TAMPEREEN
AMMATTIKORKEAKOULU

LIIKETALOUS

OPINNÄYTETYÖRAPORTTI

SNMP-Verkonvalvonta vapaan lähdekoodin ohjelmilla

Tuomas Nikka

Tietojenkäsittelyn koulutusohjelma
Joulukuu 2007
Työn ohjaaja: Harri Hakonen

TAMPERE 2007

TAMPEREEN AMMATTIKORKEAKOULU

LIIKETALOUS

Tekijä(t): Tuomas Nikka

Koulutusohjelma(t): Tietojenkäsittely / Tietoverkkopalvelut

Opinnäytetyön nimi: SNMP-Verkonvalvonta vapaan lähdekoodin ohjelmilla

Title in English: SNMP Network Monitoring with Open Source Programs

Työn valmistumis-kuukausi ja -vuosi: 12 / 2007

Työn ohjaaja: Harri Hakonen **Sivumäärä:** 56

TIIVISTELMÄ

Simple Network Management Protocol eli SNMP on yleisesti tietovekkojen hallintaan ja valvontaan tarkoitettu protokolla. Tarkoituksenani on perehtyä protokollan toimintaan, eri versioihin ja ominaisuuksiin. Lisäksi tutkin vapaan lähdekoodin ohjelmistotarjontaa verkonvalvonnan osalta. Tarkoituksenani on tutkia pääasiassa kolmea SNMP:tä käyttävää ohjelmaa, MRTG, RRDtool ja Cacti, ja päättää mikä näistä on paras vaihtoehto suuren verkkoympäristön valvontaan. Tämän jälkeen toteutan ohjelman käyttöönoton Stora Enson Imatran Tehtaiden tietoverkkoympäristössä.

Tällä hetkellä Imatran Tehtailla ei ole käytössä mitään ohjelmaa verkkolaiteiden kuormituksen reaaliaikaiseen valvontaan. Olennaisia valvottavia tietoja ovat muun muassa kytkinten ja reitittimien prosessorin ja muistin käyttöaste, lämpötila, ja ainakin tärkeimpien linkkien liikenne sekä mahdolliset virheet. Mainitsemillani ohjelmilla näistä ja monista muista asioista voidaan reaaliajassa piirtää kuvaajia ja tallentaa ne pitkällekin ajalle. Näin pystytään kartoittamaan mahdollisia pullonkauloja verkkoliikenteessä ja muodostamaan yleinen kuva verkon kuormituksesta ja suorituskyvystä. Käytännössä tämä kuitenkin tulee kyseeseen pitemmällä aikavälillä, kun ohjelma on ollut käytössä esimerkiksi useamman kuukauden tai vuoden ajan.

Esittelen työssäni edellä mainitut ohjelmat, käyttöönoton, käytön ja tärkeimmät ominaisuudet. Tämän jälkeen päätän, mikä ohjelma olisi paras ratkaisu Imatran Tehtaille, jossa verkkolaitteita on yhteensä noin 260. Kaikkia näistä ei toki ainakaan tässä vaiheessa ole tarpeellista valvoa. Kaikki ohjelmat toimivat sekä Linuxilla, Unixilla että Windowsilla, mutta suoritan käyttöönoton työnantajan toiveen mukaisesti nimenomaan Linuxilla. Koska sekä käyttöjärjestelmä että ohjelmat ovat vapaan lähdekoodin myötä ilmaisia, on kokonaisratkaisukin ilmainen.

Käytännössä paras vaihtoehto selvisi jo varsin aikaisessa vaiheessa. Ohjelman käyttöönotto sujui varsin hyvin, ja ajoin ohjelmaa testiajossa noin kaksi viikkoa ennen kun asensin lopullisen version. Työssä on myös lyhyehkö opas peruskäyttöön. Tein myös muutamia laiteprofiileja yleisimmille laitteille, joka mahdollistaa uusien laitteiden helpon lisäämisen valvottavien laitteiden joukkoon.

Ohjelmia on myös mahdollisuus laajentaa ja niillä voi käytännössä valvoa lähes mitä tahansa numeerista tietoa. Tein kuitenkin ajan puutteen takia vain sen, mitä työnantaja asetti tavoitteeksi. Mielestäni nämä tavoitteet onnistuivat kohtalaisen hyvin. Ympäristön vaatimuksista johtuen jätimme myös muutamia olennaisia asioita tekemättä.

HUOM! Työnantajan vaatimuksena oli, että kaikkia tietoverkon laitteiden IP-osoitteita, DNS-nimiä sekä SNMP-yhteisötietoja käsitellään luottamuksellisina tietoina. Siksi olen poistanut kaikki tällaiset tiedot työstä, sekä kaikki tiedot josta voisi päätellä mitään niihin liittyvää. Niinpä kaikki työssä esiintyvät IP-osoitteet, DNS-nimet, SNMP-tiedot ja muut vastaavat tiedot ovat keksittyjä.

Avainsanat: SNMP verkonvalvonta vapaan lähdekoodin ohjelmat Linux Cisco

TAMPERE
UNIVERSITY OF APPLIED SCIENCES

INTERNATIONAL BUSINESS

Author(s): Tuomas Nikka

Degree programme(s): Data Networks / Business Information Systems

Title in English: SNMP Network Monitoring with Open Source Programs

Title in Finnish: SNMP-Verkonvalvonta vapaan lähdekoodin ohjelmilla

Month and year: December 2007

Supervisor: Harri Hakonen **Pages:** 56

ABSTRACT

Simple Network Management Protocol is a protocol that is generally used for network management and monitoring. My intention is to make myself familiar with the functioning, different versions and characteristics of the protocol. I will also look at some open source programs designed for network monitoring. My main interest is three programs, MRTG, RRDtool and Cacti. I decide which of these would be the best choice for a large network, and then implement it on the Stora Enso Imatra Mills network environment.

At the moment, there is not a program in use for monitoring a load of network devices. Essential information to monitor on switches and routers includes usage levels of processor and memory, device temperature and amount of traffic in at least most important links, as well as possible errors. The programs mentioned above can graph these and many other things in real time, and store the information in archive files for quite long times. With the help of such programs, one can map possible bottlenecks on a network traffic and formulate a general picture about a network load and performance. In practice, however, this comes in account when the program has been logging the information longer time, like some months or even a year.

I will introduce these programs, their implementation, basic usage and the most important features. Then I will decide the best for the needs of the Imatra Mills, which has about 260 network devices in their network. But not all of these should be monitored, at least not yet. All the programs can be run in a Linux, Unix and Windows, but I am going to do the implementation on the Linux platform, according to the employer's wishes. As both the programs and the operating system are free, being open source, so is the whole solution.

It was very soon quite clear which program is the best. The implementation phase went quite well, and I ran a two week-long test run before installing the final spec version. This thesis also includes a short guide for a basic usage, and I made some device profiles for the most common devices, which makes the easy addition of new devices possible.

These programs can be extended in many ways, and one can use them to monitor and graph practically any numeric information. However, I only made what the employer set as a target. In my opinion, these goals were completed quite well. Because the requirements of the environment, I also had to leave out some essential things.

NB The employer demanded that all IP addresses, DNS names and SNMP community information of the network should be considered as confidential information. Therefore I have deleted all such information as well as all the information of which anything about them can be reasoned. Also, all the IP addresses, DNS names and SNMP information in the thesis have been made up.

Key words: SNMP network monitoring open source programs Linux Cisco

Sisällysluettelo

Käsitteet

SNMP – Simple Network Management Protocol: Yleisesti verkonhallintaan käytetty protokolla. SNMP:stä on versiot 1, 2 ja 3, joista viimeisin eli SNMPv3 on tietoturvaominaisuuksiensa ansiosta suositeltavin käyttää.

SMI – Structure of Management Information: Eräänlainen skeema, joka määrittää millainen SNMP-tietokannan hierarkinen rakenne on. SMI:stä on kaksi versiota, 1 ja 2, joista jälkimmäinen on laajempi, ja tehtiin vastaamaan SNMPv2:n parannuksia.

OID – Object Identifier: SMI:n määrittelemien tietoalkioiden tunnisteet. Nämä esitetään SMI:ssä numerosarjoina, esimerkiksi 1.3.6.1.4.1.9, mutta ymmärrettävyyden takia niille on määritelty myös selkokieliset vastineet kuten iso(1).org(3).dod(6).internet(1).mgmt(2).

MIB – Management Information Base: Hallinnallinen laitteella sijaitseva tietokanta, jossa olevia tietoja SNMP-protokollalla voidaan lukea ja muokata. Yleisten MIB:ien lisäksi eri valmistajilla on useita omia erilaisia laitekohtaisia MIB:ejä.

NMS – Network Management System: Hallintajärjestelmä, jonka kanssa SNMP-laitteet kommunikoivat. Tämä on tyypillisesti ohjelmisto, mutta myös itse tehtyjä SNMP:tä käyttäviä skriptejä voidaan pitää eräänlaisina NMS:inä.

RMON – Remote MONitoring: Agenttiin rakennettu luotain, joka tekee kyselyitä halutuin väliajoin laitteen MIB:lle. Tätä kutsutaan sisäiseksi kyselyksi.

RRD – Round Robin Database: RRDtoolin käyttämä tietokantatyyppi, jonka koko määritellään etukäteen. Tietyltä aikaväliltä otetuista näytteistä otetaan arvoltaan joko suurin, pienin, tai keskiarvo, joka tallennetaan RRA:han. Kun tila on käytetty, kirjoitetaan vanhojen tietojen päälle.

RRA – Round Robin Archive: RRDtoolin luoma arkistotiedostotyyppi kuvaajia varten. Oletuksena tähän on määritelty 5 minuutin aikavälillä päivittäinen, 30 minuutin välillä viikottainen, 2 tunnin välillä kuukausittainen, ja päivän välillä vuosittainen arkisto.

1 Johdanto

Tietoliikenneverkot ovat tärkeä pohja nykypäivän yritysten toiminnalle, sillä lähes kaikki tietotekniset järjestelmät toimivat niissä. Suurilla yrityksillä on usein maailmanlaajuisia tietoliikenneinfrastruktuureja, jotka yhdistävät eri puolilla maailmaa sijaitsevat toimipisteet yhdeksi suureksi loogiseksi tietoverkoksi. Näin muun muassa tiedonsiirto, IP-puhelut, videoneuvottelut ja yhteistyö onnistuvat nopeasti ja helposti työntekijöiden fyysisestä sijainnista riippumatta.

Yritykset investoivat valtavia summia tietoverkkoihinsa. Nämä koostuvat sekä omasta sisäverkosta ja sen laitteista (LAN), että eri toimipisteitä yhdistävistä linkeistä, jotka tyypillisesti hankitaan vuokraamalla tarvittavan nopea yhteys (WAN) tietoverkkooperaattoreilta. Lisäksi tietoverkkoihin kuuluu vielä muun muassa käyttäjäkohtaiset VPN-yhteydet, joilla voidaan ottaa salattu yhteys yrityksen sisäverkkoon mistä tahansa paikasta, josta pääsee internetiin.

Liikenne tietoverkoissa kasvaa jatkuvasti, ja samalla kasvaa myös tarve entistä nopeammille tietoverkoille. Laitteistoa täytyy myös uusia tehokkaammaksi ja uusia nopeampia verkkostandardeja tukevaksi. Toisaalta, jos verkko on liian nopea, voi suuri osa sen mahdollistamasta tiedonsiirtonopeudesta jäädä hyödyntämättä. Tällöin investointi ei vastaa tarvetta, ja rahaa menee hukkaan. Toisaalta verkon tulisi olla myös skaalautuva, jolloin sitä on helppo parantaa ja laajentaa tarvittaessa. Siksi yrityksille on tärkeää tietää nykyisen verkon ja sen laitteiden kuormitustilanne, jolla voidaan löytää verkossa mahdollisesti sijaitsevia pullonkauloja, muodostaa trendejä ja niiden perusteella yleinen kuva verkon kuormituksesta eri aikoina, niin sanottu baseline.

Tämä opinnäytetyö sai alkunsa Stora Enson Imatran Tehtaiden Sähkö- ja Automaatio-osaston tarpeesta saada edullinen, skaalautuva ja helppokäyttöinen verkonvalvontajärjestelmä, jolla voisi valvoa Imatran Tehtaiden verkon noin 260 aktiivilaitetta, niiden eri komponenttien ja linkkien kuormitusta, sekä muita numeerisia tietoja. Järjestelmän avulla saatava data tallennettaisiin myös myöhempää käyttöä ja analysointia varten. Lisäksi järjestelmään kuuluvat myös analysointityökalut, joilla kuormitustilanteesta voidaan piirtää erilaisia kuvaajia. Tavoitteenani on tutkia muutamaa vapaan lähdekoodin ohjelmaa, jotka mahdollistavat tämän käyttäen SNMP-protokollaa, ja valita näistä paras vaihtoehto Imatran Tehtaille. Tämän jälkeen toteutan valitsemani ohjelman käyttöönoton.

2 Kohdeyrityksen esittely

2.1 Stora Enso Oyj

Stora Enso Oyj on yksi maailman suurimmista metsäteollisuusyhtiöistä. Yhtiöllä on maailmanlaajuisesti noin 44000 työntekijää, joista Suomessa työskentelee noin 12000. Yhtiö toimii viidellä mantereella yli 40 maassa. Vuonna 2006 liikevaihto oli 14,6 miljardia euroa. Stora Enson osakkeet on listattu Helsingin, Tukholman ja New Yorkin pörsseihin.

Stora Enson päätuotteita ovat muun muassa erilaiset paperit, kuten korkealaatuinen hienopaperi, sanomalehtipaperi ja aikakauslehtiin käytettävä paperi. Muita tuotteita ovat erilaiset pakkausmateriaalit, kartonki, sekä muut puutuotteet. Yhtiön tuotantokapasiteetti on vuositasolla 16,5 miljoonaa tonnia paperia ja kartonkia, sekä 7,4 miljoonaa kuutiometriä muita puutuotteita (Stora Enso.)

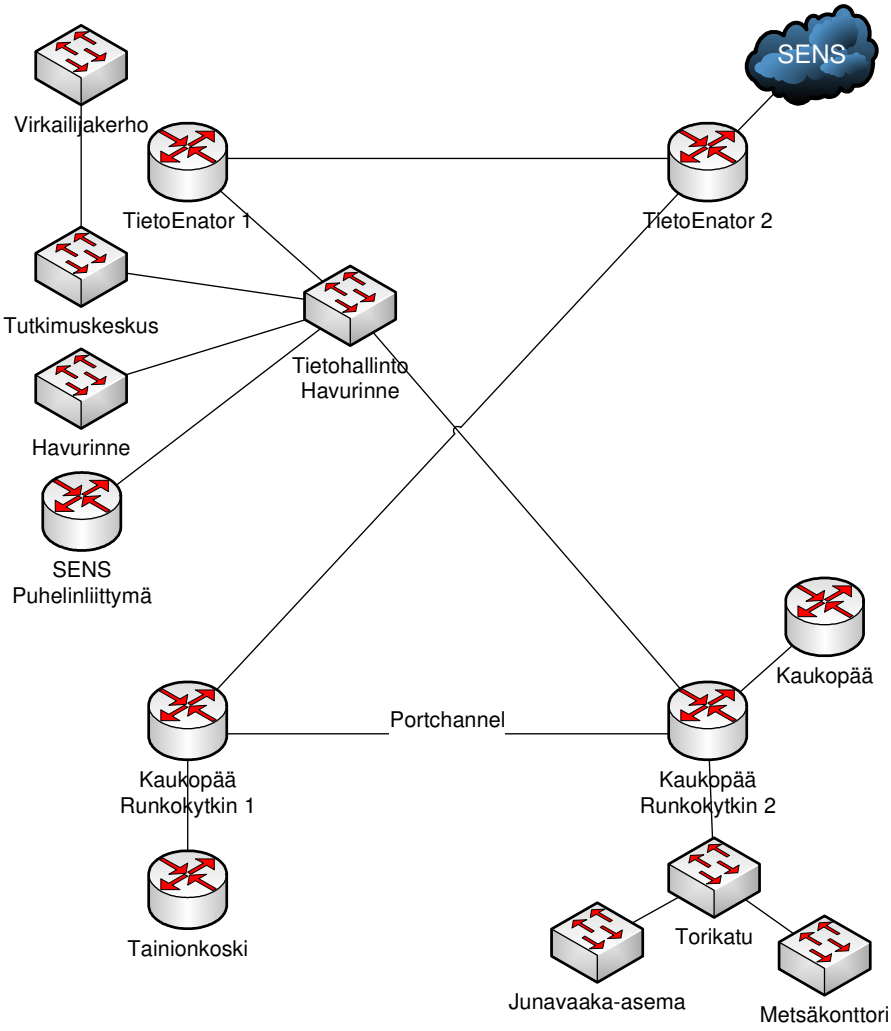
2.2 Imatran Tehtaat

Imatran Tehtaisiin kuuluu Kaukopään ja Tainionkosken tehtaat (kuva 2.1), sekä tutkimuskeskus, metsäkonttori ja Corenso Oy. Imatran Tehtaiden päätuotteita ovat erilaiset kartongit ja paperit. Vuosituotantokapasiteetti on noin 1,5 miljoonaa tonnia, josta noin 80 % on kartonkituotteita. Imatran Tehtaiden tuotteista yli 90 % menee vientiin, tärkeimpänä markkina-alueena Eurooppa. Imatran Tehtailla työskentelee noin 2600 työntekijää (Stora Enso.)

