Jyri Turkia

ASIAKASKOHTAINEN NIMIPALVELU

Opinnäytetyö
Tietotekniikan koulutusohjelma

kesäkuu 2012

MIKKELIN AMMATTIKORKEAKOULU
Mikkeli University of Applied Sciences

KUVAILULEHTI

MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences	**Opinnäytetyön päivämäärä** 6.6.2012
Tekijä(t) Jyri Turkia	**Koulutusohjelma ja suuntautuminen** Tietotekniikan koulutusohjelma
Nimike Asiakaskohtainen nimipalvelu	

Tiivistelmä

Opinnäytetyön tarkoituksena on toteuttaa asiakaskohtainen nimipalvelu BIND view-ominaisuutta hyödyntäen sekä arvioida järjestelmän skaalautuvuutta laajempaan ympäristöön. Toteuttamalla pienimuotoisessa ympäristössä halutulla tavalla toimivan nimipalvelimen pystyn dokumentoimaan kuinka sellainen toteutetaan, sekä paremmin arvioimaan toteutuksen ongelmia ja etuja.

BIND nimipalvelimen valitsin työssä käytettäväksi nimipalvelinsovellukseksi siitä syystä, että se on maailman yleisimmin käytetty nimipalvelinsovellus. Koska aihe on itsessään jokseenkin poikkeava, niin siinä käytettävien sovellusten ei tarvitse olla poikkeavia.

Huolimatta konkreettisesta osasta opinnäytetyötä, on työn luonne enemmin selvittävä ja teoreettinen koska työssä kuvattua järjestelmää ei ainakaan suoraan sellaisenaan oteta käyttöön toimestani, toki mikään ei estä jotain muuta tekemästä sitä.

Asiasanat (avainsanat) Verkkotunnus, nimipalvelin, BIND		
Sivumäärä 26	**Kieli** Suomi	**URN**
Huomautus (huomautukset liitteistä)		
Ohjaavan opettajan nimi Selin Jukka	**Opinnäytetyön toimeksiantaja** Mikkelin ammattikorkeakoulu	

KUVAILULEHTI

Date of the bachelor's thesis
6.6.2012

Author(s)
Jyri Turkia

Degree programme and option
Information Technology

Name of the bachelor's thesis
Client specific domain name server

Abstract
The goal of this bachelor's thesis is to create a client specific domain name server and to document how to do that and also review advantages and disadvantages of created system and how well it would scale in to greater environment.

The reason why BIND name server software was chosen to be used in this study is that it is the most used DNS server. The subject of this study is not convenient so the software used need not to be obscure.

Regardless to concrete part of this study the nature of this study is more investigative and theoretical as there is no plans to implement system that was studied.

Subject headings, (keywords)
Domain name system, BIND, hostname

Pages	**Language**	**URN**
26	Finnish	

Remarks, notes on appendices

Tutor
Selin Jukka

Bachelor's thesis assigned by
Mikkeli University of Applied Sciences

SISÄLTÖ

LIITTEET

SANASTO

ACL	Access control list eli lista, jolla määritetään käyttöoikeus jollekin tietylle asialle
ALIVERKKO	Koostuu useista IP osoitteista rajattuna aliverkkomaskilla. Esimerkiksi 192.168.0.0/24 tarkoittaa IPv4 osoitteita väliltä 192.168.0.0 – 192.168.0.255
ARPANET	Yhdysvaltain asevoimien tutkimusorganisaation kehittämä tietoverkko, josta kehittyi Internet
BIND	Internet Systems Consortiumin kehittämä nimipalvelinsovellus
DHCP	Dynamic Host Configuration Protocol, verkkoasetusprotokolla. Yleisimmin käytetään antamaan päätelaitteille IP osoite, oletusreititin ja nimipalvelimet
FICORA	Finnish Communications Regulatory Authority eli Viestintävirasto
FQDN	Täydellisessä muodossa esitetty verkkotunnus, esimerkiksi: www.esimerkki.com.
FTP	File Transfer Protocol, nimensä mukaisesti tiedostojen siirtoprotokolla
HOSTS	Tiedosto, joka sisältää verkko-osoitteen ja tälle määrätyn IP osoitteen
HTTP	The Hypertext Transfer Protocol. Käytetään WWW-sivujen siirtoon palvelimelta selaimelle.
ISC	Internet Systems Consortium on vuonna 1994 perustettu voittoatavoittelematon yritys kehittää Internetille tärkeitä teknologioita, esimerkiksi BIND nimipalvelin ja DHCP
IPv4	32-bittinen osoiteavaruus, jolla laitteet kommunikoivat keskenään Internetissä
IPv6	128-bittinen osoiteavaruus, jolla laitteet kommunikoivat keskenään Internetissä. Kehitetty IPv4:n korvaajaksi
LINUX	Alunperin Linus Torvaldsin kehittämä UNIXin kaltainen käyttöjärjestelmä
NAT	Network address translation mahdollistaa osoitteen muutoksen Internetissä. Mahdollistaa myös usean koneen toimimisen yhden osoitteen kautta
NIMIPALVELIN	Sovellus, joka palauttaa verkkotunnukseen liitettyjä tietoja kyselijälle
OPENSUSE	Novell-yhtiön tukema Linux-käyttöjärjestelmäprojekti. Käytännössä kokeellinen ilmaisversio yrityksen maksullisesta SUSE Linux Enterprise -versiosta
SRI-NIC	Entinen Stanford Research Institute Network Information Center. Hallinnoi aikanaan ARPANETin HOSTS.TXT tiedostoa

SSH	Secure Shell on verkkoprotokolla turvalliseen tiedonsiirtoon. Kehitettiin telnetin korvaajaksi Unixin kaltaisten käyttöjärjestelmien etähallintaa varten
TCP/IP	Transmission Control Protocol / Internet protocol – yleisimmin käytetyt tiedonsiirtoprotokollat Internetissä
UNIX	Vuonna 1969 Bell laboratorioissa kehitetty käyttöjärjestelmä, josta pohjautuu Linux
ZONE	Verkkotunnus ja tämän aliverkkotunnukset, josta nimipalvelin vastaa

1 JOHDANTO

Tutkin opinnäytetyössäni nimipalvelutoteutusta, joka palauttaa halutusti eri vastauksen riippuen kyselyn tekijästä. Opinnäytetyön aloitushetkellä aihe kiehtoi, eikä asiasta juurikaan löytynyt alkusilmäyksellä tutkimuksia tai mainintoja. Tiedossa oli kuitenkin, että jotkin suuret toimijat, kuten Google, käyttää ainankin osana järjestelmiään jonkinlaista toteutusta asiakaskohdennetuista nimipalveluista. Tämä on helppo havaita esimerkiksi tekemällä yksinkertainen nimipalvelukysely osoitteelle www.google.com kahden eri operaattorin yhteyksiä käyttäen. Esimerkiksi käyttäen Linuxissa yleistä hosts komentoa:

```
host www.google.com
```

ja vertaamalla palautuneisiin IP osoitteisiin.

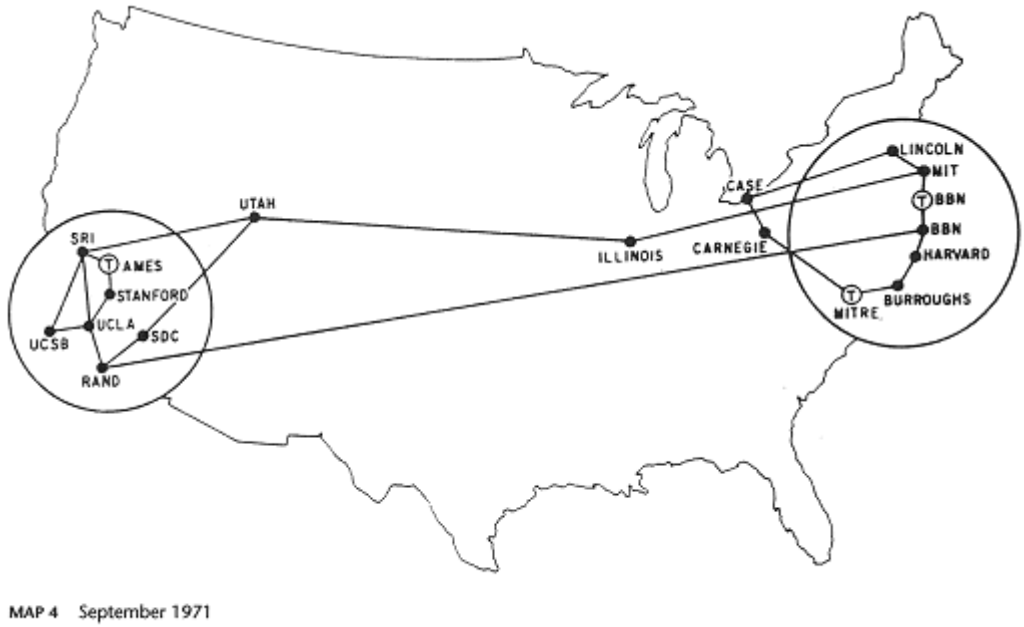
Tiesin jo aloittaessa, että voisin toteuttaa nimipalvelimen niin, että se palauttaa samalle verkkotunnukselle useita eri osoitteita. Tämä on nimipalvelimien perusominaisuuksia. Aikaisempien kokemuksieni perusteella BIND-nimipalvelimen kanssa tiesin myös, että sovellus mahdollistaa eri tiedot verkkotunnukselle sisäänrakennetulla view-ominaisuudella. Henkilökohtaisessa käytössäni olevien verkkotunnusten nimipalvelimia perustaessa, sekä lopulta päivätyöni perusteella, olin jo tutustunut BIND-nimipalvelimeen yksinkertaisessa ympäristössä sekä palveluidentarjoajan mittavammassa ympäristössä. Kuitenkaan kummassakaan ympäristössä en ole tullut törmänneeksi toteutukseen, jollainen on opinnäytetyöni tavoitteena rakentaa.

Tarkoitus on rakentaa yksinkertainen toteutus BIND-nimipalvelinsovelluksella Linux-palvelimelleni, jotta voin esittää, kuinka tälläinen nimipalvelin saadaan toteutettua. Lisäksi, rakennettuani oikeasti toimivan ympäristön, pystyn arvioimaan toteutuksen hyviä ja huonoja puolia. Toteutuksessani käytän Opensuse 12.1 Linux-julkaisua ja tämän tarjoamaa BIND 9.8.1-P1 -nimipalvelinsovellusta. En näe mitään syytä – tai hyötyä – käyttää vaihtoehtoista asennustiedostoa tai kääntää nimipalvelinta lähdekoodista itse, sillä Opensusen tarjoamasta asennuslähteestä asennettava BIND sisältää kaikki tarvitsemani ominaisuudet.

Työssäni käyn ensin läpi Internetin historiaa ja miksi koko nimipalvelujärjestelmä on aikanaan kehitetty. Mielestäni on tärkeää tietää historiaa ja taustoja, jotta voi ymmärtää kokonaisuuden. Taustojen lisäksi luonnollisesti käyn läpi nimipalvelun teoriaa sekä tärkeimmin työhöni vaikuttavia BIND-nimipalvelimen ominaisuuksia. Ilman että ymmärtää nimipalvelun teoriaa, on mahdotonta ymmärtää sovellukseni toimintaa tai tarvetta. Rakentamastani järjestelmästäni kuvaan, kuinka rakensin järjestelmän ja kuinka se toimii. Arvioin työtäni, että mikä siinä on hyvää ja mikä huonoa sekä mitä siitä voisi parantaa. Arvioin myös kuinka se toimisi suuremmassa ympäristössä ja mitä mahdollisia ongelmia silloin ilmenee. Loppuyhteenvedossa arvioin omaa suoritumistani – valmistunutta järjestelmää siltä kannalta, että mitä tekisin toisin, vai onko kenties mitään mitä tekisin muulla tavalla.

2 ARPANET

1960-luvulla Paul Baran visioi tietoverkon, joka kestäisi laajan sotilaallisen hyökkäyksen [1]. Niinpä 1960-luvun lopussa Yhdysvaltojen kehitys- ja tutkimusvirasto Department of Defense Advanced Research Projects Agency (ARPA) alkoi kehittää tietoverkkoa yhdistämään tärkeät kehitysyksiköt ja yritykset [2]. Se nimettiin tietoverkon luoneen viraston mukaan ARPANETiksi. ARPANETin alkuperäinen tavoite oli jakaa kallista laskentatehoa, mutta ARPANET mahdollisti tutkijoiden kommunikoinnin keskenään eri paikoista käsin [2]. ARPANETin toinen tavoite oli mahdollistaa tiedonsiirto vaihtuvia reittejä hyödyntäen, mahdollistaa verkon toimivuus vaikka joku tai useampi reitti lakkaisikin toimimasta [3]. Aluksi ARPANET oli vain neljän tietokoneen verkko [3]. Verkko alkoi hiljalleen kasvamaan ja kuvassa 1 1971 vuoden tilanne ARPANETin levinneisyydestä.



