TAMPEREEN
AMMATTIKORKEAKOULU

OPINNÄYTETYÖ

IBM TIVOLI NETCOOL-VERKONVALVONTAJÄRJESTELMÄN DOKUMENTAATIO
Case Corenet Oy

Tero Hyvärinen

Tietojenkäsittelyn koulutusohjelma
Toukokuu 2008
Työn ohjaaja: Harri Hakonen

HELSINKI 2008

TAMPEREEN
AMMATTIKORKEAKOULU

Tekijä(t) Tero Hyvärinen

Koulutusohjelma(t) Tietojenkäsittely

Opinnäytetyön nimi IBM Tivoli Netcool-verkonvalvontajärjestelmän dokumentaatio – Case: Corenet Oy

Työn valmistumis-kuukausi ja -vuosi Toukokuu 2008

Työn ohjaaja Harri Hakonen **Sivumäärä:** 73

TIIVISTELMÄ

Corenet Oy hankki vuoden 2007 alkupuolella IBM Tivoli Netcool -verkonvalvontajärjestelmän. Järjestelmä koostuu useammasta eri tuotteesta, joilla pystytään hoitamaan niin verkkolaitteiden valvontaa, tietoverkkojen kartoitusta kuin raportoimaan kattavasti verkon toimintaa. Jotta firman henkilöstöllä olisi selkeä kuva järjestelmän toiminnasta, täytyi järjestelmän toiminta dokumentoida. Tehtäväkseni tuli luoda kyseinen dokumentaatio Corenetin Netcool-järjestelmästä.

Opinnäytetyön tavoitteena oli kuvata Corenetin verkonvalvontajärjestelmän toiminta. Dokumentaation tuli sisältää sekä yleistä IBM Tivoli Netcool -järjestelmän toiminnankuvausta että Corenetin järjestelmään tehdyt muutokset. Lisäksi haluttiin kirjata ylös, kuinka Netcool-järjestelmä on integroitu yrityksen muihin järjestelmiin.

Koska tehtävänä oli luoda dokumentaatio verkonvalvontajärjestelmästä, teoreettiseksi viitekehykseksi valitsin yleisen verkonvalvontateorian sekä erityisesti verkonvalvonnassa käytettävät työkalut ja protokollat. Netcool-järjestelmän toiminta perustuu hyvin pitkälle juuri kyseisten protokollien, SNMP:n ja ICMP:n, käyttämiseen valvontadatan tuottamisessa.

Jo työn aloitusvaiheessa itselläni oli melko vahva tietämys verkonvalvonnasta, mutta tietämykseni IBM Tivoli Netcool -järjestelmän toiminnasta oli vielä melko vähäinen. Netcool-tietämyksen lisäämiseen käytin monia eri menetelmiä. Osallistuin kaksi viikkoa kestäneeseen koulutukseen, jossa opiskeltiin kolmen Netcool-tuotteen toimintaa. Toinen keino tiedon kartuttamiseen oli IBM:n luomien tuotedokumenttien lukeminen. Kolmas tutkimusmenetelmä oli järjestelmän ylläpitäjän Mikko Tepposen ja Netcool-projektin vetäjän Ari Löytynojan haastatteleminen. Mikkoa ja Aria haastattelin koko opinnäytetyöprojektini ajan ja sain heiltä erityisesti Corenetin järjestelmää koskevaa tietoa dokumentaatioon kirjattavaksi.

Työn tuloksena syntyi Corenetin Netcool-järjestelmän dokumentaatiosta versio 1.0, jonka luovutin yritykselle helmikuun viimeisellä viikolla. Corenetin edustajat pitivät luomaani dokumentaatiota hyvänä pohjana, jota he lähtevät laajentamaan projektin etenemisen myötä.

Dokumentaatiota luodessani huomasin, kuinka monimutkainen ja monipuolinen järjestelmä IBM Tivoli Netcool on. Jos järjestelmän toiminta haluttaisiin kuvata hyvin tarkasti, dokumentaatiosta olisi helposti tullut useamman sadan sivun mittainen. Tarkoituksena oli kuitenkin dokumentoida järjestelmän perustoiminta sekä Corenetin järjestelmäkohtaiset muutokset vakioasetuksiin. Tässä onnistuin mielestäni erittäin hyvin. Jatkossa dokumentaatioon tullaan lisäämään tarkemmin mm. kuinka Netcool-järjestelmä on integroitu yrityksen muihin järjestelmiin.

HUOM! Toimeksiantajan vaatimuksesta dokumentaatiosta tehtiin myös toinen, tämän opinnäytetyön liitteenä löytyvä, versio. Siitä poistettiin kaikki salaiseksi katsottava tieto, kuten IP-osoitteet, DNS-nimet ja Corenetin järjestelmäarkkitehtuurin kuvaukset.

Avainsanat IBM Tivoli Netcool Verkonvalvonta ICMP Traceroute SNMP

TAMPERE
POLYTECHNIC

Author(s)	Tero Hyvärinen
Degree Programme(s)	Business Information Systems
Title	Documentation of IBM Tivoli Netcool network monitoring system – Case: Corenet Oy
Month and year	May 2008
Supervisor	Harri Hakonen
Pages:	73

ABSTRACT

Corenet Inc. made a large investment at the beginning of 2007 and purchased IBM Tivoli Netcool network monitoring system. The system is made of several components that can be used to monitor network devices, discover networks and create versatile reports about network activities. My task was to create documentation for Corenet's Netcool system that company's other personnel could use to get acquainted with the system.

Goal of this thesis was to outline the operation of Corenet's Netcool system. It was decided that documentation should include both generic IBM Tivoli Netcool information as well as specific information concerning how Corenet's system has been setup. Integration of Netcool system to other systems of the company was also needed to be documented.

The task was to create documentation for network monitoring system and thus I chose network monitoring theory as the theoretic context. I set the main focus to network monitoring tools and protocols. Especially to SNMP and ICMP. Both of these protocols are also basis that Netcool system uses to create network monitoring data.

As I was starting this project I had quite a good knowledge to network monitoring. On the other hand my Netcool knowledge wasn't very strong. I used several methods to gain more knowledge to Netcool system. I took part in two-week official Netcool training which included three of the Netcool subsystems.The second method was to read official IBM product documents for different Netcool products. The third method was to interview the system administrator Mikko Tepponen and the project manager for Ari Löytynoja. I had many discussions with Mikko and Ari and got most of the system-specific information regarding Corenet's Netcool system from them.

As a result I created system documentation version 1.0 for Corenet's Netcool system. I hand it out to company during the last week of February 2008. Corenet representatives told that documentation is good and it's good basis for their future needs.

During the process I noticed how complex and versatile IBM Tivoli Netcool system is. If all system functionalities was to be documented in full detail, the documentation could easily become many hundred pages in length. My task was to document basic system functionalities and Corenet-specific changes to system defaults. In this I succeeded very well. In the future Corenet has intention to expand this documentation further, f.ex integration of Netcool system to other Corenet systems will be documented in more detail.

NOTE! The employer demanded that second version of the documentation needed to be created. This version does not include any Corenet-specific system information such as IP addresses, DNS names and system architecture. This version of the documentation can be found as attachment to this thesis.

Keywords IBM Tivoli Netcool Network monitoring ICMP SNMP

Sisällysluettelo

Käsitteet

ICMP (Internet Control Message Protocol) – Yksi TCP/IP-protokollapinon perusprotokollista. Perimmäinen tarkoitus on virhetilanteiden ja muiden huomiota vaativien tilanteiden kommunikointi laitteelta toiselle.

Traceroute – Yksinkertainen sovellus, joka käyttää joko ICMP tai UDP paketteja näyttääkseen lähde- ja kohdepisteen välillä olevien reitittävien laitteiden tiedot sekä vasteajat.

SNMP (Simple Network Management Procotol) – TCP/IP-verkoissa yleisesti verkonvalvonnassa käytettävä protokolla. Protokollasta on kolme eri versiota SNMPv1, SNMPv2 ja SNMPv3.

MIB (Management Information Base) – Määrittää muuttujat, joita verkkolaite ylläpitää. Kaksi eri versiota, MIB ja MIB-II.

SMI (Structure of Management Information) – Kokoelma yleisiä rakenteita ja identifointiskeema. Määrittää SNMP:n käyttämän tiedon muodon.

OID (Object Identifier) - MIB-muuttujien nimien määrittämiseen käytettävä ISO:n ja ITU:n hallitsemaa hierarkinen nimiavaruutta.

Hallittava laite (Managed System) - Verkkokomponentti, jota valvotaan ja hallitaan SNMP-protokollan avulla. Sisältää agentin.

Agentti (Agent) – Osa verkonvalvontajärjestelmää, joka sijaitsee hallittavassa laitteessa. Käsittelee hallittavan laitteen hallintatietoa ja muokkaa sen SNMP-yhteensopivaan muotoon

Verkonvalvontajärjestelmä (Network Management System) – Käynnistää sovelluksia, jotka valvovat ja hallitsevat hallittavia laitteita. Yksittäisille tai useille palvelimille kasattuja järjestelmäkokonaisuuksia, jotka prosessoivat verkonhallinta-informaatiota.

IBM Tivoli Netcool – Kaupallinen verkonvalvontajärjestelmä. Yksi monipuolisimmista ja pisimmälle kehitetyistä verkonvalvontajärjestelmistä.

1 Johdanto

Internetin kasvun myötä myös yritysten tietoverkoista on tullut kriittinen osa yritysmaailmaa. Nykyisin käytännössä kaikki tieto löytyy yritysten verkkopalvelimilta, joten tietoverkkojen toimivuudella on hyvin suuri merkitys. Verkkolaitteet ovat kuitenkin vain tavallisia tietokoneita, jotka ovat alttiita laite- ja ohjelmavioille, sähkökatkoksille sekä muille ongelmille. Suurilla yrityksillä on verkkolaitteita ja -palvelimia helposti sadoittain, jopa tuhansittain. Vika jo yhdessä niistä voi aiheuttaa suuria ongelmia yrityksen liiketoimintaan. Siksi onkin tärkeää myös valvoa laitteiden toimivuutta ja ongelmia. Verkonvalvontaprotokollia, kuten SNMP:tä, hyväksikäyttävät verkonvalvontaohjelmistot pystyvät kertomaan muutamissa sekunneissa, kun yrityksen verkossa tapahtuu jotain odottamatonta. Näin yrityksen IT-henkilöstö pystyy nopeasti reagoimaan tilanteeseen ja viasta aiheutunut haitta liiketoimintaan jää mahdollisimman pieneksi.

1.1 Toimeksiantaja Corenet Oy

Opinnäytetyöni toimeksiantaja on Corenet Oy (jäljempänä Corenet). Yritys on perustettu nykyisessä muodossaan vuonna 1998. Yrityksen omistavat VR-Yhtymä Oy ja TDC Oy. VR:n osuus osakkeista on 60 % ja TDC:n 40 %. (Corenet yritysesittely, 2007.) VR-omistajuuden juuret ovat kuitenkin kauempana menneisyydessä. Corenet oli alkujaan osa VR:tä, josta yritys muuttui itsenäiseksi 1980-luvulla, jolloin VR jaettiin pienimpiin yrityksiin. Alkuun yrityksen nimi oli Railtelia.

Corenet on korkealaatuisia televerkkoratkaisuja tuottava yritys. Corenetin palveluihin kuuluvat televerkkojen ja telemaattisten järjestelmien suunnittelu, rakentaminen ja ylläpito sekä toimisto- ja asiakaspalvelujärjestelmien tiedonsiirtoratkaisut. Lisäksi Corenetillä on oma siirtoverkko, jonka kapasiteetista suuri osa on vuokrattu ulkoisille asiakkaille. Corenetin tuottamien palveluiden osalta Corenet on keskittynyt suurelta osin rautatieympäristöön merkittävimpien asiakkaiden ollessa Valtion Rautatiet (VR), Ratahallintokeskus (RHK) ja muut VR-konserniin kuuluvat yritykset. Yrityksellä on kuitenkin myös monia VR-konsernin ulkopuolisia asiakkaita Suomen perusinfrastruktuurin eri osa-alueilta. Perusinfrastruktuurin alueelta Corenet on myös vahvasti hakemassa uusia asiakkuuksia ja laajentumisvaraa. Opinnäytetyöni aiheena oleva Valluvalvontajärjestelmä on suuri panostus juuri VR-konsernin ulkopuolisille markkinoille ja TCP/IP-maailmaan.

Corenetin liikevaihto vuonna 2007 oli 31,4 miljoonaa euroa ja henkilöstöä yrityksessä on n. 210 henkeä 14 paikkakunnalla. Corenetin siirtoverkolla on mittaa n. 5800 kilometriä, josta yli puolet on toteutettu kuidulla. Siirtoverkostaan Corenet tarjoaa asiakasyrityksille jopa 2,5 Gbit/s siirtoyhteyksiä. (Corenet yrityesittely, 2007)

1.2 Opinnäytetyön tavoite

Loppukesästä 2006 Corenet päätti panostaa TCP/IP-verkonvalvontaan hankkimalla Cygaten ulkoistaman n. 10-henkisen verkonvalvontatiimin. Seuraavan vuoden aikana Corenetillä tehtiin ehkäpä vieläkin suurempi panostus ja hankittiin ”täysverinen” verkonvalvontajärjestelmä, IBM Tivoli Netcool. Corenetin Netcool-järjestelmä ristittiin Vallu-valvontajärjestelmäksi. Verkonvalvonnan lisäksi järjestelmä pystyy mm. kattavien raporttien luontiin.

Tämän opinnäytetyön tavoitteeksi asetettiin Vallu-järjestelmän dokumentaation tuottaminen. Tavoitteeksi asetettiin heti alussa Vallu-valvontajärjestelmän 1.0-version dokumentointi. Luotavassa tuotoksessa oli tavoitteena dokumentoida Vallu-järjestelmän toiminnan ja konfiguroinnin ydinalueet tiivistetysti, mutta kattavasti. Toisekseen tavoitteena oli dokumentoida Corenetin järjestelmän arkkitehtuuri ja järjestelmään tehdyt muutokset vakioasetuksiin nähden.

Tämän opinnäytetyön liitteenä on versio Corenetille luodusta dokumentaatiosta. Tietoturva- ja yrityssalaisuussyistä se on kuitenkin muutettu yleiseksi kuvaukseksi IBM Tivoli Netcool -järjestelmän toiminnasta. Toisin sanoen liitteenä oleva dokumentaatio ei sisällä ollenkaan tietoa Corenetin järjestelmän arkkitehtuurista ja toiminnasta.

2 Verkonvalvonta

Viimeisten kahdenkymmenen vuoden aikana tietoliikenneverkoista on tullut hyvin tärkeä osa joka päiväistä yrityselämää. Nykyisin yksikään yritys ei kykene pärjäämään kiristyvillä markkinoilla ilman Internet-yhteyttä ja omaa tietoverkkoa. Yritykset myyvät ja markkinoivat tuotteitaan verkon kautta, monet sovellukset ja ydintieto löytyvät verkkopalvelimilta, asiakkaisiin pidetään yhteyttä sähköpostin kautta ja Voice-over-IP-järjestelmien yleistyessä yhä useammissa yrityksissä myös puhelut siirtyvät TCP/IP-verkkoihin.

Tietoliikenneverkkojen merkityksen kasvaessa myös verkon aktiivilaitteiden, kuten reitittimien, palomuurien ja verkkopalvelimien merkitys on kasvanut hyvin suureksi. Koska verkkolaitteet ja -palvelimet ovat tietokoneita, jotka voivat minä hetkenä hyvänsä toimia virheellisesti tai pahimmassa tapauksessa hajota, on myös verkkolaitteiden valvonnasta tullut tärkeä osa yritysmaailmaa. Tehokkaasti ja kannattavasti toimiakseen yritykset tarvitsevat toimivan tietoverkon, jonka ongelmatilanteet pitää pystyä havaitsemaan ja niihin täytyy pystyä reagoimaan mahdollisimman nopeasti.

2.1 Verkonvalvonnan työkalut

Suurilla monikansallisilla yrityksillä on helposti satoja toimipisteitä ympäri maailmaa. Joka toimipisteessä on yleensä useita verkkolaitteita, jolloin verkonvalvonnasta tulee lähes mahdotonta ilman toimivia työkaluja. Niinpä verkkolaitteiden ja -palvelimien toiminnan valvontaan on olemassa monenlaisia standardoituja protokollia ja niitä hyödyntäviä ohjelmistoja. Näitä protokollia ovat mm. RMON (Remote Monitoring) ja SNMP (Simple Network Monitoring Protocol). Niistä kerron tarkemmin seuraavassa luvussa.

Kuten ohjelmistomarkkinoilla yleensäkin, verkonvalvontaohjelmistoista löytyy sekä ilmaisia, esim. Cacti, että maksullisia, esim. IBM Tivoli Netcool, sovelluksia ja sovellusperheitä. Nimensä mukaisesti ilmaisohjelmistot ovat ilmaisia käyttää perusominaisuuksiensa osalta, mutta usein kehittyneemmät ja monimutkaisemmat ominaisuudet ovat niissäkin lisenssoituja – toisin sanoen maksullisia. Maksullisten ohjelmistojen hyötyjä ovat mm. valmistajan antama tuki ja takuu tuotteille sekä kattavat käyttöohjeet. Lisäksi valmistaja testaa tuotteitaan jatkuvasti ja ohjelmistobugien korjaus on useimmiten nopeampaa kuin ilmaisohjelmistojen kohdalla.

Sopivan verkonvalvontasovelluksen valinta on usein vaikeaa. Valitaanko kenties ilmainen, vähemmillä ominaisuuksilla varustettu sovellus? Vai onko kuitenkin järkevämpää tehdä kerralla kunnon panostus ja hankkia valvontajärjestelmä, jolle on mahdollisuus saada myös aina tarvittaessa

nopeasti tukea valmistajalta? Lisäksi pitäisi vielä löytyä tai palkata henkilö, yleensä useita henkilöitä, konfiguroimaan ja ylläpitämään järjestelmän toimintaa. Jo järjestelmän konfigurointi toimimaan täysin halutulla tavalla on aina aikaa ja ammattitaitoa vaativa projekti. Ja tämä ei koske ainoastaan ilmaisohjelmistoja, kuten Netcool-järjestelmän dokumentaatioita tutkiessani tulin huomaamaan. Netcool-järjestelmän, joka osaa valvonnan lisäksi luoda verkkokuvia ja monenlaisia raportteja verkon ja laitteiden toiminnasta, konfigurointi ja ylläpito on hyvin työlästä ja ammattitaitoa vaativaa työtä. Kalliiseen järjestelmään on luonnollisesti olemassa kattavat ja tarkat konfigurointiohjeet ja valmistajalta, IBM:ltä, saa aina tarvittaessa apua ongelmatilanteissa. Mutta siitä huolimatta järjestelmän täydelliseen toimintakuntoon saattaminen vaatii valtavasti aikaa ja ammattitaitoa. Voin vain kuvitella, millainen homma olisi konfiguroida samat ominaisuudet käyttöön johonkin ilmaisohjelmaan, jossa ohjeet eivät välttämättä ole yhtä kattavat ja valmistajaltakin saa tukea paljon heikommin.

2.2 Valvontapalvelun ulkoistaminen

Kuten edellisestä kappaleesta käy ilmi, tietoverkonvalvonta vaatii aikaa, rahaa, ammattitaitoa ja oikeat työkalut. Monille yrityksille se tarkoittaa kokonaisen asiantuntijatiimin palkkaamista, tuhansia työtunteja ja miljoonien eurojen menoja sovellusten ja henkilöstökulujen kautta. Tästä syystä verkonvalvonnan ulkoistaminen on yleistymässä. Verkonvalvonnan ostaminen ulkopuoliselta asiantuntijayritykseltä on huomattomasti vaivattomampi ja halvempi ratkaisu kuin oman verkonvalvontatiimin ja valvontajärjestelmän hankkiminen.

Kun yhä useammat yritykset ulkoistavat verkonvalvontansa, mitä hyötyjä ulkoistamisesta sitten oikeastaan saadaan? Ensinnäkin valvonnan ulkoistaminen on taloudellisesti hyvä ratkaisu. Kuten mainittua, oman verkonvalvonnan kasaaminen on hyvin kallista puuhaa. Varsinkin, jos asiat halutaan tehdä kunnolla. Lisäksi ammattitaitoisen henkilöstön löytäminen verkonvalvontatiimiin voi olla hyvinkin vaikeaa. Kun verkonvalvonta, ja yleensä myös verkonhallinta, ostetaan palveluna asiantuntijayritykseltä, saadaan kohtuulliseen hintaan taatusti ammattitaitoiset ihmiset ja toimivat järjestelmät hoitamaan tehtävää. Palvelu toimii myös useimmiten ympärivuorokautisena vuoden jokaisena päivinä. Tietenkin sama voidaan myös toteuttaa oman henkilökunnan avulla, mutta se kuitenkin tarkoittaa lisätyövoiman palkkaamista useampaan vuoroon sekä selvästi lisääntyviä palkkakustannuksia mm. yö- ja sunnuntailisien muodossa.

Verkonvalvontapalvelua tarjoavien yritysten merkittävä vahvuus on ammattitaito. Tuntemus aktiivilaitteiden toimintaan ja konfigurointiin on vahva jo asiantuntijaorganisaation alimmilla portailla. Todella vaikeita tilanteita varten yrityksillä on yleensä omat erikoisasiantuntijansa, jotka ovat perehtyneet juuri tietynlaisiin järjestelmiin ja laitteisiin syvällisesti.

Tämä tarkoittaa käytännössä laitteita ja järjestelmiä, joita kyseinen yritys myy ja edustaa. Näin ongelmatilanteet saadaan usein ratkottua nopeammin. Lisäksi yrityksellä on yleensä jonkinasteinen kumppanuussuhde laitteiden valmistajaan ja tämän ansiosta myös laitevalmistajan suunnalta saadaan huomattavasti nopeammin ja kattavampaa tukea.

Verkonvalvonnan ulkoistamisessa on kuitenkin myös monia huonoja puolia. Suurin osa näistä johtuu siitä, että ulkopuolinen yritys joudutaan päästämään oman verkon sisäpuolelle. Tästä seuraa luonnollisesti tietoturvauhkia. Valvonnan toteuttaminen tarkoittaa usein, että omaan verkkoon joudutaan sallimaan vieras palvelin tai palvelimia eli ns. probepalvelin, joka hoitaa itse valvontaa. Kyseinen palvelin lähettää kokoamansa tiedot eteenpäin valvontaa hoitavan yrityksen verkossa olevalle palvelimelle, jolloin palomuuriin joudutaan luomaan uusia aukkoja. Lisäksi, jos myös verkkolaitteiden hallinta on ulkoisella yrityksellä, kuinka voidaan taata, ettei salaista tietoa tule leviämään sivullisten tietoon.

Jo sana ulkoistaminen itsessään kuvaa yhtä ulkopuoliselta yritykseltä palveluna hankittavan verkonvalvonnan aiheuttamaa ongelmaa. Eli palvelu on ulkoistettu eikä se ole enää omissa käsissä. Palvelusopimuksissa on aina jonkinlainen vasteaika, esim. ongelmatiketti pitää olla avattuna 15 minuutin kuluessa ongelman havaitsemisesta ja vika korjattuna 4 tunnissa. Tämä tarkoittaa, että palvelua tarjoavan yrityksen ei tarvitse reagoida ongelmiin samantien, kuten oma henkilöstö tekisi. Palvelua tarjoava yritys ei edes yleensä halua reagoida ongelmatilanteisiin heti, vaikka siihen olisikin mahdollisuus. Ajatellaanpa tilanne, jossa asiakkaalle on myyty edellä mainittu ”15 minuuttia tiketti auki, 4 tuntia vika korjattu” -palvelu. Yrityksessä kuitenkin tartutaan ongelmatilanteisiin jo 5 minuutin sisällä ongelma ilmaantumisesta ja vikakin saadaan yleensä ratkaistua alle tunnissa. Tästä aiheutuu palveluntarjoajalle ongelmia kiiretilanteissa. Jos tällöin tiketti avataan sopimuksen mukaan 15 minuutissa ja vika saadaan 4 tunnissa korjattua, yritys on toiminut sopimuksen mukaan. Asiakkaasta kuitenkin tuntuu helposti, että palvelu oli huonoa, kun vika ei ollutkaan ohi puolessa tunnissa, kuten normaalisti. Käytännössä asiakasyritystä on ylipalveltu aiemmin ja asiakas on opetettu ”liian hyvälle”. Käytännössä tämä siis tarkoittaa, että asiakasta palvellaan sopimuksen mukaisilla vasteajoilla, josta seurauksena on pieni viive.