Kuva 2.1. Stora Enson Imatran Tehtaat, yläkuvassa Kaukopään ja alakuvassa Tainionkosken tehdas

2.3 SENS Imatran Tehtailla

SENS eli Stora Enso Network Services on Stora Enson maailmanlaajuinen tietoliikenneverkko. Pelkästään Imatran Tehtaiden kampusalueella SENS kattaa noin 260 verkkolaitetta, pääasiassa kytkimiä. Sekä Kaukopään että Tainionkosken tehtailla on molemmissa oma ATK-hallintakeskus, joissa runkokytkimet sijaitsevat. Osa verkosta on TietoEnatorin hallinnassa. Kuvassa 2.3 on Imatran Tehtaiden runkoverkon tärkeimmät laitteet.



Kuva 2.3 Imatran Tehtaiden kampusalueen runkoverkko

Imatran Tehtailla on CiscoWorks -ohjelmisto käytössään, mutta tämä on tarkoitettu nimenomaan verkkolaitteiden hallintaan. CiscoWorksin avulla saa laitteista monenlaisia raportteja lähinnä taulukkomuodossa, mutta varsinaisia graafisia kuvaajia laitteiden kuormituksesta sillä ei saa reaaliajassa piirrettyä. Siksi he tarvitsevat myös tähän tarkoitukseen soveltuvan ohjelman.

3 SNMP-protokolla

SNMP-protokolla kehitettiin 1988, kun tietoverkkojen kasvaessa huomattavaa vauhtia ilmeni tarvetta keskitetylle verkonhallinnalle. Alun perin hyvinkin yksinkertainen protokolla on sittemmin kehittynyt, ja nykyään SNMP:llä voidaan etähallita ja valvoa paitsi reitittimiä ja kytkimiä, myös lähes mitä tahansa muita verkossa olevia laitteita, kuten tulostimia, palvelimia ja jopa yksittäisiä työasemia. Esimerkiksi tulostimienhallinnan puolella ainakin Xeroxilta ja Kyoceralta on saatavilla ilmainen tulostinten hallintaan tarkoitettu ohjelma, jolla voi hallita minkä tahansa merkkisiä tulostimia. Myös nämä ohjelmat käyttävät SNMP:tä.

3.1 Protokollan toiminta

SNMP käyttää tiedonsiirtoon yhteydetöntä UDP-protokollaa toimien TCP/IP-mallin sovellustasolla. Protokollaa kehitettäessä päädyttiin UDP:hen, koska SNMP on tarkoitettu sopimaan todella suuriin verkkoihin, ja TCP:n tiedon perillemenon varmistuskeinoja pidettiin ylimääräisenä ja turhana verkon ja laitteiden kuormituksena. UDP:tä käyttämällä esimerkiksi yhteyden katkeaminen havaitaan yksinkertaisella viiveellä, ja verkon kuormitus päättyy tähän. SNMP käyttää UDP-porttinumeroa 161 käskyjen lähettämiseen ja vastaanottamiseen, ja porttia 162 trap- ja muiden ilmoitusten vastaanottamiseen hallittavilta laitteilta (Mauro & Schmidt 2005: 16.)

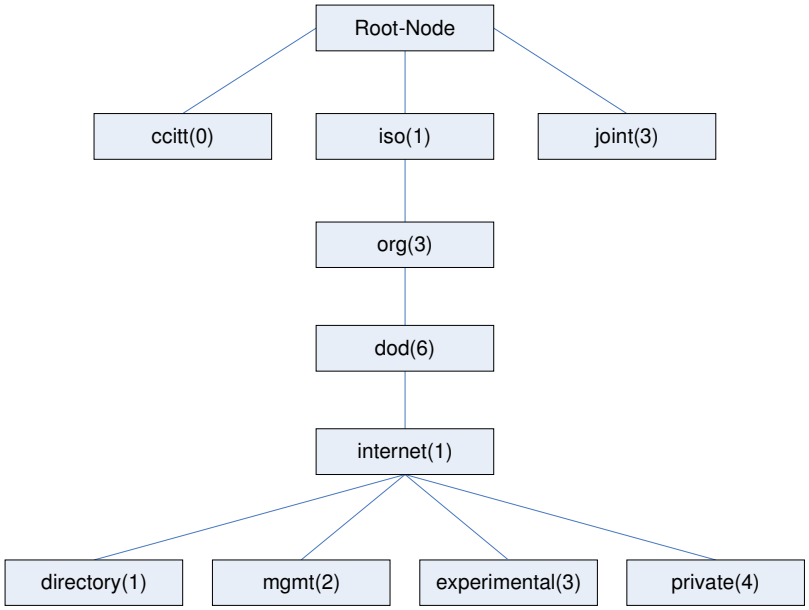
SNMP:n käyttämä tieto määritellään SMI:n (Sturcture of Management Information) mukaisesti. Tämä on eräänlainen skeema, joka määrittelee tiedon esitysmuodon. SMI on puumainen hierarkinen tietorakenne, joka sisältää itse asiassa muutakin kuin vain SNMP:n vaatimat tiedot. Hallintaobjektit voidaan jakaa kolmeen attribuuttiin:

- Nimi eli OID (Object IDentifier): tämä esitetään numeroina, mutta selkokielisyyden takia näille on annettu myös helpommin ymmärrettävät nimet, vähän samaan tapaan kuin DNS-nimet, jotka vastaavat tiettyjen laitteiden IP-osoitteita verkoissa.

- Tyyppi ja syntaksi: SMI käyttää laitteistoriippumatonta ASN.1:sta eli Abstract Syntax Notation One, joka mahdollistaa tiedonsiirron minkä tahansa SNMP:tä tukevien laitteiden ja ohjelmistojen välillä alustasta riippumatta.

- Koodaus jakaa yhden objekti-instanssin BER-sääntöjen (Basic Encoding Rules) mukaisesti oktetteihin.

Objektien nimet muodostavat kuvassa 3.1 näkyvän puumaisen rakenteen. Nimien perässä on niitä vastaavat OID-numerot.



Kuva 3.1. Osa SMI-rakennepuuta (Mauro & Schmidt 2005: 24)

Näistä nimenomaan iso eli 1 ja siitä lähtevät kuvassa näkyvät haarat ovat SNMP:n kannalta oleellisia. Kaksi muuta juuresta lähtevää haaraa, ccitt ja joint, eivät liity SNMP:hen lainkaan. Numeroiden mukaisesti mgmt:n OID on siis 1.3.6.1.2, nimettynä iso.org.dod.internet.mgmt. Muista haaroista mainittakoon vielä 4 eli private, joka on varattu eri laitevalmistajien omille SNMP-sovelluksille.

MIB (Management Information Base) on hallintatietokanta, jossa tieto laitteilla sijaitsee. Niissä olevia tietoja muokataan ja luetaan SNMP-protokollalla. MIB:ejä on lukuisia erilaisia, ja eri laitevalmistajat ovat luoneet lisäksi omia MIB:ejä. Esimerkiksi Cisco Systemsillä on satoja erilaisia laitekohtaisia MIB:ejä, jotka vastaavat heidän valtavan laitevalikoimansa toimintaa (Mauro & Schmidt 2005: 36.)

SNMP on perustoiminnaltaan varsin yksinkertainen protokolla. Verkossa täytyy olla NMS (Network Management System) eli verkonhallintajärjestelmä, jonka kanssa verkkolaitteissa oleva agentti kommunikoi. Tämä voi olla joko valtava verkonhallintaohjelmisto, kuten HP OpenView tai CiscoWorks, tai yksinkertai-

nen, itse tehty työasemalta ajettava skripti, tai lähes mitä tahansa siltä väliltä. Yleisesti käsitteellä NMS tarkoitetaan kuitenkin juuri OpenViewin tai CiscoWorksin tapaista laajaa ohjelmistoa. NMS:n ja agentin välille on määritelty seuraavanlaiset perusoperaatiot:

- Get: hae tietoalkio
- Getnext: hae seuraava tietoalkio, tunnetaan myös paremmin nimellä SNMPwalk
- Set: kirjoita uusi arvo
- GetResponse: hae virheilmoitukset

Yllä mainitut operaatiot toimivat nimenomaan NMS:stä laitteelle päin. Laitteet puolestaan aloittavat kommunikoinnin NMS:ään päin operaatiolla trap, joka on ilmoitus laitetapahtumasta. Versiossa SNMPv2 on määritelty näiden lisäksi myös muita operaatiota.

SNMP käyttää yhteisöjä (community) luottosuhteiden muodostamiseen. Laitteessa oleva SNMP-agentti voidaan konfiguroida kolmeen eri yhteisöön: vain luku, luku/kirjoitus, ja trap. Käytännössä yhteisönimiä voi ajatella salasanoina, joilla kontrolloidaan kolmea erityyppistä aktiviteettia. Näitä nimiä ei salata mitenkään, joten ne ovat paketinkaappaajalla suhteellisen helposti kaikkien selvitettävissä.

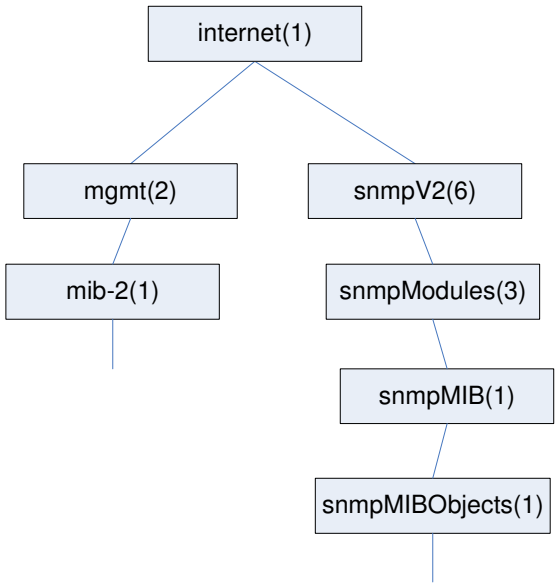
3.2 SNMPv2

Versio 2 on huomattavasti laajempi ja monipuolisempi kuin alkuperäinen SNMPv1. Uusia operaatioita ovat muun muassa seuraavat (Mauro & Schmidt 2005: 37):

- Getbulk: hakee suuren lohkon dataa kerrallaan, jonka ei tarvitse välttämättä olla yhtenäinen.
- Notification: versio 2:n generoima ilmoitus laitetapahtumasta. Tämä on standardoitu, toisin kun versio 1:n trap-ilmoitus.
- Inform: mekanismi trap-ilmoitusten vastaanoton kuittaamiseen. Samalla varmistuu ilmoituksen oikea vastaanottaja.

Lisäksi on vielä report-operaatio, jonka suunniteltiin olevan osa SNMPv2-standardia. Sitä ei kuitenkaan otettu standardiin mukaan, vaan se tuli lopullisesti käyttöön vasta versiossa 3. Sen tarkoitus on mahdollistaa eri SNMP-hallintajärjestelmien kommunikointi keskenään, lähinnä liittyen SNMP-viestien käsittelyongelmiin.

SNMPv2 luo SMI:hin uuden, nimensä mukaisen haaran OID-numerolla 6. Myös mgmt-haaraan tulee muun muassa mib-2, jonka alta löytyvät uudet hallintakomennot ja tiedot. Kuvassa 3.2 on SMIv2:na tunnettu hierarkkinen rakenne.



Kuva 3.2. SMIv2-rakennepuu, jatkoa kuvan 3.1 internet-haaraan (Mauro & Schmidt 2005: 33)

MIB:eihin viitattaessa ohjelmistoissa käytetään yleensä nimenomaan OID-numerotunnuksia, jotka voivat olla hyvinkin pitkiä. Esimerkiksi OID 1.3.6.1.2.1.25.1.5.6.0 kuvaa MRTG-ohjelman MIB:ssä muuttujaa hrSystemNumUsers. MIB:ejä voi myös luoda itse lisää, ja olemassa olevia voi laajentaa skripteillä.

3.3 SNMPv3

Uusin ja suositeltavin versio SNMP:stä on versio 3. Toiminnallisia muutoksia versiossa 3 ei ole muuten juuri lainkaan, mutta siinä on keskitytty edellisten versioiden suurimpaan heikkouteen – tietoturvaan. Yhteisönimet, eli käytännössä salasanat, välitettiin edellisissä versioissa selkotekstinä, jolloin kuka tahansa voi kaapata ne ja saada laitteet hallintaansa. SNMP v3:ssa on otettu käyttöön salasanan salaus sekä kunnon autentikointimekanismi.

Lisäksi terminologiaan on tullut melkoisia muutoksia. Versio 3:ssa on käsite SNMP-kokonaisuus (entity), joka jakautuu ko-

neistoon (engine), ja sovelluksiin. Koneisto jakautuu edelleen seuraaviin palasiin:

- Dispatcher: lähettää ja vastaanottaa viestit
- Message Processing Subsystem: käsittelee viestit lähetystä varten ja ottaa datan vastaanotetuista viesteistä talteen
- Security Subsystem: huolehtii autentikointi- ja yksityisyyspalveluista
- Access Control Subsystem: huolehtii käyttöoikeuksista

Sovelluksia puolestaan ovat muun muassa seuraavat:

- Command generator: generoi get, getnext, getbulk ja set -komennot ja käsittelee tulokset
- Command responder: vastaa yllä oleviin komentoihin
- Notification originator: generoi trap- ja notification-viestit
- Notification responder: ottaa vastaan trap- ja notification-viestit
- Proxy forwarder: välittää viestit eri entiteettien välillä

Vaikka muutos terminologiaan on melkoinen, versio 3 on toiminnaltaan täysin sama kuin aiemmissa versioissa, tietoturvaominaisuuksia lukuun ottamatta. Eri osaset on vain järjestelty uudestaan loogisemman kuuloisiksi kokonaisuuksiksi (Mauro & Schmidt 2005: 75).

Tietoturva on SNMPv3:ssa mahdollista kolmella tavalla:

- noAuthNoPriv: ei autentikointia eikä yksityisyyttä
- authNoPriv: autentikointi mutta ei yksityisyyttä
- authPriv: sekä autentikointi että yksityisyys

Autentikointi tapahtuu joko joko SHA1 (Secure Hash Algorithm 1) tai MD5 (Message Digest 5) -algoritmeillä. Ainoa tuettu kryptausalgoritmi tällä hetkellä on DES (Data Encryption Standard), mutta todennäköisesti ennemmin tai myöhemmin myös vahvemmat 3DES ja varsinkin AES (Advanced Encryption Standard) tulevat tuettujen algoritmien listalle.

3.4 RMON

Perinteisesti NMS lähettää poll- eli kyselyviestejä tietyin väliajoin niille laitteille, joita järjestelmällä halutaan valvoa. Tämä voi kuitenkin tuottaa suuren määrän liikennettä verkkoon, mikäli verkkolaitteita on paljon, ja kyselytiheys suuri. Näiden tasapai-

nottaminen on tärkeää, mutta tälle niin kutsutulle ulkoiselle kyselylle on myös vaihtoehto, joka vähentää verkkoliikennettä huomattavasti. Tämä vaihtoehto on RMON (Remote MONitoring), joka tekee luotaimen avulla sisäisiä kyselyitä laitteessa olevan agentin ja laitteen MIB-tietokannan välillä, käsittelee laitteelta saamiaan tietoja ja ilmoittaa trap-viesteinä halutuista asioista NMS:lle. Tämä vähentää sekä verkkoliikennettä että NMS:n kuormitusta.

RMON on MIB, josta on olemassa versiot RMON1 ja RMON2. Versio 1 keskittyi OSI-mallin kahden alimman kerroksen tietoihin Ethernet- ja Token Ring-verkoissa. Versiossa 2 tukea laajennettiin aina sovelluskerrokselle asti. RMON löytyy sisäänrakennettuna muun muassa useimmista Ciscon kytkimistä ja reitittimistä.

3.5 SNMP-protokollan käyttöönotto Ciscon laitteissa

SNMP-protokollan peruskäyttöönotto Ciscon kytkimillä ja reitittimillä on melko yksinkertainen toimenpide. Ensinnäkin laitteelle konfiguroidaan yhteisönimi, jotta laitteisiin saadaan SNMP-yhteys (Welscher 1999.) Tämä tapahtuu seuraavalla komennolla terminaalin konfigurointitilassa:

snmp-server community **community-string** ro|rw|view

Community-string on yhteisönimi ja ro|rw|view oikeudet, mitä kyseisen yhteisön kautta voidaan laitteelle tehdä. Ro tarkoittaa read-only, rw read/write ja view vain MIB:lle määritelty ja tarpeen mukaan rajattu näkymä. Oikeuksia voidaan tarvittaessa rajoittaa myös pääsylistoilla. Reitittimillä ja 3-tason kytkimillä voidaan myös esimerkiksi estää reititystaulun lukeminen SNMP:n avulla.

