennettä vaikuttaa haitallisesti yhteysvastuun vaihdon viiveisiin. Tutkimuksen mukaan verkkokerroksen optimointia eri tiedonsiirtorajapintojen välillä tarvittaisiin.

Julkaisussa [44] tutkittiin myös heterogeenisiä verkkoja. WLAN-, GPRS- sekä UMTS-verkkojen yhteentoimivuutta ja suorituskykyä testattiin käytännössä testi-

verkon avulla TCP-liikennettä tutkien. Myös IPv4:n ja IPv6:n yhteentoimivuutta keskenään tutkittiin. Reaaliaikasovellusten pyörittäminen MIPv6-verkoissa vaatii-kin alussa IPv4:n kanssa yhteentoimivuutta ja suorituskykyä yhteysvastuun vaihdon aikana tapahtuvalta prosessilta. Tutkimuksessa todettiin optimaalisissa olosuhteissa L2-laukaisimien avulla moniliitäntäisessä ympäristössä päästävän 0.2 sekunnin yhteysvastuun vaihdon viiveisiin.

Yhteenvetona tehdyistä tutkimuksista MIPv6:n saralla on se, että protokolla ei ole vielä kypsä tuotantokäyttöön vielä ratkaisemattomien ongelmien takia. Yhteysvastuun vaihtoon pitäisi vielä kehittää uusia ratkaisuja, joilla saataisiin rekisteröintien aiheuttamaa viivettä lyhennettyä reaaliaikasovellusten vaatimusten mukaiseksi.

4 IPv6-verifiointi

Internet-protokollan vaihtaminen IPv4:stä IPv6:een ei ole tapahdu hetkessä. IPv6-protokollaa suunniteltaessa riittävän pitkä siirtymävaihe on otettu huomioon, jotta protokollaa pystyttäisiin riittävästi kattavasti testaamaan ja löytämään sen mahdollisia ongelmakohtia. IPv6-protokollan kehityksen apuna on ollut nykyinen Internet, joka tarjoaa kehittäjille yksinkertaisen ja nopean tavan vaihtaa tietoa protokollakehityksen saralta.

Yhteentoimivuus eri järjestelmiin kehitettyjen toteutusten kesken riippuu niiden yhdenmukaisuudesta määriteltyjen spesifikaatioiden suhteen. Kun toteutus on osoittanut olevansa yhdenmukainen spesifikaation suhteen, täytyy se vielä testata ympäristössä, joka sisältää eri järjestelmien tarjoamia toteutuksia.

IPv6-spesifikaatioiden yhdenmukaisuuden tarkastamiseksi on tarjolla sekä ilmaisia että kaupallisia tuotteita. Yksi käytetyimmistä lienee japanilainen TAHI-testausympäristö [61]. Se tarjoaa kattavat testit IPv6- sekä MIPv6-järjestelmien testaamiseksi spesifikaatioiden mukaan. Seuraavissa kappaleissa kerrotaan TAHI-testeistä yleisesti, sekä tutkimusympäristössä tehdyistä yhdenmukaisuustesteistä MIPv6-ympäristössä.

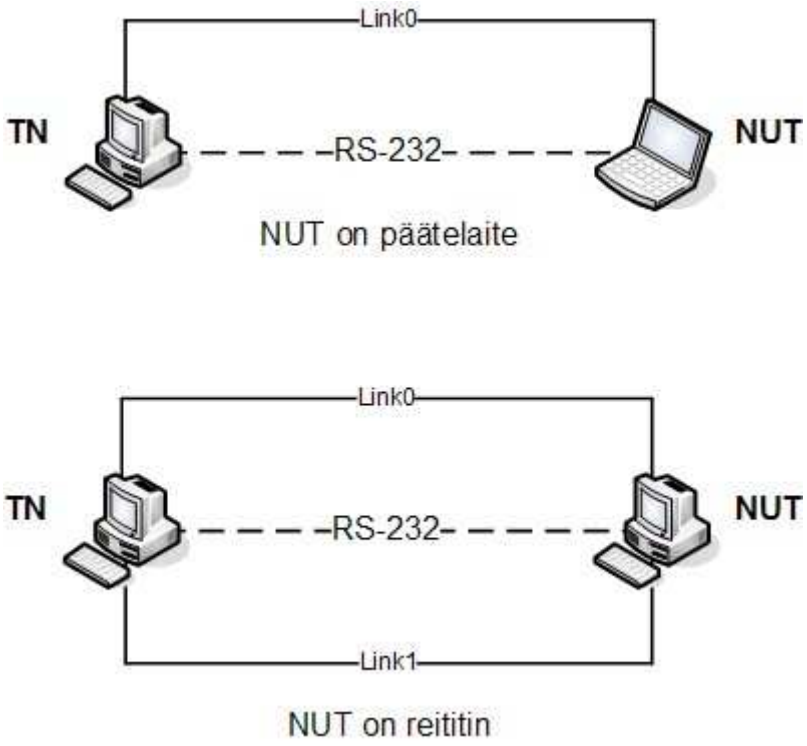
4.1 Taustaa

TAHI-projekti sai alkunsa vuonna 1998, jolloin siihen liittyin Tokion yliopisto, Yokogawa Electric Corp. ja YDC Corp. Tällä hetkellä toimintaa pyörittävät Tokion Yliopisto ja Yokogawa Electric. Projektin tavoitteena oli kehittää ohjelmisto, jonka avulla voidaan testata yhteentoimivuutta eri järjestelmien IPv6-implementaatioiden kesken ja tällä tavalla auttaa kehittäjiä parantamaan tuotteitaan. TAHI-testausympäristö on myös täysin ilmainen ja vapaasti ladattavissa projektin kotisivuilta. [61][45]

TAHI yhdenmukaisuus testit (engl. Conformance Test Suite) ovat kokoelma ohjelmistoja jotka koostuvat yhdenmukaisuus testien työkalusta *v6eval* sekä paketista *ct*, joka sisältää IPv6 yhdenmukaisuus testit.

v6eval on työkalu, joka on suunniteltu generoimaan IPv6-paketteja ja analysoimaan tuloksia. *v6eval* edellyttää, että testausympäristö on kytketty oikein. Testien suorittamiseksi testaava laite (engl. Tester Node (TN)) ja testattava laite (engl. Node Under Test (NUT)) on kytkettävä suoraan yhdellä tai useammalla Ethernet-liitännäl-

lä toisiinsa riippuen testattavan laitteen tyypistä. Testattava laite voi olla joko päätelaite, jolloin tarvitaan yksi Ethernet-kytkentä tai reititin, jolloin tarvitaan useampi Ethernet-kytkentä laitteiden välille. Kuva 4.1



Kuva 4.1: TAHI-testausympäristön kytkentä.

Testin suorittamisen aikana TN lähettää paketteja NUT:lle ja analysoi NUT:n tuottamat vasteet. Jossain tilanteissa tarvitaan myös NUT:n generoimia paketteja. Jos testien suorittaminen halutaan automaattiseksi RS-232 sarjaliitäntä tarvitaan TN- ja NUT-laitteen välille, jolloin singnalointia voidaan suorittaa laitteiden välillä vuorovaikutteisesti. *v6eval* tarjoaa myös mahdollisuuden ajaa testiskriptejä etäyhteyden takaa.

v6eval-ohjelmiston tämänhetkinen (kevät 2007) versio on 3.0.11 ja *ct*-paketin versio on 2.1.1. *ct*-paketti sisältää kattavat IPv6-spesifikaatioiden mukaiset skriptit testaukseen. Tällä hetkellä se sisältää seuraavat testit:

- IPv6 Spesification
- ICMPv6
- Neighbour Discovery for IPv6
- IPv6 Stateless Address Autoconfiguration

- Path MTU Discovery for IPv6
- IPv6 Router Selection
- IPsec
- Mobility Support in IPv6
- Default Address Selection for IPv6
- DNS Discovery
- IPv6 Prefix Options for DHCPv6
- Transition Mechanism for IPv6
- Stateless IP/ICMP Translation Algorithm
- Network Address Translation - Protocol Translation
- Robustness

Kaikki muut testit lukuun ottamatta viimeistä, ovat RFC- tai Draft-dokumenttien mukaisia. *Robustness*-testit ovat suunniteltu verkon kuormituksen keston testaukseen. Sekä *v6eval*-ohjelmistoon että *ct*-skriptipaketteihin tulee aika-ajoin päivityksiä, joista löytyy tietoa projektin kotisivuilta. [61]

TAHI-projektin sivuilta löytyy itsenäisiä testausympäristöjä tiettyjen aihealueiden verifiointiin. Näitä ovat IPv6 Ready Logo Phase 1 ja 2, IPsec Ready Logo Phase 2, MIPv6, NEMO, SIP for IPv6 ja IKE. Myös DNS ja DHCPv6:lle on tarjolla omat testit. MIPv6-paketti on IPv6 Promotion Counsil -työryhmän MIPv6-alaryhmän kehittämä testauspaketti Mobile IPv6 -toiminnallisuuden testaamiseksi. Tämä MIPv6-työryhmä koostuu japanilaisista yrityksistä, joita ovat NTT, Yokogawa Electric, Yaskawa Infromation Systems sekä NTT-AT.

MIPv6-paketti sisältää yhdenmukaisuus- sekä yhteentoimivuustestit MN-, HA- sekä CN-laitteille. Tämänhetkiset MIPv6 yhdenmukaisuus testien ct-paketin versiot ovat MN:lle ja CN:lle ovat 4.0.2 ja HA:lle 4.0.4. Nämä testit tukevat IPv6 Ready Logo 2 vaihetta.

4.2 IPv6 Ready Logo -ohjelma

Edellä jo mainittiin *IPv6 Ready Logo* ja vaiheet 1 ja 2. Mitä tämä sitten tarkoittaa? Eri järjestelmien yhteentoimivuus on kriittinen asia internetin toiminnassa. IPv6-protokollaan siirtyminen on ajan kysymys ja IPv6-implementaatioiden suuresta määrästä johtuen tämä siirtymä IPv4-pohjaisesta verkosta IPv6-protokollan käyttöön tulisi tehdä mahdollisimman selväpiirteiseksi. Epäselvyyksien välttämiseksi päätettiin kehittää maailmanlaajuinen ohjelma, jota IPv6-iplementaatioiden kehittäjät ja laitevalmistajat tulisi seurata.

Jotta IPv6-implementaatio tulee verifioiduksi IPv6 Ready Logo -ohjelman puitteissa, täytyy sen osoittaa IPv6-spesifikaatioiden mukainen yhdenmukaisuus ja yhteentoimivuus muiden IPv6-implementaatioiden kesken. TAHI-projekti alkoi julkaista IPv6 Ready Logo -ohjelman mukaisia testejä vuonna 2003. Nämä testit ovat TAHI ct-paketista, mutta testien määrää on vähennetty IPv6 Ready Logo -ohjelman vaiheiden mukaisiksi.

IPv6 Ready Logo -ohjelma sisältää 3 vaihetta, joista vaihe 3 takaa implementaatiolle täyden IPv6-tuen. Vaihe 1 julkaistiin vuonna 2003 ja se sisältää IPv6-protokollan perus toiminnallisuuden. Vaihe 2 julkaistiin vuonna 2004 ja vaihe 3 on vielä työn alla.

