STADIA
HELSINGIN AMMATTIKORKEAKOULU

TEKNIIKAN JA LIIKENTEEN TOIMIALA

Tietotekniikka

Tietoliikennetekniikka

INSINÖÖRITYÖ

TIETOVERKON VALVONTA SNMP-PROTOKOLLAN AVULLA

Työn tekijä: Lauri Jurvanen
Työn valvoja: Jukka Louhelainen
Työn ohjaaja: Jouni Meriläinen

Työ hyväksytty: __. __. 2008

Jukka Louhelainen
lehtori

STADIA
HELSINGIN AMMATTIKORKEAKOULU

ALKULAUSE

Tämä insinöörityö tehtiin Netcontrol Oy:lle, joka tarjosi mahdollisuuden insinöörityön tekemiseen. Kiitän kaikkia projektissa mukana olleita. Erityisesti haluan kiittää työn ohjaajaa tiiminvetäjä Jouni Meriläistä sekä työn valvojaa lehtori Jukka Louhelaista kannustavasta ohjauksesta työn aikana. Lisäksi kiitän projektipäällikkö Hannu Hirvistä rakentavista ideoista ja keskusteluista. Myös koko Netcontrol Oy:n henkilökunta ansaitsee kiitokset avusta ja tuesta.

Helsingissä 17.4.2008

Lauri Jurvanen

STADIA
HELSINGIN AMMATTIKORKEAKOULU

INSINÖÖRITYÖN TIIVISTELMÄ

Tekijä: Lauri Jurvanen

Työn nimi: Tietoverkon valvonta SNMP-protokollan avulla

Päivämäärä: 17.4.2008 Sivumäärä: 44 s.

Koulutusohjelma: Tietotekniikka Suuntautumisvaihtoehto: Tietoliikennetekniikka

Työn valvoja: lehtori Jukka Louhelainen

Työn ohjaaja: tiiminvetäjä Jouni Meriläinen

Tietoverkkojen valvontajärjestelmiä on nykyään käytössä monissa eri verkkoympäristöissä. Näissä verkoissa käytetään useiden laitevalmistajien tuotteita, joita täytyy pystyä valvomaan. Tähän tarkoitukseen on kehitetty verkonhallintaprotokolla SNMP (Simple Network Management Protocol). Tässä työssä valvontaprotokollaa on tarkoitus soveltaa tietoverkkojen valvontaan.

Tämä insinöörityö tehtiin osana Netcontrol Oy:n tuotekehitystä. Työn päätavoitteena oli kehittää toimiva ratkaisu verkonvalvontaan. Työssä tutkittiin myös verkonvalvonnan toteuttamista SNMP-protokollan avulla sekä eri verkonvalvonnan toteutusmalleja.

Työn teoriaosuus tehtiin kirjallisuustutkimuksena. Teoriaosuudessa käydään läpi verkonvalvonnan taustoja, SNMP-protokollan eri versioiden ominaisuuksia ja tietokantojen rakennetta. Työn loppuosassa kerrotaan Windows-käyttöjärjestelmän SNMP-ohjelmista sekä avoimen lähdekoodin Net-SNMP-ohjelmasta. Lisäksi käydään läpi näiden ohjelmien käyttöönotto. Lopussa kerrotaan myös testijärjestelmän tuloksista yleisluontoisesti.

Työ osoittaa, että SNMP:n avulla pystytään suorittamaan kattavaa verkonvalvontaa. SNMP:n toiminnallisuus saadaan aikaan työssä esitetyillä ohjelmistoilla eri tietoverkon laitteisiin. Laitteiden valvontaa pystytään suorittamaan SNMP-hallintaohjelmalla. Näiden ohjelmistojen avulla saadaan aikaan toimiva päästä päähän -ratkaisu verkonvalvontaan. Tarkemmat kuvaukset testijärjestelmistä ja kehitysmahdollisuuksista on sisällytetty liitteeseen, joka on luottamuksellinen.

Avainsanat: Verkon valvonta, SNMP, MIB, Net-SNMP

STADIA
HELSINGIN AMMATTIKORKEAKOULU

ABSTRACT

Name: Lauri Jurvanen

Title: Information Network Monitoring with SNMP protocol

Date: 17 April 2008 Number of pages: 44

Department: Information technology Study Programme: Telecommunications

Instructor: Jukka Louhelainen, Senior Lecturer

Supervisor: Jouni Meriläinen, Team Leader

Different kinds of network monitoring systems are currently used in a broad range of networks. These networks utilize many types of equipment that require monitoring. SNMP (Simple Network Management Protocol) has been developed for this particular purpose. The purpose of this study was to apply this management protocol to information network monitoring.

This study was carried out for Netcontrol Oy as a part of a research and development project. The main goal was to create an efficient solution to network monitoring. The study explores the implementation of network monitoring using SNMP and different ways to achieve this.

This study is based on the theory of network monitoring with particular focus on the properties of diverse SNMP versions and the structure of management information. The latter part of this study deals with SNMP programs operating on Windows as well as the open source program Net-SNMP. In addition, the implementation of these programs is covered along with a brief description of the test systems.

This study indicates that comprehensive monitoring can be accomplished with SNMP protocol. The functionality of SNMP can be achieved for many types of hardware with the programs illustrated in this study. The monitoring of the equipment can be carried out through the network monitoring program. The programs covered in this study can be used to provide an end-to-end solution to network monitoring. Detailed descriptions of the test systems along with future development plans can be found in the appendix, which is classified, however, as confidential.

Keywords: SNMP, Network Monitoring, MIB, Net-SNMP

SISÄLLYS

LYHENTEET JA MÄÄRITELMÄT

AES	Advanced Encryption Standard; nykyaikainen, tiedon salaukseen käytetty standardi.
ASN.1	Abstract Syntax Notation One; standardi, joka määrittelee tiedon esitysrakenteen, koodauksen, siirron ja vastaanoton.
CBC	Cipher Block Chaining; tiedon salauksessa käytetty salauslohkojen ketjutustekniikka.
CFB	Cipher Feedback; tiedon salauksessa käytetty salauksen muodostustekniikka.
CMOT	Common Management Information Protocol over TCP; historiallinen, OSI-malliin pohjautuva varmistettu verkonhallintaprotokolla.
DES	Data Encryption Standard; tiedon salausstandardi.
HEMS	High-level Entity Management System; historiallinen, verkonhallinnassa käytetty protokolla.
HMAC	Hash Message Authentication Code; tiedon todennukseen käytetty koodi.
ICMP	Internet Control Message Protocol; protokolla, jota käytetään internetissä yhteyksien havaitsemiseen ja virheilmoituksiin.
MD5	Message Digest; algoritmi, jolla pystytään muodostamaan esimerkiksi todennuksessa käytetty luku.
MIB	Management Information Base; yleisnimitys tietokannalle, joka sisältää verkonhallinta informaatiota.
OID	Object Indentifier; tunniste, tietokannan objektien tunnistamiseen.
OSI	Open Systems Interconnection; kansainvälisten standardisointiorganisaatioiden julkaisema standardi tietoverkoille ja protokollille.
PDU	Protocol Data Unit; tiedonsiirrossa käytetty informaatiolohko.
RMON	Remote Network Monitoring; tietokanta, jolla voidaan suorittaa tietoverkon valvontaa.
SGMP	Simple Gateway Management Protocol; historiallinen, verkonhallinnassa käytetty protokolla.
SHA-1	Secure Hash Algorithm; algoritmi, jolla pystytään muodostamaan esimerkiksi todennuksessa käytetty luku.

SMI	Structure of Management Information; standardi, joka määrittelee miten verkonhallintatietokanta muodostetaan.
SNMP	Simple Network Management Protocol; nykyaikainen verkonhallintaprotokolla.
TCP	Transmission Control Protocol; yhteydellinen, tiedonsiirtoon käytetty protokolla.
UDP	User Datagram Protocol; yhteydetön, tiedonsiirtoon käytetty protokolla.
USM	User-Based Security Model; käyttöoikeusmalli, jolla voidaan toteuttaa oikeudet verkonhallintatietokantaan.
VACM	View-Based Access Control Model; käyttöoikeusmalli, jolla voidaan toteuttaa oikeudet verkonhallintatietokantaan.

1 JOHDANTO

Tietoverkkojen valvontajärjestelmiä on nykyään käytössä monissa eri verkkoympäristöissä. Näissä verkoissa käytetään useiden laitevalmistajien tuotteita, joita täytyy pystyä valvomaan. Tähän tarkoitukseen on kehitetty verkonhallintaprotokolla SNMP (Simple Network Management Protocol). SNMP:n avulla pystytään toteuttamaan monipuolisia valvontajärjestelmiä. Protokolla pystyy toimimaan monissa eri ympäristöissä, joten sen soveltuvuus eri käyttötarkoituksiin on lähes rajaton.

Tässä työssä SNMP-protokollaa on tarkoitus soveltaa tietoverkkojen valvontaan. Maantieteellisesti hajautettujen prosessien valvonta on jo pitkään toteutettu hyvin monella eri tekniikalla. Tämän vuoksi verkonvalvonnan toteuttaminen näissä järjestelmissä on ollut hankalaa. Tästä johtuen kaukokäyttöjärjestelmien suunta on ollut kohti IP-pohjaista verkonvalvontaa. Nykyään IP-pohjainen liikenne on saanut jalansijaa myös kaukokäytössä. Kattava IP-pohjainen verkonvalvonta on kuitenkin vielä suurimmalta osin toteuttamatta.

Tämä insinöörityö tehtiin osana Netcontrol Oy:n tuotekehitystä. Työssä tutkittiin verkonvalvonnan toteuttamista SNMP-protokollan avulla sekä eri verkonvalvonnan toteutusmalleja. Työn osana tutkittiin myös SNMP-agenttien toimintaa eri käyttöjärjestelmissä. Työssä keskitytään pääasiassa yrityksen sisäisen IP-verkon valvontaan.

Teoriaosuudessa käydään läpi verkonvalvonnan taustoja ja itse valvontaprotokollaa. Lisäksi perehdytään SNMP-protokollan eri versioiden ominaisuuksiin sekä niiden tietoturvaominaisuuksiin. Työn loppupuolella kerrotaan käytännön testaustuloksista yleisluontoisesti. Tarkemmat kuvaukset testijärjestelmistä on sisällytetty liitteeseen, joka on luottamuksellinen.

1.1 Netcontrol Oy

Netcontrol Oy on maantieteellisesti hajautettujen prosessien valvonta- ja ohjausjärjestelmiin erikoistunut yritys. Yhtiön tuotteisiin kuuluvat mm. käyttökeskukset, ala- sekä erotinasemat. Näiden keskusten ja asemien välille on lisäksi tarjolla useita eri tyyppisiä liikennöintiratkaisuja. Nämä ratkaisut tukevat runsaasti liikennöintiprotokollia automaatio- ja prosessitekniikan aloilta. Tuotteita löytyy myös protokollien analysoitiin, sekä erilaisiin keskittimiin liittyviä tuotteita.

Yhtiö toimii tällä hetkellä Pohjoismaissa. Pääkonttori sijaitsee Helsingin Pitäjänmäellä. Netcontrol Oy:llä on lisäksi tytäryhtiö Norjassa ja sivuliike Ruotsissa. Yhtiö työllistää tällä hetkellä Suomessa noin 35 henkeä.

1.2 All-IP-konsepti

All-IP-konseptin ajatuksena on, nimensä mukaisesti, kaiken informaation siirtäminen IP-verkon avulla. Tällä hetkellä sähkönvalvontaverkoissa on käytetty paljon eri tekniikoita, joiden yhteensovittaminen on hankalaa. Aikaisemmin tarvittiin vain hitaita siirtolinkkejä, koska valvottavia elementtejä ei ollut niin paljon. Nykyään kuitenkin laitteiden määrän ja lisätoimintojen takia yhteyksien pitää olla nopeita ja joustavia. Näiden asioiden vuoksi IP-verkkoihin siirtyminen on ollut luontevaa.

Konseptin tuottaminen on tuonut kuitenkin uusia ongelmia. IP-verkkoon pääsee helposti käsiksi, joten siirrettävän informaation pitäisi olla salattua. Ainoastaan pienissä sisäverkoissa pystytään käyttämään salaamatonta tietoa, luonnollisesti palomuurien takana. Myös verkon rakenteen on oltava varmalla pohjalla. Esimerkiksi yhden laitteen tai siirtolinkin pettäminen ei pitäisi aiheuttaa paikallista suurempaa tietosiirtokatkosta. Näiden ominaisuuksien käyttöönotto tuo kuitenkin huomattavasti lisäkustannuksia. Jos taas tingitään kustannuksista, niin riskit kasvavat.

1.3 Insinöörityön tavoitteet

Insinöörityön päätavoitteena on löytää toimiva ratkaisu verkonvalvontaan. Ratkaisussa otetaan huomioon myös tietoturvaseikat sekä jatkokehitysmahdollisuudet.