Muita perustoiminnan kannalta hyödyllisiä asetuksia on trap-viestien päälle kytkeminen. Tämä tehdään komennolla

snmp-server enable traps

Tämä oletuskomento kytkee kaikki mahdolliset trap-viestit päälle. Ne voidaan myös määritellä tarkemminkin, kuten protokollakohtaisesti. Ylipäätäänkin kaikki SNMP-konfiguraatiot tehdään snmp-server -alkuisilla komennoilla. Nämä saa näkyviin konfigurointitilassa komennolla

snmp-server ?

Toinen komento konfigurointitilassa on snmp josta voidaan luoda mib-tauluja ja määritellä niiden koko. Nämä optiot kyseiselle komennolle saa näkyviin komennolla

```
snmp ?
```

4 SNMP:tä hyödyntävät ohjelmistot

4.1 MRTG

Zürichin teknillisen korkeakoulun tietoverkkopäällikkö Tobias Oetikerin jo vuodesta 1995 alkaen kehittämä MRTG (Multi Router Traffic Grapher) on yksi suosituimpia vapaan lähdekoodin SNMP:tä hyödyntäviä ohjelmia. Se on lisensoitu GNU GPL:n mukaisesti ja on ilmainen.

MRTG koostuu käytännössä Perl-skriptistä, joka lukee laitteelta SNMP-tiedoista linkkien kuormituslaskureita, sekä C:llä tehdystä osuudesta, joka kirjaa nämä tiedot ylös ja piirtää niistä kuvaajia, joita voi katsoa millä tahansa www-selaimella. Oletusnäkymänä on yksityiskohtainen päivänäkymä, jonka lisäksi ohjelma generoi näkymät viikon, kuukauden ja vuoden aikaväliltä. Vaikka tiedon vaatiman tilan määrä ei kasva, se mahdollistaa kahden vuoden kuormitustietojen tallennuksen tehokkaasti (Oetiker 2007: MRTG.)

MRTG skaalautuu keskitason Linux/Unix-koneella helposti yli 200 linkin valvontaan. Sillä ei kuitenkaan tarvitse valvoa pelkkiä verkkolaitteiden linkkejä, vaan halutessa sillä voidaan valvoa myös muita numeerisia tietoja, käytännössä pääasiassa erilaisia laskureita.

4.1.1 Asennus

Monien muiden SNMP:tä hyödyntävien sovellusten tapaan myös MRTG on kirjoitettu pääasiassa Perlillä. Se toimii joko Linuxilla, Unixilla, Novell Netwarella tai Windows NT:llä. Itse testasin ohjelmaa Debian 4.0 Linuxilla. Ohjelman lähdekoodin voi hakea MRTG-sivustolta http://oss.oetiker.ch/mrtg/, ja se täytyy kääntää konekielelle ennen käyttöönottoa.

Linux-versio MRTG:stä vaatii seuraavat kirjastot toimiakseen:

- C-kääntäjä, kuten GNU C Compiler (GCC), MRTG:n lähdekoodin kääntämistä varten
- Perl, vähintään versio 5.005 tai SNMPv3:sta käytettäessä vähintään 5.8. Uusin versio on tällä hetkellä 5.8.8
- GD library, vaaditaan kuvaajien piirtämiseen
- Libpng, jotta GD library voi tuottaa PNG-muotoisia kuvia
- Zlib, jotta Libpng voi pakata PNG-kuvat
- MRTG, uusin vakaa versio on 2.15.2

Helpoin tapa asentaa kirjastot on etsiä ne normaalin päivitysohjelman avulla (kuten apt-cache search **libpng** jne.) ja sen jälkeen hakemalla ja asentamalla niitä vastaavat paketit (apt-get install **libpng** jne.) yksitellen (Tobias Oetiker 2007: MRTG Unix Guide.) Windows-versio MRTG:stä vaatii ActiveState ActivePerl 5.8.8:n.

Kun yllä olevat kirjastot on asennettu, MRTG käännetään konekielelle ja asennetaan koneelle seuraavilla komennoilla:

```
./configure [--prefix=/usr/local/mrtg-2] [with-gd=…] [with-z=…] \
[with-png=…]
make
make install
```

Configure-komennossa prefix-komennolla voidaan tarvittaessa määritellä asennushakemisto, joka oletuksena on /usr/local/mrtg-2. With-komennoilla puolestaan määritellään GD-, Zlib- ja PNG- kirjastojen sijainnit, mikäli ne sijaitsevat oletuksesta poikkeavissa paikoissa. Jos ne on asennettu julkaisun normaalilla päivitysohjelmalla, ei niiden sijaintia tarvitse määritellä. Vaihtoehtoisesti myös koko MRTG:n voi asentaa paketinhallintaohjelmalla.

4.1.2 Käyttö

MRTG:tä käytetään komentoriviltä. Kun ohjelma asennettu, tehdään peruskonfigurointi bin-hakemistosta löytyvällä komennolla cfgmaker (Oetiker 2007: MRTG Unix Guide). Vaadittavia parametrejä ovat ainakin seuraavat:

```
cfgmaker        --global 'WorkDir: /home/httpd/mrtg' \
                --global 'Options[ ]: bits,growright' \
                --output /home/mrtg/cfg/mrtg.cfg    \
                community@router.abc.xyz
```

Output-parametriksi laitetaan www-palvelimella näkyvä hakemisto. Jos koneelle on asennettu Apache oletushakemistoon /var/www, laitetaan siihen esimerkiksi /www/htdocs/mrtg. **Community@router.abc.xyz** puolestaan määrittelee valvottavan laitteen (FQDN-nimi tai IP-osoite) ja SNMP-yhteisönimen.

Tämän jälkeen ajetaan juuri luotu tiedosto mrtg-komennolla:

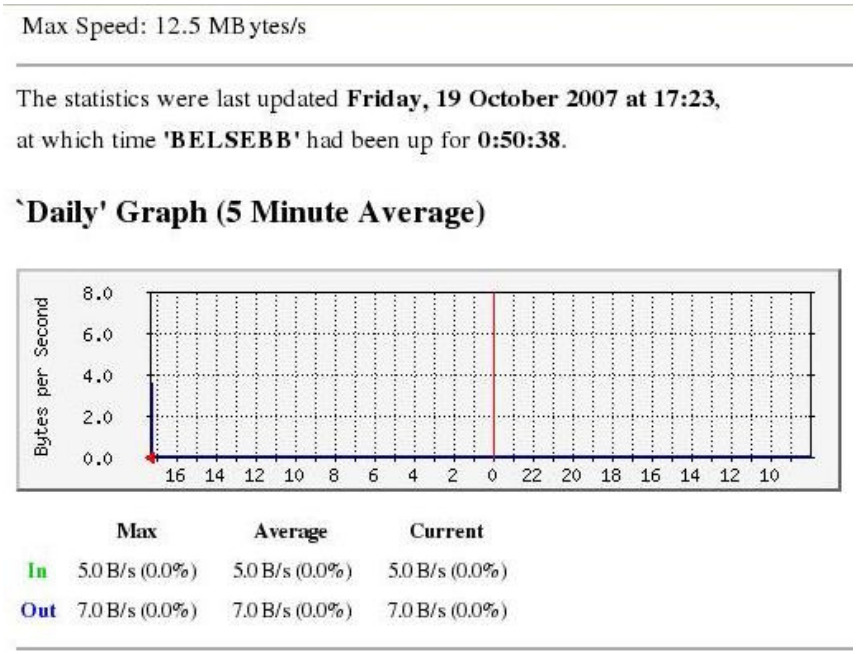
```
mrtg /home/mrtg/cfg/mrtg.cfg
```

Ohjelma luo www-hakemistoon laitteen linkkien kuormituksesta www-sivun muodossa analyysin, jota voi katsoa selaimella

osoitteessa localhost/mrtg. Ohjelma kyselee verkkolaitteelta tiedot, mutta vain kerran. Siksi tämä tulisi automatisoida ajettavaksi esimerkiksi oletusarvona 5 minuutin välein. Tämä saadaan lisäämällä /etc/crontab -tiedostoon seuraava rivi:

*/5 * * * * <mrtg-bin>/mrtg <path to mrtg-cfg>/mrtg.cfg --logging /var/log/mrtg.log

<mrtg-bin> kohtaan tulee siis hakemisto, johon MRTG on asennettu, ja <path to mrtg-cfg> kohtaan hakemisto, johon konfigurointitiedosto luotiin. Kuvassa 4.1 on MRTG:n luoma yhden portin päivittäinen linkkianalyysi.



Kuva 4.1. MRTG:n itseskaalautuva linkkianalyysi

4.1.3 Yhteenveto

MRTG on suhteellisen helppo saada toimimaan yhden tai muutaman laitteen valvonnassa, mutta hieman hankala ja työläs käyttöönotettavaksi suuressa verkkoympäristössä. Käytännössä asetukset joudutaan tekemään asetustiedostoja käsin muokkaamalla. Jotta valvottavat laitteet saataisi järkevään järjestykseen, ne pitäisi järjestellä itse tehdylle HTML-sivustolle linkeiksi.

4.2 RRDtool

RRDtool (Round Robin Database tool) on niin ikään Oetikerin kehittämä ohjelma (Oetiker 2007: About RRDtool.) Sen piti alun perin olla paranneltu, laajennettu uusi versio MRTG:stä, mutta se kasvoikin lopulta ihan omaksi kokonaisuudekseen, ja molemmat ovat nykyään laajalti käytössä. RRDtool on saavuttanut lähes vapaan lähdekoodin standardin aseman tietojen kirjaus- ja kuvausohjelmana aikajanaa vasten. Siinä missä MRTG on melko yksinkertainen Perl-skripti joka hakee SNMP-tietoja, RRDtool on huomattavasti edistyksellisempi ohjelma.

RRDtool on nimensä mukaisesti tietokanta, joka voi myös piirtää kuvaajia MRTG:n tapaan. Verrattuna esimerkiksi SQL-tietokantoihin, on RRDtool kuitenkin huomattavasti yksinkertaisempi. Round Robin tarkoittaa sitä, että tietokanta kirjoittaa vanhojen tietojen päälle kiertämällä ”ympyrää.” Tieto arkistoidaan RRA:han (Round Robin Archive) tietyn aikavälin keskiarvoina. Tiedon tarkkuus heikkenee sitä myötä kun se vanhenee, jolloin entistä suuremman aikavälin arvoista lasketaan keskiarvo, joka tallennetaan. Näin vanhojen tietojen päälle kirjoittaminen on mahdollista. Tietokannan koko määritellään luodessa se, jolloin se pysyy jatkuvasti samankokoisena.

4.2.1 Asennus

MRTG:n tavoin myös RRDtool vaatii useita kirjastoja asennettuna toimiakseen, joista osa on samoja kuin MRTG:llä (Oetiker 2007: Building RRDtool):

- LibGD
- Zlib
- Libpng
- Freetype
- Libart_lgpl

Nämä saadaan asennettua apt-get -komennolla. Kun kyseiset kirjastot on asennettu, voidaan asentaa itse RRDtool. Lähdekoodi uusimmasta vakaasta versiosta löytyy Tobias Oetikerin kotisivulta osoitteesta http://oss.oetiker.ch/rrdtool/pub/rrdtool-1.2.23.tar.gz. Tämän jälkeen paketti puretaan, käännetään ja asennetaan MRTG:tä vastaavalla tavalla haluttuun hakemistoon. Kirjastojen sijainti täytyy jälleen määritellä erikseen, mikäli ne sijaitsevat tavallisesta poikkeavissa paikoissa. Vaihtoehtoisesti myös RRDtool ja samalla kaikki sen vaatimat kirjastot voidaan hakea apt-get -komennolla.

4.2.2 Käyttö

Myös RRDtoolia käytetään komentoriviltä. MRTG:hen verrattuna huomattavasti monipuolisempana se on kuitenkin myös hieman hankala ja monimutkainen. RRDtoolissa on lukuisia funktioita, joista suurin osa on ohjelman oman RRD-tietokannan luonti- ja käsittelykomentoja, joita voi käyttää skripteillä varsin tehokkaasti.

Koska RRDtool on suunniteltu käsittelemään nimenomaan aikasidonnaista dataa, täytyy sen saada jokaista aika-arvoa vastaava data-arvo. Jos se ei saa data-arvoa, se tallentaa arvon UNKNOWN kyseiseen kohtaan. Tämä on syytä huomioida skriptejä tehdessä, sillä tietolähteenä tulisi olla jatkuvaa tietovirtaa lähettävä laite. Aika-arvot ovat aina TIMESTAMP-muotoa 01-01-1970, ja ilmaistaan sekunteina jotka ovat kuluneet tästä ajankohdasta. Niinpä uusia data-arvoja vastaavien aika-arvojen täytyy aina olla suurempia kuin edellisten (van den Bogaert 2007.)

RRDtool piirtää myös kuvaajat png-tiedostoina MRTG:n tapaan, mutta skripteissä käytön ohella varsinainen hyöty ohjelmasta saadaan sitä hyödyntävän graafisen www-käyttöliittymän avulla. Näitä on useita erilaisia, kuten Cricket ja nykyään ehkä suosituimpana ja monipuolisimpana Cacti.

4.3 Cacti

Cacti on todella monipuolinen ja laajalti käytetty ohjelma. Verrattuna esim. Cricketiin, sillä on myös laaja sivusto, dokumentaatio ja yhteisötuki foorumeineen ja postituslistoineen. Se on myös varsin laajennettava ja skaalautuva ohjelma, johon on lukuisia erilaisia lisäosia jo valmiina. Cacti käyttää RRDtoolia valvottavien tietojen kirjaus- ja tallennustoimintoihin sekä kuvaajien piirtämiseen.

4.3.1 Asennus

Asensin myös Cactin Debian 4 Linuxiin, sekä kokeilumielessä myös Fedora 7:lle. Ohjelma toimi moitteettomasti kummassakin. Tarkka ja yksityiskohtainen ohje Cactin asentamiseen Debianiin löytyy liitteestä A. Cacti tarvitsee toimiakseen RRDtoolin ja sen vaatimien kirjastojen lisäksi myös seuraavat ohjelmat (Carter 2004):

- MySQL-tietokanta
- Apache web-palvelin (tai Windowsissa IIS eli Internet Information Services)
- NetSNMP
- PHP-tulkki

Cacti tallentaa hallinnalliset tiedot MySQL-tietokantaan. Myös ohjelman asentaminen on kohtalaisen työläs ja monimutkainen projekti, ainakin MRTG:hen tai pelkkään RRDtooliin verrattuna.

Yllä olevat ohjelmat kannattaa asentaa Linuxin oman päivitystenhallinnan kautta. Tämän jälkeen varmistetaan että ne myös toimivat ja että mysqld- ja httpd-palvelut ovat päällä.

MySQL-tietokantaan luodaan Cactia varten käyttäjä, esimerkiksi **cactiuser**, sekä tietokanta, esimerkiksi **cactidb**, johon kyseiselle käyttäjälle annetaan täydet oikeudet (Carter 2004):

mysql> create database **cactidb**;
mysql> grant all on cactidb.* to **cactiuser**;
mysql> grant all on cactidb.* to **cactiuser**@localhost;
mysql> set password for **cactiuser**@localhost=password('**cactipw**');

Cactia ei tarvitse kääntää ennen asentamista. Kun asennuspaketti on haettu ja purettu, kopioidaan tiedostot Apache-asennushakemistoon luotuun uuteen hakemistoon, esimerkiksi /var/www/cacti.

MySQL:n lisäksi myös Linuxiin luodaan käyttäjä Cactia varten, joka ajaa PHP:llä tiedostoa poller.php. Tämä on siis eri käyttäjätili kuin MySQL:n **cactiuser**-käyttäjä, mutta voi olla samanniminenkin. Käytännössä ajo tapahtuu samalla tavalla kuin MRTG:llä, eli lisäämällä hakemistosta /etc löytyvään tiedostoon crontab seuraava rivi:

*/5 * * * * **cactiuser** php /var/www/cacti/poller.php > /dev/null 2>&1

Tällä asetuksella komento ajetaan viiden minuutin välein, jolloin kuvaajiin otetaan näyte kerran 5 minuutissa. Tämä on oletusarvo, mutta väli kannattaa optimoida ympäristön ja vaatimusten mukaan, huomioiden sekä siitä aiheutuva verkon ja laitteiden kuormitus, että näytteiden vaatima tila palvelimella. Linuxin **cactiuser** -käyttäjälle annetaan täydet omistus- ja muut oikeudet cacti-hakemiston alihakemistoille rra ja log:

chown -R **cactiuser** rra log
chgrp -R **cactiuser** rra log

Cactin asennushakemistosta löytyy tiedosto cacti.sql, joka ajetaan MySQL:ään komennolla

mysql –u **cactiuser** –p **cactidb** < cacti.sql

Tämä luo cactidb -tietokantaan ohjelman vaatimat taulut ja sisällön. Hakemistosta include löytyy config.php, josta löytyvät seuraavat tiedot:

$database_default = "**cactidb**";
$database_hostname = "localhost";
$database_username = "**cactiuser**";
$database_password = "**cactipw**";

Näiden tulee vastata juuri luodun tietokannan tietoja.