KUVA 1. Kaavio ARPANETistä vuonna 1971 [1]

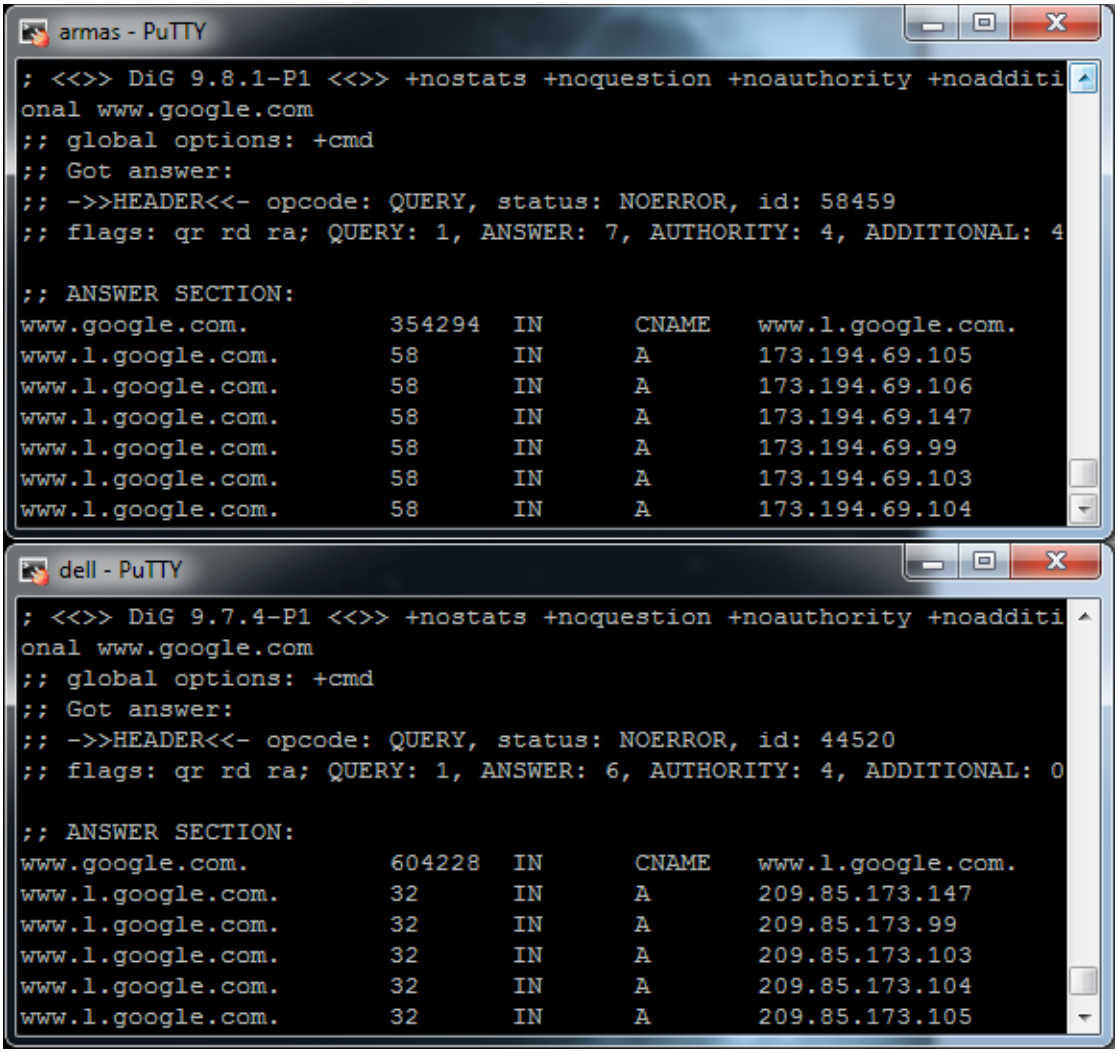
ARPANETissä tietokoneet keskustelevat keskenään numerosarjoilla. Koska ihmiset eivät luontaisesti ole hyviä muistamaan numerosarjoja, niin oli ARPANETiin luotu järjestelmä, jolla numero-osoitteen pystyi kuvaamaan melko vapaasti merkkijonolla. Tällöin osoitteesta tuli ihmiselle paljon muistettavampi. Tietokonetta pystyi kuvaamaan esimerkiksi ihmisen nimellä, ja se saattoi olla koneen merkkimuotoinen osoite. Järjestelmä toimi siten, että jokaisella verkossa olevalla tietokoneella oli tekstimuotoinen HOSTS-tiedosto, jossa tärkeimmät tiedot olivat numeerinen- ja merkkimuotoinen osoite muutosta varten. Nykyisissä käyttöjärjestelmissä on edelleen ARPANETin peruna samankaltainen HOSTS-tiedosto, joka toimii edelleen samalla tavalla. Nykyisin tosin tiedosto on usein tyhjä tai siellä on vain muutama rivi paikallisen osoitteen määrittämiseksi. Osoitteenmuutosta tehtäessä tietokone kuitenkin ensimmäisenä tarkistaa HOSTS-tiedoston, ja mikäli vastausta ei löydy, siirtyy se käyttämään modernimpia tapoja tehdä osoitteenmuutos. [2]

Koska osoitteenmuutos tehtiin paikallisen HOSTS-tiedoston perusteella, oli tiedostoa päivitettävä, mikäli verkossa tapahtui muutoksia. Usein järjestelmänvalvojat olivatkin automatisoineet HOSTS-tiedoston päivityksen SRI-NIC:n (Stanford Research Institute Network Information Center) palvelimilta, joka hallinnoi Arpanetin verkkoa [2].

Virallinen HOSTS-tiedosto oli käsin ylläpidetty ja muutoksia tuli muutamia viikossa. Etenkin 1980-luvulla UNIXin ja TCP/IP:n ansiosta ARPANETin suosion kasvaessa oli tiedoston ylläpito jo hyvin työlästä sekä virheiden määrä kasvoi. HOSTS osoitteenmuunnosjärjestelmä lakkasi toimimasta, mikäli HOSTS-tiedostossa oli sama merkkimuotoinen osoite usealle numeeriselle osoitteelle, ja käsin ylläpidettyyn tiedostoon saattoikin päätyä tälläisiä virheitä. Tämän vuoksi alettiinkin kehittää uutta järjestelmää korvaamaan HOSTS-tiedosto [2]. Vuosien kehityksen tuloksena vuonna 1984 julkaistiikin nimipalvelu (Domain Name System) [4].

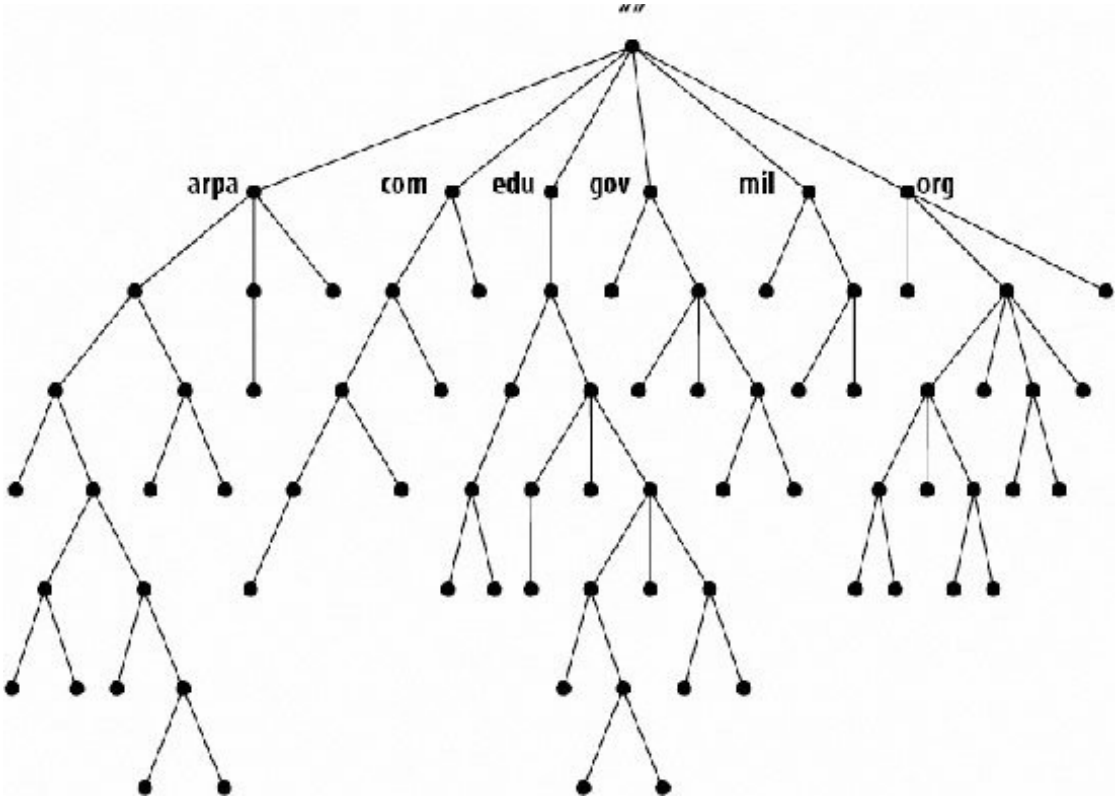
3 NIMIPALVELU

Nimipalvelu kehitettiin siis ihmisien vuoksi, jotta koneille olisi mahdollista antaa muistettavammat nimet kuin pitkät numerosarjat. Kuvassa 2 on esimerkki kahdelta eri tietokoneelta tehdystä nimipalvelukyselystä osoitteelle www.google.com. Tietokoneet sijaitsevat eri operaattoreiden IP-verkoissa. Kuvasta käy ilmi, että osoite www.google.com on vain CNAME viittaus osoitteeseen www.l.google.com. Osoitteella www.l.google.com on varsinaisesti sitten viittaukset IP-osoitteisiin, joissa Googlen hakupalvelimet sijaitsevat. Kuvasta voi huomata, että koneille palautetaan eri IP-osoitteet. Olisi hyvin epätodennäköistä, että tavalliset käyttäjät pystyisivät muistamaan näitä kaikkia Googlen IP-osoitteita. Tarjoamalla useita IP-osoitteita osoitteelle www.google.com mahdollistetaan myös hyvin yksinkertainen kuorman jako, kun käyttäjät jaetaan useammalle palvelimelle. [2]



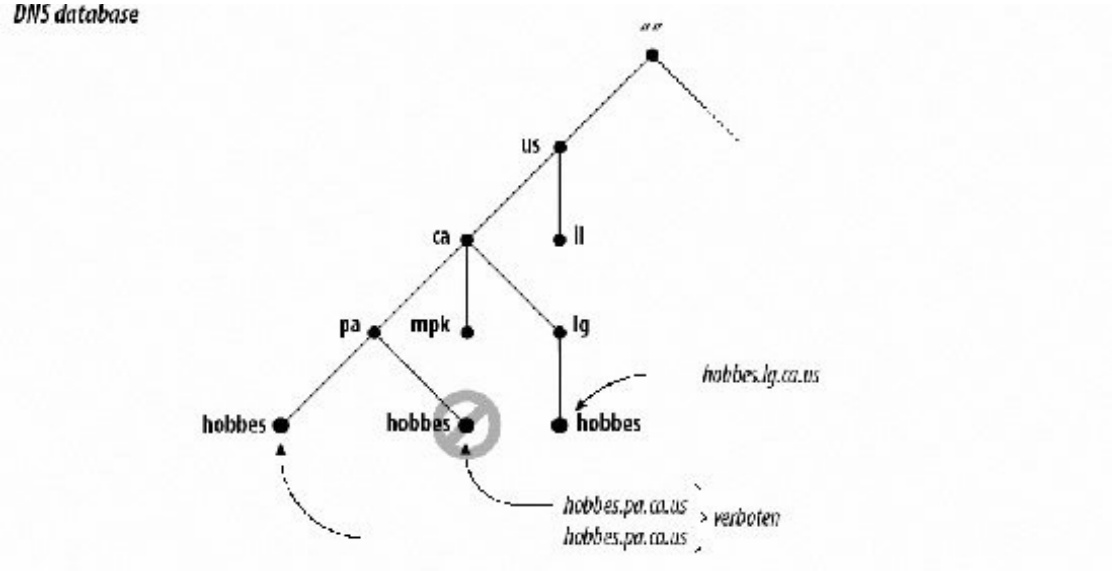
KUVA 2. Www.google.com -kysely

Nimipalvelujärjestelmän nimiavaruus on hierarkinen puurakenne. Nimiavaruuden puuta voi ajatella ylösalaisin käännettynä, sillä juuri yleensä esitetään ylimpänä ja puu kasvaa alaspäin. Ylimmällä tasolla tosiaan on juuri, joka on yhteinen piste kaikille verkkotunnuksille. Järjestelmässä tyhjä eli merkitön verkkotunnus on varattu juurelle. Koska aliverkkotunnukset eli alidomainit erotetaan isännistään pisteellä, esimerkiksi www.esimerkki.fi. on täydellinen verkkotunnus joka kuvaa verkkotunnuksen absoluuttisen pisteen puussa. Asiakasohjelmistot kuten WWW-selaimet eivät vaadi käyttäjiä kirjoittamaan verkkotunnuksien viimeistä pistettä, koska se on kaikissa verkkotunnuksissa ja täten turhaa vaatia käyttäjältä. Kuvassa 3 on esimerkki nimiavaruudesta.