Jotta implementaatio saavuttaa vaiheen *IPv6 Ready Logo Phase 1*, on sen suoritettava kaksi testiä onnistuneesti. Testissä vaaditaan, että implementaatio tukee 100%:sti IPv6 perus spesifikaatioita ja sen täytyy olla yhteentoimiva muiden järjestelmien kesken. Vaihe 1 sisältää yhdenmukaisuustestit seuraaville IPv6-spesifikaatioille:

- IPv6 Specification
- ICMPv6 for IPv6 Specification
- Neighbor Discovery
- IPv6 Stateless Address Autoconfigration

TAHI:n julkaisema vaiheen 1 testipaketti on yksinkertaistettu versio ct-paketin testeistä. Vaiheen 1 testeissä täytyy ottaa huomioon seuraavia asioita:

- Vaihe 1 käy läpi IPv6 perusspesifikaation toiminnot, jotka on määritelty MUST-vaatimuksin
- Tietyt toiminnot, kuten IPsec, on jätetty seuraaviin vaiheisiin, vaikka olisivat MUST-vaatimuksena IPv6 spesifikaatiossa

- Vain Ethernet-liitäntä otetaan huomioon
- Ylempien protokollakerroksien toimintaa ei oteta huomioon
- Vain päätelaiteet kohteena huomioidaan

Tällä hetkellä vaiheen 1 hyväksyttyjen lista on yli 270 toteutusta pitkä [62]. Nämä implementaatiot ovat luokiteltu vaiheessa 1 kolmeen kategoriaan: päätelaitteet, reitittimet ja erikoislaitteet. Vaiheessa 1 on kuitenkin syytä huomata, että vaihe 1 testit tukevat ainoastaan päätelaitteita. Listalle hyväksytyt reitittimet ovat siis IPv6-spesifikaation mukaisia päätelaite ominaisuuksien perusteella, jotka voivat olla erilaisia verrattuna reitittimet spesifikaatioihin.

Jotta implementaatio saavuttaa vaiheen *IPv6 Ready Logo Phase 2* on sen suoritettava tarkemmat yhdenmukaisuus- ja yhteentoimivuustestit kuin vaiheessa 1. Vaihe 2 sisältää yhdenmukaisuustestit seuraaville IPv6-spesifikaatioille:

- IPv6 core Protocol
- IPsec
- MIPv6
- MLD
- Transition

ja näistä MLD- ja Transition -testit ovat vielä kehitysvaiheessa. IPv6 core protocol, IPsec ja MIPv6 testit kattavat spesifikaationsa täydellisesti. Tällä hetkellä vaiheeseen 2 hyväksyttyjen lista on yli 60 implementaatiota pitkä cite [62].

4.3 Muita verifiointiympäristöjä

TAHI on tällä hetkellä kattavin ja helpoiten lähestyttävissä oleva IPv6-protokollan testausympäristö. Maailmalla on olemassa myös muita testausohjelmia IPv6-verifiointiin. TAHI:n ohella IPv6 Ready Logo -ohjelmaan kuuluu Pohjois-Amerikan IPv6-työryhmän (*North American IPv6 Task Force* (NAv6TF)) New Hampshiren yliopiston yhteentoimivuuslaboratoriossa sijaitseva IPv6-testausympäristö, jota kutsutaan lyhenteellä *UNH-IOL* [63]. Tämä kuuluu osana yhteen maailman suurimmista IPv6-verkoista, *Moonv6*-verkkoon [64].

UNH-IOL sisältää testejä, jotka ovat keskittyneet tällä hetkellä IPv6:n perus spesifikaatioihin sekä protokollista IPSec, OSPFv3, PIM-SM, RIPng, BGP, DHCPv6 ja

MLD:n. Testit kattavat yksittäisen sovelluksen säännönmukaisuuden IPv6-testauksen, että eri laiteympäristöjen yhteentoimivuuden. UNH-IOL on kaupallinen tuote ja vaatii rekisteröitymistä, toisin kuin kaikille ilmainen ja vapaasti internetistä ladattava TAHI-ohjelmisto.

Maailmalla järjestetään aika ajoin myös virallisia IPv6-testaustapahtumia, joissa kehittävät voivat käydä testaamassa omia sovelluksiaan ja hakemassa tietoa IPv6-testauksen kehityksestä. Testaustapahtumista saa tietoa testausohjelmistojen internetsivuilta, IPv6-protokollaan liittyviltä foorumeilta ja muilta alaan liittyviltä sivustoilta.

5 Yhteenveto

Internetin suosio on kasvanut räjähdysmäisesti viimeisen viidentoista vuoden aikana. Myös Internetin ja IP-protokollan käyttö mitä erilaisimmissa päätelaitteissa on lisääntynyt sekä tekniikan siirtyminen erilaiseen langattomiin verkkoihin tulee kasvamaan tulevaisuudessa. Liikkuvien päätelaitteiden merkitys tietoliikenteessä kasvaa jatkuvasti ja tämä kehitys on myös otettava huomioon suunniteltaessa tulevaisuuden tietoliikenneprotokollia.

Langattomalla tiedonsiirrolla varustetut liikkuvat päätelaitteet eroavat perinteisistä kiinteistä päätelaitteista monella tavalla. Niiden koko on yleensä pieni, jolloin niitä on helppo siirtää paikasta toiseen. Vielä tällä hetkellä myös ilmateitse käytettävissä oleva kaista on usein huomattavasti pienempi kuin kiinteillä laitteilla. Nämä ominaisuudet tuovat uusia haasteita langattomissa verkoissa käytettävien protokollien suunnittelulle.

Internetin standardointiorganisaatio IETF alkoi työryhmissään kehittää uutta protokollaa nykyisen IPv4-protokollan seuraajaksi 90-luvun puolivälissä, koska IPv4 näytti alkavan käydä ongelmalliseksi muun muassa osoiteavaruutensa takia tulevaisuuden haasteita ajatellen. Uuden protokollan nimeksi tuli IPv6 ja sen toivotaan vastaavan tuleviin haasteisiin. IPv6-protokolla tarjoaa riittävän suuren osoiteavaruuden, sisäänrakennetut tietoturva-ominaisuudet ja hyvän laajennettavuuden tulevaisuuden uusille vaatimuksille.

IPv6-protokollaan on myös kehitetty laajennus liikkuvien päätelaitteiden tarpeita ajatellen. Tämä Mobile IPv6 -protokolla tarjoaa liikkuville päätelaitteille mahdollisuuden liikkua IPv6-verkkojen välillä ilman olemassa olevien yhteyksien katkeamista verkon vaihdoksen jälkeen.

Mobile IPv6 -toteutuksia on jo jonkin aikaa kehitetty 2000-luvun alun jälkeen erilaisissa projekteissa ympäri maailmaa. Yliopistot ja yritykset ovat kehittäneet tuotteitaan erilaisille käyttöjärjestelmille ja laitealustoille standardin tai omien tarpeiden mukaisesti. Yhtään täydellistä täysin standardin mukaista ja toimivaa toteutusta ei vielä varmastikaan ole MIPv6-protokollan standardin, joka valmistui vuonna 2004, tuoreuden takia.

Tällä hetkellä kypsimmät MIPv6-toteutukset löytyvät Linux- ja BSD-käyttöjärjestelmien puolelta. Helsingin Teknillisessä korkeakoulussa on kehitetty MIPL-toteutusta (Mobile IPv6 for Linux) Linuxille. Tämä projekti tekee yhteistyötä myös vas-

taavan japanilaisen USAGI-projektin kanssa, jonka MIPv6-koodeista tiettyjä osia on liitetty myös MIPL-koodiin. MIPL on varsin kehittynyt MIPv6-toteutus ja se tukee kokonaisuudessaan MIPv6-ympäristöä sisältäen kotiagentin, liikkuvan päätelaitteen ja liikennöintikumppanin tuen. MIPL tukee myös reitin optimointia ja IPsec-salausta liikkuvan päätelaitteen ja kotiagentin välillä.

BSD-käyttöjärjestelmille kohdistettua MIPv6-toteutusta kehitetään japanissa yliopistojen ja yritysten yhteistyönä KAME-projektin puitteissa. KAME-projekti on itse asiassa kehittänyt IPv6- ja IPsec-pinot BSD-käyttöjärjestelmille. KAME-toteutukseen on lisäksi liitetty MIPv6-toiminnallisuus SHISA-projektilta, joten BSD-käyttöjärjestelmille kehitettyä MIPv6-toteutusta kutsutaan myös KAME/SHISA-protokollapinoksi. Tämä toteutus on sisältää myös MIPv6-toiminnallisuuden kokonaisuudessaan kuten MIPL-toteutuskin. Lisäksi se tukee reitin optimointia ja IPsec-salausta liikkuvan päätelaitteen ja kotiagentin välillä, aivan kuten MIPL.

Cisco ainoana suurena reititinvalmistajana tarjoaa uusimpiin IOS-versioihinsa MIPv6-tukea kotiagentin toiminnallisuuteen. Ciscon toteutuksesta kuitenkin puuttuu IPsec tietoturva ominaisuudet Mobile IPv6 -ympäristöön. Myös muilla laitevalmistajilla on osittaisia Mobile IPv6 -toteutuksia, mutta nämä voivat olla vielä kehitysasteella tai muuten toimimattomia esimerkiksi MIPL:n ja KAME:n kanssa.

Tämän tutkielman tekoa varten laboratorioon pystytettiin Mobile IPv6 -toiminnallisuuden omaava tietoverkko. Tietoverkossa tehtiin testejä, joissa tutkittiin muun muassa eri MIPv6-toteutusten yleistä toimivuutta ja viiveitä yhteysvastuun vaihdon aikana. Käytössä oli MIPL, KAME ja Ciscon MIPv6-toteutukset, joihin oli asetettu päälle reitin optimointi liikkuvan päätelaitteen ja liikennöintikumppanin välille.

Tehtyjen testien mukaan MIPL ja KAME osoittivat toimivan kohtalaisen hyvin saman ja eri toteutusten kesken. Kotiagentin ja liikennöintikumppanin toimivuuden kesken eri järjestelmillä ei ollut suuria eroja, mutta liikkuvan päätelaitteen toteutukset tuntuivat tarjoavan hieman erilaisen suorituskyvyn. MIPL:n suorityskyky yhteysvastuun vaihdon aikana rekisteröintiviiveiden osalta oli näiden testien mukaan huomattavasti parempi kuin KAME:n. Osaltaan tämä selittyy sillä, että toteutusten välillä on eroja rekisteröintiviestien lähettämisen kesken. KAME on enemmän konservatiivisempi standardin mukaan ja lähettää HoTI/CoTI-viestit vasta kun BAck-viesti on saapunut kotiagentilta liikkuvalle päätelaitteelle. MIPL:ssä tätä vaihetta on hieman optimoitu, jolloin siinä liikkuva päätelaite lähettää BU- ja CoTI-viestit samaan aikaan rekisteröintivaiheessa. Lisäksi KAME-toteutuksessa liikkuva päätelaite lähettää jokaisessa yhteysvastuun vaihdossa HoTI-viestin, kun taas MIPL lähettää sen vain ensimmäisessä yhteysvastuun vaihdon aikana. Tästä johtuen viiveiden erot olivat useita sekunteja. Cisco toimi kotiagenttina verrattain tasaväkisesti,

tai jopa nopeammin, MIPL:n ja KAME:n kanssa. Testeistä voi lukea tarkemmin kappaleesta 3.

Testejä tehtiin aluksi myös salaamalla rekisteröintiviestit kotiagentin ja liikennöintikumppanin välillä IPsec:n avulla. Näistä testeistä luovuttiin, koska liikkuvan päätelaitteen ollessa KAME ja kotiagentin MIPL, HoTI/HoT-viestien tunnelointia ei saatu yrityksistä huolimatta toimimaan ensimmäisen yhteysvastuun vaihdon jälkeen. Luultavasti ongelma johtui MIPL:n kotiagentin toteutuksesta, koska ympäristön ollessa kokonaan KAME, rekisteröinnit onnistuivat ilman ongelmia. Ongelma saattaa johtua siitä, että liikkuvan päätelaitteen ollessa KAME lähetetään siinä HoTI-viesti jokaisen yhteysvastuun vaihdon aikana, jolloin MIPL-kotiagentti saattaa sekoittaa tai laskea väärin ensimmäisen ja toisen HoTI/HoT-viestin rekisteröintien avaimet ja tämän takia hylätä kaikki ensimmäisen yhteysvastuun vaihdon jälkeen tulevat HoT-viestit.

5.1 Jatkotutkimus

Mobile IPv6 -toteutukset eivät vielä tällä hetkellä ole valmiita käyttöjärjestelmien julkaisuversioiden mukana tarjottavaksi. Koska Mobile IPv6 -protokolla standardi on vielä melko nuori ja tietyt osa-alueet vielä kehitteillä eivät nämä olemassa olevat toteutukset vielä ole tarpeeksi kypsiä toimiakseen laajassa käytössä.