Laitteita hallitaan yhdeltä työasemalta, jota kutsutaan hallinta-asemaksi. Tämä asema kykenee vastaanottamaan laitteiden automaattisesti lähettämiä viestejä. Hallinta-aseman kautta pystyy lisäksi selvittämään yksityiskohtaisia tietoja kustakin verkon laitteesta sekä kontrolloimaan automaattisten sanomien lähettämistä. Lisäksi asema voi selvittää säännöllisin väliajoin jokaisen verkon laitteen tilan.

Valvottavan laitteen tehtävä on toimia normaalisti, sille tarkoitetulla tavalla. Kun laitteessa tapahtuu jokin epätavallinen toiminto, laite muodostaa automaattisesti sanoman. Tämän sanoman se lähettää verkon yli hallinta-asemalle, joka tekee tarvittavat toimenpiteet viestin mukaan, esimerkiksi ilmoittaa kriittisestä tilanteesta verkonvalvojalle. Valvottavan koneen on myös vastattava hallinta-aseman kyselyihin.

Testijärjestelmässä valvotaan IP-verkon laitteita SNMP-protokollan avulla. Tiedon kerääminen tapahtuu palvelimella, joka tarjoaa informaatiota eteenpäin valvontasovellukseen. Näiden komponenttien avulla on tarkoitus kehittää toimiva ratkaisu All-IP-konseptiin.

Insinöörityön toissijaisena tavoitteena mainitaan tiedon keskittäminen. Tällöin laitteissa, joissa ei ole tarpeeksi tehoa tai toiminnallisuutta, ei pystytä ajamaan kokonaista SNMP-ohjelmaa. Tämän vuoksi tarvitaan tietokone, joka kyselee näiden laitteiden tilaa ja raportoi niistä hallinta-asemalle.

2 VERKONVALVONTA

Tietoverkkojen nopea kasvu on saanut aikaan myös valvonnan laajuuden nopean kasvun. Tietoverkkojen laajuus ja monimutkaisuus ovat johtaneet siihen, ettei verkkoja pystytä enää valvomaan ilman tietokoneita. Tästä johtuen valvontajärjestelmiä on kehitettävä jatkuvasti.

Valvontajärjestelmän tarkoituksena on havainnoida verkossa tapahtuvia muutoksia. Nämä muutokset voivat olla käyttäjien tai verkossa tapahtuvan vian aikaansaamia. Jotta näitä muutoksia pystytään valvomaan, on myös verkonvalvontatiedon rakenteella oleellinen merkitys. Tiedon rakenne auttaa tuomaan informaation helposti esille sekä vertailemaan sitä. Verkonvalvonta voidaan jakaa esimerkiksi seuraaviin osa-alueisiin: suorituskyvyn ja käyttäjätilien valvontaan sekä virheiden havainnointiin.

Suorituskyvyn valvonta on yksi verkonvalvonnan oleellisimmista valvontakohteista. Se käsittelee mm. palvelu- sekä tehokkuuskeskeistä verkonvalvontaa. Suorituskyky ilmoittaa verkon kyvyn siirtää informaatiota. Sen mittareina toimivat esimerkiksi, kuinka monta prosenttia ajasta verkko on vapaana käyttäjälle tai kuinka kauan kestää vastauksen saanti pyyntöön. Tehokkuus kertoo ylläpitäjälle, kuinka paljon siirtokaistaa käytetään.

Virheiden havainnointi on paljon monimutkaisempaa kuin edellä kuvatut suorituskyvyn valvontaan liittyvät seikat. Virheiden havainnointia voi toteuttaa monella protokollapinon tasolla. Esimerkiksi siirtotien vika voi olla yksinkertainen fyysinen vika, jolloin havainnointi on helpompaa. Tämä ei kuitenkaan läheskään aina ole kyseessä. Ongelmien ilmetessä ylemmillä OSI-mallin kerroksilla virheen paikallistaminen tai havainnointi voi olla hankalaa. Fyysisen siirtotien vian ilmetessä voi koko tiedon siirto tiettyyn osaan verkkoa olla poikki liian pitkän ajan. Tämä laskee verkon suorituskykyä huomattavasti. Tällöin informaation siirto voidaan joutua toteuttamaan eri reittiä, jota ei välttämättä löydy.

Käyttäjätilien valvonta on yksi nykyaikaisen verkon valvontakohteista. Sillä valvotaan esimerkiksi käyttäjien laitteistoa, ohjelmia ja palveluita. Käyttäjistä pystytään keräämään informaatiota monesta eri lähteestä. Hyvin suunnitellulla informaation keräämisellä voidaan esimerkiksi ennustaa jonkin

siirtotien ylikuormittuminen. Käyttäjien valvontaan liittyy myös tietoverkon tunnistus- ja turvallisuusseikat. [1, s. 23 - 45.]

Verkonvalvonta on periaatteessa verkonhallinnan muoto. Verkonhallinnassa korostuu enemmän tiedonsiirto molempiin suuntiin eli hallinta-asemalta valvottavaan kohteeseen sekä toisinpäin. Verkonvalvonnassa tieto liikkuu pääasiassa valvottavalta kohteelta hallinta-asemalle. Verkonhallinta on tässä työssä jätetty vähemmälle huomiolle. Työssä keskitytään verkonhallintaan ainoastaan SNMP-toiminnallisuuden muodossa.

Verkonvalvonnan sijoittuminen OSI-malliin

OSI-mallin mukaan verkonvalvonta SNMP-protokollan muodossa sijoittuu mallin ylimmälle kerrokselle eli sovelluskerrokseen. Protokollapinon sovelluskerros on valittu siksi, että alemmilla tasoilla toimii paljon eri protokollia. Protokollien suuren määrän takia verkonvalvonta olisi paljon hankalampi toteuttaa.

Sovelluskerroksen valintaan on myös vaikuttanut verkonvalvonnan luonne. Yhteyden valvominen päästä päähän ei ole riittävää alemmilla kerroksilla, jolloin ei saataisi helposti tietoa ylempien kerroksien toimivuudesta. Tällöin ainoa looginen ratkaisu verkonvalvontaan on sijoittaa valvonta ylimpään kerrokseen. Sovelluskerroksen yhteys valvoo samalla myös kaikkien alempien kerrosten yhteyttä.

3 SNMP-VALVONTAJÄRJESTELMÄ

3.1 SNMP-protokollan kehittyminen

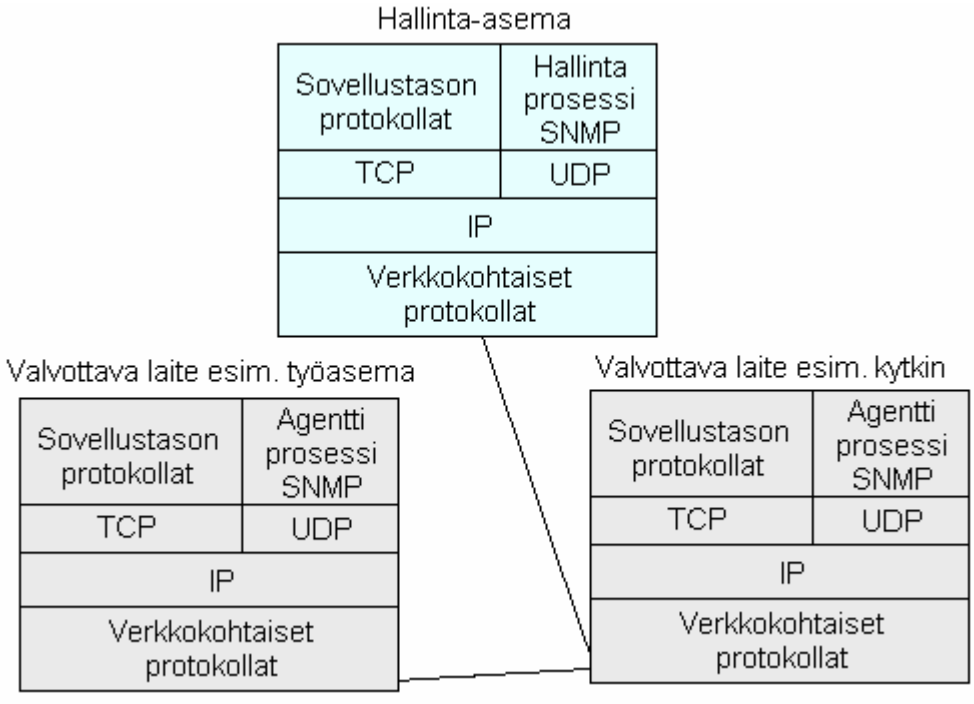
1980-luvun alkupuolella oli käytössä valvontaprotokollia, jotka oli tarkoitettu pieniin verkkoihin. Näiden protokollien avulla suurempien verkkojen valvominen olisi hankalaa tai työlästä. Internetin kasvua ei osattu ennustaa, jolloin valvontaprotokollien kehitys jäi nopeasti jälkeen verkkojen kehityksestä.

1980-luvun alkupuolella luotettiin edelleen ICMP-protokollaan. ICMP:llä toteutettu yhteyden varmistaminen päästä päähän oli aluksi täysin toimiva ratkaisu. Verkkojen kasvaessa ICMP-protokolla alkoi kuitenkin jäädä liian yksinkertaiseksi. Tämän vuoksi vuonna 1987 julkaistiin SGMP-protokolla, joka toi esiin laajempaan verkonvalvontaan soveltuvia ominaisuuksia. Tämä

protokolla jäi kuitenkin lyhytaikaiseksi, kun jo seuraavana vuonna päätettiin jatkokehittää HEMS-, CMOT- ja SNMP-protokollia. Näistä protokollista CMOT:n piti olla pitkäaikaisempi vaihtoehto ja SNMP-protokollaa alettiin kehittämään vain väliaikaiseksi ratkaisuksi. SNMP-protokollan ensimmäinen virallinen standardi julkaistiin 1990. SNMP-protokolla alkoi yleistymään vauhdilla, sen yksinkertaisuuden vuoksi. Nykyään SNMP-protokolla ja sen eri versiot ovat vallanneet verkonvalvonta-alan. [1, s. 71 - 75.]

3.2 SNMP-valvontajärjestelmän toiminta

SNMP-verkonvalvontajärjestelmään kuuluu perustapauksessa kaksi osapuolta, hallinta-asema ja valvottava laite. Valvontatilanteessa hallinta-asemia on yksi tai kaksi, ja valvottavien kohteiden määrä voi olla jopa satoja. Yleensä hallinta-asemien lukumäärä riippuu siitä, kuinka laajalla alueella asemia tarkastellaan. SNMP-valvontajärjestelmään liittyy SNMP-protokollan lisäksi myös monta tiedonrakenteeseen ja -siirtoon liittyvää standardia.



Kuva 1. Valvontajärjestelmän perusrakenne [1, s. 80 mukaillen]

Yleensä verkossa on yksi hallinta-asema, johon useampi valvottava laite lähettää tietoja. Valvottavat laitteet voivat sisältää SNMP-toiminnallisuuden, joka tarkoittaa että laitteessa on myös agentti. Agentteja sisältävät laitteet voivat olla esimerkiksi PC:itä, kytkimiä tai reitittimiä (kuva 1). Jos SNMP-toiminnallisuutta ei saada laitettua johonkin tietoverkon laitteeseen, tarvitaan jokin muu SNMP-agentti valvomaan kyseistä laitetta. [1, s. 75 - 80.]

Valvontajärjestelmän tietokannat

Valvottava tieto on jokaisella agentilla tietokannassa. Tämä tietokanta on nimeltään MIB (Management Information Base). MIB sisältää hierarkkisen järjestyksen erilaisia objekteja. Nämä objektit voivat olla esimerkiksi tekstikenttiä, lukuja tai taulukoita. Jokaiseen MIB-objektiin on olemassa oma osoite, joka on nimeltään OID (Object Identifier). Tietokannan rakenteen määrittelee SMI-standardi (Structure of Management Information).

3.3 Valvontajärjestelmän toimintatilat

Valvontajärjestelmällä on periaatteessa kaksi tapaa toimia. Ensimmäisenä näistä voisi mainita kiertokyselyn. Kiertokyselyn tarkoitus on kysellä jokaiselta agentilta vuorollaan tietoja. Kysely kuluttaa kuitenkin verkon tehokkuutta, joten laitteiden läpikäynti ei ole kovin kannattavaa. Kiertokyselyä voidaan kuitenkin käyttää, kun otetaan huomioon verkon käyttöaste. Verkon laitteiden läpikäynti voidaan suorittaa esimerkiksi yöllä. Tällöin verkon kuormitus valvontajärjestelmän toimesta ei tuota paljon haittaa normaalille käyttäjälle. Kiertokysely voidaan toteuttaa myös päivällä, jos verkon kuormitus antaa siihen mahdollisuuden.

Yleensä kiertokyselyn määräväliaika on riippuvainen verkon kuormituksesta. Aikaan vaikuttaa myös, kuinka paljon verkkoa käytetään ja kuinka paljon kiertokysely kuluttaa verkon kapasiteettia. Tätä aikaa voidaan pienentää tai suurentaa riippuen siirrettävän informaation muuttumisesta. Tällöin on otettava huomioon, kuinka paljon valvottavia kohteita on. Tiedon kriittisyyttä on myös hyvä harkita. Jos tiedolla ei ole kiirettä, se voidaan siirtää myös vähemmän ruuhkaiseen aikaan. [1, s. 79.]