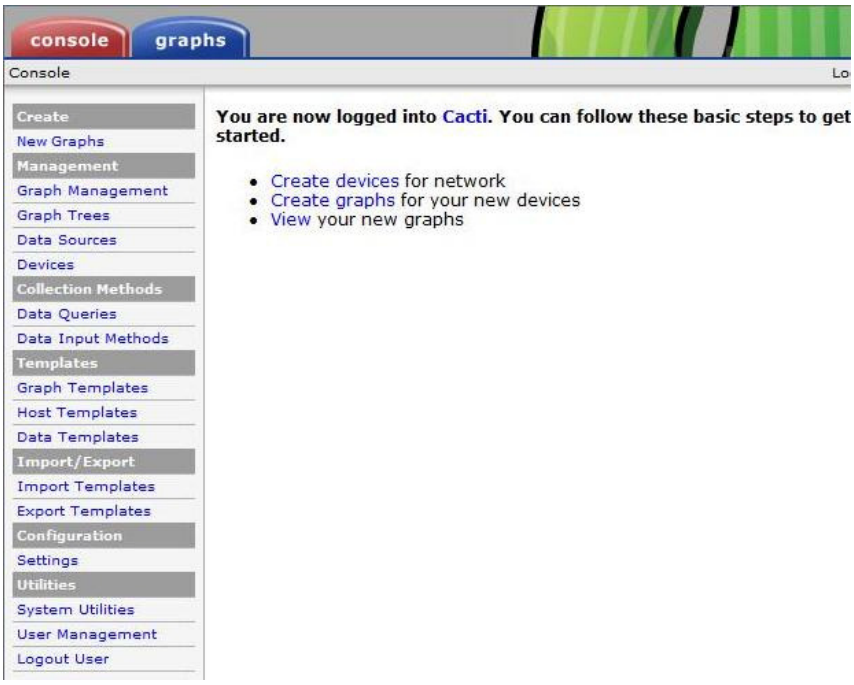
Tämän jälkeen mennään www-selaimella osoitteeseen localhost/cacti tai toiselta koneelta koneennimitaiip-osoite/cacti josta asennus jatkuu graafisen käyttöliittymän kautta. Mikäli tässä vaiheessa tulee virheilmoitus puuttuvista oikeuksista tietokantaan, tarkoittaa se sitä, että config.php -tiedostossa olevat tiedot ja/tai niitä vastaavat tietokannan tiedot eivät ole oikein.

Kun kaikki on kunnossa, voidaan aloittaa asentaminen asennusvelhon kautta. Ohjelma kysyy PHP:n, RRDtoolin ja muiden vaadittavien komponenttien sijaintipolut. Ensimmäistä kertaa kirjautuessa käytetään oletustunnusta admin ja salasanaa admin, joka pyydetään välittömästi vaihtamaan.

Linuxin asetuksissa kannattaa huolehtia myös siitä, että ohjelman vaatimat palvelut on laitettu päälle pysyvästi, jolloin ohjelma lähtee käyntiin myös konetta uudelleenkäynnistäessä automaattisesti.

4.3.2 Käyttö

Kun Cactiin kirjaudutaan sisään, avautuu kuvassa 4.2 oleva näkymä.

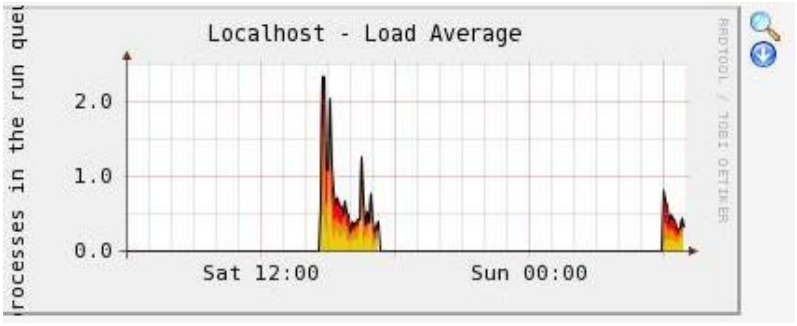


Kuva 4.2. Cactin etusivu

Cactin käyttäminen on kohtalaisen helppoa selkeän graafisen käyttöliittymän ansiosta. Etusivulta voidaan luoda järjestelmään uusia laitteita, kuvaajia jo luotuihin laitteisiin, sekä katsoa luotuja kuvaajia. Vasemmalla (kuva 4.2) on konsoli, josta löytyy kaikki ohjelman keskeiset hallintatoiminnot. Ylhäältä voi vaihtaa välilehtien console ja graphs välillä, joista jälkimmäisessä tulee oikeaan reunaan lisää välilehtivaihtoehtoja graafien tarkastelua helpottamaan.

Yleistä

Cacti tekee oletuksena isäntäkoneesta 4 kuvaajaa: keskimääräinen kuormitus, kirjautuneet käyttäjät, muistin käyttö, sekä päällä olevien prosessien määrä. Crontab-tiedostoon määritellystä kyselyvälistä riippuen voi kuitenkin kestää kahden (oletuksena 5 minuutin pituisen) syklin verran, ennen kuin ohjelma alkaa piirtämään minkäänlaista kuvaajaa. Kuvassa 4.3 on näistä keskimääräistä kuormitusta kuvaava käyrä.



Kuva 4.3. Ohjelman oletuksena luoma kuormituskäyrä

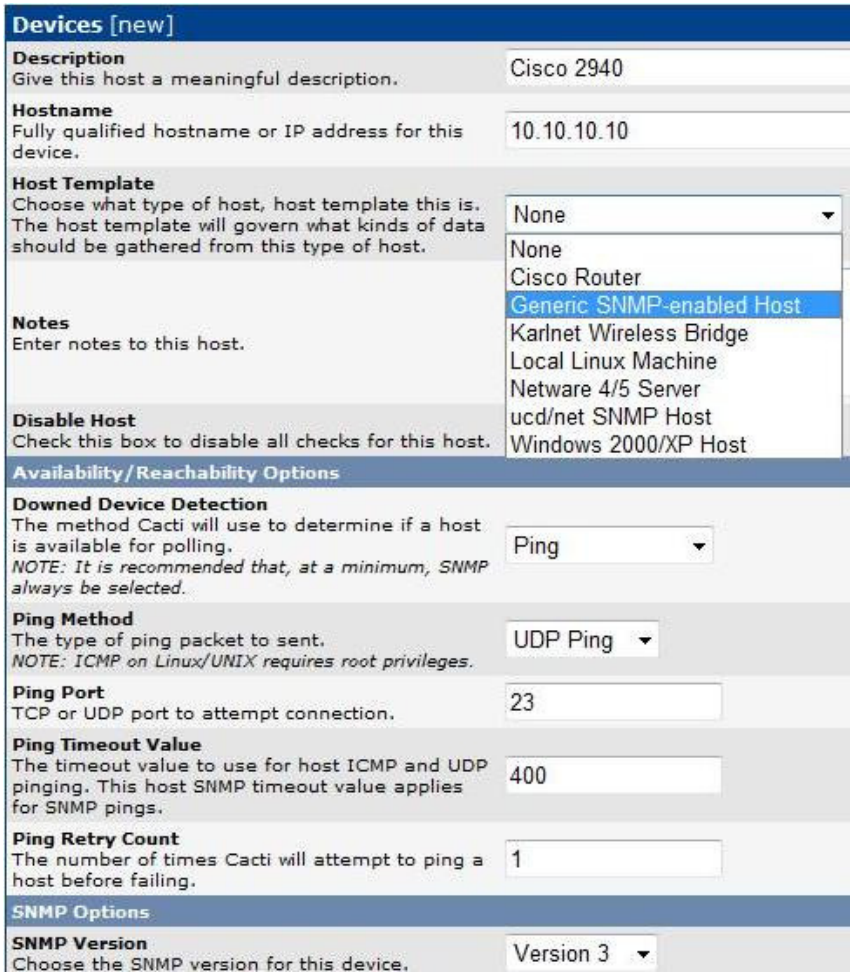
MRTG:n tavoin Myös Cacti luo jokaisesta kuvaajasta oletuksena seuraavat versiot, joita pääsee tutkimaan tarkemmin klikkaamalla haluttua käyrää:

- Päivittäinen näkymä 5 minuutin näytevälillä
- Viikoittainen näkymä 30 minuutin näytevälillä
- Kuukausittainen näkymä 2 tunnin näytevälillä
- Vuosittainen näkymä 1 päivän näytevälillä

Uusia laitteita luodaan joko etusivun ”Create Devices For Network” -tekstiä painamalla tai vasemman laidan Management -> Devices alta ja valitsemalla Add (kuva 4.4).

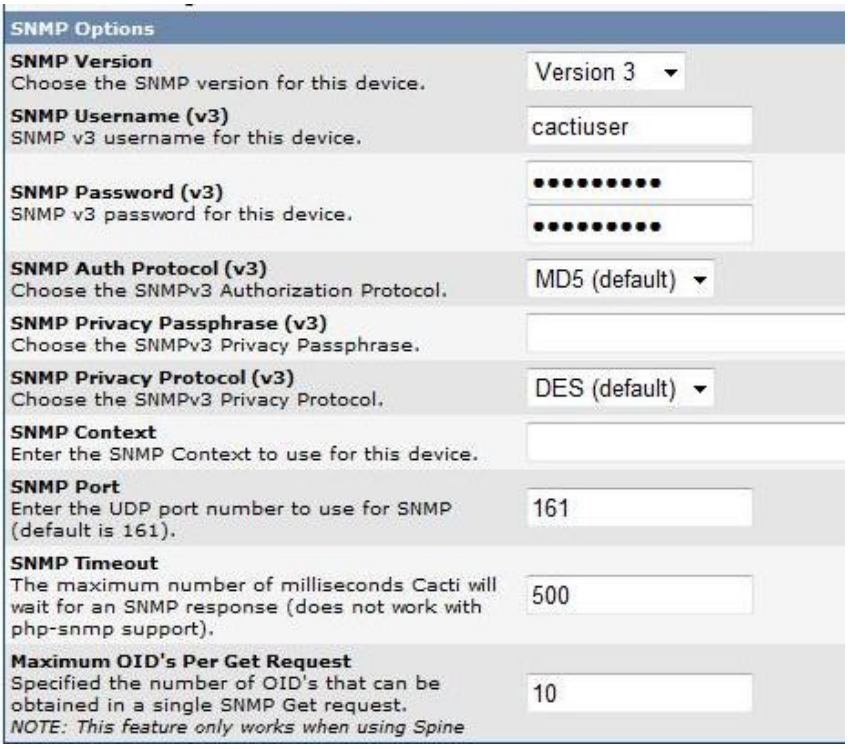
Kuva 4.4. Cactin laitehallinta. Oikeassa yläkulmassa olevasta Add -painikkeesta lisätään uusia laitteita.

Tärkeimpiä asetuksia uutta laitetta luodessa ovat Hostname (joko täyspitkä toimialuenimi eli FQDN tai IP-osoite), Host Template (joka määrää mitä laiteelta voidaan valvoa) sekä SNMP-optiot versio ja SNMP -yhteisö. Kuvassa 4.5 ovat uuden luotavan laitteen asetukset.



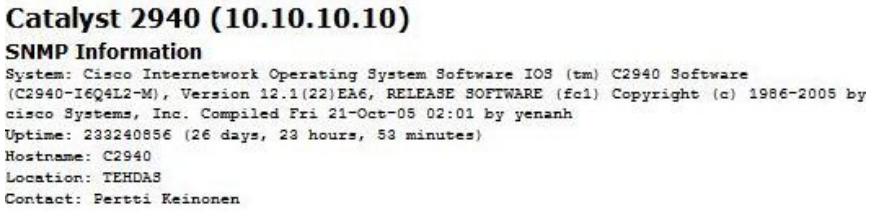
Kuva 4.5. Uuden laitteen luonti

Jos SNMP:stä on käytössä versio 3, asetetaan myös autentikointi-, salaus- ja salasana-asetukset kuntoon. SNMP-asetukset näkyvät kuvassa 4.6.



Kuva 4.6. SNMP-optiot uuden laitteen luonnissa

Tämän jälkeen painetaan Create, jolloin ohjelma ottaa yhteyttä luotuun laitteeseen. Jos tämä onnistuu, tulee laitteen tietoja näkyviin yläreunaan. Ciscon verkkolaitteissa nähdään esimerkiksi laitteen nimi, IOS-käyttöjärjestelmän (Cisco Internetwork Operating System) versio, aika, jonka laite on ollut ylhäällä, sekä muuta tietoa, kuten kuvassa. 4.7.



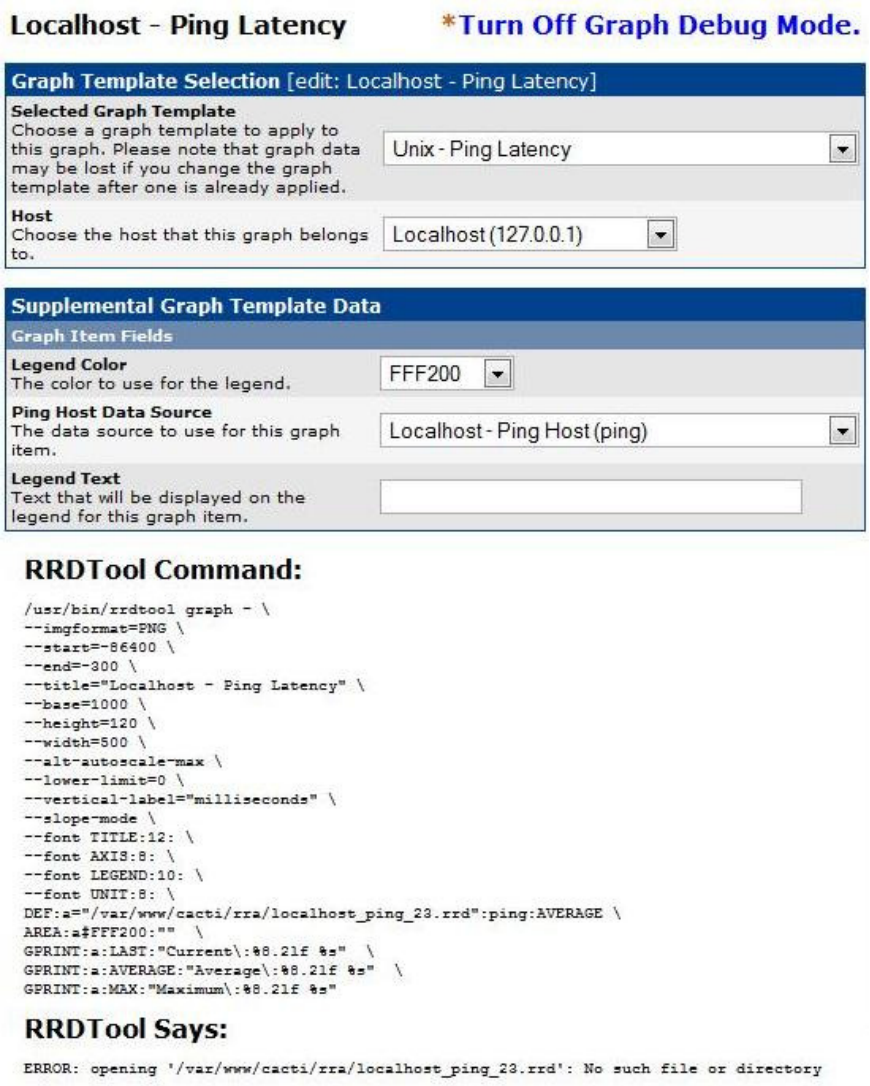
Kuva 4.7. Testilaitteen tiedot, jotka tulevat näkyviin uutta laitetta luodessa, mikäli yhdeydenotto onnistuu.

Oikealta painettaessa ”Create Graphs for this Host” voidaan luoda käyriä esimerkiksi porttikohtaisesta kuormituksesta. Tämä näkyy kuvassa 4.8.

Kuva 4.8. Kuvaajia uusille laitteille luodaan täältä.

Jos SNMP-yhteyden otto laitteeseen ei onnistu, tulee virheilmoitus ”SNMP error”. Tämä voi johtua joko siitä, että NetSNMP -paketti ei toimi tai ei ole kokonaan asennettu, tai siitä, että autentikointi- tai yhteisötiedot ovat väärin joko ohjelmassa tai laitteessa. Mahdollisia muita ongelmia voi tutkia kuvaajien luonnissa debugging-tilassa. Ohjelma luo jokaista kuvaajaa kohden RRDtoolilla rrd-päätteisen tiedoston Cactin rra-hakemistoon. Debugging-tilassa nähdään muun muassa, mikäli kirjoitusoikeudet kyseiseen hakemistoon eivät ole kunnossa.