Tässä testissä tehtyjen tutkimusten mukaan MIPL ja KAME ovat osoittaneet kumminkin olevan jo varsin hyvällä mallilla kehityksen kärkipäässä. Suurempia ongelmia ei näiden toiminnassa Mobile IPv6 -ympäristöissä ole, mutta rekisteröintivaiheen viiveet ovat vielä melko suuret reaaliaikasovellusten sujuvaan toimintaan nähden. Myös reititinvalmistaja Cisco on kehittänyt varsin vakaan ja kotiagenttina hyvin toimivan toteutuksen.

Jatkotutkimuksissa voitaisiin keskittyä tutkimaan Mobile IPv6 -protokollan rinnalla yhteysvastuun vaihdon rekisteröintivaiheen optimointiin liittyviä toteutuksia. Näitä ovat muun muassa FHMIPv6, HMIPv6 sekä FFHMIPv6. Tehdyissä tutkimuksissa näistä onkin saatu hyviä kokemuksia ja niiden on todettu olevan hyviä vaihtoehtoja MIPv6 rekisteröintivaiheen viiveiden minimoimista ajatellen.

Laboratoriossa tehdyissä testeissä ilmeni ongelmia myös IPsec:n toimivuudessa KAME:n ja MIPL:n kesken. Tulevissa tutkimuksissa olisi selvitettävä, mistä johtuu testeissä ilmennyt ongelma MIPL-kotiagentin hylätessä HoT-viestit ensimmäisen yhteysvastuun vaihdon jälkeen.

Viitteet

[1] Gustafsson, E., Jonsson, A., *Always best connected*, Wireless Communications, IEEE 2003.

[2] Berezdivin, R., Breinig, R., Topp, R., *Next-Generation Wireless Communications Concepts and Technologies*, Communications Magazine, IEEE 2002.

[3] Chiussi, F. M., Khotimsky, D. A., Krishnan, S., *Mobility Management in Third-Generation All-IP Networks*, Communications Magazine, IEEE 2002.

[4] Zhen, L., Wenan, Z., Junde, S., Chunping, H., *Consideration and Research Issues For the Future Generation of Mobile Communication*, Canadian Conference on Electrical and Computer Engineering 2002.

[5] Evans, B. G., Baughan, K., *Visions of 4G*, Electronics & Communication Engineering Journal 2000.

[6] Varshney, U., Jain, R., *Issues in Emerging 4G Wireless Networks*, Computer 2001.

[7] Wisely, D., Aghvami, H., Gwyn, S.L., Zahariadis, T., Manner, J., Gazis, V., Houssos, N., Alonistioti, N.,*Transparent IP radio access for next-generation mobile networks*, Wireless Communications, IEEE 2003.

[8] Hui, S. Y., Yeung, K. H., *Challenges in the Migration to 4G Mobile Systems*, Communications Magazine, IEEE 2003.

[9] Frattasi, S., Fathi, H., Fitzek, F.H.P., Prasad, R., Katz, M.D., *Defining 4G Technology from the Users Perspective*, Network, IEEE 2006.

[10] 6bone-projekti, *6bone-projekti*,
<URL: `http://www.6bone.net/`>, 1.1.2007.

[11] Hexago-projekti, *Hexago-projekti*,
<URL: `http://www.hexago.com/`>, 1.1.2007.

[12] go6-projekti, *go6-projekti*,
<URL: `http://www.go6.net/`>, 1.1.2007.

[13] IPv6 Forum, *IPv6 Forum*,
<URL: http://www.ipv6forum.com/>, 1.1.2007.

[14] IPv6 Portal, *IPv6 Portal*,
<URL: http://www.ipv6tf.org/>, 1.1.2007.

[15] IPv6 Promotion Council, *IPv6 Promotion Council*,
<URL: http://www.v6pc.jp/en/index.phtml>, 1.1.2007.

[16] IPv6 Style, *IPv6 Style*,
<URL: http://www.ipv6style.jp/en/>, 1.1.2007.

[17] 6net-projekti, *6net-projekti*,
<URL: http://www.6net.org/>, 1.1.2007.

[18] 6DISS-projekti, *6DISS-projekti*,
<URL: http://www.6diss.org/>, 1.1.2007.

[19] Postel, J., *Internet protocol*, RFC 791, IETF 1981.

[20] Bradney, S., Mankin, A., *The Recommendation for the IP Next Generation Protocol*, RFC 1752, IETF 1995.

[21] Kent, S., Atkinson, R., *Security Architecture for the Internet Protocol*, RFC 2401, IETF 1998.

[22] Harkins, D., Carrel, D., *The Internet Key Exchange (IKE)*, RFC 2409, IETF 1998.

[23] Kent, S., Atkinson, R., *IP Authentication Header*, RFC 2402, IETF 1998.

[24] Kent, S., Atkinson, R., *IP Encapsulating Security Protocol (ESP)*, RFC 2406, IETF 1998.

[25] Deering, S., Hinden, R., *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460, IETF 1998.

[26] Narten, T., Nordmark, E., Simpson, W., *Neighbor discovery for IP version 6*, RFC 2461, IETF 1998.

[27] Hinden, R., Thaler, D., *IPv6 Host-to-Router Load Sharing*, RFC 4311, IETF 2005.

[28] Thomson, S., Narten, T., *IPv6 Stateless Address Autoconfiguration*, RFC 2462, IETF 1998.

[29] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., Carney, M., *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, RFC 3315, IETF 2003.

[30] Conta, A., Deering, S., Gupta, M.,*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC 4443, IETF 2006.

[31] Conta, A., Deering, S., *Generic Packet Tunneling in IPv6 Specification*, RFC 2473, IETF 1998.

[32] Johnson, D., Perkins, C., Arkko, J., *Mobility Support in IPv6*, RFC 3775, IETF 2004.

[33] Arkko, J., Devarapalli, V., Dupont, F., *Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents*, RFC 3776, IETF 2004.

[34] Koodli, R., *Fast Handovers for Mobile IPv6*, RFC 4068, IETF 2005.

[35] Hinden, R., Deering, S., *IP Version 6 Addressing Architecture*, RFC 4291, IETF 2006.

[36] Perkins, C., *IP Mobility Support for IPv4*, RFC 3344, IETF 2002.

[37] Manner, J., Kojo, M., *Mobility Related Terminology*, RFC 3753, IETF 2004.

[38] Soliman, H., *Mobile IPv6, Mobility in a Wireless Internet*, Addison Wesley 2004.

[39] Dunmore, M., *Mobile IPv6 Handovers: Performance Analysis and Evaluation*, 6net 2005, <URL:`http://www.6net.org/publications/deliverables/D4.1.3v2.pdf`>, 1.4.2007.

[40] Dunmore, M., *An IPv6 Deployment Guide*, 6net 2005, <URL:`http://www.6net.org/book/deployment-guide.pdf`>, 1.4.2007.

[41] Vogt, C., *A Comprehensive Delay Analysis for Reactive and Proactive Handoffs with Mobile IPv6 Route Optimization*, Institute of Telematics, University of Karlsruhe 2006.

[42] Montavont, N., Noel, T., *Analysis and Evaluation of Mobile IPv6 Handovers over Wireless LAN*, Mobile Networks and Applications: Journal on special issues on Mobility of Systems, Users, Data and Computing 2003, s. 643–653.

[43] Chakravorty, R., Vidales, P., Subramanian, K., Pratt, I., Crowcroft, J., *Performance Issues with Vertical Handovers - Experiences from GPRS Cellular and WLAN Hot-spots Integration*, Proceedings of the 2nd IEEE Annual Pervasive Computing and Communications Conference 2004, s. 155–164.

[44] Bernaschi, M., Cacace, F., Pescape, A., Za, S., *Analysis and experimentation over heterogeneous wireless networks*, Proceedings of the International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities 2005, s. 182–191.

[45] Ruiz, J., Vallejo, A., Abella, J., *IPv6 Conformance and Interoperability Testing*, Proceedings of the 10th IEEE Symposium on Computers and Communications 2005, s. 83–88.

[46] Nokia, *Nokian kotisivut*,
<URL: `http://www.nokia.com`>, 15.1.2007.

[47] Symbian, *Symbian kotisivut*,
<URL: `http://www.symbian.com`>, 15.1.2007.

[48] IPv6-toteutuksia, *IPv6 Implementations*,
<URL: `http://www.ipv6.org/impl/index.html`>, 18.1.2007.

[49] Cisco Systems, *Cisco-verkkolaitevalmistajan kotisivut*,
<URL: `http://www.cisco.com`>, 18.1.2007.

[50] Microsoft Internet Protocol Version 6, *Microsoftin IPv6-sivut*,
<URL: `http://www.microsoft.com/technet/network/ipv6/default.mspx`>, 18.1.2007.

[51] Microsoft Research IPv6 Implementation, *Microsoftin IPv6-tutkimuksen kotisivut*,
<URL: `http://research.microsoft.com/msripv6/`>, 18.1.2007.

[52] Treck Incorporation, *Treck Incorporation*,
<URL: `http://www.elmic.com/`>, 18.1.2007.

[53] Linux IPv6 HOWTO, *Linux IPv6 ohjesivut*,
<URL: `http://tldp.org/HOWTO/Linux+IPv6-HOWTO/`>, 20.12.2006.

[54] Mobile IPv6 for Linux, *MIPL-projektin www-sivut*,
<URL: `http://www.mipl.mediapoli.com/`>, 1.7.2006.

[55] USAGI(UniverSAl playGround for Ipv6) Project, *USAGI-projektin www-sivut*, <URL: http://www.linux-ipv6.org/>, 20.12.2006.

[56] USAGI MIPv6 Project, *UMIP-projektin www-sivut*, <URL: http://www.linux-ipv6.org/umip-0.1-ann.html>, 20.12.2006.

[57] KAME Project, *KAME-projektin www-sivut*, <URL: http://www.kame.net/>, 1.6.2006.

[58] KAME Newsletter, *KAME-projektin Newsletter-sivut*, <URL: http://www.kame.net/newsletter/>, 10.4.2007.

[59] SHISA Project, *SHISA-projektin www-sivut*, <URL: http://www.mobileip.jp/>, 1.7.2006.

[60] WIDE Project, *WIDE-projektin www-sivut*, <URL: http://www.wide.ad.jp/>, 1.7.2006.

[61] TAHI Project, *TAHI-projektin www-sivut*, <URL: http://www.tahi.org/>, 1.8.2006.

[62] IPv6 Ready Logo Program, *IPv6 Ready Logo -ohjelman kotisivut*, <URL: http://www.ipv6ready.org/>, 12.10.2006.

[63] University of New Hampshire, InterOperability Laboratory, *UNH-IOL*, <URL: http://www.iol.unh.edu/services/testing/ipv6/>, 1.3.2007.

[64] Moonv6-projekti, *Moonv6-projektin kotisivut*, <URL: http://www.moonv6.org/>, 1.3.2007.

[65] Naval Research Laboratory PROTEAN Research Group, *MGEN the Multi-Generator Toolset, NIST Net network emulator*, <URL: http://mgen.pf.itd.nrl.navy.mil/, viitattu 13.12.2006.

[66] Ethereal-verkkoanalysaattori, *Ethereal-ohjelman kotisivut*, <URL: http://www.ethereal.com/, viitattu 15.2.2007.