KUVA 3. Nimipalvelun nimiavaruus [2]

Jokainen verkkotunnus on uniikki. Toisinsanoen kahta identtistä verkkotunnusta ei voi olla olemassa. Tämä saattaa hämärtyä, sillä yhdellä verkkotunnuksella voi olla useita tietoja, kuten www.l.google.com esimerkissä verkkotunnukselle on liitetty useita IP-osoitteita. Verkkotunnukset ovat uniikkeja, kun niitä tarkastellaa FQDN- eli täydellisessä muodossa. Esimerkiksi palvelin.testidomain.com. ja palvelin.hurdur.com. ovat uniikkeja, vaikka alimman tason verkkotunnus on sama, kuten kuvassa neljä kuvattu hobbes.lg.ca.us ja hobbes.pa.ca.us.

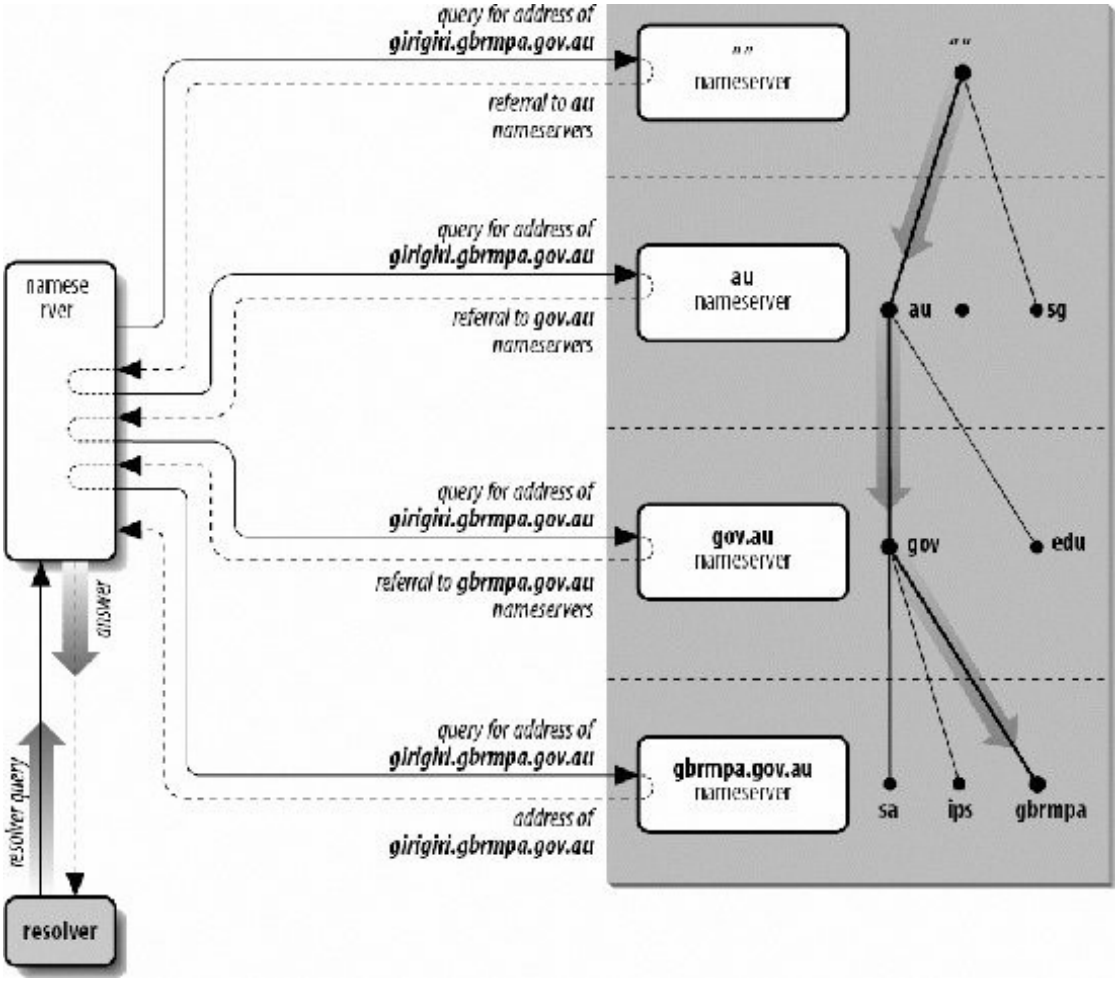


KUVA 4. Uniikit verkkotunnukset [2]

Juuriverkkotunnusta lukuunottamatta jokainen verkkotunnus eli domain on jonkin domainin alidomain. Esimerkiksi fi-verkkotunnus on juuren alidomain ja domain.fi on fi-verkkotunnuksen alidomain [2]. Verkkotunnus voi olla korkeintaan 255 merkkiä pitkä, kuitenkin niin, että alidomainin pituus voi olla korkeintaan 63 merkkiä [5][6]. Verkkotunnukset ovat incasesensitiivisiä eli toisinsanoen ei ole merkitsevää kirjoittaako isoilla vai pienillä kirjaimilla verkkotunnuksen [5]. Verkkotunnuksessa sallittuja merkkejä ovat kirjaimet A-Z ja numerot sekä -. Kansalliset merkit kuten ä, ö ja å ovat olleet mahdollisia vuodesta 2005 lähtien, mutta nämä toteutetaan enkoodaamalla haluttu merkki alkuperäisen standardin mukaiseksi [7]. Esimerkiksi ääkkönen.fi on todellisuudessa xn–kknen-fraa0m.fi [7].

Kun käyttäjä syöttää WWW-selaimensa osoiteriville www.google.com siirtyäkseen tähän osoitteeseen, käynnistää selain nimipalvelukyselyn. Rekursiivinen nimipalvelukysely tehdään verkkoasetuksissa määritetylle nimipalvelimelle. Nimipalvelimelta kysytään käytännössä ”mikä on verkko-osoitteelle liitetty IP-osoite, jos et tiedä niin voitko selvittää sen?”. Nimipalvelin alkaakin selvittämään lopullista vastausta muilta nimipalvelimilta, mikäli tietoa ei löydy tämän omista tiedoistaan. Nimipalvelin aloittaa kysymällä juurinimipalvelimelta ”Mikä on verkko-osoitteen www.google.com IP-osoite tai kuka tietää sen?”. Juurinimipalvelin vastaa com-verkkotunnuksen nimipalvelimella, koska se tietää vastauksen tai tietää ohjata edelleen eteenpäin. Com-verkkotunnuksen nimipalvelin puolestaan vastaa samaan

kysymykseen google.com-verkkotunnuksen nimipalvelimella. Lopulta on löydetty nimipalvelin, joka tietää vastauksen ja palauttaa sen kyselyä tehneelle nimipalvelimelle, joka puolestaan pystyy palauttamaan vastauksen selaimelle, joka alunperin aloitti kyselyn. Kuvassa 5 on kuvattu girigiri.gbrmpa.gov.au verkkotunnuksen osoitteen selvitys. [2]



KUVA 5. Resolution [2]

Kuvassa 5 resolver siis suorittaa rekursiivisen kyselyn omalta nimipalvelimeltaan. Rekursiivinen nimipalvelukysely käytännössä tarkoittaa pyyntöä nimipalvelimelle selvittää kysytyn verkkotunnuksen IP-osoite. Näin yleensä siksi, että resolver on yksinkertainen sovellus, joka ei itse osaa selvittää tarvittua tietoa. Nimipalvelin, jolta resolver kysyy, alkaa sitten tekemään iteratiivisia kyselyitä muilta nimipalvelimilta. Nimipalvelin nimittäin osaa selvittää halutun tiedon muiden nimipalvelimien vihjeiden perusteella. [2]

3.1 BIND view

BIND on käytetyin nimipalvelinsovellus ja siksi erittäin hyvä ohjelmisto opinnäytetyön toteutusta varten [8]. BIND9 ja uudemmat versiot tukevat view-ominaisuutta, jolla nimipalvelin voi esittää eri dataa riippuen mistä kysely tulee. Opinnäytetyössäni teen esimerkkisovelluksen kuinka hyödyntää view-ominaisuutta. View-ominaisuutta hyödyntäen on mahdollista määrittää nimipalvelin niin, että tietty osa verkosta voi esimerkiksi tehdä rekursiivisia nimipalvelukyselyjä palvelimelle ja toiselle osalle verkkoa vastataan vain virheilmoituksella. [9] View määritetään nimipalvelimen asetustiedostossa esimerkiksi seuraavalla tavalla:

```
view "external" {
  match-clients { !localnets; any; };
  recursion no;
  allow-transfer { dns_slaves; };
  zone "example.com" {
    type master;
    file "external/example.com.zone";
  };
};
```

Liitteessä 1 on kommentoitu ja kokonainen esimerkkiasetustiedosto. View on siis asetusmääritys, miten nimipalvelimen tulee toimia määritykseen täsmäävän asiakkaan kohdalla. View-määrityksellä on mahdollista tehdä asiakaskohtaisesti poikkeavia asetuksia, kuten sallia rekursiiviset nimipalvelukyselyt vain tietyille käyttäjille. Vastaavasti view-määrityksellä on mahdollista määrittää zonelle asiakaskohtaisesti eri zonetiedostot.

BIND-nimipalvelimen toinen ominaisuus ACL (access control list) on äärimmäisen tärkeässä osassa view-ominaisuutta käytettäessä, sillä näillä listoilla määritetään minkä view:n määrityksillä kyselyyn vastataan. ACL-määrityksellä määritetään ne osoitteet, jotka täsmäävät kyseiseen listaan. [10]. Listaan voi määrittää yksittäisen IP-osoitteen tai IP-osoite alueen, esimerkiksi 192.168.0.1/24. Merkintätapa 192.168.0.1/24 tarkoittaa IP-verkkoaluetta aliverkon maskilla 255.255.255.0. 24 on aliverkon maskin 1-bittien lukumäärä. Tällä aliverkolla tarkoitetaan osoitteita väliltä 192.168.0.0 – 192.168.0.255. Todellisuudessa käyttökelpoisia osoitteita alueelta eivät ole 192.168.0.0 ja 192.168.0.255, koska ne on varattu verkko-osoitteiksi ja broadcast-osoitteiksi. Listaan on myös mahdollista määrittää osoitteita, johon lista ei täsmää.