Verkonvalvonnan ulkoistamisessa on myös toinen ongelma vasteaikojen suhteen. Kuinka palvelua tarjoava yritys pystyy takaamaan vasteaikojen mukaisen palvelun? Tietenkin näihin tilanteisiin sovitaan sanktiot, mutta siltikin oma henkikökunta pyrkii aina nopeammin reagoimaan ja ratkomaan ongelmatilanteet. Varsin ongelmalliseksi tämä tilanne muuttuu varsinkin silloin, kun palveluntarjoajalla on paljon asiakkaita ja kuitenkin rajatut resurssit. Eli kuinka voidaan taata, että juuri meidän yrityksemme ongelmatilanteita pyritään ratkomaan suurella prioriteetilla? Yksinkertainen vastaus tähän on tietenkin raha, mutta ulkoistuksen yhtenä tärkeimpänä motiivina oli juuri säästöjen saaminen.

3 Verkonvalvonnassa käytettävät työkalut

Ilman oikeanlaisia työkaluja verkonvalvonta on lähes mahdotonta. Tästä syystä onkin kehitetty monenlaisia protokollia verkonvalvonnan käyttöön. Osa näistä protokollista, kuten SNMP, on kehitetty juuri verkonvalvontaa varten. Osa protokollissa, kuten ICMP, taas on kehitetty jostain muista lähtökohdista, mutta on myöhemmin huomattu hyödylliseksi myös valvontakäytössä. Seuraavassa esittelen kolme näistä protokollista. Niistä erityisesti SNMP ja ICMP ovat merkittävässä roolissa myös IBM Tivoli Netcool -järjestelmissä verkonvalvontadatan tuottajina.

3.1 Internet Control Message Protocol

Internet Control Message Protocol eli ICMP on yksi TCP/IP-protokollapinon perusohjelmista. ICMP:n tunnetuinta sovellusta kutsutaan nimellä ping sen toimintaperiaatteen vuoksi. Protokollan toimintatapa on hyvin tuttu sukellusveneistä, joissa Sonar-tekniikan avulla pinnan alla olevat sukellusveneet pystyivät löytämään pinnalla purjehtivia laivoja. Sonar käyttää kaikuluotausta eli ääniaaltoja ja niiden takaisinheijastumista kohteiden sijainnin paikantamiseen. Samoin ICMP kaikuluotaa eli lähettää ICMP echo request -paketteja kohdeosoitteille. Jos osoitteen omaava laite vastaa ICMP echo reply -paketilla, saadaan selville laitteen olemassa olo. Ping myös mittaa ajan, joka vastauksen saamiseen kuluu ja pystyy näin raportoimaan tavoitettavuuden lisäksi vasteajan eli RTT:n (Round-Trip-Time) (Ballew 1998: 201).

3.1.1 ICMP:n tarkoitus

ICMP:tä ei alun perin kehitetty verkonvalvontaa silmällä pitäen. ICMP:n perimmäinen tarkoitus on virhetilanteiden ja muiden huomiota vaativien tilanteiden kommunikointi eli ICMP:n tarkoitus on olla virheiden raportointijärjestelmä (Comer 2002: 130). ICMP:n avulla esim. reitittimet pystyvät lähettämään virhettä koskevan sanoman alkuperäiselle lähettäjälle. Näin myös alkuperäisen paketin lähettänyt laite saa selville, että vastaanottava laite ei esim. ollut tavoitettavissa ja paketti ei mennyt perille. Alkuperäisessä paketissa on sekä lähettäjän että vastaanottajan osoitetiedot. Kun reititin havaitsee ongelman ja toteaa, että vastaanottaja ei ole tavoitettavissa, se lähettää ICMP:n avulla tiedotuksen alkuperäiselle lähettäjälle kohdatusta ongelmasta.

3.1.2 ICMP-sanomien kuljetus ja rakenne

ICMP-viestit kuljetaan normaaleissa IP-paketeissa. Ne kapseloidaan kahteen otteeseen ja kuljetaan IP-tietosähkeen data-osassa (Comer 2002:

131). ICMP-sanomia on useanlaisia ja niiden muoto on erilainen kolmea ensimmäistä kenttää lukuunottamatta. Nämä kolme kenttää ovat type (8 bittiä), joka yksilöi sanoman tyypin, code (8 bittiä), joka sisältää tarkempaa tietoa sanoman tyypistä sekä checksum (16 bittiä), jota käytetään sanoman eheyden tarkistamiseen (Postel 1981: 2). Type-kenttä siis määrittää viestin tarkoituksen ja rakenteen. RFC 792-dokumentissa määritellyt Type-kentän arvot näemme taulukosta 3.1.2:

Taulukko 3.1.2: ICMP-viestien tyypit (Postel 1981: 20)

Tyyppi	ICMP-viestin laji
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect (change a route)
8	Echo Request
11	Time Exceeded for a Datagram
12	Parameter Problem on a Datagram
13	Timestamp Request
14	timestamp Reply
15	Information Request
16	Information Reply

3.1.3 Kohteen tavoitettavuuden testaaminen (ping)

Ping on ehkäpä yleisimmin ja laajimmin verkonvalvonnassa käytetty työkalu. Sen toiminta perustuu ICMP echo request - ja ICMP echo reply -sanomien käyttöön. Pingatessa laite lähettää echo request -sanoman kohdeosoitteeseen. Laite, jolle osoite kuuluu, vastaa saamaansa pakettiin echo reply -sanomalla. Näin saadaan selville, onko vastaanottava laite tavoitettavissa. Vastauksen saaminen osoittaa, että laitteiden välillä on IP-tason yhteys toisiinsa. Tarkemmin ottaen vastaus todistaa, että lähettävä laite pystyy lähettämään paketteja verkkoon, yhteys ja reititys kohdeverkkoon ja takaisin toimii sekä vastaanottava laite on toiminnassa ja sen ICMP- sekä IP-ohjelmat toimii (Comer 2002: 133).

Pingin käytössä on kuitenkin omat ongelmansa. Sen avulla saadaan vain melko vähän tietoa verkosta. Lisäksi saadut tiedot saattavat olla harhaanjohtavia. Lähtevä ja palautuva paketti eivät välttämättä kulje samaa reittiä. Epäsymmetrisen reitityksen takia ei tiedetä, mikä reiteistä on vikaantunnut (Ballew 1998: 202). Jos vastaus saadaan, tiedetään vain, että jotain reittiä pitkin laite on tavoitettavissa, mutta käytetystä meno- sekä paluureitistä ei ole tietoa. Laitteen tavoitettavuuden valvonnassa ping onkin omimmillaan ja hyvin hyödyllinen. Esim. IBM Tivoli Netcool -järjestelmä lähettää ICMP-pakettaja valvottaviin osoitteisiin ja saamiensa vastausten perusteella pystyy valvomaan, ovatko laitteet ”hengissä”.

3.2 Traceroute

Ping on käytännöllinen työkalu, kun halutaan tietää, onko kohdelaite tavoitettavissa. Tämä on myös pingin ongelma, sillä se kertoo ainoastaan kohdelaitteen tavoitettavuuden, mutta ei ollenkaan matkan varrella olevia laitteita. Toisin sanoen ping ei kerro, mitä reittiä paketti kulkee kohteeseensa, jolloin matkan varrella olevat reititysongelmat ja pääreitin varrella olevat ”kaatuneet” reitittimet jäävät huomaamatta. Traceroute korjaa juuri tämän ongelman. Sen avulla pystytään pakottamaan jokainen matkan varrella oleva OSI-mallin 3-kerroksella toimiva laite (yleensä reititin) vastaamaan ja kertomaan oman osoitteensa.

3.2.1 Tracerouten toiminta

Traceroutesta on olemassa useita implementaatioita. Tracerouten yleisin versio käyttää hyödykseen ICMP-paketteja. Traceroute perustuu IP-paketin TTL (time-to-live) -kentän käyttöön. Lähettävä laite asettaa jokaiseen IP-pakettiin TTL-arvon, jonka arvoa jokainen reititin matkan varrella pienentää yhdellä. Kun TTL-arvo saavuttaa arvon nolla, reititin hylkää paketin ja lähettää ICMP Time-to-live Exceeded -paketin lähettäjälle. Tässä paketissa lähdeosoitteena on paketin hylänneen reitittimen oma osoite. Traceroute toimii siis siten, että se lähettää ensin paketin TTL-arvolla 1, jolloin ensimmäinen paketin käsittelevä reititin lähettää takaisin Time-to-live Exceeded -sanoman. Seuraavaan pakettiin TTL-arvoksi asetetaan 2, jolloin TTL Exceeded -sanoma saadaan reitin varrella seuraavalta reitittimeltä. (Understanding the Ping and Traceroute Commands 2006.) Seuraavaan pakettiin TTL-arvoksi laitetaan 3 ja jokaiseen seuraavaan pakettiin TTL-arvoa aina kasvatetaan yhdellä, kunnes kohdeosoite saavutetaan tai TTL-arvo ylittää määritetyn maksimiarvon, joka on oletuksena yleensä 30, ja jäljitys lopetetaan.

ICMP-tracerouten lisäksi on olemassa muitakin versioita traceroutesta. UDP-protokollan sijaan TCP-protokollaa apunaan käyttää esimerkiksi Tcptraceroute-sovellus. Tcptraceroute ja muut TCP-traceroute-sovellukset ovat käteviä, kun ICMP- ja UDP-protokollien avulla ei pystytä selvittämään yhteysongelmaa. Sovelluksia käytettäessä voidaan antaa monenlaisia asetuksia lähetettäville paketeille, joista tärkeimpänä käytettävä kohdeporttinumero. Ongelma voi olla esimerkiksi, että kohdelaite vastaa pingatessa, mutta jokin tietty verkkosovellus ei toimi. Koska ICMP-paketit menevät läpi, laitteiden välillä on IP-tason yhteys, mutta jostain syystä verkkosovelluksen käyttämään porttinumeroon menevä liikenne ei pääse läpi. Tällöin tcptracerouten avulla voidaan löytää laite, joka estää kyseisen liikenteen ja pyrkiä korjaamaan ongelma.

Kolmas tracerouten implementaatio on pingin ja tracerouten yhdistävät sovellukset. Windows-käyttöjärjestelmissä on käytössä PathPing-sovellus (PathPing 2008). Toinen hyvä esimerkki on My traceroute.

3.2.2 Tracerouten ongelmat

Traceroute on siis hyvä työkalu, kun halutaan selvittää, mitä reittiä pitkin paketti kulkee kohdeosoitteeseen. Sen toiminnassa on kuitenkin omat ongelmansa, jotka johtuvat useimmiten matkalla olevien laitteiden toteutuksesta. Jotkin laitteet eivät toimi täysin standardin mukaisesti ja eivät generoi Time-to-live Exceeded -sanomaa (Ballew 1998: 202). Lisäksi useissa laitteissa voidaan määrittää, että laite ei vastaa ICMP-paketteihin. Molemmissa tapauksissa kyseiseltä laitteelta ei saada Tracerouten avulla vastausta. Tämä ei kuitenkaan yleensä ole ongelma, sillä useimmiten reitillä seuraava laite vastaa, jolloin reitin varrelle jää yksi laite, joka ei vastaa ja jää tunnistamatta.

Suurempi ongelma tracerouten kannalta on palomuurit. Monet traceroute-ohjelmat lähettävät UDP-datagrammin sattumanvaraisesti valittuun korkeaan porttinumeroon. Koska palomuurien on tarkoitus pitää ylimääräiset ja luvattomat yhteydet kurissa, ne konfiguroidaan estämään nämä korkeaan porttiin tulevat UDP-yhteydet. Tällöin palomuuri yksinkertaisesti pysäyttää tracerouten. (Ballew 1998: 202)

Nykyisin yritykset yhdistävät lisääntyvässä määrin eri toimipisteensä VPN-tunnelien avulla. Tähän liittyy tracerouten kolmas ongelma. Kun liikenne lähetetään VPN-tunneliin, paketin sisältö kryptataan ja puretaan vasta VPN-tunnelin toisessa päässä. Koska paketin sisältö on salattu, eivät matkan varrella olevat reitittimet näe siitä kuin osoitetiedot ja tracerouten toiminta estyy.

3.3 Simple Network Management Protocol

Simple Network Management Protocol eli SNMP on selvästi yleisimmin käytetty verkonhallintaprotokolla. SNMP-protokollan yksi suurimmista vahvuuksista on, että se on suunniteltu helposti käytettäväksi verkonhallintatyökaluksi. Erityisesti TCP/IP-verkoissa SNMP-protokollan käyttö on melko yksinkertaista, eikä se vaadi mitään monimutkaisia komentorakenteita hallittaviin laitteisiin.

3.3.1 SNMP:n historia

Vaikka idea verkonhallintaprotokollasta ja -työkalusta on ollutkin olemassa jo 70-luvun lopulta alkaen, SNMP-protokollan historia alkaa vasta 80-luvun lopulla. Idea SNMP-protokollasta voidaan katsoa syntyneen vuonna 1987 järjestetyssä tietoverkkojohtajien tapaamisessa. Kyseisen tapahtuman ruokailun aikana SNMP-protokollan keksijöinä yleisesti pidetyt Jeffrey Case, James Davin, Mark Fedor ja Marty Schoffstall kehittivät SNMP-protokollan edeltäjän Simple Gateway Monitoring Protocol

(SGMP) ja hakivat sille IETF:n määrittelyä RFC1028-dokumentilla. (Simoneau 1999: 8)

SGMP-protokollan kanssa samaan aikaan toisaalla kehitettiin kahta muuta verkonhallintaprotokollaa – High-Level Entity Management System (HEMS) ja Common Management Information Protocol (CMIP). HEMS määriteltiin RFC-dokumenteissa 1021 – 1024 ja CMIP-protokollan ominaisuudet julkaistiin kolmessa ISO-dokumentissa. (Simoneau 1999: 8)

IAB:n (Internet Activities Board) kokouksessa helmikuussa 1988 päätettiin, että SNMP olisi verkonhallinnan lyhyen tähtäimen ratkaisu, kun taas CMIP olisi pidemmän tähtäimen ratkaisu (Simoneau 1999: 8). Tuolloin ajateltiin vielä OSI-protokollien vievän nopeasti voiton TCP/IP-protokollista. Koska SNMP oli suunniteltu juuri TCP/IP-maailmaan, sen ei nähty saavuttavan pidemmän päälle jalansijaa. (Hautaniemi 1994.) Kuten nykyisin tiedämme, kyseinen näkemys oli täysin väärä. TCP/IP-protokollapino saavutti huikean suosion ja se valloitti lähes koko Internetin.

Vuoden 1989 aikana SNMP-protokollasta tuli merkittävin verkonhallintaprotokolla TCP/IP-verkoissa. Suurin osa laitevalmistajista tuki protokollaa ja myös käytti sitä omassa verkossaan. Myös seuraava vuosi oli merkittävä SNMP-protokollan kannalta, sillä vuonna 1990 se hyväksyttiin viralliseksi Internet-standardiksi. Virallisiksi standardeiksi tulivat RFC1157- (SNMP), RFC1155- (SMI) ja RFC1156- (MIB) dokumentit. MIB-standardi korvattiin vielä seuraavan vuoden huhtikuussa parannellulla ja lisäominaisuuksia sisältävällä MIBII-standardilla (RFC1213). (Hautaniemi 1994)

Vuonna 1991 SNMP-siirtokerroksen protokollaksi määriteltiin UDP RFC1270-dokumentissa. UDP valittiin TCP:n sijaan, koska SNMP suunniteltiin toimimaan myös erittäin isoissa verkoissa. Protokollan kehittäjät näkivät, että TCP:n käytöstä syntyisi vain ylimääräistä ja turhaa kuormitusta verkkoon. Samana vuonna tehtiin määrittelyt myös SNMP-protokolla käyttämisestä OSI-verkoissa (RFC1283). Seuraavan vuoden aikana tehtiin määrittelyjä mm. FDDI- ja Frame Relay -verkkojen integroinnista SNMP-protokollaan. (Simoneau 1999:9)

Ensimmäiseksi standardiksi hyväksytyssä SNMP-protokollassa oli kuitenkin vielä joitain puutteita. Erityisesti protokollan turvallisuuteen oli kehitettävä parannuksia. Tässä vaiheessa SNMP ei sisältänyt ollenkaan autentikointia. Pääasiassa tästä syystä alettiin kehittää uutta versiota SNMP-protokollasta ja se sai nimekseen SNMPv2. Siitä ei kuitenkaan tullut kovinkaan onnistunut protokolla, joten uudempaakin protokollaa, nimeltään SNMPv3, on kehitelty ja siitä toivotaan muodostuvan alkuperäisen SNMP:n korvaaja. SNMPv2- ja SNMPv3-protokollista kerron tarkemmin kappaleissa 3.3.6 ja 3.3.7.

3.3.2 TCP/IP-verkonhallinnan komponentit

TCP/IP-verkonhallinta koostuu kolmesta osatekijästä, jotka määrittelevät SNMPv1-protokollan (Stevens 2000: 359 – 360):

1. *Management Information Base* (MIB), joka määrittää muuttujat, joita verkkolaite ylläpitää. RFC 1213 määrittää toisen MIB-version (MIB-II) toiminnan.
2. Kokoelma yleisiä rakenteita ja identifointiskeema, joita käytetään MIB-muuttujiin viittaamisessa ja kutsutaan nimellä *Structure of Management Information* (SMI). Toisin sanoen SMI määrittää SNMP:n käyttämän tiedon muodon. RFC 1155 määrittelee SMI:n.
3. Protokolla, joka toimii verkonhallintajärjestelmän ja laitteen välillä, on *Simple Network Management Protocol*. RFC 1157 määrittelee protokollan.

3.3.3 SNMP:n peruskomponentit

SNMP-protokollassa on kolme peruskomponenttia:

1. hallittavat laitteet
2. agentit
3. verkonhallintajärjestelmät

Hallittavat laitteet ovat verkkokomponentteja, joita SNMP-protokollan avulla valvotaan ja hallitaan. Hallittavat laitteet sisältävät agentin. Ne myös keräävät ja tallentavat hallintainformaatiota, joka on verkonhallinta-järjestelmien käytettävissä. (Cisco Internetworking Technology Handbook: Simple Network Management Protocol 2008.) Informaatiota ovat esimerkiksi laitteiden konfiguraatiot sekä lokiin kerätyt tiedot. Hallittavia laitteita ovat mm. reitittimet, palomuurit ja palvelimet.

Agentti on verkonhallintasovelluksen osa, joka sijaitsee hallittavassa laitteessa. Agentti tuntee ja käsittelee kyseisen laitteen hallintatietoa ja muokkaa sen SNMP-yhteensopivaan muotoon. (Cisco Internetworking Technology Handbook: Simple Network Management Protocol 2008)

Verkonhallintajärjestelmän tehtävä on käynnistää sovelluksia, jotka valvovat ja hallitsevat hallittavia laitteita (Cisco Internetworking Technology Handbook: Simple Network Management Protocol 2008). Verkonhallintajärjestelmät ovat yksittäisille tai monesti useille palvelimille kasattuja järjestelmäkokonaisuuksia, jotka prosessoivat verkonhallinta-informaatiota. Järjestelmät ovat yleensä paljon muistia ja prosessoriaikaa käyttäviä kokonaisuuksia. Jotta verkonhallinta olisi ylipäätään mahdollista SNMP-protokollan avulla, yksi tai useampia verkonhallintajärjes-

telmiä täytyy sijaita hallittavassa verkossa (Cisco Internetworking Technology Handbook: Simple Network Management Protocol 2008).

3.3.4 SNMP:n toiminta

SNMP on OSI-mallin sovellustason protokolla. Sen tärkein tehtävä on välittää verkonhallintainformaatiota valvottavilta laitteilta valvontaa hoitaville palvelimille. Tämä kommunikaatio voi olla kahdenlaista (Stevens 2000: 359):

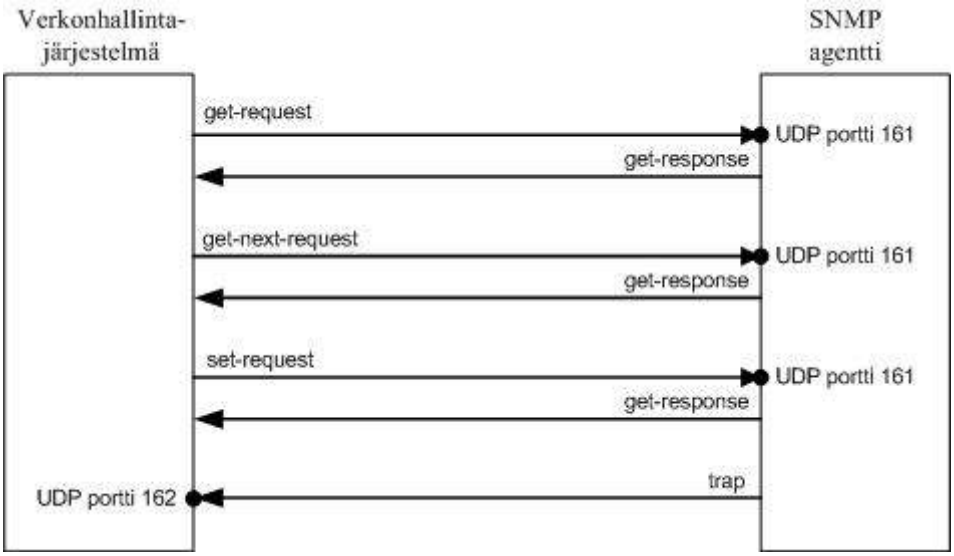
- Verkonhallintajärjestelmä pyytää agentilta tiettyä muuttujan arvoa (”Onko liitäntä X ylhäällä?”).
- Agentti kertoo verkonhallintajärjestelmälle, että jotain tärkeää on tapahtunut (”Liitäntä meni alas”).

Kaikki valvottavat laitteet, jotka on konfiguroitu käyttämään SNMP:tä, ylläpitävät hallintatietokantaa eli MIB:iä (Management Information Base). MIB sisältää tiedot, joita se käyttää omassa toiminnassaan, esim. liitäntä- ja reititystiedot. Laitteiden valvonta toimii siten, että verkonhallintajärjestelmä pyytää laitteelta MIB-kannasta tarvittavia tietoja (Ballew 1998: 209).

SNMP käyttää tiedonhakemiseen laitteelta viiden tyyppisiä viestejä (Case ym. 1990: 29):

1. Nouda yhden tai useamman muuttujan arvo: *get-request* -operaattori
2. Nouda seuraava muuttuja yhden tai useamman määritellyn muuttujan jälkeen: *get-next-request* -operaattori
3. Aseta yhden tai useamman muuttujan arvo: *set-request* -operaattori
4. Palauta yhden tai useamman muuttujan arvo: *get-response* -operaattori
5. Ilmoita verkonhallintajärjestelmälle, kun jotain tapahtuu agentissa: *trap* -operaattori

Verkonhallintajärjestelmä lähettää kolme ensimmäistä viestiä agentille. Agentti vastaa näihin get-response -operaattorilla. Jos laite on konfiguroitu lähettämään SNMP trap -viestejä, agentti lähettää trap-viestin verkonhallintajärjestelmälle, kun laitteella tapahtuu muutos, josta on määritelty trap lähetettäväksi. Kuvio 3.3.4 tekee yhteenvedon SNMP:n käyttämien operaattorien toiminnasta:



Kuvio 3.3.4: SNMP:n operaattorit ja porttinumerot (Stevens 2000: 361)

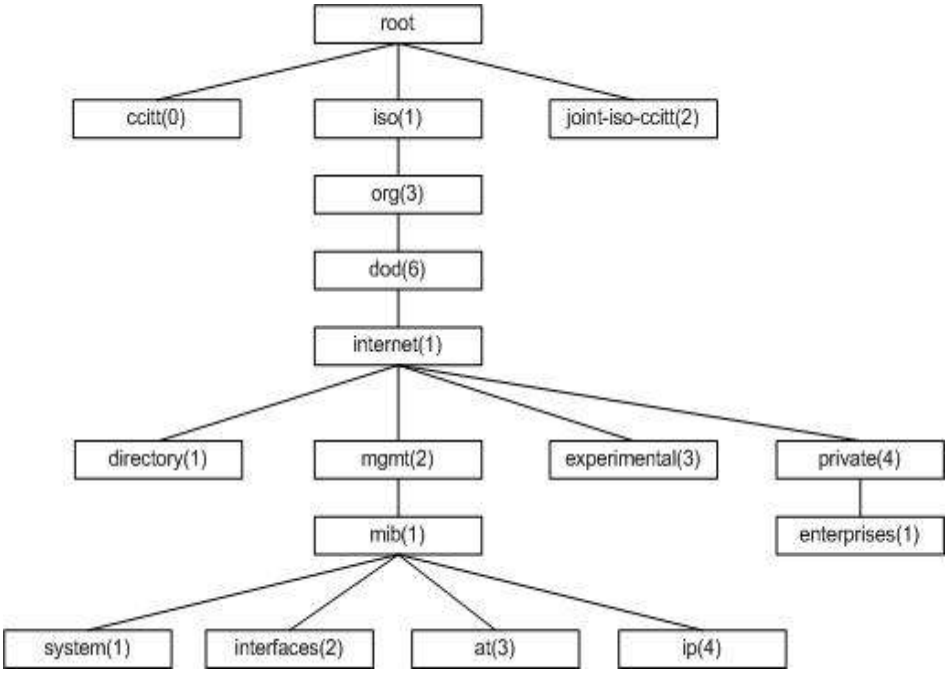
Verkonhallintajärjestelmä lähettää pyynnöt UDP porttiin 161. Agentti lähettää trap-viestit UPD porttiin 162.