A Laitteiden konfiguraatiot - MIPL

Kotiagentin konfiguraatiot

```
########################################
# Mobile IPv6 config file: Home Agent #
#                                      #
# filename: /etc/mip6d.conf            #
########################################

NodeConfig HA;

## If set to > 0, will not detach from tty
DebugLevel 10;

## List of interfaces where we serve as Home Agent
Interface "eth2";

## IPsec config
UseMnHaIPsec enabled;

## Key management Mobility disabled
#KeyMngMobCapability disabled;

IPsecPolicySet {
    HomeAgentAddress 3ffe:5::1;
    HomeAddress 3ffe:5::2/64;

    IPsecPolicy HomeRegBinding UseESP 1 2;
    IPsecPolicy MobPfxDisc UseESP 3 4;
    IPsecPolicy TunnelMh UseESP 5 6;

}
```

```
IPsecPolicy vaihtoehdot:
- HomeRegBinding: käytetään kuljetusmoodissa BU/BA:lle
- Mh: Käytetään kuljetusmoodissa kaikille MH-paketeille
- TunnelMh: kaikki MH-paketit tunneloidaan ilman erottelua
            (BU/BA, HoTI/HoT, CoTI/CoT,...)
- TunnelHomeTesting: vain RR-prosessissa käytetyttyille
                     tunneloiduille MH-paketeille
                     (HoTI/HoT, CoTI/CoT)
- TunnelPayload: Hyötykuorman tunnelointiin
- ICMP: kaikille ICMP-paketeille
- MobPfxDisc: vain MPS/MPA:lle
- Any: kaikille paketeille


#########################################################
# Router Advertisement Daemon config file: Home Agent   #
#                                                       #
# filename: /etc/radvd.conf                             #
#########################################################

interface eth2
{
    AdvSendAdvert on;

    MinRtrAdvInterval 0.05;
    MaxRtrAdvInterval 3;

    AdvIntervalOpt off;

    AdvHomeAgentFlag on;
    HomeAgentLifetime 10000;
    HomeAgentPreference 20;
    AdvHomeAgentInfo on;

    prefix 3ffe:5::1/64
    {
        AdvRouterAddr on;
        AdvOnLink on;
```

```
        AdvAutonomous on;
        AdvPreferredLifetime 10000;
        AdvValidLifetime 20000;
    };
};

################################################
# Security Association config file: Home Agent #
#                                              #
# (MN:lle sama tiedosto)                       #
#                                              #
# filename: /etc/sa.conf                       #
# 3ffe:5::1 is home address of MN              #
# 3ffe:5::2 is address of HA                   #
################################################

# MN -> HA transport SA for BU
add 3ffe:5::2 3ffe:5::1 esp 2000
    -u 1
    -m transport
    -E des-cbc "MNHAtran"
    -A hmac-sha1 "MNHAtransportMNHAtra" ;

# HA -> MN transport SA for BA
add 3ffe:5::1 3ffe:5::2 esp 2001
    -u 2
    -m transport
    -E des-cbc "HAMNtran"
    -A hmac-sha1 "HAMNtransportHAMNtra";

# MN -> HA transport SA for MPS
add 3ffe:5::2 3ffe:5::1 esp 2002
    -u 3
    -m transport
    -E des-cbc "MNHAtran"
    -A hmac-sha1 "MNHAtransportMNHAtra";
```

```
# HA -> MN transport SA for MPA
add 3ffe:5::1 3ffe:5::2 esp 2003
    -u 4
    -m transport
    -E des-cbc "HAMNtran"
    -A hmac-sha1 "HAMNtransportHAMNtra";

# MN -> HA tunnel SA for HoTI
add 3ffe:5::2 3ffe:5::1 esp 2004
    -u 5
    -m tunnel
    -E des-cbc "MNHAtunn"
    -A hmac-sha1 "MNHAtunnelMNHAtunnel";

# HA -> MN tunnel SA for HoT
add 3ffe:5::1 3ffe:5::2 esp 2005
    -u 6
    -m tunnel
    -E des-cbc "HAMNtunn"
    -A hmac-sha1 "HAMNtunnelHAMNtunnel";


##########################################
# Liitäntöjen konfiguraatiot: Home Agent #
#                                        #
# filename: set_interfaces.sh            #
##########################################
#!/bin/sh

## eth0
ip link set dev eth0 down
ip link set dev eth0 up

ip -6 addr add 3ffe:1::2/64 dev eth0
ip -6 route add default via 3ffe:1::1
ip -6 route add 3ffe:2::/64 via 3ffe:1::1
ip -6 route add 3ffe:3::/64 via 3ffe:1::1
ip -6 route add 3ffe:4::/64 via 3ffe:1::1
```

```
## eth2
ip link set dev eth2 down
ip link set dev eth2 up

ip -6 addr add 3ffe:5::1/64 dev eth2
########################################


##############################
# Proc-asetukset: Home Agent #
#                            #
# filename: set_proc.sh      #
##############################
#!/bin/sh

echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
echo 0 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

Liikkuvan päätelaitteen konfiguraatiot

```
########################################
# Mobile IPv6 config file: Mobile Node #
#                                      #
# filename: /etc/mip6d.conf            #
########################################

#NodeConfig MN;
NodeConfig CN;

## If set to > 0, will not detach from tty
DebugLevel 10;
#DebugLevel 0;

#MnDiscardHaParamProb enabled;
```

```
#SendMobPfxSols disabled;

## Support route optimization with other MNs
DoRouteOptimizationCN enabled;

## Use route optimization with CNs
#DoRouteOptimizationMN enabled;


## Interface "eth0" - Ethernet
Interface "eth0" {
    MnIfPreference 1;
}

MnHomeLink "eth0" {
    HomeAgentAddress 3ffe:5::1;
    HomeAddress 3ffe:5::2/64;
}

## Interface "eth1" - Wlan
Interface "eth1" {
    MnIfPreference 2;
}

## IPsec config
#UseMnHaIPsec enabled;

#KeyMngMobCapability disabled;

IPsecPolicySet {
    HomeAgentAddress 3ffe:5::1;
    HomeAddress 3ffe:5::2/64;

    IPsecPolicy HomeRegBinding UseESP 1 2;
    IPsecPolicy MobPfxDisc UseESP 3;
    IPsecPolicy TunnelMh UseESP;
}
```

```
##############################
# Proc-asetukset: Mobile Node #
#                             #
# filename: set_proc.sh       #
##############################
#!/bin/sh

echo 0 > /proc/sys/net/ipv6/conf/all/forwarding
echo 1 > /proc/sys/net/ipv6/conf/all/autoconf
echo 1 > /proc/sys/net/ipv6/conf/all/accept_ra
echo 1 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

Liikennöintikumppanin konfiguraatiot

```
##############################################
# Mobile IPv6 config file: Correspondent Node #
#                                             #
# filename: /etc/mip6d.conf                   #
##############################################

NodeConfig CN;
DoRouteOptimizationCN enabled;


##############################################
# Liitännän konfiguraatio: Correspondent Node #
#                                             #
# filename: set_interfaces.sh                 #
##############################################
#!/bin/sh

## eth0
ip link set dev eth0 down
ip link set dev eth0 up

ip -6 addr add 3ffe:4::2/64 dev eth0
```

```
ip -6 route add default via 3ffe:4::1
#####################################


#######################################
# Proc-asetukset: Correspondent Node #
#                                     #
# filename: set_proc.sh               #
#######################################
#!/bin/sh

echo 0 > /proc/sys/net/ipv6/conf/eth0/forwarding
echo 0 > /proc/sys/net/ipv6/conf/eth0/autoconf
echo 0 > /proc/sys/net/ipv6/conf/eth0/accept_ra
echo 1 > /proc/sys/net/ipv6/conf/eth0/accept_redirects
```

Reitittimen ”R” konfiguraatiot

```
###############################################
# Liitäntöjen konfiguraatiot: Reititin "R" #
#                                             #
# filename: set_interfaces.sh                 #
###############################################
#!/bin/sh

## eth1
ip link set dev eth1 down
ip link set dev eth1 up

ip -6 addr add 3ffe:1::1/64 dev eth1
ip -6 route add 3ffe:5::/64 via 3ffe:1::2

## eth3
ip link set dev eth3 down
ip link set dev eth3 up

ip -6 addr add 3ffe:2::1/64 dev eth3
```

```
ip -6 route add 3ffe:6::/64 via 3ffe:2::2

## eth4
ip link set dev eth4 down
ip link set dev eth4 up

ip -6 addr add 3ffe:3::1/64 dev eth4
ip -6 route add 3ffe:7::/64 via 3ffe:3::2

## eth5
ip link set dev eth5 down
ip link set dev eth5 up

ip -6 addr add 3ffe:4::1/64 dev eth5
########################################


##################################
# Proc-asetukset:  Reititin "R" #
#                               #
# filename: set_proc.sh         #
##################################
#!/bin/sh

echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
echo 0 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

Reitittimen ”AR1” konfiguraatio

```
##############################################
# Liitäntöjen konfiguraatiot: Reititin "AR1" #
#                                            #
# filename: set_interfaces.sh                #
##############################################
#!/bin/sh
```

```
## eth2
ip link set dev eth2 down
ip link set dev eth2 up

ip -6 addr add 3ffe:2::2/64 dev eth2
ip -6 route add default via 3ffe:2::1
ip -6 route add 3ffe:1::/64 via 3ffe:2::1
ip -6 route add 3ffe:3::/64 via 3ffe:2::1
ip -6 route add 3ffe:4::/64 via 3ffe:2::1

## eth1
ip link set dev eth1 down
ip link set dev eth1 up

ip -6 addr add 3ffe:6::1/64 dev eth1
########################################


##################################
# Proc-asetukset: Reititin "AR1" #
#                                #
# filename: set_proc.sh          #
##################################
#!/bin/sh

echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
echo 0 > /proc/sys/net/ipv6/conf/all/accept_redirects


#############################################################
# Router Advertisement Daemon config file: Reititin "AR1" #
#                                                           #
# filename: /etc/radvd.conf                                 #
#############################################################
```

```
interface eth1
{
    AdvSendAdvert on;
    AdvIntervalOpt on;

    MinRtrAdvInterval 0.05;
    MaxRtrAdvInterval 3;
    AdvHomeAgentFlag off;

    prefix 3ffe:6::0/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

Reitittimen ”AR2” konfiguraatiot

```
################################################
# Liitäntöjen konfiguraatiot: Reititin "AR2" #
#                                              #
# filename: set_interfaces.sh                  #
################################################
#!/bin/sh

## eth1
ip link set dev eth1 down
ip link set dev eth1 up

ip -6 addr add 3ffe:3::2/64 dev eth1
ip -6 route add default via 3ffe:3::1
ip -6 route add 3ffe:1::/64 via 3ffe:3::1
ip -6 route add 3ffe:2::/64 via 3ffe:3::1
ip -6 route add 3ffe:4::/64 via 3ffe:3::1
```

```
## eth2
ip link set dev eth2 down
ip link set dev eth2 up

ip -6 addr add 3ffe:7::1/64 dev eth2
########################################


############################################################
# Router Advertisement Daemon config file: Reititin "AR2" #
#                                                          #
# filename: /etc/radvd.conf                                #
############################################################

interface eth2
{
    AdvSendAdvert on;
    AdvIntervalOpt on;

    MinRtrAdvInterval 0.05;
    MaxRtrAdvInterval 3;
    AdvHomeAgentFlag off;

    prefix 3ffe:7::0/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

B Testaus-skripti - MIPL MN

```
##########################################
# Suorituskykytesteissä käytetty skripti #
#                                        #
# filename: run_test.sh                  #
##########################################
#!/bin/sh

set -e
export PATH

########### Interface settings ###########

# proc variable
proc=/proc/sys/net/ipv6/conf

eth0()
{
  echo "=> Setting eth0 up."
  #ip link set dev eth0 up
  ifconfig eth0 up
  while [ ! -d ${proc}/eth0 ]; do : ; done
  echo 0 > ${proc}/eth0/forwarding
  echo 1 > ${proc}/eth0/autoconf
  echo 1 > ${proc}/eth0/accept_ra
  echo 1 > ${proc}/eth0/accept_redirects
}

eth1_AP1()
{
  echo "=> Connecting eth1 to AP1."
  iwconfig eth1 essid AP1
}
```

```
eth1()
{
  echo "=> Setting eth1 up."
  #ip link set dev eth1 up
  ifconfig eth1 up
  while [ ! -d ${proc}/eth1 ]; do : ; done
  echo 0 > ${proc}/eth1/forwarding
  echo 1 > ${proc}/eth1/autoconf
  echo 1 > ${proc}/eth1/accept_ra
  echo 1 > ${proc}/eth1/accept_redirects
}

eth1_AP2()
{
  echo "=> Connecting eth1 to AP2."
  iwconfig eth1 essid AP2
}

################ Handover ################

mipl()
{
  echo "=> Starting mip6d."
  #echo "-> Configuring IPsec-keys..."
  #echo ""
  #if [ -f /etc/sa.conf ]; then
  #  setkey -F
  #  setkey -FP
  #  setkey -f /etc/sa.conf
  #else
  #  echo "Can't find sa.conf. Can't run setkey."
  #  echo "Add /etc/sa.conf."
  #  exit 1
  #fi
  echo "-> Starting mip6d-daemon."
  echo ""
```

