Teemu Väisänen

# PK-yrityksen lähiverkon suunnittelu, toteutus ja valvonta

Metropolia Ammattikorkeakoulu
Insinööri
Tietotekniikka
Insinöörityö
9.5.2012

Metropolia

**Tiivistelmä**

| | |
|---|---|
| Tekijä(t)<br>Otsikko<br><br>Sivumäärä<br>Aika | Teemu Väisänen<br>PK-yrityksen lähiverkon suunnittelu, toteutus ja valvonta<br><br>41 sivua + 1 liite<br>9.5.2012 |
| Tutkinto | Insinööri (240 op AMK) |
| Koulutusohjelma | Tietotekniikka |
| Suuntautumisvaihtoehto | Tietoverkot |
| Ohjaaja(t) | Lehtori Marko Uusitalo |


Tämä insinöörityö tehtiin eräälle yritykselle, jonka päätoimisto sijaitsee Keravalla. Työn tarkoituksena oli parantaa kyseisen yrityksen lähiverkon toteutusta sekä sen hallinnassa olevien laitteiden valvontaa. Heillä oli valmiina käytössä oleva verkko, jolle haluttiin päivitystä internetoperaattorin vaihdon yhteydessä. Yrityksen hallintaan kuului myös isompi yrityskeskuksen internet-yhteyden ylläpito, ja tälle toteutettiin valvonta.

Työn teoriaosuudessa käsitellään lähiverkon suunnittelua, sen vaiheita sekä siihen liittyviä ideologioita. Työ on kuitenkin hyvin käytännönläheinen ja keskittyy enemmän itse verkon toteutukseen Ciscon IOS:n kautta sekä Nagioksella hoidettavaan laitteiden valvontaan. Koska selainpohjainen reitittimen asetusten muokkaus on ainakin vanhemmissa käyttöjärjestelmissä hyvin toimimaton ja epäluotettava, on suositeltavaa käyttää joko SSH:ta tai laitteen konsoliporttia.

Työ alkoi laatimalla yrityksen sisäiseen verkkoon perusasetukset sekä internetyhteydet. Lähiverkon toiminnan tarkastuksen jälkeen siirryttiin laajentamaan verkon sisäisiä palveluita, joihin kuuluivat muun muassa etäyhteyden luonti sekä etähallinta. Verkkoa luodessa lisättiin myös lähiverkon palveluita luomalla verkolle radius-palvelu sekä sille sisäinen DNS-nimipalvelin.

Lähiverkon toteutuksen jälkeen ryhdyttiin järjestämään verkolle valvontajärjestelmää. Yritykselle oli valmiina käytössä Linux-käyttöjärjestelmälle asennettu Nagios Core -valvontaohjelmisto. Kyseisellä ohjelmistolla voidaan valvoa verkon laitteita muun muassa ping-testillä. Sillä saadaan myös lähetettyä tekstiviestejä sekä sähköpostia vikatilanteen ilmentyessä. Nagios on open source -ohjelmisto, eli sen asennettua on lähdekoodi vapaasti käytettävissä.

Tuloksena saatiin yritykselle toimiva lähiverkkototeutus, ja sille kuuluva reaaliaikainen valvonta- sekä hälytysjärjestelmä. Toteutus loi myös tilaa verkon kasvulle.



| | |
|---|---|
| Avainsanat | Cisco, lähiverkko, PK-yritys, Nagios, valvonta |

Metropolia

**Abstract**

| | |
|---|---|
| Author(s)<br>Title<br><br>Number of Pages<br>Date | Teemu Väisänen<br>Planning, implementing and monitoring of SME LAN<br><br>41 pages + 1 appendice<br>9 May 2012 |
| Degree | Bachelor of Engineering (240 cu AMK) |
| Degree Programme | Information Technology |
| Specialisation option | Data Networks |
| Instructor(s) | Marko Uusitalo, Senior Lecturer |


This final project was done for a company currently residing in Kerava. This projects aims to improve an already existing networks implementation and to improve the monitoring of the equipment in their networks, such as switches and routers. The company had an existing local area network that they wanted to improve at the same time they were changing their current internet provider. They also had a larger network under their control that needed better supervision that was also implemented.

The theoretical part deals with the planning of a local area network, its stages and ideologies. This project is focused on the actual implementation of the LAN and it concentrates on the Cisco IOS as well as the monitoring software called Nagios. It is not recommended to use browser-based configuration because it is, at least on older versions, really buggy and unreliable. This is why it is better to use either SSH or the console port on Cisco routers for configuration.

The implementation started on bringing the network online with internet connection. After ensuring the proper working of the basic services the LANs internal services were extended with for example creating a remote-access to the LAN, and implementing a secure remote control with SSH. Later on a radius service and a DNS service were added.

When the LAN was working as it was supposed to a monitoring service needed to be implemented. For this the company had a program called Nagios Core installed on their Linux-server. This program allows monitoring of network devices with for example a simple ping command. It also can send emails or text messages for configured users when a problem occurs. Nagios is an open source program, which means that once installed, users have free control over the source code.

The result of this project was a working local area network with real-time monitoring and alarm system. This implementation allows for growth on the network without extra costs.



| | |
|---|---|
| Keywords | Cisco, local area network, SME, Nagios, monitoring |

Sisällys


1 Johdanto 1

2 Suunnittelu 2

2.1 Kartoitus 3
2.2 Prosessin vaiheet 4

3 Toteutus 7

3.1 Perusasetukset 7
3.2 Internetyhteyden luonti 7
3.3 DHCP-asetukset 8
3.4 NAT/PAT 9
3.4.1 NAT/PAT-portinohjaukset 11
3.4.2 Epävirallinen osoitekäännöstapa 11
3.5 DNS-palvelu 12
3.6 VPN-tunnelin luonti (point-to-point) 15
3.6.1 Tunnelin ensimmäinen luontivaihe 15
3.6.2 Tunnelin toinen luontivaihe 16
3.7 PPTP/L2TP-yhteys yrityksen lähiverkkoon 17
3.8 Etähallinta 23
3.9 OSPF 24

4 Valvonta 25

4.1 Yleisiä työkaluja 25
4.2 Nagios 26
4.2.1 Käyttäjien määritys 28
4.2.2 Laitteiden konfigurointi 32

5 Kehitysideoita 37

5.1 NetFlow ja NBAR 37
5.2 QoS 38
5.3 Vikasietoisuuden lisääminen 40

Lähteet 42

Liitteet

Liite 1. Nagios-ohjelman konfiguraatiotiedoston sisältö

## Termistö

**VPN** – **V**irtual **P**rivate **N**etwork. Tapa yhdistää kaksi tai useampi sisäinen verkko julkisen verkon yli salatusti. Luoden näin turvallisen yhteyden, jolla voidaan siirtää tietoja verkkojen välillä. (Andersson & Madsen: VPN.)

**DHCP** – **D**ynamic **H**ost **C**ontrol **P**rotocol. Verkkoprotokolla jolla jaetaan lähiverkon laitteille muun muassa IP-osoitteita. Tämä mahdollistaa samassa verkossa olevien laitteiden kommunikaation. (Droms: DHCP.)

**OSPF** – **O**pen **S**hortest **P**ath **F**irst. Dynaaminen verkkojen välinen reititysprotokolla. Mahdollistaa kahden tai useamman eri verkon kommunikaation. (Moy: OSPF.)

**Nagios** – Linux käyttöjärjestelmälle pohjautuva, vapaassa levityksessä oleva IT-infrastruktuurin valvontaohjelma. Ohjelmaan sisältyy mm. tekstiviestihälytykset, sähköpostihälytykset sekä sitä voi vapaasti muokata omien taitojen mukaisesti.

**Interface** – Verkkolaitteen portti.

**VLAN** – **V**irtual **L**ocal **A**rea **N**etwork. Tekniikka, jolla voidaan jakaa virtuaalisesti fyysinen tietoliikenneverkko halutulla tavalla osiin riippumatta siitä, missä laitteet sijaitsevat. (Chown:VLAN.)

**SSH** – **S**ecure **S**hell. Salattuun tietoliikenteeseen suunniteltu protokolla. Tällä voidaan mm. muodostaa yhteys verkkolaitteeseen ilman fyysistä yhteyttä. (Ylonen: SSH.)

**NAT/PAT** – **N**etwork **A**ddress **T**ranslation / **P**ort **A**ddress **T**ranslation. Tekniikka, joka vähentää julkisten IP-osoitteiden tarvetta. Tällä tekniikalla useampi verkon laite pääsee julkiseen verkkoon saman julkisen IP-osoitteen kautta hyödyntämällä eri portteja.

**WAN** – **W**ide **A**rea **N**etwork. Tiedonsiirtoverkko, joka kattaa suuren maantieteellisen alueen, ääriesimerkkinä Internet. (Cisco DocWiki: Wan.)

**Access-list** – Verkon liikenteen hallintaan liittyvä pääsylista. Tällä hallitaan sallittuja portteja ja protokollia verkon liikenteessä.

**VPDN** - **V**irtual **P**rivate **D**ial-up **N**etwork. PPTP/L2TP-protokollien mahdollistavan VPN-yhteyden hallintaprotokolla.

**NTP** - **N**etwork **T**ime **P**rotocol. UDP-pohjainen verkkoprotokolla, jolla saadaan tahdistettua samassa verkossa olevan laitteiden kellonajat.

**PING -** TCP/IP-protokollan työkalu, jolla testataan määritellyn laitteen saavutettavuutta. Ping lähettää kyseiselle laitteelle paketin, jolla pyydetään vastausta kyseiseltä laitteelta. Mikäli laite on saatavilla, lähettää se vastauspaketin ja samalla varmistaa yhteyden toimivuuden. (Cisco DocWiki: Details of Technician Commands.)

**HASH -** Algoritmi, joka laskee tietylle merkkijonolle ns. tiivisteen ja käyttää sitä jonon eheyden varmistamiseen.

**RADIUS - R**emote **A**uthentication **D**ial **I**n **U**ser **S**ervice. Verkkoprotokolla, joka soveltuu sisäänsoittopalveluiden (esimerkiksi VPN) käyttäjän tunnistuksessa.

**SNMP - S**imple **N**etwork **M**anagement **P**rotocol. Verkkoprotokolla, jota käytetään verkkolaitteiden tilan tarkasteluun sekä hallintaan.

**NBAR** - **N**etwork **B**ased **A**pplication **R**ecognition. Ciscon oma metodi verkon liikenteen dynaamiseen tunnistukseen. (Cisco: Quality of Service Networking.)

**QoS** - **Q**uality **o**f **S**ervice. Termi, jota käytetään kuvaamaan verkon kykyä tarjota parempaa palvelua valituille protokollille. (Cisco: Quality of Service Networking.)

# 1 Johdanto

Tämän työn tehtävänä oli siirtyä Zyxelin reitittimestä Ciscon 871-sarjan reitittimeen. Työ pätee suurimmaksi osaksi myös kaikkiin muihin Ciscon reitittimiin 12.4 IOS -versioilla. Teoriaosuudessa käsitellään lähiverkkoprojektin suunnittelua ja sen vaiheita. Työn käytännön osuudessa esitellään erinäköisiä verkkoprotokollia ja niiden käyttötarkoituksia, niistä on myös lyhyt kuvaus termistössä. Tämä projekti keskittyy lähiverkon toteutukseen Ciscon reitittimellä ja sille valvonnan toteuttamiseen.

Työ on jaettu suunnitteluun, toteutukseen ja valvontaan. Suunnitteluvaiheessa käydään läpi senhetkistä toteutusta, mietitään, miten sitä voisi parannella, ja käydään läpi yleisesti suunnitteluvaiheen toimenpiteet. Toteutusvaiheessa käydään läpi lähiverkolle tarpeelliset komennot perusasetuksista lähtien. Siinä otetaan myös huomioon yrityksen tarpeet ja laajennetaan asetuksia sen mukaisesti. Viimeisenä käydään läpi verkon valvonnan perusteita ja sitä, kuinka tämän yrityksen verkkoa haluttiin valvoa.

Projektissa eniten aikaa vei lähiverkon toteutus ja sille toimivien asetusten konfigurointi. Perusasetusten jälkeen ryhdyttiin toteuttamaan lisätoimintojen käyttöönottoa. Samalla hoidettiin myös yrityksen verkon dokumentoinnin hoitaminen ajan tasalle huomatessamme sen olevan hyvin vajaa. Tämän toteutusmallin tarkoituksena on olla helposti ylläpidettävä, selkeä sekä täyttävän kaikki yrityksen sen hetkiset tarpeet jättäen tilaa myös tarpeen mukaan kasvulle yrityksen laajetessa.

Työ on rajattu näihin kolmeen vaiheeseen ja lopputuloksena oli yrityksen senhetkisiin tarpeisiin soveltuva lähiverkko. Käyn myös lopuksi läpi parannusehdotuksia verkon toteutukselle. Projektissa saatiin aikaan lähiverkko ja sille valvonta sekä valvonta yrityksen hallinnassa olevalle isommalle verkolle. Toteutus jätti tilaa myös kasvulle ja oli näin ollen odotusten mukainen.

# 2 Suunnittelu

Jokainen lähiverkkoprojekti lähtee suunnittelusta. Suunnittelu tulee tehdä mahdollisimman perusteellisesti, joskin täydellisen suunnitelman tekeminen ei ole kustannustehokasta, ja on lähes mahdotonta. Tärkeintä on saada laitteet fyysisesti järkeviin paikkoihin sekä kriittisimmät palvelut toimimaan. Käyttöönoton jälkeen tulee kerätä käyttäjiltä palautetta puuttuvista palveluista. Lähiverkkoa rakentaessa on tarpeellista selvittää verkon tärkeimmät tehtävät, jotta sen vaatimukset voidaan määritellä. Kukin verkon suunnitteluun osallistuva näkee verkon ensisijaiset tehtävät oman käyttökokemusten ja toimintaympäristönsä näkökulmasta. Esimerkiksi yrityksen näkökulma on erilainen kuin oppilaitoksen (Jaakohuhta: 275).

Tietoverkon suunnitteluun voidaan käyttää perinteistä systeemityön vaihejakomallia. Kyseisessä mallissa jaetaan verkon rakentaminen seitsemään eri vaiheeseen: esitutkimukseen, määrittelyyn, suunnitteluun, toteutukseen, testaukseen, käyttöönottoon sekä viimeisenä ylläpitoon. Nämä vaiheet eivät välttämättä seuraa toisiaan. Ne saattavat olla samanaikaisia ja on ehkä jaettu useammaksi aliprojektiksi, joilla kullakin on oma aikataulunsa.

Esitutkimuksessa kerätään projektin kannalta tarpeelliset dokumentaatiot, kuten esimerkiksi vanhan verkon kytkennät sekä asetukset. Sen lisäksi selvitetään verkolle tarpeelliset palvelut sekä ohjelmistot. Ohjelmistojen ohjekirjat on hyvä olla saatavilla, mikäli niille tulee tarvetta. Esimerkiksi verkon palomuuriasetuksia säädettäessä on hyvä tietää, mitä portteja kukin ohjelma käyttää. Määrittelyvaiheessa on tarkoituksena selvittää tietojärjestelmältä vaadittavat ominaisuudet. Tähän vaiheeseen kuuluvat muun muassa erilaiset kartoitukset sekä laite- että ohjelmistopuolelta, ja analyysit muun muassa verkon käytöstä sekä sen kuormituksesta. Suunnitteluvaiheessa pyritään löytämään vaihtoehtoisia ratkaisuja määrittelyssä selvitettyjen ominaisuuksien toteutukseen. Tärkein osa suunnittelussa on pohtia erilaisia toteutusvaihtoehtoja, niiden kustannuksia sekä toteutukseen kuluvan ajan määrää. Toteutusvaiheessa ryhdytään rakentamaan tietojärjestelmää ja siihen liittyvää tietoverkkoa, joka määriteltiin suunnitteluvaiheessa. Tässä vaiheessa joudutaan usein vielä muuttamaan suunnitelmia joidenkin yksityiskohtien osalta, mutta onnistuneen suunnitteluvaiheen jälkeen ei tulisi olla tarvetta suuriin muutoksiin. (Hakala - Vainio: 406-409.)