3.3.5 Hallintatiedon rakenne (SMI)

SMI määrittää SNMP:n käyttämän hallintatiedon rakenteen. Toisin sanoen se kertoo, millaisia sääntöjä käyttäen MIB-muuttujia määritetään ja tunnistetaan (Comer 2002: 558). Liiallisen monimutkaisuuden välttämiseksi SMI määrittää vain muutamia MIB-muuttujatyyppejä sekä niiden määrittämis- ja nimeämissäännöt. Näitä ovat esim. *IpAddress* (4 oktetin merkkijono) ja *Counter* (kokonaisluku väliltä 0 – 2^{32}-1) (Stevens 2000: 363).

MIB-muuttujien nimien määrittämiseen käytetään ISO:n ja ITU:n hallitsemaa nimiavaruutta, joka tunnetaan nimellä ObjectIdentifier (OID). OID-nimiavaruus on hierarkinen. Sen pohjalta pystytään määrittämään kaikki tarvittavat objektien nimet, jotka ovat kaikki yksilöllisiä. Hierakisen rakenteen takia nimiavaruuden eri osien hallinta voidaan delegoida (Comer 2002: 559.) Näin myös eri laitevalmistajat saavat oman ”oksansa” nimiavaruudesta ja voivat luoda omat laitevalmistajaspesifiset MIB:nsa.

OID-nimiavaruuden hierarkinen juuri on nimetön. Sitä kutsutaan kuitenkin yleensä nimellä juuri (root) (Stevens 2000: 364). Kokonaisuudessaan rakenne on kuvion 3.3.5 mukainen:



Kuvio 3.3.5: OID nimiavaruuden rakenne (Stevens 2000: 365)

Edellä mainitut laitevalmistajakohtaiset MIB:it löytyvät enterprises-oksan alta. Nimet alkavat aina *iso.org.dod.internet.private.enterprises* ja vastaavasti OID on 1.3.6.1.4.1.

3.3.6 SNMP versio 2

Vuonna 1993 julkaistiin 11 uutta RFC-dokumenttia, jotka määrittävät SNMP version 2 (SNMPv2). SNMPv2 on kehittyneempi versio alkuperäisestä SNMP:stä. Uusi versio sisälsi myös useita parannuksia alkuperäiseen protokollaan. Tärkeimmät muutokset SNMPv1 ja SNMPv2 välillä ovat (Stevens 2000: 387 – 388):

- Uusi pakettityyppi *get-bulk-request*, joka mahdollistaa verkonhallintajärjestelmälle suurten tietomäärien tehokkaan hakemisen.
- Uusi pakettityyppi *inform-request*, jonka avulla yhden verkonhallintajärjestelmän on mahdollista lähettää tietoa toiselle.
- Kaksi uutta MIB:iä: SNMPv2 MIB ja SNMPv2-M2M MIB (Manager-to-Manager).
- SNMPv2 sisältää turvallisuuspäivityksiä alkuperäiseen protokollaan. SNMPv1-protokollassa community string-merkkijonot verkonhallintajärjestelmän ja agentin välillä ovat normaaliteksti salasanoja. SNMPv2-protokollan myötä on mahdollisuus käyttää autentikointia ja salausta.

SNMPv2 hallintatiedon rakenne, SMI, määriteltiin RFC 1902 – dokumentissa (Cisco Internetworking Technology Handbook: Simple

Network Management Protocol 2008). Siinä määritellään tiettyjä lisäyksiä ja parannuksia alkuperäisiin SMI-tietotyyppeihin, kuten bittimerkkijonot, verkko-osoitteet ja laskurit. Lisäksi MIB-puuhun määritellään kaksi uutta oksaa (Simoneau 1999: 229):

- Security (1.3.6.1.5)
- SNMPv2 (1.3.6.1.6)

3.3.7 SNMP versio 3

Myös kolmas versio, SNMPv3, on kehitetty SNMP-protokollasta, mutta se ei ole vielä saanut kovinkaan suurta suosiota. Erityisesti tähän vaikuttaa vähäinen määrä laitteita, joihin SNMPv3-tuki on implementoitu. SNMPv3 on taaksepäin yhteensopiva edellisten kahden version kanssa ja laajentaa protokollan toimintaa (Comer 2002: 572). Suurimmat parannukset uudessa versiossa koskevat tietoturvaa ja hallintaa. SNMPv3 suojauskäytännöt on suunniteltu joustaviksi ja helposti käyttöönotettaviksi, jolloin verkonhallintajärjestelmän ja hallittavien laitteiden välinen kommunikaatio on helpompi huomioida yrityksen tietoturvasuunnittelussa (Comer 2002: 572).

Muita SNMPv3:n mukanaan tuomia uusia ominaisuuksia ovat mm. sanomien todennus ja verkonhallintajärjestelmän auktorisointi (Comer 2002: 572). Sanomien todennuksella varmistetaan, että komennot tulevat verkonhallintajärjestelmältä, jolla on oikeus komentoja antaa ja tietoa hakea. Auktorisoinnilla taas voidaan määrittää, mitä tietoja jokin tietty verkonhallintajärjestelmä on oikeutettu pyytämään ja saamaan. SNMPv3 pystyy myös tarkistamaan viestien eheyden. Tällä varmistetaan, että viestejä ei ole muokattu matkan varrella (SNMPv3 2008).

4 IBM Tivoli Netcool

Yksi pisimmälle kehitetyistä kaupallisista verkonvalvontajärjestelmistä on IBM Tivoli Netcool. Järjestelmällä on takanaan usean vuoden menestyksekäs historia, josta kertoo jo sekin, että Netcool-järjestelmän keskeisimmän yksikön, OMNIbus:in, viimeisin versio on 7.2. Järjestelmä on käytössä mm. monilla suurilla kansainvälisillä Internet palveluntarjoajilla ja huhujen mukaan myös NASA:lla.

Tuote tunnettiin pitkään pelkäästään Netcool-nimellä. Tällöin tuotteen omisti ja sitä kehitti Micromuse-niminen Yhdysvaltalainen yritys. Vuoden 2005 aikana IBM kuitenkin päätti, että yritys haluaa vahvistaa Tivoli-tuoteperheettään ja vuoden lopulla IBM osti Micromusen 865 miljoonalla dollarilla ja muutti Micromusen yhdeksi Tivoli-bisnesyksikökseen. Samalla tuoteperheen nimeksi muutettiin IBM Tivoli Netcool.

IBM Tivoli Netcool -järjestelmä koostuu useista tuotteista, jotka hoitavat verkonvalvonnan eri osa-alueita. IBM Tivoli Netcool -järjestelmä yhdistää eri verkonvalvonnan osat yhden tuoteperheen alle. Tuoteperheeseen kuuluu seuraavat tuotteet (IBM Tivoli Network Management Documentation 2008):

Verkon saatavuudenhallinta

- IBM Tivoli Netcool/OMNIbus
- IBM Tivoli Netcool/Webtop
- IBM Tivoli Network Manager IP Edition
- IBM Tivoli Network Manager Transmission Edition
- IBM Tivoli Network Manager Entry Edition
- Netcool/Precision IP
- Netcool/Precision TN
- Netcool/Precision IP Asset Discovery
- Netcool/Precision IP Cramer Intergration Module
- Netcool/Portal
- Netcool/Reporter
- Netcool/Visionary
- Netcool for Wireless User Quality

Verkon suorituskyvynhallinta

- IBM Tivoli Netcool/Proviso
- IBM Tivoli Netcool Carrier VoIP Manager
- IBM Tivoli Netcool Enterprise VoIP Manager
- IBM Tivoli Netcool IP Multimedia Subsystem Manager
- IBM Tivoli Netcool Performance Management for Wireless Products

Palvelunlaadunhallinta

- *IBM Tivoli Netcool Service Quality Management Products*

Infrastruktuurin dokumentointi

- Netcool GUI Foundation
- IBM Tivoli Netcool/Security Manager

Saatavuuden- ja palveluidenhallinta

- *Netcool/Impact*
- *Composite Application Manager for Internet Service Monitoring*
- *Netcool/RAD*
- Service Monitors Reporter/Netcool

Seuraavaksi tutustumme lähemmin tärkeimpiin IBM Tivoli Netcool -tuoteperheen komponentteihin.

4.1 IBM Tivoli Netcool/OMNIbus

IBM Tivoli Netcool -järjestelmän ydin on IBM Tivoli Netcool/OMNIbus. Sen tehtävä on kerätä tapahtumainformaatiota useista verkon eri tietolähteistä ja esittää informaatio kootusti verkon operaattoreille ja pääkäyttäjille. Järjestelmä seuraa ja kerää tapahtumadataa korkean suorituskyvyn omaavaan muistissa pyörivään tietokantaan ja esittää merkityksellisen tiedon käyttäjille filtterien ja näkymien avulla. (IBM Tivoli Network Management Documentation 2008.) IBM Tivoli Netcool/OMNIbus koostuu useista alikomponenteista, joita ovat:

- ObjectServer
- Probet
- Yhdyskäytävät
- Desktop-työkalut
- Hallintatyökalut

ObjectServer on muistissa pyörivä tietokantapalvelin, joka on koko järjestelmän ydin. Kaikki verkosta kerätty tapahtumatieto lähetetään ObjectServerille, jossa tieto tallennetaan tietokantataulukoihin. Tämän jälkeen tieto esitetään tapahtumalistoissa.

Probet ovat sovelluksia, jotka hoitavat tiedon keruuta. Niitä ovat mm. SNMP- ja ping-probet. Merkittävin probeista on SNMPd-probe, joka nimensä mukaisesti kerää SNMP-protokollan avulla tietoa verkon tapahtumista. Tiedonkeruun lisäksi probe parsii tiedon ObjectServerin ymmärtämään muotoon. Toisin sanoen probe muuttaa SNMP-protokollan avulla kerätyn tiedon muotoon, jota ObjectServer osaa lukea.

Yhdyskäytävien tehtävä on välittää tietoa ObjectServerin ja tiettyjen kolmannen osapuolen sovellusten välillä. Sovelluksiin kuuluu mm. tietokantoja, helpdesk- ja tiketöintijärjestelmiä. Yhdyskäytävää tarvitaan myös ObjectServerien kahdennukseen. Yhdyskäytävät mahdollistavat useamman ObjectServerin välisen kommunikoinnin.

Desktop on kokoelma graafisia työkaluja, joiden avulla voidaan tarkastella ja hallita tapahtumia sekä määrittää, millälailla tapahtumainformaatio esitetään. Desktop-työkaluja on mahdollista käyttää sekä Windows- että Unix-käyttöjärjestelmissä.

Hallintatyökalut mahdollistavat ObjectServerin hallinnan. Ne ovat kokoelma graafisia käyttöliittymiä, joiden avulla pystymään konfiguroimaan ja hallitsemaan ObjectServereitä. Työkaluihin kuuluu Netcool/OMNIbus Administrator, interaktiivinen SQL-käyttöliittymä, prosessihallinta sekä import/export-sovellus.

4.2 IBM Tivoli Netcool Webtop

Netcool/Webtop on selainpohjainen sovellus, joka käsittelee verkkotapahtumia yhdestä tai useammasta tietolähteestä ja esittää tapahtumatiedon käyttäjille monenlaisissa graafisissa muodoissa (IBM Tivoli Netcool/Webtop version 2.1 Administration Guide 2008: 1). Tärkein käyttökohde on verkonvalvontakeskukset (Network Operations Center, NOC), joissa NOC-työntekijät pystyvät Webtopin kautta seuraamaan keskitetysti tapahtumia kaikista valvottavista verkoista.

Webtop voi esittää tapahtumadataa tapahtumalistojen, taulukoiden ja IBM Tivoli Network Managerin avulla luotujen karttojen sekä kuvien muodossa. Tapahtumalistojen, taulukoiden ja karttojen katseluun käytetään selainyhteyttä palvelimelle.

Webtop käyttää palvelin/asiakas-arkkitehtuuria ja pyörii IBM Tivoli Netcool GUI Foundation -palvelimessa. Netcool GUI Foundation on palvelinsovellus, joka tarjoaa graafisia käyttöliittymiä eri Netcool-tuotteille ja mahdollistaa näiden käyttöliittymien käytön yhdellä kirjautumisella (IBM Tivoli Netcool/Webtop version 2.1 Administration Guide 2008: 2).

Webtop kommunikoi suoraan ObjectServerin kanssa ja esittää ObjectServerin tietokannassa olevan tapahtumadatan visuaalisessa muodossa. Webtop ja ObjectServer käyttävät kommunikointiinsa IDUC (Insert, Delete, Update, Control)-protokollaa. Protokollan avulla ObjectServer kehottaa Webtop-palvelinta hakemaan päivitetyn tapahtumatiedon ja päivittämään tapahtumalistat, kun muutoksia tapahtuu.

4.3 IBM Tivoli Network Manager

IBM Tivoli Network Manager tuo Netcool-järjestelmään verkonmallintamis- ja juurisyyanalyysiominaisuudet. Tuote sai viimeisimmän 3.7-version myötä uuden IBM Tivoli Network Manager -nimen, sillä 3.6-versio tunnettiin vielä nimellä Netcool/Precision.

Network Managerista on tarjolla kolme eri versiota erilaisiin verkkotyyppeihin (IBM Tivoli Network Management Documentation 2008):

- IBM Tivoli Network Manager IP Edition soveltuu käytettäväksi normaaleissa IP-verkoissa.
- IBM Tivoli Network Manager Transmission Edition on suunniteltu palveluntarjoajia silmällä pitäen siirtoverkkojen valvontaan (toimivat OSI-mallin 1. ja 2. kerroksella).
- IBM Tivoli Network Manager Entry Edition out-of-box-ratkaisu, joka asentaa automaattisesti kaikki tarvittavat komponentit ja tuotteet Network Managerin peruskäyttöä varten.

Network Manager tuo järjestelmän käyttäjälle seuraavat ominaisuudet (IBM Tivoli Network Manager IP Edition version 3.7 Administration Guide 2008: 1):

- Verkkojen ja verkkolaitteiden haku
- Verkkolaitteiden monitorointi
- Verkkotopologian visualisointi
- Juurisyytapahtumien tunnistaminen

Järjestelmän ylläpitäjä voi määritellä ja ajaa hakuja luodakseen verkkotopologiakuvia. Network Managerin löytämä topologia tallennetaan topologiatietokantaan (IBM Tivoli Network Manager IP Edition version 3.7 Administration Guide 2008: 3). Ylläpitäjä voi myös visualisoida topologiaa karttojen ja muiden kuvien avulla. Tapahtumalistojen visualisoinnin avulla ylläpitäjä voi muokata erilaisia näkymiä esim. hälytysten kriittisyyden tai asiakkaan pohjalta.

Network Manageria voidaan pitää älykkäänä probena, sillä se pystyy myös hoitamaan SNMP- ja ping-valvontaa. Järjestelmä valvoo, ovatko laitteet ja liitännät ylhäällä sekä toimiiko laitteet asetettujen parametrien mukaisesti. Jos ongelmia ilmenee, järjestelmä luo asiasta hälytyksen tapahtumalistaan.

Juurisyyn tunnistaminen on prosessi, jossa selvitetään perimmäinen syy, miksi yksi tai useampia laitteita hälyttää (IBM Tivoli Network Manager IP Edition version 3.7 Administration Guide 2008: 3). Koska useimpia laitteita ja yhteyksiä ei kahdenneta, vika yhdessä laitteessa aiheuttaa helposti yhteyden katkeamisen myös muihin laitteisiin. Kun näin käy, kaikilta laitteilta tulee hälytykset. Juurisyyanalysointi korreloi tapahtumia

verkkotopologiaan ja pystyy sen perusteella päättelemään, mitkä laitteet ovat välillisesti tavoittamattomissa verkkovian vuoksi (IBM Tivoli Network Manager IP Edition version 3.7 Administration Guide 2008: 3).

4.4 IBM Tivoli Netcool/Proviso

Provison avulla järjestelmän ylläpitäjä pystyy luomaan monenlaisia raportteja verkon toiminnasta. Erityisen hyödyllinen Proviso on verkonvalvonta- ja verkonhallintapalveluita tarjoaville yrityksille. Provison avulla yritys pystyy luomaan kattavan kuvan asiakkailleen heidän verkon toiminnasta ja tapahtumista, ja tarjoamaan huomattavaa lisäarvoa asiakkailleen. Luotujen raporttien pohjalta on helppoa parantaa palvelunlaatua ja pienentää operatiivisia kustannuksia.

Proviso koostuu useammasta eri komponentista, jotka usein sijoitetaan omille palvelimilleen. Järjestelmän alikomponenttejä ovat:

- Proviso DataMart
- Proviso DataLoad
- Proviso DataChannel
- Proviso DataView

Proviso DataMart on kokoelma järjestelmän hallintaan tarvittavia käyttöliittymiä. Käyttöliittymät ovat graafisia ja niiden avulla järjestelmän ylläpitäjä määrittää järjestelmän toimintaperiaatteet ja tarkistaa sekä ratkoo ongelmia toiminnassa.

Proviso kerää itse dataa raporttien pohjaksi. Tiedonkeruuta Provisossa hoitaa DataLoad. Se hoitaa hajautettua tiedonkeruuta niin SNMP kuin muistakin lähteistä. DataLoad kerää tiedot erilliseen keskitettyyn tietokantaan.

Proviso DataChannel toimii tiedon välittäjänä DataLoadin ja DataViewin välillä. DataChannel kokoaa DataLoadin keräämän tiedon ja tarjoaa sen DataViewin raportointiominaisuuksien käyttöön. DataChannelin toinen tehtävä on tehdä tosiaikaisia laskelmia ja havaita, jos asetut kynnysarvot ylittyvät.

DataView on sovelluspalvelin, jonka avulla luodaan selainpohjaisia verkkoraportteja. Monenlaisten palvelunlaaturaporttien luonti onnistuu kätevästi DataViewin avulla. Kun raportit ovat valmiita, ne voidaan julkaista selainpohjaisen Portalin kautta asiakkaiden saataville.

4.5 IBM Tivoli Netcool/Impact

IBM Tivoli Netcool/Impact on Netcool-järjestelmän korrelaatiomoottori. Impactin avulla ristiintaulukoidaan ObjectServerille tulevia tapahtumia ja jossain toisessa tietolähteessä olevaa tietoa. Tätä kutsutaan myös tapahtumien rikastamiseksi. Toisin sanoen Impactin avulla lisätään tietoja valvontaa suorittavilta komponenteilta, kuten probeilta, tuleviin tapahtumiin. Tapahtumiin voidaan lisätä esim. laitteen sijainti- ja kontaktitiedot ennen kuin ne esitetään tapahtumalistoilla. Yleensä nämä tiedot on tallennettu laite- ja konfiguraatiotietokantaan eli CMDB:hen (Configuration Management Database).

Impactia voidaan myös käyttää OMNIbusin ja kolmannen osapuolen järjestelmien integrointiin. Se soveltuu mm. useiden tietokantojen, viestintäjärjestelmien ja verkkoinventaarisovellusten yhdistämiseen OMNIbusin kanssa.

5 Netcool-dokumentaation luonti Corenetille

Tämän opinnäytetyön tuotteena loin dokumentaation Corenet Oy:lle heidän uudesta IBM Tivoli Netcool -verkonvalvontajärjestelmästään. Tarve dokumentaatiolle syntyi jo alkuvuodesta 2007, jolloin Corenetin yritysjohto päätti lähteä tekemään suuren panostuksen ja hankki uuden verkonvalvontajärjestelmän. Oman verkon ja sen hetkisten asiakkaiden laitteet haluttiin saada monipuolisemman järjestelmän valvontaan ja hallintaan. Toisena tavoitteena Corenetilla oli lähteä tarjoamaan verkonvalvonta- ja verkonhallintapalveluita myös uusille asiakasyrityksille.

5.1 Aiheen saaminen ja alkurajaus

Kun Corenet päätti panostaa uuteen verkonvalvontajärjestelmään, työskentelin itse Corenetin verkonvalvontatiimissä. Idea opinnäytetyön aiheeseen tuli silloiselta esimieheltäni Dan Torckellilta. Tiedustelin Danilta toukokuussa 2007, olisiko Corenetilla tarjota aihetta opinnäytetyölleni ja tarve dokumentoida Valluksi ristitty IBM Tivoli Netcool -järjestelmä oli hänen ensimmäinen ideansa. Tästä asiat lähtivät rullaamaan ja ideoimme jo toukokuun aikana dokumentaation sisältöä yhdessä Danin ja Vallu-projektinvetäjän Ari Löytynojan kanssa.

Päätimme heti alkuun, että tuotoksen tulisi dokumentoida, millainen Corenetin Netcool-järjestelmä on. Toisin sanoen, se tulisi pitämään sisällään paljon yleistä Netcool-tietämystä ja siihen dokumentoitaisiin myös kaikki oletusasetuksiin tehtävät muutokset. Yksi idea oli myös luoda ohjeita verkonvalvontatiimille, kuinka he pystyisivät järjestelmän vikatilanteissa, varsinkin yö- ja viikonloppuaikaan, tarkistamaan järjestelmän tilan ja mahdollisesti palauttamaan sen takaisin toimintakuntoon. Tästä ideasta kuitenkin luovuttiin myöhemmin. IBM Tivoli Netcool -järjestelmä on hyvin monimutkainen ja useat korjaustoimenpiteet vaativat järjestelmään admin-tason oikeuksia. Niitä ei kuitenkaan haluttu antaa liian usean käyttöön. Tästä syystä dokumentaatiosta jätettiin pois kyseiset ohjeet.

Suurin aiheen rajaus tehtiin heti alkuun. Silloin päätimme, että opinnäytetyönäni tekemä tuotos tulisi koskemaan ainoastaan järjestelmän 1.0-versiota. Yrityksen johto ja projektissa mukana olevat henkilöt olivat jo aiemmin päättäneet, mitä ominaisuuksia ja toiminnallisuuksia järjestelmässä tulisi olla versiossa 1.0. Tehtäväkseni tuli dokumentoida kyseiset ominaisuudet ja toiminnallisuudet.

5.2 Netcool-tuntemus ja -koulutus

Alkutietämykseni IBM Tivoli Netcool -järjestelmän toiminnasta oli melko heikko. Edellisellä työnantajallani, jolle tässä vaiheessa edelleen teimme ulkoistettuna verkonvalvontaa, oli käytössään Netcool-järjestelmä. Verkonvalvontatiimin jäsenenä pääsin näkemään valvontanäkymän ja SSH-yhteydellä asiakasverkkojen probeille sekä tiesin, että probet hoitavat varsinaista laitevalvontaa. Muuta tietoa Netcool-järjestelmästä minulla ei vielä tässä vaiheessa ollut.

Corenetin johto päätti heti alusta lähtien hankkia kunnollisen koulutuksen Vallu-järjestelmän ylläpitäjille Mikko Tepposelle ja Markku Riikoselle. Suurin osa koulutuksista järjestettiin Iso-Britanniassa. Myös Helsingissä järjestettiin kuitenkin osa koulutuksista ja koska tehtävänäni oli dokumentoida Corenetin järjestelmä, pääsin mukaan Helsingissä IBM:n tiloissa järjestettyihin kolmeen koulutukseen. Ne järjestettiin syyskuussa 2007 seuraavasti:

- 3. – 5.9. IBM Tivoli Netcool/OMNIbus User and Administration and Configuration
- 6. – 7.9. IBM Tivoli Netcool/Webtop Fundamentals
- 10.9. – 14.9. IBM Tivoli Netcool/Precision IP Fundamentals

Kouluttajana molempien viikkojen ajan toimi David Cunningham Iso-Britanniasta.

Koulutukset olivat erittäin tarpeellisia opinnäytetyöni kannalta. Sain koulutusten aikana erittäin hyvät perustiedot Netcool-tuotteista. Tietenkin täytyy huomioida, että Impact- ja Proviso-tuotteita koulutus ei käsittänyt. Siitäkin huolimatta koulutusten anti opinnäytetyön kannalta oli korvaamaton. Ilman näitä kolmea koulutusta dokumentaation tekeminen olisi ollut selvästi vaikeampi urakka.

5.3 Asiat lähtevät rullaamaan

Opinnäytetyöni lähti kunnolla käyntiin vasta syyskuun koulutusten jälkeen. Myös Netcool-projektinvetäjä Ari Löytynoja oli mukana kursseilla ja meidän molempien kuva IBM Tivoli Netcool –järjestelmän toiminnasta kehittyi huomattavasti koulutusten seurauksena.