Merkintätapa tälle on lisätä huutomerkki ennen osoitetta. [11]. Kuvassa 6 on esimerkki listasta ”good-guys”, joka täsmää kaikkiin verkkoalueisiin, joihin itse palvelin on kyteytynyt, IPv6-verkkoalueeseen 2001:db8:0:1::/64 ja IPv4-verkkoalueeseen 192.168.2.0/24 poislukien osoitteet 192.168.2.0 – 192.168.2.15.

```
acl "good-guys" {
     !192.169.2.5/28; // denies first 16 IPs
     192.168.2/24;    // allows rest of subnet
     localnets;       // allows our network
     2001:db8:0:1::/64; // allows this subnet only
};
```

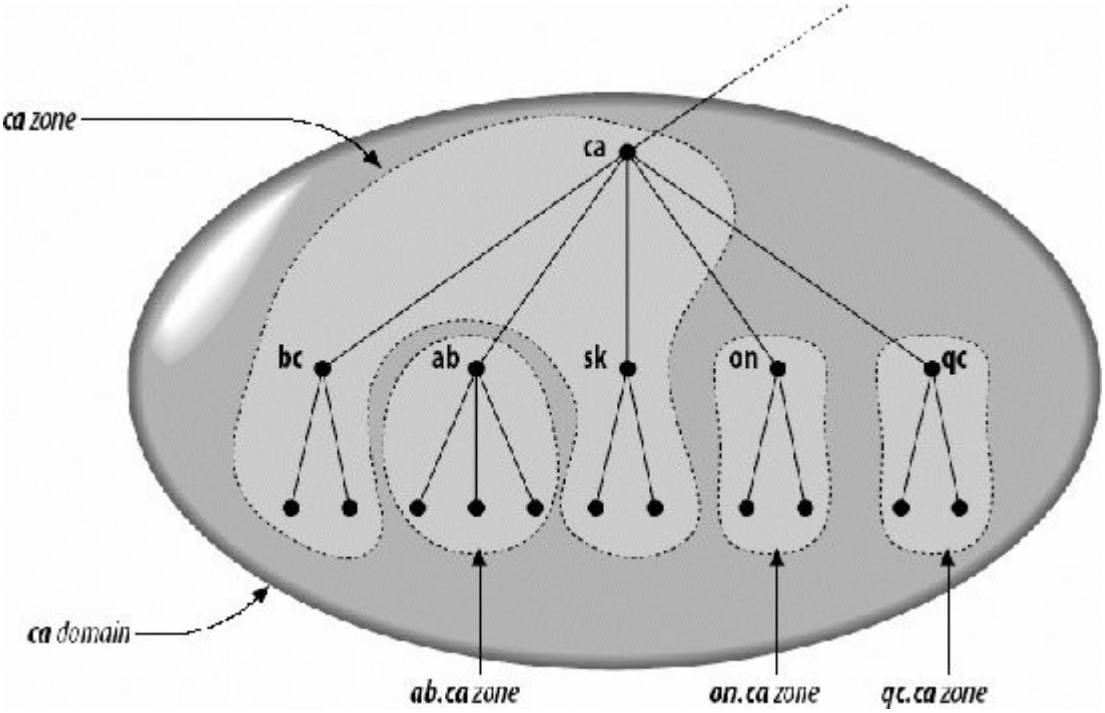
KUVA 6. ACL good guys [11]

View-lohkon match-clients-määrityksellä määritetään mistä tuleviin kyselyihin kyseinen view vastaa. Esimerkiksi määrityksellä match-clients { localhost; good-guys; }; voidaan määrittää view vastaamaan palvelimelta itseltään tuleviin kyselyihin sekä good-guys listassa määritetyistä verkko-osoitteista saapuviin kyselyihin. BIND suorittaa ACL vertailun siten, että vertailu lopetetaan aina ensimmäisen täsmäävän ehdon löydyttyä. Esimerkiksi, jos good-guys listan vertailua tehtäessä kysely tulee IPv4 osoitteesta 192.168.2.5, niin kyseinen view ei palvele tätä kyselyä. Mikäli ehto ! 192.168.2.5/28 siirrettäisiin listassa mihin tahansa väliin, kuitenkin niin että 192.168.2/24 on määritetty ensin, ei kielteinen määritys vaikuttaisi koskaan, sillä vertailu loppuisi aina sallivaan 192.168.2/24 määritykseen [11]. Tämä tekee BIND:n ACL:t kankeiksi, sillä osoitteita ei voi määrittää mielivaltaisessa järjestyksessä, vaan ne on aina järjestettävä niin, että täsmällisin on ensin ja täsmällisyys järjestyksessä kohti epätäsmällisintä.

3.2 Verkkotunnuksen osittaminen

Verkkotunnuksen osittamisen ymmärtämiseksi on ymmärrettävä verkkotunnuksien zonet. Yksinkertaistettuna zone tarkoittaa verkkotunnusta ja tämän aliverkkotunnuksia, joista nimipalvelin on vastuussa ja joista se lähettää vastauksia. Korkeimman tason verkkotunnukset kuten fi ja com ovat ositettu juuresta erilleen. Osittamisen ansiosta fi-verkkotunnuksia hallinnoi Suomen Viestintävirasto, eikä esimerkiksi jokin amerikkalainen organisaatio. Koska jokainen verkkotunnus, juurta lukuunottamatta, on jonkin toisen verkkotunnuksen aliverkkotunnus, kuuluvat aliverkkotunnukset emäverkkotunnuksensa zoneen. Tilanne voidaan muuttaa

tekemällä ositus, jolloin aliverkkotunnus ja sen aliverkkotunnukset lohkotaan omaan zoneensa.



KUVA 7. Ca verkkotunnuksen zonet [2]

Kuvassa 7 on kuvattu mahdollinen tilanne ca-verkkotunnukselle. Verkkotunnuksella ca on useita aliverkkotunnuksia, joista osa on ositettu omiin zoneihinsa. Toisinsanoen on mahdollista osittaa vain osa aliverkkotunnuksista. Verkkotunnus ca on Canadan verkkotunnus ja esimerkissä osittamattomat verkkotunnukset ovat bc (British Columbia) ja sk (Saskatchewan).

Verkkotunnuksen osittaminen tapahtuu niin, että nimipalvelimilla vain määritetään jollekin aliverkkotunnukselle omat nimipalvelimet. Tällöin nimipalvelukyselyn yhteydessä emäverkkotunnuksen nimipalvelin osaa ohjata kyselijän edelleen ositetun verkkotunnuksen vastaavalle nimipalvelimelle. Kuvassa 8 on esimerkki verkkotunnuksen osittamisesta zone-tiedostossa.

```
; zone fragment for example.com
; name servers in the same zone
$TTL 2d ; default TTL is 2 days
$ORIGIN example.com.
@               IN      SOA   ns1.example.com. hostmaster.example.com. (
                2003080800 ; serial number
                2h         ; refresh =  2 hours
                15M        ; update retry = 15 minutes
                3W12h      ; expiry = 3 weeks + 12 hours
                2h20M      ; minimum = 2 hours + 20 minutes
                )
; main domain name servers
                IN      NS     ns1.example.com.
                IN      NS     ns2.example.com.
; main domain mail servers
                IN      MX     10 mail.example.com.
; A records for name servers above
ns1             IN      A      192.168.0.3
ns2             IN      A      192.168.0.4
; A record for mail server above
mail            IN      A      192.168.0.5
....

; sub-domain definitions
$ORIGIN us.example.com.
; we define two name servers for the sub-domain
@               IN      NS     ns3.us.example.com.
; the record above could have been written without the $ORIGIN as
; us.example.com. IN NS ns3.us.example.com.
; OR as simply
;       IN NS   ns3
; the next name server points to ns1 above
                IN      NS     ns1.example.com.
; sub-domain address records for name server only - glue record
ns3             IN      A      10.10.0.24 ; 'glue' record
; the record above could have been written as
; ns3.us.example.com. A 10.10.0.24 if it's less confusing
```

KUVA 8. Delegoiva zone [12]

Verkkotunnus us.example.com. ositetaan omaan zoneensa, jonka vastaava nimipalvelin on ns3.us.example.com sekä ns1.example.com. Koska verkkotunnuksilla on syytä olla useampia nimipalvelimia häiriötilanteiden varalta, on ns1.example.com nimipalvelin tarkoitettu tässä lisättävän orjanimipalvelimena eli toissijaisena nimipalvelimena. Toissijainen nimipalvelin hakee zonen tiedot ensisijaiselta nimpalvelimelta ja toimii näillä tiedoilla zonen vastaavana nimipalvelimena. [12]

3.3 Zonetiedosto

BIND säilyttää verkkotunnukseen, tai tarkemmin sanottuna zoneen, liittyvät tiedostot tekstimuotoisessa tiedostossa eli zonetiedostossa [13]. Zonetiedoston yksittäistä määritettyä tietoa kutsutaan resurssitietueeksi (Resource record) [13]. RFC1033 määrittää zonetiedoston syntaksin, joka on:

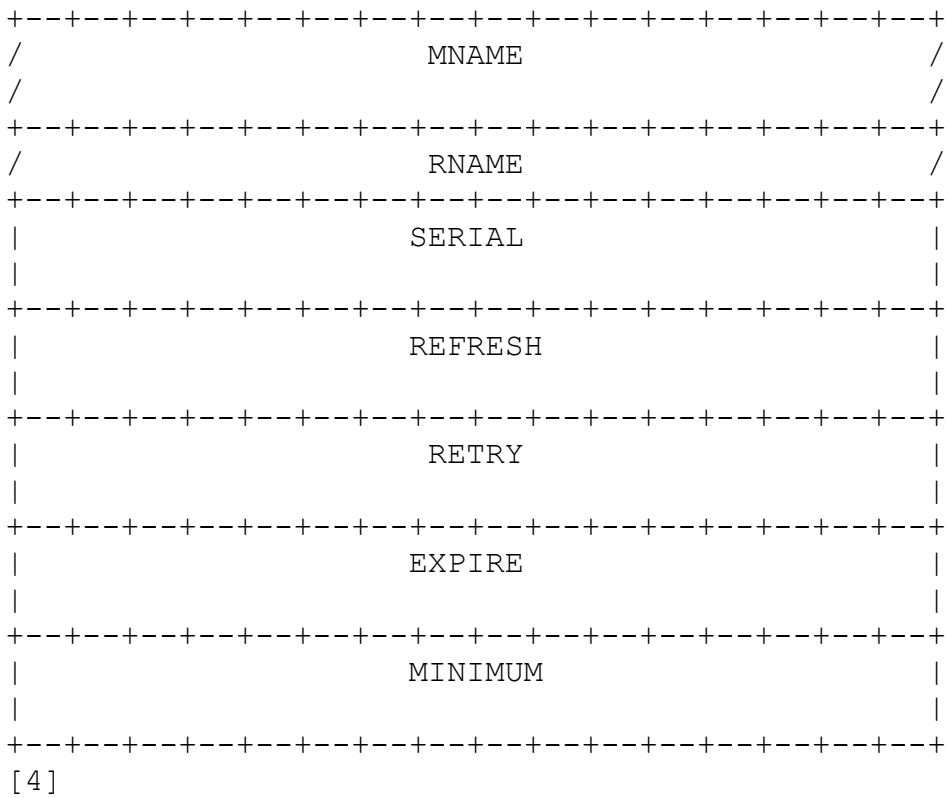
```
<name>    [<ttl>]    [<class>]    <type>    <data>
```

Tietueet erotetaan toisistaan välilyönneillä [14]. Esimerkki resurssitietueesta:

```
mail.example.com. IN A 10.0.0.1
```

- NAME, kenttä määrittää mille verkkotunnukselle kyseinen resurssitietue on määritetty
- TTL, tarkoittaa elinaikaa (Time To Live). Tällä määritetään kuinka kauan resurssin tietoa säilytetään päätelaitteiden ja nimipalvelimien välimuistissa
- CLASS, määrittää protokolla ryhmän, käytännössä aina IN eli Internet
- TYPE, määrittää minkä tyyppisestä resurssitietueesta on kyse. Tyyppi määrittelee kuinka dataosion arvoa käsitellään.
- DATA, varsinainen resurssitietueen data

Joka zonessa on oltava SOA-resurssitietue (start of authority), joka määrittää zonen ylläpitäjän yhteystiedot, pääasiallisen nimipalvelimen ja zonen säilömiseen liittyviä asioita [4]. SOA-tietueen rakenne:



[4]

jossa

- MNAME, pääasiallisesti verkkotunnuksen tiedosta vastaavan nimipalvelimen verkkotunnus

- RNAME, zonesta vastaavan henkilön tai henkilöiden sähköpostiosoite
- SERIAL, etumerkitön 32 bittinen zonen versionumero. Toissijaiset nimipalvelimet käyttävät tätä arvoa vertaillaakseen zonetiedostojen ajantasaisuutta
- REFRESH, 32 bittinen ajanjakso, jonka välein toissijaisen nimipalvelimen tulee päivittää zonetietonsa
- RETRY, 32 bittinen ajanjakso, jonka välein toissijaisen nimipalvelimen tulee yrittää uudelleen, mikäli edellinen zonetietojen päivitys epäonnistui
- EXPIRE, 32 bittinen ajanjakso, jonka jälkeen zone ei ole enää voimassa, mikäli päivitykset ovat epäonnistuneet
- MINIMUM, pienin TTL arvo, jonka yksikään resurssitietue voi saada

[4]

BIND kuitenkin käyttää MINIMUM arvoa negatiivisen vastauksen säilömiseen RFC1033:n vastaisesti [13]. Negatiivinen vastaus tarkoittaa tilannetta, jossa nimipalvelimelle ei ole määritetty kysyttyä resurssitietuetta ja palvelin palauttaa tiedon, ettei tietoa ole olemassa. BIND käyttää MINIMUM arvoa tällaisen vastauksen välimuistissa säilyttämisen ajan määrittelyyn. Esimerkki SOA-resurssitietueesta:

```
@ IN SOA muori.distortionturtle.net. 94734.mail.mamk.fi. (
                          1337716175
                          1200
                          180
                          1209600
                          900)
```

SOA-tietueen lisäksi jokaiselle zonelle tulee määrittää vastaavat nimipalvelimet ja se tapahtuu zonetiedostossa määrittämällä NS-tyyppinen resurssitietue. Resurssitietueella määritetään vastaavan nimipalvelimen osoite.