Testausvaiheessa käydään läpi tietojärjestelmän sekä siihen kuuluvan verkon toimintaa. Testaus on syytä toteuttaa järjestelmällisesti sekä tarpeen mukaan jaotella toiminnalliseen, määritystenmukaisuus- ja standardinmukaisuustestaukseen. Tämän vaiheen tarkoituksena on karsia suurin osa verkon ongelmista pois ja varmistaa verkon toimivuus. Käyttöönottovaiheessa tietojärjestelmä ja siihen kuuluvat laitteistot otetaan loppukäyttäjien käytettäväksi. Tähän vaiheeseen kuuluu aina tarkkailujakso, jonka aikana IT-ammattilaiset seuraavat järjestelmän toimintaa ja neuvovat sen käyttäjiä. Käyttöönottovaiheeseen kuuluu myös hyvin läheisesti käyttäjiltä tulevan palautteen käsittely. Vaikka projekti olisi hyvin suunniteltu ja toteutettu, voivat yksittäiset käyttäjät silti löytää puutteita järjestelmästä, ja riippuen niiden laadusta, saattaa olla tarpeellista korjata kyseessä olevat puutteet. Tietojärjestelmäprojektit päättyvät käyttöönottovaiheeseen. Käyttöönoton jälkeen alkaa kuitenkin tietojärjestelmän ja siihen kuuluvien verkkojen rutiininomainen ylläpito. Projektin valmistumisen jälkeen laaditaan dokumentit järjestelmään tehdyistä muutoksista ja laajennuksista. Projektin jatkokehityksen kannalta on hyvin tärkeää ylläpitää sekä tehdä mahdollisimman perusteelliset dokumentaatiot. Tulee kuitenkin miettiä, minkälaisia dokumentteja tehdään ja mitä tietoja otetaan ylös, sillä kaiken mahdollisen tiedon laittaminen esimerkiksi yhdelle Excel-sivulle ei ole viisas ratkaisu. (Hakala & Vainio: 409-411.)

Verkko tulee aina mahdollisuuksien mukaan ylimitoittaa, jotta saadaan yrityksen kasvaessa verkkoon helposti lisättyä käyttäjäpäätteitä tai muita verkon laitteita, ilman lisähankintoja varsinaiseen runkoverkkoon. Ylimitoitusta miettiessä kannattaa kuitenkin pitää mielessä mahdolliset lisäkustannukset sekä miettiä niiden hyötysuhdetta.

Tässä työssä oli toimiva lähiverkko käytössä, joten suunnitteluun ei käytetty suuremmin aikaa. Perustana käytettiin vanhaa lähiverkkoa ja siihen vaihdettiin reititin ja lisättiin lähiverkon palveluita. Toteutusta tehdessä huomattiin yrityksellä olevan hallussa muutamia tarpeettomia kytkimiä ja niitä käytettiin verkon laajentamiseen ilman lisäkustannuksia.

## 2.1 Kartoitus

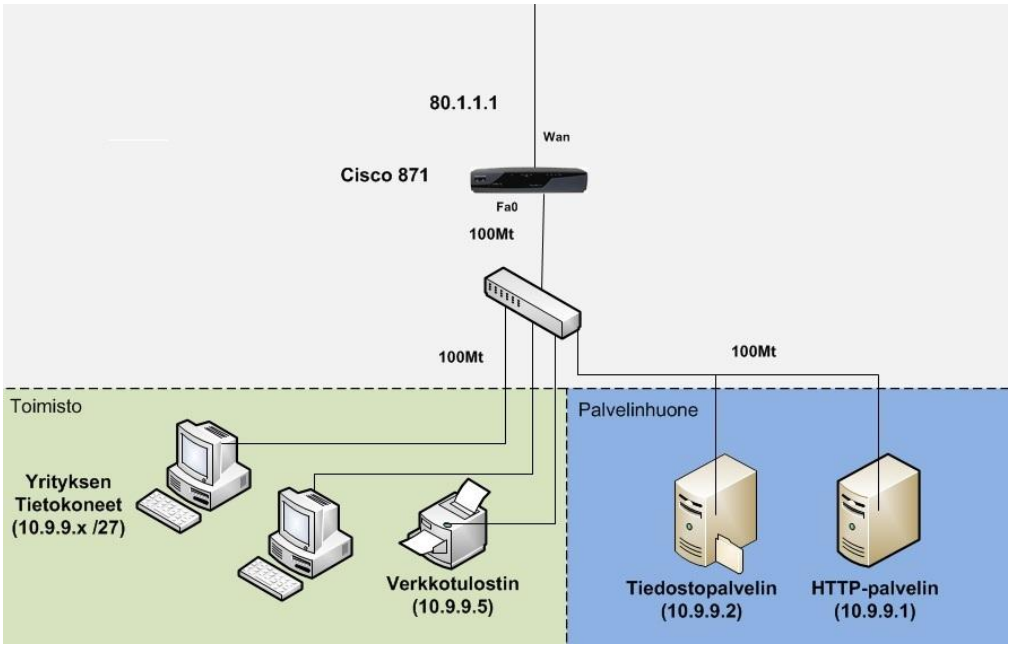
Yrityksen verkkoa kartoittaessa huomattiin, että dokumentointi sen hetkisestä verkosta oli hyvin heikko. Tästä syystä dokumentaation päivittämiseen kului runsaasti aikaa.

Kartoitusta aloittaessa oli selvillä, mitä laitteita verkkoon kuului, mutta merkintöjen vaillinaisuuden takia senhetkisen verkon rakenteen ja laitteiden paikallistamiseen kului ylimääräistä aikaa. Asiaa kuitenkin korjailtiin projektin edetessä ja kytkennöistä sekä laitteista tehtiin ajan tasalla olevat dokumentaatiot ja niitä päiviteltiin projektin edetessä.

Tämän yrityksen verkkoon kuului yksi tiedostopalvelin sekä yksi HTTP-palvelin. Näiden lisäksi oli toimistotiloissa käytössä neljästä seitsemään konetta (neljä pöytäkonetta ja kannettavia tietokoneita tarpeen mukaan) sekä toimiston tulostin. Internetoperaattorin vaihdon ja jatkon kasvumahdollisuuksien takia siirryttiin vanhasta Zyxelin reitittimestä monipuolisempaan ja vakaampaan Ciscon 871-reitittimeen. Tämä mahdollisti paremman verkon vianselvityksen, lisäsi ominaisuuksia verkkoon sekä antoi paremman hallittavuuden esimerkiksi sähköpostin spam-virusten hallinnointiin.

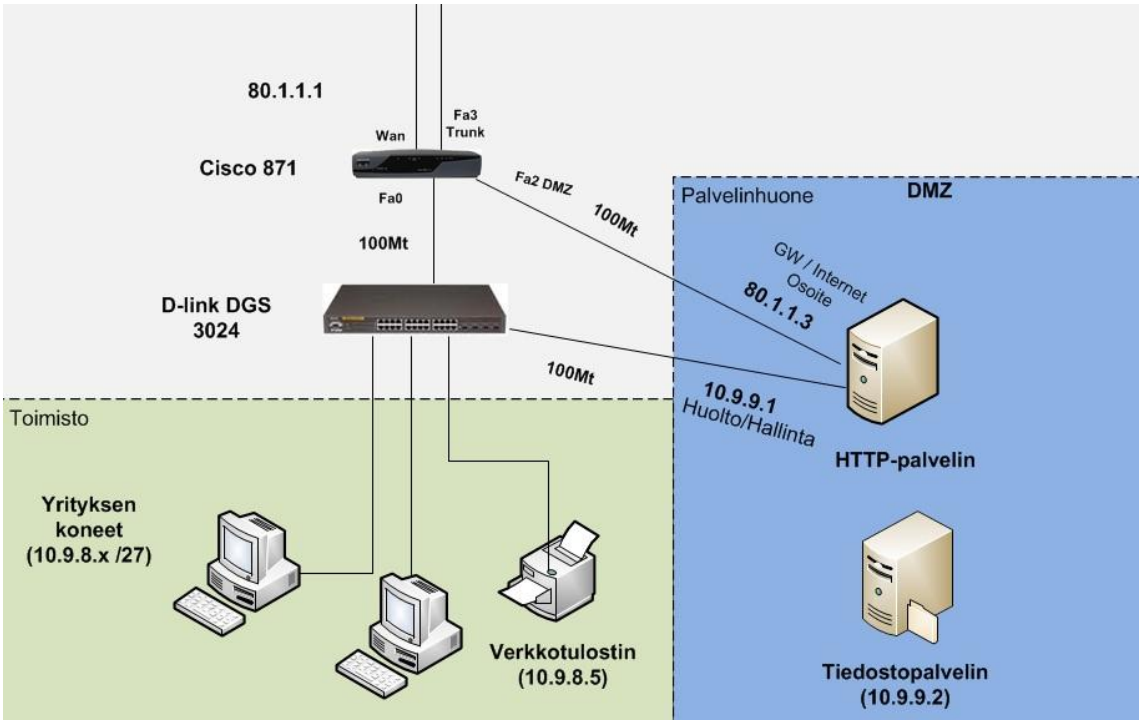
## 2.2 Prosessin vaiheet

Koska verkko tuli saada toimimaan mahdollisimman nopeasti, aloitettiin verkon pystytys alla olevan kuvan mukaisesti. Tässä vaiheessa verkon topologiaa ei muutettu vielä, vaan vaihdettiin vanha reititin uuteen.




Kuva 1. Yrityksen sisäverkon kaavio. Ensimmäinen iteraatio.

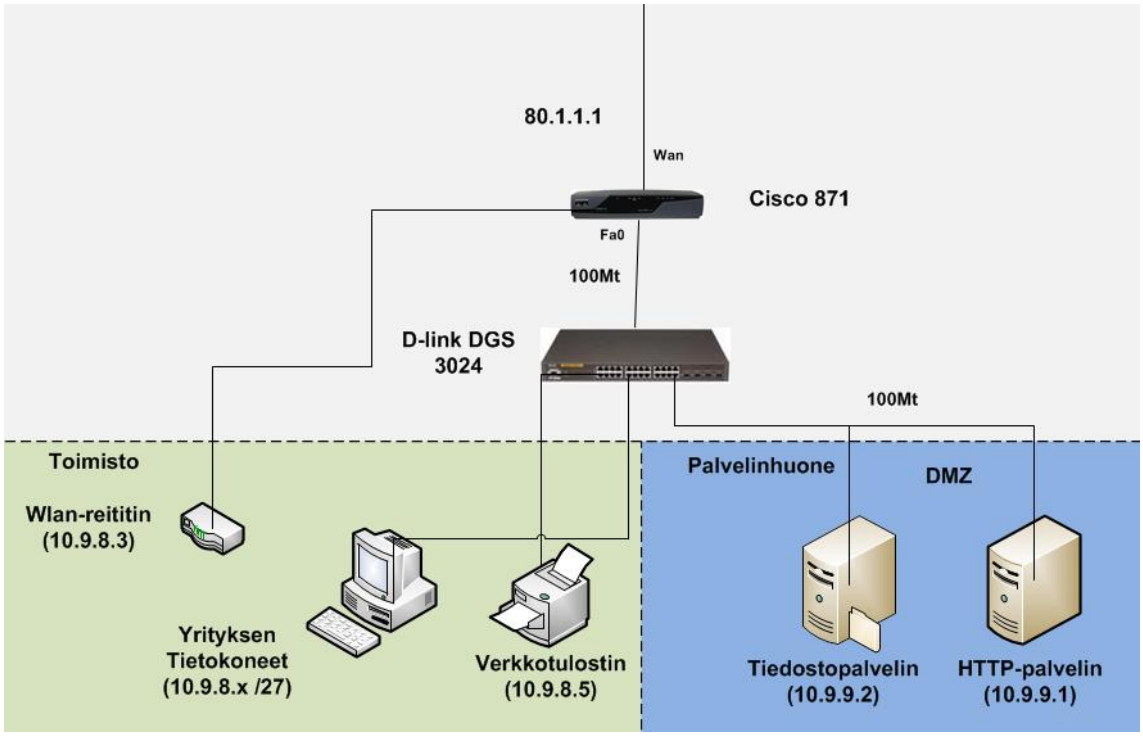
Tällä saatiin kaikki kriittisimmät palvelut ja toiminnot päälle nopeasti. Kyseisessä kokoonpanossa ilmeni kuitenkin nopeasti ongelmia yrityksen internetsivujen kanssa. Yrityksen verkkosivut toimivat muuten asiallisesti, mutta sisäverkosta ei päässyt niihin käsiksi, ellei syöttänyt sisäverkon osoitetta manuaalisesti hosts.ini-tiedostoon tai syöttänyt sitä selaimen osoiteriville. Tätä koetettiin korjata seuraavan kuvion osoittamalla tavalla vian löydyttyä. Vika ei niinkään ollut verkon mallissa vaan sen toteutuksessa, josta lisää myöhemmin. Seuraavan kuvion osoittama muutos oli sillä hetkellä paras vaihtoehto.




Kuva 2. Yrityksen sisäverkko, toinen iteraatio.

Tämän muutoksen tarkoituksena oli antaa web-palvelimelle oma julkinen IP, jonka toivottiin auttavan edellä mainitun ongelman verkkosivujen kanssa. Koska Ciscon reititin ei suostu reitittämään sisäverkosta ulkoisen DNS-palvelimen kautta takaisin sisäverkkoon, koetettiin sitä korjata kuvan mukaisesti. Nopeasti huomattiin, että tämä ei tilannetta parantanut, joten teimme tilapäisen korjauksen muuttamalla verkon osoitekäännöksen toteutustapaa. Tämä korjaus kuitenkin tuotti lisäongelmia muun muassa VPN-tunneleiden kanssa, joten pysyvämpi ratkaisu tuli keksiä. Ongelmiin löydettiin ratkaisu, kun tiedostopalvelimesta tehtiin myös yrityksen verkon DNS-palvelin. Verkon hallinnan

helpottamiseksi ja toiminnan parantamiseksi siirryttiin samalla standardin mukaiseen DMZ-alueeseen. Tätä vaihetta toteuttaessa siirryttiin vanhasta kytkimestä D-linkin DGS 3024 -malliin, joka laajensi verkkoporttien määrää 24:stä 48:aan ilman erillisiä lisähankintoja. D-linkin kytkin oli ollut tarpeettomana yrityksen toisen toimipisteen varastossa, joten päätettiin ottaa se käyttöön lisäporttien takia.




Kuva 3. Yrityksen sisäverkko, viimeisin iteraatio

Tällä muutoksella saatiin aikaan verkon vakaannuttaminen, palveluiden toimivuus ja parannettu hallittavuus. Normaalisti ei ole hyötyä tai tarvetta olla tiedostopalvelimia DMZ-alueella, mutta tässä tapauksessa päätettiin jättää tiedostopalvelin kyseiselle alueelle ajan ja vaivan säästämiseksi. Päätöstä tehdessä otettiin huomioon, että DMZ-alueelle ei pääse ulkoapäin, paitsi yrityksen verkkosivujen selaamista varten. Yrityksen käyttöön otettiin myös wlan-reititin, jotta saatiin tarpeen mukaan testattua yrityksen hallinnassa olevan verkon julkisia IP-osoitteita ja saatiin langattomat laitteet liitettyä lähiverkkoon.

# 3 Toteutus

Lähiverkon toteutus aloitetaan hankkimalla oikeanlaiset laitteet verkon suunnittelun ja kartoituksen jälkeen. Tässä tapauksessa laitteet oli etukäteen hankittu ja tarpeen mukaan otettiin käyttöön valmiina yrityksen hallussa olevia laitteita, kuten D-link DGS 3024 -kytkin. Kytkin on layer 2 -laite, joka käytännössä tarkoittaa sitä, että sitä ei tarvitse erikseen konfiguroida, joten suurin osa ajasta meni Ciscon 871-sarjan reitittimen kanssa.

## 3.1 Perusasetukset

Reitittimen asennus aloitettiin peruskonfigurointikomennoilla. Ensimmäiseksi sille annettiin verkossa helposti tunnistettava nimi komennolla *hostname <laitteen verkkonimi>*. Tämän jälkeen rajattiin laitteen asetusten hallitsijoita laittamalla salasana komentoriville pääsemiseksi komennolla *enable secret <salasana>.* Salasanat ovat hyvä suojata komennolla *service password-encryption*. Tämä salaa kaikki reitittimessä olevat salasanat, jotta niitä ei voi lukea suoraan konfiguraatiotiedostosta. Tämä salaus on kuitenkin heikko ja murrettavissa helposti, mutta suojaa kuitenkin nopealta läpiluvulta. Verkkolaitteiden asetuksia sisältävät tiedostot tulee tästä huolimatta pitää turvallisessa paikassa väärinkäyttöjen ehkäisemiseksi. Jos verkossa on käytössä Windows toimialue (domain), on se hyvä ilmoittaa myös reitittimelle: *ip domain name <toimialueen nimi>* tämä auttaa reititintä toimimaan muiden toimialueella olevien laitteiden kanssa. Perusasetuksia säätäessä on hyvä laittaa myös reitittimen kello oikeaan aikaan. Tämä auttaa muun muassa virheiden etsinnässä. Kellon voi laittaa toimimaan joko NTP-protokollan avulla: *ntp server <IP-osoite>* tai syöttämällä se reitittimelle manuaalisesti: *clock timezone GMT 2, Clock summer-time CEST.* Tässä tapauksessa ei käytetty NTP-palvelinta, kellon tarkkuuden ollessa sivuseikka. Kelloa ei tässä tilanteessa käytetty kuin debug-komentoja tarkastellessa. Suomalaisia palvelimia löytyy esimerkiksi NTP Pool Project:in kotisivuilta (Hansen: NTP Pool Project).