Koulutusten jälkeen pidimme Arin kanssa muutamia palaverejä, joissa mietimme tarkemmin, mitä kaikkea dokumentaation tulisi sisältää. Myös järjestelmän ylläpitäjistä Mikko oli väliin mukana ja häneltä tuli myös paljon hyviä ideoita dokumentaatioon. Käydyissä keskusteluissa päätimme mm. liittää dokumentaatioon tiedot, kuinka Corenetin vanha ver-

konvalvontajärjestelmä ja ongelmatikettijärjestelmä yhdistetään Netcool-järjestelmään. Molemmat näistä tosin jäivät dokumentaation 1.0-versiosta pois suurelta osin. Joitain tietoja kirjattiin ylös, mutta integroinnissa ilmenneiden ongelmien takia kaikkea tietoa ei saatu kirjattua ylös.

Kun olimme saaneet Arin ja Mikon kanssa rajattua, mitä kaikkea dokumentaation tulisi sisältää, alkoi itse dokumentaation kirjoittaminen. Lähteinä dokumentaatioon käytin pääasiassa Netcool-koulutuksista saatua materiaalia sekä IBM:n dokumentaatiokantaa, joka löytyy osoitteesta:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?toc=/com.ibm.netcool_OMNIbus.doc/toc.xml

Yksi tärkeä tiedonlähde olivat Ari ja Mikko. Heiltä sain erityisesti tietoa Corenetin järjestelmän asetuksista ja arkkitehtuurista.

Mitään selkeää aikataulua dokumentaation valmistumiselle en missään vaiheessa saanut Corenetin puolelta. Vuorotyön takia minulla oli tässä vaiheessa arkisinkin vapaapäiviä, jolloin pystyin dokumentaatiota kirjoittamaan. Kirjoittaminen ei kuitenkaan edennyt ihan tahdilla, jota olin ajatellut. Syynä tähän olivat varsinkin vuorotyö ja 12-tuntiset vuorot, jolloin vapaapäivinä ei useinkaan ollut energiaa ja motivaatiota työn tekemiseen.

5.4 Työn loppuvaiheet

Pikkuhiljaa työ kuitenkin eteni ja dokumentaatiossa alkoi piirtyä kuva Corenetin Vallu-järjestelmästä. Tammikuussa 2008 työ oli jo melko valmiissa tilassa. Lähetin aina silloin tällöin Arille ja Mikolle viimeisimmän version luettavaksi ja tätä kautta sain hyvää palautetta dokumentaation vioista ja puutteista.

Viimeinen sysäys dokumentaation valmiiksi saattamiselle syntyi, kun sain uuden työpaikan Stonesoft Oyj:stä. Dokumentaatio oli onneksi tässä vaiheessa jo niin hyvässä kunnossa, että pystyin sopimaan Arin kanssa, että toimitan hänelle 1.0-version vielä ennen lähtöäni Corenetiltä. Sain työn valmiiksi toiseksi viimeisenä työpäivänäni Corenetillä.

Corenetin varsinaisen version lisäksi jouduin tämän jälkeen luomaan dokumentaatiosta vielä julkisen version, joka on tämän työn liitteenä. Kun aloittelin työn tekemistä kesällä 2007, oli heti selvää, että dokumentaatiosta täytyy luoda kaksi versiota. Tietoturva- ja yrityssalaisuussyistä julkiseen versioon ei saanut laittaa laitteiden IP-osoitteita, DNS-nimiä ja sijaintipaikkakuntia. Myös osa käytetyistä porttinumeroista sekä kaikki hallintakäyttöliittymien URL-osoitteet piti poistaa julkisesta versiosta.

Palautin julkisen version Arille vielä viikonlopun aikana, mutta sitä jouduin vielä muuttamaan useampaan kertaan jälkeenpäin. Corenetin uudella turvallisuuspäälliköllä oli tiukka linja julkista versiota kohtaan. Siitä piti vielä poistaa kaikki tiedot, jotka mitenkään viittasivat Corenetin järjestelmän arkkitehtuuriin ja konfigurointiin. Sivumäärässä tämä tarkoitti tekstin poistamista ja muokkaamista n. 10 sivun verran. Tämän työn liitteenä olevasta julkisesta versiosta tuli näin yleinen dokumentaatio IBM Tivoli Netcool -järjestelmän toiminnasta. Kaikki siinä olevat tiedot on löydettävissä IBM-dokumenteista.

6 Johtopäätökset

IBM Tivoli Netcool -järjestelmässä on paljon hyviä ominaisuuksia. Järjestelmä on hyvin monipuolinen kokonaisuus, joka tarjoaa todella paljon mahdollisuuksia käyttäjälleen. Verkkolaitteiden valvonta ja hallinta onnistuu keskitetysti. Lisäksi TCP/IP-verkkojen valvonnan ohella myös siirtoverkkolaitteiden valvontaan on tarjolla oma tuotteensa, IBM Tivoli Network Manager Transmission Edition. Järjestelmä pystyy itse etsimään verkkolaitteet ja piirtämään niistä topologiakuvat mm. OSI-mallin 2- ja 3-kerroksella sekä MPLS-verkkojen perusteella. Lisäksi Provison avulla pystytään luomaan hyvin monenlaisia raportteja.

Kun IBM Tivoli Netcool -järjestelmä on saatu konfiguroitua toimintakuntoon, on se vakaa ja hyvin toimiva järjestelmä. Tuotteella on takanaan jo usean vuoden historia ja se on käytössä useissa suurissa tietoliikenneyrityksissä ympäri maailman. Tämän lisäksi järjestelmää käyttävät myös monet muiden alojen suuryritykset, sillä IBM Tivoli Netcool -järjestelmä pystyy valvomaan tietoverkkolaitteiden lisäksi myös muitakin laitteita ja järjestelmiä. Käytännössä järjestelmän avulla voidaan valvoa kaikkea, mistä pystytään saamaan sähköinen impulssi, joka lisäksi pystytään muuntamaan muotoon, jossa IBM Tivoli Netcool -järjestelmä ymmärtää impulssin tapahtumana.

Kolmas selkeästi positiivinen asia on hyvä tuotetuki valmistajan puolelta. Tämä on varsinkin totta, jos vertailuun otetaan pariksi jokin kilpailevista ilmaisista järjestelmistä. IBM Information Centeristä löytyy joka tuotteelle useita käyttöönotto-, konfigurointi- ja ylläpito-oppaita. Ongelmatilanteissa asiakkaille tarjotaan myös apua niin puhelimitse kuin sähköpostitse nopealla vasteajalla.

Monipuolisessa ja toimivassa IBM Tivoli Netcool -järjestelmässä on myös omat miinuspuolensa. Suurin este järjestelmän todella laajamittaiselle ja yleisemmälle käytölle on laitteen mukana tuleva kallis hintalappu. IBM Tivoli Netcool -järjestelmää ei saa alle miljoonan euron hintaan. Kaikilla järjestelmään kuuluvilla tuotteilla on omat hintansa ja lisenssinsä. Lisäominaisuuksiakin on paljon tarjolla, mutta jokainen lisenssi maksaa aina useiden tuhansien eurojen verran. Peruslisenssin avulla saa mm. valvoa tiettyä määrää verkkolaitteita. Jos kyseinen raja ylitetään, täytyy yrityksen ostaa uusi lisenssi.

Toinen IBM Tivoli Netcool -järjestelmän heikkous on, että järjestelmää on monimutkaista konfiguroida ja ylläpitää. Graafisia käyttöliittymiä on todella monta, joten järjestelmän kohdalla ei todellakaan voida puhua keskitetystä hallintakonsolista. Osa muutoksista täytyy myös tehdä suoraan konfiguraatiotiedostoihin. Toisin sanoen tiedostoja joudutaan muokkaamaan tekstieditorin avulla, mikä on työlästä ja hidasta. Käymieni tuotekoulutusten aikana sain myös huomata, että tiettyjen asetus-

ten luominen vaatii sekä graafisen käyttöliittymän käyttöä että konfiguraatiotiedostojen suoraa muokkaamista.

Viimeisenä heikkoutena mainitsisin sen, että järjestelmä koostuu useamman eri valmistajan tuotteesta, vaikka nykyisin kaikki tuotteet kuuluvat IBM:n Tivoli-tuoteperheeseen. IBM on hankkinut yritysostoilla oikeudet järjestelmään kuuluviin tuotteisiin ja esim. järjestelmän ydin, joka tunnettiin aiemmin pelkästään Netcool-nimellä, oli pitkään Micromuse-nimisen yrityksen tuote. Koska järjestelmän eri tuotteet ovat alun perin eri yritysten kehittämiä, eivät käyttöliittymät luonnollisestikaan ole yhdenmukaisia ja tästä seuraa edellisessä kappaleessakin mainittu ongelma eli järjestelmän monimutkainen ylläpito.

7 Yhteenveto

Tietoverkoista on tullut erittäin tärkeä osa liikemaailmaa. Samalla verkkolaitteiden toimivuuden ja toimimattomuuden nopeasta havaitsemisesta on tullut yhtälailla tärkeää. Laitteiden valvontaa hoitavista protokollista tärkein on SNMP. Protokollan keräämien tietojen hyödyntämiseen tarvitaan oikeanlaiset ohjelmistot, joista yksi monipuolisimpia, mahdollisesti kaikkein monipuolisin, on IBM Tivoli Netcool.

Tämä opinnäytetyö lähti liikkeelle tilanteessa, jossa toimeksiantaja Corenet hankki IBM Tivoli Netcool -järjestelmän ja järjestelmän toiminta tuli dokumentoida. Opinnäytetyönä tekemäni IBM Tivoli Netcool -dokumentaatio oli osa isompaa projektia. Teimme yrityksen edustajien kanssa heti alkuun suuren rajauksen, sen perusteella opinnäytetyöni tulisi käsittämään dokumentaation järjestelmän 1.0-versiosta. Versio 1.0 aikataulutettiin valmistumaan alkuvuodesta 2008.

Sain dokumentaation valmiiksi helmikuussa 2008 ja pienten muutosten jälkeen toimeksiantaja hyväksyi sen. Dokumentaatiosta tuli kattava ja se kuvaa hyvin Corenetin hankkimien IBM Tivoli Netcool-tuotteiden toimintaa sekä Corenetin järjestelmän toimintaa ja arkkitehtuuria. Joitain suunniteltuja osia dokumentaatiosta jäi vielä uupumaan osittain järjestelmän kanssa kohdattujen ongelmien ja osittain lopussa tulleen kiireen takia. Kuten mainittua, dokumentaatiosta kuitenkin valmistui hyväksytty 1.0-versio. Yrityksen edustajat pitivät sitä hyvänä pohjana, kun he lähtevät kehittämään verkonvalvontajärjestelmäänsä eteenpäin seuraaviin versioihin. Erityisesti kiitosta tuli siitä, että dokumentaatio kuvaa hyvin Netcool-tuotteiden sisäistä toimintaa. Siitä tulee olemaan hyötyä varsinkin, jos Corenet palkkaa uusia ylläpitäjiä Netcool-järjestelmää hallinnoimaan.

Tietoa dokumentaatioon keräsin useammalla eri tavalla. Tärkeimpiä lähteitä olivat IBM:n WWW-dokumentaatiokannasta löytyvät tuotedokumentaatiot, kuten Administration Guide -oppaat. Niistä sain kerättyä kattavasti tietoa Netcool-tuotteiden toiminnasta ja hallinnasta.

Toinen tärkeä tiedonlähde olivat koko projektin aikana käymäni keskustelut projektinvetäjä Ari Löytynojan ja järjestelmän ylläpitäjä Mikko Tepposen kanssa. Sain heiltä suurimmaksi osaksi kaikki Corenetin Netcool-järjestelmää koskevat tiedot, jotka tulivat dokumentaatioon. Ari ja Mikko auttoivat kiitettävästi aina tarvittaessa, kun kiireiltään ehtivät. Projektin alussa, varsinkin loka- ja marraskuussa, Mikolla oli usein hyvin kiire. Mikko vastasi suurelta osin järjestelmän konfiguroinnista, joten tässä vaiheessa hänellä riitti tekemistä, koska järjestelmä piti saada toimintakuntoon mahdollisimman nopeasti.

Varsinkin dokumentaation kirjoittamisen aloitusvaiheessa tärkeä tiedonlähde olivat IBM Tivoli Netcool -koulutukset syyskuussa 2007. Ennen

koulutuksia tietämykseni IBM Tivoli Netcool -järjestelmästä oli melko vähäinen. Koulutuksista sain hyvän pohjan lähteä kirjoittamaan dokumentaatiota.

Dokumentaatiosta jouduin luomaan kaksi erillistä versiota. Corenetille toimitettu virallinen versio sisältää paljon salaiseksi luokiteltavaa tietoa, joten jouduin tekemään erillisen julkisen version dokumentaatiosta. Kyseinen versio on tämän työn liitteenä. Tämä julkinen versio voi hyvin kiinnostaa myös muita alalla toimivia yrityksiä. Varsinkin sellaisia yrityksiä, jotka ovat hankkimassa IBM Tivoli Netcool -tuotteita. Dokumentaation julkinen versio tekee kattavan yhteenvedon OMNIbus-, Impact-, Webtop-, Network Manager IP Edition - ja Proviso-tuotteiden toiminnasta, joten siitä on varmasti hyötyä myös muille yrityksille.

Itselleni työn tekemisestä on ollut paljonkin hyötyä. Omaan nykyisin melko kattavan tietämyksen IBM Tivoli Netcool -tuoteperheeseen. Siitä on varmasti hyötyä tulevaisuudessa, jos tulee tarvetta vaihtaa työpaikkaa. Lisäksi olen saanut työtä tehdessäni paljon tietoa erityisesti SNMP-protokollasta ja tietämykseni verkonvalvonnasta ja -hallinnasta on lisääntynyt selvästi.

Lähdeluettelo

Ballew, Scott M., 1998. IP-verkkojen hallinta Ciscon reitittimillä. Espoo: Suomen atk-kustannus

Case, J., Davin, J., Fedor, M. & Schoffstall, M., 1990. RFC 1098 - A Simple Network Management Procotocol (SNMP). [online][viitattu 21.5.2008]. http://tools.ietf.org/html/rfc1157

Cisco Internetworking Technology Handbook: Simple Network Management Protocol, 2008. [online][viitattu 13.4.2008]. http://www.cisco.com/en/US/docs/internetworking/technology/handbook/SNMP.html

Comer, Douglas E., 2002. TCP/IP. Helsinki: IT Press.

Corenet yritysesittely, 2007. [online][viitattu 24.2.2008]. http://www.corenet.fi/default.asp?docId=11891&rnd=8979932817853699

Hautaniemi, Mika 1994. TKK/Atk-keskuksen TCP/IP-verkon valvonta ja hallinta. [online][viitattu 26.9.2007]. http://www.netlab.tkk.fi/julkaisut/tyot/diplomityot/611/johdanto.html

IBM Tivoli Netcool/Webtop version 2.1 Administration Guide, 2008. [online][viitattu 20.4.2008]. http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_wt.doc/ag/Administration_Guide.pdf

IBM Tivoli Network Management Documentation, 2008. [online][viitattu 20.4.2008]. http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?toc=/com.ibm.netcool_OMNIbus.doc/toc.xml

IBM Tivoli Network Manager IP Edition version 3.7 Administration Guide, 2008. [online][viitattu 20.4.2008]. http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.itnetmantip.doc_3.7/7960/NMIP_administration.pdf

PathPing, 2008. [online][viitattu 21.5.2008]. http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/cnet/cnbd_trb_vxmb.mspx?mfr=true

Postel, J. 1981. RFC 792 – Internet Control Message Protocol. [online][viitattu 21.5.2008]. http://www.faqs.org/rfcs/rfc792.html

Simoneau, Paul, 1999. SNMP network management. New York: McGraw-Hill.

SNMPv3, 2008. [online][viitattu 21.4.2008]. http://www.cisco.com/en/US/docs/ios/12_0t/12_0t3/feature/guide/Snmp3.html#wp4363

Stevens, Richard W., 2000. TCP/IP Illustrated Volume 1: The Protocols. Reading (Massachusetts): Addison-Wesley

Understanding the Ping and Traceroute Commands, 2006. [online][viitattu 24.5.2008]. http://www.cisco.com/warp/public/63/ping_traceroute.html#traceroute

Liitteet

Liite 1: Netcool-järjestelmän kuvaus: Versio 1.0.1

Netcool-järjestelmän kuvaus

Versio 1.0.1

6.4.2008

SISÄLLYSLUETTELO

1 NETCOOL-DOKUMENTAATION TARKOITUS

Tämän dokumentin tarkoitus on kuvata Netcool-verkonvalvontajärjestelmän toimintaa ja sen käyttöön liittyviä asioita. Netcool-järjestelmä koostuu IBM Tivoli Netcool -ohjelmistoista. Tämä dokumentti keskittyy IBM Tivoli Netcool/OMNIbus-, IBM Tivoli Netcool/Webtop-, IBM Tivoli Network Manager IP Edition- ja IBM Tivoli Netcool/Proviso-ohjelmistojen toiminnan kuvaamiseen. Dokumentaatio tehtiin alunperin Corenet Oy:lle kuvaamaan Netcool-järjestelmää.

1.1 IBM:N TIVOLI NETCOOL-DOKUMENTAATIO

IBM:ltä löytyy runsain määrin tarkempaa tietoa mm. Netcool-tuotteiden asennuksesta ja konfiguroinnista. Dokumentit on luettavissa sekä suoraan selaimella HTML-muodossa että ladattavissa PDF-muodossa IBM:n dokumentaatiokannasta:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp

1.2 DOKUMENTISSA KÄYTETTYJÄ LYHENTEITÄ

Seuraavassa on lueteltuna tässä dokumentaatiossa yleisesti käytettyjä lyhenteitä:

Lyhenne	**Koko nimike**
CEF	Common Event Format
CMDB	Configuration Management Database
CVS	Concurrent Version System
DC	Distribution Chart
EMP	Electromagnetic Pulse
GLF	Generic Log File
GST	Group Summary Table
GUI	Graphical User Interface
MIB	Management Information Base
NCIM	Netcool Common Information Model
NCKL	Netcool Knowledge Library
NCSM	Netcool Security Manager
NGF	Netcool Graphical User Interface (GUI) Foundation
NOC	Network Operations Center
RST	Resource Summary Table
RTT	Resource Threshold Table
SNMP	Simple Network Management Protocol
WAAPI	Webtop Administration Application Programming Interface (API)

2 NETCOOL-JÄRJESTELMÄN TEKNISET TIEDOT

Tässä osioissa luetellaan eri Netcool-komponenttien komentoja sekä järjestelmän vakioasetuksilla käyttämät porttinumerot.

2.1 NETCOOL-KOMPONENTTIEN KÄYNNISTYS

Komponentti	**Komento**	**Käyttötarkoitus**
License Server	$NCHOME/license/bin/nc_start_license -background	Käynnistää License Serverin
ObjectServer	$OMNIHOME/bin/nco_objserv –name <*server_name*> &	Käynnistää ObjectServerin
Probe	$OMNIHOME/probes/nco_p_<*probe_name*>	Käynnistää proben
Security Manager	$NCHOME/security/bin/ncsm_server &	Käynnistää Security Managerin
Netcool GUI Foundation	$NCHOME/bin/ngf_server start	Käynnistää NGF-serverin
Impact	$NCHOME/bin/wasce start	Käynnistää Impactin
Gateway	$OMNIHOME/bin/nco_g_objserv_<*type*> -name <*server_name*>	Käynnistää gatewayn
Network Manager	$PRECISION_HOME/bin/ncp_ctrl -domain <domain_name>	Käynnistää Network Managerin
DataLoad	pvmdmgr start	Käynnistää DataLoadin SNMP-keräilijän (collector)
DataChannel	dccmd start <DC component>	Käynnistää DataChannel-komponentin, esim: dccmd start BCOL.2.2
DataMart	provisoinfod start	Käynnistää Proviso Info Daemonin DataMart-palvelimella
DataView / SilverStream	/etc/init.d/SilverStream start	Käynnistää DataViewin SilverStreamin
Oracle	lsnrctrl start	Käynnistää Oracle kuuntelijan (listener), jolloin Proviso voi ottaa yhteyttä kantaan

2.2 NETCOOL-KOMPONENTTIEN PYSÄYTYS

Komponentti	**Komento**	**Käyttötarkoitus**
License Server	$NCHOME/license/bin/nc_stop_license	Pysäyttää License Serverin
ObjectServer	*HUOM! Tämä tehdään root-käyttäjänä* $OMNIHOME/bin/nco_sql –server <*server_name*> "alter system shutdown;" "go" "exit"	Pysäyttää ObjectServerin
Security Manager	$NCHOME/security/bin/ncsm_shutdown	Pysäyttää Security Managerin
Netcool GUI Foundation	$NCHOME/bin/ngf_server stop	Pysäyttää NGF-palvelimen
Impact	$NCHOME/bin/wasce stop	Pysäyttää Impactin
Network Manager	No shutdown command -> pkill ncp	Pysäyttää Network Managerin
DataLoad	pvmdmgr stop	Pysäyttää DataLoadin SNMP-keräilijän (collector)
DataChannel	dccmd stop <DC component>	Pysäyttää DataChannel-komponentin, esim: dccmd stop all
DataMart	provisoinfod stop	Pysäyttää Proviso Info Daemonin DataMart-palvelimella
DataView / SilverStream	/etc/init.d/SilverStream stop	Pysäyttää DataViewin SilverStreamin
Oracle	lsnrctrl stop	Pysäyttää Oracle kuuntelijan

2.3 NETCOOL-KOMPONENTTIEN MUITA KOMENTOJA

Komponentti	**Komento**	**Käyttötarkoitus**
License Server	$NCLICENSE/bin/nc_print_license	Tulostaa lisenssit
ObjectServer	$OMNIHOME/bin/nco_dbinit -server <*obj_server_name*>	Luo uuden ObjectServerin
Security Manager	$NCSM_HOME/bin/ncsm_status	Tulostaa Security Managerin tilan
Netcool GUI Foundation	$NCHOME/bin/ngf_server status	Tulostaa NGF-palvelimen tilan
Network Manager	$PRECISION_HOME/bin/ncp_userconfig -domain <*domain_name*> -username <*username*> -password <*password*>	Luo Network Manager-käyttäjän

2.4 NETCOOL-KOMPONENTTIEN KÄYTTÄMIÄ PORTTINUMEROITA

Komponentti	**Portti**	**Käyttötarkoitus**
Network Manager	161	Portti, johon SNMP-kyselyt lähetetään
Probe, ITNM, Proviso	162	Portti, jota SNMP trap probe-sovellus (MTTrapd) kuuntelee
Proviso	514	Log forward port (syslog forwarding)
Impact	1099	WASCE:n käyttämä RMI nimeämisportti
Security Manager	1275	NCSM-palvelinportti
Impact	1389	WASCE:n Apache Directory Service -portti
Proviso / Oracle	1521	Database port for Oracle
Impact	1527	WASCE:n Apache Derby -portti
Proviso	3002	Bulk collector serviceport used for internal communication
Proviso	3113	DataMart ProvisoInfo RMI port
Proviso	3114	DataMart ProvisoInfo server port
Network Manger / MySQL	3306	MySQL- portti, TopoViz - MySQL -kommunikaatiolle
ObjectServer	4100	Portti, jota ObjectServer kuuntelee
Process Agent	4200	Portti, jota Process Agent kuuntelee
Impact	4201	WASCE:n Enterprise Java Beans -portti
Impact	4242	WASCE:n Java Authentication and Authorization Service -portti
Impact	5050	WASCE:n Common Object Services -portti
Security Manager	5600	NCSM-palvelimen tietokantaportti
Network Manager	7500	Vakioportti Tibco Rendezvouz-väylän käyttöön
Network Manager	7600	RVA-vakioportti, jota käyttää Discovery Configuration GUI
Impact	8009	WASCE:n Tomcat AJP -yhdistäjäportti
Security Manager	8077	NCSM-palvelimen HTTP-portti
Security Manager	8080	NCSM-rekisteröitymisportti
Impact, Proviso, Webtop	8080	HTTP-portti selainyhteydelle
NGF	8085	NGF-palvelimen kontrolliportti
NGF	8089	NGF-palvelimen tietokantaportti
Impact	8443	HTTPS-portti selainyhteydelle
Proviso	9001	CORBA channel manager port
Proviso	9002	CORBA application manager port
Proviso	9005	CORBA channel name service port
Proviso	25000	Log server port
License Server	27000	Portti, jota License Server kuuntelee
Network Manager	34535	Network Manager SNMP Helper
ObjectServer	> 1024	Dynaaminen IDUC -portti tapahtumalistojen päivitykseen
Proviso	45107	CORBA name server access port
Proviso	54890	Name Service Port used by SilverStream
Proviso	54891	Cache Manager RMI port
Impact	61616	WASCE:n Java Message System-portti

3 NETCOOL-JÄRJESTELMÄN TOIMINTA

Tässä osiossa kuvataan eri Netcool-ohjelmistojen toimintaa.