Toinen vaihtoehto on toteuttaa tietojen siirto agentin automaattisten viestien välityksellä. Tällöin verkkoa kuormitetaan vähemmän kuin kiertokysely menetelmällä. Yleensä automaattiset viestit toimitetaan UDP-protokollan avulla. Tällöin huonona puolena on se, että viestien perille menosta ei ole varmaa tietoa.

Parhaimmat tulokset saavutetaan yhdistämällä nämä kaksi toimintamuotoa. Kiertokyselyn toteuttaminen vähemmän ruuhkaiseen aikaan ei haittaa verkon normaalia käyttäjää. Ruuhkaiseen aikaan siirretään vain kriittisimmät viestit automaattisten viestien muodossa. Jos hallinta-asema vastaanottaa

jonkin kriittisen tiedon automaattisen viestin muodossa, se voi lisäksi kysyä lisätietoja kyseiseltä agentilta. Tällöin verkon valvojalla on aina suhteellisen tuoretta tietoa. [1, s. 27 - 29].

SNMP-protokolla voidaan laittaa myös toimimaan jonkin yhteydellisen protokollan päälle, esimerkiksi TCP:n. Tällöin saadaan varmuutta tiedonsiirtoon, mutta ylimääräisten viestien osuus voi olla suuri verrattuna hyödylliseen informaatioon. Lopulta kysymys on siitä, halutaanko yhteys varmistaa vai ei. Tämän valinnan tekemiseen vaikuttavat valvontaan käytetty siirtokaista ja tiedonsiirron varmistaminen.

Välityspalvelin

Välityspalvelimeen joudutaan turvautumaan esimerkiksi silloin, kun kaikilla verkon laitteilla ei ole mahdollisuutta keskustella SNMP-valvontajärjestelmän kanssa. Tällaisia laitteita ovat esimerkiksi modeemit. Tällöin SNMP-agentti voi toimia välittäjäagenttina, joka varastoi valvomiensa laitteiden tietoja. Tämä kuitenkin edellyttää agentin toiminnalta uusia ominaisuuksia.

Välityspalvelin voi myös toimia SNMP-puolella salaamassa informaatiota. Tällöin palvelin ottaa vastaan salaamattomia paketteja ja salaa ne seuraavalle siirtotielle. Palvelin voi toimia myös tiedon keskittäjänä, jolloin se hakee informaatiota aliagenteilta. Tällöin agentti suodattaa informaatiosta merkityksettömät tiedot pois ja lähettää loput hallinta-asemalle. [1, s. 81 - 82].

4 SNMP-PROTOKOLLA VERSIO 1

Ensimmäinen versio SNMP-protokollasta julkaistiin vuonna 1991. Tätä versiota kutsutaan nykyään nimellä SNMPv1. Standardi sisältää protokollan perustoiminnan. Hallinta-asema ja valvottava laite voivat vaihtaa tietoja tiettyjen standardissa määriteltyjen sanomien mukaan. Ensimmäinen versio määrittelee kolme pääviestityyppiä.

- Get-viestillä haetaan halutut objektit agentin tietokannasta.
- Set-viestillä voidaan siirtää tietoa agentin tietokantaan.
- Trap-viestillä agentti ilmoittaa automaattisesti hallinta-asemalle muuttuneesta tilanteesta.

Viesteissä voidaan siirtää useamman objektin tieto samalla kertaa. Hallinta-asemalta agentille suunnassa, viestien nimet ovat GetRequest, GetNextRequest ja SetRequest. Agentilta hallinta-asemalle liikkuvat GetResponse ja trap-viesti. Agentin vastausmuoto GetResponse vastaa siis molempiin sekä get- että set-viesteihin. SetRequest-viestillä pystytään muuttamaan haluttuja objekteja agentin tietokannasta. Tämä kuitenkin edellyttää kirjoitusoikeutta. Kirjoitusoikeuden lisäksi objektin täytyy olla kirjoitettavaa muotoa. Automaattisista viesteistä käytetään tästä lähtien trap-viesti-nimeä. [1, s. 78.]

4.1 SNMP-protokollan yhteisöt

Verkonvalvontayhteisöillä kontrolloidaan verkon käyttäjien pääsyä käsiksi verkonvalvontainformaatioon. Yhteisö on mukana jokaisessa SNMP-paketissa. Sen tarkoituksena on toimittaa todentaminen (Authentication). Tiedon ja käyttöoikeuksien rajaaminen eri käyttäjille on helppo toteuttaa yhteisön avulla. Lisäksi eri käyttäjäryhmille pystytään antamaan eri käyttöoikeudet tietokantaan.

Viestien vastaanotossa agentti tarkistaa, minkä yhteisön nimi on paketissa. Jos yhteisön nimi löytyy tietokannasta ja yhteisölle löytyy luku- tai kirjoitusoikeudet, voi agentti toimia viestin mukaan. Myös viestiin vastaus tapahtuu samalla yhteisön nimellä. Trap-viestien tapauksessa voidaan määritellä hyväksyykö agentti kaikkien yhteisöjen paketit tai ainoastaan tiettyjen.

Yhteisön käyttäminen käyttöoikeusperusteena ei kuitenkaan ole kovin varma tapa. Käyttöoikeuden siirtäminen paketissa on toteutettu täysin selväkielisenä. Tämä johtaa siihen, että kirjoitusoikeuden antaminen joudutaan harkitsemaan tapauskohtaisesti. Koska kirjoitusoikeuden avulla pystyy kirjoittamaan tietokantaan esimerkiksi prosessin sammutuspyynnön, tuo kirjoitusoikeuden salliminen tietoturvauhkia. Myös lukuoikeuden antaminen täytyy harkita tapauskohtaisesti. Oikeuksien salliminen riippuu täysin siitä, missä ympäristössä toimitaan ja löytyykö tietokannasta arkaluontoista informaatiota. [1, s. 163 - 166.]

4.2 Valvontatietokannan rakenne

Valvontatiedon rakenteen määrittelee SMI-standardi (Structure of Management Information). Standardi pohjautuu ASN.1-standardiin (Abstract Syntax Notation). SMI voidaan jakaa kolmeen pääluokkaan: moduuli- ja objektimäärittelyihin sekä automaattisten viestien määrittelyihin.

SMI määrittelee minkälaisia objekteja tietokanta voi sisältää. Tietokantaan voidaan määritellä ASN.1-standardiin pohjautuvia muuttujia. ASN.1-standardi määrittelee seuraavat muuttuja tyypit: kokonaisluku (integer), merkkijono (octetstring), tyhjä (null), tunniste (object idenfier) ja sekvenssit (sequence, sequence-of). Näistä ASN.1-standardin määrittelemistä muuttujista johdettiin SMI-standardin muuttujat.

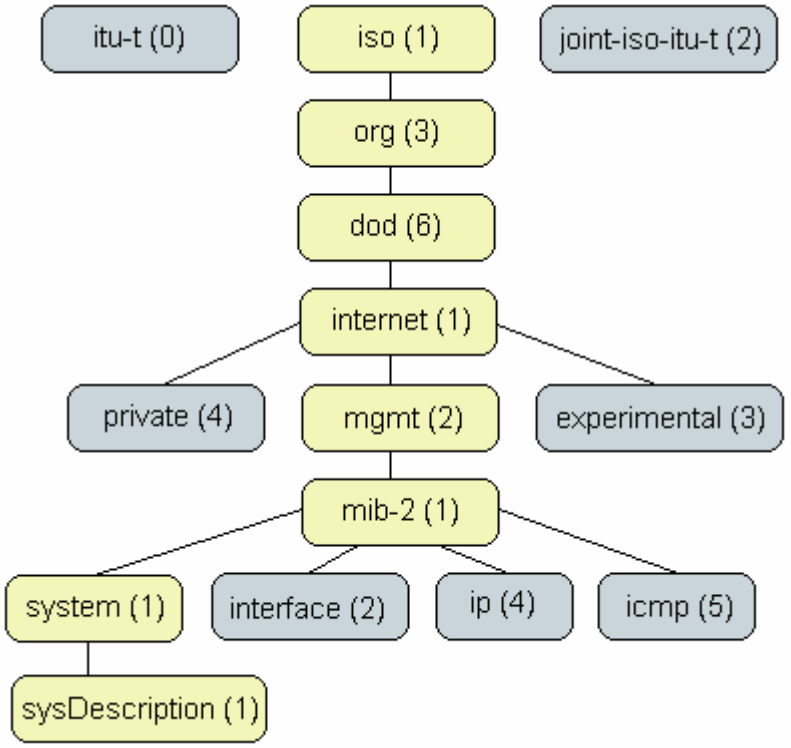
SMI:n määrittelemä tietokanta sisältää seuraavat ohjelmakohtaiset muuttujatyypit: laskuri, nollattava laskuri (Gauge), aikalaskuri (TimeTicks), yleinen (Opaque) ja IP-osoite. Lisäksi siinä voidaan käyttää ASN.1-muuttujia: tunniste ja kokonaisluku. Yleinen muuttujatyyppi voi sisältää voi sisältää mitä tahansa tietoa. Kun se otetaan käyttöön, siitä muodostetaan ASN.1-standardin merkkijonomuuttuja. Kaikki SMI:n määrittelemät luvut ovat 32-bittisiä.

SMI määrittelee myös sen, miten informaatio tulisi esittää ja nimetä. Tietokanta voi sisältää vain yksinkertaisia tietotyyppejä kuten lukuja tai kaksiulotteisia taulukoita. SMI ei tue myöskään monimutkaisten datarakenteiden luomista eikä hakemista tietokannasta. Standardi ei sisällä monimutkaisia tietorakenteita, koska se voisi tuoda hankaluuksia käyttöönottoon sekä toimintaan. [4; 1, s. 86 - 96.]

Yleensä koko tietokannasta käytetään nimeä MIB. Tietokantojen määrä riippuu alemmalla tasolla käytettävistä protokollista tai siitä mitä informaatiota agentin halutaan sisältävän. Esimerkiksi TCP/IP-verkkoihin kehitettiin standardi, joka pystyy tarjoamaan tietoja IP-verkon tilasta ja yhteysprotokollista. MIB-standardi TCP/IP-verkkoihin julkaistiin vuonna 1988. Nykyään käytetään MIB-2-versiota TCP/IP-tietokannasta. [5, 6.]

Valvontatietokannan rakenne on puumainen. Puumalli järjestää objektit loogiseksi kokonaisuudeksi. Tämän ansiosta tiedot on helpompi löytää, koska objektit on jaettu pienempiin ryhmiin. Tällöin myös suurempien tietomäärien

siirto onnistuu vaivattomammin, koska koko ryhmän voi siirtää yhdellä tiedonsiirtokäskyllä.



Kuva 2. Tietokannan rakenne

Tietokannan alimmalla tasolla on määritelty kolme järjestöä ITU-T, ISO ja joint-ISO-ITU-T. Jokainen näistä puun haaroista jakautuu omiin osioihinsa. Mielenkiintoisin haara verkonvalvonnassa on ISO, koska sen alla on myös internet sekä verkonhallintaosio (mgmt tai management). Tietokanta on niin laaja, ettei siitä pystytä toteuttamaan yhdelle agentille kuin pieni osa. Se sisältää monia eri tarkoituksiin luotuja pienempiä tietokantoja. Yleensä agenttiin sisällytetään vaan oleellisimmat osat kuten prosessien valvontaan, tiedonsiirtoon ja reititykseen liittyvät tietokannat. (Kuva 2.)

.1	.3	.6	.1	.2	.1
iso (1)	org (3)	dod (6)	internet (1)	mgmt (2)	mib-2 (1)

Kuva 3. OID-tunniste

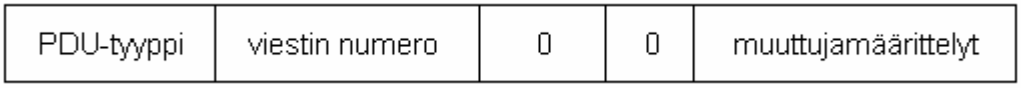
Jokainen objekti tietokannassa on merkitty OID-tunnistusnumerolla. OID-numeron määrittelee ASN.1-standardi. OID-numero on yksinkertainen sarja pisteillä erotettuja numeroita (kuva 3).

MIB-tietokannasta pystytään hakemaan yksittäinen tieto viittaamalla esimerkiksi sen nimeen tai OID-tunnukseen. Käyttöjärjestelmän nimi (sysDescription) voidaan hakea OID-numerolla 1.3.6.1.2.1.1.1 tai sysDescpriction-nimellä. [1, s. 97 - 115.]