## 3.2 Internetyhteyden luonti

Seuraavaksi ryhdyttiin asentamaan asetuksia, joilla mahdollistetaan lähiverkon sekä internetyhteyden toimivuuden. Ensimmäiseksi laitettiin kuntoon ulospäin menevän,

julkisen IP-osoitteen portti (WAN-port): *interface fastethernet 4*. Tällä valittiin oikea portti, jonka jälkeen siihen laitettiin IP-osoite aliverkkomaskineen *ip address 80.1.1.1 255.255.255.248.* Tämän toimivuutta testattiin käyttämällä ping-komentoa julkisen IP-verkon yhdyskäytävään. Yhdyskäytävä vastasi, joten osoite oli kunnossa. Jos kyseinen osoite olisi ollut jo käytössä, olisi yhdyskäytävä lähettänyt virheilmoitusta päällekkäisistä IP-osoitteista.

Jotta päästäisiin sisäverkkoon päin olevista ethernet-porteista WAN-portin kautta internetiin, tulee verkkoon konfiguroida virtuaalilähiverkko (vlan), jota ethernet-portti käyttää: *interface vlan12, ip address 10.9.9.1,* virtuaalilähiverkon osoitteeksi on tapana laittaa joko ensimmäinen tai joku viimeisistä osoitteista lähiverkon osoitteistosta (pool). Tämä helpottaa asetusten muistamista tilanteissa, joissa joudutaan käsin syöttämään tiedot päätelaitteille tai esimerkiksi internetyhteyden toimivuuden testailussa. Kyseistä portin liikennettä voidaan myös halutessaan rajoittaa luomalla siihen oma pääsylista (access-list) ja laittamalla se porttiin *ip access-group <pääsylistan nro tai nimi> <in/out>.* Tämä komento on hyödyllinen muun muassa portin 25, joka on sähköpostin lähetys- sekä vastaanotto-portti, hallinnassa. In-lisäyksellä pääsylista vaikuttaa sisääntulevaan liikenteeseen kun taas out-lisäys vaikuttaa ulospäin menevään liikenteeseen. Hallinnan tärkeydestä on hyvänä esimerkkinä spam-viruksen joutuminen lähiverkkoon. Jotta se ei häiritsisi muuta internet-liikennettä, voidaan sen käyttämän portin toimintaa tilapäisesti rajoittaa, kunnes viruksesta päästän eroon. Seuraavaksi määritellään, mitkä portit kuuluvat edellä määriteltyyn virtuaalilähiverkkoalueeseen: *interface fastethernet 0,* switchport access vlan12.

### 3.3 DHCP-asetukset

Verkon toimivuuteen liittyy oleellisesti myös Dynamic Host Configuration Protocol -palvelu. Tämän ansiosta saadaan syötettyä verkon laitteille dynaamisesti oikeat verkkoasetukset. Sillä säästetään aikaa uusien verkkolaitteiden asentamisessa verkkoon. Joillekin laitteille tulee kuitenkin määritellä pysyvät asetukset, jotta niitten käyttö olisi ongelmatonta, tässä tapauksessa kyseisiä laitteita oli kolme, verkon tiedostopalvelin, HTTP-palvelin sekä tulostin. Tiedostopalvelimen sekä HTTP-palvelimen IP-osoitteet olivat jo määritelty palvelimilla eikä niitä tarvinnut erikseen laittaa reitittimen asetuksiin. DHCP määritellään seuraavasti: Ensiksi aloitetaan kieltämällä osoitteet, jotka halu-

taan varata tulevaisuuden käyttöä varten staattisiksi osoitteiksi tai ne osoitteet, jotka ovat jo käytössä *ip dhcp excluded-address <alin osoite><ylin osoite>.* Esimerkiksi *ip dhcp excluded address 10.9.8.1 10.9.8.10*. Tähän komentoon ei siis kuulu verkkomaski, jos sen laittaa epähuomiossa jälkimmäisen osoitteen tilalle, kieltää komento liikaa osoitteita.

Kiellettyjen osoitteiden jälkeen määritellään verkon asetukset: *ip dhcp pool vlan12,* nimen tulisi olla helposti yhdistettävissä siihen verkkoon, jota se käsittelee. Seuraavaksi määritellään DHCP-osoitteiston osoitteet: *network <verkon ensimmäinen IP-osoite> <aliverkkomaski>*. Verkon yhdyskäytävä: *default router <verkossa olevan rajapinnan osoite>*. Sille mahdolliset nimipalvelimet määritellään komennoilla *dns-server <osoite 1> <osoite 2>*. Jos käytössä on Windows-toimialue, on se hyvä lisätä myös DHCP-asetuksiin: *domain-name <verkkoon kuuluva dns-liite (suffix)>.* Näiden komentojen lisäksi voidaan tehdä DHCP-osoitteistoja yksittäisille osoitteille. Jos samaan verkkoon tehdään useampi kuin yksi osoitteisto (esimerkiksi yksittäisille osoitteille), perii nämä ominaisuudet ensisijaiselta osoitteistolta. Tässä tapauksessa haluttiin luoda printterille oma osoite, joka onnistuu komennoilla: *host <osoite> <aliverkkomaski>, client-identifier <mac-osoite>.*

Tämän verkon asetukset:
Ip dhcp pool vlan12
Network 10.9.8.1 255.255.255.0
Default router 10.9.8.1
Dns server 10.9.9.2 8.8.8.8
Host 10.9.8.2 255.255.255.0
Client-identifier 0123.4567.89ab

### 3.4 NAT/PAT

Jotta internetyhteys toimisi useammasta kuin yhdestä koneesta, tulee käyttää osoite-käännöstä. Sen avulla saadaan useampi kone saman julkisen IP-osoitteen alle, mikä puolestaan mahdollistaa julkisen verkon käytön jokaiselta koneelta. Sitä varten tulee määritellä osoitelista, jossa määritellään sitä käyttävät sisäverkon IP-osoitteet. Osoite-lista määritellään seuraavalla tavalla: *ip access-list extended <nimi tai numero>* tai

*access-list <numero>,* tällä luodaan pohja listalle ja nimetään se. Sen jälkeen määritellään osoitteet, joihin listaa sovelletaan: *permit ip <10.9.8.0> <0.0.0.255> any*, tämä sallii kaikkien 10.9.8.0-verkon osoitteiden käännöksen. Sen jälkeen poistutaan pääsylistan konfiguroinnista exit-komennolla ja syötetään peruskonfiguraatiotilaan (configurate terminal) komento, jossa määritellään, mistä listasta katsotaan IP-osoitteet, missä portissa käytetään osoitekäännöstä ja onko kyse sisäisestä vai ulkoisesta portista: *ip nat inside source list <nimi> interface <portin nimi, esim. Fasthethernet 4> overload.* Sanalla overload komennon perässä tarkoitetaan, että käytetään myös Port Address Translation -protokollaa. PAT-protokolla sallii useamman kuin yhden koneen käyttää samaa julkista IP-osoitetta käyttämällä jokaiselle koneelle omia portteja. Reititin merkitsee NAT-tauluun koneet ja portit jotka lähettävät pyyntöjä julkiseen verkkoon. Seuraamalla NAT-taulua reititin osaa reitittää takaisin tulevat paketit oikeille koneille, joka mahdollistaa julkisen verkon toiminnan yhdellä julkisella IP-osoitteella. Protokollan heikkoutena on se, että julkisesta verkosta ei saada yhteyttä koneisiin, jotka ovat NAT/PAT-protokollan alla, ilman että reitittimelle kerrotaan erikseen portinohjauksilla, minne halutaan kunkin portin liikenteen johtavan.

Tämän jälkeen määritellään reitittimelle, mitkä portit ovat sisäverkon portteja ja mitkä ulkoverkon portteja. Nämä kertovat reitittimelle, mistä sisäverkon liikenne tulee ja mistä se menee julkiseen verkkoon. Tämä auttaa reititintä reitittämään paketteja oikeisiin portteihin ja on pakollinen osa NAT/PAT-konfigurointia. Osoitekäännöstä käyttäviin portteihin laitetaan joko komento *ip nat inside*, tai *ip nat outside*, riippuen siitä, kummassa suunnassa portti on. Tässä työssä WAN-porttiin tulee komento: *ip nat outside* ja vlan 12 -virtuaaliporttiin *ip nat inside*. Näin mahdollistetaan internetyhteyden toimiminen vlan 12:sta sijaitsevissa laitteissa.

Sisäverkon HTTP-palvelimelle luotiin oma DMZ ja sinne luotiin sekä oma sisäverkko-osoitteisto että vlan (Kappaleessa 3.3 esitetyllä tavalla) ja oma NAT-osoite: *ip nat pool DMZ 80.1.1.3 <viimeinen julkinen osoite> netmask 255.255.255.248*. Osoitekäännöstä käyttäville osoitteille luotiin oma pääsylista, joka salli vain demilitarisoidun alueen (DMZ) sisällä olevat koneet: *ip access list dmz*, *permit 10.9.9.0 0.0.0.255*. Tämän jälkeen määriteltiin NAT/PAT: *ip nat inside source list <DMZ> <dmz> overload*. Viimeisenä määriteltiin kyseinen virtuaalilähiverkon portti sisäiseksi osoitekäännösportiksi *ip nat inside* -komennolla.

### 3.4.1 NAT/PAT-portinohjaukset

Portinohjauksia käytetään ohjaamaan internetistä tulevaa liikettä portin perusteella oikeaan lähiverkkosoitteeseen. Esimerkiksi jos lähiverkossa on sähköpostipalvelin, voidaan kyseisen julkisen IP-osoitteen portin 25 liikenne ohjata oikeaan lähiverkon laitteeseen oikealla porttiohjauksella. Portinohjausta voidaan käyttää myös ilman osoitekäännöstä, mutta tässä työssä sitä käytetään, koska osoitekäännös on tarpeellinen lähiverkon toimivuuden kannalta.

Portinohjaukset voidaan hoitaa useammalla eri tapaa, joko määritellään, mistä laitteen portista tulee liikenne sisään, tai mikäli käytössä on useampi julkinen IP-osoite, voidaan kontrolloida liikennettä myös sen perusteella. Jos halutaan ohjata liikennettä laitteen julkisen IP-osoitteen portin kautta, onnistuu se komennolla *ip nat inside source static tcp <sisäisen osoitteen ip> <portti> interface <laitteen portti esim. fastethernet4> <portti>*. Vaihtoehtoisesti voidaan määritellä ohjaus ulkoisen IP:n kautta: *ip nat inside source static tcp <sisäisen osoitteen ip> <portti> <ulkoinen osoite> <portti>.* Tässä työssä käytettiin komentona *ip nat inside source static tcp 10.9.9.2 25 80.1.1.3 25.*

### 3.4.2 Epävirallinen osoitekäännöstapa

Kävimme läpi työn edetessä erilaisia ratkaisuja internetsivujen ongelmien korjaamiseksi. Yksi näistä ratkaisuista oli ulkopuolisen konsultin ehdottaman epävirallinen osoitekäännöstapa. Kokeilimme epävirallista tapaa kuvassa 1 ja kuvassa 2 esitettävien kytkentöjen ollessa käytössä. Tätä tapaa ei suositella käytettäväksi pitempiaikaisena ratkaisuna. Sillä koetettiin korjata lähiverkko nopeasti ilman sen suurempia muutoksia. Ongelma, jota tällä pyrittiin korjaamaan oli se, että sisäverkosta ei päässyt julkisella IP-osoitteella verkossa sijaitsevan HTTP-palvelimen hostaamalle verkkosivulle, koska reititin ei suostu reitittämään liikennettä takaisin samaan verkkoon, mistä se on tullut. Ongelman korjaus olisi myös onnistunut muokkaamalla jokaisen koneen hosts.ini-tiedostoa, mutta se puolestaan aiheuttaisi lisävaivaa liikkuvien verkkolaitteiden kanssa. Tällä ratkaisulla saatiin korjattua tilanne, mutta samalla selvisi myös, miksi kyseistä tapaa ei suositella käytettäväksi. Vaikka tämän jälkeen pystyttiin selaamaan verkkosivuja myös www-osoitteen kautta, esti se muun muassa ssh- sekä ipsec/l2tp -

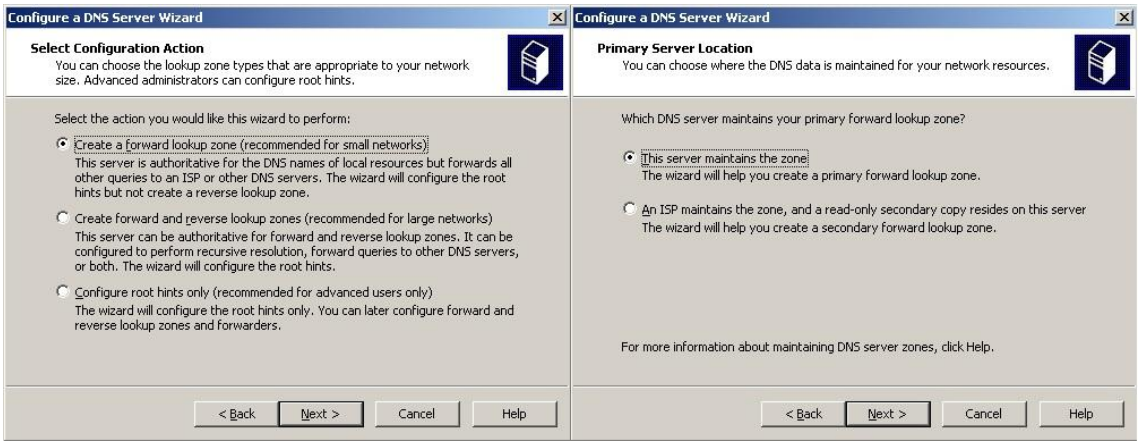
yhteydet sisäverkosta. Näitä käytettiin asiakkaiden laitteiden etähallintaan ja näin ollen ne olivat yritykselle kriittisiä protokollia. Tätä puolestaan koetettiin korjata käyttämällä osoitekäännöstä yrityksen verkon julkiseen IP-osoitteeseen. Sen toteuttamiseen käytettiin yrityksen hallinnassa ollutta pääreititintä, jonka avulla saatiin yrityksen osoitteet käännettyä. Osoitekäännöksen toteuttamiseen piti luoda myös staattinen reitti takaisin yrityksen sisäverkkoon: *ip route 10.9.8.0 255.255.255.0 80.1.1.1*. Tämä ei kuitenkaan toiminut halutulla tavalla, joten osoitekäännös laitettiin takaisin Ciscon 871:een. Ongelma sijaitsi väärässä reitityksessä, eli pääreititin ei osannut ohjata liikennettä oikein takaisin. Yllä olevan oletus-reitin ja osoitekäännöksen poistaminen auttoi asiaan ja palattiin takaisin epäviralliseen osoitekäännökseen. Sitä käytettiin useampi viikko pysyvämpää ja toimivampaa ratkaisua miettiessä.

Konfigurointi hoidetaan muuten samalla tavalla kuin virallinen ja suositeltava tapa, mutta siitä jätetään määrittämättä sekä inside- että outside-verkkoportit. Sen sijaan, että määritellään erikseen sisäiset ja ulkoiset portit laitetaan jokaiseen osoitekäännöstä käyttävään porttiin vain *ip nat enable*. Esimerkiksi: *ip nat source list <nimi> interface <verkkoportin nimi, esim. Fasthethernet 4> overload*, tämän jälkeen asetetaan osoitekäännöstä käyttävät portit: *interface fastethernet 4*, *ip nat enable* ja *interface vlan 12, ip nat enable*. Tämä mahdollistaa käännöksen toimivuuden, mutta se todennäköisesti aiheuttaa enemmän ongelmia kuin mitä ratkaisee. Tässä työssä sitä käytettiin tilapäisenä ratkaisuna. Viimeisimmässä lähiverkon toteutuksessa (kuva 3) tästä luovuttiin ja implementoitiin virallinen tapa. Paras ratkaisu internetsivujen toimivuuden parantamiseen oli ottaa käyttöön omassa verkossa sijaitseva DNS-palvelu.

### 3.5 DNS-palvelu

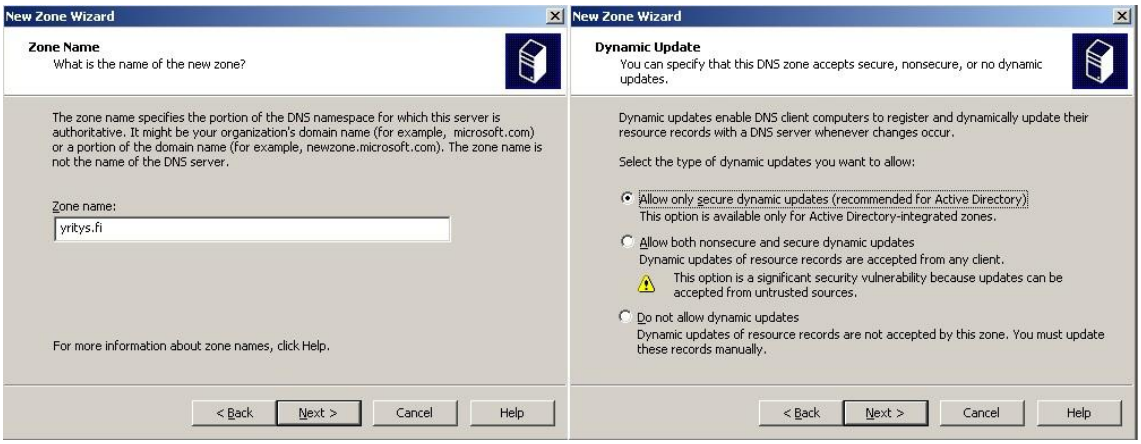
Domain Name System -protokollan tarkoituksena on helpottaa muun muassa internetin selausta mahdollistamalla nimien käytön IP-osoitteiden sijaan, esimerkiksi osoite 173.194.35.184 voidaan kääntää nimelle google.com. Sillä luodaan myös mekaniikka, jolla voidaan nimetä resursseja niin, että niitä voidaan käyttää eri verkkolaitteiden, protokollien, verkkojen sekä organisaatioiden välillä (Mockapetris: Domain names - implementation and specification). Tässä työssä käytimme nimipalvelua parantamaan verkon toimivuutta. Otimme käyttöön verkon sisäisten osoitteiden määrittelyn ja jätimme verkon ulkoisten DNS-osoitetietojen hallinnan yrityksen ulkopuolisille DNS-palvelimille.