4 TESTIYMPÄRISTÖN TOTEUTUS

Opinnäytetyön tavoitteena on muodostaa nimipalvelin, joka osaa vastata eritavalla riippuen siitä, mistä nimipalvelukysely saapuu. Tarkoituksena tässä on käyttää BIND-nimipalvelinta siintä syystä että BIND on yleisin nimipalvelin, ainankin BIND:n kehittäjän ISC:n (Internet Systems Consortium) mukaan [8]. BIND mahdollistaa halutun toiminnallisuuden käyttämällä BIND:n view- ja acl-ominaisuuksia. Tällöin on mahdollista määrittää tietylle verkkotunnukselle useita erillaisia zoneja ja jopa jättää

määrittämättä osalle verkosta. Tarkoitus on tehdä yksinkertainen toteutus ja sen perusteella tutkia järjestelmää sekä pohtia mahdollisesti parempaa toteutusta sekä toteutuksen skaalautuvuutta.

Toteutuksessa käytän OpenSuse 12.1 Linux-palvelinta ja palvelinsovelluksena BIND 9.8.1-P1. BIND on myös OpenSusen oletusnimipalvelin, joten tämän asentaminen ei ole vaikeaa. Asennustiedostot löytyvät oletusasennuslähteistä. Asennus onnistuukin yksinkertaisesti YaST:n ohjelmistojenhallinnan kautta, tai kuten itse tykkään ohjelmistoja hallita, konsoliohjelman zypperin avulla, yksinkertaisella komennolla

```
sudo zypper in named
```

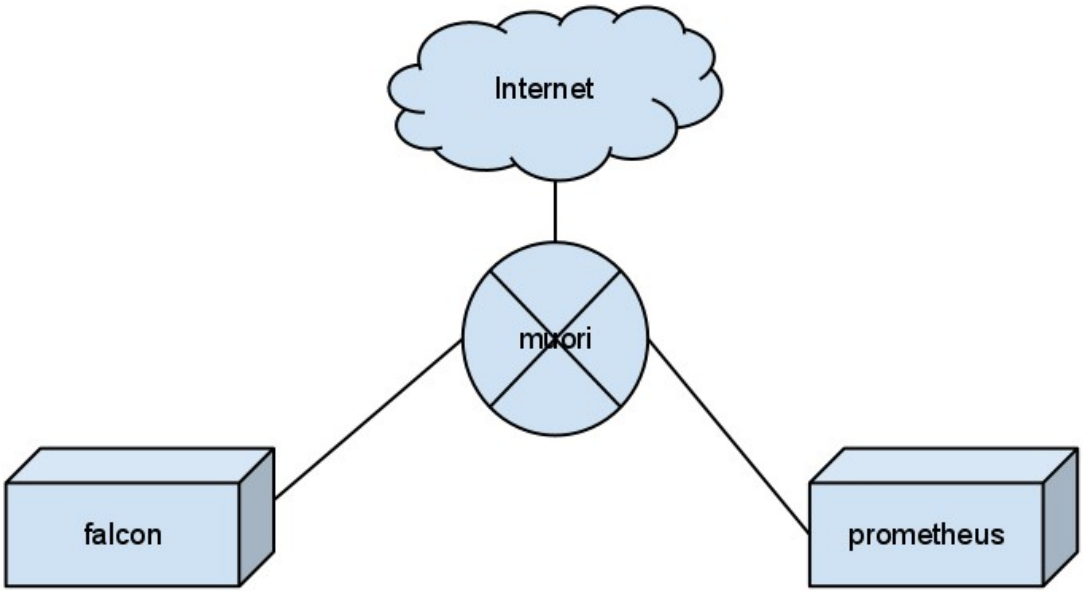
OpenSuse-versiossa BIND:n asetusten määrittämiseksi tärkeitä tiedostoja ja kansioita ovat:

- /etc/named.conf
- /etc/named.conf.include
- /etc/named.d
- /etc/sysconfig/named
- /var/lib/named/

BIND ajetaan niinsanotusti hiekkalaatikoidusti, jolloin palvelinsovellus ei pysty puuttumaan muihin järjestelmän palveluihin. BIND:n hiekkalaatikko eli chroot-kansio on /var/lib/named/. Chroot-kansion alta löytyvät myös zone-tiedostot. Nimipalvelimen kannalta sillä on kolme merkitsevää verkkoaluetta. Julkinen osio, joka kattaa kaikkialta Internetistä tulevat kyselyt, toinen verkko on ensimmäinen osa puoliksi lohkotusta sisäverkosta ja kolmas luonnollisestikin tuo sisäverkon toinen puolikas.

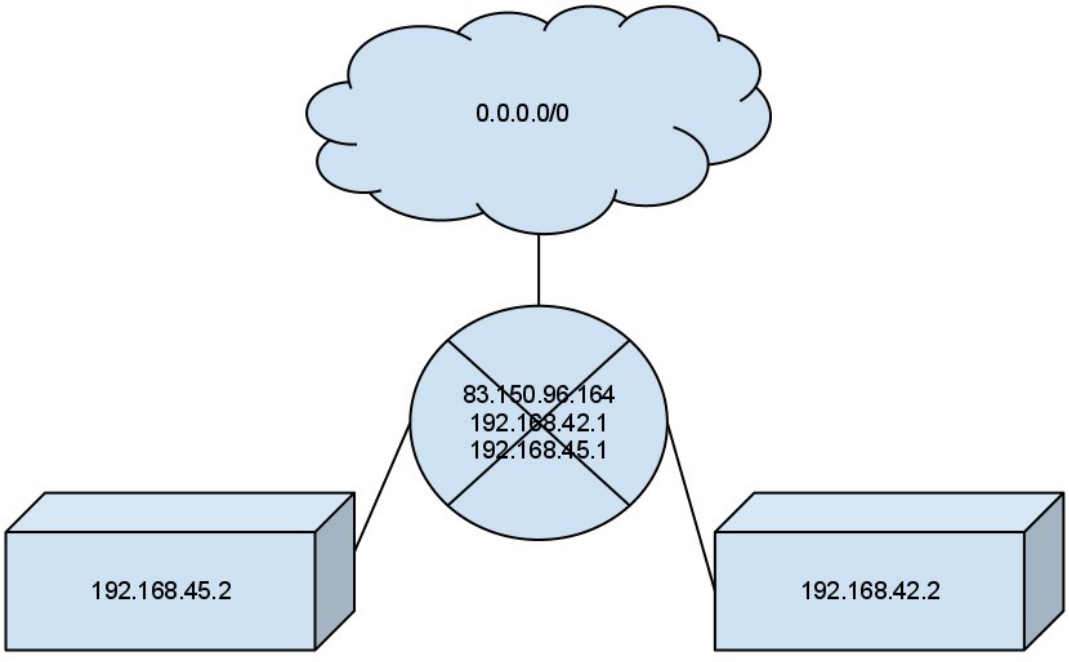
Opinnäytetyössä käytettävä verkkoympäristö koostuu kolmesta osasta. Verkkoja ydistävänä tekijänä on reititin, Muori. Reitittimessä itsessään on muutamia palveluja. Osa niistä on verkkoteknisiä ratkaisuja ja osa on sovellustason ohjelmistoja. Sisäverkkoja on kaksi kappaletta: 192.168.42.0/24 ja 192.168.45.0/24. Merkintä /24 tarkoittaa aliverkon bittien lukumäärää. Merkintä /24 on toisin sanottuna desimaalilukuna esitettynä 255.255.255.0. Tällöin verkon verkko-osoite on 192.168.42.0 ja broadcast-osoite 192.168.42.255 ja tavalliseen käyttöön sallitut IP osoitteet ovat välillä 192.168.42.1 - 192.168.42.254. Molemmista verkoista liikennöinti Internetiin tapahtuu Muorin kautta. Kuvissa 9 ja 10 on opinnäytetyön

verkko kuvattuna. Kuvassa 9 esitetään laitteiden sijainti verkkossa.



KUVA 9. Verkkokaavio 1

Kuvassa 10 on laitteiden IPv4 osoitteet.



KUVA 10. Verkkokaavio 2

Reitin eli Muori on yhteydessä Internetiin IP-osoitteella 83.150.96.164. Muori jakaa Internet-yhteyden niin sanottuihin sisäverkkoihin eli IP-alueille 192.168.42.0/24 sekä 192.168.45.0/24. Sisäverkoista siis pääsee Muorin kautta ulos sekä toisiinsa.

Palomuurilla voisi tehdä rajoituksia verkkojen välille, mutta niiden tekeminen ei ole opinnäytetyössäni tarkoituksenmukaista. Palomuurirajoitukset oikeastaan vain estäisivät osan testeistä. Muorin toimiessa sekä reitittimenä että nimipalvelimena sisäverkoille sekä verkkotunnukselle oppari.distortionturtle.net se on verkkotopologiassa äärimmäisen keskeisellä paikalla. Tarvitaan vain yksi laite ja osassa sovelluksia yksi nimipalvelusovellus palvelemaan kaikkia verkkoja.

4.1 Nimipalvelimen asetustiedosto

Nimipalvelimen asetukset määritetään /etc/named.conf -tiedostossa. Asennuspaketissa on valmis esimerkkiasetustiedosto, jolla pääsee jo hyvin alkuun. Asetustiedostoon on kuitenkin lisättävä view-määritykset, jotta saavutetaan haluttu toiminnallisuus. Nimipalvelin toimii vallan mainiosti jo pelkästään oletusasetuksilla, joten välttämättä asetustiedostossa ei tarvitse kuin lisätä zone-määritykset, ja opinnäytetyöni tapauksessa view-määritykset joihin zone-määritykset. Mikäli käyttää view-määrityksiä, niin kaikki zone-määritykset ovat sijoitettava view-määrityksien sisälle. Tämä on mielestäni vähän typerää, sillä olisi huomattavasti kätevämpää, mikäli pystyisi määrittämään osan zoneista glopaaleiksi ja ne olisivat käytössä kaikissa view-määrityksissä ilman, että ne täytyy erikseen määrittää. Asetustiedoston options-lohkon voi jättää sellaisekseen ellei erikseen halua muuttaa sieltä jotain asetusta. Opinnäytetyöni kannalta options-lohkon asetukset ovat melkolailla yhdentekevät, joten en niitä erityisemmin tässä rupea läpi käymään. Tein kuitenkin asetuksiin pieniä muutoksia, jotka lähinnä lisää palvelimen tietoturvaa. Nämä muutokset ovat nähtävissä Liite 3:ssa, jossa on koko named.conf -tiedosto.

BIND-nimipalvelimen asetustiedostossa named.conf-tiedostossa options-lohko tulee pitää ensimmäisenä, sillä se määrittää palvelimen oletusasetukset, joita voi sitten osittain uudelleenmäärittää esimerkiksi view-kohtaisesti. Options-lohkon jälkeen kannattaa määrittää ACL-määritykset, jotta ne on määritetty ennen kuin niitä kutsutaan view-määrityksissä. On toki mahdollista määrittää ACL:n sisältö suoraan view:ssä match-clients-määrityksellä, mutta mielestäni on selkeämpää, että tehdään erikseen ACL, johon sitten viitataan. Seuraavassa ACL 42 verkolle:

```
acl "42" {
```

```
        192.168.42.0/24;
};
```

Koska ACL sisältää vain yhden aliverkkomäärityksen, olisi oikeastaan aika sama määrittää sama aliverkko view-määrityksen match-clients-määrityksellä. Kuitenkin, heti kun ACL:ää tarvitsee lähteä muokkaamaan, esimerkiksi poislukemaan aliverkon jokin tietty IP tai lisäämään muita osoitteita, pysyy asetukset selkeämpinä ja täten helpommin ylläpidettävimpinä, kun käytetään ACL-määrityksiä. Loput ACL-määritykset ovat nähtävissä liitteessä kolme.