Tietokantaa voidaan myös laajentaa omilla objekteilla. Objektin lisääminen tietokantaan vaatii itse ohjelmakoodin sekä uuden tietokantamäärittelyn. Tietokantamäärittely voidaan sijoittaa esimerkiksi enterprises-ryhmään, jonka OID-numero on 1.3.6.1.4.1. Jos laitteeseen jätetään pysyvästi kyseinen testiobjekti, on tulevaisuudessa huomioitava uusien laitteiden sisältämät tietokannat. [3.]

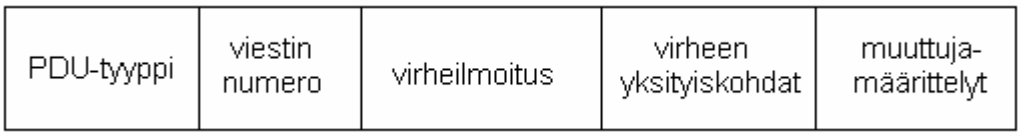
4.3 Tiedonsiirtoviestien rakenne

Tiedonsiirtoviestien rakenne riippuu viestin tyypistä, mutta viesteissä on myös yhteinen osa. Kaikissa viesteissä on ensimmäisenä SNMP-protokollan versionumero ja yhteisönimi. SNMPv1-viestin tapauksessa versionumero on 0. Näiden kenttien jälkeen tuleva viestin loppuosa on nimeltään PDU-viesti. Viestien rakenne jakautuu kolmeen tyyppiin, jotka ovat yleisviesti (request), vastausviesti (response) ja trap-viesti. Jokainen viestityyppi käydään seuraavaksi läpi yksityiskohtaisesti.



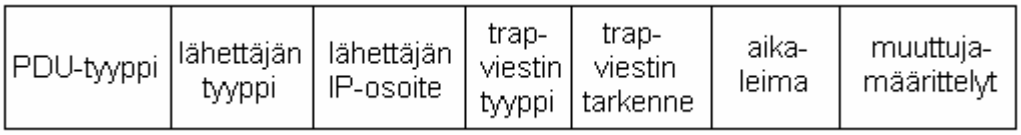
Kuva 4. Yleisviestin rakenne [1, s.174 mukaillen]

Yleisviestin rakenne on viestityypeistä kaikkein yksinkertaisin. Siinä määritellään vain PDU-tyyppi, viestin numero ja muuttujamäärittelyt (kuva 4). PDU-tyyppi kertoo, mikä viesti on kyseessä. Mahdollisia vaihtoehtoja ovat GetRequest, GetNextRequest ja SetRequest. Viestin numero määrittelee yksittäisen numeron joka viestille, jotta viestiä ei sekoiteta muihin viesteihin. Muuttujamäärittelyt-kenttä sisältää tiedot haettavista muuttujista. Jokainen muuttujamäärittely sisältää muuttujan tyypin sekä muuttujan arvon. Kun kyseessä on tiedonhakuviesti on arvo merkittävä null-tyyppiseksi.



Kuva 5. Vastausviestin rakenne [1, s.174 mukaillen]

Vastausviesti on tyypiltään GetResponse. Siinä määritellään samat kentät kuin yleisviestin tapauksessakin, mutta tällä kertaa kaikille kentille tulee jokin merkitys. Uusina kenttinä tulevat virheilmoitus ja virheen yksityiskohdat. Virheilmoitus määrittelee virheen yleistyypin virhetilanteessa. Esimerkiksi, jos muuttujaa ei löydy tietokannasta, on virheilmoituksen arvo 2. Tässä tapauksessa virheen yksityiskohdat -kenttä kertoo, mikä muuttujista puuttui tietokannasta. (Kuva 5.)



Kuva 6. Trap-viestin rakenne [1, s. 174 mukaillen]

Trap-viestissä on määritelty enemmän kenttiä kuin aikaisemmissa yleis- ja vastausviestissä (kuva 6). Viestin alussa on informaatiota lähettäjästä, josta kertovat lähettäjän tyyppi ja IP-osoite -kentät. Lähettäjän tyyppi kertoo, kuinka monella OSI-mallin tasolla lähettäjälaite toimii. Tämän jälkeen viestissä on yksityiskohtaista tietoa trap-viestistä. Trap-viestin tyyppi (generic-trap) määrittelee viestin yleistyypin (0-6). Yleistyyppejä ovat esimerkiksi coldStart (0) ja linkUp (3). Yleistyyppi 6 määrittelee käyttöön loput trap-viesteistä, näiden viestien tarkempi informaatio löytyy trap-viestin tarkenne -kentästä (specific-trap). Aikaleima on lähettävän agentin sysUpTime-kentän arvo lähetys hetkellä. [1, s. 173 - 175.]

5 SNMP-PROTOKOLLA VERSIO 2

SNMPv1:n käyttöönoton jälkeen alkoi arvostelu sen suurista tietoturvapuutteista. Verkkolaitteiden valmistajat eivät yleensä uhkien vuoksi tukeneet protokollan kaikkia ominaisuuksia. Yleensä laitteet laitettiin tukemaan vain lukuominaisuutta, kirjoitusominaisuuden ollessa tietoturvaltaan liian heikko. Tämän vuoksi protokollan toisen version tavoitteena oli mm. kehittää sen turvallisuusominaisuuksia.

Kehitystyö aloitettiin vuonna 1992 julkaisemalla 8 dokumenttia, jotka eivät olleet standardeja. Päätavoitteena oli kehittää protokolla hallitsemaan suurempaa määrää verkon resursseja. Lisäksi tavoitteena oli pitää protokolla yksinkertaisena, helppona ottaa käyttöön sekä parantaa tietoturvaa. Standardi saatiin nopeasti valmiiksi ja se julkaistiin jo seuraavana vuonna.

Useamman vuoden jälkeen, vuonna 1996, protokolla päätettiin julkaista uudestaan ilman tietoturvaominaisuuksia. Tähän ratkaisuun päädyttiin sen vuoksi, että valmistajat eivät pitäneet protokollan turvallisuusominaisuuksien muodosta. Tämän vuoksi IETF päätti protokollan uudelleen julkaisemisesta. Tämä versio tunnetaan myös nimellä SNMPv2 ja SNMPv2C. [1, s. 331-357.]

5.1 SNMP-versio kahden ominaisuudet

SNMPv2 tukee täysin samoja ominaisuuksia kuin SNMPv1. Lisäksi siihen saatiin suurten tietomäärien siirtoon uusi käsky sekä uutena ominaisuutena hallinta-asemien välille omat tiedonsiirtokäskyt. Valvontainformaation rakenne uusittiin myös käyttämään versiota SNMPv2 SMI.

SNMPv2-protokollan tiedonsiirtoviestit

Aikaisempaan versioon verrattuna toisen version käskyt ovat muuttuneet selkeämmiksi. Versiossa on määritelty yhteinen vastausmuoto kaikille käskyille agentilta hallinta-asemalle. Seuraava taulukko esittelee kaikki tiedonsiirtoviestit sekä niiden toiminnan.

Taulukko 1. SNMPv2-viestien tyypit ja niiden toiminnot

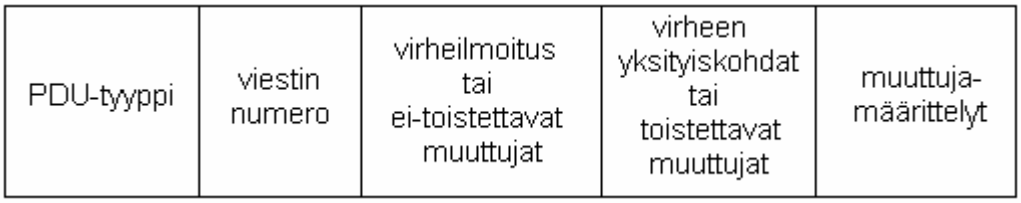
Viestin tyyppi	Suunta	Informaatio
Response	Agentilta hallinta-asemalle tai hallinta-asemalta toiselle	Vastaus pyydettyihin informaatioihin
GetBulkRequest	Hallinta-asemalta agentille	Pyytää useamman arvon kerralla
GetRequest	Hallinta-asemalta agentille	Pyytää informaatiota jokaiselle listatulle objektille
GetNextRequest	Hallinta-asemalta agentille	Pyytää jokaisen listatun objektin seuraavan arvon
SetRequest	Hallinta-asemalta agentille	Muuttaa listattujen objektien arvoja
InformRequest	Hallinta-asemalta toiselle	Pyytää informaatiota toiselta hallinta-asemalta
SNMPv2-Trap	Agentilta hallinta-asemalle	Tapahtuman takia lähetetty automaattinen viesti

Peruskäskyllä GetRequest pystytään hakemaan ennalta tiedettyjen objektien tiedot. Kun kyseessä taulukko, jonka pituudesta ei ole tietoa, voidaan käyttää GetNextRequest- tai GetBulkRequest-pyyntöjä.

Käskyjen käytössä kannattaa miettiä mitä pyyntöä käyttää esimerkiksi ohjelmassa. Kun tiettyyn käskyyn saadaan sisällytettyä mahdollisimman monta objektia, voidaan kaikki tieto siirtää yhdellä pyyntö-vastaus -parilla. Jos ohjelmaa ei saada toteutettua näin, voi tiedon hakeminen luoda turhan paljon verkkoliikennettä. Tämä tulee esille varsinkin kiertokyselyviesteissä, joita lähetetään monelle agentille. [1, s. 365 - 385].

Tiedonsiirtoviestien rakenne

SNMP-protokollan toisessa versiossa on määritelty yhteinen viestirakenne kaikille viesteille. Tämä helpottaa viestien käsittelyä, koska viesteistä löytyvät tiedot aina samasta kohtaa. Viestin rakenteessa on ensimmäisenä protokollan versio numero sekä yhteisökenttä. SNMPv2-viestin tapauksessa version arvo on 1. Näiden jälkeen tulevat PDU-viesteistä riippuvaiset kentät.



Kuva 7. SNMPv2-viestin rakenne

Vastausviesti eli Response-PDU käyttää viestissä olevia virhekenttiä (kuva 7). Nämä virhekentät toimivat täysin samalla tavalla kuin aikaisemmassa versiossa, jotka käytiin läpi luvussa 4.3. Virhekentät ovat molemmat nollia, jos viesti on onnistunut. Tällöin kaikki haetut objektit löytyivät tietokannasta tai informaation kirjoittaminen tietokantaan onnistui.

Suuremman tietomäärän hakemiseen tarkoitettu GetBulkRequest-PDU-viestityyppi käyttää ei-toistettavat ja toistettavat muuttujat -kenttiä (kuva 7). Informaation hakeminen toimii esimerkiksi seuraavalla tavalla, kun muuttujia on 5, ei-toistettavat muuttujat -kentän arvo on 3 ja toistettavat muuttujat -kentän arvo on 4. Tällöin ensimmäiset kolme muuttujaa haetaan yksittäisinä muuttujina. Lopuista kahdesta muuttujasta haetaan toistettavat muuttujat -kentän määrittelemä muuttujien lukumäärä, joka tässä tapauksessa on 4. Yhteensä haettavia muuttujia tulee tässä tapauksessa 11. Tällä hakuviestillä pystytään hakemaan helposti suuri määrä tietoa yhdellä pyyntö-vastaus -parilla.

Kaikki loput viestit, jotka on esitelty taulukossa 1, käyttävät myös samaa viestimuotoa. Tällöin viestissä on määritelty ainoastaan PDU-tyyppi, viestin numero sekä muuttujamäärittelyt (kuva 7). Muihin viestissä oleviin kenttiin on laitettu arvo 0. [1, s. 366 - 385.]

5.2 Valvontatietokannan rakenne

Uuden SNMP-protokollan myötä myös tietokannan rakenne päivitettiin. Vuonna 1993 julkaistiin uusi rakennemäärittely SNMPv2-protokollalle. Tietokannan rakennemäärittelyn lisäksi protokolla sisälsi määrittelyt uusille muuttujatyypeille sekä uusille MIB-tietokannoille.

Tietokannan rakenne on periaatteessa täysin samanlainen kuin ensimmäisessäkin tietokannassa. Tietokannan muuttuja tyyppeihin lisättiin 64-bittiset laskurit ja etumerkittömät 32-bittiset luvut. Lisäksi rakennemäärittelyissä

tuli uusia makromäärittelyjä. Automaattisten viestien makromäärittelyihin lisättiin esimerkiksi ilmoituksen tyyppi (notification-type). Muihin makromäärittelyihin lisättiin esimerkiksi moduuli ja objektin tunnistus makrot (module-identity, object-identity). [7.]

6 SNMP-PROTOKOLLA VERSIO 3

SNMP-prokollan versio 3 julkaistiin 1998. Sitä kutsutaan myös turvallisuuspäivitysten versioksi, koska SNMPv2:n kehityksessä hylätyt turvallisuusjulkaisut olivat pohjana kolmannelle versiolle. Päivitykset on sisällytetty SNMP-protokollaan, vaikka itse protokollan toiminnallisuus ei muuttunut mitenkään.