3.1 OMNIBUS

Netcool/OMNIbus on Netcool-järjestelmän ydin. Tai tarkemmin sanottuna OMNIbus ObjectServerit ovat järjestelmän ydin. OMNIbus hoitaa reaaliaikaisesti keskitettyä valvontaa hyvin monentyyppisten verkkojen osalta ja pystyy käsittelemään kymmeniä miljoonia tapahtumia (event) vuorokaudessa.

OMNIbus koostuu useista eri osamoduuleista, joista jo mainittu ObjectServer on kaikkein tärkein. OMNIbus käsittää lisäksi Netcool-järjestelmän ja ulkoisten tietokantojen välistä kommunikointia hoitavia yhdyskäytäviä (gateway) sekä valvontatietoa keräävät probe-palvelimet.

Kaikki OMNIbus-komponentit pitää konfiguroida Interfaces-tiedostoon, jotta järjestelmän sisäinen kommunikaatio toimii. Interfaces-tiedoston konfigurointi tapahtuu NCO_XIGEN-työkalun avulla. Alla oleva komento avaa Interfaces-tiedoston konfigurointikäyttöliittymän:

$OMNIHOME/bin/nco_xigen

3.1.1 Probet

HUOM! Tässä dokumentissa puhutaan sekä fyysisistä probe-palvelimista että probe-ohjelmistokomponenteista, kuten MTTrapD. Puhuttaessa fyysisestä palvelimesta käytetään tässä dokumentissa sanaa probe-palvelin. Kun kyseessä on ohjelmistokomponentti, käytetään sanaa probe-sovellus.

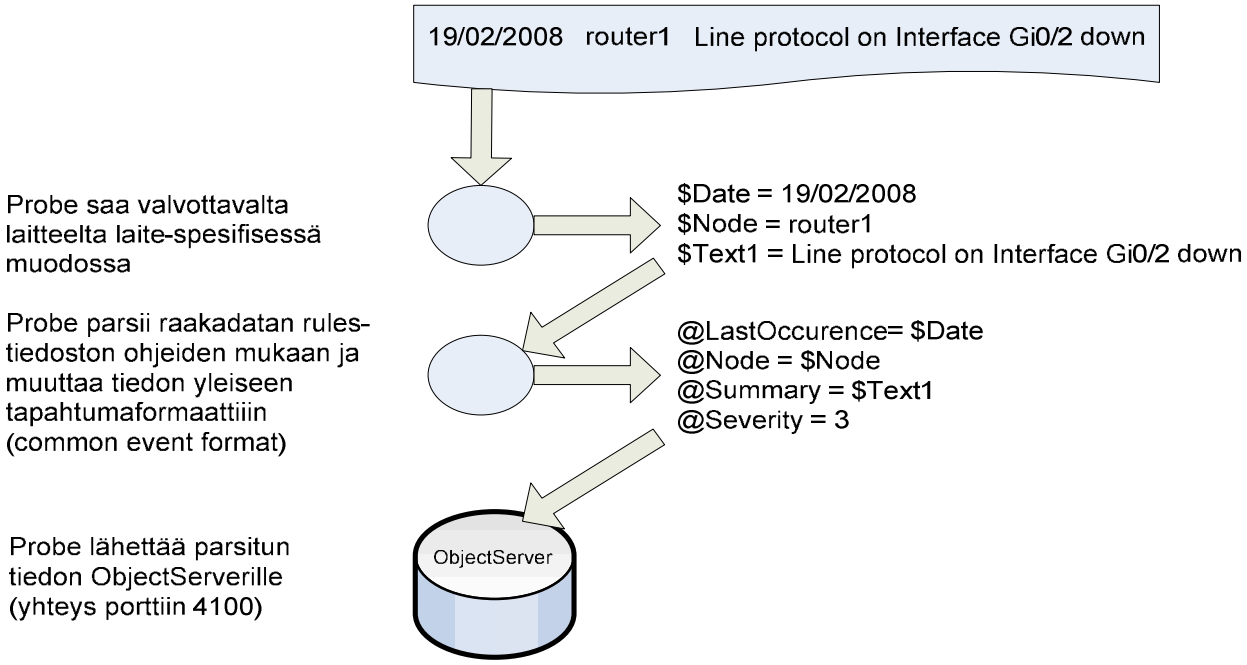
Netcool probe-sovellukset lukevat dataa valvomastaan kohteesta mm. SNMP-protokollan avulla. Kun uutta valvontatietoa syntyy, probe-palvelimet muuttavat tiedon yleiseen tapahtumaformaattiin (Common Event Format) ja toimittavat sen ObjectServerille, jossa tapahtumien käsittely suoritetaan.

Probe-palvelimen ydin on ajettavat tiedostot. Ne keräävät tapahtumatiedot valvottavista laitteista, prosessoivat tapahtumat ja lähettävät ne eteenpäin ObjectServerille hälytyksinä. Ajettavat tiedostot löytyvät $OMNIHOME/probes/<platform>-hakemistosta. Esimerkiksi SNMP trap daemon (MTTrapd) on ajettava tiedosto. Käynnistyessään probe-palvelin lukee <*probe-nimi*>.props ja <*probe-nimi*>.rules-nimiset tiedostot $OMNIHOME/probes/<*platform*>/ -kansiosta ja alkaa toimia niissä olevien konfiguraatioiden mukaisesti.

Järjestelmässä on käytössä Netcool Knowledge Library (NCKL). Se on kokoelma standardimuotoisia rules-tiedostoja ja mahdollistaa OMNIbusin SNMP-probe-sovelluksen saumattoman toiminnan Network Managerin kanssa ilman erillisiä konfigurointeja. NCKL:n käytön hyödyt on myös nähtävissä mm. siten, että se lisää korkeamman tason korrelaatiota. NCKL:n liittyen käytössä on myös MIB2RULES-työkalu (v5.25). Sen avulla NCKL:n rules-tiedostoihin voidaan lisätä uudenlaisten laitteiden MIB:ejä ja määritellä niiltä saatujen trapien asetuksia, esim. trapin ilmoittaman tapahtuman kiireellisyys (severity).

<*probe-nimi*>.props-tiedosto sisältää probe-palvelimen konfiguraatioon liittyvät asiat. Näitä ovat esim. käytettävä ObjectServer, luettava tiedosto, tiedoston sisällön käsittely ja kuunneltava portti. Konfigurointimahdollisuudet vaihtelevat sovelluskohtaisesti ja nämä tiedot on kuvattu probe-sovellusten dokumentaatiossa.

<*probe-nimi*>.rules-tiedosto määrittää, kuinka probe-sovellusten keräämää dataa käsitellään ennen sen lähettämistä Objectserverille. Tiedostossa määritellään mm. Objectserverin kenttien sisältö, Identifier-kenttä (yksilöi tapahtuman) ja tapahtuman kiireellisyys. Probe-sovellusten keräämä tapahtumatieto on muokkaamatonta, jota ei sellaisenaan voida lähettää ObjectServerille. Tästä syystä probe-sovellus katkoo tapahtumadatan tokeneihin, jolloin jokainen token esittää yhden osan tapahtumadatasta. Tämän jälkeen probe-sovellus parsii eli muuttaa tokenit elementeiksi ja prosessoi ne rules-tiedoston sääntöjen mukaisesti. Parsittujen elementtien pohjalta lisätään arvot ObjectServerin kenttiin. Kenttien arvot sisältävät tapahtuman yksityiskohdat ObjectServerin ymmärtämässä muodossa. Kentistä syntyy hälytykset. Ne lähetetään eteenpäin ObjectServerille, jossa tiedot talletetaan alerts.status-taulukkoon. Alla on kuvaus, kuinka probe parsii tokenit elementeiksi:



3.1.2 ObjectServer

Kuten mainittua, Netcool-järjestelmän ytimenä toimii Netcool/OMNIbusin ObjectServer-moduuli. ObjectServer on RAM-muistissa toimiva tietokantapalvelin, jossa käsitellään kaikista probe-palvelimista ja muista lähteistä saadut tapahtumat. ObjectServerissä kaikki tieto on tietokannoissa ja taulukoissa, jotka pyörivät ObjectServerin muistissa. Tietokannoista tärkein on alerts, joka sisältää tapahtumatiedot. Alerts-kannan tärkein taulukko taas on alerts.status, koska se sisältää aktiiviset hälytykset.

ObjectServerin ja Webtop-palvelimen välillä kulkee suuret määrät tapahtumatietoa. Jotta ObjectServer ei ylikuormittuisi, kehitettiin IDUC-protokolla (Insert, Delete, Update & Control).

Kun ObjectServerin tapahtumatietokannassa tapahtuu muutoksia, se ei lähetä uusia tietoja Webtop-palvelimelle, vaan ilmoittaa sille, että ”tietokanta on muuttunut - tule hakemaan uudet tiedot". Vakioasetuksilla ObjectServer päivittää IDUC-clienttinsa uudelle tapahtumadatalla joka 60 sekunti. Vakiona IDUC myös käyttää sattumanvaraista porttinumeroa. Jos kuitenkin ObjectServerin ja Webtop-palvelimen välinen liikenne kulkee palomuurin lävitse, voidaan, ja myös täytyy, tämä porttinumero määrittää staattiseksi. Porttinumero voi olla mikä tahansa vapaa porttinumero väliltä 1025 – 65535.

ObjectServerin konfigurointi tapahtuu lähes yksinomaan erillisellä nco_config-työkalulla, jonka avulla voidaan mm. luoda automaatioita (automation), liipaisimia (trigger) ja filttereitä (filter), hallita käyttäjiä sekä heidän käytössään olevia menuja ja työkaluja. Filtterien toimintaa on kuvattu tarkemmin jäljempänä kappaleessa 4 - Tapahtumasta laitteella valvontanäkymään. nco_config-työkalu käynnistetään ObjectServer-palvelimen komentoriviltä komennolla:

$OMNIHOME/bin/nco_config &

Ohjeet nco_config-työkalun käyttöön löytyvät IBM-dokumenttikannan Netcool/OMNIbus Administration Guide v7.1 -oppaasta. Opas on saatavilla PDF-muodossa osoitteesta:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_OMNIbus.doc/ag/om71ag.pdf

Yksi tärkeimmistä Objectserverin triggereistä on tapahtumien deduplikaatiota hoitava triggeri. Deduplikaatiolla tarkoitetaan samalta laitteelta samasta viasta johtuvien tapahtumien tunnistamista. Tällä estetään esim. että fläppäävästä linkistä ei tule joka kerta uutta tapahtumariviä valvontanäkymään, kun liitäntä menee alas tai nousee ylös. Tapahtumien deduplikointiin käytetään Node-, AlertKey-, AlertGroup- ja Manager-kenttien arvoja. Jos Objectserveriltä löytyy jo samasta viasta tapahtumarivi, jossa edellä mainittujen kenttien arvot ovat samat, kasvaa tapahtuman Count-kentän arvo (Tally). Lisäksi tapahtumarivin muiden kenttien, kuten LastOccurence, arvot päivittyvät. Muussa tapauksessa kyseessä on uusi tapahtuma, josta luodaan oma tapahtumarivinsä.

3.1.3 Yhdyskäytävät

Yhdyskäytäviä (gateway) käytetään tiedon siirtämiseen ObjectServeristä tiettyihin muihin Netcool-komponentteihin. Yhdyskäytävät mahdollistavat myös tapahtumien välittämisen ObjectServerin ja kolmannen osapuolen sovellusten, kuten tietokantojen, välillä.

Yhdyskäytäviä voidaan käyttää tapahtumien replikointiin ja backup-ObjectServerin ylläpitoon. Sovellusyhdyskäytävät mahdollistavat erilaisten bisnestoimintojen yhteensovittamisen, yhdyskäytävä voidaan esim. konfiguroida lähettämään tapahtumainformaatiota Helpdesk-sovellukselle. Yhdyskäytäviä voidaan myös käyttää tapahtumien arkistointiin tietokantaan

Yhdyskäytäviä on kahta päätyyppiä:

- Yksisuuntainen yhdyskäytävä
- Kaksisuuntainen yhdyskäytävä

Yksisuuntaiset yhdyskäytävät sallivat tapahtumadatan siirtymisen ainoastaan yhteen suuntaan. Muutokset lähteenä käytettävässä ObjectServerissä replikoidaan toiseen ObjectServeriin tai

sovellukseen. Sen sijaan kohteena olevassa ObjectServerissä tehdyt muutokset eivät replikoidu lähteenä käytettävään ObjectServeriin. Yhdensuuntainen yhdyskäytävä on siis käytännössä arkistoiva yhdyskäytävä.

Kahdensuuntaiset yhdyskäytävät sen sijaan mahdollistavat tapahtumadatan siirtymisen molempiin suuntiin. Lähteenä olevan ObjectServerin tiedot voidaan replikoida kohde-ObjectServerille tai sovellukselle ja myös kohteena oleva ObjectServer tai sovellus voi lähettää tietoa lähde-ObjectServerille. Tästä syystä kahdensuuntaisia yhdyskäytäviä pidetään synkronisointityökaluina.

Yhdyskäytävien kohteena voivat olla mm.:

6. Toinen ObjectServer (backup)
7. Tietokanta
8. Helpdesk-sovellus
9. Tikettijärjestelmä

Kaikilla luoduilla yhdyskäytävillä on oma konfiguraatiotiedosto. Tiedostot sijaitsevat $OMNIHOME/etc-hakemistossa ja niiden pääte on -conf.

3.2 IMPACT

Netcool/Impact on Netcool-järjestelmän analysointi- ja korrelointimoottori. Impactin avulla voidaan hoitaa tapahtumien rikastamista ja tiedon korrelointia. Rikastamisella tarkoitetaan probe-sovelluksilta tulevien tapahtumien tietojen täydentämistä. Rikastamisella voidaan tapahtumiin lisätä esim. laitteen osoite- ja kontaktihenkilötiedot. Lisäksi Impactin avulla OMNIbus voidaan integroida monenlaisiin kolmannen osapuolen ohjelmistoihin, kuten tietokantoihin ja verkon inventointisovelluksiin.

Impactin konfiguraatiot löytyvät kahdesta ryhmästä konfiguraatiotiedostoja. Globaalit konfiguraatiotiedostot kontrolloivat kaikkien Impact-palvelinten asetuksia ja löytyvät $OMNIHOME/impact -hakemistosta. Palvelininstanssikohtaiset konfiguraatiotiedostot kontrolloivat yksittäisen palvelininstanssin asetuksia. Palvelininstansseja on kaksi - Impact- ja GUI-palvelininstanssi.

Impact käyttää tiedostojärjestelmään pohjautuvaa versionhallintaa (Concurrent Version System - CVS) tarvitsemien tiedostojen hallintaan. Tämän vuoksi Impactin käynnistäminen oikeana käyttäjänä (netcool) on äärimmäisen tärkeää.

3.2.1 Impact-palvelininstanssi

Impact-palvelininstanssi on järjestelmän ydinkomponetti. Se koostuu alikomponenteista, jotka hoitavat koko Impact-järjestelmän toimintaa. Impact-palvelininstanssi pyörii sovellusinstanssina IBM WASCE -sovelluspalvelimessa. Kun järjestelmä käynnistetään WASCE:n kautta, käynnistyy myös Impact-palvelininstanssi.

Impact-palvelininstanssin konfiguraatiotiedostot löytyvät $NCHOME/impact/etc -hakemistosta. Vakioasetuksilla se käyttää seuraavia porttinumeroita:

Nimi	**Portti**	**Käyttötarkoitus**
Komentoriviportti	2000	Portti, joka tarjoaa komentoriviyhteyden
ObjectServer-portti	4100	Portti, jota ObjectServer kuuntelee

Yksittäinen Impact-palvelininstanssi voidaan käynnistää, pysäyttää ja poistaa seuraavien komentojen avulla

Toiminto	**Komento**
Palvelininstanssin käynnistys	$NCHOME/impact/bin/nci_server <*server*>
Palvelininstanssin pysäytys	$NCHOME/impact/bin/nci_shutdown <*server*>
Palvelininstanssin poistaminen	$NCHOME/impact/bin/nci_removeserver <*server*>

3.2.2 GUI-palvelininstanssi

GUI-palvelininstanssi on sovellus, joka tarjoaa selainpohjaisen graafisen käyttöliittymän Impactin hallintaan. GUI-palvelininstanssi pyörii sovellusinstanssina IBM WASCE -sovelluspalvelimessa. Kun järjestelmä käynnistetään WASCE:n kautta, käynnistyy myös GUI-palvelininstanssi.

GUI-palvelininstanssin asetustiedostot löytyvät $NCHOME/guiserver/etc -hakemistosta. GUI-palvelininstanssi käyttää seuraavia porttinumeroita:

Nimi	**Portti**	**Käyttötarkoitus**
HTTP-portti	8080	HTTP-portti selainyhteydelle
HTTPS-portti	8443	HTTPS-portti selainyhteydelle
NSCM-portti	8077	Portti, jota Security Manager kuuntelee
Nimeämisportti	1099	WASCE:n käyttämä RMI nimeämisportti
EJB-portti	4201	WASCE:n Enterprise Java Beans -portti
COS-nimeämisportti	5050	WASCE:n Common Object Services -portti
JMS-portti	61616	WASCE:n Java Message System-portti
Derby-portti	1527	WASCE:n Apache Derby -portti
Directory Service -portti	1389	WASCE:n Apache Directory Service -portti
JAAS-portti	4242	WASCE:n Java Authentication and Authorization Service -portti
Tomcat AJP-portti	8009	WASCE:n Tomcat AJP -yhdistäjäportti

Impactin tehtävä on siis rikastaa probe-sovelluksilta tulevia hälytyksiä. Probe-palvelin lähettää hälytykset ObjectServerille. Impact lukee aktiiviset tapahtumat ObjectServeriltä DefaultEventReader-sovelluksen avulla. Siinä määritellään myös, kuinka Impactiin luotuja toimintaperiaatteita (policy) käytetään.

3.2.3 Netcool-tietokantapalvelin

Netcool-tietokantapalvelin on erityiskäyttöön luotu versio PostgreSQL-tietokannasta. Se on valmistettu käytettäväksi Impactin ja muiden Netcool-tuotteiden kanssa. Impact käyttää tietokantaa GUI raportointityökalujen käyttämän tiedon tallennuspaikkana.

Tietokantapalvelin käynnistetään ajamalla seuraavat skriptit:

$NCHOME/bin/nc_db_postgres setupinit
$NCHOME/impact/bin/nci_db setupinit

Tietokantapalvelimen asetustiedosto on seuraava:

$NCHOME/bin/.nc_dbconf

Tietokannan käyttämä porttinumero voidaan määritellä kyseisessä tiedostossa.

Tietokantapalvelimeen liittyvät tärkeimmät komennot löytyvät seuraavasta taulukosta:

Komento	Käyttötarkoitus
$NCHOME/bin/nc_db_postgres start	Käynnistää Netcool-tietokantapalvelimen
$NCHOME/bin/nc_db_postgres stop	Pysäyttää tietokantapalvelimen
$NCHOME/bin/nc_db_postgres status	Tulostaa tietokantapalvelimen tilan
$NCHOME/impact/bin/nci_db setupinit	Resetoi tietokantapalvelimen
$NCHOME/impact/bin/nci_db connect	Avaa komentoriviyhteyden tietokantapalvelimeen
$NCHOME/impact/bin/nci_db backup -backupfile <filename>	Tekee tietokannasta backup-tiedoston
$NCHOME/impact/bin/nci_db restore -backupfile <filename>	Palauttaa tietokannan backup-tiedostosta

3.2.4 Impact-palvelinten käynnistys ja pysäytys

Impact- ja GUI-palvelininstanssien lisäksi Impact vaatii toimiakseen Netcool Security Manageria. Lisäksi on mahdollista käyttää erillistä Netcool tietokantapalvelinta ja Impact JRExec -palvelinta. Palvelimet tulisi käynnistää seuraavassa järjestyksessä:

Komponentti ja toiminto	**Komento**
NCSM käynnistys	$NCHOME/security/bin/ncsm_server
Tietokantapalvelimen käynnistys (ei välttämätön)	$NCHOME/bin/nc_db_postgres start
Impact- ja GUI-palvelininstassien käynnistys	$NCHOME/bin/wasce start
Impact- ja GUI-palvelininstassien pysäytys	$NCHOME/bin/wasce stop
Impact JRExecin käynnistys (ei välttämätön)	$NCHOM/impact/bin/nci_jrexec

Yksittäisen palvelininstanssin käynnistys- ja pysäytyskomennot löytyvät edeltä kappaleesta 3.2.1.

3.2.5 Impactin hallinta

Impactin hallinta tapahtuu selainpohjaisen käyttöliittymän kautta. Hallinta on mahdollista myös telnet-yhteyden kautta avautuvalta komentoriviltä.

Tarkemmat ohjeet Impactin hallintaan löytyvät IBM-dokumentaatiokannasta Netcool/Impact User Interface Guide - ja Netcool Impact Policy Reference Guide -oppaista:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_impact.doc/im401ug.pdf
http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_impact.doc/im401pr.pdf

HUOM! Käytettäessä Impactia selaimen kautta on varmistettava aina että kaikki tiedostot vapautetaan Impactin omassa CVS-järjestelmässä ennen kuin selainkäyttö lopetetaan. Selainyhteyttä ei myöskään saa sulkea ilman uloskirjautumista, sillä tämä voi lukita tiedoston Impactin CVS:ssä

3.3 WEBTOP

Webtop on selainpohjainen ohjelmisto, joka prosessoi verkkotapahtumia yhdestä tai useammasta tietolähteestä ja esittää tiedon graafisessa muodossa. Webtopin kautta verkonhallintatiimi pystyy hoitamaan valvontatehtäviään.

Webtop ei itse ylläpidä tapahtumatietoa. Se käyttää OMNIbus Objectservereitä tietolähteenään. Merkittävimmät tiedon esittämiskomponentit ovat:

- Aktiivinen tapahtumalista (Active Event List)
  - Java-pohjainen
  - Esittää aktiiviset tapahtumat
  - Voidaan luoda näkymiä ja filttereitä, joilla saadaan esim. ainoastaan yhden asiakkaan tapahtumat tapahtumalistaan

- Kevyt tapahtumalista (Lightweight Event List)
  - HTML-pohjainen
  - Näyttää aktiivisen tapahtumalistan

- Taulukkonäkymä (tableview)
  - Näyttää taulukoina, esim. pylväsdiagrammeina, hälytysten määrän

Aktiivisia tapahtumia esittävän valvontanäkymän lisäksi Webtopin kautta voidaan luoda ja seurata tapahtumia myös karttojen ja kuvien avulla. Esim. verkkokuvia voidaan laittaa Webtop-karttojen pohjaksi. Kun kuvassa olevien laitteiden yhteyteen liitetään laitteen tai porttien tilaa kuvaava aktiivinen objekti, voidaan suoraan verkkokuvasta nähdä, mikä laite tai portti menee alas ja onko sillä vaikutuksia muuhun verkkooon.

3.3.1 Webtop ja Netcool GUI Foundation

Webtopin asennusvaiheessa palvelimelle asennettiin automaattisesti myös Netcool GUI Foundation (NGF), jota Webtop tarvitsee toimiakseen. Webtop käyttää client-server-arkkitehtuuria ja pyörii NGF:n sisällä. Client-koneet ottavat yhteyden NGF:iin, kun ne haluavat yhteyden Webtopiin. NGF on palvelinohjelma, joka tarjoaa graafisia käyttöliittymiä (GUI = Graphical User Interface) eri Netcool-tuotteille. Autentikointiin ja auktorisointiin NGF käyttää Netcool Security Manager -ohjelmaa (NCSM). Webtop-käyttäjille pitää siis olla luotuna käyttäjätunnukset Security Manageriin, että he saavat selainyhteyden Webtopiin.

HUOM! Jos GUI Foundation-palvelin buutataan, Network Manager hukkaa SNMP MIB Browserin helper-osoitteen.