View-määrityksillä päästäänkin jo sitten asian ytimeen, alku 192.168.42.0/24 verkolle räätälöidystä view määrityksestä:

```
view "42" {
        match-clients { 42; };
        recursion yes;
        zone "oppari.distortionturtle.net" {
                type master;
                file "master/42.oppari.zone";
        };
        # The following zone definitions don't need any
        # modification.  The first one
        # is the definition of the root name servers.
        # The second one defines
        # localhost while the third defines the reverse
        # lookup for localhost.


        zone "." in {
                type hint;
                file "root.hint";
        };
        zone "localhost" in {
                type master;
                file "localhost.zone";
        };
        zone "0.0.127.in-addr.arpa" in {
                type master;
                file "127.0.0.zone";
        };
        zone
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.
0.0.0.ip6.arpa" IN {
                type master;
                file "127.0.0.zone";
        };
};
```

View 42:n ensimmäinen määritys ”match-clients” määrittää, että tämä view palvelee ACL 42:ssa määritetyille osoitteille. Seuraava määritys ”recursion yes” mahdollistaa 192.168.42.0/24 verkossa olevat päätelaitteet käyttämään nimipalvelinta omana nimipalvelimenaan, joka suorittaa rekursiiviset nimipalvelukyselyt. Kolmas määritys onkin sitten opinnäytetyön zonen määrittäminen. Nimipalvelin toimii päänimipalvelimena verkkotunnukselle oppari.distortionturtle.net ja tiedosto, josta tämän zonen tiedot ladataan, on master/42.oppari.zone. Tiedoston polku on viittaus nimipalvelimen chroot-kansion juuresta lähtöisin, joten todellinen polku tiedostolle on /var/lib/named/master/42.oppari.zone.

Aliverkolle 192.168.45.0/24 teen samanlaisen view-määrittelyn, mutta siinä match-clients on luonnollisesti acl 45, jotta se täsmää oikealle aliverkolle ja zonetiedoston polku on master/45.oppari.zone. Seuaraavaksi määrittelen oman view:n itse nimipalvelimelle. Nimipalvelimen oma zone on hyvin pitkälti samanlainen kuin 42 ja 45 aliverkkojen, pienellä poikkeuksella, että nimipalvelin ei yritä lähettää zonen muutoksen ilmoitusviestiä tämän view:n perusteella. Zonetiedostona nimpalvelimen omalle view:lle käytetään zonetiedostoa, jonka polku on master/internet.oppari.zone. Viimeinen view-määritys on Internetistä saapuville kyselyille tarkoitettu ja se on määritetty seuraavasti:

```
view "internet" {
        match-clients { any; };
        zone "oppari.distortionturtle.net" {
                type master;
                file "master/internet.oppari.zone";
        };
};
```

Nimipalvelimeni oletusasetusten mukaisesti Internetistä saapuvat kyselyt eivät saa olla rekursiivisiä ja Internetistä katsottuna nimipalvelin tietää ainoastaan oppari.distortionturtle.net zonesta, ei mistään muusta. Internetistä saapuville kyselyille tarkoitettu view sisältää match-clients-määrityksen, jolla sallitaan kaikkialta tuleville kyselyille vastaaminen. Tästä syystä tämä view on määritettävä viimeisenä, muuten tämä view saattaisi täsmätä johonkin kyselyyn jolle on määritetty oma view.

4.2 Zone-tiedostot

Zonetiedostot ovat tässä toteutuksessa asia, jotka vaihtuvat view-kohtaisesti.Samalle zonelle on nyt kolme eri zonetiedostoa hieman eroavin tiedoin. Internetistä ja itse nimipalvelimelta tarkasteltuna oppari.distortionturtle.net verkkotunnuksen zone näyttää seuraavalta:

```
$TTL 86400
@       IN      SOA     muori.distortionturtle.net.
94734.mail.mamk.fi. (1337716175 1200 180 1209600 900 )
@       IN      NS      muori.distortionturtle.net.
@       IN      NS      armas.distortionturtle.net.
muori   IN      A       83.150.96.164
www     IN      A       83.150.96.164
```

Kun vastaavasti aliverkosta 192.168.42.0/24 tarkasteltuna sama zone näyttää seuraavalta:

```
$TTL 86400
@       IN      SOA     muori.distortionturtle.net.
94734.mail.mamk.fi. ( 1338643492 1200 180 1209600 900 )
@          IN      NS      muori.distortionturtle.net.
@          IN      NS      armas.distortionturtle.net.
muori      IN      A       192.168.42.1
www        IN      A       192.168.42.1
prometheus IN      A       192.168.42.2
falcon     IN      A       192.168.45.2
```

Aliverkosta 192.168.45.0/24 vastaavasti zone näyttää seuraavalta:

```
$TTL 86400
@       IN      SOA     muori.distortionturtle.net.
94734.mail.mamk.f. ( 1338643461 1200 180 1209600 900 )
@       IN      NS      muori.distortionturtle.net.
@       IN      NS      armas.distortionturtle.net.
muori   IN      A       192.168.45.1
falcon  IN      A       192.168.45.2
ssh     IN      A       192.168.45.1
```

Selkeimpinä eroina on, että Internetistä käsin ei ole mitään tietoa prometheus.oppari.distortionturtle.net ja falcon.oppari.distortionturtle.net -verkkotunnuksista sekä muori.distortionturtle.net verkkotunnuksen muuttuva IPv4 osoite. Verkkotunnus falcon.oppari.distortionturtle.net on määritetty sekä 192.168.42.0/24 että 192.168.45.0/24 verkkoihin. Tällöin molemmilla verkoilla on nimipalvelussaan käytettävissä tämä osoite. Verkkotunnus ssh.oppari.distortionturtle.net on nähtävissä ainoastaan 192.168.45.0/24-verkon puolella ja www.oppari.distortionturtle.net ei ole lainkaan käytettävissä tästä verkosta.

4.3 Järjestelmä toiminnassa

Verkon toiminnan kannalta tärkeiden sovellusten, kuten DNS-palvelimen ja NAT:n, lisäksi Muorissa on Apache HTTP-palvelin. HTTP-palvelimen lisäksi hyvä esimerkkipalvelinsovellus on SSH-etäkirjautumispalvelin. Molemmilla palveluilla on helppo demonstroida kohdennettua nimipalveluvastausta. Hyvänä esimerkkinä voi toimia Muorissa sijaitseva WWW-sivu, joka on määritetty näkymään vain esimerkiksi toiselle sisäverkolle, jolloin korostuu tarve määrittää verkko-osoitteelle oikea osoite.

Nimipalvelin toimii niin kuin olettaa sopiikin eli asetusten mukaisesti. Internetistä nimipalvelukyselyä tehdessä saa vastaukseksi esimerkiksi falcon.oppari.distortionturtle.net-verkkotunnukselle ”nxdomain”:

```
Host falcon.oppari.distortionturtle.net not found:
3(NXDOMAIN)
```

Aliverkosta 192.168.45.0/24 tehtynä vastaukseksi tuleekin falcon.oppari.distortionturtle.net-osoitteelle määritetty IPv4-osoite:

```
falcon.distortionturtle.net has address 192.168.45.2
```

Saman vastauksen saa myös, mikäli kyselyn tekee 192.168.42.0/24-aliverkosta. Vastaavasti prometheus.oppari.distortionturtle.net saa IPv4-osoitteen ainoastaan 192.168.42.0/24-aliverkosta kysellessä. Verkkotunnus ssh.oppari.distortionturtle.net on myös käytettävissä ainoastaan 192.168.45.0/24-aliverkosta.

5 ARVIOINTI KUINKA SOVELLETTAVA ISOMMASSA MITTAKAAVASSA

BIND view:llä toteutetut asiakaskohtaiset nimipalveluvastaukset toimivat hyvin suurissakin verkoissa silloin, kun omiin lohkoihinsa eriytetään IP-verkkoalueita tai yksittäisiä IP-osoitteita, kuitenkin välttäen samanaikaisten verkkoalueiden ja yksittäisten osoitteiden määrittämistä. ACL-määritykset ovat mielestäni epäkäytännöllisiä, koska yksittäiset IP-osoitteet on määritettävä ennen verkkoalueita. Oikeastaan yksittäiset IP-osoitteet ja verkkoalueet eivät vielä ole ongelmallisia, kunhan muistaa, että yksittäiset osoitteet on määritettävä ensin. Sama pätee

verkkoalueiden verkkomaskin suhteen, eli suppeammpan maskin omaavat verkkoalueet on määriteltävä ensin. Mitä pienempi alue, sitä aiemmin se on määritettävä ennen laajempia verkkoalueita. Tämä tekee ACL:n hallinnan raskaaksi kun ACL listoja on useita ja niissä on useita verkkoalueita. ACL-määrityksien hallinta on myös päivityksien osalta raskasta, koska määritykset on annettava BIND:n asetustiedostossa ja muutoksien jälkeen nimipalvelin on käytännössä käynnistettävä uudelleen.

Useita view-määrityksiä käyttäessä useiden verkkotunnuksien kanssa toivoisi mahdolliseksi määritellä ACL useaan view-määritykseen siten, että kysellyn zonen tietoja ei välttämättä löydy ensimmäisestä view määrityksestä. Määrityksiä läpi käytäessä zonen tiedot löytyisivät mahdollisesti jostain toisesta view-määrityksestä. Viimeisessä view:ssä voisi mahdollisesti olla määritetty sitten sääntö kuinka toimitaan mikäli yksikään aiempi view ei täsmännyt. View:n hallinta muuttuu tosiaan raskaaksi, mikäli nimipalvelimella on useita verkkotunnuksia. View määritykset toimivat kohtuullisen yksinkertaisessa ympäristössä, mutta ylläpito muuttuu raskaaksi ympäristön monimutkaistuttua.

Osoittaakseni kuinka ylläpito raskautuu view:n ja verkkotunnuksien määrän kasvaessa, käytän täysin kuvitteellista esimerkkiä demonstroidakseni tämän. Voi olla hieman vaikea hahmottaa ylläpidon raskaus ennen kuin saa pienen käsityksen kuinka työmäärä kasvaa. Sanotaan, että nimipalvelimella on autoritäärinen nimipalvelin verkkotunnuksille esimerkki1.com ja esimerkki2.com. Esimerkki1.com WWW-palvelu on sen kaltainen, että eri puolella maailmaa sijaitsevat käyttäjät halutaan ohjata eri palvelimelle. Esimerkiksi WWW-palvelin, joka sijaitsee Aasiassa, kykenee palvelemaan nopeammin japanilaisia sivuston käyttäjiä kuin esimerkiksi palvelin Suomessa. Tästä syystä järjestelmän ylläpitäjä on luonut view:n, joka kattaa käytännössä kaikki japanilaisten operaattoreiden IP-verkot. Tähän view:n on määritetty molemmat verkkotunnukset, jolloin japanilaiset käyttäjät voivat käyttää sekä esimerkki1.com että esimerkki2.com palveluja. Japanilaiset operaattorit määrittäneen view:n jälkeen on määritetty view, joka täsmää kaikkiin osoitteisiin. Myös tässä view:ssä on määritetty molemmat verkkotunnukset. Koska verkkotunnuksella esimerkki2.com palveluja ei haluta sijainnin tai minkään muunkaan perusteella ohjata eri palvelimille tai muutenkaan eritellä, voidaan esimerkki2.com

verkkotunnukselle tehdä yksi zone-tiedosto, joka on määritetty verkkotunnukselle molemmissa view määrityksissä.