SNMPv3-protokollassa on otettu huomioon aiemmissa versioissa esille tulleet tietoturvapuutteet. Version kolme dokumentit eivät määrittele koko SNMP-toiminnallisuutta. Koko toiminnallisuuden aikaan saamiseksi on käytettävä pohjana SNMPv1-tai SNMPv2-protokollan dokumentteja. SNMPv3 määrittelee PDU-viestien hallinnan. Lisäksi se ottaa kantaa valvontajärjestelmän osien rakenteeseen. Esimerkiksi hallinta-aseman ja agentin rakenneosat on määritelty yksityiskohtaisesti. Näillä määrittelyillä saavutetaan SNMP-järjestelmän integrointi muihin järjestelmiin. [1, s. 447 - 473.]

6.1 Tietoturva

Standardin mukaan SNMPv3-protokollaan on sisällytetty viestien autentikointi eli todentaminen ja salaaminen. Jos molemmat ovat käytössä, protokolla pystyy torjumaan monet tietoturvauhkat. Todentaminen voidaan suorittaa käyttäen HMAC-todennusta (Hash Message Authentication Code). Salaaminen on puolestaan standardin mukaan toteutettavissa DES-salauksella (Data Encryption Standard). [1, s. 429]

Nämä protokollassa mainitut menetelmät alkavat kuitenkin olla nykypäivänä hieman vanhentuneita. Tämän vuoksi eri valmistajat ovat ottaneet käyttöön omiin tuotteisiinsa varmempia salaus- ja todennusmenetelmiä. Esimerkiksi verkkolaitevalmistaja Cisco Systems Inc. tukee uusimmissa IOS-järjestelmissä mm. CFB-AES (Cipher Feedback - Advanced Encryption Standard) salauksia 128, 192 ja 256 bitin vahvuuksilla. Lisäksi vaihtoehtoisena menetelmänä on 3DES-salaus. [9]

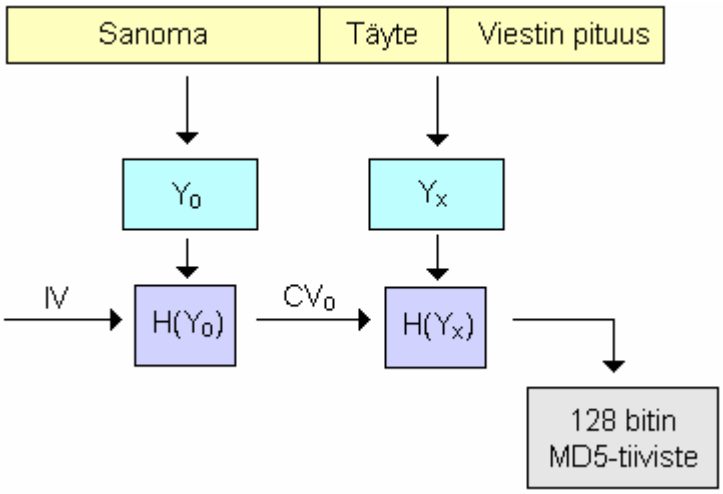
6.2 Tiedon todentaminen

Todentamisen periaatteena on varmistaa identiteetti. SNMP:n tapauksessa on tarpeellista varmistaa tiedon aitous ja se, että viesti on tullut perille oikeasta osoitteesta. Todennus suoritetaan käyttäen paketissa olevaa todennuskenttää. Todennuskenttä sisältää lähettäjän päässä lasketun arvon viestin sisällöstä. Vastaanottajan päässä viestin perusteella lasketaan uusi todennuskentän arvo. Todennus on onnistunut, jos viestistä laskettu todennusarvo vastaa viestissä olevaa.

Todennuskentän arvon laskemiseen käytetään SNMPv3-standardin mukaan HMAC-todennusta. HMAC:n aikaansaamiseksi on tarjolla kaksi algoritmia MD5 ja SHA-1. Näitä algoritmeja kutsutaan myös yksisuuntaisiksi tiivisteiksi.

6.2.1 MD5-algoritmi

MD5 (Message Digest 5) ottaa vastaan määräämättömän pituisen datan ja tuottaa 128-merkkisen bittijonon. Menetelmää kutsutaan yhden suunnan hajakoodaus-algoritmiksi. Yhdellä suunnalla tarkoitetaan sitä, ettei lopputuloksesta pysty laskemaan, mikä alkuperäinen viesti oli. Algoritmin lopputulosta kutsutaan hajautusarvoksi tai tiivisteeksi.



Kuva 8. MD5-algoritmi

Algoritmin perustoiminta on seuraavanlainen. Koska sisään otettu informaatio otetaan vastaan erimittaisissa lohkoissa, jokaiseen lohkoon lisätään täytebittejä. Täyteen tavoitteena on muodostaa 448 bittiä pitkä lohko. Täytebittejä lisätään 1 - 512, vaikka lohko olisi jo valmiiksi oikean mittainen. Jos syötetty informaatio on jo 448 bittiä pituudeltaan, siihen lisätään silti 512 bitin

täyte. Tällöin loput 512 bitin viestistä ilmoittaa informaation pituuden 64:llä bitillä, jolloin viestin pituudeksi tulee 2 * 512 bittiä. Täyte koostuu yhdestä ykkös-bitistä sekä tarvittavasta määrästä nolla-bittejä. Loput lohkosta sisältää tiedon, kuinka monta täytebittiä lisättiin sekä alkuperäisen informaation bittien lukumäärän. Koko lohkon pituudeksi muodostuu n kertaa 512 bittiä (Y_0), jossa n on positiivinen kokonaisluku.

Täytteen ja lohkonpituuden muuntelun jälkeen on vuorossa MD5-algoritmi. Ennen algoritmia joudutaan kuitenkin alustamaan koodin tuottaja. Tämän jälkeen informaatio syötetään 32 bitin lohkoissa algoritmiin. Algoritmi sisältää neljä erilaista loogista funktiota, jotka suoritetaan 16 kertaa. Yhteen näistä funktioista syötetään ¼ * 512 bitin lohko sekä edellisestä MD-algoritmista syntynyt 128 bitin tulos CV_0. Jos kyseessä on ensimmäinen MD-muunnos, tarvitaan edellisen tuloksen sijasta aloituskoodivektori IV. Tämä vektori voi olla ennalta sovittu tai satunnainen, jonka viestin vastaanottaja ja lähettäjä tietävät. Lisäksi prosessissa on mukana sini-funktiosta johdettu 32-bittinen luku. Tämä luku tuo prosessiin lisää satunnaisuutta, jolloin lähdeinformaation yksinkertaisuus ei pysty heijastumaan lopputulokseen. (Kuva 8). [1, s. 433 - 438.]

6.2.2 SHA-1-algoritmi

SHA-1-algoritmi on hyvin samankaltainen kuin aiemmin kuvattu MD5. SHA perustuu MD5-menetelmän aikaisempaan versioon. Tämän vuoksi sen perusrakenne muistuttaa hyvin paljon MD5-menetelmää. SHA-1-algoritmi pystyy prosessoimaan kerralla 2^{64} bittiä. Prosessointi tapahtuu 512-bitin lohkoissa kuten aiemmin kuvatussa MD5-menetelmässä. Lopputuloksena saadaan 160-bittinen hajautusarvo. Prosessointi aloitetaan täyttämällä lohkot 512-bitin mittaisiksi. Lohkojen täyttäminen tapahtuu samalla lailla kuin luvussa 6.2.1. Tämän jälkeen lohko jaetaan 64-bitin ryhmiin, jotka syötetään funktioon. Funktiossa on viisi 32-bittistä alustettua rekisteriä, jotka sisältävät aina tietyt arvot. Funktio ottaa lisäksi vastaan 64-bitin luvun sekä aikaisemmasta SHA-vaiheesta tulleen 160-bittisen luvun. Jokaisessa vaiheessa on myös mukana ennalta määrätty vakio. Näiden avulla lasketaan ensimmäinen välitulos.

Ensimmäisen välituloksen aikaansaamiseksi tarvitaan 20 kierrosta, jotta funktio pystyy käymään läpi kaikki syötetyt bitit. Ensimmäisen välituloksen jälkeen on vielä kolme eri funktiota, joissa käytetään eri vakiota. Lopullisen

SHA-1-koodin muodostamiseen tarvitaan siis neljä funktiota, joissa kussakin on 20 laskuvaihetta. Neljännen laskuvaiheen jälkeen on lisäksi summaus tuloksesta sekä edellisestä SHA-tuloksesta. Näiden summasta syntyy yhden lohkon lopullinen tulos. Kun kaikki 512-bitin lohkot on käyty läpi, saadaan lopputuloksena 160-bittinen SHA-1-hajautusarvo. [1, s. 438 - 439.]

6.2.3 HMAC-todentaminen

Todentamisessa voidaan käyttää periaatteessa mitä tahansa salaus- tai algoritmimenetelmää. Symmetristen salausalgoritmimenetelmien käytössä on kuitenkin yksi huono puoli: ne on suunniteltu toimimaan molempiin suuntiin eli ne pystytään salaamaan ja purkamaan samalla avaimella. Tällöin salattuun informaatioon voidaan löytää tietty avain suhteellisen helposti.

Yksisuuntaiset tiivisteet ovat tarkoitettu vain tiedon tiivistämiseen, joten niiden käyttö on suositeltavampaa. Lisäksi salausmenetelmät ovat hitaampia suorittaa kuin yksisuuntaiset tiivisteet.

Yksisuuntaisia tiivisteitä ei voida käyttää suoraan todentamiseen, koska niihin ei sisälly salasana-ominaisuutta. Tämän vuoksi on kehitelty HMAC-menetelmä. HMAC muodostetaan kahdessa eri vaiheessa. Ensimmäisessä vaiheessa syötetään algoritmiin n (positiivinen kokonaisluku) kertaa 512-bittiä sekä salasanan sisältävä lohko. Salasanaan on tuotu lisää satunnaisuutta tekemällä XOR-funktio vakion kanssa. Menetelmässä käytetyille vakioille löytyy arvot standardista.

Toiseen algoritmiin syötetään ensimmäisestä vaiheesta saatu algoritmin tulos sekä salasana, joka on muunnettu uudella vakiolla. Toisen vaiheen tuloksena saadaan koko HMAC-menetelmän tulos. [1, s. 442 - 444].

6.2.4 HMAC-menetelmän turvallisuus

HMAC-menetelmän turvallisuuden pohjana on siinä käytetty algoritmi. Jos menetelmässä käytetään esimerkiksi edellä esitettyä MD5-algoritmia, on HMAC:n turvallisuus verrannollinen MD5:n 128-bittiin. Algoritmin murtamiseen voidaan tällöin käyttää menetelmää, jossa etsitään kahta lukua jotka tuottavat saman MD5-koodin. 128-bitin tapauksessa läpikäytävien vaihtoehtojen lukumäärä on 2^{64}.

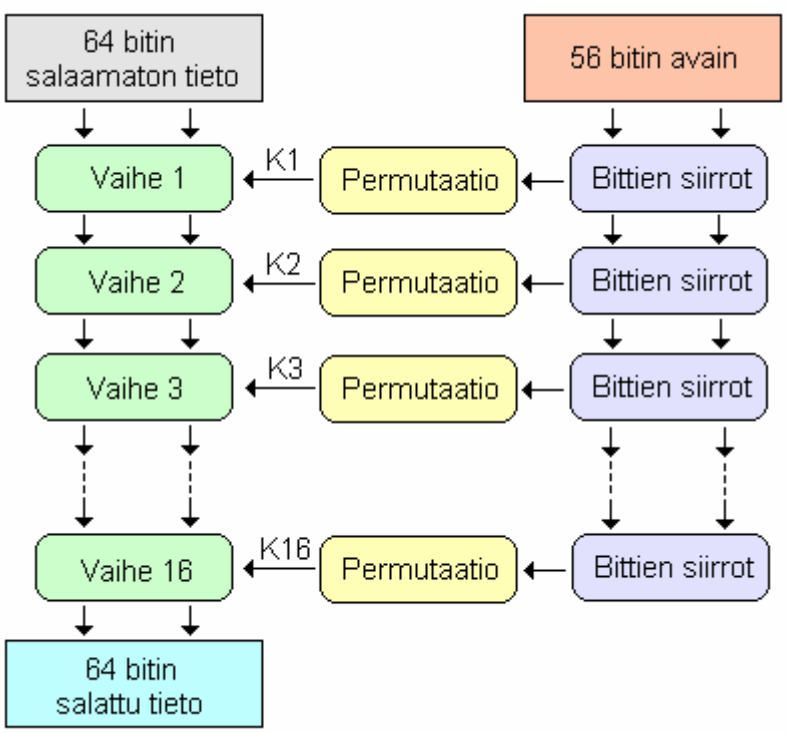
Koodinmurtamisen tapauksessa tarvitaan lisäksi salausavain. Koska kolmas osapuoli ei tiedä avainta, se joutuu tarkkailemaan läpi kaikki 2^{64} vaihtoehtoa. Esimerkiksi normaalissa 1 Gbps yhteydessä, tämä tarkoittaisi useiden tuhansien vuosien tarkkailujaksoa. Tämä edellyttää tietenkin, ettei salausavainta muuteta tarkkailujakson välillä. Näiden esimerkkien perusteella HMAC-menetelmä on täysin turvallinen. [1, s. 445].