```
  #if [ -f /etc/mip6d.conf ]; then
    mip6d -c /etc/mip6d.conf
  else
    echo "Can't find /etc/mip6d.conf. Can't run mip6d."
    echo "Add /etc/mip6d.conf."
    exit 1
  fi
  sleep 5
}

teth()
{
  echo "=> Starting tethereal."
  tethereal -t a -i any -w testdump &
}

start()
{
  echo "Echoing \"Start\" to testdump"
  echo "Start" > datedump &
  date +%k:%M:%S:%N >> timedump &
  echo ""
}

ho1()
{
  echo "=> Setting eth 0 down"
  #ip link set dev eth0 down
  ifconfig eth0 down
  echo "...Doing HO to AP1."
  echo ""
}

ho2()
{
  echo "=> Doing HO to AP2."
  eth1_AP2
```

```
  echo ""
}


home()
{
  echo "=> Returning home"
  #ip link set dev eth0 up
  #ip link set dev eth1 down
  ifconfig eth0 up
  ifconfig eth1 down
  echo ""
}

kl()
{
  echo "=> Killing tethereal & mip6d -processes"
  killall tethereal
  #killall tcpdump
  killall mip6d
  killall wget
}

############### Test round ###############

t1=15

echo ""
echo "Performing MIPv6 test."
echo "======================"
echo ""

teth
eth0
eth1_AP1
eth1
mipl
```

```
#wget -b http://[3ffe:4::2]/file.iso &
#sleep 3
start
sleep $t1
ho1
sleep $t1
ho2
sleep $t1
home
sleep $t1
kl

echo "Test run done."