Yrityksellä oli käytössä Windows server 2003, joten lisäsimme siihen DNS-palvelun. Seuraavat kuvat illustroivat prosessin etenemistä Server 2003 -version avustajalla.




Kuva 4. DNS server -luontivelhon ensimmäinen ja toinen vaihe

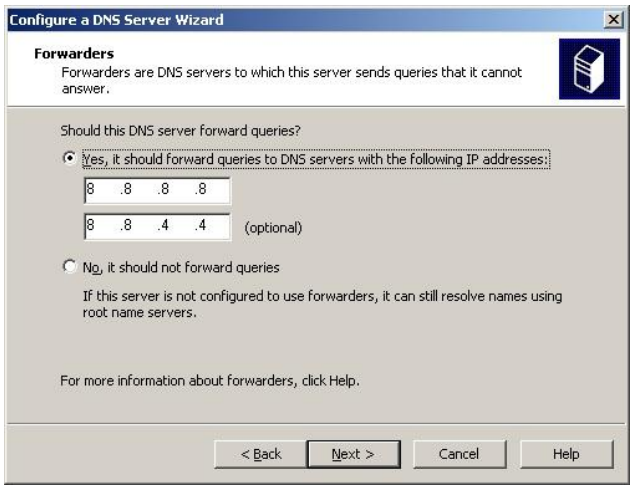
Asennuksen ensimmäisiin vaiheisiin kuuluu DNS-alueen valinta. Tässä tapauksessa otimme käyttöön Forward Lookup Zonen, joka mahdollistaa omien osoitteiden lisäämisen tarpeen mukaan sekä kaikkien muiden osoitteiden tietojen hakemisen ulkoiselta palvelimelta. Sen jälkeen määriteltiin palvelimen hallitsevan aluetta, edellä mainitusta syystä.




Kuva 5. DNS server -luontivelhon kolmas ja neljäs vaihe

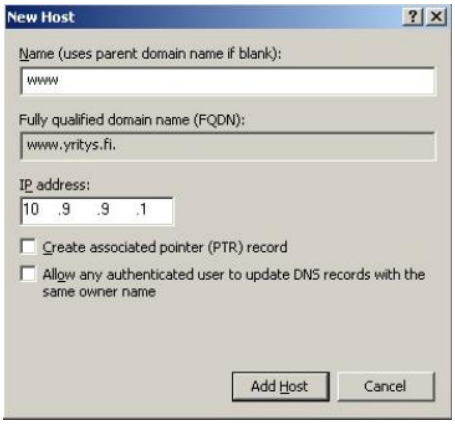
Seuraavana määritellään yritykselle DNS-alue, jonka alle voidaan määrittää kyseisessä alueessa sijaitsevat koneet, sekä tässä tapauksessa verkkosivulle oma IP-osoite. Sen jälkeen valitaan turvalliset dynaamiset päivitykset Active Directory käyttäjille. Tämä

mahdollistaa nopean automaattisen päivityksen käyttäjille DNS-palvelimen tietojen päivittyessä.




Kuva 6. DNS server -luontivelhon viimeinen vaihe

Viimeiseksi määritellään, mikäli palvelin hakee DNS-tietoja myös ulkoisilta palvelimilta. Tähän on lähes aina laitettava ulkoisen DNS-palvelimen osoite, mikäli kyseessä ei ole palvelin, johon määritellään lähes kaikki internetosoitteiden tiedot. Näiden jälkeen velho näyttää yhteenvedon asetuksista. Mikäli kaikki on kunnossa, painetaan "Finish"-nappulaa. Seuraavaksi palvelin luo DNS-palvelun ja sen jälkeen sille voidaan säätää tarkempia asetuksia.




Kuva 7. DNS-tietueen luonti palvelimelle

Viimeisenä vaiheena oli yrityksen internetsivun DNS-tietueen lisäys palvelimen tietoihin. Tämän jälkeen pystyttiin selaamaan yrityksen verkkosivuja ongelmitta sekä verkon ulko- että sisäpuolelta.

## 3.6 VPN-tunnelin luonti (point-to-point)

Virtual Private Network on protokolla, jolla yhdistetään kaksi tai useampi lähiverkko yhteen julkisen IP-verkon yli muodostamalla tunneli näiden pisteiden välillä. VPN-tunnelin liikenne voidaan salata, jolloin ulkopuolisten on vaikea päästä siihen käsiksi. Tunneleita voidaan luoda monella tapaa kuten paikasta paikkaan, paikasta moneen paikkaan tai monesta paikasta yhteen paikkaan. Tässä tapauksessa otettiin käyttöön paikasta paikkaan (point-to-point) oleva VPN-tunneli. Tällä saatiin etäyhteys toimimaan erään toisen yrityksen palvelimiin. Yhteyttä käytettiin yhteisprojektin jaettujen tiedostojen käyttöön.

Tunnelin luonti on monimutkainen ja yleensä ongelmallinen prosessi. Tekoa nopeuttaa se, että yhdellä henkilöllä on mahdollisuus säätää molempia tunnelin päitä samanaikaisesti. Tämän projektin VPN-tunnelin luontia hidasti se, että yhteys toimi menosuuntaan niin kuin kuuluisikin, mutta takaisin tuleva liikenne ei suostunut toimimaan kunnolla. Muutaman päivän selvittelyn jälkeen selvisi, että vastapuolella oli ollut erillinen palomuuri välissä, joka esti yhteyden oikein toimimisen. Sen asetusten muutoksien jälkeen yhteys alkoi toimia.

### 3.6.1 Tunnelin ensimmäinen luontivaihe

Tunnelin luonti aloitetaan ISAKMP (Internet Security Association and Key Management Protocol) -prosessin ensimmäisen vaiheen luomisella. Ensiksi luodaan salauskäytäntö *crypto isakmp policy <tärkeysnro>*. Tämän alle määritellään käytettävä salaus *encryption <3des/aes/des>,* todennusmenetelmä *authentication <pre-share/rsa-encr/rsa-sig>*. Todennus voidaan hoitaa joko jaetulla avaimella tai rsa-sertifikaateilla. Seuraavaksi määritellään käytettävä tarkiste: *hash <md5/sha>.* Huomioitavaa tässä on se, että hash sha -komento ei näy tarkastellessa käytössä olevia asetuksia *show running configuration* -komennolla. Sen jälkeen määritellään Diffie-Hellman-avaimenvaihtoryhmä: *group <1/2/5>*. Kun tunneli yhdistyy, luovat molemmat laitteet

niin sanotun security association -yhteyden, jonka avulla tunnelin päät pystyvät luottamaan toisiinsa ja luomaan salatun yhteyden. Tälle määritellään yleensä elinaika, mutta se ei ole pakollinen: *lifetime <sekunneissa>*. Elinaika pitää huolen siitä, että yhteys pysyy turvallisena, sen loppuessa joutuvat laitteet tarkastamaan uudestaan toimiiko yhteys ja vieläkö vastapään asetukset ovat kunnossa. Viimeiseksi luodaan yhteydelle tunniste, jolla tunnelin päät varmistuvat vastapuolen oikeudesta luoda yhteys. Tähän voidaan käyttää joko ennalta jaettua avainta (preshared key), kolmannen osapuolen varmennusta tai rsa-avainta. Tässä yhteydessä oli käytössä jaettu avain: *crypto isakmp key <avain> address <vastapään osoite>*. Tämän jälkeen siirrytään seuraavaan vaiheen luontiin. Vasta kun molemmat vaiheet on luotu, voidaan testata yhteyden toimivuutta.

### 3.6.2 Tunnelin toinen luontivaihe

Tunnelin luonnin toisessa vaiheessa määritellään salaus ja liikenteelle tarkiste. Ensimmäiseksi määritellään salaukselle käytettävä "transform-set", jolla määritellään ryhmä attribuutteja, joita käytetään tunnelin luonnin yhteydessä. Komennolla *crypto ipsec transform-set <nimi> <käytettävät tyypit, esimerkiksi: esp-3des esp-sha-hmac>* saadaan aikaan tarvittavien protokollien määritys. Tästä jatketaan luomalla pääsylista osoitteille, joiden sallitaan käyttävän tunnelia *access-list <nro>* tai *ip access-list extented <nimi>.* Tämän alle laitetaan *permit ip <halutun osoitteiden IP-verkko> <vastapään ip-verkko>*. Eli oman verkon ollessa 10.9.8.x ja vastapään esimerkiksi 11.9.8.x tulee komennoksi *permit ip 10.9.8.0 0.0.0.255 11.9.8.0 0.0.0.255*. Seuraavaksi luodaan salausta varten kartta, jossa määritellään tunnelin toinen pää ja siihen sallitut verkko-osoitteet edellä määritetyssä pääsylistassa ja otetaan salaukset käyttöön: *crypto map <nimi> <tärkeysnro> ipsec-isakmp.* Määritellään vapaaehtoinen kuvaus *description <kuvaus>*. Asetetaan vastapään osoite *set peer <vastapään julkinen IP-osoite>,* otetaan salaus käyttöön: *set transform-set <edellä tehdyn transform-setin nimi>* sekä määritellään siihen sallitut osoitteet: *match address <edellä tehdyn pääsylistan nro/nimi>*. Tämän jälkeen otetaan juuri tehty kartta ja liitetään se ulospäin menevään porttiin: *interface fastEthernet 4, crypto map <mapin nimi>.* Lopuksi tulee muistaa poistaa kyseinen yhteys osoitekäännöksestä: *(ip) access-list <nat-listan nimi>*, *deny ip 10.9.8.0 0.0.0.255 11.9.8.0 0.0.0.255*. Mikäli asetukset ovat molemmissa päissä yhtenevät ja internetyhteys toimii, tulisi VPN-yhteyden toimia tämän jälkeen. Häiriötilan-

teissa voidaan käyttää Ciscon omaa debug-järjestelmää. Esimerkiksi debug crypto isakmp sa -komennolla saadaan näkyviin tunnelin luonnissa tapahtuvat virheet, kuten esimerkiksi turvallisuusyhteyden (security association) luomisvirheet. *Crypto isakmp sa* puolestaan kertoo kaikki sillä hetkellä käytössä olevat turvallisuusyhteydet (Cisco: Security Commands).

## 3.7 PPTP/L2TP-yhteys yrityksen lähiverkkoon

Yrityksellä oli tarvetta saada etäyhteys ulkoverkosta yrityksen sisäisessä verkossa sijaitsevaan tiedostopalvelimeen. Tähän käytettiin PPTP/L2TP-protokollia. PPTP (Point-To-Point Tunneling Protocol) on VPN-protokolla, joka on kehitetty PPP-protokollan päälle. Tällä mahdollistetaan suorat yhteydet kahden verkkolaitteen välillä. PPTP on mahdollistanut muiden kuin TCP/IP-protokollan yhtäaikaisen liikenteen, ja näin ollen sallii turvallisen VPN-yhteyden. Protokollaa käytetään päätelaitteiden etäyhteyksissä.

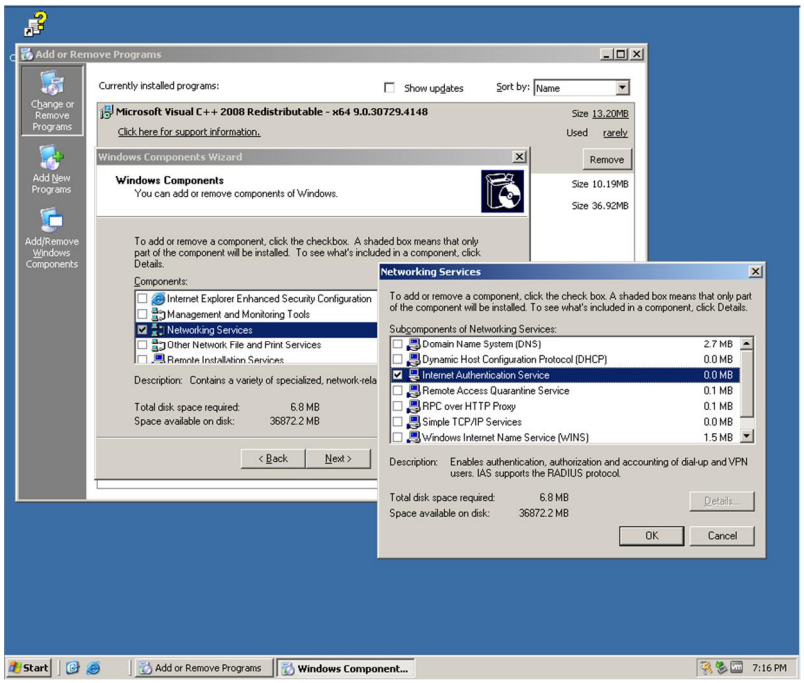
VPN-yhteyden konfigurointi aloitetaan ottamalla käyttöön Virtual Private Dial-up Network (VPDN) laitteessa: *vpdn enable*. Seuraavaksi luodaan VPDN-ryhmä, jossa määritellään yhteyden asetukset: *vpdn-group <numero>, accept-dialin, protocol pptp, virtual template <nro>.* Tämän jälkeen tehdään osoitteisto, josta etäyhteyden läpi tulevat tietokoneet ottavat osoitteen: *ip local pool <poolin nimi> <osoitteet jotka annetaan VPN:n kautta tuleville koneille>*. Tässä tapauksessa otettiin osoitteet lähiverkon loppupäästä: *ip local pool vpn-clients 10.9.8.220 10.9.8.245.*

Jotta saadaan yhteydenluontioikeudet vain halutuille henkilöille, tulee määrittää käyttäjät, joille annetaan oikeus käyttää verkon resursseja. Tämä voidaan hoitaa joko reitittimellä tai erillisellä radius-palvelimella. Tässä tapauksessa käytimme sekä yrityksen sisäistä radius-palvelinta, että hätätilanteita varten luotuja reitittimen paikallisia tunnuksia: *aaa authentication ppp default local, aaa authorization network default if-authenticated, aaa authorization network users local.* Yhteyttä ottaville käyttäjille on myös hyvä antaa dns-palvelinten tiedot, niin he voivat tarpeen vaatiessa käyttää internetiä ilman, että joutuvat katkaisemaan etäyhteyttä: *async-bootp dns-server 10.9.9.2 8.8.8.8.* Koska etäkäyttäjille ei haluta antaa oikeuksia reitittimen hallintaan käytettiin seuraavaa kahta komentoa, jolla estetään ylimääräiset oikeudet etäkäyttäjiltä: *Line vty 0 4*, *no privilege 15*. Reitittimen etähallintaa varten on ssh-yhteys admin-tunnuksilla.

Tästä jatketaan luomalla yhteydelle pohja, jossa määritellään VPN-yhteyden protokollat, salaus ja IP-osoiteavaruus: *interface Virtual-Template <edellä laitettu nro>, ip unnumbered <lähiverkkoportin osoite tai nimi esimerkiksi vlan12>, peer default ip address pool <edellä oleva ip local poolin nimi>*. Käyttäjien sallittiin myös käyttää tämän verkon internetyhteyttä lisäämällä komento *ip nat enable.* Yhteyksille on hyvä laittaa elinaika, jotta ylimääräisiä sessioita ei jäisi roikkumaan: *keepalive <sekunneissa>.* Tämän jälkeen syötetään tiivistykseen, salaukseen ja yhteydenottoon käytettävät protokollat: *compress mppc, ppp encrypt mppe auto*, *ppp authentication pap chap ms-chap*. Todennukseen on olemassa myös lisää protokollia, mutta nämä olivat tälle yritykselle tarpeelliset.