Toinen mahdollinen ongelma on, ettei kuvaajaan tule käyrää lainkaan näkyviin. Tällöin ohjelma valittaa ”No such file or directory.” Tämä johtuu crontab-tiedostoon asetetusta aikavälistä, ja on ihan normaalia. Uuden kuvaajan luonnista kestää jonkin aikaa (yleensä kaksi aikasykliä, eli oletusasetuksilla 5-10 minuuttia) ennen kuin ohjelma luo kyseisen kuvaajan. Mikäli kuvaajaa ei tässä ajassa tule, kannattaa mahdollista ongelmaa yrittää selvittää debugging-tilasta. Kuvassa 4.9 on vasta luodun käyrän debug-tila päällä.



Kuva 4.9. Kuvaajan debug-tila päällä.

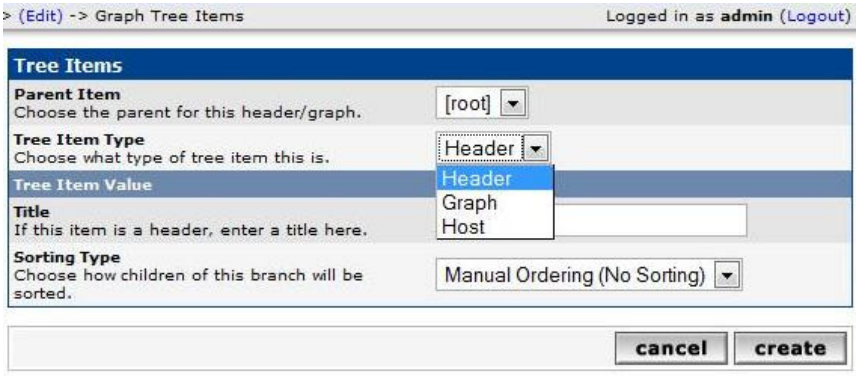
Hallintakonsoli

Kun konsoli on valittuna, vasemmalla ovat keskeisimmät hallintatoiminnot. Ylimpänä on Create -otsikon alla New Graphs josta näkyy viimeisimmät luodut laitteet ja niille luodut käyrät. Tästä voi myös luoda uusia käyriä.

Kuva 4.10 Management -valikko

Toisena on Management, joka näkyy kuvassa 4.10. Graph Management-valikossa näkyy kaikki luodut kuvaajat, ja siitä voi siirrellä, lisätä ja poistaa kuvaajia puurakenteessa. CDEFs sisältää käsittelytoimintoja tiedon käsittelyyn, kuten oletuksena toimiva 5 minuutin kysely, bittien kääntäminen tavuiksi (eli kertomalla 8:lla) jne. Colors -valikosta voi muuttaa käyrien väriä.

Graph Trees -valikosta hallitaan Graphs -näkymän puurakennetta. Painamalla Add -painiketta oikeassa yläkulmassa voidaan puurakenteeseen luoda lisää haaroja. Näihin voidaan edelleen luoda alihaaroja, lisätä yksittäisiä, jo luotuja kuvaajia, sekä ehkä parhaimpana vaihtoehtona kokonaisia laitteita. Nämä näkyvät kuvassa 4.11. Kokonaisen laitteen lisäämällä ohjelma laittaa laitetta vastaavan otsikon alle automaattisesti kaikki kyseiselle laitteelle luodut kuvaajat loogiseen, mutta edelleen muokattavaan järjestykseen. Näin säästyy paljon aikaa ja vaivaa eri kuvaajien järjestelemisessä, varsinkin kun Graph Management -näkymä muuttuu varsin sekavaksi, kun kuvaajia alkaa olla enemmän.



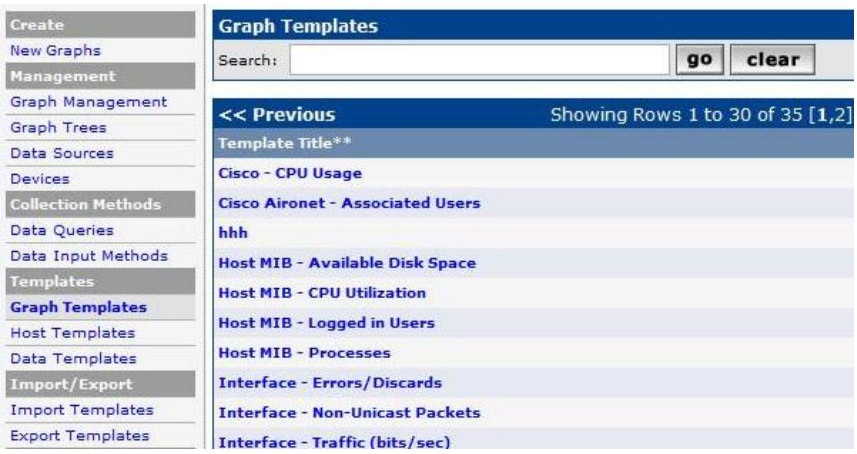
Kuva 4.11. Puurakenteen hallinta

Data Sources -sivulla on käytössä olevat tiedonkeräystavat. Alivalikko RRA eli Round Robin Archive näyttää jokaisesta käyrästä luotavat tietokannat eri tiedonhakuvälillä ja aikajaksolla. Viimeisenä vielä Devices -sivulla näkyy valvottavat laitteet, jonka kautta myös uusia laitteita luodaan.

Mallit

Templates- eli mallivalikosta löytyy kolme kohtaa: Graph Templates, Host Templates sekä Data Templates. Näistä Graph Templates on kuvassa 4.12. Oleellista Cactin laajennettavuuden kannalta on, että näitä voi luoda lisää, sekä tuoda ja viedä alapuolella olevasta Import/Export-valikosta (kuva 4.12.) Cactin sivustolta löytyy runsaasti valmiiksi tehtyjä malleja eri laitteille ja ohjelmistoille. Nämä luovat yleensä useita uusia malleja jokaiseen osioon. Data kerätään yleensä tietyllä OID:llä, joka saat-

taa olla eri laitteissa olla erilainen. Tästä datasta luodaan kuvaajamalli. Lopulta nämä linkitetään yhteen.



Kuva 4.12. Kuvaajamallien hallinta

Kuten tästä voi päätellä, voi uusien mallien luominen olla hieman työlästä. Ensin on luotava tietotyyppi (Data Templates –valikosta), joka haetaan useinmiten OID:n avulla. Lisäksi valitaan kuinka usein tietoa haetaan, mitä RRA:ta se käyttää (oletuksena kaikki paitsi tunti) ja nimetään tietotyyppi. OID:n toimivuus kannattaa ensin testata isäntäkoneen komentoriviltä. Tämä onnistuu komennolla

snmpwalk –Os **laitteen_ip** –c **community** –v **snmp-ver OID_nro**

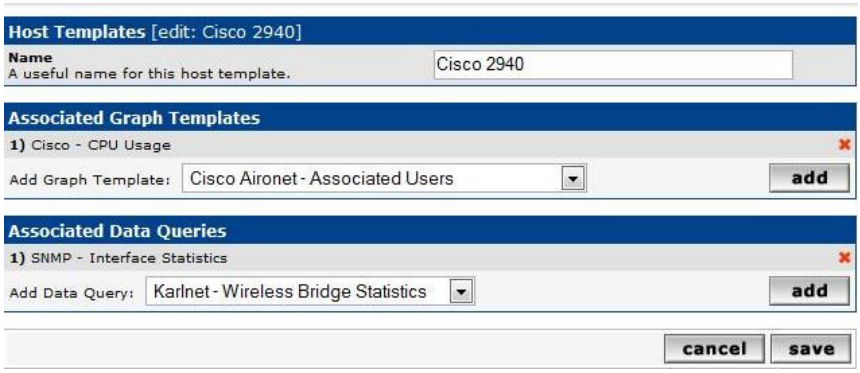
Esimerkiksi:

```
cerv:~# snmpwalk -Os 1.2.3.4 -c joo -v 2c 1.3.6.1.4.1.9.9.48.1.1.1.5.1
enterprises.9.9.48.1.1.1.5.1 = Gauge32: 3855228
cerv:~#
```

Graph Templates eli kuvaajamallit sisältävät käytössä olevat kuvaajatyypit. Tässä esimerkiksi määritellään kuvaajan koko Graphs -näkymässä. Tallentamisen jälkeen linkitetään siihen haluttu tietotyyppi Graph Item inputs -kohdasta. Graph Template Items -kohdasta lisätään kuvaajaan käyriä. Näitä myös voi ja kannattaa (harkitusti) laittaa useita samaan kuvaajaan esimerkiksi siten, että yhtä muuttujaa kuvataan alueena ja muita käyrinä. Näin ne eivät jää toistensa alle.

Host Templates eli laitemallit mahdollistavat erilaisten laiteprofiilien luonnin, jolla tietyillä, kyseisellä laitteella toimivilla menetelmillä (yleensä OID:t) luotuja kuvaajia voidaan yhdistää helposti hallittavaksi kokonaisuudeksi Nämä näkyvät kuvassa

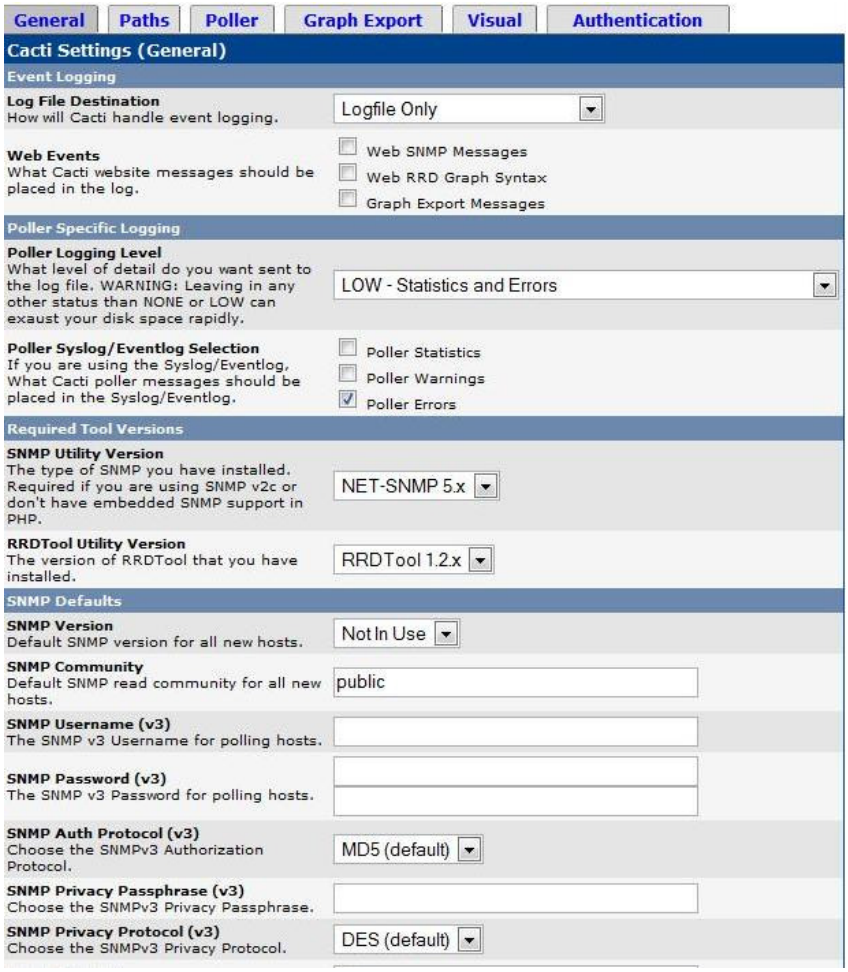
4.13. Näin kyseisellä laiteprofiililla voi luoda valvottavaksi uuden laitteen, jossa luodaan oletuksena halutut kuvaajat. Esimerkiksi paljon käytetylle Cisco Catalyst 2950 -kytkimelle kannattaa luoda profiili, joka tekee kuvaajat automaattisesti muun muassa prosessorin käytöstä, muistin käytöstä ja laitteen keskimääräisestä kuormituksesta. Lisäksi valitaan portit, joista halutaan käyrät sisään ja ulos menevästä liikenteestä. Tarkemmat ohjeet laiteprofiilien luontiin on liitteessä B.



Kuva 4.13. Laitemallien hallinta

Asetukset

Configuration-otsikon alta löytyy Settings-sivu, josta löytyy ohjelmiston yleiset asetukset (General), polut Cactin tarvitsemille ohjelmille (Paths), kyselymetodin asetukset (Poller), kuvaajien vientiasetukset (Graph Export), yleisiä asetuksia kuvaajien ulkonäköön (Visual) sekä autentikointiin (Authentication) liittyen. Utilities-otsikon alla on System Utilities, josta löytyy ohjelman lokitiedostot, kysely- ja SNMP-välimuistit sekä Technical Support-sivu, josta löytyy paljon hyödyllistä tietoa ohjelman toiminnasta. Asetusten etusivu on kuvassa 4.14.



Kuva 4.14. Cactin asetussivu

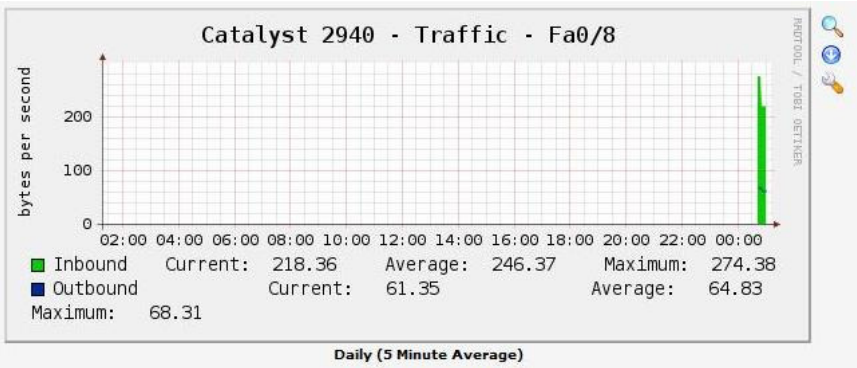
Viimeisenä on vielä User Management, josta voi järjestelmään luoda käyttäjiä ja esimerkiksi antaa vain tiettyjä oikeuksia varsin monipuolisesti. Oletuksena ohjelmaan on luotu admin- ja guest-tunnukset, joille on vastaavasti annettu täydet oikeudet ja vain kuvaajien katseluoikeudet. Guest-tunnus on oletuksena pois käytöstä.

Käyrien katselu

Valmiita kuvaajia voi katsella valitsemalla Graphs-välisivun, jolloin oikeaan reunaan avautuu lisää muokkausmahdollisuuksia, kuten kuvassa 4.15.

Kuva 4.15. Kuvaajien katselutilan valikko

Settings -valikosta tehdään yleisluontoisia asetuksia kuvaajien katseluun ja puurakenteeseen liittyen, kuten esimerkiksi mitä kuvaajia ja miltä aikaväliltä oletusarvoisesti näytetään. Settings-välilehden oikealla puolella seuraava on puurakenne, joka on epäilemättä hyödyllisin. Seuraava on listarakenne jossa näkyy listana kaikkien laitteiden kaikki kuvaajat niminä. Viimeisessä eli esikatselutilassa näkyy kaikkien laitteiden kaikki kuvaajat pienennettyinä, kahteen riviin järjestettynä.



Kuva 4.16. Kuvaajanäkymä

Kuvaajanäkymässä (kuva 4.16) oikealla olevasta suurennuslasista voi tarkentaa kuvaajaa maalaamalla kuvaajasta halutun alueen ja alaspäin osoittavasta nuolesta tallentaa datan csv-muodossa tarkempaan tarkasteluun esimerkiksi Microsoft Excelillä. Lisäksi kun kuvaajaa painaa tarkempaa tutkailua varten, tulee vielä jakoavaimen kuva (Properties). Tästä näkee ohjelman suorittaman RRDtool-skriptin kuvaajan piirtämiseen (kuva 4.17.) Tämä on sama minkä näkee debug-tilassa kuvaajan luonnissa.

Kuva 4.17. Kuvaajan Properties-ikkuna. Tästä näkee millaista skriptiä Cacti ajaa RRDtoolilla

4.3.3 Yhteenveto

Kaiken kaikkiaan Cacti on todella laaja ja monipuolinen ohjelma. Lähes kaikkea voi asetuksilla muuttaa haluamansalaiseksi ja jo oletusasetuksilla pääsee pitkälle.

Tulee kuitenkin huomioida, että Cacti ei ole varsinainen NMS, vaan nimenomaan verkonvalvontaohjelma. Se rajoittuu vain ja ainoastaan yhdentyyppisten SNMP-funktioiden, eli get ja snmpwalk, hyödyntämiseen. Se ei siis ota vastaan Trap-viestejä laitteilta ainakaan ilman laajennuksia, eikä sillä voi tehdä minkäänlaista SNMP-hallintaa laitteille. Tässä mielessä kyseessä on varsin rajoittunut ohjelma. Näistä periaatteessa rajoittuneista funktioista se kuitenkin ottaa kaiken irti.