3.3.2 Webtopin tärkeimmät hakemistot

Seuraavassa on lueteltu hakemistoja, joista löytyy Webtopin, Netcool GUI Foundationin ja Security Managerin tärkeimmät tiedostot:

Komponentti	Hakemisto	Sisältö
NCSM	$NCHOME/security/bin	Skriptit ja ajettavat tiedostot
NCSM	$NCHOME/security/db	PostGres-tietokantatiedostot
NCSM	$NCHOME/security/etc	Konfigurointi- ja käynnistystiedostot
NCSM	$NCHOME/security/install	Default-tiedostot, mm. skripti NCSM:n konfigurointiin
NCSM	$NCHOME/security/log	Loki-tiedostot
NGF	$NCHOME/guifoundation/conf	NGF-konfigurointitiedostot
NGF	$NCHOME/log/guifoundation	NGF-lokitiedostot
Webtop	$NCHOME/webtop/bin	Webtop skriptit ja ajettavat tiedostot
Webtop	$NCHOME/etc/webtop	Webtop-konfigurointitiedostot
Webtop	$NCHOME/etc/webtop/resources/__common	Hakemisto, johon kopioidaan kartoissa ym. näkymissä käytettävät elementit

3.3.3 Webtopin tärkeimmät tiedostot

Seuraavassa on lueteltuna Webtop-komponenttien tärkeimpiä tiedostoja:

Komponentti	**Tiedosto**	**Tiedoston käyttötarkoitus**
Webtop	$NCHOME/etc/webtop/datasources/ncwDataSourceDefinitions.xml	Sisältää Webtopin tietolähteen konfigurointitiedot
Webtop	$NCHOME/etc/webtop/server.init	Sisältää Webtopin sessio- ja ympäristöasetukset
Webtop	$NCHOME/log/webtop.log	Sisältää Webtopin debug- ja käyttäjälokitietoja
NCSM	$NCSM_HOME/etc/<*server_name*>_server.props	Security Managerin asetukset
NCSM	$NCSM_HOME/etc/smParentType.type	NCSM tietotyyppiasetukset

3.3.4 Selainyhteys Webtop-palvelimeen

Selainyhteyden Webtop-palvelimeen voidaan ottaa seuraavilla selaimilla:

Käyttöjärjestelmä	**Selain**
Windows XP	Internet Explorer 6.0, Mozilla Firefox 1.5 ja 2.0
Windows 2003 Server	Internet Explorer 6.0, Mozilla Firefox 1.5 ja 2.0
Windows Vista	Internet Explorer 7.0, Mozilla Firefox 1.5 ja 2.0
Linux (Red Hat Enterprise Linux 4.0, SUSE Linux Enterprise Server 9 ja 10)	Mozilla Firefox 1.5 ja 2.0
Solaris 9 ja 10	Mozilla Firefox 1.5 ja 2.0

Koneelle täytyy lisäksi olla asennettuna Sun Java Virtual Machine plug-in. Tällä hetkellä tuetut Java-versiot ovat 1.4.2, 1.5 ja 1.6.0 Update 2. Lisäksi selain täytyy olla asetettu hyväksymään kaikki cookiet.

HUOM! Java plug-in 1.6.0 Update 1 -versiota ei saa käyttää!

3.3.5 Webtop-näkymien muokkauksesta

Webtop-näkymien (mm. tapahtumalistojen ja karttojen) muokkaus tapahtuu selaimella Webtop-palvelimeen otetun yhteyden kautta. Näkymien muokkaajalla täytyy olla admin- tai muokkausoikeudet haluamaansa näkymään. Webtopissa on valmiina ja sinne voidaan luoda eritasoisilla oikeuksilla varustettuja käyttäjätilejä. Kaikki kartoissa ja muissa näkymissä käytettävät kuvaelementit, kuten verkkokuvat, pitää kopioida Webtop-palvelimelle kansioon:

$NCHOME/etc/webtop/resources/__common

Lisäksi resurssit pitää määritellä selaimen kautta avautuvassa karttaresurssiselaimessa:

Webtop Admin-page --> Pages --> Map Resource Browser

Ohjeet Webtop-näkymien muokkaukseen löytyvät IBM-dokumentaatiokannassa olevasta Netcool/Webtop Administration Guide version 2.1 -oppaasta ja Webtop-kurssilla jaetusta ja läpikäydystä materiaalista. Administration Guide löytyy PDF-muodossa osoitteesta:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_wt.doc/ag/Administration_Guide.pdf

3.3.6 Weptop Administration API

Webtop Administration API (WAAPI) -client on Java-pohjainen työkalu, jonka avulla Webtopia voidaan konfiguroida etänä. WAAPI-clienttia voidaan käyttää monimutkaisten konfiguraatioiden nopeaan lähettämiseen Webtop-palvelimelle. Sillä voidaan myös ottaa esimääritellyt konfiguraatiot heti käyttöön Webtop-palvelimella. WAAPI lähettää XML-muotoisen komentotiedoston suoraan Webtop-palvelimelle, joka päivittää itsensä sen mukaisesti. WAAPI-clientin kautta voidaan muokata seuraavia ominaisuuksia Webtopissa:

- Oliot ja olionäkymät (entities and entity views)
- Tiedostojen hallinta (file management)
- Kartat (maps)
- Menut (menus)
- Resurssit (resources)
- Työkalut (tools)
- Käyttäjätieto ja -oikeudet (user information and privileges)

WAAPI-client käynnistetään komennolla:

$NCHOME/webtop/bin/runwaapi

Tarkemmat ohjeet WAAPI-clientin käytöstä löytyvät edellisessä kappaleessa mainitusta IBM:n Netcool/Webtop Administration Guide -oppaasta.

3.4 IBM TIVOLI NETWORK MANAGER IP EDITION

IBM Tivoli Network Manager IP Edition (myöhemmin Network Manager) on Netcool-tuoteperheen verkonmallintamis- ja juurisyyanalyysimoottori. Vielä edellisessä 3.6-versiossa Network Managerin nimi oli Precision ja tuote tunnetaankin yleisesti juuri Precision-nimellä. Se tuo Netcool-järjestelmään seuraavat toiminnallisuudet:

4. Verkkojen automaattinen löytäminen
5. Verkkojen valvonta
6. Verkkotopologian visualisointi
7. Juurisyyanalysointi eli juurisyyn määrittäminen

Network Managerin tärkein toiminnallisuus on verkkojen automaattinen löytäminen ja mallintaminen. Sen avulla järjestelmän ylläpitäjä voi konfiguroida ja ajaa verkkojen löytöhakuja ja uudelleenlöytöhakuja. Network Manager etsii kaikki määritellyn haun piiriin kuuluvat laitteet, joihin se saa yhteyden, ja piirtää kerättyjen tietojen perusteella verkosta tai verkoista topologiakuvan. Network Manager pystyy piirtämään topologiakuvat TCP/IP-verkoista OSI-kerroksen 2- ja 3-kerroksella. Topologiatiedoista voidaan luoda edelleen topologiakartta, jota voidaan muokata näyttämään esim. aliverkot, VLAN:it tai vain tuotannossa olevat laitteet. Tämän lisäksi Network Manager voidaan myös konfiguroida valvomaan löytämiään laitteita. Network Manageria voidaankin pitää älykkäänä probe-palvelimena.

Topologian visualisointiin Network Manager käyttää Topoviz-ohjelmaa. Topoviz sisältää monenlaisia graafisia käyttöliittymiä, joiden avulla topologiakuvia voidaan muokata tehokkaasti.

Network Managerin luomia topologiakuvia ja -karttoja muokataan ja katsellaan Webtopin kautta. Näin ollen Webtopin kautta verkonvalvonta voi helposti seurata sekä aktiivista tapahtumalistaa että Network Managerin luomia topologiakuvia.

Kerätty topologiatieto tallennetaan topologiatietokantaan, josta käytetään myös nimitystä NCIM (Netcool Common Information Model). Se on relaatiotietokanta (MySQL), joka pitää sisällään TCP/IP-verkkojen 2- ja 3-kerrosten topologiatiedot.

Juurisyyanalysointi tarkoittaa perimmäisen syyn määrittämistä, miksi yksi tai useampia laitteita hälyttää. Vika yhdessä laitteessa aiheuttaa monesti useita hälytyksiä kyseisestä verkosta. Kun yhteys yhteen vialliseen laitteeseen katkeaa, myös yhteys muihin laitteisiin voi katketa. Juurisyyanalyysi korreloi saadut hälytykset keräämäänsä topologiatietoon ja pystyy sen perusteella määrittämään, mitkä laitteet ovat saavuttamattomissa jonkin muun laitteen vikatilanteen vuoksi. Tällaisten laitteiden hälytykset ”tukahdutetaan” väliaikaisesti. Toisin sanoen ne esitetään oireina juuriviasta.

3.4.1 Avainkomponentit

Network Managerin sisäinen kommunikointi on toteutettu Tibco Rendezvous -väylän avulla. Network Managerin eri palvelut ja alikomponentit ottavat yhteyttä Tibco-väylään, kun ne haluavat välittää tai kuunnella tietoa. Network Managerin alikomponentteja ovat seuraavat:

- ncp_auth
  - autentikoi Network Managerin Object Query Language (OQL)-moottorin käyttäjiä ja prosesseja

- ncp_ctrl
  - prosessikontrolleri, joka käynnistää Network Manager-komponentit prosessiriippuvuuksien määrittämässä järjestyksessä
  - uudelleenkäynnistää automaattisesti prosessit, jotka kaatuvat
  - asetukset löytyvät $NCHOME/etc/CtrlServices.<*domain_name*>.cfg –tiedostosta

- ncp_class
  - tarjoaa hierarkisen määrittelyrakenteen löydetyille verkkolaitteille
  - muodostuu ryhmästä Active Object Class (AOC)-tiedostoja
  - jokainen AOC määrittää laitenimen, jäsenyyden, pollausasetukset ja säännöt, kuinka tapahtumaa käsitellään

- ncp_model
  - säilyttää toimivaa aktiivista topologiaa, jota koko järjestelmä käyttää
  - jokaisella laitteella on luokkanimi, joka määrittää, kuinka laitetta luokitellaan, esitetään, monitoroidaan ja kuinka juurisyyanalyysi prosessoi laitteelta tulevat virhetilanteet

- ncp_store
  - ylläpitää recovery-tiedostoja tapahtuma- ja topologiatiedoista

- ncp_disco
  - määrittelee ja hoitaa verkkojen automaattista löytämistä

- rakentaa löydetyistä laitteista topologian ncp_model-prosessin käyttöön
- konfigurointi tapahtuu Webtop-selainyhteyden kautta (ks. 3.4.4)
- Seed-asetus määrittää, mistä osoiteavaruuksista laitteita etsitään

- ncp_d_helpserv
  - sisältää apuprosesseja, joita hakuagentit käyttävät hyväkseen

- ncp_monitor
  - hoitaa valvottavien laitteiden ICMP- ja SNMP-pollausta

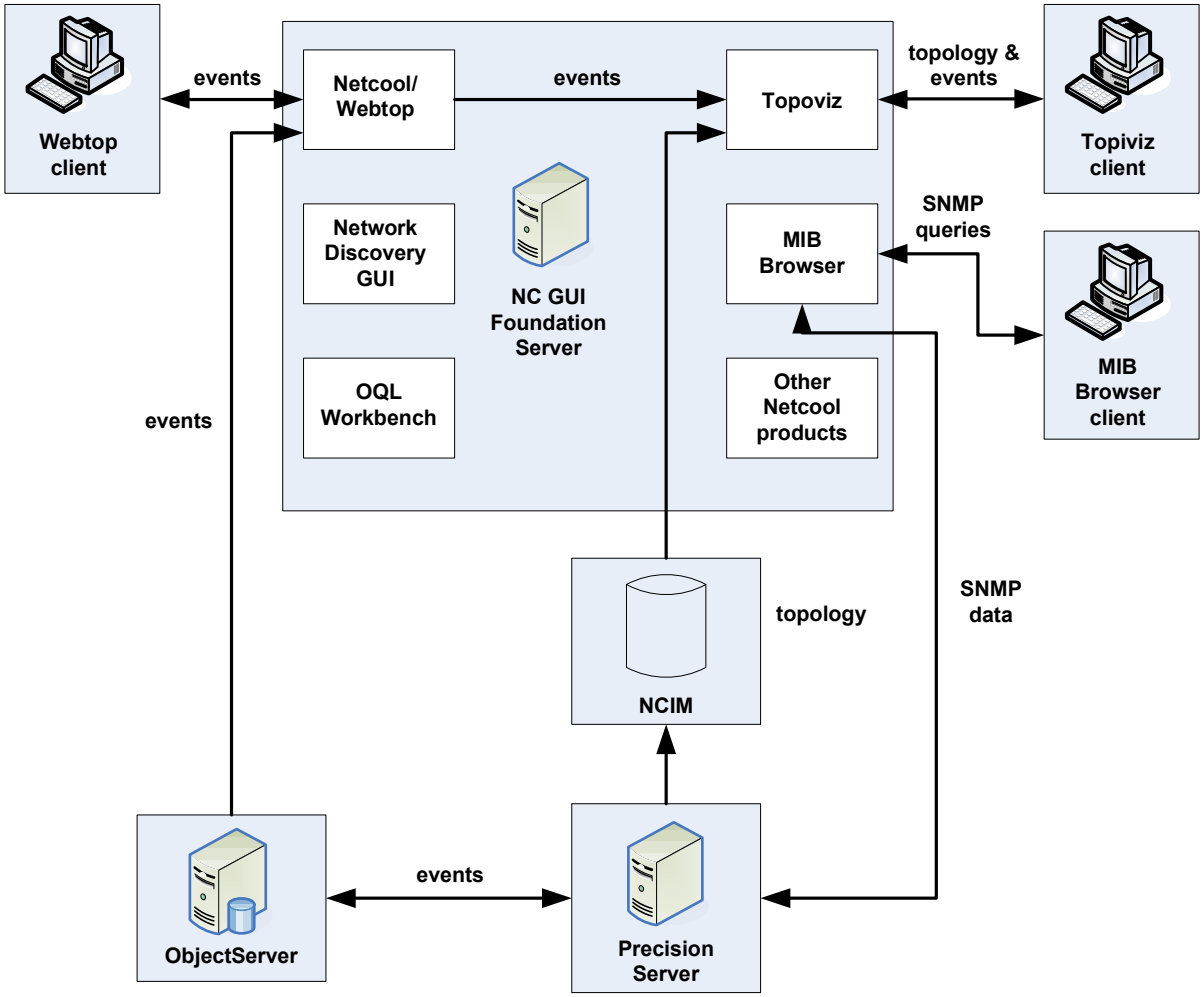
Verkkojen löytämisprosessi sisältää omat alikomponenttinsa, joiden toiminnan ymmärtäminen on tärkeää järjestelmän ylläpitäjälle. Löytämisprosessi koostuu seuraavista alikomponenteista:

- Finders
  - ottavat selvää, mitä laitteita verkosta löytyy
  - mm. ping-finder, file-finder, traps-finder

- Agents
  - selvittävät laitteiden liitäntätiedot ja kuinka laitteet ovat kytketty toisiinsa
  - käyttää SNMP-. Telnet- ja SSH-protokollia
  - mm. details-agent, joka kerää laitekuvauksen (sysDescr) ja objekti-ID:n (OID)

- Stitchers
  - yhdistää löydetyt laite- ja liitettävyystiedot ja kokoaa niiden pohjalta verkon topologian
  - SendTopologyToModel-stitcher lähettää kootut tiedot ncp_model-prosessille

3.4.2 Network Managerin arkkitehtuuri



3.4.3 Network Managerin tärkeimmät hakemistot

Seuraavassa taulukossa on lueteltuna Network Managerin tärkeimpiä hakemistoja:

Hakemisto	**Sisältö**
$NCHOME/var/precision	väliaikaisesti tallennetut (cache) topologia-, tapahtuma- ja discovery failover-tiedostot
$NCHOME/etc/precision	Network Managerin konfiguraatiotiedostot
$NCHOME/log/precision	Network Managerin lokitiedostot
$PRECISION_HOME/disco/agents	Discovery agenttien konfiguraatiotiedostot
$PRECISION_HOME/disco/stitchers	Discovery stitcherien konfiguraatiotiedostot
$PRECISION_HOME/aoc	Active Object Class-tiedostot
$PRECISION_HOME/aoc/rca_rules	Tiedostot, jotka sisältävät juurianalyysisäännöt

3.4.4 Network Managerin käynnistys, sammutus ja prosessien tarkistaminen

HUOM! Ennen Network Manager-palvelimen käynnistystä pitää Security Manager- ja Netcool GUI Foundation-palvelimet olla käynnistettynä.

Komento	**Käyttötarkoitus**
$PRECISION_HOME/bin/ncp_ctrl -domain <domain_name>	Käynnistää Network Manager-prosessin
pkill ncp	Network Managerille ei erillistä pysäytyskomentoa
$PRECISION_HOME/bin/ncp_oql	Prosessien statuksen tarkistaminen Ctrl-prosessista: 1. $PRECISION_HOME/bin/ncp_oql -domain <*domainname*> -service ctrl -username <*admin_username*> 2. select * from services.inTray; -tulostaa kaikki prosessit, jotka CTRL-prosessin hallinnassa 3. select serviceName, domainName, processId from services.inTray where serviceState = 4; -tulostaa kaikki CTRL:n kontrolloimat käynnissä olevat prosessit 4. select * from services.unManaged; -prosessit, jotka käynnissä, mutta ei CTRL:n kontrolloimia

3.4.5 Network Managerin käyttämät porttinumerot

Nimi	**Portti**	**Käyttötarkoitus**
SNMP	161	Portti, johon SNMP-kyselyt lähetetään
SNMP Trap	162	SNMP Trap-portti, jota käyttää Trap Finder ja Trap Polling Agent
MySQL	3306	MySQL-portti, käytetään TopoViz – MySQL-kannan -kommunikaatiossa
ObjectServer	4100	Portti, johon tapahtumatieto lähetetään ObjectServerille
Tibco Rendezvouz	7500	Vakioportti Tibco Rendezvouz-väylän käyttöön
RVA	7600	RVA-vakioportti, jota käyttää Discovery Configuration GUI
NCSM	8077	Security Managerin porttinumero
License Server	27000	License Serverin porttinumero

3.4.6 Network Managerin hallinta

Network Managerin hallinta ja topologiakuvien katselu tapahtuu selainyhteydellä Webtopin kautta. Network Manageriin voidaan luoda eritasoisia admin-tunnuksia, joille voidaan antaa järjestelmään vain tietyt muokkausoikeudet. Webtop:in kautta admin-käyttäjät voivat:

- hallita käyttäjiä ja käyttäjäryhmiä
- hallita verkkojen löytämishakuja (Discovery)
- luoda erilaisia valvonta- ja hallintanäkymiä, jotka voivat sisältää esim. tapahtumalistoja ja karttoja
- tehdä hakuja tietokantoihin ja hallita niitä OQL Workbench:in avulla
- katsella luotuja togologiakuvia ja -näkymiä

3.5 PROVISO

Proviso on Netcool-tuoteperheen raporttienluontiohjelmisto. Provison avulla saadaan tehokkaampi ja parempi näkymä palvelunlaatuun ja -käyttöön. Samalla järjestelmä mahdollistaa ongelmatilanteiden nopean havaitsemisen ja korjaamisen. Järjestelmän tuottamien raporttien pohjalta on helppoa parantaa palvelunlaatua ja pienentää operatiivisia kustannuksia ja sitä myötä parantaa asiakastyytyväisyyttä.

3.5.1 Provison alikomponentit

Proviso kerää, säilyttää ja raportoi suoritustietoa. Se koostuu useammasta alikomponentista. Provison neljä keskeistä komponenttia ovat:

- Proviso DataMart
  - kokoelma graafisia hallinta-, konfigurointi- ja vianselvityskäyttöliittymiä järjestelmän ylläpitäjän käyttöön
  - GUI käynnistetään komennolla: $PVMHOME/bin/pvm

- Proviso DataLoad
  - suorittaa luotettavaa, politiikan mahdollistavaa, jaettua SNMP- ja bulk-tiedon keruuta sekä tiedon tietokantaan tallennusta

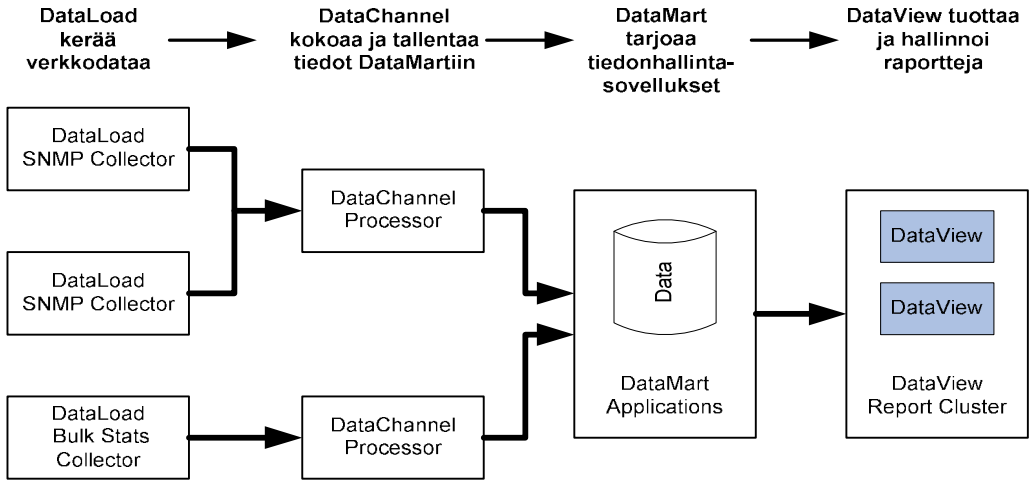
- Proviso DataChannel
  - kokoaa yhteen Proviso DataLoad -komponentin kautta kerätyn tiedon ja tarjoaa sen Proviso DataView -komponentin raportointiominaisuuksien käyttöön
  - prosessoi myös on-line-laskelmia ja havaitsee tosiaikaisesti, kun asetetut kynnysarvot ylittyvät

- Proviso DataView
  - sovelluspalvelin selainpohjaisten verkkoraporttien luontiin

Lisäksi Proviso-järjestelmää varten on erillisiä sovelluspaketteja. Ne ovat kustomoituja ohjelmanpätkiä, jotka koostuvat MIB:eistä, hakukaavioista, keruukaavioista, ryhmittelysäännöistä, raporttipohjista ja muista Provison toiminnallisuuksista yhden tietyn teknologian tai laitevalmistajan laitteiden tiedon keruuseen, kokoamiseen ja raportointiin (bulk). Ne mahdollista sen, että Provison avulla voidaan raportoida mitä tahansa teknologiaa ja mitta-arvoa. Esimerkkinä sovelluspaketista voisi mainita esim. Cisco QoS Application Pack ja Lotus Domino Application Pack.