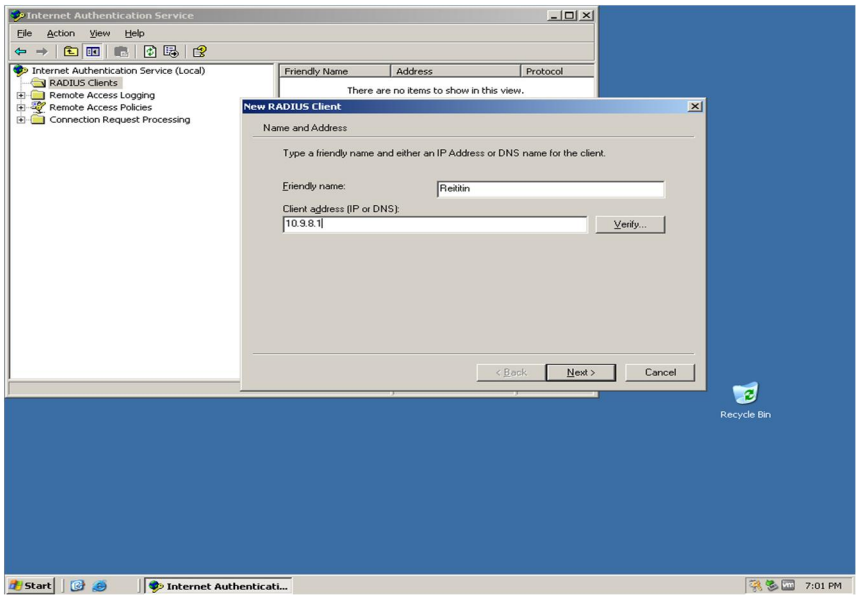
Reitittimen käyttäjien todennus ei kuitenkaan riittänyt, sillä haluttiin hallita käyttäjiä tarkemmin ja helpommin, joten otimme käyttöön IAS-palvelun tiedostopalvelimella. Internet Authentication Service on Microsoftin vastine Radius-palvelulle. Molemmilla pystytään varmentamaan käyttäjiä, IAS otettiin käyttöön tässä tapauksessa sen helpon toteutuksen takia ja koska yrityksellä oli jo käytössä Microsoftin Windows Server 2003. Seuraavat komennot lisättiin, jotta saatiin todennus toimimaan myös kyseisellä palvelimella: *aaa authentication ppp default local group radius, aaa authorization network default local group radius*. Jotta yhteys verkkolaitteen ja palvelimen välillä toimisi, tulee lisätä seuraavat asetukset: *radius-server host <ip> auth-port 1645 acct-port 1646 key 0 <tunnus>.* Oletusportteja käytettäessä porttien määritykset eivät ole tarpeen. Tämän jälkeen asennettiin Windows-palvelimelle IAS-palvelu.

IAS-palvelu on Windows Server 2003 ja aikaisempien ominaisuus, Server 2008:ssa tämän korvaa NPS. Koska yrityksellä oli käytössä Server 2003, toteutettiin Radius-toiminnot IAS:lla. Asennus aloitetaan lisäämällä IAS-protokolla palvelimen Windows-lisäosista. Windows-lisäosia voidaan lisätä "Add or Remove Programs" -valikon "Add/Remove Windows Components" -kohdasta kuvan 8 mukaisesti.



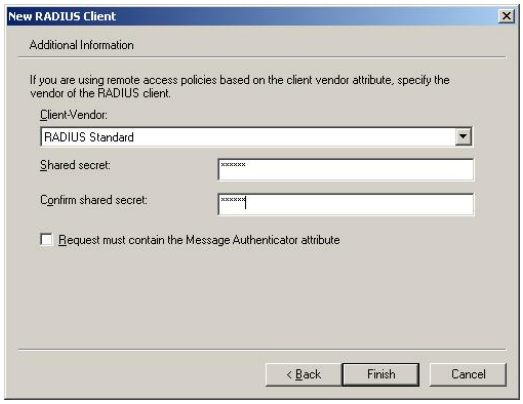

Kuva 8. IAS-protokollan asennus.

Asennuksen jälkeen määritellään asetukset radiuksen asiakaslaitteelle. Sen IP-osoitteeksi tulee verkkolaitteen lähiverkkoportin IP-osoite.

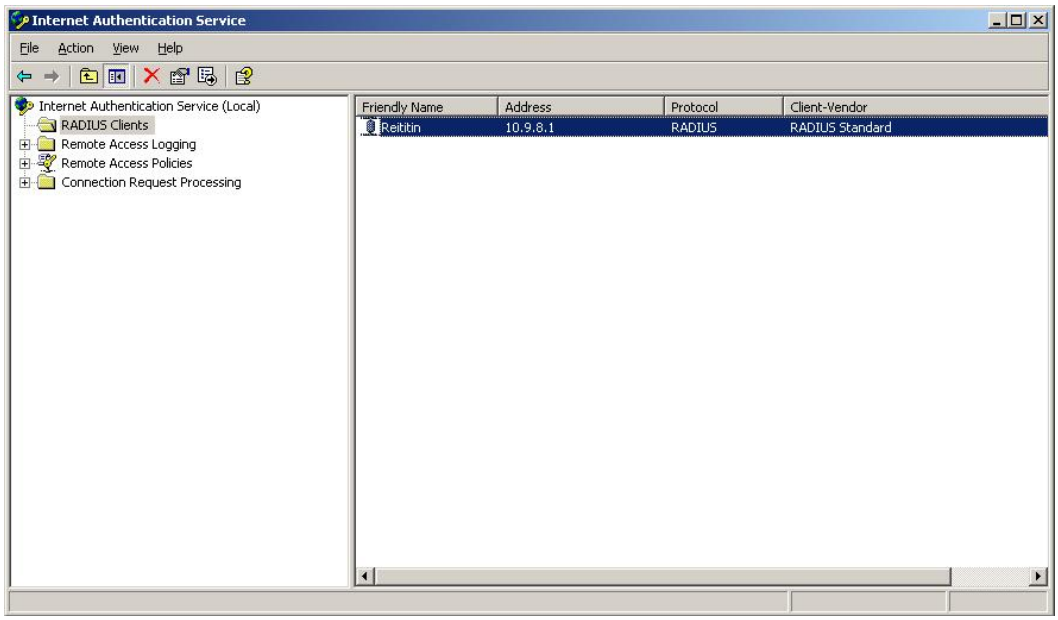



Kuva 9. Radius-palvelun reitittimeen kohdistuvat asetukset.

Tämän jälkeen määritellään palvelimen ja reitittimen välinen jaettu salasana, jolla varmistetaan vain sallittujen laitteiden yhteydenotot.

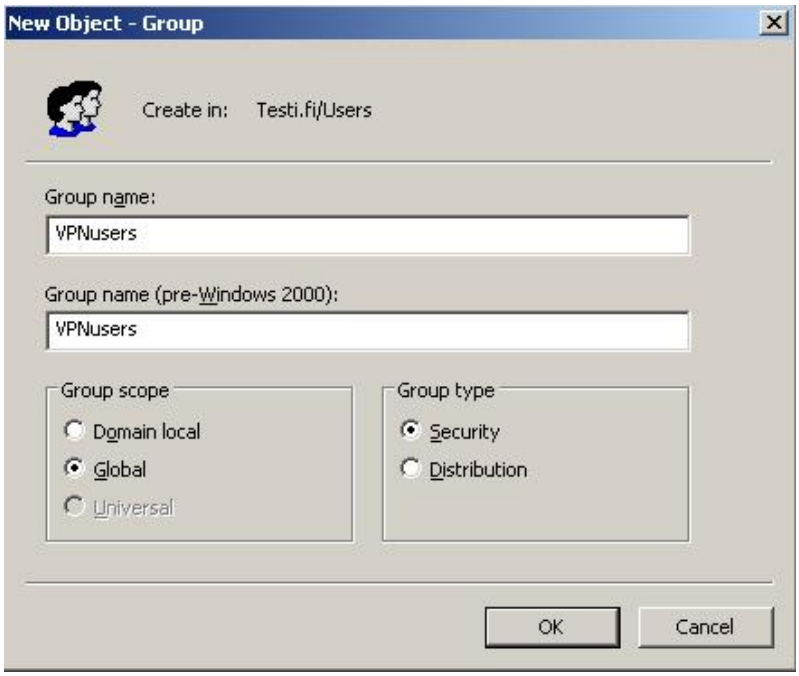



Kuva 10. Radiuksen jaettu salasana.

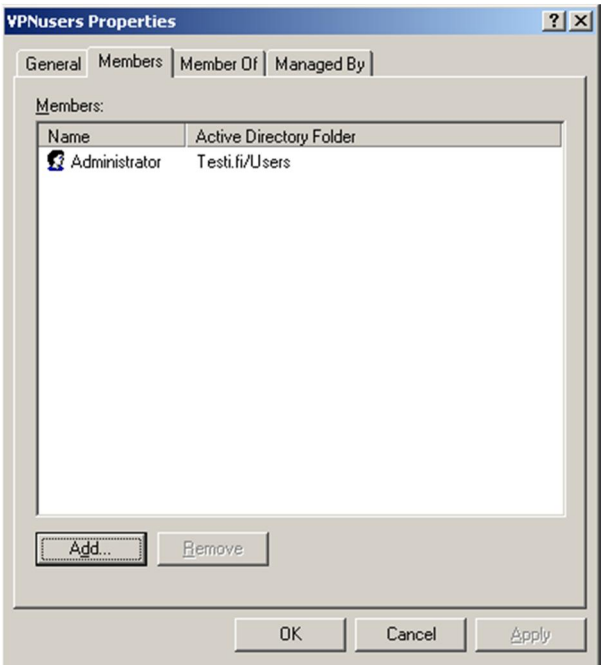



Kuva 11. Asetusten jälkeinen kuvaruutu.

VPN-yhteyksille tulee luoda Active Directory -käyttäjäryhmä ja ryhmälle lisätään jäseniä tarpeen mukaan.

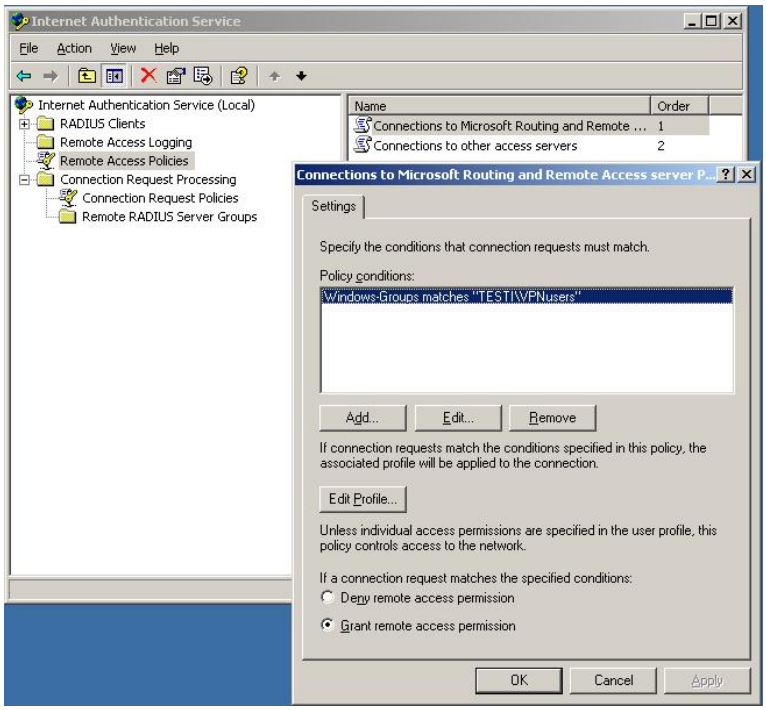



Kuva 12. VPN-käyttäjien Active Directory -käyttäjäryhmä.

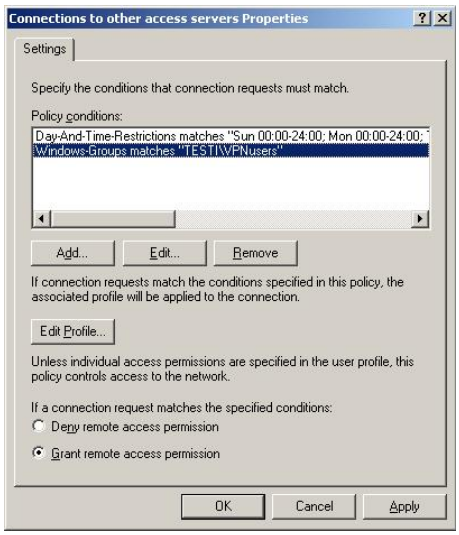



Kuva 13. VPN-käyttäjien Active Directory -ryhmän jäsenet.

Seuraavaksi lisätään tälle palvelulle oikeudet käyttää etäyhteyttä. Helpoiten tämä onnistuu poistamalla vanhat remote access policyt, mikäli siellä ei ole muuta kuin default policyt, ja luomalla seuraavissa kuvissa näkyvät käytännöt:

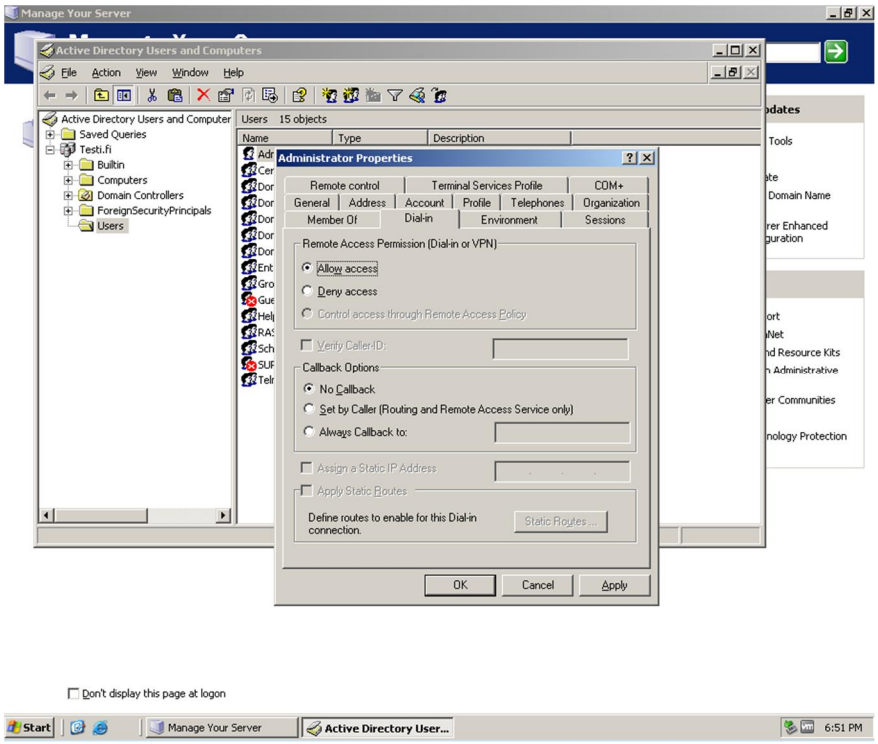



Kuva 14. Remote Access Policy -asetukset.




Kuva 15. Remote Access Policy -asetukset.

Remote Access Policyjen jälkeen sallitaan jokaiselle halutulle käyttäjälle erikseen oikeus ottaa etäyhteys laitteeseen.




Kuva 16. Käyttäjien VPN-yhteydenoton salliminen.

Tämän jälkeen käyttäjät voivat ottaa etäyhteyden reitittimen kautta sisäverkkoon ja saavat näin ollen sisäverkon palvelut käyttöön.

## 3.8 Etähallinta

Koska kaikki PAT:ia käyttävät koneet näkyvät ulospäin yhden IP-osoitteen takana ja ulkoapäin tulevien yhteyksien kanssa yksi TCP tai UDP -portti voidaan ohjata vain yhdelle koneelle tulee tiettyjen porttien kanssa ristiriitoja. Tässä projektissa yrityksellä oli käyttöä portille 22, jota käytetään oletusarvoisesti ssh-yhteyksille, joten reitittimestä otettiin käyttöön uusi portti ssh-yhteyksiä varten. Portti 22 oli tässä kokoonpanossa ohjattu HTTP-palvelimen etähallinnalle. Kyseisen portin olisi voinut ottaa käyttöön reitittimen hallintaa varten, mutta siitä olisi aiheutunut palvelimen ylläpitäjälle lievää vaivaa hallintaohjelman kanssa. Koska reititin oli uusi lisäys verkon infrastruktuuriin, päätettiin siihen vaihtaa portti ja pitää portti 22 ennallaan. Otimme käyttöön helposti muistettavan portin ja kirjasimme sen ylös hallintaan liittyvään ohjelmistoon sekä verkon dokumentaatioon.

Ensimmäiseksi tulee määritellä perusasetukset ssh:n käytettävyyttä varten: *ip ssh time-out 60.* Aikakatkaisu on tärkeä laittaa huolehtimaan siitä, ettei ylimääräisiä sessioita jäisi roikkumaan laitteelle. Niiden purku jälkeenpäin onnistuu vain verkkolaitteen konsoliportista. Seuraavaksi määritellään perusasetukset turvallisuutta varten: *ip http secure-server, ip http timeout-policy idle <sekunteja> life <sekunteja> requests <sallittujen yhteyksien määrä>.* Tämän jälkeen määritellään käyttäjät ssh-yhteydelle: *aaa new-model, aaa authentication login default local, aaa authentication login ssh local, aaa authorization network users local.* Seuraavaksi lisätään ssh:n käyttäjille täydet oikeudet reitittimeen: *line vty 0 4, privilege level 15* ja määritellään käytettävät protokollat *transport input ssh, transport output ssh*. Näiden jälkeen määritettiin rotary-ryhmä, jolla reititin tunnistaa portin ja sallitaan vain ssh sisään: *line vty 0 4, rotary 1, transport input ssh*. Viimeisenä määritellään portti tälle rotary-ryhmälle: *ip ssh port 2022 rotary 1*. Tämän jälkeen pääsee julkisella IP:llä tulemaan portista 2022 ssh-yhteydellä reitittimelle etähallintaa varten.