6.3 Tiedon salaaminen

Sähköisen tiedon salaaminen on kehittynyt paljon viimeisen parin kymmenen vuoden aikana internetin yleistymisen johdosta. Tiedon salaamisen perusteena on, ettei kolmas osapuoli pääse käsiksi tietoon. Tiedonsalaus jaetaan kahteen pääluokkaan julkisen ja salaisen avaimen menetelmiin. [1, s. 427 - 428.]

6.3.1 DES-salausmenetelmät

Informaationsalausstandardi DES (Data Encryption Standard) on ollut yksi käytetyimmistä salausmenetelmistä. Salaus perustuu yhden avaimen menetelmään, jota kutsutaan myös symmetriseksi salaukseksi. Diffie ja Hellman kehittivät protokollan 70-luvun lopulla, jolloin se otettiin Yhdysvaltojen hallituksen viralliseksi salausstandardiksi. DES säilyi pitkään erittäin vahvana salausmenetelmänä. Sitä käytetään edelleen joihinkin sovelluksiin, mutta avaimen koon vuoksi sitä ei ole enää pidetty tarpeeksi turvallisena.



Kuva 9. DES-salausmenetelmä

Salaus toteutetaan käyttämällä yhtä 56 bitin avainta. Tämä avain jakautuu 16 aliavaimeen K, joista kukin on pituudeltaan 48 bittiä. Aliavaimen muodostuksessa pääavain jaetaan ensin kahteen 28 bitin haaraan. Tämän jälkeen bittejä siirretään muutamalla pykälällä vasemmalle, jonka jälkeen tulokset yhdistetään permutaatiolla. Seuraavalle tasolle siirretään siirroista saatu tulos. Permutaatiosta tullut tulos viedään salausvirtaan.

Toinen osapuoli salauksessa on syötetty selväkielinen osa, joka on pituudeltaan 64 bittiä. Bitit syötetään aluksi permutaatioon. Sen jälkeen bitit jaetaan kahteen 32-bitin haaraan, jotka yhdistetään kussakin vaiheessa aliavainten permutaatiosta tulleisiin bitteihin. Kukin vaihe sisältää monimutkaisen yhdistelmän permutaatioita sekä XOR-funktioita. Viimeisen vaiheen jälkeen tehdään vielä käänteispermutaatio. Näiden syötteiden avulla muodostetaan 16-vaiheisessa operaatiossa salattu sanoma (kuva 9). [1, s. 429 - 432.]

3-DES-salausmenetelmä

3-DES tai Triple DES on nykyaikaisempi versio DES-salauksesta. Tässä DES-menetelmässä käytetään kolmea erillistä DES-salausvaihetta tiedon salaukseen. Jokaisessa DES-vaiheessa käytetään kaikkia 16 vaihetta. Tieto

salataan kolme kertaa peräkkäin avaimilla A, B ja C. Tällöin salauksesta tulee kolme kertaa vahvempi kuin DES-salauksessa. Käytössä on myös monia eri variaatioita. Vaihtoehtona voi olla esimerkiksi salaaminen avaimella A, purkaminen avaimella B ja salaaminen avaimella A. [9.]

6.3.2 Tiedon salauksen turvallisuus

Tiedon salauksen turvallisuus on noussut esille tietokoneiden huiman kehityksen vuoksi. Esimerkiksi DES-menetelmästä on julkaistu monia artikkeleita liittyen sen murtamiseen. Nykyaikaisilla salauksen murtamiseen tarkoitetuilla laitteilla kuluu murtamiseen aikaa vain muutamia tunteja.

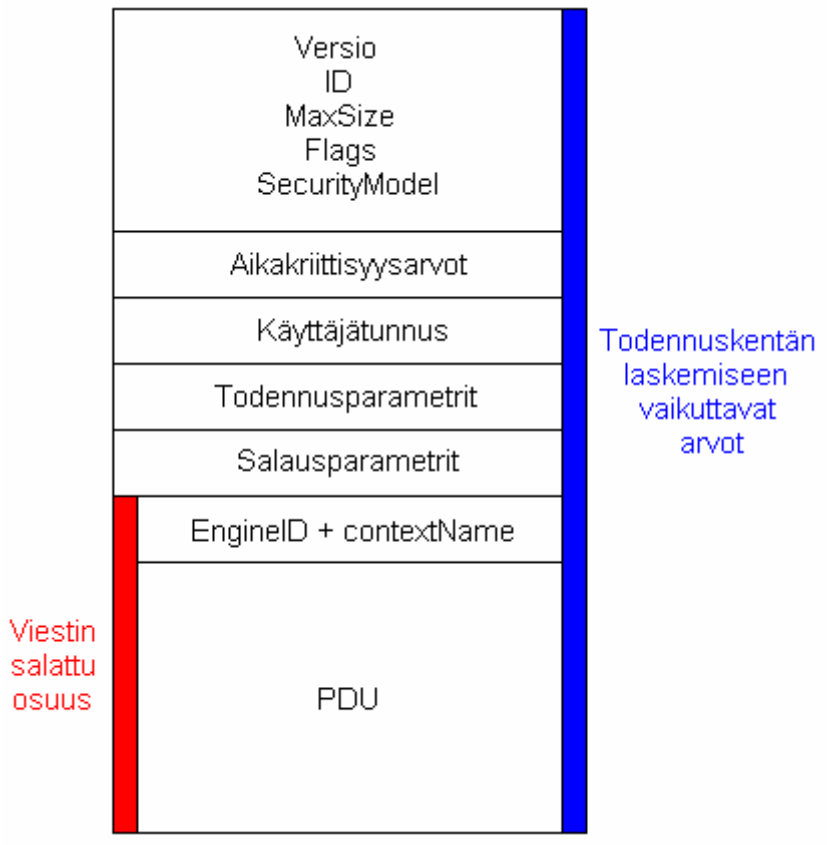
Nykypäivänä on käytössä monia muita tiedon salaukseen liittyviä standardeja. Näistä yksi käytetyimmistä on AES-standardi. Se määrittelee tällä hetkellä Rijndael-algoritmin käytettäväksi salaukseen. Salaukseen on määritelty mm. 256 ja 512 bitin avaimet. Tätä salausmenetelmää sovelletaan myös joissakin SNMPv3-tuotteissa. [10].

6.4 Käyttäjäkohtainen turvallisuusmalli

Käyttäjäkohtainen turvallisuusmalli eli USM (User-Based Security Model) määrittelee miten salaus- ja todennusmenetelmiä hyödynnetään. Lisäksi USM keskittyy muihin tietoturvaseikkoihin, SNMP-paketin muotoon sekä siihen liittyviin kenttiin. Aiemmin kuvatut todentamis- ja salausmenetelmät pätevät myös USM-mallissa. USM määrittelee lisäksi eri menetelmiä julkisille ja salaisille avaimille sekä niiden jakamisen ja päivittämisen verkossa. USM:n toiminnallisuuden määrittelee agentin USM-tietokanta.

Aikakriittisyys

Aikakriittisyydellä varaudutaan lähetettävien pakettien manipulaatioon. Tällöin verkon agentit ovat synkronoituneet hallinta-asemaan. Koska paketti sisältää lähettävän verkkolaitteen ajan, ei viestiä pystytä viivästyttämään tai toistamaan.



Kuva 10. SNMPv3-paketin rakenne

Agentteja valvotaan kolmen kentän avulla. Nämä kentät ovat snmpEngineBoots, snmpEngineTime ja latestReceivedEngineTime. Ensimmäinen näistä kertoo agentin uudelleenkäynnistysten määrän, toinen ylläpitää agentin kelloa hallinta-aseman päässä ja kolmas kertoo viimeisen vastaanotetun kellonajan agentin päässä. Kellonaikojen ja uudelleen käynnistyksien perusteella hallinta-asema pystyy ajoittamaan päivitykset. (Kuva 10.)

Viestin salaaminen

USM-malli käyttää viestin salaamiseen DES-salausta ja CBC-ketjutusta (Cipher Block Chaining). Salaukseen syötetään 128-bittinen privKey. Koska luku on liian pitkä DES-salauksen avaimeksi, siitä käytetään vain ensimmäiset 8 oktettia. Jokaisesta oktetista otetaan vain 7 eniten merkitsevää bittiä avaimeen. Loput 8 oktettia hyödynnetään IV:n ensimmäiseen versioon. Lisäksi salauksessa on mukana IV:n muuttamiseen tarkoitettu salt-luku. Luku muodostetaan snmpEngineBoots-kentän avulla. Lopullinen IV saadaan XOR-funktiolla ensimmäisen IV:n ja salt-luvun avulla.

Salt-luku lähetetään vastaanottajalle viestin mukana, jotta vastaanottaja pystyy laskemaan oikean IV:n. Koska salt-luku muuttuu jokaisen käynnistyskerran myötä, selväkieliselle tekstille saadaan enemmän vaihtelua. Kolmannen osapuolen kannalta ei ole hyötyä selväkielisenä lähetetystä salt-luvusta, koska se ei ole suoraan IV. [1, s. 498 - 524].

6.5 Näkymäkohtainen turvallisuusmalli

Näkymäkohtainen turvallisuusmalli eli VACM (View-Based Access Control Model) on toinen SNMPv3-protokollan turvallisuusmalleista. VACM määrittelee, mitä oikeuksia kullakin käyttäjällä on tietokantojen katseluun sekä todentamis- ja salausmenetelmät.

VACM käyttää oikeuksien toteuttamiseen ryhmiä. Kullakin ryhmällä on tietyt oikeudet esimerkiksi tietokantaan. Nämä oikeudet määritellään SNMP-paketin securityModel- ja securityName-kentillä. Paketin salaaminen ja todentaminen toimii samoilla periaatteilla kuin aiemmin kuvatussa USM-mallissa.

Ryhmien rajoitetut tietokantaoikeudet

Yleensä halutaan rajoittaa käyttäjien pääsyä tiettyihin MIB-osiin, jolloin VACM-mallista löytyy ratkaisu tähän. VACM-oikeudet pystytään luomaan VACM-taulukoiden avulla. Tämä tietenkin edellyttää sitä, että agentti tukee kyseistä tietokantaa.

VACM-tietokanta jakaantuu neljään eri taulukkoon. Ensimmäinen taulukko määrittelee eri contextName-muuttujat. Context-kentät määrittelevät muiden agenttien tietokannat, jotka sisältyvät kyseiseen agenttiin. Toinen VACM-taulukko määrittelee ryhmien luokitukset. Kolmas taulukko sisältää kunkin ryhmän oikeudet tietokantaan ja viimeinen niiden näkymät tietokantaan. [1, s. 525 - 539].

7 STANDARDIEN MUKAISET TIETOKANNAT

7.1 Tietokanta TCP/IP-verkkojen valvontaan

Standardin tarkoituksena oli luoda sopiva kokoelma objekteja TCP/IP-verkkojen valvontaan. SNMP:n ohjausta noudattaen myös tietokanta pidettiin yksinkertaisena. Tähän vaikutti myös protokollan heikko turvallisuus, joka

esti hallitsevien objektien sisällyttämisen tietokantaan. Tämän vuoksi MIB sisältää vain yksinkertaisemmat objektit, joilla ei ole paljon vaikutusta verkon hallitsemiseen.

Tietokantaan sisällytettiin ainoastaan objekteja, joita aidosti tarvittiin verkonvalvonnassa. Turhien objektien luomista vältettiin tutkimalla, voidaanko ne johtaa joistain muista verkonvalvonnan objekteista. Tietokanta yritettiin myös tehdä ohjelmallisesti yksinkertaiseksi, ettei se rasita agenttia liikaa. Esimerkiksi laskureita ei sisällytetty kuin yksi kerroksen kriittistä sektoria kohden.

Valvontatietokannan puumallissa TCP/IP-MIB löytyy mgmt (2) alta, joka OID-merkintänä on 1.3.6.1.2.1. Siinä on määritelty 11 ryhmää. Ryhmät ovat seuraavat:

- system (1)
- interfaces (2)
- at (Address Translation) (3)
- ip (4)
- icmp (5)
- tcp (6)
- udp (7)
- egp (8)
- cmot (9)
- transmission (10)
- snmp (11).

System-ryhmä kuuluu ns. pakollisiin ryhmiin. Se on pakko sisällyttää tietokantaan, jos MIB-2-tietokanta on käytössä. Yleensä ryhmiä sisällytetään agentin tietokantaan sen mukaan, missä ympäristössä agentti sijaitsee. Laitetyyppi, jossa agentti sijaitsee vaikuttaa myös ryhmien valintaan. Esimerkiksi normaali työasema tarvitsee vain verkon normaaliin toimintaan sisältyvät ryhmät. Yleisperiaate ryhmien lisäyksessä on, että jos tarvitaan yksi objekti kyseisestä ryhmästä, on lisättävä koko ryhmä.