Halutessa Cactia voi myös laajentaa skripteillä jopa SNMP:n ulkopuolellekin, käyttäen PHP:ta, Perliä ja Bash-skriptausta. Tähän en kuitenkaan paneudu tässä työssä sen tarkemmin. Cactin kotisivuston foorumeilta löytyy runsaasti erilaisia laajennuksia, joita voi hyödyntää kohtalaisen helpostikin.

5 Cactin käyttöönotto Imatran Tehtailla

Kaiken kaikkiaan kävi jo lyhyen kokeilun perusteella varsin selväksi, että Cacti on selkeästi paras vaihtoehto verrattuna varsin pelkistettyyn MRTG:hen, tai siihen, että RRDtoolia käyttäisi yksinään. Cricketiä, joka on Cactin tapaan RRDtoolia käyttävä web-käyttöliittymä, en lopulta edes kokeillut, koska Cricketistä saatavilla oleviin tietoihin tutustumisen perusteella Cacti vaikutti ylivertaiselta siihenkin verrattuna.

Cacti on monipuolisuudessaan, helppokäyttöisyydessään ja ominaisuuksiltaan lähes kaupallisten ohjelmien tasoa. Ohjelmalla on myös todella laaja käyttäjäyhteisö sekä foorumit. Yhteisösivuilta löytyy myös todella paljon tutoriaaleja ja oppaita huomattavasti peruskäyttöä pidemmälle ja laajennuksien itse tekemiseen.

Ohjelma on selkeästi tehty suurten verkkojen valvontaa varten. Kaupallisiin ohjelmistoihin verrattuna suurin ero on ehkä siinä, ettei koko ohjelmassa ole minkäänlaisia Help-toimintoja. Onneksi valikot ovat selkeitä ja melko hyvin dokumentoituja, ja vähän asioista perillä oleva henkilö pärjää ohjelman kanssa varmasti. Mutta tarvittaessa kaiken lisäavun joutuu etsimään Cactin viralliselta sivustolta tai muualta internetistä.

Valitettavasti monipuolisuus ja monen eri osa-alueen hyödyntäminen näkyy myös asennuksessa, ainakin MRTG:hen verrattuna. Siinä missä MRTG on kohtalaisen helppo asentaa toimintakuntoon (pienelle laitemäärälle), Cactin toimimaan saaminen voi olla hieman hankalampaa. Ongelmia voi tulla useissa asennuksen vaiheissa, eikä niiden ratkaiseminen ole välttämättä kovinkaan helppoa. Siksi onkin hyvä asentaa ohjelman vaatimat osat yksi kerrallaan ja testata niiden toimivuus kunnolla, ennen kuin siirtyy eteenpäin. Ne kannattaa asentaa nimenomaan Linuxin oman ohjelmistonhallinnan kautta.

5.1 Huomioitavaa käyttöönotossa

Kun Cacti implementoidaan suuren yrityksen tuotantoverkkoon, tulee ottaa huomioon useita asioita, liittyen lähinnä palvelimeen sekä Cactin konfiguraatioon.

Kokeilujen perusteella Cacti toimi moitteettomasti Debian 4:lla, jossa on SELinux-tila päällä. Tämä pakottaa käyttöön tiettyjä tietoturva-asetuksia, jotka rajaavat sitä, mitä koneella sallitaan

tehdä. Valitsin Debianin alustaksi, koska sillä on yleisesti hyvä maine vakaana ja tietoturvallisena julkaisuna.

Kiintolevyn osiointivaiheessa olisi hyvä ajatus laittaa ainakin /var-hakemisto erilliselle osiolle. /var/www/cacti/rra-hakemistoon tulee kaikki ohjelmaan luodut RRA:t, jotka vievät oletuksena tilaa noin 50 kilotavua per käyrä. Jos samassa kuvaajassa on useampi käyrä, vie kuvaajaa vastaava RRA-tiedosto tilaa noin 50 kilotavua kerrattuna käyrien määrällä.

Cacti ei vaadi isäntäkoneelta kovinkaan paljoa tehoa eikä kiintolevytilaa. Asensin ohjelman ja käyttöjärjestelmän vanhalle käytöstä poistetulle IBM T40 -kannettavalle, jossa on 512 MB muistia ja 40 GB kiintolevy. Tältä koneelta se siirretään virtuaaliseksi myöhemmin saapuvaan uuteen Sun Microsystemsin palvelimeen.

Vaihtoehtoisesti Cactin voisi asentaa jo käytössä olevalle Linux-palvelimelle yhdeksi uudeksi palveluksi. Koko asennus toimintakunnossa vie käyttöjärjestelmineen tilaa vain noin 1,3 GB. Tämän jälkeen melkeinpä ainoa lisätilaa vievä tekijä ovat RRA:t, joita luodaan yksi jokaista käyrää kohden, mutta joiden koko määritellään ennalta. Oletusarvoisesti luotu RRA vie 47840 tavua tilaa.

Jos esimerkiksi jokaista Imatran Tehtaiden kampusalueen 260 laitetta kohden luodaan kuvaajia, jotka sisältävät yhteensä 100 käyrää, vievät nämä tilaa yhteensä vain noin 1,2 GB. Mikäli kuvaajien tallennustarkkuutta muuttaa tarkemmaksi, eli käytännössä RRA-tietokannan kokoa suuremmaksi, on yhden kuvaajan tilan kulutus hieman suurempi, mutta silti nykymittapuulla minimaalista.

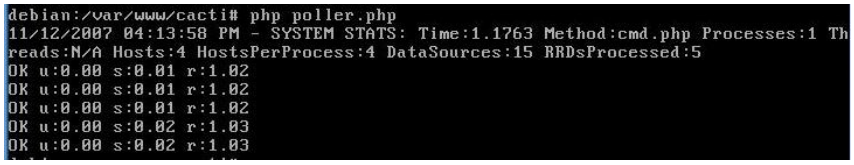
Asensin käyttöjärjestelmän puhtaalta pöydältä, ilman työpöytää, ja sen jälkeen asensin siihen vain Cactin vaatimat ohjelmat sekä SSH-palvelun palvelimen etähallintaa varten. Root-, MySQL:n Root- ja Cactin Admin- käyttäjille kannattaa laittaa vahvat salasanat. Sekä Debian että Cacti mahdollistavat myös LDAP-autentikoinnin käytön, mutta tätä emme ottaneet käyttöön.

Koska Cacti käyttää SNMP:tä ja tämä on tietoturvan kannalta potentiaalisesti riskialtis protokolla, kannattaa harkita SNMP v3:n käyttöä. Lisäksi, koska Cacti vain lukee tietoa laitteilta, sitä varten kannattaa laitteille luoda pelkästään lukuoikeudet sisältävä yhteisötunnus.

5.2 Käyttöönotto

Kun palvelin, jossa Cactia ajetaan, on kunnossa, voidaan alkaa miettimään muita käyttöönottoon liittyviä asioita. Kokeiluvaiheessa valvoin Cactilla parin viikon ajan runkoverkon Catalyst 6000-kytkinten, kahden Catalyst 2950:n sekä yhden Aironet 1130:n kuormitusta. Tässä testiajossa kävi ilmi, että verkon kuormitus on hämmästyttävän pientä, joten SNMP-kyselyistä aiheutuva liikenne ei ole ongelma.

Cactille on kaksi metodia kyselyiden tekemiseksi, oletuksena käytössä oleva cmd.php, sekä spine (tunnetaan myös nimellä cactid), joista jälkimmäinen on kirjoitettu ja optimoitu toimimaan niin nopeasti kuin mahdollista. Kyselynopeuden voi testata kirjoittamalla Linuxin komentokehoitteessa komennon php /var/www/cacti/poller.php kuten kuvassa 5.1.



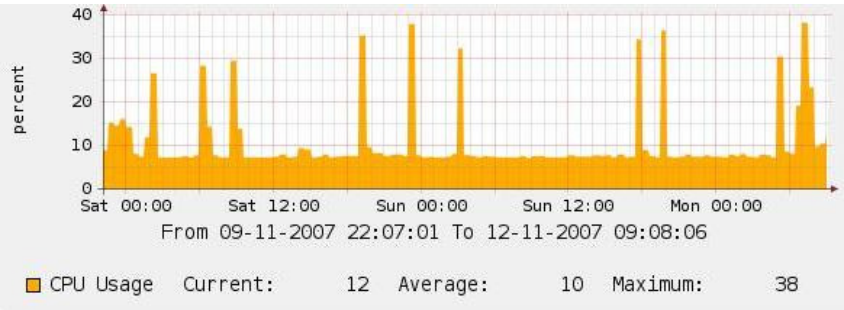
Kuva 5.1. php poller.php

Tämä ajaa käytössä olevan metodin, ja laskee siihen kuluvan keskimääräisen ajan. Tehtaalla testatessani cmd.php:llä aikaa kului noin 3 sekuntia 5 laitteen valvonnalla. Mikäli arvo on kohtuuttoman suuri tai kasvaa sellaiseksi, kannattaa käyttää spineä. Cactin dokumenttien mukaan kriittinen raja on 5 minuuttia eli 300 sekuntia, koska tämä on oletusarvoinen kyselyväli. Mikäli spineä aikoo käyttää, tulee se hakea ja asentaa erikseen joko apt-get-komennolla tai Cactin sivustolta (The Cacti Group 2007: 1.)

Totesimme myös oletusaikavälin eli viisi minuuttia kyselyille sopivaksi. Oletusarvot kuvaajille puolestaan ovat seuraavat:

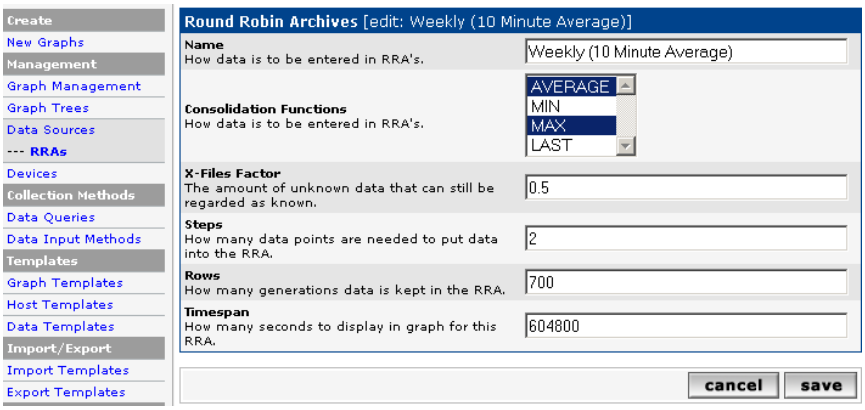
- Hourly 1 min average (ei oletuksena käytössä)
- Daily 5 min average
- Weekly 30 min average
- Monthly 2 hour average
- Yearly 1 day average

Käytännössä ohjelmaa käytetään menneen ajan tutkimiseen, ja pidempien aikavälien näytetarkkuus on melko huono, kuten kuvassa 5.2 näkyy.



Kuva 5.2. Toisen runkokytkimen prosessorin käyttö viikon ajalta tarkennettuna

Tärkein kuvaajista on viikoittainen, jotta pystytään esimerkiksi alkuviikosta tutkimaan tarkemmin, mikäli viikonlopun aikana tapahtunut jotain tavallisuudesta poikkeavaa. Siksi päätin tarkentaa tätä vastaavan RRA:n tallennusvälin 10 minuuttiin. Tämä tapahtuu hallintakonsolin Management-otsikon alta valitsemalla Data Sources ja RRAs (kuva 5.3.) Valitaan Weekly (30 Minute Average) ja muutetaan Steps oletusarvosta 6 (6 X 5 Min = 30 Min) arvoon 2 (2 X 5 Min = 10 Min). Samalla kannattaa myös nimetä RRA:n nimi uudestaan, esimerkiksi ”Weekly (10 Minute Average)”. Lisäksi kannattaa kasvattaa RRA:n kokoa, koska näytteitä tulee nyt kolme kertaa enemmän, joten muutamme Rows-kohdan arvon 700:sta 2100:n.



Kuva 5.3. Oletusaikavälien muuttaminen

5.3 Verkonvalvonta Cactilla

Koska Cactilla voi seurata melkein mitä tahansa numeerista aikasidonnaista tietoa, kannattaa hieman rajata valvottavia asioita. Tässä keskityn pääasiassa verkkolaitteiden valvontaan,

mutta ohjelmasta löytyy jo oletusasetuksina useita malleja myös Windows- ja Linux/Unix -koneille muun muassa muistin ja prosessorin kuormituksen, järjestelmään kirjautuneiden käyttäjien, sekä kiintolevytilan valvontaan.

Valvonta-OID:t

Kytkimiltä ja reitittimiltä tärkeitä valvottavia ovat vastaavanlaiset muuttujat, perässä vastaavat objektinimet ja OID:t (Cisco Systems 2007):

Prosessorin käyttöaste
cpmCPUTotal5min 1.3.6.1.4.1.9.9.109.1.1.1.1.5
cpmCPUTotal5minRev 1.3.6.1.4.1.9.9.109.1.1.1.1.8

Muistin käyttöaste
ciscoMemoryPoolUsed 1.3.6.1.4.1.9.9.48.1.1.1.5
ciscoMemoryPoolFree 1.3.6.1.4.1.9.9.48.1.1.1.6
ciscoMemoryPoolLargestFree 1.3.6.1.4.1.9.9.48.1.1.1.7

Laitteen lämpötila (Nämä toimivat vain joillakin laitteilla, esim. 6000-sarjan kytkimillä)
ciscoEnvMonTemperatureStatusValue 1.3.6.1.4.1.9.9.13.1.3.1.3
ciscoEnvMonTemperatureThreshold 1.3.6.1.4.1.9.9.13.1.3.1.4

Laitteen todellinen kuormitusaste (Nämä toimivat vain joillakin laitteilla, esim. 6000-sarjan kytkimillä)
sysTrafficPeak 1.3.6.1.4.1.9.5.1.1.19
sysTrafficPeakTime 1.3.6.1.4.1.9.5.1.1.20

WLAN-tukiasemilta (Cisco Aironet) kiinnostavaa tietoa ovat lisäksi siihen liittyneiden käyttäjien määrä sekä signaalin vahvuus:

cDot11ActiveWirelessClients 1.3.6.1.4.1.9.9.273.1.1.2.1.1
cDot11ClientSignalStrength 1.3.6.1.4.1.9.9.273.1.3.1.1.3

Lisäksi kiinnostavaa tietoa ovat tietysti tärkeimpien linkkien liikenne sisään ja ulos, sekä mahdolliset virheet, mutta nämä porttikohtaiset tiedot löytyvät ohjelmasta oletuksena. Joitakin muita näistä tiedoista puolestaan löytyy Cactin yhteisösivustolta. Lista käyttäjien tekemistä malleista löytyy osoitteesta http://forums.cacti.net/about15067.html.

Teemme valmiit mallit seuraaville laitteille käyttäen osittain apuna valmiita malleja:

Cisco Catalyst 2950/2960/3500/3550:

- CPU Usage (löytyy ohjelmasta valmiina)
- Memory Usage (löytyy ohjelmasta valmiina)
- IO Memory Usage (Cactin sivustolta Catalyst 6500 Template)

Cisco Catalyst 6000/6500 (Cactin sivustolta 6500 Template):

- CPU usage
- Temperature
- Sys Traffic Usage Peak
- IO Memory Usage

Cisco Aironet 1131 A6/1242 A6:
Associated Users (luodaan itse, tarkat ohjeet liitteessä B)

Järjestely

Tämän jälkeen luodaan looginen puurakenne johon laitteet järjestellään. Imatran Tehtailla tämä kannattaa tehdä kampusalueen eri osien mukaan esimerkiksi seuraavasti:

- Runkoverkko
- Kaukopää
- Tainiokoski
- Research Center
- InnoCentre
- Metsäkonttori
- Tietohallinto
- Virkailijakerho

Näiden alle tehdään vielä esimerkiksi tukiasemille (Access Point) oma haara ja kytkimille omansa. Kun laitteita on luotu, järjestellään ne puurakenteen alle sopiviin paikkoihin esimerkiksi laitteen nimen mukaiseen järjestykseen. Helpoin tapa tehdä tämä on lisätä Graph Trees -sivulta haluttuihin haaroihin Host -tyyppisiä otsikoita joiden alle ohjelma järjestelee kaikki kyseiselle laitteelle luodut kuvaajat automaattisesti.

Tämä ei kuitenkaan välttämättä riitä. Tukiasemilta tärkein tieto on nimenomaan se, kuinka paljon niiden kautta on WLANiin liittynyt käyttäjiä, joten kaikilta tukiasemilta kannattaa käyttäjien määrän kertovat kuvaajat järjestellä yhdelle sivulle, vaikkapa AP-otsikkonäkymään. Näin eri tukiasemien kuormitusta pystyy helposti vertailemaan ja näkee, mitkä tukiasemat ovat paljon tai liikaa käytettyjä, ja mitkä vähemmän tai ei lainkaan. Saman kuvaajan voi siirtää useaankin eri paikkaan, joten tämä ei ole ongelma.