## 3.9 OSPF

DMZ- sekä toimistoverkon välille tuli saada yhteys toimimaan, joten tätä varten käytettiin OSPF-protokollaa. OSPF on reititysprotokolla, joka mahdollistaa eri IP-verkkojen väliset yhteydet. Toteutus on hyvin yksinkertainen. Ensimmäiseksi määritellään protokollan käyttöönotto, sekä alueelle numero. OSPF otettiin käyttöön komennolla *router OSPF 1*. Seuraavaksi sille määritellään IP-verkot, jotka kuuluvat kyseiselle OSPF-alueelle: *network 10.9.9.0 0.0.0.255 area 0 sekä 10.9.8.0 0.0.0.255 area 0.* Area 0 komennon lopussa tarkoittaa sitä, että kyseessä on runkoverkon reitityksestä, mutta alueita voi luoda tarpeen mukaan lisää vaihtamalla numeroa.

Verkko toimi tämän jälkeen halutulla tavalla ja sen jälkeen pystyimme siirtymään valvonnan toimintakuntoon saattamiseen. Valvontaa varten lisättiin Linux-koneen toisen verkkokortin portti yrityksen valvoman internetyhteyden runkoverkkoon ja sille laitettiin runkoverkkoa vastaava IP-osoite.

# 4 Valvonta

Tietoliikenneprojektit eivät varsinaisesti ole koskaan ohi. Niiden viimeisenä vaiheena on valvonta ja ylläpito. Näihin kuuluvat muun muassa käyttäjien, tietokantojen, jaettujen tiedostojen, ohjelmistojen sekä verkon hallinta. Tässä työssä keskitytään enimmäkseen verkkolaitteiden valvonnan toteutukseen. Muille verkon palveluille oli jo toteutettu hallinta ja tarpeellinen valvonta. Valvontaan otettiin mukaan yrityksen oman Ciscon reitittimen lisäksi heillä hallinnassa olevan yrityskeskuksen lähiverkostoa ylläpitävä laitteisto. Siihen kuului viisi HP Procurve 2524 -kytkintä, Cisco 1841 -reititin, jota käytettiin myös verkon pääpalomuurina sekä HP Procurve 5304xl -pääkytkin palvelinhuoneessa. Näille laitteille oli aikaisemmin valvonta olemassa, mutta vanhan, verkkoa ylläpitävän, yrityksen oston jälkeen se oli lähes käyttökelvottomassa kunnossa. Uudella omistajalla oli käytössä Nagios-valvontaohjelmisto, joten päätettiin hyödyntää sitä ja liittää nämäkin laitteet valvonnan alle. Vanhan valvontajärjestelmän kunnostaminen olisi ollut kustannustehotonta.

Se mitä halutaan tietoliikenneprojekteissa valvoa vaihtelee projektikohtaisesta, mutta yleisimpiä valvonnan kohteita ovat laitteiden verkkoyhteydet, http-, smtp-, ssh-portit, laitteiden virransaanti sekä verkon liikennemäärä sekä -laatu. Näiden lisäksi voidaan valvoa muita verkolle kriittisiä tapahtumia. On myös hyvä luoda valvonnan alaisista tapahtumista pitempiaikaisia logitiedostoja, jotta voidaan tarpeen mukaan parantaa verkon toimivuutta. Toteutus myös vaihtelee projektikohtaisesti. On olemassa monta eri tapaa toteuttaa valvontaa, sille voidaan joko tehdä oma ohjelmisto, hyväksikäyttää valmiita valvontaprotokollia tai ostaa valmis palvelu muualta. Valvonnan toteutus riippuu yleensä yrityksen rahan ja projektiin käytettävän ajan määristä. Tässä työssä säästettiin molemmissa yrityksen valmiin järjestelmän hyväksikäytöllä.

## 4.1 Yleisiä työkaluja

Verkon hallintaan voidaan käyttää useita eri ohjelmistoja, mutta lähes kaikki käyttävät SNMP-protokollaa. SNMP:n (Simple Network Management Protocol) avulla saadaan verkkolaitteille lähetettyä tilakyselyitä sekä määrittää laite lähettämään hälytyksiä valvontaohjelmalle. Se myös mahdollistaa verkon suorituskykyvyn seurannan, ongelmien etsimisen ja ratkaisun sekä verkon kasvun suunnittelun. SNMP:n kolme avainkompo-

nenttia ovat hallintaohjelma, joka vastaanottaa SNMP:ltä tulevat paluuviestit; agentit, joissa käytetään SNMP-protokollaa; MIB (Management Information Base) -taulukko, joka käsittelee agentin SNMP-tietoja. Näiden kolmen komponentin avulla voidaan valvoa verkon laitteita ja asettaa ne tarpeen mukaan hälyttämään tietyistä tapahtumista, kuten esimerkiksi kaistankäytön rajaprosentin ylittymisestä. Muita käytettäviä protokollia ovat muun muassa RMON (Remote Network Monitoring) sekä WBEM (Web-Based Enterprise Management) (Jaakohuhta: 312-322).

Liikenteen tarkasteluun puolestaan voidaan käyttää muun muassa Wireshark-, Ether-Eal- tai TCP Dump -ohjelmistoja. Näillä pystytään valvomaan verkossa liikkuvaa dataa yksityiskohtaisesti. Esimerkiksi Wiresharkilla voidaan seurata dataa protokollan, ajan, lähteen ja kohteen perusteella. Tätä dataa voidaan myös suodattaa tarpeen mukaan sekä tallentaa myöhempää tarkastelua varten. Tämä mahdollistaa dokumentoinnin myös verkkoliikenteestä, jota voidaan käyttää pohjana verkon parantamisessa sekä hallinnassa.

Tämän yrityksen käytössä oli valmiiksi Linux-käyttöjärjestelmälle asennettu Nagios-valvontaohjelmisto sen muita ylläpitoa vaativia laitteita valvomassa, joten päätettiin laajentaa se valvomaan myös aiemmin valvontaa vailla olevia verkkolaitteita. Kyseinen ohjelmisto sijaitsi yrityksen sisäverkkokaavioissa näkyvällä HTTP-palvelimella. Ohjelmiston lisämäärittely vaati parin tunnin harjoittelun, sillä edellinen hallinnoija ei ollut enää yrityksen palveluksessa. Ohjelmisto ei kuitenkaan ole hirveän monimutkainen, ja sen vaatimat muutokset saatiin valmiiksi muutamassa illassa. Asetusten muuttajalta vaadittiin Linux-käyttöjärjestelmän osaamista sekä itse Nagioksen hallintatiedostojen ymmärtämistä.

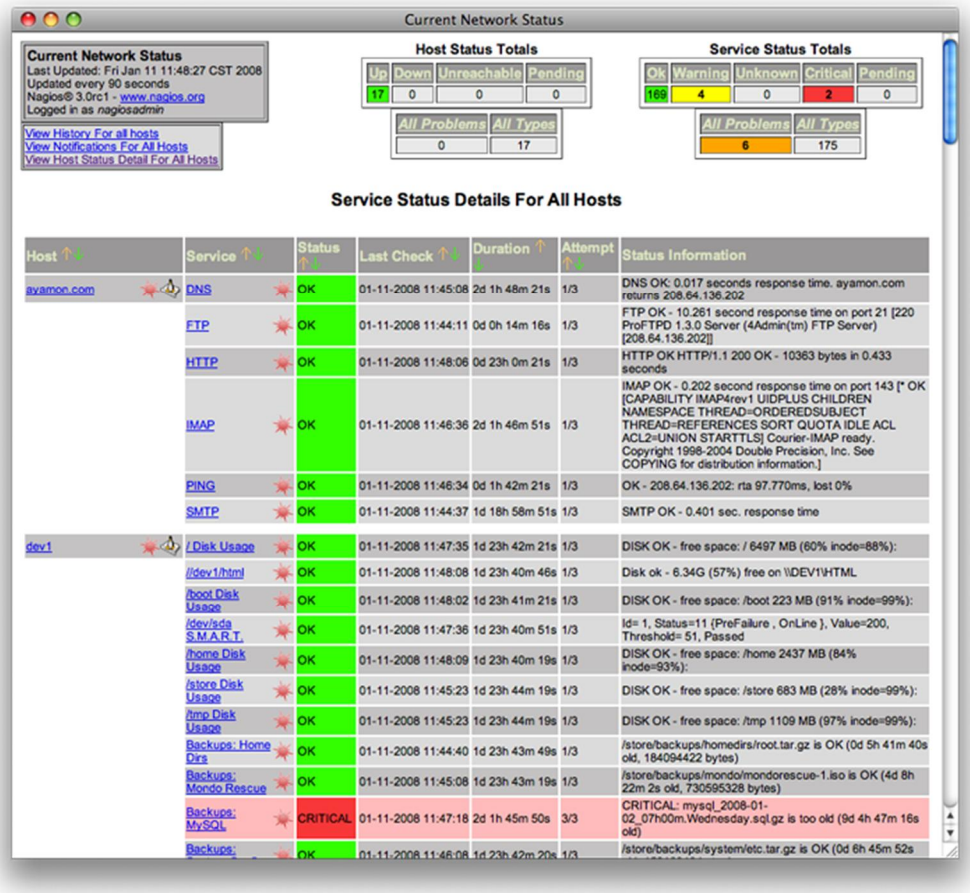
## 4.2 Nagios

Tämän yrityksen valvontaa varten käytettiin Nagios-nimistä ohjelmistoa ja sen Core-versiota. Nagioksen ominaisuuksiin kuuluu muun muassa ohjelmien, palveluiden, käyttöjärjestelmien sekä lähiverkkoprotokollien monitorointi. Tälle on myös mahdollista hommata maksullinen versio nimeltä Nagios XI, joka laajentaa Coren ominaisuuksia. Nagios XI:n maksullisten versioiden hinta alkaen 1 295 Amerikan dollaria (1-50 palvelinta). Tämä kattaa ensimmäiset 12 kuukautta, jonka jälkeen lisenssi voidaan uusia.

Uusiminen puolestaan maksaa 1 196 Amerikan dollaria per vuosi. Hyötyinä maksullisessa versiossa on muun muassa tekninen tuki 3-10 tapaukselle riippuen lisenssistä, ilmaisia lisäosia sekä koulutusresursseja. (Nagios XI Pricing.)

Ohjelmalla on olemassa selkeä graafinen käyttöliittymä, joka näyttää nopeasti, mitkä valvonnan alaiset laitteet ovat kaatuneet tai muuten laukaisseet hälytyksen. Hälytyksistä saadaan tarpeen mukaan ilmoitukset käyttäjille sekä sähköpostilla että tekstiviestillä. Ohjelma perustuu vapaaseen lähdekoodiin, joten se on hyvin pitkälti muokattavissa käyttäjän tarpeiden mukaan. Tämän ansiosta on kehitelty monia liitännäisiä mukaan lukien DNX, jolla voidaan jakaa nagioksen kuormitusta muiden palvelinten kanssa, NSClient++, jolla helpotetaan Windows-laitteiden valvontaa, sekä NagVis, jolla voidaan visualisoida ohjelmiston tietoja kuten esimerkiksi verkon rakennetta.

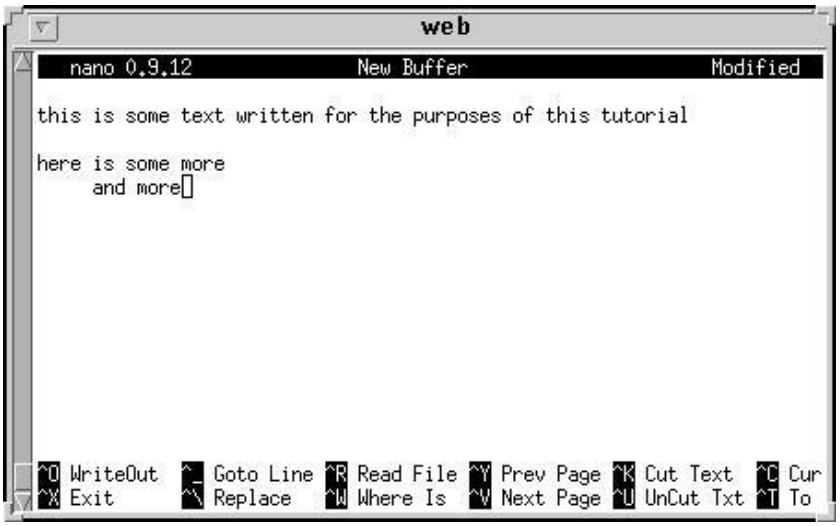
Yrityksellä oli käytössä muun muassa HTTP- sekä ssh-portin valvonnat verkon ulkopuolisille laitteille. Jos valvontaosioon olisi käytetty enemmän aikaa olisi ollut mahdollista ottaa selville nagioksen hallintatiedostoista, kuinka ne toimivat. Näin olisi voitu laajentaa yrityksen sisäistä valvontaa ainakin kyseisten porttien osalta. Myös POP3/IMAP-portit olisi ollut hyvä saada valvontaan HTTP-palvelimella sijaitsevan sähköpostiohjelman toiminnan varmistamista varten. Kuvassa 17 esimerkki valmiiden asennusten jälkeisestä Nagios Coren graafisesta käyttöliittymästä.




Kuva 17. Nagios-valvontaohjelmiston graafinen käyttöliittymä (Nagios Core Screenshots)

### 4.2.1 Käyttäjien määritys

Asetusten määrityksen ensimmäisenä vaiheena on käyttäjien lisäys. Vasta sen jälkeen voidaan siirtyä laitteisiin. Yritys oli määrittänyt jokaiselle valvonnan alla olleelle keskukselle oman määrittelytiedoston ja kaikki löytyvät */etc/nagios/custom*-kansiosta. Heidän omalle verkolle sekä verkolla, joka sijaitsi samassa talossa, lisättiin uusi valvontatiedosto sekaannuksien välttämiseksi. Tiedoston kirjoitus ja muokkaus hoidettiin unix-käyttöjärjestelmistä löytyvällä nano-tekstieditorilla. Komennolla *nano /etc/nagios/custom/yritys.cfg* saatiin tehtyä uusi tiedosto, josta nagios ottaa asetukset laitteiden valvontaan. Tämä tiedosto saatiin tallennettua ctrl+s-näppäinyhdistelmällä.