rm file.iso*
rm wget-log*

exit 0

##########################################
```

C Suorituskykytestit: MN – MIPL

Skenaario: MIPL-MIPL-MIPL

Skenaario: MIPL-MIPL-MIPL		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.045741	MD *	1.058207	MD *	2.246911
BA	1.012734	BU => HA	1.001066	BU => HA	0.003902
Pkt. uusi verkko => HoA	0.000859	CoTI	0.001103	BA	0.108754
HoTI	0.004408	CoT	0.006260	BU => CN	0.004013
CoTI	0.005598	BA	0.000717	Pkt. uusi verkko => HoA	0.005138
HoT	0.002432	BU => CN	0.002340		
CoT	0.001358	Pkt. uusi verkko => NCoA	0.011233		
BU => CN	0.003059				
Pkt. uusi verkko => CoA	0.006909				
Kokonaisviive	**1.083098**		**2.080926**		**2.368718**

Skenaario: MIPL-MIPL-MIPL		Testi: CN => MN Ping 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.004579	MD *	4.045504	MD *	3.856846
BA	1.012013	BU => HA	1.001827	BU => HA	1.001702
Pkt. uusi verkko => HoA	0.006108	CoTI	0.001111	BA	0.061949
HoTI	0.005138	CoT	0.006567	BU => CN	0.043612
CoTI	0.001741	BA	0.000619	Pkt. uusi verkko => HoA	0.005215
HoT	0.002729	BU => CN	0.003195		
CoT	0.001396	Pkt. uusi verkko => NCoA	0.008422		
BU => CN	0.005597				
Pkt. uusi verkko => CoA	0.004973				
Kokonaisviive	**1.044274**		**5.067245**		**4.969324**

Skenaario: MIPL-MIPL-MIPL		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.009963	MD *	1.246384	MD *	3.351342
BA	1.013321	BU => HA	1.000914	BU => HA	0.003512
HoTI	0.004739	CoTI	0.001534	BA	0.099065
CoTI	0.001225	CoT	0.006967	BU => CN	0.002593
Pkt. uusi verkko => HoA	0.000664	BA	0.000638	Pkt. uusi verkko => HoA	0.002295
HoT	0.002424	BU => CN	0.003117		
CoT	0.000601	Pkt. uusi verkko => NCoA	0.016380		
BU => CN	0.001374				
Pkt. uusi verkko => CoA	0.007272				
Kokonaisviive	**1.041583**		**2.275934**		**3.458807**

* MD sisältää liikkeen havaitsemisen vaiheessa lähetettäviä viestejä, kuten RA-, MLD- ja NS-viestit.

Skenaario: MIPL-MIPL-MIPL		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.005964	MD *	3.548291	MD *	2.379431
Pkt. uusi verkko => HoA	1.012568	BU => HA	1.001057	BU => HA	0.459942
CoTI	0.003007	CoTI	0.001523	BA	0.018612
CoT	0.009481	CoT	0.006716	BU => CN	0.006720
BU2 => HA	0.488384	BA	0.000658	Pkt. uusi verkko => HoA	0.003791
BA	0.010829	BU => CN	0.003137		
HoTI	4.494830	Pkt. uusi verkko => NCoA	0.008909		
HoT	0.009314				
BU => CN	0.013110				
Pkt. uusi verkko => CoA	0.030765				
Kokonaisviive	**6.078252**		**4.570291**		**2.868496**

Skenaario: MIPL-MIPL-MIPL		Testi: MN <= CN, TCP tiedostonsiirto HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.007189	MD *	1.444498	MD *	2.968220
BA	1.086339	BU => HA	1.001065	BU => HA	0.003891
Pkt. uusi verkko => HoA	2.089187	CoTI	0.001534	BA	0.061720
HoTI	0.079934	CoT	0.006886	BU => CN	0.108234
CoTI	0.001269	BA	0.000849	Pkt. uusi verkko => HoA	0.279187
HoT	0.017081	BU => CN	0.003197		
CoT	0.000216	Pkt. uusi verkko => NCoA	0.889189		
BU => CN	0.089770				
Pkt. uusi verkko => CoA	0.016580				
Kokonaisviive	**3.387565**		**3.347218**		**3.421252**

MIPL-MIPL-MIPL IPsec-suojauksella

Skenaario: MIPL-MIPL-MIPL / IPsec suojaus		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.004928	MD *	4.265101	MD *	2.967928
Pkt. uusi verkko => HoA	1.013644	BU => HA	1.001249	BU => HA	0.001220
BA	0.001431	CoTI	0.001755	BA	0.095163
HoTI	0.020335	CoT	0.006784	BU => CN	0.009945
CoTI	0.001672	BA	0.001077	Pkt. uusi verkko => HoA	0.005227
HoT	0.009711	BU => CN	0.002164		
CoT	0.004567	Pkt. uusi verkko => NCoA	0.013657		
BU => CN	0.004591				
Pkt. uusi verkko => CoA	0.033016				
Kokonaisviive	**1.093895**		**5.291787**		**3.079483**

Skenaario: MIPL-MIPL-KAME

Skenaario: MIPL-MIPL-KAME	Testi: CN => MN Ping 100 x 128 tavua / s				
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.008232	MD *	1.008058	MD *	1.210311
BA	1.011197	BU => HA	0.802695	BU => HA	1.001492
Pkt. uusi verkko => HoA	0.002112	CoTI	0.001105	BA	0.062217
HoTI	0.007322	CoT	0.006359	BU => CN	0.128484
CoTI	0.002255	BA	0.000762	Pkt. uusi verkko => HoA	0.007086
HoT	0.004894	BU => CN	0.002258		
CoT	0.000740	Pkt. uusi verkko => NCoA	0.008654		
BU => CN	0.007379				
Pkt. uusi verkko => CoA	0.007325				
Kokonaisviive	**1.051456**		**1.829891**		**2.409590**

Skenaario: MIPL-MIPL-KAME	Testi: CN => MN UDP 100 x 128 tavua / s				
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.044793	MD *	3.473308	MD *	2.336981
BA	1.011207	BU => HA	1.000975	BU => HA	0.000100
Pkt. uusi verkko => HoA	0.005953	CoTI	0.001108	BA	0.016433
HoTI	0.003915	CoT	0.006924	BU => CN	0.080186
CoTI	0.001126	BA	0.000683	Pkt. uusi verkko => HoA	0.004004
HoT	0.002589	BU => CN	0.002429		
CoT	0.000595	Pkt. uusi verkko => NCoA	0.014607		
BU => CN	0.001510				
Pkt. uusi verkko => CoA	0.009882				
Kokonaisviive	**1.081570**		**4.500034**		**2.437704**

Skenaario: MIPL-MIPL-KAME	Testi: CN => MN UDP 1000 x 128 tavua / s				
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.011148	MD *	1.775391	MD *	1.364773
Pkt. uusi verkko => HoA	1.012232	BU => HA	1.001092	BU => HA	0.856717
CoTI	0.002897	CoTI	0.001547	BA	0.016691
CoT	0.002968	CoT	0.006741	BU => CN	0.007700
BU2 => HA	0.490038	BA	0.000819	Pkt. uusi verkko => HoA	0.012337
BA	0.023004	BU => CN	0.003092		
HoTI	4.489329	Pkt. uusi verkko => NCoA	0.007954		
HoT	0.012455				
BU => CN	0.011430				
Pkt. uusi verkko => CoA	0.000591				
Kokonaisviive	**6.056092**		**2.796636**		**2.258218**

Skenaario: MIPL-MIPL-KAME	Testi: MN <= CN TCP Tiedostonsiirto HTTP-palvelimelta				
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.925954	MD *	1.525596	MD *	1.001971
BA	1.013590	BU => HA	1.001015	BU => HA	0.001027
Pkt. uusi verkko => HoA	3.101660	CoTI	0.001099	BA	0.098327
HoTI	0.005526	CoT	0.006541	BU => CN	0.002611
CoTI	0.002594	BA	0.000710	Pkt. uusi verkko => HoA	0.440381
HoT	0.005982	BU => CN	0.006592		
CoT	0.001771	Pkt. uusi verkko => NCoA	0.315275		
BU => CN	0.002897				
Pkt. uusi verkko => CoA	0.020114				
Kokonaisviive	**5.080088**		**2.856828**		**1.544317**

Skenaario: MIPL-KAME-MIPL

Skenaario: MIPL-KAME-MIPL	Testi: CN => MN Ping 100 x 128 tavua / s				
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.025333	MD *	3.144603	MD *	2.380646
BA	1.007692	BU => HA	1.001242	BU => HA	0.000177
Pkt. uusi verkko => HoA	0.000985	CoTI	0.001134	BA	0.003074
HoTI	0.003696	BA	0.005708	BU => CN	0.128890
CoTI	0.001117	CoT	0.000880	Pkt. uusi verkko => HoA	0.007778
HoT	0.002485	BU => CN	0.001566		
CoT	0.000842	Pkt. uusi verkko => NCoA	0.014858		
BU => CN	0.001121				
Pkt. uusi verkko => CoA	0.016669				
Kokonaisaika	**1.059940**		**4.169991**		**2.520565**

Skenaario: MIPL-KAME-MIPL	Testi: CN => MN Ping 1000 x 128 tavua / s				
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.006240	MD *	1.057538	MD *	0.306800
BA	1.003210	BU => HA	1.001079	BU => HA	0.000250
Pkt. uusi verkko => HoA	0.001137	CoTI	0.001523	BA	0.003052
HoTI	0.004631	BA	0.006309	BU => CN	0.004219
CoTI	0.001116	CoT	0.000711	Pkt. uusi verkko => NCoA	0.009807
HoT	0.002475	BU => CN	0.001757		
CoT	0.000895	Pkt. uusi verkko => NCoA	0.014719		
BU => CN	0.002520				
Pkt. uusi verkko => CoA	0.014191				
Kokonaisaika	**1.036415**		**2.083636**		**0.324128**

Skenaario: MIPL-KAME-MIPL	Testi: CN => MN UDP 100 x 128 tavua / s				
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.010625	MD *	4.759723	MD *	2.116954
BA	1.008000	BU => HA	0.357216	BU => HA	0.000426
Pkt. uusi verkko => HoA	0.003159	CoTI	0.001111	BA	0.003003
HoTI	0.003710	CoT	0.006579	BU => CN	0.002342
CoTI	0.001596	BA	0.000770	Pkt. uusi verkko => HoA	0.008220
HoT	0.004579	BU => CN	0.002699		
CoT	0.000858	Pkt. uusi verkko => NCoA	0.012237		
BU => CN	0.004312				
Pkt. uusi verkko => CoA	0.012705				
Kokonaisaika	**1.049544**		**5.140335**		**2.130945**

Skenaario: MIPL-KAME-MIPL	Testi: CN => MN UDP 1000 x 128 tavua / s				
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.039580	MD *	1.005678	MD *	2.217522
Pkt. uusi verkko => HoA	1.001294	BU => HA	0.950729	BU => HA	0.000094
CoTI	0.007971	CoTI	0.001108	BA	0.003127
CoT	0.011191	BA	0.006556	BU => CN	0.022417
BU2 => HA	0.500933	CoT	0.000569	Pkt. uusi verkko => NCoA	0.004263
BA	0.009240	BU => CN	0.001880		
HoTI	4.509201	Pkt. uusi verkko => NCoA	0.007087		
HoT	0.008827				
BU => CN	0.016528				
Pkt. uusi verkko => CoA	0.000568				
Kokonaisaika	**6.105333**		**1.973607**		**2.247423**

Skenaario: MIPL-KAME-MIPL		Testi: MN <= CN TCP tiedostonsiirto HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.021200	MD *	2.031621	MD *	1.856757
BA	1.004191	BU => HA	0.767003	BU => HA	0.000477
Pkt. uusi verkko => HoA	2.038115	CoTI	0.001531	BA	0.003128
HoTI	0.070429	BA	0.006931	BU => CN	0.029863
CoTI	0.006806	CoT	0.000571	Pkt. uusi verkko => NCoA	1.660712
HoT	0.002174	BU => CN	0.001688		
CoT	0.001340	Pkt. uusi verkko => NCoA	0.606613		
BU => CN	0.024659				
Pkt. uusi verkko => CoA	0.019561				
Kokonaisaika	**3.188475**		**3.415958**		**3.550937**

Skenaario: MIPL-KAME-KAME

Skenaario: MIPL-KAME-KAME		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.009649	MD *	1.008872	MD *	1.178956
BA	1.008575	BU => HA	0.753753	BU => HA	0.003647
Pkt. uusi verkko => HoA	0.004637	CoTI	0.001109	BA	0.002974
HoTI	0.006051	BA	0.005826	BU => CN	0.002550
CoTI	0.001126	CoT	0.000741	Pkt. uusi verkko => HoA	0.010344
HoT	0.004051	BU => CN	0.001797		
CoT	0.000890	Pkt. uusi verkko => NCoA	0.007790		
BU => CN	0.007238				
Pkt. uusi verkko => CoA	0.009120				
Kokonaisviive	**1.051337**		**1.779888**		**1.198471**

Skenaario: MIPL-KAME-KAME		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.041223	MD *	3.807296	MD *	0.882082
BA	1.000445	BU => HA	0.064528	BU => HA	0.317778
Pkt. uusi verkko => HoA	0.000927	CoTI	0.001013	BA	0.003380
HoTI	0.005201	BA	0.006515	BU => CN	0.033945
CoTI	0.001129	CoT	0.000832	Pkt. uusi verkko => HoA	0.020297
HoT	0.002474	BU => CN	0.001748		
CoT	0.000722	Pkt. uusi verkko => NCoA	0.013662		
BU => CN	0.001439				
Pkt. uusi verkko => CoA	0.007907				
Kokonaisviive	**1.061467**		**3.895594**		**1.257482**

Skenaario: MIPL-KAME-KAME		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.024157	MD *	1.818210	MD *	1.450796
BA	1.003224	BU => HA	1.001068	BU => HA	0.982715
Pkt. uusi verkko => HoA	0.018053	CoTI	0.001537	BA	0.003184
HoTI	0.008305	BA	0.006370	BU => CN	0.002689
CoTI	0.004584	CoT	0.000777	Pkt. uusi verkko => HoA	0.019811
HoT	0.006506	BU => CN	0.001557		
CoT	0.018054	Pkt. uusi verkko => NCoA	0.024436		
BU => CN	0.005943				
Pkt. uusi verkko => CoA	0.015823				
Kokonaisviive	**1.104649**		**2.853955**		**2.459195**

Skenaario: MIPL-KAME-KAME		Testi: MN <= CN TCP Tiedoston lataaminen HTTP-palvelimelta.			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.196461	MD *	4.683123	MD *	2.946120
BA	0.999060	BU => HA	0.471742	BU => HA	0.003478
Pkt. uusi verkko => HoA	3.840342	CoTI	0.001116	BA	0.002958
HoTI	0.140023	BA	0.006021	BU => CN	0.002991