3.5.2 Provison arkkitehtuuri

Seuraava kaavio kuvaa Provison sisäistä arkkitehtuuria ja näyttää Proviso-komponenttien keskenäiset vuorovaikutussuhteet:



3.5.3 Provison tärkeimmät hakemistot

Hakemisto	**Sisältö**
/opt/datamart/bin	DataMartin ajettavat komentorivisovellukset
/opt/dataload/bin	DataLoadin ajettavat komentorivisovellukset
/opt/datachannel/bin	DataChannelin ajettavat komentorivisovellukset
/opt/datachannel/conf	DataChannelin konfiguraatiotiedostot
/opt/dataview/bin	DataViewin ajettavat komentorivisovellukset

3.5.4 Proviso-komponenttien käynnistys ja pysäytys

Komponentti	**Komento**	**Käyttötarkoitus**
DataLoad	pvmdmgr start	Käynnistää DataLoadin SNMP-keräilijän (collector)
DataLoad	pvmdmgr stop	Pysäyttää DataLoadin SNMP-keräilijän (collector)
DataChannel	dccmd start <DC component>	Käynnistää DataChannel-komponentin, esim: dccmd start BCOL.2.2
DataChannel	dccmd stop <DC component>	Pysäyttää DataChannel-komponentin, esim: dccmd stop all
DataMart	provisoinfod start	Käynnistää Proviso Info Daemonin DataMart-palvelimella
DataMart	provisoinfod stop	Pysäyttää Proviso Info Daemonin DataMart-palvelimella
DataView / SilverStream	/etc/init.d/SilverStream start	Käynnistää DataViewin SilverStreamin
DataView / SilverStream	/etc/init.d/SilverStream stop	Pysäyttää DataViewin SilverStreamin
Oracle	lsnrctrl start	Käynnistää Oracle-kuuntelijan (listener), jolloin Proviso voi ottaa yhteyttä kantaan
Oracle	lsnrctrl stop	Pysäyttää Oracle-kuuntelijan

3.5.5 Provison hallinta- ja raporttienluontikäyttöliittymät

Järjestelmän hallinta ja konfigurointi tapahtuu DataMart-palvelimen kautta. Graafiset käyttöliittymät käynnistetään komentoriviltä ja ne vaativat toimiakseen X-ikkunoinnin käyttöä:

$PVMHOME/bin/pvm

- Käynnistää DataMart-palvelimen ja avaa DataMart-palvelimen graafisen konfigurointikäyttöliittymän

$PVMHOME/bin/pvmstat

- Avaa DataMart-palvelimen Status Tool -työkalun graafisen käyttöliittymän

$PVMHOME/bin/collectinfo

- Avaa DataMart-palvelimen Collector Information Tool -työkalun graafisen käyttöliittymän

Raporttien luomista varten järjestelmässä on oma graafinen käyttöliittymänsä. Etäyhteyttä varten työasemalle täytyy asentaa SilverJRunner-ohjelmisto (ks. 6.2), jonka jälkeen käyttöliittymä käynnistetään seuraavasti:

4. Työaseman komentorivillä navigoi hakemistoon, johon SilverJRunner on asennettu

5. Käynnistä ohjelma komennolla:

   SilverJRunner <*palvelin*>:X PV DVNavigator

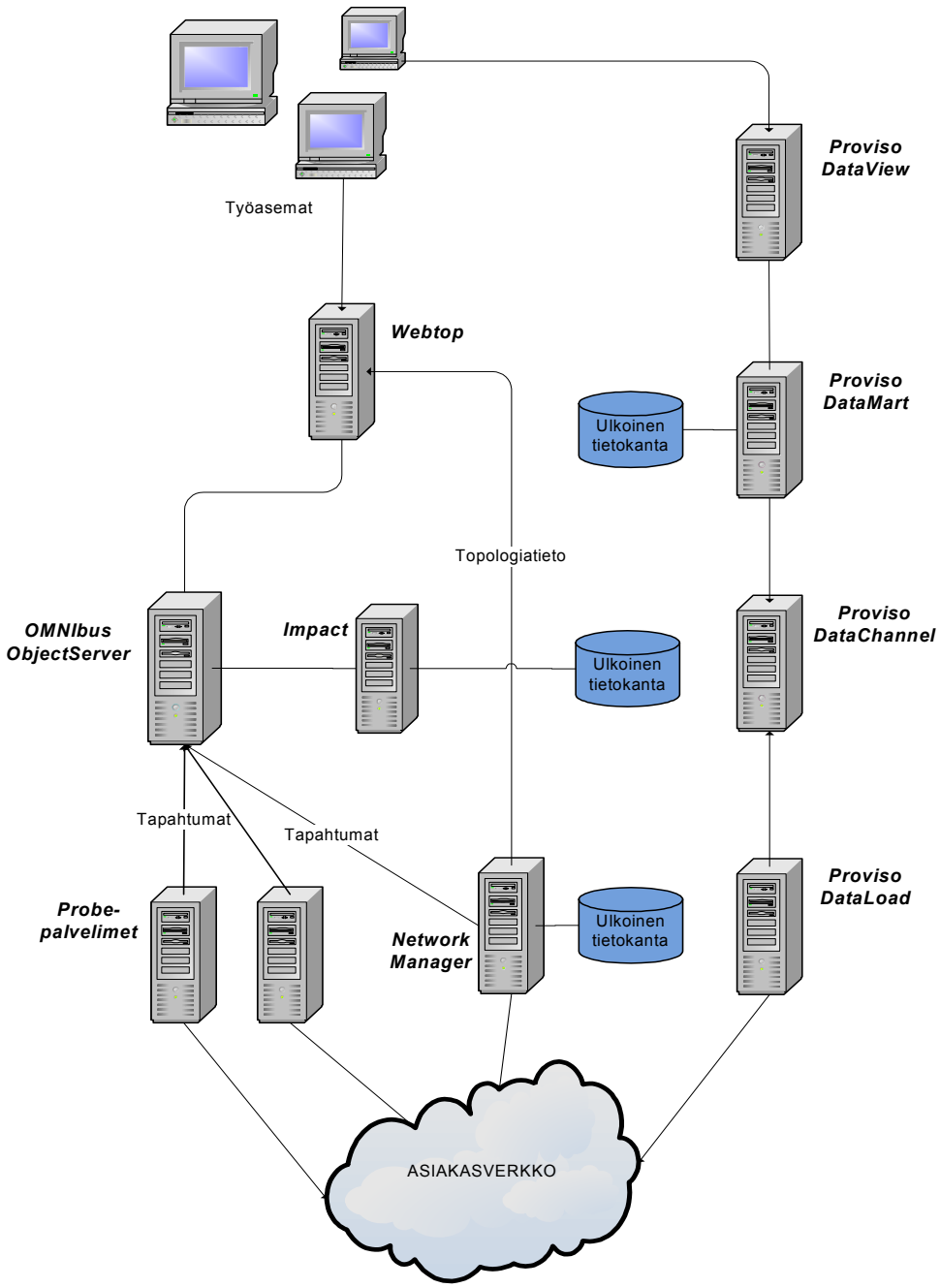
6. Anna käyttäjätunnus ja salasana ja klikkaa OK.

Sovellus on mahdollista käynnistää myös suoraan DataView-palvelimen komentoriviltä komennolla:

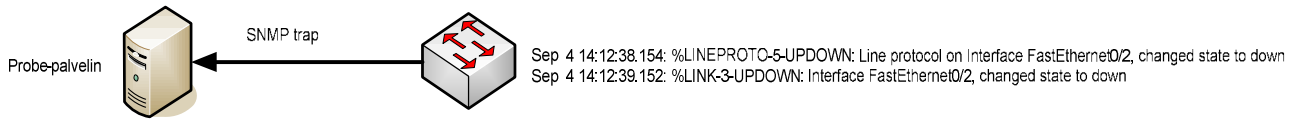
/opt/silverstream/bin/SilverJRunner localhost:X <*database_name*> DVNavigator

4 NETCOOL-KOMPONENTTIEN VÄLINEN KOMMUNIKOINTI

Alla on kokonaiskuva Netcool-järjestelmän eri komponenttien välisestä kommunikaatiosta ja sen perässä kuvaus toiminnasta:



Kun valvottavalla laitteella tapahtuu odottamatonta, esim. liitäntä menee alas, laite lähettää tapahtumasta SNMP trapin probe-palvelimelle.



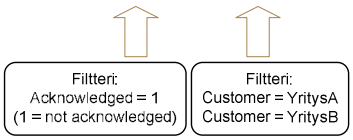
Probe-palvelin parsii saadun tapahtuman rules-tiedostojen sääntöjen mukaisesti yleiseen tapahtumaformaattiin (Common Event Format). Tämän jälkeen probe-palvelin ottaa TCP-yhteyden ObjectServerin kuuntelemaan porttiin ja lähettää tiedot tapahtumasta ObjectServerille. ObjectServer laittaa tapahtuman muistissa pyörivään alerts-tietokantaan alerts.status-taulukkoon.

Koska Probe lähettää ainoastaan perustiedot tapahtumasta, kuten laitteen identifointitiedon ja tiedon ongelmasta, ennen tapahtuman esittämistä aktiivisella tapahtumalistalla, Impact lisää hälytykseen laitetietoja erillisestä laitetiedot sisältävästä tietokannasta. Tällaista tietoa voi olla esim. laitteen sijainti- ja kontaktitiedot. Impact ottaa yhteyden ObjectServerin kuuntelemaan porttiin ja lukee ObjectServerillä olevat aktiiviset tapahtumat DefaultEventReaderillä. Tämän jälkeen Impact lisää tapahtumiin tietoa siihen luotujen toimintaperiaatteiden (policy) mukaisesti laitetietokannasta ja palauttaa hälytykset ObjectServerin alerts.status-taulukkoon. Lisäksi Impact kirjoittaa samat hälytystiedot PostgreSQL-tietokantapalvelimella pyörivään tietokantaan. Tähän kantaan tapahtumatiedot kirjoitetaan aina, kun tulee uusia tapahtumia, olemassa olevia tapahtumia muokataan ja hälytys poistuu.

Hälytysten seuraamiseen käytetään Webtop-palvelimen kautta selaimella avautuvia tapahtumalistoja. Client-koneet ottavat Webtopiin yhteyden ja pääsevät sitä kautta näkemään tapahtumalistoja. Tapahtumalistojen ylläpitämistä varten Webtop ottaa TCP-yhteyden ObjectServeriin tapahtumatietojen saamiseksi. Kun uutta tapahtumadataa on saatavilla ja Webtop tarvitsee päivitystä tietoihin, ObjectServer lähettää Webtop kehotuksen hakea päivitetyt tapahtumatiedot. Tähän käytetään Insert, Delete, Update or Control (IDUC) -protokollaa. ObjectServerin lähettämä kehotus ohjeistaa Webtopia päivittämään kaikki tapahtumalistanäkymät.

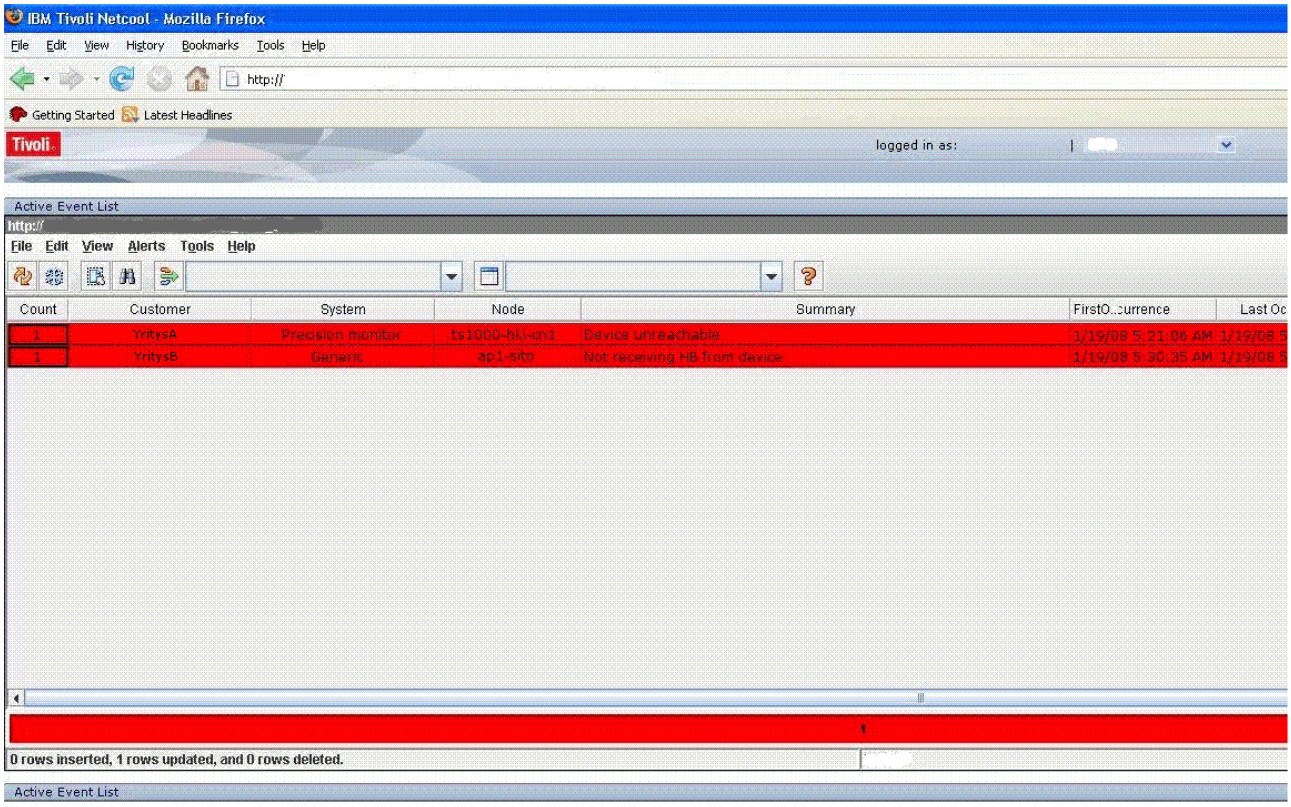
Webtop-käyttäjille tapahtumalistoilla näkyviin hälytyksiin voidaan vaikuttaa entiteettien, kuten filtterien, avulla. Toisin sanoen filtteröinnillä vaikutetaan siihen, mitä hälytyksiä kyseiselle käyttäjälle näkyy, esim. aktiivisten kuittaamattomien hälytysten -näkymässä filtteröidään seuraavia kenttiä:

Severity	Count	Acknowledged	Customer	Node	Alarm	Freetext	Address
5	1	1	YritysA	A-router1	Device unreachable		Katu 1, 00500 Helsinki
5	2	1	YritysB	B-device1	Not receiving HB from device		Katu 2, 00500 Espoo
4	1	1	Yritys C	router1	Interface Gi0/2 down		Kuja 3, 00500 Imatra
5	5	2	YritysA	A-router2	Device unreachable		Polku 5, 00500 Vantaa



Käytössä on siis kaksi filtteriä. Ensimmäinen filtteri määrittää, että kyseessä tulee olla uusi tapahtuma, jota ei ole vielä kuitattu (acknowledge). Kun tapahtuma kuitataan, se siirtyy erilliseen kuitatut Acknowledged–näkymään ja Acknowledged-kentän arvoksi muuttuu 2. Toinen filtteri

määrittää, että tässä näkymässä esitetään vain rivit, joissa asiakas eli Customer-kentän arvo on joko YritysA tai YritysB. Kun filtterit yhdistetään, kyseinen näkymä näyttää Webtopin ruudulla seuraavalta:



5 DOKUMENTTIEN LUONTI PROVISOLLA

Provisoa käytetään raporttien luontiin. Raportit luodaan Proviso DataView Navigator -sovelluksen avulla. Sovelluksessa on valmiita raporttirunkoja sekä joitain perustyylipohjia, joita voidaan käyttää hyväksi. Suurin osa tyylipohjista jaetaan Proviso Application Pack-sovelluspakettien mukana. Sovelluspaketit ovat erillisiä softapakkauksia, jotka on suunniteltu keräämään ja esittämään tietynlaista tietoverkkodataa, esim. Cisco QoS ja MPLS MIB:ien tuottamaa dataa.

5.1 RAPORTTIEN LUONNIN JA TARKASTELUN PERUSVAIHEET

Raporttien luonnin ja katselemisen perusvaiheet ovat seuraavat:

1. Luodaan uusi Proviso-tyylipohja ja asennetaan se Proviso DataView -palvelimelle tai muokataan olemassa olevaa tyylipohjaa.

2. DataView Navigator -sovellusta käyttäen:

    a. Luodaan raporttipohja ja yhdistetään siihen tyylipohja.

    b. Luodaan yksittäinen raportti-instanssi määrittelemällä useita raporttipohjan parametrejä sekä asetuskohtaisia tiedonhakuja ja filttereitä.

    c. Testataan, miltä luotu raportti näyttää selaimella.

3. Viimeistelty raportti tuodaan Portal-näkymään – tämän tekee yleensä Proviso-ylläpitäjä.

4. Käyttäjät, joilla on oikeus raportin katseluun, näkevät sen Portal-näkymän kautta – Proviso DataView -palvelin lukee raporttipohjan tiedot, hoitaa tarvittavat tiedonkeruut, täyttää raporttipohjan kerätyllä tiedolla ja esittää raportin käyttäjän selaimella.

5.2 RAPORTTIEN LUONTI REPORTER SET WIZARD -TYÖKALUN AVULLA

Reporter Set Wizard on Provison työkalu, jonka avulla voidaan luoda ryhmä raportteja, jotka antavat kuvan tietyn teknologian toiminnasta.

Reporter Set Wizard -työkalun käyttäminen:

1. Käynnistä Proviso DataView Navigator seuraavasti:

    Paikallisesti DataView-palvelimelta
    a. Kirjaudu Proviso DataView -palvelimelle
    b. Siirry hakemistoon, johon SilverJRunner on asennettuna:

        cd /opt/silverstream/bin

    c. Käynnistä DataView Navigator komennolla:

/SilverJRunner localhost:X <*database_name*> DVNavigator

d. Anna käyttäjätunnus ja salasana ja klikkaa OK

Etänä omalta työasemalta

e. Jos työasemalle ei ole vielä asennettuna SilverJRunner-sovellusta, lataa se DataView-palvelimelta:

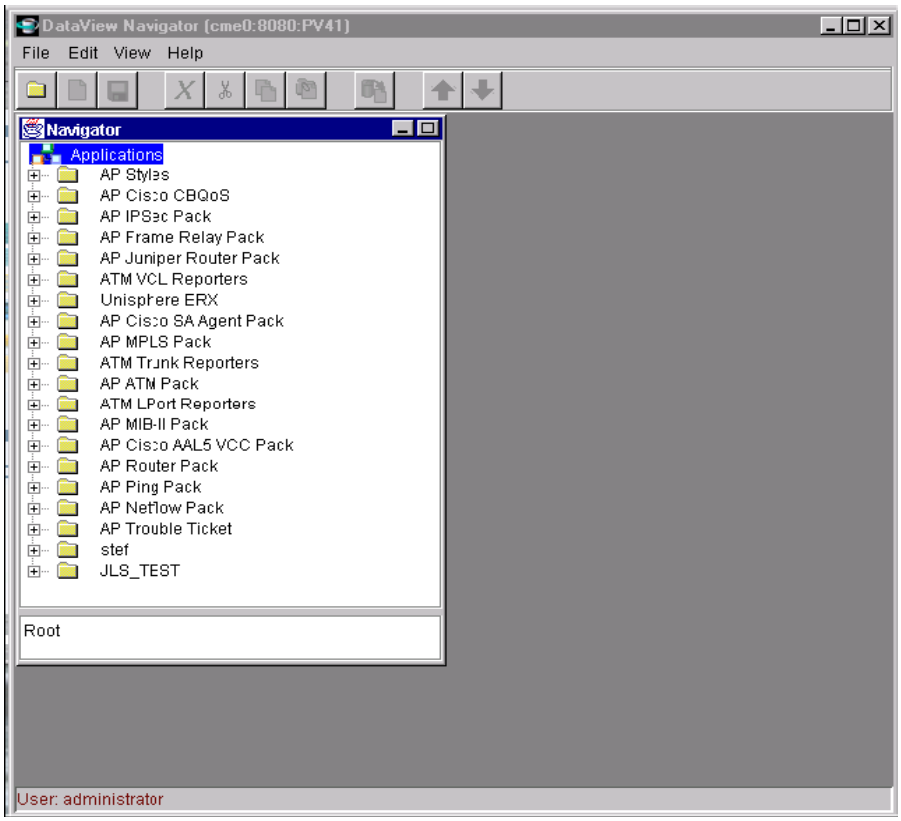
http://<*server_name*>:X/SilverStream/Pages/SilverJRunner.html

f. Kun SilverJRunner-sovellus on asennettuna työasemalle, avaa komentokehote ja navigoi hakemistoon, johon asensit SilverJRunnerin ja käynnistä sovellus komennolla:

SilverJRunner < *server_name*>:X PV DVNavigator

g. Anna käyttäjätunnus ja salasana ja klikkaa OK

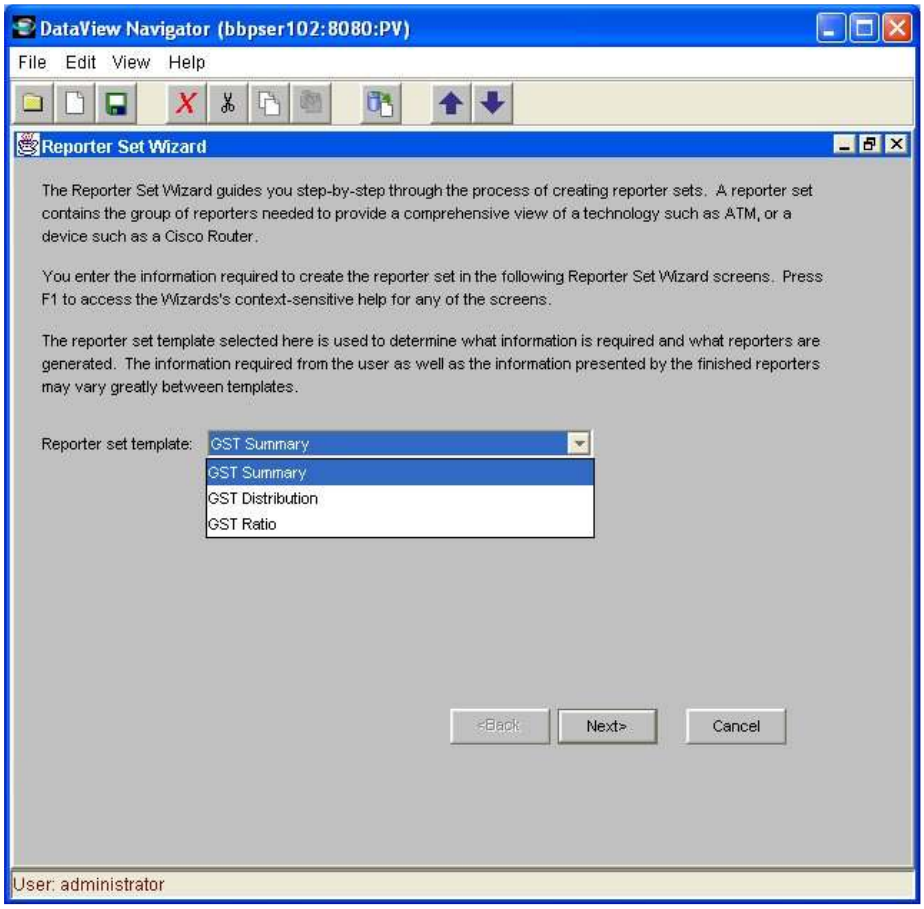
2. DataView Navigator -käyttöliittymä aukeaa.



3. Luo ja tallenna Navigator-ikkunassa uusi kansio, jonka haluat assosioida luotavan raporttiryhmän kanssa.

4. Valitse File-valikosta Reporter Set Wizard.

5. Määritä, mitä mallia käytetään raporttiryhmän pohjana Reporter Set Template-alasvetovalikosta ja klikkaa Next.



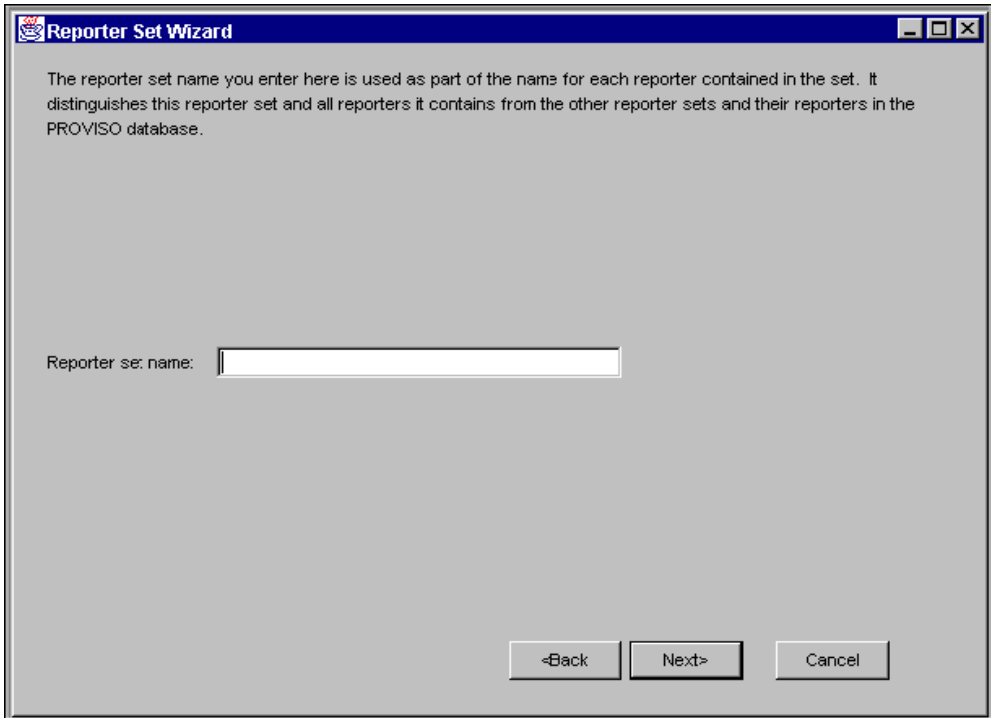
a. GST Summary sisältää perusraportit:

   i. GST (Group Summary Table), joka tarjoaa tiivistetysti tietoa resurssiryhmistä.
   ii. RST (Resource Summary Table), joka listaa yksittäisten resurssien arvot.
   iii. RTT (Resource Threshold Table), joka listaa resurssit, jotka ovat ylittäneet määritellyt raja-arvot
   iv. DC (Distribution Chart) esittää piirakkadiagrammin muodossa resurssien jakautumisen tietyllä vaihteluvälillä

b. GST Distribution esittää GST- ja RST -raportit jakaumadiagrammien avulla.