Nykyään käytössä on toinen versio samasta TCP/IP-verkkoihin suunnitellusta tietokannasta. Tämä versio julkaistiin vuonna 1990. Lisäksi sitä on päivitetty pienin muutoksin sen jälkeen. Tietokantaan kutsutaan myös MIB-2 nimellä. Viralliselta nimeltään tietokanta on RFC1213-MIB. Verrattuna aikaisempaan versioon se on lähes samanlainen, muutamaa ryhmää lukuun

ottamatta. Uudemmassa versiossa on kolme uutta ryhmää CMOT, transmission ja SNMP. [5, 6.]

7.2 RMON-tietokanta

RMON (Remote network Monitoring) on laajennus normaalin SNMP-agentin toimintaan. Laajennus on toteutettu RMON-tietokannalla. RMON-tietokanta sisältää enemmän objekteja normaalien lähiverkkojen valvontaan. Aiemmin kuvattu MIB TCP/IP-verkkoihin sisältää tietoa verkon toiminnasta yhden verkon laitteen kannalta. Koska tietoverkkoa on tarkkailtava kokonaisuutena, on laajemman tietokannan käyttöönotto kannattavaa. RMON antaa verkon valvojalle yksityiskohtaista tietoa koko verkon toiminnasta.

RMON-tietokantojen soveltaminen nykyaikaisiin verkkoihin on hankalaa. Koska RMON-laite valvoo vain yhtä törmäysaluetta, ei tekniikkaa saada sovellettua kovin helposti. Mahdollisuutena on lisätä RMON-laite johonkin lähiverkkojen yhdistämispisteeseen, jolloin se pystyy analysoimaan verkkojen välillä liikkuvaa informaatiota. Tällöin olisi hyvä käyttää eri yhteyttä viestien lähettämiseen, jolloin verkkojen välinen liikenne ei häiriintyisi. Toisena mahdollisuutena on ohjata kytkimeltä kaikki tiedot yhteen porttiin. Tämä voi kuitenkin johtaa agentin ylikuormittumiseen.

RMON 1

RMON-tietokannan ensimmäinen versio tarjoaa tiedon keskittämiseen tarkoitetun elementin verkonvalvontaan. Yleensä agentit hoitavat valvonnan perustietokannoilla kuten MIB-2-tietokannalla. RMON-tietokannan avulla saadaan verkon tilasta lisätietoa. RMON pystyy valvomaan verkon tilaa seuraamalla verkossa liikkuvia paketteja. Tietokanta sisältää osioita mm. ethernet-pakettien varastoimiseen, tilastojen keräämiseen sekä yksittäisien työasemien pakettitietoja. Näiden tietojen keräämiseen on lisäksi tarjolla suodatuksia ja niistä pystytään johtamaan myös trap-viestejä.

Tietokantaa pystytään soveltamaan hyvin monien eri objektien valvontaan sekä trap-viestien luomiseen näistä objekteista. Erityisen hyödylliset ryhmät RMON-tietokannassa ovat alarm- ja event-ryhmät. Alarm-ryhmällä pystytään asettamaan mille tahansa RMON-tietokannan laskurille tai muuttujalle päivitysnopeus. Arvoille pystytään asettamaan rajat, joiden välillä muuttujan arvo saa vaihdella. Rajojen ylittymisen pohjalta event-ryhmällä pystytään muo-

dostamaan trap-viesti tai merkintä tietokantaan. Myös mistä tahansa muusta MIB:n osasta pystytään samaan aikaan trap-viesti.

Tietokannan sijainti on puumallissa TCP/IP MIB-tietokannan alla. Sen OID-numero on 1.3.6.1.2.1.16, jonka alla on kymmenen RMON-ryhmää. Ryhmät ovat seuraavat:

- statistics (1)
- history (2)
- alarm (3)
- host (4)
- hostTopN (5)
- matrix (6)
- filter (7)
- capture (8)
- event (9)
- tokenring (10).

Tietokannan soveltamismahdollisuuksia on useita. Hetkellisistä arvoista pystytään samaan aikaan trap-viestejä, joka mahdollistaa nopean reagoinnin vikatilanteisiin. Pitkällä aikavälillä pystytään keräämään tilastoja verkon käytöstä sekä muista siihen liittyvistä asioista. Näiden tilastojen avulla pystytään ennustamaan esimerkiksi verkon tulevat vikatilanteet. [1, s. 209-276].

RMON 2

RMON 2 -tietokanta laajentaa ensimmäisen version tiedonkeräystä. Tietokanta määrittelee 9 uutta ryhmää entisten 10 ryhmän perään. Ryhmät ovat seuraavat:

- protocolDir (11)
- protocolDist (12)
- addressMap (13)
- nlHost (14)
- nlMatrix (15)
- alHost (16)
- almatrix (17)
- usrHistory (18)
- probeConfig (19).

RMON 2 -tietokanta laajentaa ensimmäisen version tietojenkeräystä protokollariippuvaiseksi. Lisäksi agenttien välinen tiedonkeruu on suurennettu verkko- ja sovelluskerroksien tasolle. Tietokanta muuttui myös verrattuna edellisen versioon rakenteeseen, koska siinä on käytetty SMIv2-rakennemäärittelyä. Uusi määrittely saa tiedonrakenteeseen uusia ominaisuuksia, joita hyödynnetään RMON 2 -tietokannassa.

RMON-ryhmien kaksi ensimmäistä määrittelee protokolla riippuvaisen toiminnot. Nämä ryhmät sisältävät yksityiskohtaista tietoa muun muassa siitä, mitä protokollia on käytössä sekä niiden pakettimäärät.

Host- ja matrix-ryhmät on tarkoitettu yksittäisten koneiden valvontaan. Ryhmät jakautuvat seuraavasti: nlHost kertoo pakettitietoa verkkokerrokselta ja alHost selvittää sovelluskerroksen tiedot. Näiden ryhmien matrix-versioissa voidaan määritellä, minkä kahden koneen välillä paketteja seurataan.

Ryhmä 8 eli usrHistory sisältää datan varastoimiseen liittyviä toimintoja. Sillä pystytään esimerkiksi seuraamaan jonkun tietyn muuttujan arvoa ja varastoimaan sitä määrävälein. Viimeinen probeConfig-ryhmä määrittelee tietoja laitteesta, johon RMON-tietokanta on lisätty. [1, s. 277 - 328.]

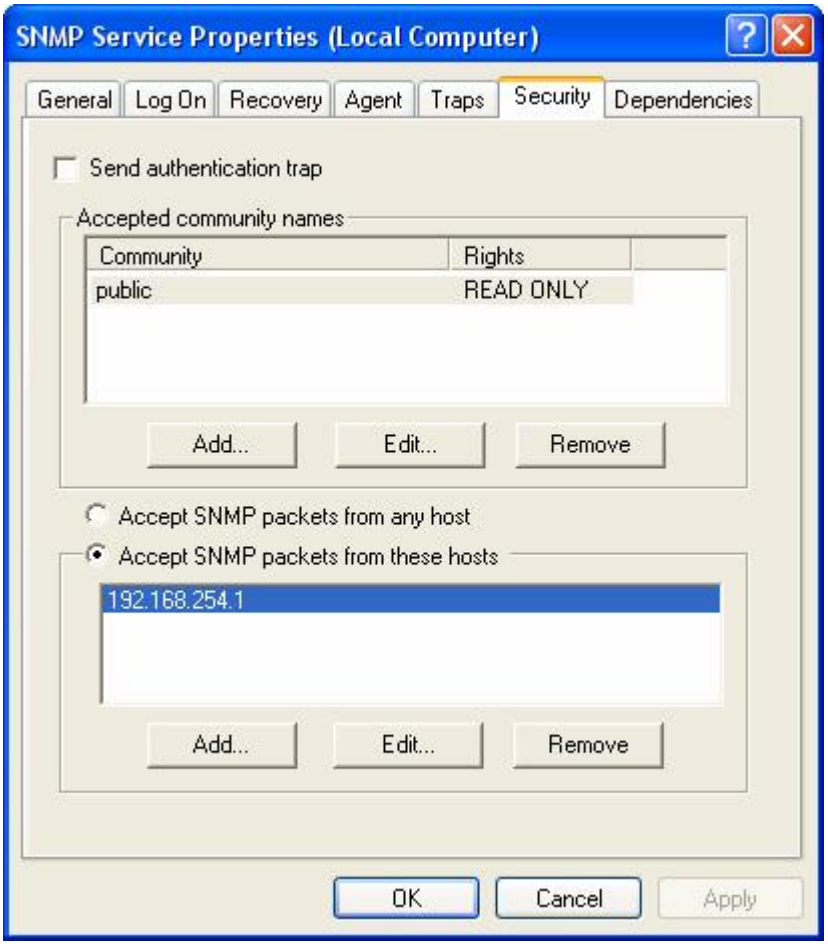
8 KÄYTTÖJÄRJESTELMIEN SNMP-AGENTIT

8.1 Windows-käyttöjärjestelmien SNMP-agentti

Windows-käyttöjärjestelmissä on tarjolla Microsoftin oma SNMP-agentti. Agentin saa käyttöön ohjauspaneelin kautta. Palvelut listasta löytyy SNMP-agentti sekä SNMP-trap-palvelut. Näitä ohjelmia testattiin Windows 2000 sekä Windows XP-käyttöjärjestelmissä. Jos agenttia ei löydy palvelulistasta, täytyy se asentaa erikseen Windows-käyttöjärjestelmän asennus-CD:ltä tai vaihtoehtoisesti valita jokin muu ohjelma toimimaan agenttina.

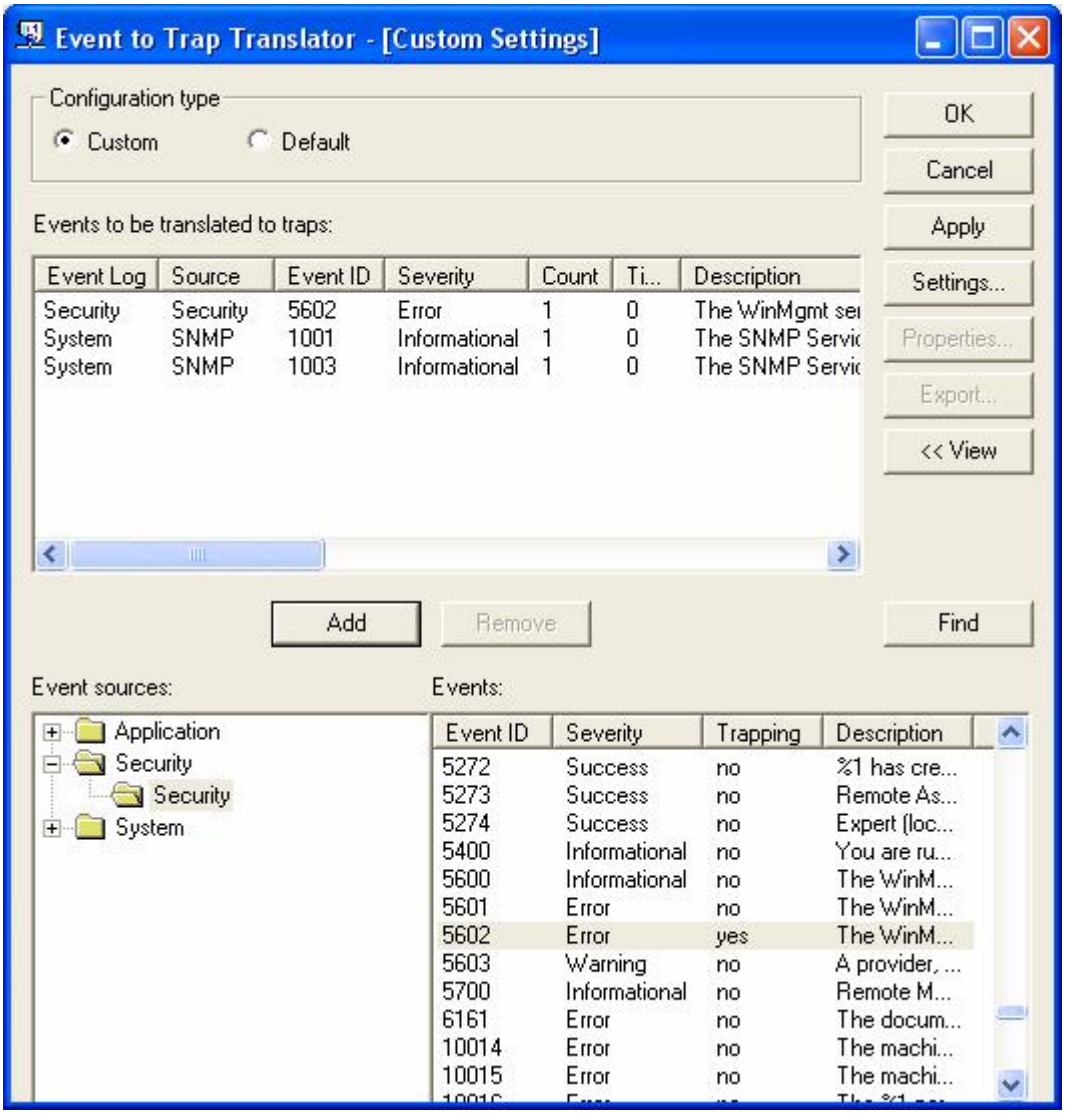
8.1.1 Windows-käyttöjärjestelmän SNMP-agentti

SNMP-agentti sopii myös verkonhallintaan. Tämän vuoksi SNMP-agentin käyttöönotossa on huomioitava tietoturva. Tietoturvan kannalta kannattaa miettiä, mitkä verkon tietokoneet saavat kirjoitusoikeuden kyseiseen tietokoneeseen.