Kuva 18. Nano-tekstieditori (University of Greenwich's CMS Unix Team: GNU Nano)

Tiedoston kirjoitus aloitetaan määrittelemällä ryhmä, joka huolehtii laitteiden ylläpidosta. Tälle keskukselle luotiin ryhmä, jossa oli kaksi ylläpitäjää, toinen, jolle tulee hälytykset sekä puhelimeen että sähköpostiin vuorokauden ympäri, sekä toinen, jolle tulee vain sähköpostihälytykset. Lauseet, joiden edessä on #-merkki, tarkoittaa sitä, että kyseinen lause on kommentti. Kommenteille helpotetaan tiedoston lukua ja tässä tapauksessa uusien käyttäjien, laitteiden tai ryhmien lisäämistä. Hallintaryhmän teko onnistuu seuraavanlaisesti:

```
# 'Yritys-admins' contact group definition
define contactgroup{
contactgroup_name       yritys-admins
alias yritys            Admins
members                 admin1_sms,admin1_email,admin2_email
}
```

Tässä määriteltiin yrityksen ylläpitäjät ja luotiin yhteys heidän profiileihin. Tätä ryhmää käytetään myöhemmin laitteille määriteltävien hälytysten lähetykseen. Admin-ryhmän lisäksi tehtiin profiili, jonne lähetetään vuorokauden ympäri hälytykset:

```
# 'Yritys-admin1-247' contact group definition
define contactgroup{
contactgroup_name       Yritys-admin1-247
```

```
alias                   yritys admin1 24/7
members                 admin1_sms_247,admin1_email,admin2_email
}
```

Ryhmien luonnin jälkeen tehdään käyttäjien profiilit. Käyttäjien profiileihin tulee luoda palveluiden sekä palvelinten hälytysajankohdat, yhteydenottotapa, yhteystiedot sekä tilat, joista halutaan hälytys. Yhteystietoihin voidaan laittaa joko sähköpostiosoite tai tekstiviestejä vastaanottavan laitteen puhelinnumero. Yhteydenottotavan sekä yhteystietojen tulee olla yhtenevää muotoa, muuten asennustiedosto ei toimi.

```
# 'admin1_sms' contact definition
define contact{
contact_name            admin1_sms
alias                   admin1 SMS

service_notification_period             daytime_r
host_notification_period                daytime_r
service_notification_options            c,r
host_notification_options               d,r
service_notification_commands           notify-by-sms
host_notification_commands              host-notify-by-sms
pager                                   0491234567
}
```

Ensimmäisenä nimen ja aliaksien jälkeen on määritelty viite käytettävään hälytysten lähetysaikatauluun komennoilla *service_notification_period* sekä *host_notification_period*. Aikataulu itsessään määritellään myöhemmin sivulla 32. Seuraavaksi määritetään, mistä lähetetään hälytykset *service_notification_options*, sekä *host_notification_options* -lauseilla. Perässä olevalla kirjaimella määrittelee hälytyksen asteen. Kelpaavia vaihtoehtoja ovat palveluille: w (varoitustila, esimerkiksi hidas toiminta), u (tuntematon tila), c (kriittiset tilat, esimerkiksi palvelu ei toimi ollenkaan) sekä n (ei hälytyksiä). Palvelimille kelpaavat vaihtoehdot ovat puolestaan: d (palvelin alhaalla), u (palvelimeen ei saada yhteyttä, unreachable), r (palvelin palautuu normaalitilaan) sekä n (ei hälytyksiä). Tämän jälkeen määritellään hälytysten lähetystapa sekä

yhteystiedot. Tässä profiilissa määriteltiin tekstiviestit hälytystavaksi sekä määriteltiin sille puhelinnumero. Seuraavana malli profiilista, johon lähetetään 24 tuntina vuorokaudessa seitsemänä päivänä viikossa hälytyksiä niiden tapahtuessa.

```
# 'admin1_sms_247' contact definition
define contact{
contact_name                    admin1_sms_247
alias admin1                    SMS 24/7
service_notification_period     24x7
host_notification_period        24x7
service_notification_options    c,r
host_notification_options       d,r
service_notification_commands   notify-by-sms
host_notification_commands      host-notify-by-sms
pager                           0491234567
}
```

Seuraavaksi luotiin sähköpostihälytyksiä varten profiilit molemmille käyttäjille alla olevan esimerkin mukaisesti. Suurimpana erona tekstiviesti- ja sähköpostiprofiilien välillä oli hälytysten määrä. Sähköpostihälytyksiin otettiin mukaan myös warning-viestit, jotta tarpeen mukaan saatiin tietoon, jos jokin palvelu tai palvelin on jatkuvasti suurella kuormituksella tai lähellä käyttökelvottomuutta.

```
# 'admin1_email' contact definition
define contact{
contact_name                    admin1_email
alias                           admin1 email alarms
service_notification_period     24x7
host_notification_period        24x7
service_notification_options    c,w,u,r
host_notification_options       d,u,r
service_notification_commands   notify-by-email
host_notification_commands      host-notify-by-email
email                           admin1@yritys.fi
```

```
}
```

Ennen kuin siirryttään itse laitteiden profiileiden luontiin, määriteltiin viimeisenä vielä kellonaika profiileille, joille ei haluta hälytyksiä muuta kuin viikolla säädyllisiin kellonaikoihin. Tätä profiilia voidaan käyttää, jos halutaan lisätä valvojia, jotka ovat tarpeen mukaan saatavilla työviikon aikana.

```
# 'daytime_r' timeperiod definition
define timeperiod{
timeperiod_name         daytime_r
alias                   daytime_r 7-22
monday                  07:00-22:00
tuesday                 07:00-22:00
wednesday               07:00-22:00
thursday                07:00-22:00
friday                  07:00-22:00
}
```

Kyseisessä taulussa määritellään sille nimi sekä jokaiselle halutulle viikonpäivälle tunnit, joina halutaan hälytysten tulevan. Tässä työssä ei käytetty kyseistä aikataulua, koska valvojia oli vain muutama ja heille haluttiin hälytykset aina. Näillä asetuksilla saatiin hälytysten vaatimat asetukset toimimaan ja pystyttiin siirtymään valvottavien laitteiden vaatimien asetusten säätämiseen.

### 4.2.2 Laitteiden konfigurointi

Verkkolaitteiden osalta tiedoston muokkaaminen on hyvin pitkälti yhtenevää käyttäjien tietojen lisäämisen kanssa. Suurimpana erona on se, että joudutaan luomaan laitteiden tietojen lisäksi niille ryhmä sekä määritellä palvelu, millä tarkistetaan laitteen toimivuus. Ensimmäisenä luodaan ryhmä, johon lisätään valvottavat laitteet:

```
# 'Yritys' host group definition
define hostgroup{
hostgroup_name          Yritys
```

```
alias                   Yritys
contact_groups          admins,yritys-admins
members                 Yritys-FW,yritys-router,yritys-MDF,yritys-IDF1,yritys-IDF1B,
yritys-IDF2, yritys-IDF3
}
```

Ryhmälle luodaan nimi ja aliaksia tarpeen mukaan. Sen jälkeen lisätään sille käyttäjät tai käyttäjäryhmät, joille hälytykset lähtevät ja viimeisenä määritellään verkkolaitteiden profiilien nimet. Laiteryhmän luonnin jälkeen tehdään laitteille valvontakomento. Tässä tapauksessa otettiin käyttöön kymmenen ping-komennon sarja. Ohjelmisto lähettää siis laitteelle kymmenen ping-komentoa ja saatujen vastausten perusteella päättelee, onko tarvetta lähettää hälytyksiä laitteesta. Kokeneen linux-käyttäjän, sekä lisäajan kanssa valvontaa olisi voinut laajentaa esimerkiksi HTTP-palvelimen HTTP- ja SSH-portin valvomiseen. Komennon luonti on seuraavanlainen:

```
# Yritys service definition template
define service{
; The 'name' of this service template, referenced in other service definitions
name                    yritys-ping-service
use                     server-service ; Name of service template to use
max_check_attempts      10
contact_groups          admins,yritys-admins,yritys-admin1-247
service_description     PING
check_command           check_ping!500.0,50%!1000.0,80%

register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A
TEMPLATE!
}
```

Ensimmäisenä luodaan komennolle nimi. Seuraavaksi otetaan käyttöön Nagioksessa oleva "server-service" käyttöön, joka määrittelee valvottavan kohteen olevan verkkolaite eikä esimerkiksi palvelu. Tämän jälkeen tulee enimmäisyritysten määrä, hälytysten kontaktihenkilöt tai ryhmät, käytettävä komento sekä komennolle hälytysrajat. Hälytysrajoista ensimmäinen on "warning"-rajapinta ja toinen "critical". Ensimmäinen numero

määrittää vastinajan ja toinen epäonnistuneiden komentojen prosentin. Eli jos pingien vastauksissa kestää yli 500 ms tai niistä 50 % epäonnistuu, lähetetään siitä vastaanottajille warning-tason hälytys, vastaavat arvot critical-hälytykselle ovat 1000 ms tai 80 %. Viimeisenä määritellään laitteiden tiedot valvontaohjelmistolle:

```
# Yritys MDF
define host{
use                     server-host ; Name of host template to use
host_name               yritys-MDF
alias                   Yritys MDF HP-runkokytkin
address                 15.9.1.1
parents                 yritys-router
}
```

Ensimmäisenä kerrotaan kyseessä olevan yksittäinen verkkolaite "server-host"-komennolla. Sen jälkeen määritellään nimi, joka näkyy visuaalisessa käyttöliittymässä. Nimen ei tarvitse olla sama kuin mikä sen verkkonimeksi on laitettu. Laitteesta voidaan antaa aliaksena enemmän tietoja, jotka näkyvät käyttöliittymän lisätiedoissa. Tämän jälkeen tulee laitteen IP-osoite sekä mahdolliset isäntälaitteet, joiden läpi tämä laite kuljettaa verkkoliikennettä. Isäntälaitteiden lisääminen helpottaa muun muassa vianetsintää yhteysongelmia ratkottaessa. Seuraavaksi määritellään laite käyttämään edellä määriteltyä ping-valvontaa:

```
# service definition
define service{
use                     yritys-ping-service ; Name of service template to use
host_name               yritys-MDF
}
```

Tämä toteutettiin jokaiselle verkon kytkimelle, verkon pääpalomuurille, sekä yrityksen omalle palomuurille. Määrittelytiedoston koko sisältö löytyy liitteestä 1. Toisena esimerkkinä yrityksen IDF1-nimellä kulkeva kytkin:

```
# yritys IDF1
define host{
use                     server-host ; Name of host template to use
host_name               yritys-IDF1
alias                   yritys IDF1 HP-kuitukytkin
address                 15.9.1.3
parents                 yritys-MDF
}

#Service definition
define service{
use                     yritys-ping-service ; Name of service template to use
host_name               yritys-IDF1
}
```

Palomuurien valvonta toteutettiin julkisen IP-osoitteen kautta. Näin varmistettiin myös niiden alla olevien verkkojen internetyhteys. Sekaannuksien välttämiseksi nimettiin toinen routeriksi ja toinen firewalliksi. Tässä tapauksessa router-nimellä oli pääverkon palomuuri-reititin ja firewall yrityksen Cisco 871. Alla molempien laitteiden asetukset:

```
# yritys router
define host{
use                     server-host ; Name of host template to use
host_name               Yritys-router
alias main              Firewall
address                 80.1.1.2
}


# Service definition
define service{
use                     yritys-ping-service ; Name of service template to use
host_name               yritys-router
}
```

```
# yritys fw
define host{
use                     server-host ; Name of host template to use
host_name               Yritys-FW
alias                   Yritys871 (Cisco 871)
address                 80.1.1.1
parents                 yritys-MDF
}
# Service definition
define service{
use                     yritys-ping-service ; Name of service template to use
host_name               yritys-FW
}
```

Kun kaikki oli saatu syötettyä, tallennettiin tiedosto */etc/nagios/custom/* -kansioon nimellä *yritys.cfg*. Tämän jälkeen tuli koko ohjelmisto käynnistää uudestaan komennolla: */etc/init.d/nagios restart*. Mikäli tiedosto on kunnossa, käynnistyy Nagios uudelleen ilman virheilmoituksia. Jos siinä on jotain vikaa, kertoo se käynnistyessään viallisen tiedoston rivinumeron ja jättää tiedoston lataamatta. Kaikki toimivat tiedostot ladataan, ja ohjelma käynnistyy muuten normaalisti. Jos tiedostoa tarvitsee muuttaa, voi viallisen rivin saada helposti esiin komennolla *head -<rivinumero> /etc/nagios/custom/yritys.cfg*. Sen jälkeen aukaisee tiedoston tekstieditorilla, hakee kyseisen rivin esille ja tekee tarpeelliset muutokset. Nagioksen pluginit löytyvät puolestaan kansiosta */etc/nagios-plugins/config* tai */usr/lib/nagios/plugins*, mikäli niitä on tarve muutella.

Ohjelman ladattua toimivan konfiguraatiotiedoston tarvitsee käydä graafisen käyttöliittymän puolella laittamaan jokaisen laitteen kohdalla "Enable all service notifications from this host". Tämän jälkeen Nagios valvoo laitteita ja lähettää tarpeen mukaan hälytyksiä. Toimintaa testattiin ottamalla yksi kytkin hetkeksi irti, hälytyksien tulon jälkeen kytkin laitettiin takaisin. Ping-komennon toimintaa testattiin myös, ja se antaa seuraavanlaisen vastauksen:

```
palvelin:/usr/lib/nagios/plugins# ./check_ping -H 10.9.9.2 -w 500,50% -c 1000,80%
PING OK - Packet loss = 0%, RTA = 0.08 ms|rta=0.083000ms;500.000000;1000.000000;0.000000 pl=0%;50;80;0
```

Näiden testausten jälkeen todettiin verkon täyttävän sen hetkiset vaatimukset, ja projekti oli saatu valmiiksi. Dokumentaatio päivitettiin valmiin konfiguraation mukaiseksi, ja se talletettiin tiedostopalvelimelle. Palvelimesta otettiin automaattinen varmuuskopio yrityksen verkkolevylle.

# 5 Kehitysideoita

Yrityksen valvonta oli toteutettu hieman vaatimattomattomasti. Pyrittiin vain valvomaan laitteiden pystyssäolemista. Sitä voitaisiin laajentaa valvomaan muun muassa kaistan käyttöä ja käyttäjiä sekä tiettyjä protokollia, esimerkiksi SMTP-yhteyksiä. Esimerkiksi Nagiokseen voisi lisätä Check SNMP Cisco Traffic tai Check Interfaces Operational Status -liitännäiset. Molemmat ovat lisäosia, joilla voidaan valvoa IF-MIB yhteensopivien laitteiden porttien pystyssä olemista. Osaava Nagios sekä Linux-käyttäjä voisi puolestaan kirjoittaa ohjelman, jolla pystyy valvomaan verkon laitteita MIB-taulun rajoituksien mukaisesti. Myös lisäohjelmistojen hankkiminen voisi olla hyvä ottaa huomioon. Esimerkiksi Paessler on kehittänyt oman työkalun verkkoliikenteen valvomiseen. Se kulkee nimellä PRTG Network Monitor (Paessler: PRTG Network Monitor), ja sillä pystytään valvomaan verkon liikennettä useammalla eri tavalla.

Verkon kaistan käytön valvomiseen olisi hyvä olla myös oma ohjelma, jota voitaisiin käyttää verkon analysointiin sekä käytön dokumentaatioon. Analyysin jälkeen olisi hyvä miettiä, tarvitaanko QoS-palvelua tai onko tarpeellista kahdentaa verkon laitteistoa. Analyysin perusteella voisi myös määritellä, onko tarpeellista rajoittaa tiettyjen verkon käyttäjien yhteyttä, esimerkiksi virusten tai liiallisen verkon kuormituksen takia.

## 5.1 NetFlow ja NBAR

Ciscolta löytyy oma protokolla verkon liikenteen valvomiseen nimeltään NetFlow. Sillä pystytään keräämään verkon liikenteestä tilastotietoja. Niihin kuuluvat muun muassa

verkon käyttäjät, liikenteen tyyppi, sen ajankohta sekä lähtö- että päätepiste. Näitä tietoja voidaan hyväksikäyttää verkon vikatilanteissa, liikenteen analysoinnissa, suorituskyvyn seurannassa sekä turvallisuusuhkien ehkäisemisessä. NetFlow:lla valvotussa ympäristössä esimerkiksi DDOS-yritykset, virukset sekä mahdolliset madot on helpommin havaittavissa ja niitä pystytään seuraamaan reaaliajassa. Tämän ansiosta verkon ylläpitäjät pystyvät tekemään tarpeellisia muutoksia verkon turvallisuuteen sekä luomaan verkosta luotettavamman. Se mahdollistaa myös tilastotietojen viemisen ulkoiseen tiedostoon, mikä puolestaan helpottaa toistuvien uhkien, sekä riskikäyttäjien tunnistamista. Tilastotietojen ansiosta saadaan aikaan pitkäaikainen verkon käytön dokumentaatio, jolla pystytään parantamaan verkon toimivuutta, sekä estämään sille haitallista liikennettä. Lisäetuina on mahdollista hyödyntää käytön määrään perustuvaa laskutusta siihen soveltuvissa verkkoratkaisuissa. NetFlow on integroitu Ciscon laitteisiin ja sen käytöstä on saatavilla ohjeen Ciscon virallisilta internetsivuilta. (Cisco: Netflow & Network-based application recognition.)

Netflowin kanssa käytetään yleensä NBAR-protokollaa, joka tunnistaa suuren osan yleisimmistä verkkoa käyttävistä ohjelmistoista. Tämän lisäksi on ohjelmaan luotu mahdollisuus lisätä tunnistettavia protokollia Packet Description Language Modulen kautta. Sen ansiosta voidaan luokitella liikennettä ja käyttää muiden ohjelmien kanssa verkon liikenteen hallitsemiseen. NBAR mahdollistaa myös tilastojen luonnin ohjelmien kaistankäytöstä, sekä IP-arvojärjestyksen merkitsemisen paketeille. Protokollalla ei kuitenkaan voida yksin valvoa verkon tapahtumia, se tarvitsee sitä hyväksikäyttävän protokollan tai ohjelman seuraamaan siltä tulevaa dataa. NBAR vaatii kuitenkin verkkolaitteelta normaalia enemmän käyttömuistia, joka on otettava huomioon protokollan käyttöönottoa harkittaessa. Se saattaa vaatia myös ensimmäistä kertaa käynnistäessä huomattavan pitkän ajan kalibrointia varten. Tämän aikana verkkolaitteen käyttö on todennäköisesti hidasta tai mahdotonta. Ensimmäisen käynnistyksen jälkeen laite toimii normaalisti mikäli muisti ei ole täynnä. (Cisco: Quality of Service Networking.)

## 5.2 QoS

QoS (Quality of Service) on termi, jota käytetään kuvaamaan tekniikoita, joilla hallitaan verkon liikennettä. QoS:llä hallitaan neljää verkon osa-aluetta: Kaistankulutus, latenssi eli viive datan lähetys- ja vastaanottopisteen välillä, latenssiajan vaihtelu sekä luotetta-

vuus. Luotettavuudella tarkoitetaan pudotettavien pakettien lukumäärää, jota QoS-tekniikat vähentävät. QoS:n käyttöönotto voisi olla tarpeellista myös tässä järjestelmässä, mikäli verkon käyttö on korkealla tiettynä aikana päivästä tai viikosta. Sen ansiosta pystyttäisiin välttymään verkon tukoksilta sekä varmistamaan tärkeän liikenteen kulkeminen myös kiireaikoina. (Cisco: CCNP: Optimizing Converged Networks.)