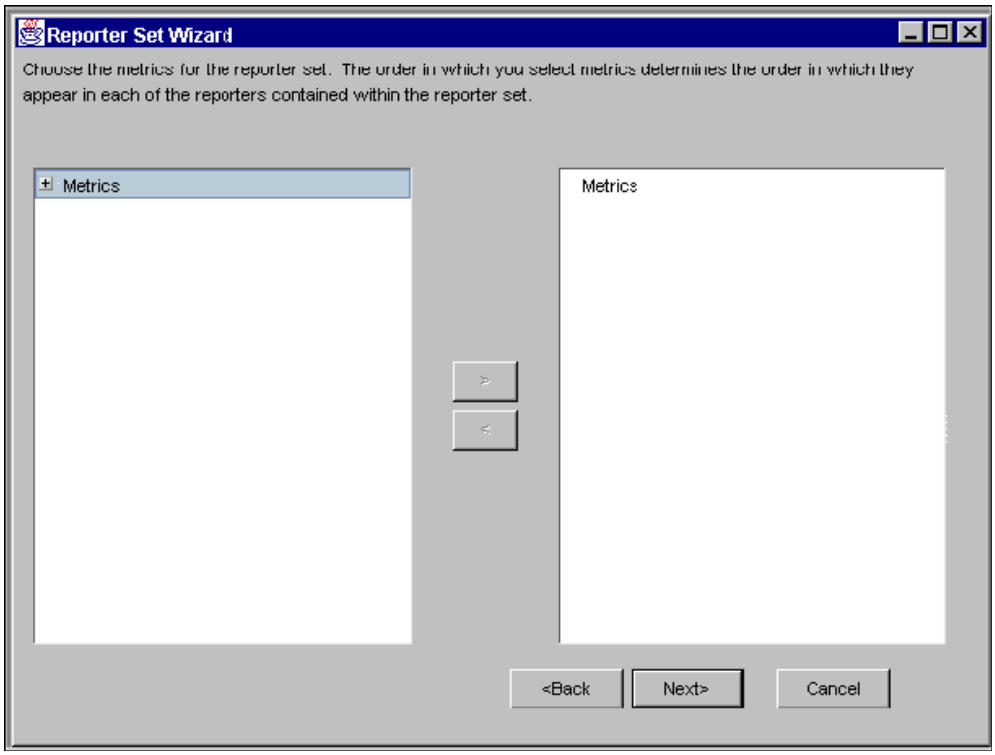
c. GST Ratio esittää GST- ja RST-raportit suhdediagrammien avulla.

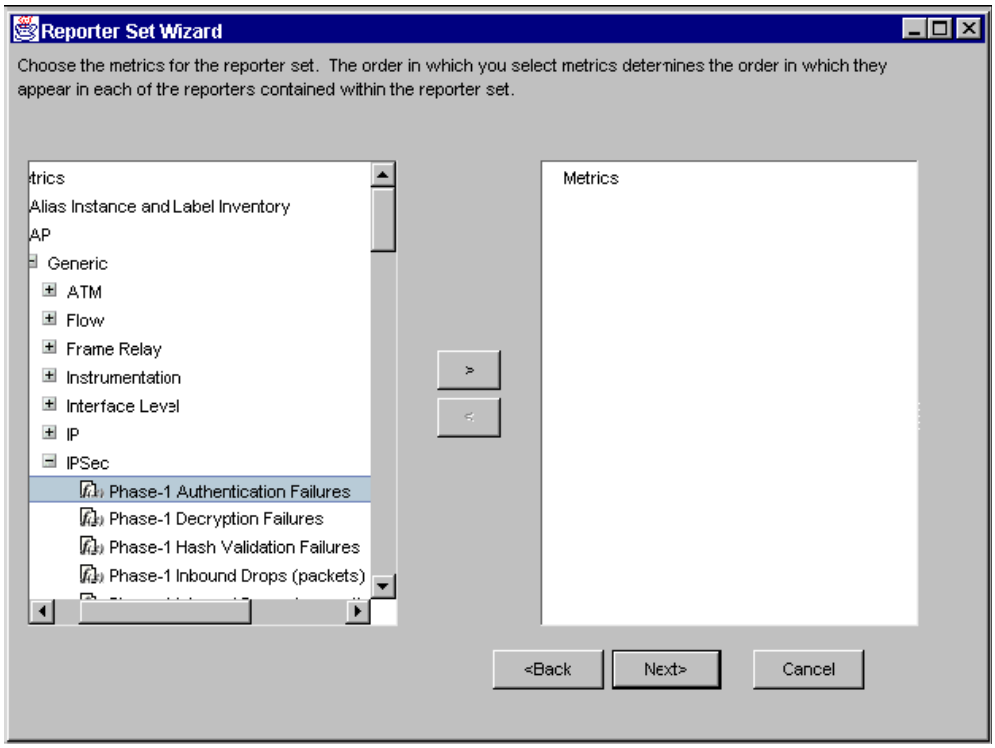
6. Anna raporttiryhmälle nimi ja klikkaa Next.



7. Seuraavaksi annetaan käytettävän mittasuureet (metric).

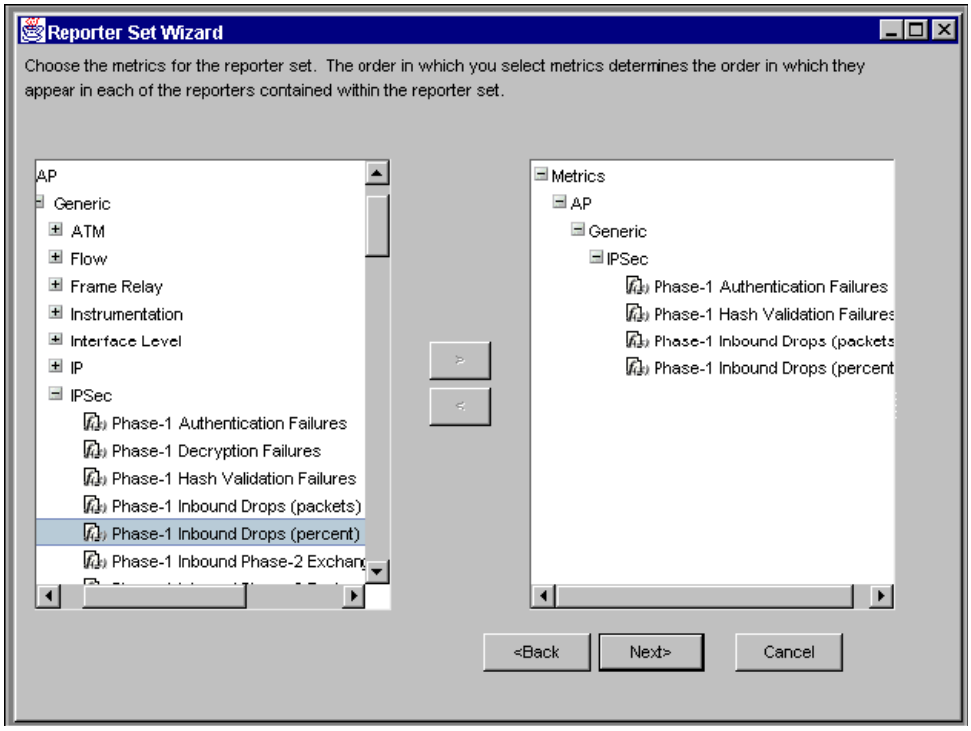
    a. Vasemmassa ikkunaruudussa klikkaa + -merkkejä laajentaaksesi näkymää.



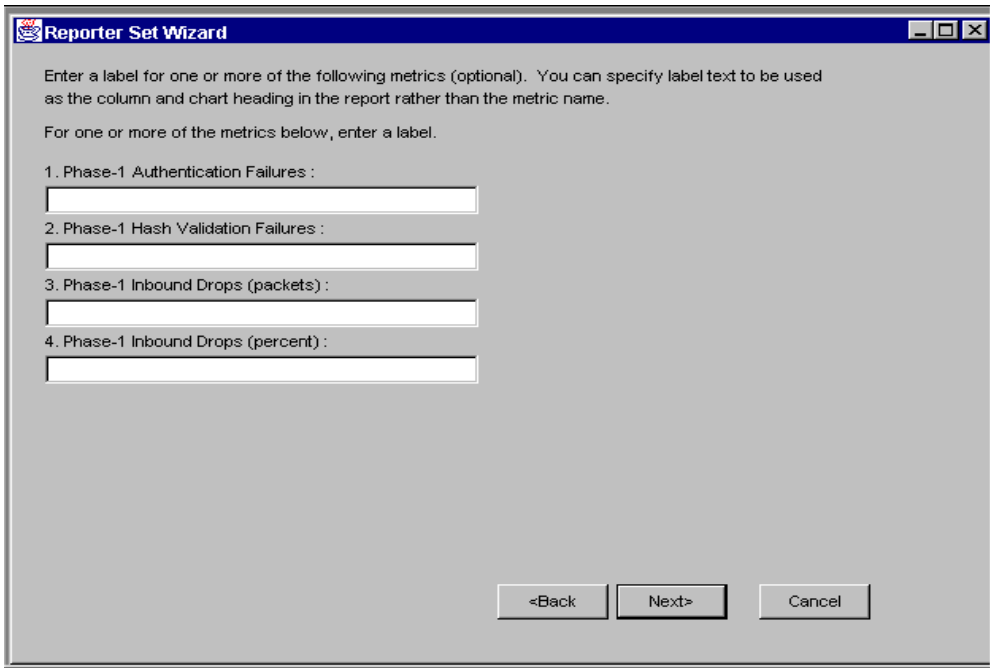


b. Klikka  -painiketta lisätäksesi mittasuureen valittujen listaan.

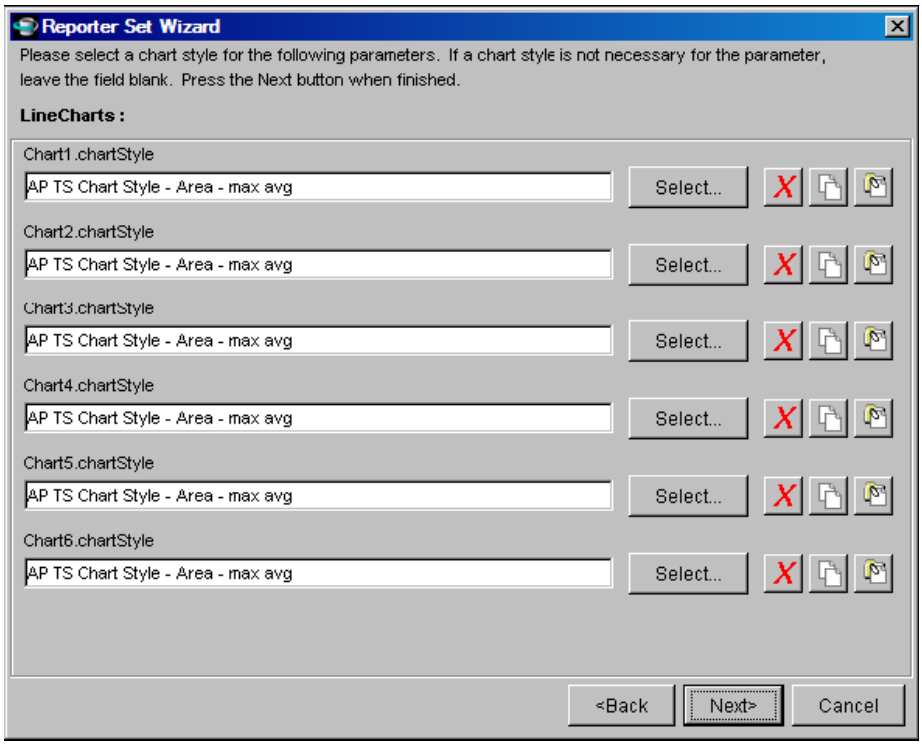
c. Toista menetelmä kaikille mittasuureille, jotka haluat mukaan raporteihin.



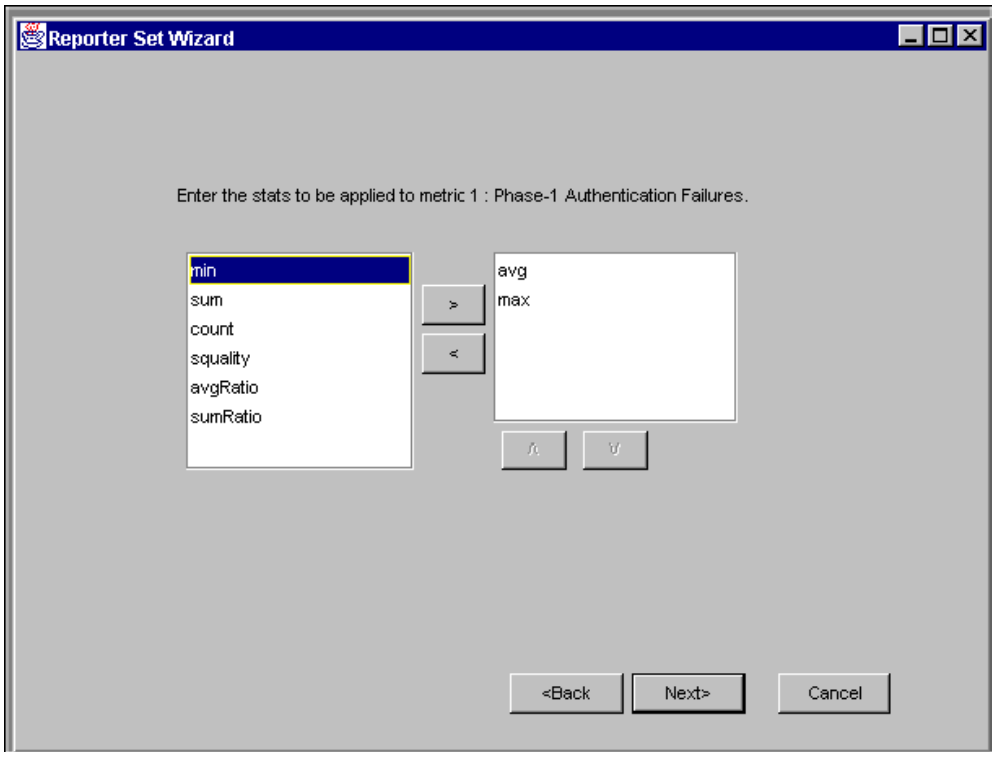
8. Seuraavassa ikkunassa valittujen mittasuureiden otsikot voidaan muuttaa haluttaessa. Jos otsikoita ei erikseen nimetä, käytetään mittasuureen nimeä.



9. Seuraavaksi valitaan lopullisen raportin kaavioiden ja diagrammien tyylit. Tämä tehdään valitsemalla jokaisen luetteloidun kaavion kohdalla Select-painiketta ja valitsemalla haluttu kaaviotyyli.



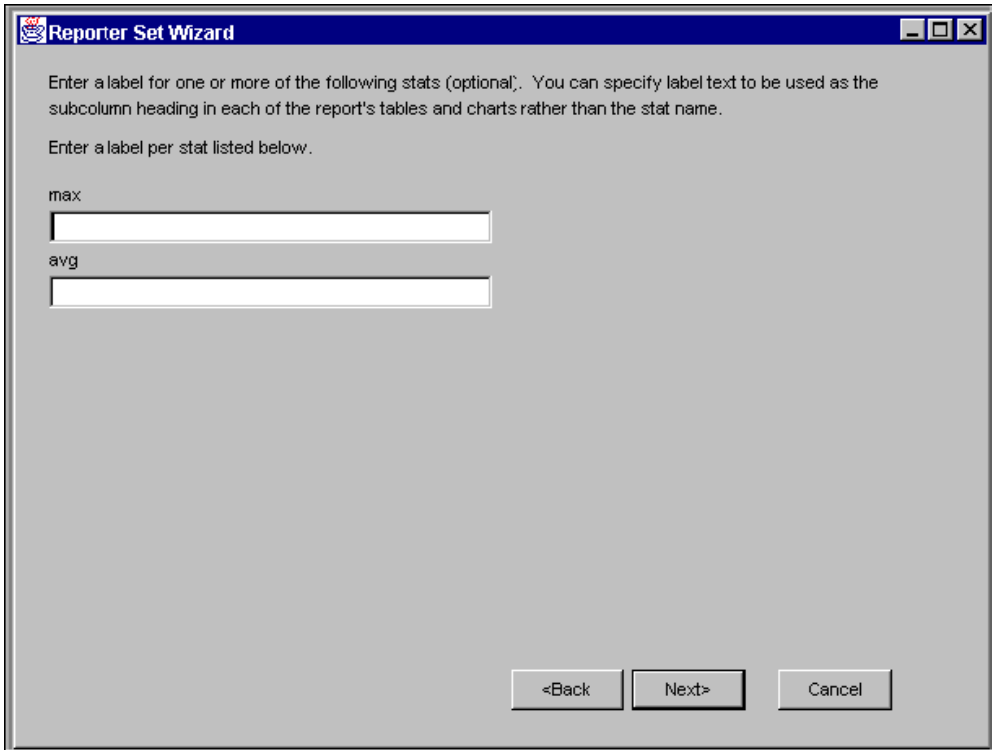
10. Seuraavaksi mittasuureille valitaan statistiikat (stats), joita ovat mm. keskiarvo (avg), summa (sum) ja lukumäärä (count). Statistiikat pitää valita jokaiselle mittasuureelle erikseen, jos esim. valittiin 5 mittasuuretta, määritellään ensin statistiikat ensimmäiselle suureelle, painetaan next, määritellään statistiikat toiselle suureelle, painetaan next ja niin edelleen.



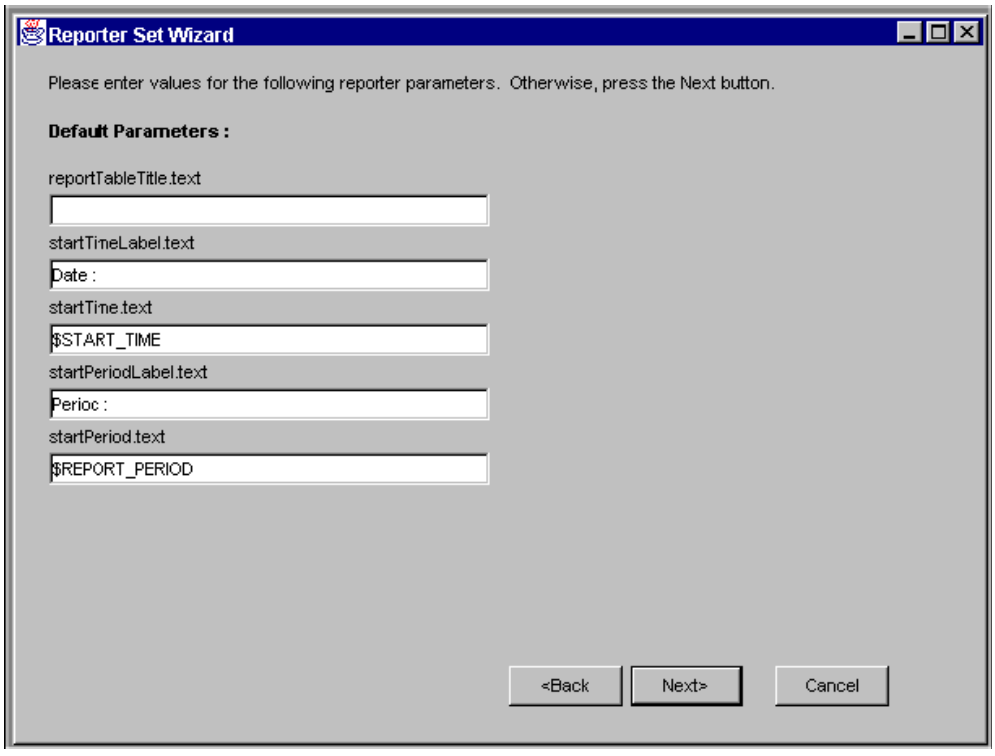
a. Valitaan statistiikka vasemmasta laatikossa ja lisätään se mukaan [>] painikkeella.

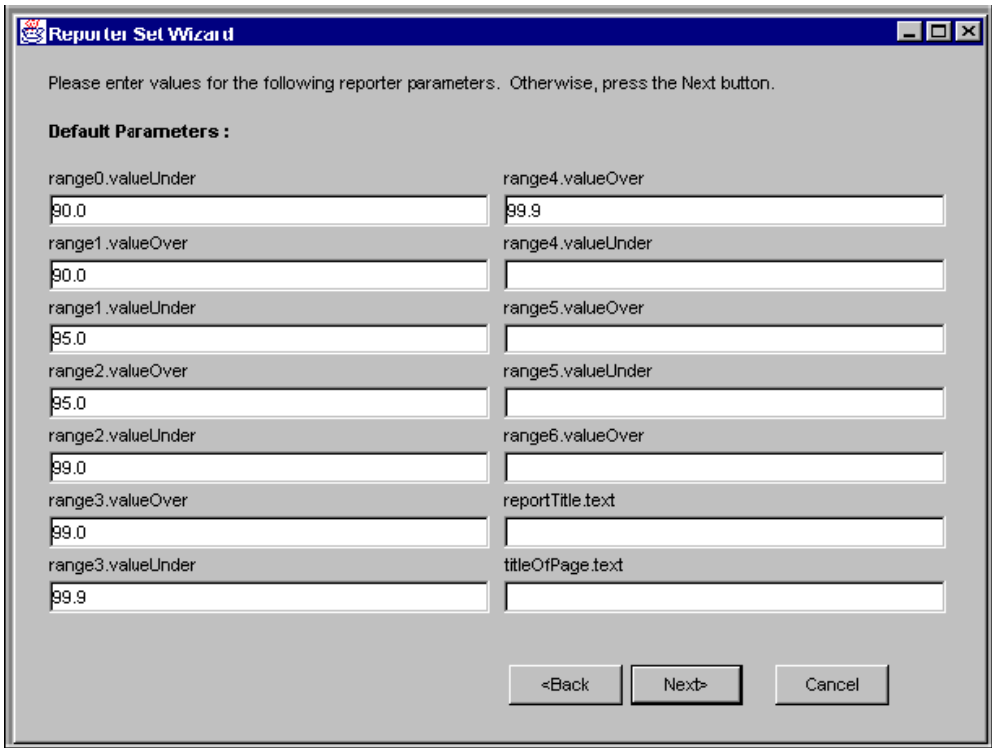
b. Kun halutut statistiikat on valittu ja ne näkyvät oikeanpuoleisessa laatikossa, voidaan niiden järjestystä muuttaa [/\ ] ja [\/] painikkeilla.

11. Kun statistiikat on valittu, pitää niille valita vielä otsikot. Syötä otsikot riveille ja paina next.

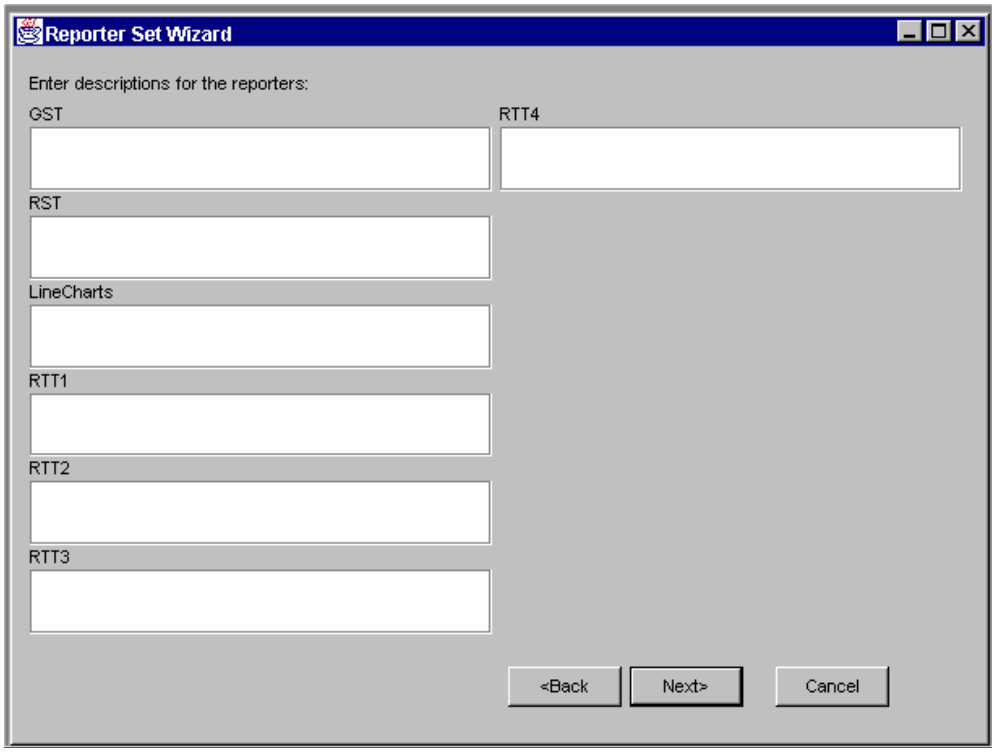


12. Seuraavassa vaiheessa lisätään raporttien parametrien arvoja. Kaikkien parametrien arvoja ei pysty määrittelemään. Ne, joita pystyy, arvokseen merkkijonoja.





13. Viimeisessä vaiheessa raporteille lisätään kuvaukset.



14. Lopuksi Reporter Set Wizard-työkalu koostaa yhteenvedon luoduista raporteista.