HoT	0.039292	CoT	0.000735	Pkt. uusi verkko => HoA	0.402411
CoTI	0.005017	BU => CN	0.001643		
CoT	0.021628	Pkt. uusi verkko => NCoA	0.355632		
BU => CN	0.045639				
Pkt. uusi verkko => CoA	0.017226				
Kokonaisviive	**5.304688**		**5.520012**		**3.357958**

Skenaario: MIPL-Cisco-MIPL

Skenaario: MIPL-Cisco-MIPL		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.046190	MD *	1.019300	MD *	0.806650
BA	1.010155	BU => HA	0.730784	BU => HA	0.000449
Pkt. uusi verkko => HoA	0.000857	CoTI	0.001114	BA	0.004471
HoTI	0.004288	BA	0.006694	BU => CN	0.002822
CoTI	0.001158	CoT	0.000803	Pkt. uusi verkko => HoA	0.016067
CoT	0.003973	BU => CN	0.001803		
HoT	0.000792	Pkt. uusi verkko => NCoA	0.030146		
BU => CN	0.010064				
Pkt. uusi verkko => CoA	0.081574				
Kokonaisaika	**1.159051**		**1.790644**		**0.830459**

Skenaario: MIPL-Cisco-MIPL		Testi: CN => MN Ping 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.004121	MD *	3.980581	MD *	1.344046
BA	1.010272	BU => HA	1.001093	BU => HA	0.000636
Pkt. uusi verkko => HoA	0.001002	CoTI	0.001116	BA	0.004436
HoTI	0.004600	BA	0.006401	BU => CN	0.002679
CoTI	0.001255	CoT	0.000839	Pkt. uusi verkko => HoA	0.008664
CoT	0.003582	BU => CN	0.001645		
HoT	0.001511	Pkt. uusi verkko => NCoA	0.022345		
BU => CN	0.001617				
Pkt. uusi verkko => CoA	0.010489				
Kokonaisaika	**1.038449**		**5.014020**		**1.360461**

Skenaario: MIPL-Cisco-MIPL		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.010534	MD *	4.399569	MD *	1.159003
BA	1.010193	BU => HA	1.001088	BU => HA	0.000425
Pkt. uusi verkko => HoA	0.000849	CoTI	0.001117	BA	0.004495
HoTI	0.005290	BA	0.006891	BU => CN	0.005287
CoTI	0.001114	CoT	0.000606	Pkt. uusi verkko => HoA	0.012167
CoT	0.003142	BU => CN	0.001726		
HoT	0.002618	Pkt. uusi verkko => NCoA	0.009377		
BU => CN	0.001702				
Pkt. uusi verkko => CoA	0.016465				
Kokonaisaika	**1.051907**		**5.420374**		**1.181377**

Skenaario: MIPL-Cisco-MIPL		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover - HN => FN1		**Handover - FN1 => FN2**		**Handover - FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU. => HA	0.006609	MD *	4.294087	MD *	0.985314
Pkt. uusi verkko => HoA	1.009573	BU => HA	0.268477	BU => HA	0.002663
CoTI	0.007841	CoTI	0.001543	BA	0.004516
CoT	0.013480	BA	0.006275	BU => CN	0.002884
BU2	0.486496	CoT	0.000815	Pkt. uusi verkko => HoA	0.007983
BA	0.010035	BU => CN	0.001726		
HoTI	4.489854	Pkt. uusi verkko => NCoA	0.008361		
HoT	0.014407				
BU => CN	0.025047				
Pkt. uusi verkko => CoA	0.019046				
Kokonaisaika	**6.082388**		**4.581284**		**1.003360**

Skenaario: MIPL-Cisco-MIPL		Testi: MN <= CN TCP Tiedoston lataus HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.008509	MD *	3.852955	MD *	0.912864
BA	1.005603	BU => HA	1.001094	BU => HA	0.001729
Pkt. uusi verkko => HoA	0.506911	CoTI	0.001249	BA	0.004470
HoTI	0.005203	BA	0.005962	BU => CN	0.002615
CoTI	0.002390	CoT	0.000620	Pkt. uusi verkko => HoA	0.737706
HoT	0.005657	BU => CN	0.001632		
CoT	0.000447	Pkt. uusi verkko => NCoA	2.204319		
BU => CN	0.001752				
Pkt. uusi verkko => CoA	0.652330				
Kokonaisaika	**2.188802**		**7.067831**		**1.659384**

Skenaario: MIPL-Cisco-KAME

Skenaario: MIPL-Cisco-KAME		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.008673	MD *	1.926971	MD *	1.449519
BA	1.009675	BU => HA	1.001042	BU => HA	0.000498
Pkt. uusi verkko => HoA	0.002456	CoTI	0.001113	BA	0.004471
HoTI	0.052922	CoT	0.006411	BU => CN	0.002577
CoTI	0.001106	BA	0.000590	Pkt. uusi verkko => HoA	0.015389
CoT	0.003727	BU => CN	0.002409		
HoT	0.000959	Pkt. uusi verkko => NCoA	0.011369		
BU => CN	0.001139				
Pkt. uusi verkko => CoA	0.000438				
Kokonaisaika	**1.081095**		**2.949905**		**1.472454**

Skenaario: MIPL-Cisco-KAME		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.024089	MD *	1.383518	MD *	0.842148
BA	1.007422	BU => HA	1.001058	BU => HA	0.001139
Pkt. uusi verkko => HoA	0.010149	CoTI	0.001118	BA	0.004550
HoTI	0.002482	BA	0.006182	BU. => CN	0.002116
CoTI	0.001489	CoT	0.000608	Pkt. uusi verkko => HoA	0.011300
CoT	0.002939	BU => CN	0.001719		
HoT	0.001008	Pkt. uusi verkko => NCoA	0.005007		
BU => CN	0.001511				
Pkt. uusi verkko => CoA	0.009365				
Kokonaisaika	**1.060454**		**2.399210**		**0.861253**

Skenaario: MIPL-Cisco-KAME		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	0.019476	MD *	1.026113	MD *	0.704628
BA	1.007355	BU => HA	1.001098	BU => HA	0.000034
Pkt. uusi verkko => HoA	0.010815	CoTI	0.001113	BA	0.005368
HoTI	0.075058	BA	0.006187	BU => CN	0.002686
CoTI	0.003770	CoT	0.000784	Pkt. uusi verkko => HoA	0.013866
CoT	0.024663	BU => CN	0.001741		
HoT	0.000996	Pkt. uusi verkko => NCoA	0.023691		
BU => CN	0.002273				
Pkt. uusi verkko => CoA	0.034793				
Kokonaisaika	**1.179199**		**2.060727**		**0.726582**

Skenaario: MIPL-Cisco-KAME		Testi: MN <= CN TCP Tiedoston lataaminen HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	1.673972	MD *	1.007246	MD *	0.763722
BA	1.005792	BU => HA	0.840850	BU => HA	0.001492
Pkt. uusi verkko => HoA	0.263888	CoTI	0.001103	BA	0.004496
HoTI	0.004857	CoT	0.006304	BU => CN	0.003218
CoTI	0.001156	BA	0.000778	Pkt. uusi verkko => HoA	0.043278
CoT	0.007477	BU => CN	0.002490		
HoT	0.022231	Pkt. uusi verkko => NCoA	0.076071		
BU => CN	0.007961				
Pkt. uusi verkko => CoA	2.622316				
Kokonaisaika	**5.609650**		**1.934842**		**0.816206**

D Suorituskykytestit: MN – KAME

Skenaario: KAME-KAME-KAME

Skenaario: KAME-KAME-KAME		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	5.376249	MD *	1.987251	MD *	0.126043
BA	1.000993	BU => HA	0.003003	BU => HA	0.000128
Pkt. uusi verkko => HoA	0.004866	BA	6.027734	BA	0.001939
HoTI	0.002409	HoTI	0.001431	HoTI	3.021845
CoTI	0.001235	CoTI	0.000337	HoT	0.000902
HoT	0.002258	HoT	0.003032	BU => CN	0.000999
CoT	0.000746	CoT	0.000650	Pkt. uusi verkko => HoA	0.005669
BU => CN	0.001081	BU => CN	0.000996		
Pkt. uusi verkko => CoA	0.012071	Pkt. uusi verkko => NCoA	0.005590		
Kokonaisviive	**6.401908**		**8.030024**		**3.157525**

Skenaario: KAME-KAME-KAME		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	4.342964	MD *	1.367105	MD *	1.052742
Pkt. uusi verkko => HoA	0.999828	BU => HA	0.002930	BU => HA	0.000100
BA	0.002551	BA	6.027835	BA	0.001903
HoTI	0.013981	HoTI	0.001432	HoTI	3.019545
CoTI	0.001845	CoTI	0.000333	HoT	0.000873
HoT	0.001693	HoT	0.003120	BU => CN	0.000975
CoT	0.001320	CoT	0.000748	Pkt. uusi verkko => HoA	0.004793
BU => CN	0.000834	BU => CN	0.001035		
Pkt. uusi verkko => CoA	0.013332	Pkt. uusi verkko => NCoA	0.014729		
Kokonaisviive	**5.378348**		**7.419267**		**4.080931**

Skenaario: KAME-KAME-KAME		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	2.730416	MD *	3.810077	MD *	1.496707
BA	1.000599	BU => HA	0.003057	BU => HA	0.000092
Pkt. uusi verkko => HoA	0.017798	BA	6.028743	BA	0.001962
HoTI	0.000736	HoTI	0.001391	HoTI	3.016391
CoTI	0.002307	CoTI	0.000333	HoT	0.000879
HoT	0.016553	HoT	0.003042	BU => CN	0.000988
CoT	0.000611	CoT	0.000565	Pkt. uusi verkko => HoA	0.014547
BU => CN	0.001828	BU => CN	0.001022		
Pkt. uusi verkko => CoA	0.018472	Pkt. uusi verkko => NCoA	0.017186		
Kokonaisviive	**3.789320**		**9.865416**		**4.531566**

Skenaario: KAME-KAME-KAME		Testi: MN <= CN TCP Tiedoston lataaminen HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	4.939424	MD *	1.513216	MD *	0.669467
BA	1.006744	BU => HA	0.003094	BU => HA	0.000102
Pkt. uusi verkko => HoA	3.781380	BA	6.027585	BA	0.001688
HoTI	0.001579	HoTI	0.001418	HoTI	3.021115
CoTI	0.000318	CoTI	0.000352	HoT	0.000944
HoT	0.010324	HoT	0.003658	BU => CN	0.001002
CoT	0.001344	CoT	0.000667	Pkt. uusi verkko => HoA	1.393435
BU => CN	0.035809	BU => CN	0.000950		
Pkt. uusi verkko => CoA	0.015872	Pkt. uusi verkko => NCoA	2.636488		
Kokonaisviive	**9.792794**		**10.187428**		**5.087953**

Skenaario: KAME-KAME-MIPL

Skenaario: KAME-KAME-MIPL		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	5.924916	MD *	1.183342	MD *	0.465474
BA	0.009153	BU => HA	0.002814	BU => HA	0.000106
Pkt. uusi verkko => HoA	0.999420	BA	6.022538	BA	0.003070
HoTI	0.002352	HoTI	0.001795	HoTI	3.014539
CoTI	0.001234	CoTI	0.000330	HoT	0.000910
HoT	0.002286	HoT	0.003084	BU => CN	0.001102
CoT	0.000863	CoT	0.000712	Pkt. uusi verkko => HoA	0.008964
BU => CN	0.001374	BU => CN	0.001028		
Pkt. uusi verkko => CoA	0.062852	Pkt. uusi verkko => NCoA	0.018819		
Kokonaisviive	**7.004450**		**7.234462**		**3.494165**

Skenaario: KAME-KAME-MIPL		Testi: CN => MN Ping 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	5.018406	MD *	2.369582	MD *	0.447760
BA	0.007964	BU => HA	0.002882	BU => HA	0.000093
Pkt. uusi verkko => HoA	1.002013	BA	6.020301	BA	0.003024
HoTI	0.001108	HoTI	0.001767	HoTI	3.018805
CoTI	0.000421	CoTI	0.000330	HoT	0.000861
HoT	0.004203	HoT	0.003060	BU => CN	0.000950
CoT	0.001596	CoT	0.000741	Pkt. uusi verkko => HoA	0.006534
BU => CN	0.000873	BU => CN	0.000881		
Pkt. uusi verkko => CoA	0.005401	Pkt. uusi verkko => NCoA	0.015860		
Kokonaisviive	**6.041985**		**8.415404**		**3.478027**

Skenaario: KAME-KAME-MIPL		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	2.397012	MD *	1.350628	MD *	1.939751
BA	0.007567	BU => HA	0.001596	BU => HA	0.000098
Pkt. uusi verkko => HoA	0.990632	BA	6.026069	BA	0.003067
HoTI	0.000705	HoTI	0.001756	HoTI	3.006404
CoTI	0.000340	CoTI	0.000332	HoT	0.000926
HoT	0.004007	HoT	0.003080	BU => CN	0.000959
CoT	0.000736	CoT	0.000550	Pkt. uusi verkko => HoA	0.006446
BU => CN	0.000956	BU => CN	0.000927		
Pkt. uusi verkko => CoA	0.016924	Pkt. uusi verkko => NCoA	0.015841		
Kokonaisviive	**3.418879**		**7.400779**		**4.957651**

Skenaario: KAME-KAME-MIPL		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	3.962205	MD *	1.935561	MD *	0.004646
BA	0.008820	BU => HA	0.002897	BU => HA	0.000751
Pkt. uusi verkko => HoA	0.997392	BA	6.028029	BA	0.001980
HoTI	0.002228	HoTI	0.001796	HoTI	7.174309
CoTI	0.001311	CoTI	0.000332	HoT	0.001407
HoT	0.009000	HoT	0.003144	BU => CN	0.000976
CoT	0.000823	CoT	0.000731	Pkt. uusi verkko => HoA	0.002181
BU => CN	0.000880	BU => CN	0.000921		
Pkt. uusi verkko => CoA	0.000404	Pkt. uusi verkko => NCoA	0.006203		
Kokonaisviive	**4.983063**		**7.979614**		**7.186250**

Skenaario: KAME-KAME-MIPL		Testi: MN <= CN TCP Tiedoston lataaminen HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	3.372558	MD *	2.410889	MD *	0.575560
BA	0.007540	BU => HA	0.003063	BU => HA	0.000095
Pkt. uusi verkko => HoA	3.177324	BA	6.018908	BA	0.003084
HoTI	0.001252	HoTI	0.001776	HoTI	3.017418
CoTI	0.000342	CoTI	0.000350	HoT	0.000904
HoT	0.003149	HoT	0.003331	BU => CN	0.000959
CoT	0.000841	CoT	0.000836	Pkt. uusi verkko => HoA	2.948998
BU => CN	0.001045	BU => CN	0.000823		
Pkt. uusi verkko => CoA	6.521328	Pkt. uusi verkko => NCoA	5.169411		
Kokonaisviive	**13.085379**		**13.609387**		**6.547018**

Skenaario: KAME-MIPL-KAME

Skenaario: KAME-MIPL-KAME		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	4.743719	MD *	2.362191	MD *	0.840047