5.4 Cactin varmistaminen ja palauttaminen

Cactin varmistaminen on kohtalaisen helppo toteuttaa ja myös tärkeää, mikäli ohjelma on tarkoitus ottaa pitkäaikaiseen käyttöön tuotantoverkossa. Helpoin tapa toteuttaa varmistaminen on ottaa käyttöön Samba File System ja liittää joku toisella koneella tai palvelimella oleva verkkojako Cacti-koneelle. Tämän jälkeen tehdään skripti, joka kopioi tarvittavat tiedot verkkojaolle, ja ajetaan tätä skriptiä halutuin väliajoin crontab-tiedoston avulla, aivan kuten Cactin poller.php:tä, muttei toki yhtä usein.

Tärkeitä varmistettavia tietoja ovat Cactin käyttämä MySQL-tietokanta, jossa sijaitsevat kaikki asetukset, kuvaajatiedot ja hallinnalliset tiedot, sekä Cactin asennushakemisto ja varsinkin siellä sijaitseva rra-hakemisto. Viimeksi mainittu tulisi varmistaa mahdollisimman usein, koska siellä oleva tieto muuttuu joka kerta kun poller.php –tiedosto ajetaan, eli oletusarvoisesti joka 5. minuutti. Sopiva tallennusväli tälle hakemistolle on esimerkiksi kerran vuorokaudessa.

Tietokannassa olevat tiedot muuttuvat aina kun asetuksia muutetaan tai uusia kuvaajia luodaan, joten sekin on hyvä varmistaa kohtalaisen usein mysqldump-komennolla. Cactin hakemisto kannattaa myös varmistaa, jotta saadaan kaikki skriptit ja asetukset (muut kuin tietokannassa olevat) talteen, mutta rra- ja log-hakemistoja lukuunottamatta siellä olevat tiedot muuttuvat todella harvoin.

Varmistusta suunniteltaessa tulee huomioida, että mikäli skriptiä ajetaan crontab-tiedoston kautta, tulee jokaista varmistettavaa osa-aluetta kohden luoda oma skripti, jos niitä aiotaan ajaa eri aikoina. Käytännössä MySQL-tietokanta ja muu Cactin asennushakemisto vievät niin vähän tilaa, että ne voidaan mainiosti varmistaa yhdellä kertaa samalla skriptilläkin. Näin tein myös Imatran Tehtailla.

Myös tietojen palauttaminen on helppoa, kunhan on vain kone jossa on kaikki Cactin vaatima valmiiksi asennettuna. Ajetaan vain mysql-komento mysqldump-komennon luomaan tiedostoon ja kopioidaan rra- ja mielellään vaikka koko Cacti-hakemisto takaisin paikoilleen. Tällöin kuvaajien piirtäminen jatkuu automaattisesti. Tarkat ohjeet varmistuksesta, varmistusskriptistä ja palauttamisesta on liitteessä C.

6 Yhteenveto

Tämä opinnäytetyö lähti alun perin liikkeelle tehtävänä, jossa oli tarkoitus kokeilla kolmea vapaan lähdekoodin ohjelmaa, MRTG:tä, RRDtoolia ja Cactia, ja arvioida näiden soveltuvuutta suuren verkkoympäristön valvontaan. Varsin pian kuitenkin kävi ilmi, että Cacti itse asiassa käyttää RRDtoolia, ja että se on todella ylivertainen verrattuna MRTG:hen tai pelkkään RRDtooliin monipuolisuutensa, graafisen käyttöliittymänsä ja ylipäätään kaiken suhteen.

Itselleni kyseessä oli ennen kaikkea opettelutehtävä. Tähän kuului itselleni ennestään enemmän tai vähemmän tuntemattomat Linux, MySQL, Apache, SNMP ja lähes kaikki mitä työssä tuli vastaan. Sellaisena se olikin todella monipuolinen ja opettavainen tehtävä.

Lisäksi tämä oli ensimmäinen kerta kun kunnolla tutustuin vapaan lähdekoodin tarjontaan. Sieltä löytyy todella monipuolisia ohjelmia lähes kaikkiin mahdollisiin käyttötarkoituksiin, mutta ohjelmien käyttöönotto voi olla hieman vaikeampaa kuin vastaavien kaupallisten ohjelmistojen. Ne myös voivat vaatia opettelua huomattavasti enemmän kuin esimerkiksi Microsoftin helppokäyttöisyyteen panostavat ohjelmat. Mutta se on usein vaivan arvoista.

Koska Cactia voi viritellä ja laajentaa lähes loputtomiin, eikä pelkästään SNMP-tietoja tallentavaksi, olisi opinnäytetyöstäkin saanut loputtoman laajan. Lopetin kuitenkin työn kun toimeksiantajan toiveet tulivat toteutettua.

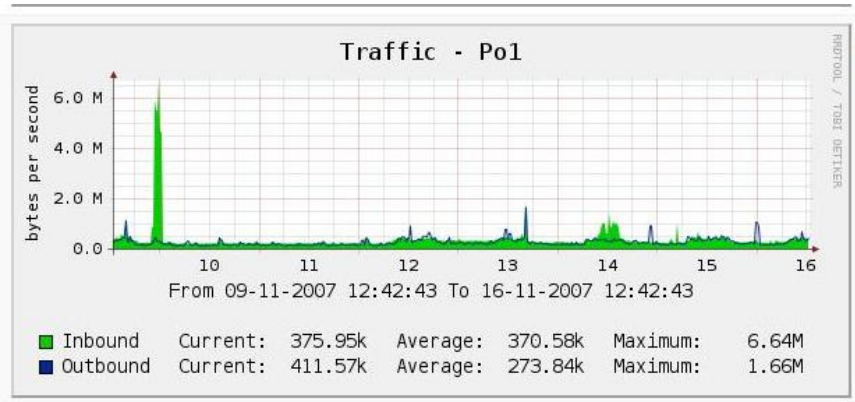
6.1 Johtopäätökset

Cacti on loistava esimerkki siitä mitä vapaa lähdekoodi parhaimmillaan tarjoaa. Se on todella monipuolinen mutta silti laitteistovaatimuksiltaan ja resurssien kulutukseltaan minimaalinen ohjelma verkonvalvontaan. Sitä myös parannetaan, laajennetaan ja päivitetään jatkuvasti. Täytyy kuitenkin muistaa että Cacti ei ole NMS. Se on loppujen lopuksi varsin rajoittunut ohjelma, mutta siinä mihin se keskityy, se on todella monipuolinen.

Vapaan lähdekoodin puolelta löytyy myös verkonhallintaohjelmia, kuten OpenNMS. Imatran Tehtailla on kuitenkin heidän laajan verkkoympäristönsä laitteiden hallintaa varten käytössään CiscoWorks, joten minkäänlaisille hallintaominaisuuksille

ei ollut tarvetta. Sen sijaan ainakaan Imatran Tehtailla käytössä olevalla versiolla CiscoWorksista ei pystytä piirtämään kuvaajia laitteiden käyttöasteesta ja kuormituksesta. Tähän tarkoitukseen puolestaan Cacti on erinomainen vaihtoehto, ja Linuxia alustana käytettäessä myös täysin ilmainen.

Kun ajoin Cactia testikäytössä viikon verran, kävi ilmi että Imatran Tehtaiden verkon kuormitus on varsin minimaalista. Esimerkiksi 3-tason runkokytkinten prosessorikuormitus kävi muutaman kerran 80 % tasolla, mutta muuten se on ollut lähes poikkeuksetta alle 20 % tasolla. Myös tärkeimpien verkkoporttien kuormitus on ollut satunnaisia piikkejä lukuun ottamatta yllättävän vähäistä. Tämä näkyy muun muassa toiselta runkokytkimeltä otetusta kuvaajasta, joka on kuvassa 6.1.



Kuva 6.1. Kahden HSRP:llä yhdistetyn runkokytkimen välinen liikenne viikon ajalta

Havaitsin testiaikana myös yhden Catalyst 3550 -kytkimen ilmeisesti käynnistyneen uudelleen yön aikana, koska kuvaajiin oli tullut kahden syklin väli. Tämä tarkoittaa sitä, että yhteys laitteeseen on katkennut. Kun yhteys palaa, jatkaa ohjelma käyrän piirtämistä välittömästi, mutta jättää kuluneen ajan verran tyhjää. Näin ohjelmalla nähdään mihin aikaan jotain outoa on saattanut tapahtua. Sen sijaan siitä puuttuu ainakin oletuksena varsinainen ominaisuus tehdä hälytyksiä, mikäli tietty raja kuormituksessa ylittyy.

6.2 Parannus- ja kehitysideoita

Sekä Debian että Cacti tukevat LDAP-autentikointia, joten Active Directoryn hyödyntäminen voisi useimmissa ympäristöissä olla hyvä idea. Imatran Tehtailla ohjelma kuitenkin tulee vain

muutaman henkilön käyttöön, joten emme katsoneet sitä tarpeelliseksi.

Koska ympäristö on suuri, ei kovin suuria muutoksia tämän ohjelman takia luonnollisestikaan aleta tekemään. Esimerkiksi SNMP:stä käytössä on versio SNMPv2. Tietoturvan takia SNMPv3 on suositeltavin versio, mutta sitä ei vielä ole otettu käyttöön. Myös uuden, pelkät lukuoikeudet sisältävän yhteisön luominen kaikille valvottaville laitteille Cactia varten olisi ollut melkoinen urakka, joten käytin samaa täydet oikeudet sisältävää yhteisöä kuin mitä muun muassa CiscoWorks käyttää.

7 Lähdeluettelo

Stora Enso Oyj 2007. [online] [viitattu 27.11.2007] www.storaenso.com

Mauro, Douglas R., Scmidt, Kevin J. 2005. Essential SNMP. O'Reilly

Peter J. Welcher 1999. Netcraftsmen. Configuring SNMP in Cisco routers [online] [viitattu 21.10.2007] http://www.netcraftsmen.net/welcher/papers/snmprouter.html

Tobi Oetiker's MRTG – The Multi Router Traffic Grapher. [online] [viitattu 1.10.2007] http://oss.oetiker.ch/mrtg/

Tobi Oetiker's About RRDTool. [online] [viitattu 14.10.2007] http://oss.oetiker.ch/rrdtool/

Alex van den Bogaert 2007. Tobi Oetiker's RRDtutorial. [online] [viitattu 8.11.2007] http://oss.oetiker.ch/rrdtool/tut/rrdtutorial.en.html

Tobi Oetiker 2007. Building RRDtool. [online] [viitattu 3.10.2007] http://oss.oetiker.ch/rrdtool/doc/rrdbuild.en.html

Lee Carter, 2004. Cacti SNMP Management Installation HOW-TO For Linux [online] [viitattu 4.10.2007] http://docs.cacti.net/node/70/

The Cacti Group 2007: 1. Spine Information. [online] [viitattu 12.11.2007] http://www.cacti.net/spine_info.php

The Cacti Group 2007: 2. Why Templates? [online] [viitattu 1.11.2007] http://docs.cacti.net/node/80

Cisco Systems, 2007. SNMP Object Navigator. [online] [viitattu 13.11.2007] http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en

Liite A: Cactin asennus

Tällä ohjeella saa asennettua Cactin puhtaalta pöydältä. Tarkoituksena on tehdä palvelin, josta on karsittu työpöytä ja huomioitu pääasiassa vain Cactin toimivuus. Halutessa voi toki asentaa muutakin, sillä useimmissa koneissa resurssit riittävät varmasti myös muuhun käyttöön kuin vain Cactin ajoon. Tässä ohjelma asennetaan Debian 4.0 Linuxille. Asennusprosessi myös muille Linux-versioille on pääpiirteittäin samanlainen, mutta vaaditut paketit ja niiden nimet voivat olla hieman erilaisia. Cactin voi asentaa myös apt-get-ohjelmalla, mutta tätä ei kannata tehdä, koska käyttöjärjestelmä ei osaa tehdä Cactin vaatimia käyttäjätunnuksia ja tietokanta-asetuksia.

Asennetaan Debian 4.0 tekstimuodossa koneelle. Asennusmediaksi riittää mainiosti minimaalinen verkkoasennuslevyimage. Oletusasetuksilla web-palvelin on /var/www -hakemistossa, johon myös Cacti ja sen keräämä data tullaan asentamaan. Mikäli aikomuksena on osioida levy oletusasetuksista poikkeavaksi, kannattaa tämä huomioida.

Koneelle ei kannata antaa DHCP:n kautta osoitetta, vaan määritellään verkkoasetukset käsin. Package mirroriksi voi laittaa vaikkapa Jyväskylän Yliopiston palvelin ftp.jyu.fi. Otetaan asennettavista paketeista pois kaikki paitsi Web Server ja Standard System. Grubin voi myös asentaa.

Asennuksen jälkeen kun kone käynnistetään uudelleen, käynnistyy tietoturva- ja muiden asennusten konfigurointiohjelma. Sallitaan palomuurista HTTP-, HTTPS- ja SSH-liikenne sisään, ja laitetaan palvelut httpd ja mysqld käynnistymään automaattisesti.

Asennuksen jälkeen kirjaudutaan root-käyttäjällä ja haetaan järjestelmään päivitykset komennolla

apt-get upgrade

Tämän jälkeen asennetaan Cactin tarvitsemat löytyviä kirjastot ja paketit komennolla

apt-get install **paketin_nimi** [**paketin_nimi2 paketin_nimi3** ...]

Seuraavat paketit pitäisi jo olla asennettuna. Mikäli näin on, ilmoittaa asennuskomento että kyseinen paketti on jo uusin versio:

- zlib1g
- perl
- libpng12-0
- libfreetype6

Asennetaan seuraavaksi seuraavat paketit:

- libgd2-xpm
- libart-2.0-2
- mysql-server
- php5
- php5-mysql
- php5-gd
- php5-cli
- snmp
- snmpd
- php5-snmp
- libsnmp-perl
- rrdtool
- librrds-perl
- dbconfig-common
- libphp-adodb
- php5-pgsql
- php4-pgsql
- php5-sybase
- php4-sybase
- php5-odbc
- php4-odbc
- apache2

Nämä asentavat muitakin paketteja, joten kun käyttöjärjestelmä kysyy, halutaanko ne asentaa, vastataan kaikkiin kyllä. Seuraavaksi käynnistetään kone uudelleen.

Luodaan käyttäjäryhmä cacti, siihen käyttäjä cactiuser ja MySQL:ään tietokanta cactidb. Asennetaan myös MySQL:ään root-käyttäjälle mielellään vahva salasana:

groupadd **cacti**
useradd -g **cacti cactiuser**
mysql
mysql> set password for root@localhost=password(‘**salasana**’);
mysql> create database **cactidb**;
mysql> grant all on **cactidb**.* to **cactiuser**;
mysql> grant all on **cactidb**.* to **cactiuser**@localhost;
mysql> set password for **cactiuser**@localhost=password('**cactipw**');

Seuraavaksi haetaan cacti ja puretaan se hakemistoon /var/www/cacti:

```
cd /var/www/
wget http://www.cacti.net/downloads/cacti-0.8.7.tar.gz
tar -zxvf cacti-0.8.7.tar.gz
mv cacti-0.8.7 cacti
rm cacti-0.8.7.tar.gz (poistaa paketin)
```

ajetaan cactiuser-käyttäjällä cacti-hakemistosta löytyvä cacti.sql joka luo cactidb-tietokantaan sisällön:

```
cd /var/www/cacti
mysql -u cactiuser -p cactidb < cacti.sql
```

Cactiuser -käyttäjä tarvitsee omistusoikeudet Cacti-hakemiston hakemistoihin rra ja log:

```
chown -R cactiuser rra log
chgrp -R cacti rra log
```

Include-hakemistosta löytyy tiedosto config.php, johon laitetaan tiedot tietokannasta, käyttäjtilistä ja salasanasta kuntoon:

```
nano /var/www/cacti/include/config.php
```

```
$database_default = "cactidb";
$database_hostname = "localhost";
$database_username = "cactiuser";
$database_password = "cactipw";
```

Käyttäjä cactiuser ajaa tiedostoa poller.php joten laitetaan tämä menemään ajastettuna lisäämällä /etc/crontab -tiedostoon rivi:

```
*/5 * * * * cactiuser php /var/www/cacti/poller.php > /dev/null 2>&1
```

Tällä asetuksella tiedosto ajetaan viiden minuutin välein. Tätä kannattaa toki muuttaa tilanteen ja tarpeiden mukaan.

Sitten mennään selaimen kautta (toiselta koneelta) osoitteeseen esim. 10.10.10.10/cacti josta asennus jatkuu graafisesti. Ohjelman pitäisi löytää kaikki haluamansa ohjelmat oletuspaikoista. Mikäli ne on asennettu muualle, kerrotaan sijaintipolku tässä. Tämän jälkeen Cacti on valmis käyttöön.