Normaalitilanteissa, joissa ei ole määritelty QoS-protokollaa, käytetään FIFO-jonotusjärjestelmää. Tämä tarkoittaa sitä, että laitteelle tulevat paketit reititetään eteenpäin siinä järjestyksessä, missä ne saapuivat sinne. Tällä periaatteella ei pystytä varmistamaan kaikkien tai edes tärkeiden pakettien perille saapumista ruuhkatilanteissa. Kun käyttöön otetaan parempi jonotusjärjestelmä voidaan korottaa tiettyjen pakettien tärkeyttä ja laskea toisten. Hyvänä esimerkkinä liikenteen pakettien kulun tärkeydestä on ero tiedostonsiirron ja ääniliikenteen välillä. Tiedonsiirrossa on hyväksyttävää siirtonopeuden vaihtelut ja vaikka hidastuminen saattaa haitata käyttäjää ei se ole kuitenkaan kriittistä. Ääniliikenteessä, esimerkiksi VOIP (Voice Over IP), puolestaan jo yksi kymmenesosasekunnin viive aiheuttaa käyttäjälle huomattavan eron puhelun laadussa. (Cisco: CCNP: Optimizing Converged Networks.)

Ciscon IOS:ssa on neljä erityylistä tapaa hoitaa QoS-palvelua: ruuhkanhallinta, jononhallinta, linkin tehokkuuden hallinta sekä liikenteen muokkaus ja hallinta. Ruuhkanhallintatyökaluilla voidaan määritellä liikenteelle tärkeysjärjestykset. Niiden avulla varmistetaan tärkeän liikenteen kulku myös verkon ruuhkautuessa sekä järjestetään muuta liikennettä paremmin. Näihin työkaluihin kuuluu muun muassa Priority Queuing, Class-based Queuing, Weighted Fair Queuing sekä Class-based Weighted Fair Queuing. Jononhallintatyökalut puolestaan yrittävät varmistaa, että laitteen jono ei täyty, jotta siellä on tilaa tärkeille paketeille. Ne käyttävät ennalta määrättyä kriteeriä alhaisen tärkeystason pakettien pudottamiselle ennen korkean tärkeystason pakettien pudotusta. Tästä esimerkkinä toimii Weighted Random Early Detection, joka pudottaa paketteja sattumanvaraisesti ruuhkan lisääntyessä. Se käyttää kuitenkin ennalta määriteltyjä sääntöjä välttääkseen tärkeämpien tiedostojen pudottamista. Linkin tehokkuutta voidaan puolestaan hallita pienentämällä aikaa, joka reitittimellä menee paketin laittamiseen laitteesta linkille. Tehokkuuteen voidaan myös vaikuttaa RTP- ja CRTP-protokollilla. Nämä vähentävät paketeissa olevaa otsikkoa, joka pahimmillaan voivat olla kaksi kertaa niin suuri kuin paketissa oleva hyötytieto. Liikenteen muokkailulla

(shaping) ja hallinnalla (policing) saadaan aikaan rajat halutuille liikennetyypeille. Esimerkiksi frame-relay-liikenne voidaan rajoittaa vaikka 512 kilobittiin sekunnissa, jos vastapää ei pysty sitä nopeammin ottamaan vastaan tai halutaan muuten rajoittaa sen kaistankäyttöä. Paketit menee puskuroituna muistiin odottamaan vuoroaan tarpeen mukaan. Hallinta toimii muuten samaan tapaan, mutta sen sijaan että paketit olisi puskuroituna muistissa, ne yleensä pudotetaan kokonaan. (Cisco: CCNP: Optimizing Converged Networks.)

QoS-tekniikat ei kuitenkaan ratkaise ongelmia, mikäli ne riippuvat laitteista tai kaistan riittämättömyydestä. Näissä tilanteissa se auttaa kuitenkin tilapäisenä korjauksena. Tästä syystä olisi hyvä saada verkolle asianmukainen kaistankäytön valvonta. Se nopeuttaisi verkon kuormitusongelmien huomaamista, sekä niiden ratkaisemista.

## 5.3 Vikasietoisuuden lisääminen

Yrityksen hallinnassa olevan runkoverkon kahdennusta olisi hyvä harkita, jotta saataisiin nostettua verkon toimintavarmuutta ja käytettävyysaikaa. Kahdennus mahdollistaisi myös kuormituksen jaon ja varayhteyksien teon. Varayhteyksillä ja kahdennuksella pystyttäisiin myös välttymään yksittäisten laitteiden tilapäisistä häiriöistä. Näitä tilanteita tuli yksi kuuden kuukauden aikana ja silloin vian selvittämisessä meni noin tunti. Tänä aikana koko verkko oli alhaalla, josta harmia syntyi sekä yritykselle itselleen että yrityksen asiakkaille.

Nykyajan hallittavissa laitteissa on laitteiden tekijöillä menetelmiä, joilla saadaan aikaan vikasietoisuutta, jos verkkoa on kahdennettu. Näihin kuuluu muun muassa Ciscon HSRP, Extremen ESRP sekä RFC sertifioitu VRRP (RFC 2338). Näillä protokollilla pystytään varmistamaan laitteiden välisten yhteyksien toimivuus, mutta ei kuitenkaan yksittäisten linkkien toimivuutta. Protokollien toiminta perustuu yhteyspulssien seurantaan, mikäli pulssia ei ensisijaisen linkin läpi havaita siirrytään käyttämään varayhteyttä. Varayhteydestä voidaan palata pääyhteyden käyttöön joko manuaalisesti tai automaattisesti linkin noustessa pystyyn. (Jaakohuhta: 175-177.)

Yhteyksien toimivuutta on myös mahdollista parantaa niputtamalla kaksi tai useampi samalle laitteelle menevä linkkiä. Tämä mahdollistaa kaistan lisäämisen sekä toimivuu-

den parantamisen. Yhteysnopeus putoaa kuitenkin, mikäli linkkinipusta yksi tai useampi linkki katkeaa, mutta yhteys pysyy päällä niin kauan, kuin edes yksi nipun linkeistä on pystyssä. Linkkien niputukseen on olemassa monta erinimistä tekniikkaa, mutta ne ovat käytännössä kuitenkin hyvin samanlaisia, vain toteutukset eroavat toisistaan. Näitä tekniikoita ovat muun muassa Ciscon FastEther Channel tai GigaEther Channel, Nortelin MultiLink Trunking sekä IEE 802.3ad Link Aggregation. (Jaakohuhta: 177-178.)

Tätä projektia harkittiin yhdessä vaiheessa ja käytiin läpi sen toteutusmahdollisuuksia, -tapoja sekä kuluja. Kävimme myös läpi yrityksen omistamia käyttämättömiä laitteita siinä toivossa, että niillä olisi saanut kuluja ajettua alas. Toteutukseen sopivia laitteita ei kuitenkaan löytynyt, joten kulut verkon kahdennuksesta eivät helpottuneet. Toteutukseen käytettävän ajan ja rahan määrästä ei kuitenkaan päästy yhteisymmärrykseen, joten projekti hyllytettiin.

# Lähteet

CCNP: Optimizing Converged Networks. Cisco. Verkkodokumentti. 2009. <http://cisco.netacad.net>. Luettu 16.1.2011.

Check Interfaces Operational Status. Nagios Exchange. 2009. <http://exchange.nagios.org/directory/Plugins/Network-Connections%2C-Stats-and-Bandwidth/Check-Interfaces-Operational-Status/details>. Luettu 27.11.2011.

Check SNMP Cisco Traffic. Nagios Exchange. 2009. Verkkodokumentti. <http://exchange.nagios.org/directory/Plugins/Hardware/Network-Gear/Cisco/Check-SNMP-Cisco-Traffic/details>. Luettu 27.11.2011.

Details of Technician Commands. Cisco DocWiki. 2008. Verkkodokumentti. <http://docwiki.cisco.com/wiki/Cisco_Unified_MeetingPlace,_Release_6.x_--_Details_of_Technician_Commands>. Luettu 26.11.2011.

DHCP. R. Droms. 1997. Verkkodokumentti. <http://www.ietf.org/rfc/rfc2131.txt>. Luettu 26.11.2011.

Domain Names - Implementation and Specification. Mockapetris. 1987. Verkkodokumentti. <http://www.ietf.org/rfc/rfc1035.txt> Luettu 22.2.2012.

GNU Nano. University of Greenwich's CMS Unix Team. Verkkodokumentti. <http://unix.cms.gre.ac.uk/pdf/software/editors/nano.pdf>. Luettu 7.11.2011.

Hakala, Mika ja Vainio, Mika. 2005. Tietoverkon rakentaminen. Jyväskylä: Docendo Finland Oy.

Jaakohuhta Hannu. 2005. Lähiverkot - Ethernet. Helsinki: Edita publishing Oy.

Nagios Core. Verkkodokumentti. 2009. <http://nagios.com/products/nagioscore>. Luettu 7.11.2011.

Nagios Core Screenshots. Verkkodokumentti. 2009.
<http://www.nagios.com/products/nagioscore/screenshots>. Luettu 27.11.2011.

Nagios Core 3.x documentation. Verkkodokumentti.
<http://nagios.sourceforge.net/docs/3_0/toc.html>. Luettu 7.11.2011.

Nagios XI Pricing. Verkkodokumentti. 2009.
<http://www.nagios.com/products/nagiosxi/pricing>. Luettu 21.2.2012.

Netflow & Network-based application recognition. Cisco. 2003. Verkkodokumentti.<http://www.cisco.com/application/vnd.ms-powerpoint/en/us/guest/tech/tk362/c1482/ccmigration_09186a00801da7de.ppt>. Luettu 27.11.2011.

NTP Pool Project. A. Hansen. 2004. Verkkodokumentti.
<http://www.pool.ntp.org/en/>. Luettu 2.2.2012.

OSPF. J. Moy. 1998. Verkkodokumentti. <http://www.ietf.org/rfc/rfc2328.txt>. Luettu 26.11.2011.

PRTG Network Monitor. Paessler. 1998. Verkkodokumentti.
<http://www.paessler.com/prtg>. Luettu 27.11.2011.

Quality of Service Networking. Cisco. 2009. Verkkodokumentti.
<http://docwiki.cisco.com/wiki/Quality_of_Service_Networking>. Luettu 2.2.2012

Security Commands: reverse-route through show crypto isakmp sa. Ciscon verkkotietokanta. 1987. Verkkodokumentti.
<http://www.cisco.com/en/US/docs/ios/12_3/security/command/reference/sec_r1g.html#wp1074075>. Luettu 7.11.2011.

SSH. T. Ylonen. 2006. Verkkodokumentti. <http://www.ietf.org/rfc/rfc4251.txt>. Luettu 26.11.2011.

VLAN. T. Chown. 2006. Verkkodokumentti. <http://www.ietf.org/rfc/rfc4554.txt>. Luettu 26.11.2011.

VPN. L. Andersson & T. Madsen. 2005. Verkkodokumentti. <http://www.ietf.org/rfc/rfc4026.txt >. Luettu 26.11.2011.

WAN. Cisco DocWiki. 2009. Verkkodokumentti. <http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies>. Luettu 26.11.2011.

Nagios-ohjelman konfiguraatiotiedoston sisältö

```
# 'yritys-admins' contact group definition
define contactgroup{
  contactgroup_name       yritys-admins
  alias                   yritys Admins
  members                 admin1_sms, admin1_email, admin2_email
}


# 'yritys- admin1-247' contact group definition
define contactgroup{
  contactgroup_name       yritys- admin1-247
  alias                   yritys admin1 24/7
  members                 admin1_sms_247, admin1_email, admin2_email
}


# 'Admin2_sms' contact definition
define contact{
  contact_name                    admin1_sms
  alias                           admin1 SMS

  service_notification_period     daytime_r
  host_notification_period        daytime_r
  service_notification_options    c,r
  host_notification_options       d,r
  service_notification_commands   notify-by-sms
  host_notification_commands      host-notify-by-sms
  pager                           0491234567
}
```

```
# ' admin1_sms_247' contact definition
define contact{
  contact_name                    admin1_sms_247
  alias                           admin1 SMS 24/7

  service_notification_period     24x7
  host_notification_period        24x7
  service_notification_options    c,r
  host_notification_options       d,r
  service_notification_commands   notify-by-sms
  host_notification_commands      host-notify-by-sms
  pager                           0491234567
}


# ' admin2_email' contact definition
define contact{
  contact_name                    admin2_email
  alias                           admin2 Maili halytykset

  service_notification_period     24x7
  host_notification_period        24x7
  service_notification_options    c,w,u,r
  host_notification_options       d,u,r
  service_notification_commands   notify-by-email
  host_notification_commands      host-notify-by-email
  email                           admin2@yritys.fi
}


# ' admin1_email' contact definition
define contact{
  contact_name                    admin1_email
```

```
  alias                           admin1 email alarms

  service_notification_period     24x7
  host_notification_period        24x7
  service_notification_options    c,w,u,r
  host_notification_options       d,u,r
  service_notification_commands   notify-by-email
  host_notification_commands      host-notify-by-email
  email                           admin1@yritys.fi
}


# 'daytime_r' timeperiod definition
define timeperiod{
  timeperiod_name       daytime_r
  alias                 daytime_r 7-22
  monday                07:00-22:00
  tuesday               07:00-22:00
  wednesday             07:00-22:00
  thursday              07:00-22:00
  friday                07:00-22:00
}


# 'yritys' host group definition
define hostgroup{
  hostgroup_name        yritys
  alias                 yritys
  contact_groups        admins, yritys -admins
  members               yritys-FW, yritys-router, yritys-MDF, yritys-IDF1, yritys-
IDF1B, yritys-IDF2, yritys-IDF3
}
```

```
# yritys service definition template
define service{
  ; The 'name' of this service template, referenced in other service definitions
  name        yritys-ping-service
  use         server-service  ; Name of service template to use

  max_check_attempts     10
  contact_groups         admins, yritys-admins, yritys-admin2-247

  service_description    PING
  check_command          check_ping!500.0,50%!1000.0,80%

  register  0  ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A
TEMPLATE!
}


# yritys 1. vlan valvonta definition
#define service{
#               name                    yritys-ping-service
#               use                     host-service
#               max_check_attempts      10
#               contact_groups          admins, yritys-admins, yritys-admin2-247
#
#               service_description     PING
#               check_command           check_ping!500.0,50%!1000.0,80%
#               register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SER-
VICE, JUST A TEMPLATE!
#}

# yritys MDF
define host{
  use                     server-host  ; Name of host template to use
```

```
  host_name               yritys-MDF
  alias                   yritys MDF HP-runkokytkin
  address                 15.9.1.1
  parents                 yritys-router
}

# service definition
define service{
  use                     yritys-ping-service ; Name of service template to use

  host_name               yritys-MDF
}

# yritys IDF1
define host{
  use                     server-host  ; Name of host template to use

  host_name               yritys-IDF1
  alias                   yritys IDF1 HP-kuitukytkin
  address                 15.9.1.3
  parents                 yritys-MDF
}

#Service definition
define service{
  use                     yritys-ping-service  ; Name of service template to use

  host_name                yritys-IDF1
}

# yritys IDF1
define host{
```

```
  use                     server-host  ; Name of host template to use

  host_name               yritys-IDF1B
  alias                   yritys-IDF1B HP-kuitukytkin
  address                 15.9.1.7
  parents                 yritys -MDF
}

#Service definition
define service{
  use                     yritys-ping-service  ; Name of service template to use

  host_name               yritys-IDF1B
}

# yritys IDF2
define host{
  use                     server-host  ; Name of host template to use

  host_name               yritys-IDF2
  alias                   yritys IDF2 HP-kuitukytkin
  address                 15.9.1.4
  parents                 yritys-MDF
}

# Service definition
define service{
  use                     yritys-ping-service  ; Name of service template to use

  host_name               yritys-IDF2
}

# yritys IDF3
```

```
define host{
  use                     server-host  ; Name of host template to use

  host_name               yritys-IDF3
  alias                   yritys IDF3 HP-kuitukytkin
  address                 15.9.1.5
  parents                 yritys-MDF
}

# Service definition
define service{
  use                     yritys-ping-service  ; Name of service template to use

  host_name               yritys-IDF3
}

# yritys router
define host{
  use                     server-host  ; Name of host template to use

  host_name               yritys-router
  alias                   yritys Firewall
  address                 80.1.1.2
}

# Service definition
define service{
  use                     yritys-ping-service  ; Name of service template to use

  host_name               yritys-router
}

# yritys fw
```

```
define host{
  use                     server-host  ; Name of host template to use

  host_name               yritys-FW
  alias                   Cisco 871
  address                 80.1.1.1
  parents                 yritys-MDF
}
# Service definition
define service{
  use                     yritys-ping-service  ; Name of service template to use

  host_name                yritys-FW
}
```