BA	1.012888	BU => HA	0.002876	BU => HA	0.000125
Pkt. uusi verkko => HoA	0.000903	BA	6.019856	BA	0.015143
HoTI	0.008567	HoTI	0.002213	HoTI	3.013130
CoTI	0.001243	CoTI	0.000329	HoT	0.000799
HoT	0.004447	HoT	0.003057	BU => CN	0.000949
CoT	0.001378	CoT	0.000776	Pkt. uusi verkko => HoA	0.005491
BU => CN	0.001721	BU => CN	0.000712		
Pkt. uusi verkko => CoA	0.018852	Pkt. uusi verkko => NCoA	0.007933		
Kokonaisviive	**5.793718**		**8.399943**		**3.875684**

Skenaario: KAME-MIPL-KAME		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	3.033619	MD *	4.382095	MD *	2.758866
BA	1.013046	BU => HA	0.002842	BU => HA	0.000095
Pkt. uusi verkko => HoA	0.000968	BA	6.031155	BA	0.015347
HoTI	0.001869	HoTI	0.001273	HoTI	3.021021
CoTI	0.000324	CoTI	0.000326	HoT	0.000936
HoT	0.003365	HoT	0.003138	BU => CN	0.000856
CoT	0.000840	CoT	0.000802	Pkt. uusi verkko => HoA	0.019213
BU => CN	0.000885	BU => CN	0.000746		
Pkt. uusi verkko => CoA	0.008273	Pkt. uusi verkko => NCoA	0.016837		
Kokonaisviive	**4.063189**		**10.439214**		**5.816334**

Skenaario: KAME-MIPL-KAME		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	5.033272	MD *	0.638207	MD *	1.049932
BA	1.011590	BU => HA	0.003070	BU => HA	0.000090
Pkt. uusi verkko => HoA	0.011413	BA	6.029004	BA	0.015470
HoTI	0.000674	HoTI	0.002228	HoTI	3.016312
CoTI	0.002264	CoTI	0.000480	HoT	0.001200
HoT	0.015972	HoT	0.002892	BU => CN	0.000889
CoT	0.000647	CoT	0.000827	Pkt. uusi verkko => HoA	0.018265
BU => CN	0.001532	BU => CN	0.000715		
Pkt. uusi verkko => CoA	0.019229	Pkt. uusi verkko => NCoA	0.020468		
Kokonaisviive	**6.096593**		**6.697891**		**4.102158**

Skenaario: KAME-MIPL-KAME		Testi: MN <= CN TCP Tiedoston lataaminen HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	5.562451	MD *	1.208543	MD *	3.268068
BA	1.007555	BU => HA	0.002939	BU => HA	0.000102
Pkt. uusi verkko => HoA	0.012196	BA	6.025805	BA	0.015356
HoTI	0.000743	HoTI	0.001282	HoTI	3.023820
CoTI	0.001566	CoTI	0.000324	HoT	0.000903
HoT	0.002400	HoT	0.003751	BU => CN	0.000888
CoT	0.007143	CoT	0.000780	Pkt. uusi verkko => HoA	0.297424
BU => CN	0.035000	BU => CN	0.000783		
Pkt. uusi verkko => CoA	0.015833	Pkt. uusi verkko => NCoA	2.608678		
Kokonaisviive	**6.644887**		**9.852885**		**6.606561**

Skenaario: KAME-MIPL-MIPL

Skenaario: KAME-MIPL-MIPL		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	1.546851	MD *	1.035103	MD *	4.688423
BA	1.007433	BU => HA	0.002924	BU => HA	0.000088
Pkt. uusi verkko => HoA	0.005795	BA	6.029832	BA	0.062459
HoTI	0.001038	HoTI	0.001374	HoTI	3.024140
CoTI	0.000384	CoTI	0.000335	HoT	0.000785
HoT	0.003554	HoT	0.003308	BU => CN	0.000895
CoT	0.000759	CoT	0.000608	Pkt. uusi verkko => HoA	0.009641
BU => CN	0.000956	BU => CN	0.000880		
Pkt. uusi verkko => CoA	0.011005	Pkt. uusi verkko => NCoA	0.015045		
Kokonaisviive	**2.577775**		**7.089409**		**7.786431**

Skenaario: KAME-MIPL-MIPL		Testi: CN => MN Ping 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	2.600098	MD *	1.897557	MD *	
BA	1.009651	BU => HA	0.003063	BU => HA	
Pkt. uusi verkko => HoA	0.006965	BA	6.032770	BA	
HoTI	0.008048	HoTI	0.001545	HoTI	
CoTI	0.000405	CoTI	0.000340	HoT	
HoT	0.003881	HoT	0.003064	BU => CN	
CoT	0.001379	CoT	0.000781	Pkt. uusi verkko => HoA	
BU => CN	0.000783	BU => CN	0.000755	Epäonnistui, koska HoTI-viestin lähetet-	
Pkt. uusi verkko => CoA	0.001642	Pkt. uusi verkko => NCoA	0.004747	tiin Wlan-liitännän kautta	
Kokonaisviive	**3.632852**		**7.944622**		

Skenaario: KAME-MIPL-MIPL		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	3.492289	MD *	1.542435	MD *	0.927273
BA	1.008546	BU => HA	0.001207	BU => HA	0.000093
Pkt. uusi verkko => HoA	0.002867	BA	6.028236	BA	0.108842
HoTI	0.000662	HoTI	0.001449	HoTI	3.020838
CoTI	0.000361	CoTI	0.000328	HoT	0.000796
HoT	0.003414	HoT	0.003059	BU => CN	0.000900
CoT	0.000838	CoT	0.000655	Pkt. uusi verkko => HoA	0.010033
BU => CN	0.000923	BU => CN	0.000863		
Pkt. uusi verkko => CoA	0.014579	Pkt. uusi verkko => NCoA	0.008296		
Kokonaisviive	**4.524479**		**7.586528**		**4.068775**

Skenaario: KAME-MIPL-MIPL		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	5.918450	MD *	3.604497	MD *	0.280548
Pkt. uusi verkko => HoA	1.009913	BU => HA	0.003016	BU => HA	0.000105
BA	0.004150	BA	6.024096	BA	0.014927
HoTI	0.003853	HoTI	0.001264	HoTI	3.014450
CoTI	0.000952	CoTI	0.000326	HoT	0.000817
HoT	0.017934	HoT	0.003768	BU => CN	0.000868
CoT	0.000763	CoT	0.000827	Pkt. uusi verkko => HoA	0.001993
BU => CN	0.001010	BU => CN	0.000728		
Pkt. uusi verkko => CoA	0.012357	Pkt. uusi verkko => NCoA	0.007310		
Kokonaisviive	**6.969382**		**9.645832**		**3.313708**

Skenaario: KAME-MIPL-MIPL		Testi: MN <= CN TCP Tiedoston lataaminen HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	5.159647	MD *	2.101154	MD *	0.752085
BA	1.011827	BU => HA	0.002940	BU => HA	0.000092
Pkt. uusi verkko => HoA	6.626107	BA	6.025797	BA	0.097754
HoTI	0.001550	HoTI	0.001335	HoTI	3.021773
CoTI	0.000315	CoTI	0.000340	HoT	0.000907
HoT	0.006764	HoT	0.003030	BU => CN	0.000894
CoT	0.000471	CoT	0.000650	Pkt. uusi verkko => HoA	3.256925
BU => CN	0.017105	BU => CN	0.000892		
Pkt. uusi verkko => CoA	0.082948	Pkt. uusi verkko => NCoA	6.234124		
Kokonaisviive	**12.906734**		**14.370262**		**7.130430**

Skenaario: KAME-Cisco-KAME

Skenaario: KAME-Cisco-KAME		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	6.276328	MD *	2.261633	MD *	0.119429
BA	1.006140	BU => HA	0.001322	BU => HA	0.000147
Pkt. uusi verkko => HoA	0.001454	BA	6.026326	BA	0.003427
HoTI	0.001113	HoTI	0.001343	HoTI	3.028940
CoTI	0.000413	CoTI	0.000334	HoT	0.003825
CoT	0.004435	CoT	0.003696	BU => CN	0.000927
HoT	0.002354	HoT	0.001386	Pkt. uusi verkko => HoA	0.002526
BU => CN	0.000931	BU => CN	0.000862		
Pkt. uusi verkko => CoA	0.000414	Pkt. uusi verkko => NCoA	0.013133		
Kokonaisviive	**7.293582**		**8.310035**		**3.159221**

Skenaario: KAME-Cisco-KAME		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	2.549674	MD *	2.295296	MD *	0.655854
BA	1.005880	BU => HA	0.003002	BU => HA	0.000141
Pkt. uusi verkko => HoA	0.009969	BA	6.019694	BA	0.003444
HoTI	0.001787	HoTI	0.001322	HoTI	3.028815
CoTI	0.000853	CoTI	0.000339	HoT	0.003728
CoT	0.003135	CoT	0.003895	BU => CN	0.000930
HoT	0.001442	HoT	0.001099	Pkt. uusi verkko => HoA	0.022002
BU => CN	0.000809	BU => CN	0.000832		
Pkt. uusi verkko => CoA	0.010489	Pkt. uusi verkko => NCoA	0.012826		
Kokonaisviive	**3.584038**		**8.338305**		**3.714914**

Skenaario: KAME-Cisco-KAME		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	6.380879	MD *	3.972095	MD *	0.572652
BA	1.006918	BU => HA	0.002885	BU => HA	0.000158
Pkt. uusi verkko => HoA	0.010477	BA	6.023770	BA	0.003434
HoTI	0.000680	HoTI	0.001450	HoTI	3.021211
CoTI	0.000334	CoTI	0.000351	HoT	0.003752
HoT	0.019215	HoT	0.003573	BU => CN	0.000925
CoT	0.000903	CoT	0.001403	Pkt. uusi verkko => HoA	0.023992
BU => CN	0.001481	BU => CN	0.000831		
Pkt. uusi verkko => CoA	0.000419	Pkt. uusi verkko => NCoA	0.025129		
Kokonaisviive	**7.421306**		**10.031487**		**3.626324**

Skenaario: KAME-Cisco-KAME		Testi: MN <= CN TCP Tiedoston lataaminen HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	3.772352	MD *	1.931667	MD *	0.995470
BA	1.007081	BU => HA	0.002823	BU => HA	0.000159
Pkt. uusi verkko => HoA	3.650107	BA	6.033223	BA	0.003482
HoTI	0.001333	HoTI	0.001330	HoTI	3.021334
CoTI	0.000329	CoTI	0.000323	HoT	0.004186
CoT	0.004237	CoT	0.004199	BU => CN	0.000955
HoT	0.001415	HoT	0.001519	Pkt. uusi verkko => HoA	0.188231
BU => CN	0.000884	BU => CN	0.000832		
Pkt. uusi verkko => CoA	0.257796	Pkt. uusi verkko => NCoA	4.430641		
Kokonaisviive	**8.695534**		**12.406557**		**4.213817**

Skenaario: KAME-Cisco-MIPL

Skenaario: KAME-Cisco-MIPL		Testi: CN => MN Ping 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	3.152467	MD *	2.873744	MD *	0.431022
BA	1.005551	BU => HA	0.002963	BU => HA	0.000158
Pkt. uusi verkko => HoA	0.010327	BA	6.020733	BA	0.003445
HoTI	0.001129	HoTI	0.001320	HoTI	3.029020
CoTI	0.000413	CoTI	0.000344	HoT	0.003609
CoT	0.003923	CoT	0.004224	BU => CN	0.000907
HoT	0.000877	HoT	0.001516	Pkt. uusi verkko => HoA	0.016311
BU => CN	0.001014	BU => CN	0.002343		
Pkt. uusi verkko => CoA	0.016325	Pkt. uusi verkko => NCoA	0.005441		
Kokonaisviive	**4.192026**		**8.912628**		**3.484472**

Skenaario: KAME-Cisco-MIPL		Testi: CN => MN Ping 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	2.645586	MD *	1.890820	MD *	
BA	1.002367	BU => HA	0.002874	BU => HA	
Pkt. uusi verkko => HoA	0.007353	BA	6.020906	BA	
HoTI	0.001128	HoTI	0.001380	HoTI	
CoTI	0.000405	CoTI	0.000347	HoT	
CoT	0.004253	CoT	0.003601	BU => CN	
HoT	0.002582	HoT	0.001414	Pkt. uusi verkko => HoA	
BU => CN	0.000882	BU => CN	0.000814	Epäonnistui, koska HoTI-viestin lähetet-	
Pkt. uusi verkko => CoA	0.004956	Pkt. uusi verkko => NCoA	0.017464	tiin Wlan-liitännän kautta.	
Kokonaisviive	3.669512		7.939620		

Skenaario: KAME-Cisco-MIPL		Testi: CN => MN UDP 100 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	5.853591	MD *	3.760907	MD *	0.632010
BA	1.008002	BU => HA	0.002878	BU => HA	0.000174
Pkt. uusi verkko => HoA	0.001145	BA	6.021839	BA	0.003464
HoTI	0.003449	HoTI	0.001341	HoTI	3.019689
CoTI	0.000331	CoTI	0.000332	HoT	0.003768
CoT	0.004040	CoT	0.003616	BU => CN	0.000931
HoT	0.001112	HoT	0.001469	Pkt. uusi verkko => HoA	0.008878
BU => CN	0.000878	BU => CN	0.000824		
Pkt. uusi verkko => CoA	0.004424	Pkt. uusi verkko => NCoA	0.007324		
Kokonaisviive	6.876972		9.800530		3.668914

Skenaario: KAME-Cisco-MIPL		Testi: CN => MN UDP 1000 x 128 tavua / s			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	3.123618	MD *	2.591090	MD *	0.165149
BA	1.006219	BU => HA	0.001412	BU => HA	0.000141
Pkt. uusi verkko => HoA	0.001368	BA	6.026451	BA	0.003456
HoTI	0.006078	HoTI	0.001340	HoTI	3.023753
CoTI	0.000990	CoTI	0.000335	HoT	0.003800
CoT	0.015689	CoT	0.003526	BU => CN	0.000923
HoT	0.004572	HoT	0.001473	Pkt. uusi verkko => HoA	0.004499
BU => CN	0.001511	BU => CN	0.000837		
Pkt. uusi verkko => CoA	0.000388	Pkt. uusi verkko => NCoA	0.005159		
Kokonaisviive	4.160433		8.631623		3.201721

Skenaario: KAME-Cisco-MIPL		Testi: MN <= CN TCP Tiedoston lataaminen HTTP-palvelimelta			
Handover: HN => FN1		**FN1 => FN2**		**FN2 => HN**	
Pkt. vanha verkko	0.000000	Pkt. vanha verkko => OCoA	0.000000	Pkt. vanha verkko => OCoA	0.000000
BU => HA	6.170531	MD *	2.880993	MD *	0.237231
BA	1.004618	BU => HA	0.002906	BU => HA	0.000159
Pkt. uusi verkko => HoA	0.125400	BA	2.008458	BA	0.003379
HoTI	0.001182	HoTI	0.001371	HoTI	3.029148
CoTI	0.001440	CoTI	0.000328	HoT	0.003867
CoT	0.002851	CoT	0.003631	BU => CN	0.000948
HoT	0.000989	HoT	0.001446	Pkt. uusi verkko => HoA	0.042781
BU => CN	0.000929	BU => CN	0.000802		
Pkt. uusi verkko => CoA	0.096381	Pkt. uusi verkko => NCoA	2.175016		
Kokonaisviive	7.404321		7.074951		3.317513