Poikkeavat määritykset japanilaisten käyttäjien vuoksi aiheuttaa sen, että esimerkki1.com verkkotunnukselle tarvitsee tehdä kaksi eri zone-tiedostoa: toinen rajattua view-määritystä ja toinen yleistä määritystä varten. Zone-tiedostot voivat olla muulta osin identtiset, mutta esimerkiksi vain osoitteen www.esimerkki1.com määrityksellä on eri IP-osoitteet. Tämä johtaa siis siihen, että yhden verkkotunnuksen vuoksi tulee ylläpitää kahta zone-tiedostoa. Mikäli tätä laajennettaisiin esimerkiksi niin, että meksikolaiset käyttäjät halutaan ohjautuvan heitä lähellä olevalle palvelimelle, tarvitsisi heitä varten määrittää oma view ja zone-tiedosto. Mikäli esimerkiksi verkkotunnuksen esimerkki2.com zonen unohtaisi määritellä Meksikon view:stä, tällöin Meksikolaiset käyttäjät eivät voisi lainkaan käyttää esimerkki2.com verkkotunnuksen palveluja, sillä nimipalvelin palauttaisi aina heidän kyselyihinsä ”ei tietoa”-vastauksen. Tämä toki on tehokas keino, mikäli halutaan estää vastaukset jollekin alueelle kokonaan, mutta aiheuttaa lisätyötä ylläpidon kannalta. Esimerkissä kuvaillussa käytössä alkaa jo mielestäni ylläpidollinen taakka olla sen verran suuri, että view:n lukumäärän kasvaessa inhimillisten virheiden mahdollisuus kasvaa suureksi. Tällöin olisi järkevää luoda ohjelma, joka hoitaisi BIND-asetusten ja zone-tiedostojen kirjoittamisen ja ylläpitäjä syöttäisi tiedot tämän lisäohjelman käyttöliittymän kautta.

6 LOPPUPÄÄTELMÄ

Aloitin omaan tapaani opinnäytetyön suurella innostuksella ja kovin tavoittein, kuitenkin melko nopeasti oli nähtävissä, että aihe on mielenkiintoinen vaikka sitä vähän rajaakin ja miettii tarkemmin millä tavalla toteutan opinnäytetyöni. Mikäli olisin voinut tehdä työni jollekin yritykselle tai organisaatiolle, niin olisi se luultavasti ollut mahdollista toteuttaa laajempana ja täten mielestäni hyödyllisempänä. Verkkotunnukset ja nimipalvelu oli perusteiltaan minulle jo entuudestaan tuttu, joten oli mielenkiintoista lähteä laajentamaan tietämystäni aiheesta. Opinnäytetyöstä on ollut minulle selkeästi hyötyä, sillä tehdessäni opinnäytetyötä tuli nimipalvelimen toiminta huomattavasti tutummaksi ja näin ollen päivätyössäni olen osannut toimia entistä paremmin nimipalvelinten kanssa.

Mielestäni opinnäytetyöni lopulta supistui turhan suppeaksi. Erityisesti itseäni jäi harmittamaan, että en saanut tehtyä opinnäytetyötäni millekään yritykselle, jolle työstäni olisi suoraa hyötyä ja työ olisi mahdollisesti pystytty toteuttamaan laajemmassa mittakaavassa. Tällöin olisi ollut mahdollista tehdä esimerkiksi eri palvelinsovellusten vertailu soveltuvuudeltaan tälläiseen käyttöön. Nimipalvelin sovelluksena ei ole kovinkaan monimutkainen, jonka vuoksi olisi myös ollut mielenkiintoista, edes vain vertailukohdan hakemiseksi, ohjelmoida täysin oma toteutus. Oman nimipalvelimen ohjelmointi on kuitenkin sen verran laaja ja aikaa vievä homma, ettei se ehkä edes yksinään olisi ollut järkevää tehdä omana opinnäytetyönään. Tässä olisikin auttanut laajemman ryhmän tuki, jossa olisi ollut useampia osallistujia ja oma osuuteni olisi voinut pysyä riittävän pienenä, jotta olisi ollut mahdollista käyttää nimipalvelimen ohjelmointia vain osana tätä työtä.

Opinnäytetyötä tehdessäni aloin myös epäillä todellista tarvetta asiakaskohtaiselle nimipalvelulle, sillä pienissä ympäristöissä se voi aiheutta ennemmin ongelmia kuin ratkoa niitä. Tilanteet, joissa asiakas voi helposti ja useasti vaihtaa eri määritysten alueelle, voi aiheuttaa käyttäjälle epäselvyyttä ja pahimmillaan käyttökatkoja. Voisin kuvitella esimerkiksi tilanteen, jossa yrityksen nimipalvelimilla olisi estetty jokin palvelu ja nimipalvelin palauttaisi palvelun verkkotunnukselle kohtalaisen pitkän negatiivisen vastauksen esimerkiksi muutamiksi tunneiksi. Käyttäjän lähtiessä pois yrityksen verkosta ja halutessa käyttää palvelua muusta verkosta, olisi hänellä edelleen taakkanaan palvelun esto, kunnes negatiivinen vastaus vanhenisi. Toinen ongelmaa aiheuttava tilanne, joka tulee helposti mieleen, on jokin yrityksen oma palvelu, jota yrityksen sisäverkosta käytetään mahdollisesti verkonrakenteen vuoksi eri IP-osoitteella. Tälle voi helpostikin määräytyä jopa päivien TTL. Käyttäjän lähtiessä esimerkiksi työmatkalle, ja tällöin yrityksen verkon ulkopuolelle, IP-osoite voi muuttua ja yrityksen verkon sisällä ollut osoite voi lakata kokonaan toimimasta. Toki mikäli yrityksen verkossa käytetään asiakaskohdennettua nimipalvelua, niin tälläiset tilanteet on otettava huomioon ongelmien minimoimiseksi.

BIND view-ominaisuudella toteutettu kohdennettu nimipalvelin toimii hyvin ja riittävän tehokkaasti esimerkiksi jonkin organisaation verkkotunnuksien pääasiallisena nimipalvelimena sekä sisäverkkojen nimipalvelimena. Tällaisessa ympäristössä on

tehokasta hoitaa yhdellä palvelimella usean nimipalvelimen tehtävät. Oletuksena nimipalvelin voi vastata Internetistä tuleville kyselyille ja jakaa zonea tavallisilla zonen jakomenetelmillä toissijaisille nimipalvelimille. Samalla organisaation sisäverkko voi olla lohkottu useisiin osiin ja näillä voi olla erillaisia tarpeita ja rajoituksia, johon nimipalvelin on helppo mukauttaa. Nimipalveluiden tarjoaminen niin, että vaikka maanosittain olisi mahdollista määrittää verkkotunnuksen eroavaisuuksia, tai jopa tarkemmin, kasvattaa BIND:n tapauksessa ylläpidon kuormaa niin paljon, että mielestäni tähän opinnäytetyön toteutus ei juurikaan sovellu. Järjestelmälle on tarvetta esimerkiksi Googlen tapauksessa, jossa on järkevää ohjata käyttäjiä maantieteellisesti lähemmille tai vähemmän ruuhkaisille palvelimille. Luulenkin, että Googlen ja tämän kaltaisten toimijoiden järjestelmät on toteutettu huomattavasti ylläpidettävämmällä toteutuksella. Googlen tapauksessa en ihmettelisi mikäli heillä olisi kehitetty kokonaan oma nimipalvelinsovellus tarkoitusta varten.

LÄHTEET

[1] Martin Dodge. Historical Maps of Computer Networks. WWW-dokumentti. http://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html. 2007. Lainattu 24.5.2012.
[2] Albitz, Paul, Liu, Cricket. DNS and BIND, 5th Edition. 2006.
[3] Wikipedia. ARPANET. WWW-dokumentti. http://en.wikipedia.org/wiki/ARPANET. 2012. Lainattu 3.6.2012.
[4] Mockapetris P.. Domain names - Implementation and specification. Tekninen dokumentti. http://www.ietf.org/rfc/rfc1035.txt. 1987.
[5] Kozierok, Charles M.. DNS Labels, Names and Syntax Rules . WWW-dokumentti. http://www.tcpipguide.com/free/t_DNSLabelsNamesandSyntaxRules.htm. 2005. Lainattu 1.4.2012.
[6] Kozierok, Charles M.. DNS Labels, Names and Syntax Rules. WWW-dokumentti. http://www.tcpipguide.com/free/t_DNSLabelsNamesandSyntaxRules-3.htm. 2005. Lainattu 1.4.2012.
[7] FICORA. Native language characters (å, ä, ö and Lappish) in domain names. WWW-dokumentti. http://www.ficora.fi/en/index/palvelut/palvelutaiheittain/fiverkkotunnukset/aakkostenkaytto.html. 22.7.2010. Lainattu 1.4.2012.
[8] Internet Systems Consortium. BIND. WWW-dokumentti. http://www.isc.org/software/bind. 2012. Lainattu 24.4.2012.
[9] Shane Tzen. Split Views with Bind 9 Howto. WWW-dokumentti. http://www.knowplace.org/pages/howtos/split_view_with_bind_9_howto.php. . Lainattu 26.4.2012.
[10] ZyTrax Inc.. DNS BIND acl clause. WWW-dokumentti. http://www.zytrax.com/books/dns/ch7/acl.html. 2011. Lainattu 1.5.2012.
[11] ZyTrax Inc.. BIND Definition of Address List Match. WWW-dokumentti. http://www.zytrax.com/books/dns/ch7/address_match_list.html. 2011. Lainattu 1.5.2012.
[12] ZyTrax Inc.. HOWTO - Delegate a Sub-domain (a.k.a. subzone). WWW-dokumentti. http://www.zytrax.com/books/dns/ch9/delegate.html. 2011. Lainattu 22.5.2012.
[13] Internet Systems Consortium. BIND 9 Administrator ReferenceManual. Tekninen dokumentti. http://ftp.isc.org/isc/bind9/cur/9.8/doc/arm/Bv9ARM.pdf. 2012.
[14] Lottor M.. Domain Administrators Operations Guide. Tekninen dokumentti. http://tools.ietf.org/html/rfc1033. 1987.

LIITE 1 (1).
BIND nimipalvelimen esimerkki asetustiedosto

```
// This is the primary configuration file for the BIND DNS server named.
// This is for example only
acl "dns_slaves" {
  172.16.1.2;   # IP of the slave DNS nameserver
  172.16.101.2  # ditto
};
acl "lan_hosts" {
  192.168.0.0/24;  # network address of your local LAN
  127.0.0.1;       # allow loop back
};
options {                  # this section sets the default options
  directory "/etc/namedb"  # directory where the zone files will reside
  listen-on {
    192.168.0.1;           # IP address of the local interface to listen
    127.0.0.1;             # ditto
  };
  auth-nxdomain no;        # conform to RFC1035
  allow-query { any; };    # allow anyone to issue queries
  recursion no;            # disallow recursive queries unless over-ridden below
  version "0";             # obscures version reporting - can't hurt
};
key "rndc-key" {
  algorithm hmac-md5;
  secret "nOzUd7+Hwdq6k6CQq7SbDw==";  # DO NOT USE THIS KEY - example only
};
controls {
  inet 127.0.0.1 allow { localhost; }
  keys { rndc-key; };
};
view "internal" {
  match-clients { lan_hosts; };  # match hosts in acl "lan_hosts" above
  recursion yes;                 # allow recursive queries
  notify no;                     # disable AA notifies

  // prime the server with knowledge of the root servers
  zone "." {
    type hint;
    file "db.root";
  };
  // be authoritative for the localhost forward and reverse zones, and for
  // broadcast zones as per RFC 1912
  zone "localhost" {
    type master;
    file "db.local";
  };
  zone "127.in-addr.arpa" {
    type master;
    file "db.127";
  };
  zone "0.in-addr.arpa" {
    type master;
```

```
    file "db.0";
  };
  zone "255.in-addr.arpa" {
    type master;
    file "db.255";
  };
  zone "example.com" {
    type master;
    file "internal/example.com.zone";
  };
};
view "external" {

  // "localnets" and "any" are special reserved words
  // "localnets" mean any network address (as opposed to host address) configured
  // on the local network interfaces - "!" means to negate
  match-clients { !localnets; any; };
  recursion no;                   # disallow recursive queries
  allow-transfer { dns_slaves; }; # allow "hosts in acl "dns_slaves" to transfer zones

  zone "example.com" {
    type master;
    file "external/example.com.zone";
  };
};
```

Määräys

FI-VERKKOTUNNUKSEN TEKNISISTÄ MÄÄRITTELYISTÄ JA SALLITUISTA MERKEISTÄ

Annettu Helsingissä 8 päivänä elokuuta 2006

Viestintävirasto on määrännyt 13 päivänä maaliskuuta 2003 annetun verkkotunnuslain (228/2003) 4 §:n 1 momentin ja 4 a §:n 2 momentin, sellaisena kuin se on laissa (187/2006), nojalla:

1 §

Soveltamisala

Tätä määräystä sovelletaan Internet-tietoverkon fi-maatunnukseen päättyviin verkkotunnuksiin.