Isäntäkoneen hallintaa varten kannattaa asentaa myös SSH-palvelu etäyhteyttä varten (apt-get install ssh) sekä Samba File

System varmistusta varten (apt-get install smbfs). Cactin varmistuksesta on kerrottu tarkemmin liitteessä C.

Liite B: Mallien tekeminen

Jotta Cactilla voidaan piirtää kuvaajia muistakin kuin ohjelman valmiiksi tukemista asioista, täytyy ensin luoda tietotyyppimalli (Data Template) jossa määritellään muun muassa tiedonhakutapa, tietolähdetyyppi, ja SNMP:n ollessa kyseessä haluttua tietoa vastaava OID, sekä mitkä RRA:t tietotyyppiin liitetään. Tämän jälkeen luodaan kuvaajamalli (Graph Template), jossa tietotyyppimallia tai -malleja käytetään. Näitä voi olla useita samassa kuvaajamallissa. Tämän jälkeen kuvaajamallien mukaan voi luoda kuvaajia jo luoduille laitteille laitteille ja/tai luoda laitemalleja (Host Template) joihin yhdistetään automaattisesti tietyt kuvaajamallit kun uusi laite luodaan. Cactin sivuston foorumeilta löytyy runsaasti valmiita malleja lähes jokaiseen käyttötarpeeseen. Nämä ovat lähes poikkeuksetta laitemalleja, jotka luovat samalla niin tietotyyppi- kuin kuvaajamalleja.

Ennen kuin aletaan luoda uutta SNMP-tietotyyppimallia, kannattaa snmpwalk-komennolla kokeilla, että kyseinen OID toimii. Mikäli kyseessä on Ciscon laite, kannattaa myös varmistaa että laitteissa, joita aiotaan valvoa, on sellainen versio IOS:ista joka tukee kyseistä OID:tä. Mikäli nämä ovat väärin, Cacti kyllä voi luoda kuvaajan mutta siihen ei ilmesty mitään käyriä.

Ciscon sivustolta löytyy SNMP Object Navigator (tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en), jolla voi etsiä halutunlaisia tietotyyppejä, tutkia SNMP -puurakennetta, ja kääntää OID:t objektinimiksi ja päinvastoin. Tämä on erinomainen työkalu tietotyyppien luontia helpottamaan.

Perussyntaksi snmpwalk -komennolle on seuraava:

snmpwalk –Os **IP_tai_FQDN** –c **yhteisö** –v **versio OID**

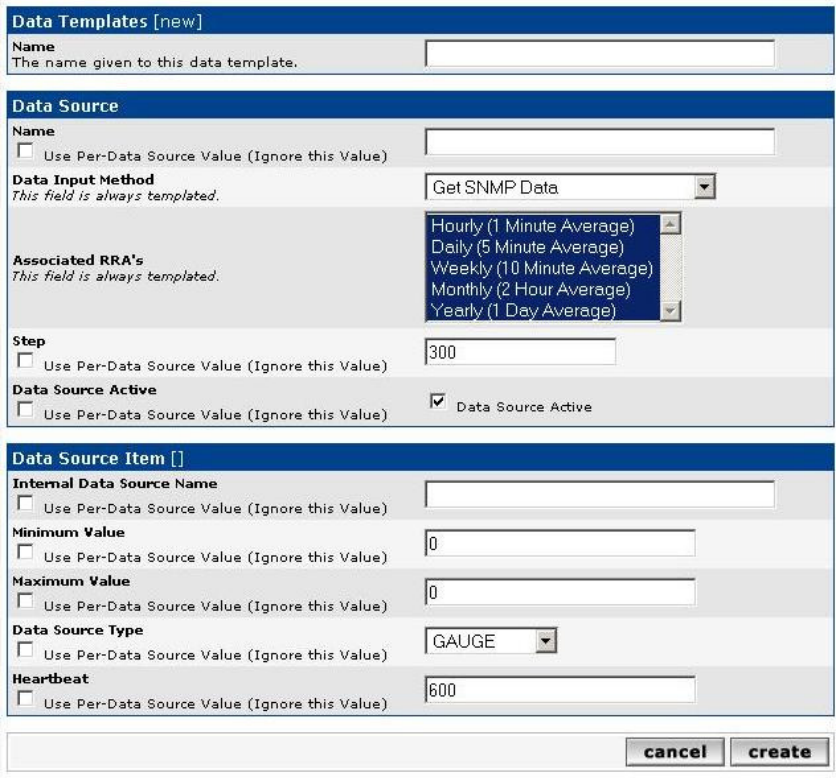
jossa IP_tai_FQDN on halutun laitteen IP-osoite tai DNS-nimi, yhteisö SNMP-yhteisönimi, versio SNMP-versionumero (Huom. SNMPv2 on tässä 2c) ja OID haluttu SNMP OID-numerosarja. SNMPv3 vaatii vielä lisäoptioita.

Esimerkiksi seuraava komento hakee laitteelta 1.2.3.4 laitteen kautta WLANiin liittyneiden käyttäjien määrän (Cisco Aironetin tukema OID):

```
snmpwalk –Os 1.2.3.4 –c public –v 2c 1.3.6.1.4.1.9.9.273.1.1.2.1.1
enterprises.9.9.273.1.1.2.1.1.1 = Gauge32: 6
enterprises.9.9.273.1.1.2.1.1.2 = Gauge32: 0
```

Tuloksesta nähdään, että haku toimii, ja että kyseinen OID on Gauge -tyyppinen. Tämä on Cactin tyyppisen valvonnan kannalta hyödyllisin tietotyyppi, sillä sen arvo muuttuu jatkuvasti molempiin suuntiin. Cactin valmiiksi tehdyistä tietotyyppimalleista kaikissa käytetään Gauge-tyyppistä tietolähdettä, ainoastaan prosessorikuormaa mittaavissa käytetään Counter-tyyppistä tietolähdettä. Lisäksi tuettuna on vielä Derive ja Absolute. Esimerkiksi Absolutea voidaan käyttää INT-tyyppisen arvon seuraamisessa mutta se on käytännössä pelkkää suoraa viivaa, joka saattaa muuttua lähinnä uudelleenkäynnistyksen yhteydessä.

Luodaan Cisco Aironet -tukiasemia varten tietotyyppimalli, joka seuraa tukiasemaan liittyneiden käyttäjien määrää. Ensin valitaan hallintakonsolista Data Templates ja Add.



Kuva B.1 Data Templaten luominen

Kirjoitetaan Otsikkokenttään Name haluttu nimi mallille, esimerkiksi Cisco Aironet – Associated Users. Saman voi laittaa Tietolähteen (Data Source) nimeksi. Tiedonhakumetodi on Get SNMP data, Associated RRA:s kaikki paitsi ensimmäinen. Internal Data Source Name voi olla mitä tahansa. Jos lopulliseen kuvaajaan halutaan useita käyriä, tällä erotetaan ne. Suu-

rimmaksi arvoksi kannattaa laittaa 0, sillä tämä tarkoittaa, ettei ylärajaa ole. Cacti jättää pois muuten kaikki tähän asetettua arvoa suuremmat arvot (The Cacti Group 2007: 2.)

Tietolähdetyyppi on siis Gauge, muut asetukset voidaan jättää oletusarvoiksi. Kun nyt painetaan Create, tulee lisäoptioita, joista tärkein on OID. Tähän laitetaan 1.3.6.1.4.1.9.9.273.1.1.2.1.1. Muihin ei tarvitse laittaa mitään. Tämän jälkeen tallennetaan, jonka jälkeen voidaan tarkistaa että luotu malli näkyy Data Templates-listassa.

Seuraavaksi luomme kuvaajamallin joka käyttää äsken luomaamme tietotyyppimallia. Vastaavalla tavalla valitsemme Graph Templates ja Add. Mallin nimeksi voi jälleen laittaa Cisco Aironet - Associated Users. Kuvaajan nimeksi kannattaa laittaa esimerkiksi ympäristömuuttujalla |host_description| - Associated Users jolloin tiedetään laitteen nimi. Jos kuvaajasta haluaa oletusarvoista 500 pikseliä leveämmän, voi sitä muuttaa. Ihan alhaalta löytyy Vertical Label, johon voidaan laittaa esimerkiksi number of users. Muihin ei tarvitse laittaa mitään. Tallennetaan.

Nyt ylimmäisestä otsakkeesta Graph Template Items voi lisätä käyriä kuvaajaan painamalla Add. Data Source -valikosta pitäisi löytyä äsken luomamme Cisco Aironet – Associated Users tietotyyppi, jonka valitsemme. Sitten valitaan haluttu väri ja käyrätyyppi. Koska tämä on ainoa käyrä kuvaajassa, kannattaa selkeyden takia valita Area jolloin kuvaajaan maalataan alue käyrän alapuolelta.

Text Format -kohtaan laitetaan valittua väriä vastaava selitys, esimerkiksi Aironet Users. Tämän jälkeen tallennetaan ja nähdään että kuvaajamalliin tuli samalla myös Graph Item Inputs –otsikon alle tietolähde. Nyt malli on valmis käytettäväksi.

Laitemallin luonti on todella helppoa. Valitaan Host Templates, Add ja annetaan laitteelle nimi, esimerkiksi Cisco Aironet. Tallennetaan jonka jälkeen valitaan tähän malliin (tai profiiliin) linkitetyt kuvaajat ”Associated Graph Templates” otsikon alta. Kun halutut kuvaajat on valittu ja lisätty Add -napista, tallennetaan jonka jälkeen malli on valmis käytettäväksi. Tämänkään jälkeen kuvaajia ei kuitenkaan luoda automaattisesti laitteita luodessa, vaan ne tulee valita yksitellen laitteen luonnin jälkeen Graph Templates -otsikon alta.

Liite C: Cactin varmistaminen ja palauttaminen

Cactin varmistaminen ja palauttaminen on tärkeää ja myös kohtalaisen helppoa. Varmistettavia tietoja ovat Cactin tietokanta MySQL:ssä, ja Cactin asennushakemisto, varsinkin siellä oleva hakemisto /rra. Tämä tulisi varmistaa mahdollisimman usein, koska rra-tiedostoissa oleva tieto muuttuu jatkuvasti.

Suositeltavaa on kuitenkin varmistaa samalla koko Cactin hakemisto, jotta kaikki asetukset, skriptit ja muu siellä oleva data saadaan mukaan. Muut asennushakemiston tiedot rra:ta lukuunottamatta vievät noin 5 MB tilaa. Nämä tosin muuttuvat harvoin, joten jos haluataan minimoida varmistettavan datan määrä, ne voi toki varmistaa erikseenkin esimerkiksi kerran kuukaudessa, mutta tämä täytyy tehdä erillisellä skriptillä.

Paikka, johon varmistettavat tiedot tallennetaan, olisi tietysti hyvä olla toisella koneella. Jos kohdekone on Windows-kone, tarvitaan Samba File System -moduuli, jonka jälkeen voidaan liittää Windows-koneella oleva hakemisto Cactia ajavalle Linux-koneelle. Luodaan etäkoneelle vaikkapa hakemisto **cactibackup**, jaetaan se, ja luodaan käyttäjätunnus jolle annetaan hakemistoon täydet luku- ja kirjoitusoikeudet.

Tämän jälkeen asennetaan SMBFS komennolla

```
apt-get install smbfs
```

Laitetaan se käynnistymään automaattisesti aina uudelleen käynnistettäessäkin:

```
echo ’smbfs’ >> /etc/modules
```

Sitten luodaan hakemisto johon Windows-palvelimella oleva hakemisto liitetään, esimerkiksi:

```
mkdir -p /mnt/kohdekone
```

Tietoturvasyistä kannattaa tehdä erillinen tiedosto, jossa on käyttäjätunnus ja salasana etäkoneelle. Tehdään tämä vaikkapa /etc-hakemistoon tiedostoon nimellä kohdekone.smbpass:

```
cat > /etc/kohdekone.smbpass << EOF
username=cactibackup
password=cactipw
EOF
```

Tässä siis cactibackup on Windows-etäjärjestelmään luotu käyttäjätunnus ja cactipw sen salasana. Muutetaan tiedoston oikeudet siten, että vain root-käyttäjällä on oikeus lukea sitä:

chmod 600 /etc/**kohdekone.smbpass**
chown root.root /etc/**kohdekone.smbpass**

Sitten lisätään /etc/fstab -tiedostoon seuraava rivi, joka liittää hakemiston automaattisesti konetta käynnistettäessä, ja lukee tarvitsemansa oikeudet juuri luodusta tiedostosta:

//kohdekone/cactibackup /mnt/**kohdekone** smbfs defaults,credentials=/etc/**kohdekone.smbpass** 0 0

Tämän jälkeen ajetaan seuraava komento, joka kannattaa myös lisätä skriptin alkuun:

mount /mnt/**kohdekone**

Nyt kun etäkoneelle voidaan kopioida tiedostoja, tehdään skripti joka tekee paketin sekä MySQL-tietokannasta, että Cactin hakemistosta. Cactin foorumilta löytyy valmis skripti, jota muokkaamalla saadaan nämä tehtyä. Seuraavassa on muokattu versio, #-merkillä alkavat rivit voi laittaa skriptiin kommenteiksi.

#!/bin/bash

#Määritellään aluksi muutamia ympäristömuuttujjia:

mount /mnt/**kohdekone**

BACKUPDIR=/mnt/**kohdekone**
BASEDIR=/var/www/
MSDUMP=/usr/bin/mysqldump
CACTIDB=**cactidb**
DBUSER=**cactiuser**
DBPASS=**cactipw**

#Nyt voidaan näitä käyttäen ajaa seuraavat komennot, ensiksi #Cacti-hakemiston ja seuraavaksi MySQL-tietokannan #tallennus:

cd $BASEDIR
tar -zcvf $BACKUPDIR/cacti.`date +%d%m%y`.tgz cacti/

$MSDUMP --add-drop-table $CACTIDB -u$DBUSER -p$DBPASS > $BACKUPDIR/cacti.`date +%d%m%y`.sql

#Skriptiin voidaan myös lisätä seuraavat rivit, joka poistaa #(tässä esimerkissä) yli 31 päivää vanhat tiedot, jottei tilaa kulu #liikaa:

```
find $BACKUPDIR/ -name '*.tgz' -mtime +31 | xargs rm
find $BACKUPDIR/ -name '*.sql' -mtime +31 | xargs rm
```

#Skripti päättyy tähän

Kun nämä komennot tallentaa esim. bash-skriptiksi nimeltä cactibackup.sh jonka voi tallentaa vaikka cacti-hakemistoon, ja laittaa crontab-tiedostoon suoritettavaksi halutulla välillä, alkaa ohjelma tekemään tallennuksia. Crontab-rivi on tällöin seuraava, joka päivä klo 23 tehtävällä tallennuksella:

```
* 23 * * * root /var/www/cacti/cactibackup.sh
```

Tuloksena on siis kaksi tiedostoa, päivämäärästä riippuen esim. cacti.02.11.07.tgz ja cacti.02.11.07.sql. On tärkeää huomata, että tässä esimerkissä tiedostot eroitellaan toisistaan päivämäärän avulla. Jos varmistus halutaan ottaa useammin kuin kerran päivässä, tulisi tiedostot nimetä lisäksi vielä esim. kellonajan mukaan, tai skripti ylikirjoittaa vanhojen tietojen päälle aina samoihin kyseistä päivää oleviin tiedostoihin. Mikäli tiedot rra-hakemistosta jostain syystä tuhoutuvat, skripti siis kirjoittaa vanhojen tietojen päälle tyhjää jolloin kaikki (kyseisen päivän) tiedot menetetään.

Palauttaminen on helppoa, jos kohdekoneella on kaikki Cactia varten valmiina, eli oma tietokanta täysillä oikeuksilla omalle käyttäjälle ja kaikki vaaditut kirjastot asennettuna, sekä crontab-tiedostoon ajastettu poller.php-komennon suoritus. Periaatteessa Cactin asetuksineen ja tietokantoineen päivineen voisi myös siirtää samalla tavalla koneelta toiselle, mutta tässä tapauksessa kannattaa ensin asentaa ohjelma koneelle toimintakuntoon ja vasta sen jälkeen palauttaa tiedot sinne.

Tietokanta voidaan luoda helposti uudelleenkin seuraavilla komennoilla MySQL-kehoitteessa:

```
create database cactidb;
grant all on cactidb.* to cactiuser;
grant all on cactidb.* to cactiuser@localhost;
set password for cactiser@localhost=password('cactipw');
```

Tämän jälkeen puretaan cacti.02.11.07.tgz komennolla

```
cd /var/www/html
```

tar -zxvf cacti.02.11.07.tgz

SQL-tietokanta puolestaan palautettaan komennolla

mysql -u **cactiuser** -p **cactidb** < cacti.02.11.07.sql

Crontab-rivi on seuraava:

*/5 * * * root php /var/www/cacti/poller.php > /dev/null 2>&1

Tämän jälkeen ohjelma alkaa taas ajamaan itseään, päivittäen tilanteen kahden aikasyklin aikana. Toimivuus kannattaa toki vielä tarkistaa kirjautumalla ohjelmaan ja testaamalla.