2 §

Nimipalvelimet

Verkkotunnus on määriteltävä vähintään kahteen ja enintään kymmeneen toisistaan riippumattomaan nimipalvelimeen. Verkkotunnus on määriteltävä kaikkiin nimipalvelimiin tämän määräyksen mukaisesti ja niihin on saatava yhteys Internet-tietoverkosta. Määrittelyt on voitava tarkastaa Viestintäviraston tekemillä automaattisilla nimipalvelukyselyillä. Nimipalvelimissa on oltava NS-tietueet (Name Server), joissa on määritelty kaikki verkkotunnuksen nimipalvelimet. NS-tietueiden on osoitettava palvelimiin, joille on määritelty A-tietueella IP-osoite nimipalvelussa.

3 §

Sähköpostiyhteydet

Verkkotunnusta koskevien MX-tietueiden (Mail Exchanger) määritteleminen nimipalvelimiin on vapaaehtoista. Jos nimipalvelimiin määritellään verkkotunnusta koskevat MX-tietueet, on MX-tietueiden osoitettava palvelimiin, joille on määritelty A-tietueella IP-osoite nimipalvelussa. Tällöin myös palvelimien sähköpostijärjestelmät on määriteltävä vastaanottamaan verkkotunnukselle lähetetyt sähköpostit.

4 §

SOA-tietue

Verkkotunnuksen nimipalvelimen asetukset määrittelevän SOA-tietueen (Start of Authority) on oltava seuraavien vaatimusten mukainen:

1) MNAME (Master Name) -kentässä on oltava verkkotunnuksen ensisijaisen nimipalvelimen nimi;

2) RNAME (Responsible Name) -kentässä on oltava toimiva sähköpostiosoite nimipalvelimien ylläpidosta vastaavalle taholle; sekä

3) sarjanumerot ja ajastimet eivät saa olennaisesti poiketa julkaistuista Internet-standardeista ja -suosituksista.

5 §

Verkkotunnuksen pituus ja sallitut merkit

Verkkotunnuksessa voi olla enintään 63 merkkiä.

Verkkotunnuksessa sallittuja merkkejä ovat kirjaimet a – z, numerot 0 - 9 ja yhdysmerkki (-, tavuviiva-miinusmerkki) sekä seuraavat kansalliset merkit:

Merkki	**Koodi**	**Nimi**
á	00E1	Latinalainen pienaakkonen a ja akuutti
â	00E2	Latinalainen pienaakkonen a ja sirkumfleksi
ä	00E4	Latinalainen pienaakkonen a ja treema (yleiskielessä pieni ä)
å	00E5	Latinalainen pienaakkonen a ja yläpuolinen ympyrä (yleiskielessä pieni ruotsalainen o)
č	010D	Latinalainen pienaakkonen c ja hattu (yleiskielessä pieni hattu-c)
đ	0111	Latinalainen pienaakkonen d ja poikkiviiva
ǥ	01E5	Latinalainen pienaakkonen g ja poikkiviiva
ǧ	01E7	Latinalainen pienaakkonen g ja hattu
ǩ	01E9	Latinalainen pienaakkonen k ja hattu
ŋ	014B	Latinalainen pienaakkonen äng
õ	00F5	Latinalainen pienaakkonen o ja tilde
ö	00F6	atinalainen pienaakkonen o ja treema (yleiskielessä pieni ö)
š	0161	Latinalainen pienaakkonen s ja hattu (yleiskielessä pieni hattu-s)
ŧ	0167	Latinalainen pienaakkonen t ja poikkiviiva
ž	017E	Latinalainen pienaakkonen z ja hattu (yleiskielessä pieni hattu-z)
ʒ	0292	Latinalainen pienaakkonen ezh
ǯ	01EF	Latinalainen pienaakkonen ezh ja hattu

Verkkotunnus ei saa alkaa yhdysmerkillä eikä päättyä yhdysmerkkiin.

Verkkotunnuksen kolmas ja neljäs merkki eivät saa molemmat olla

yhdysmerkkejä. Kansallisia merkkejä sisältävän verkkotunnuksen ACEmuodon (ASCII Compatible Encoding) kolmas ja neljäs merkki saavat kuitenkin molemmat olla yhdysmerkkejä.

6 §

Voimaantulo

Tämä määräys tulee voimaan 14 päivänä elokuuta 2006 ja se on voimassa toistaiseksi.

Tällä määräyksellä kumotaan Viestintäviraston 23 päivänä maaliskuuta 2006 antama määräys fi-verkkotunnuksen teknisistä määrittelyistä ja sallituista merkeistä (Viestintävirasto 37 D/2006 M).

7 §

Tiedonsaanti ja julkaiseminen

Tämä määräys on julkaistu Viestintäviraston määräyskokoelmassa ja se on saatavissa Viestintäviraston asiakaspalvelusta:

Käyntiosoite Itämerenkatu 3 A, HELSINKI
Postiosoite PL 313, 00181 HELSINKI
Puhelin (09) 6966 500
Telekopio (09) 6966 410
WWW-sivusto http://www.ficora.fi/
Y-tunnus 0709019-2

Helsingissä 8 päivänä elokuuta 2006

Pääjohtajan estyneenä ollessa
Johtaja Tapani Rantanen
Johtaja Timo Lehtimäki

LIITE 3 (1)
named.conf

```
options {
        # The directory statement defines the name server's working directory
        directory "/var/lib/named";
        # Write dump and statistics file to the log subdirectory.  The
        # pathenames are relative to the chroot jail.
        dump-file "/var/log/named_dump.db";
        statistics-file "/var/log/named.stats";
        # disallow recursive queries unless over-ridden later
        recursion no;
        managed-keys-directory "/etc/named.d";
};
# Include the meta include file generated by createNamedConfInclude.  This
# includes all files as configured in NAMED_CONF_INCLUDE_FILES from
# /etc/sysconfig/named
include "/etc/named.conf.include";
acl "42" {
        192.168.42.0/24;
};
acl "45" {
        192.168.45.0/24;
};
view "42" {
        match-clients { 42; };
        recursion yes;
        zone "oppari.distortionturtle.net" {
                type master;
                file "master/42.oppari.zone";
        };
        # The following zone definitions don't need any modification.  The first one
        # is the definition of the root name servers.  The second one defines
        # localhost while the third defines the reverse lookup for localhost.
        zone "." in {
                type hint;
                file "root.hint";
        };
        zone "localhost" in {
                type master;
                file "localhost.zone";
        };
        zone "0.0.127.in-addr.arpa" in {
                type master;
                file "127.0.0.zone";
        };
          zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa"
IN {
                type master;
                file "127.0.0.zone";
        };
};
view "45" {
        match-clients { 45; };
```

```
        recursion yes;
        zone "oppari.distortionturtle.net" {
                type master;
                file "master/45.oppari.zone";
        };
        # The following zone definitions don't need any modification.  The first one
        # is the definition of the root name servers.  The second one defines
        # localhost while the third defines the reverse lookup for localhost.
        zone "." in {
                type hint;
                file "root.hint";
        };
        zone "localhost" in {
                type master;
                file "localhost.zone";
        };
        zone "0.0.127.in-addr.arpa" in {
                type master;
                file "127.0.0.zone";
        };
          zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa"
IN {
                type master;
                file "127.0.0.zone";
        };
};
view "server" {
        match-clients { localhost; };
        recursion yes;
        notify no;
        # The following zone definitions don't need any modification.  The first one
        # is the definition of the root name servers.  The second one defines
        # localhost while the third defines the reverse lookup for localhost.
        zone "." in {
                type hint;
                file "root.hint";
        };
        zone "localhost" in {
                type master;
                file "localhost.zone";
        };
        zone "0.0.127.in-addr.arpa" in {
                type master;
                file "127.0.0.zone";
        };
          zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa"
IN {
                type master;
                file "127.0.0.zone";
        };
        zone "oppari.distortionturtle.net" {
```

```
                type master;
                file "master/internet.oppari.zone";
        };
};
view "internet" {
        match-clients { any; };
        zone "oppari.distortionturtle.net" {
                type master;
                file "master/internet.oppari.zone";
        };
};
```

```
$TTL 86400
@       IN      SOA     muori.distortionturtle.net. 94734.mail.mamk.fi. (
                                1338643492      ; Unsigned 32 bit value in range 1 to 4294967295 with a maximum
                                                ; increment of 2147483647. In BIND implementations this is defined
                                                ; to be a 10 digit field. This value MUST increment when any
                                                ; resource record in the zone file is updated. A slave (Secondary)
                                                ; DNS server will read the master DNS SOA record periodically,
                                                ; either on expiry of refresh (defined below) or when it receives a
                                                ; NOTIFY and compares arithmetically its current value of sn with
                                                ; that received from the master DNS. If the sn value from the master
                                                ; is arithmetically HIGHER than that currently stored by the slave
                                                ; then a zone transfer (AXFR/IXFR) is initiated. If the value of sn
                                                ; from the master DNS SOA is the same or LOWER then no zone transfer
                                                ; is initiated.

                                1200            ; Indicates the time when the slave will try to refresh the zone from
                                                ; the master

                                180             ; Defines the time between retries if the slave (secondary) fails to
                                                ; contact the master when refresh (above) has expired.

                                1209600         ; Indicates when the zone data is no longer authoritative. Used by
                                                ; Slave or (Secondary) servers only.

                                900             ; The negative caching time - the time a NAME ERROR = NXDOMAIN result
                        )                       ; may be cached by any resolver.
@               IN      NS      muori.distortionturtle.net.
@               IN      NS      armas.distortionturtle.net.
```

```
muori           IN      A       192.168.42.1
www             IN      A       192.168.42.1
prometheus      IN      A       192.168.42.2
falcon          IN      A       192.168.45.2
```

```
$TTL 86400
@       IN      SOA     muori.distortionturtle.net. 94734.mail.mamk.f. (
                                1338643461      ; Unsigned 32 bit value in range 1 to 4294967295 with a maximum
                                                ; increment of 2147483647. In BIND implementations this is defined
                                                ; to be a 10 digit field. This value MUST increment when any
                                                ; resource record in the zone file is updated. A slave (Secondary)
                                                ; DNS server will read the master DNS SOA record periodically,
                                                ; either on expiry of refresh (defined below) or when it receives a
                                                ; NOTIFY and compares arithmetically its current value of sn with
                                                ; that received from the master DNS. If the sn value from the master
                                                ; is arithmetically HIGHER than that currently stored by the slave
                                                ; then a zone transfer (AXFR/IXFR) is initiated. If the value of sn
                                                ; from the master DNS SOA is the same or LOWER then no zone transfer
                                                ; is initiated.

                                1200            ; Indicates the time when the slave will try to refresh the zone from
                                                ; the master

                                180             ; Defines the time between retries if the slave (secondary) fails to
                                                ; contact the master when refresh (above) has expired.

                                1209600         ; Indicates when the zone data is no longer authoritative. Used by
                                                ; Slave or (Secondary) servers only.

                                900             ; The negative caching time - the time a NAME ERROR = NXDOMAIN result
                        )                       ; may be cached by any resolver.
@       IN      NS      muori.distortionturtle.net.
@       IN      NS      armas.distortionturtle.net.
```

```
muori   IN      A       192.168.45.1
falcon  IN      A       192.168.45.2
ssh     IN      A       192.168.45.1
```

LIITE 6
Internet-zone

```
$TTL 86400
@       IN      SOA     muori.distortionturtle.net. 94734.mail.mamk.fi. (
                                1337716175      ; Unsigned 32 bit value in range 1 to 4294967295 with a maximum
                                                ; increment of 2147483647. In BIND implementations this is defined
                                                ; to be a 10 digit field. This value MUST increment when any
                                                ; resource record in the zone file is updated. A slave (Secondary)
                                                ; DNS server will read the master DNS SOA record periodically,
                                                ; either on expiry of refresh (defined below) or when it receives a
                                                ; NOTIFY and compares arithmetically its current value of sn with
                                                ; that received from the master DNS. If the sn value from the master
                                                ; is arithmetically HIGHER than that currently stored by the slave
                                                ; then a zone transfer (AXFR/IXFR) is initiated. If the value of sn
                                                ; from the master DNS SOA is the same or LOWER then no zone transfer
                                                ; is initiated.
                                1200            ; Indicates the time when the slave will try to refresh the zone from
                                                ; the master
                                180             ; Defines the time between retries if the slave (secondary) fails to
                                                ; contact the master when refresh (above) has expired.
                                1209600         ; Indicates when the zone data is no longer authoritative. Used by
                                                ; Slave or (Secondary) servers only.
                                900             ; The negative caching time - the time a NAME ERROR = NXDOMAIN result
                        )                       ; may be cached by any resolver.
@       IN      NS      muori.distortionturtle.net.
@       IN      NS      armas.distortionturtle.net.
muori   IN      A       83.150.96.164
www     IN      A       83.150.96.164
```