Kuva 11. SNMP-palvelun ominaisuudet

Windows-järjestelmän oma SNMP-agentti ei tue kuin SNMP:n versioita yksi ja kaksi. Tämän vuoksi SNMP-palvelun ominaisuudet kohdasta löytyy turvallisuutta käsittelevä välilehti (Security), jossa saa muokattua listat, millä yhteisönimellä sekä mistä IP-osoitteesta SNMP-paketin on tultava, jotta se hyväksytään. Agentin omaisuuksista löytyy myös Traps-välilehti, jonka avulla Windows-käyttöjärjestelmä saadaan lähettämään automaattisia sanomia. Traps-välilehteen määritellään mihin osoitteeseen ja millä yhteisöllä viestejä lähetetään. (Kuva 11.)



Kuva 12. Event to Trap Translator -ohjelman graafinen näkymä

Automaattisten sanomien lähettämiseen tarvitaan vielä jokin lähde, josta saadaan informaatio lähetettäviin sanomiin. Yleensä lähteenä käytetään Windows-järjestelmän omaa ohjelmaa, joka pystyy muuttamaan järjestelmän tapahtumat trap-viesteiksi. Ohjelman saa näkyville kirjoittamalla komentorivil-le evntwin-komennon. Tämän ohjelman avulla pystytään määrittelemään kuinka monta tiettyä järjestelmän tapahtumaa tarvitaan, että trap-viesti muo-dostetaan (kuva 12). Ohjelma sisältää useita tuhansia eri tapahtumia, jotka on jaettu loogisiksi kokonaisuuksiksi. Evntwin-ohjelman ja SNMP-agentin avulla saadaan aikaan toimiva automaattisten sanomien lähetyspalvelu.

SNMP-agentti tukee asennuksen jälkeen yleisimpiä SNMP-tietokantoja. Näi-den tietokantojen avulla pystytään hakemaan tietoja esimerkiksi TCP-, UDP- ja IP-protokollien toiminnasta. Tietokanta sisältää myös prosessitietoja. Pro-

sessien lisäksi tietokannasta löytyvät asennetut ohjelmat sekä niiden ajoparametrit. Tietokannasta löytyy myös informaatiota muistien täyttöasteesta sekä tietokoneen sisältämistä lisälaitteista.

8.1.2 Windows-käyttöjärjestelmän SNMP-trap-palvelu

SNMP-trap-palvelulla pystytään vastaanottamaan verkosta tulevia SNMP-trap-sanomia. Sanomat hyväksytään, jos niiden yhteisökenttä vastaa trap-palvelun ominaisuuksissa määritettyä yhteisöä. Palvelussa pystytään lisäksi määrittelemään, mistä IP-osoitteesta viestien täytyy saapua, jotta ne hyväksyttäisiin. Vastaanotettuja viestejä ei kuitenkaan pysty havaitsemaan ilman jotain toimivaa hallintaohjelmaa. Windows-järjestelmästä ei löydy tällaista ohjelmaa, joten tulkitsemiseen täytyy etsiä joku muu ohjelma.

8.2 Linux-käyttöjärjestelmien SNMP-agentti

8.2.1 Avoimen lähdekoodin Net-SNMP-ohjelma

Linux-käyttöjärjestelmien tapauksessa ohjelmaksi valittiin Net-SNMP. Net-SNMP on avoimen lähdekoodin SNMP-paketti. Se on varustettu monipuolisilla ominaisuuksilla. Siihen on myös lisätty monia ohjelmia, jotka pystyvät laajentamaan SNMP:n toiminnallisuutta. Ohjelma on tarkoitettu pääasiassa Linux-käyttöjärjestelmiin, mutta se toimii myös muilla alustoilla. Tämän insinöörityön aikana Net-SNMP-pakettia testattiin seuraavissa Linux-distribuutioissa: Gentoo, Ubuntu (Debian) ja Red Hat Enterprise Linux 5.

Net-SNMP-paketti sisältää snmpd- ja snmptrapd-daemonit. Lisäksi siihen on sisällytetty paljon automaattisia ohjelman käyttöönottotiedostoja. Ohjelman perustoiminnot määritellään kolmen tiedoston avulla. Toiminnan määrittelytiedostot ovat snmp.conf, snmpd.conf ja snmptrapd.conf. Näiden tiedostojen avulla saadaan aikaan ohjelman perustoiminnot. Ohjelma sisältää myös valmiiksi asennettuna tarpeelliset tietokannat perusvalvontaan.

Net-SNMP-ohjelma tukee kahta eri symmetristä salausalgoritmia, DES-salausta sekä AES-salausta 128-bitin vahvuudella. Todennusmenetelmistä se hyödyntää MD5- sekä SHA-tiivistettä.

8.2.2 Ohjelman yleismäärittelyt

Ohjelman yleismäärittelyt voidaan toteuttaa snmp.conf-tiedostolla. Tiedosto voidaan luoda käyttämällä valmista snmpconf-ohjelmaa. Ohjelmalla pystytään tekemään myös kaksi muuta toiminnanmäärittelytiedostoa: snmpd.conf ja snmptrapd.conf. Toinen mahdollisuus tiedostojen luomiseen on tehdä ne ohjeiden avulla. Myös tiedostojen sijainti järjestelmässä on oleellinen. Ohjelma lukee käynnistysvaiheessa monta eri hakemistoa läpi, toiminnanmäärittely tiedostojen vuoksi. Tässä työssä niiden sijainti tiedostojärjestelmässä on /etc/snmp/. Jos tiedostoissa on määritelty SNMPv3 käyttäjiä sekä salasanoja, täytyy tiedoston sijaita hakemistossa /var/net-snmp/. Tällöin ohjelma poistaa automaattisesti salasana- ja käyttäjien määrittelyt käynnistyessä.

Yleismäärittelytiedosto snmp.conf sisältää hakukäskyissä käytettäviä vakioparametreja. Näitä parametreja voidaan käyttää helpottamaan hakutoimenpiteitä agentilla. Jos hakukäskyssä ei käytetä näitä vakioparametreja, joudutaan ne määrittelemään uudestaan joka käskyssä. Tiedostolla pystytään myös määrittelemään tietokantojen käyttöä sekä SNMPv3-käyttäjiä. Mahdollisuuksina on myös virheilmoitusten tulostaminen. Normaalien käskyjen tulostuksia pystytään säätelemään eri parametrien avulla. Snmp.conf-tiedostoa ei ole pakko käyttää, mutta se helpottaa hakukäskyjen käyttöä.

8.2.3 Agentin toiminnanmäärittely tiedostojen avulla

Agentin toimintaa ohjataan snmpd.conf-tiedoston avulla. Agentin toiminnan määrittely on hyvä aloittaa verkonvalvonta-ajatuksella. Agentin on siis tarkoituksena vastata pyyntöihin. Tätä varten on määriteltävä mistä verkosta tai IP-osoitteesta agentti hyväksyy paketteja. Yleensä määrittelyyn liittyy myös luku- tai kirjoitusoikeudet sekä kohde, mistä tietoa saa hakea ja mihin kirjoittaa. Tämän lisäksi on määriteltävä mihin yhteisöön itse agentti kuuluu. Yhteisö määrittelee myös, millä yhteisöillä on oikeus lukea tietokannasta. Tämä pätee myös kirjoitusoikeuksiin. Näiden määrittelyiden avulla saadaan agentti vastaamaan tuleviin SNMP-paketteihin.

```
rocommunity public 192.168.1.0/24 1.3.6.1.2.1.1
```

Määrittelyn syntaksi on seuraava. Ensin määritellään halutaanko sallia luku- ja kirjoitus oikeudet agentin tietokantaan vai vain lukuoikeus. Tässä tapauksessa sallittiin vain rocommunity (read-only community) eli pelkkä lukuoi-

keus. Seuraavaksi määritellään agentin yhteisö joka on public. Sitten määritellään IP-osoite tai verkko ja verkon maski. Viimeisenä on tietokannan osa. Tässä tapauksessa agentilta pystytään lukemaan vain MIB-2-tietokannan system-ryhmä.

```
rwcommunity private 192.168.1.13 1.3.6.1.2.1
```

Jos halutaan sallia jollekin muulle yhteisönimelle esimerkiksi kirjoitusoikeus, voidaan lisätä toinen rivi aloitusmääreellä rwcommunity. Tässä tapauksessa pitää kuitenkin ottaa huomioon tietoturva. Kuten luvussa 4.1 mainittiin, SNMPv1-ja SNMPv2-pakettien yhteisökentät kulkevat verkossa selväkielisinä. Tämän vuoksi kirjoitusoikeuden väärinkäyttö on tietoturvauhka.

```
#          sec.name  source    community
com2sec    kaikki    default   public

#          sec.model           sec.name
group      Luku      v1        kaikki
group      Luku      v2c       kaikki

#          viewname  incl/excl subtree   mask
view       all       included  .1        80

#     context sec.model sec.level match read write notif
access Luku "" any      noauth    exact all  none  none
```

Tietokannan käyttöoikeuksia voidaan käyttöönottaa myös taulukkomuodossa. Tätä määrittelyä kutsutaan VACM-malliksi. Tällä menetelmällä voidaan myös sallia luku- tai kirjoitusoikeuksia tiettyihin tietokannan osiin. Parametrit voivat sisältää myös SNMPv3-määrittelyjä.

```
agentaddress 192.168.254.2:tcp:161
```

Agentin käynnistysparametreja voidaan myös määritellä snmpd.conf-tiedostossa. Net-SNMP-paketti tukee useita siirtotason protokollia. Myös niiden määrittelyt ovat asetettavissa samassa tiedostossa. Jos siirtoprotokolla- ja porttimäärittelyä ei ole, agentti käyttää UDP-protokollaa ja porttia 161. Lisäksi samalla parametrilla voidaan määritellä, mitä IP-osoitetta agentti kuuntelee.

```
syslocation 3. kerros, laitekaappi 2
syscontact admin@domain.com
```

Tiedostossa voidaan määritellä myös TCP/IP-MIBin system-ryhmän objekteja. Normaalisti nämä objektit ovat kirjoitettavia. Kun objektit määritellään

tiedostossa, niitä ei voi muuttaa agentin käynnistyksen jälkeen set-komennolla. [12.]

8.2.4 Laitteen resurssien valvonta agentin avulla

Snmpd.conf-tiedostoa pystytään käyttämään myös monien eri resurssien valvontaan. Resurssien valvontaa pystytään suorittamaan Californian yliopiston luomalla UCDavis-tietokannalla. Tietokanta sisältää valmiit objektit virheilmoituksille sekä lippuobjektit. Lippuobjektien avulla saadaan helposti selville vikatilanteet. UCDavis-tietokanta sisältää valvontakohteita esimerkiksi muisteille, laitteille, prosesseille sekä prosessorin kuormitukselle. Seuraavissa esimerkeissä on esitelty näistä muutamia.

Agentti pystyy valvomaan, että tietyt prosessit ovat toiminnassa. Prosesseista pystytään myös valvomaan, kuinka monta saman nimistä prosessia on käynnissä ja ilmoittamaan, jos prosesseja on virheellinen määrä.

```
proc snmptrapd
proc mountd 2 1
```

Prosessin tarkistaminen voidaan suorittaa yksinkertaisella määrittelyllä. Prosessien tarkistukset aloitetaan proc-määrittelyllä, jonka jälkeen tulee prosessin nimi. Jos halutaan tarkistaa vain, että prosessi on käynnissä, riittää pelkkä prosessin nimi. Kun päätetään, että prosessien lukumäärä täytyy pysyä tietyllä välillä, määritellään prosessinimen perään maksimi- ja minimimäärät.

```
load 70 60 50
```

Prosessorin kuormituksen tarkistaminen onnistuu yhdellä määrittelyllä. Määrittelyn perässä voidaan antaa arvot prosessorin kuormituksessa prosentteina eri aikajaksoille. Ensimmäinen luku load-sanan perässä määrittelee kuormituksen minuutin keskiarvolle, toinen viiden minuutin ja kolmas 15 minuutin keskiarvolle.

```
includeAllDisks 20%
```

Kolmas tässä osiossa esiteltävä valvontakohde on kovalevyjen tilan valvonta. Kovalevyjen osioita pystytään valvomaan yksitellen tai määräämään kaikille sama vapaantilan määrä. Tilan määrä pystytään antamaan kB-muodossa tai prosentteina. Tiedostolla pystytään myös valvomaan esimerkiksi tiedostojen maksimikokoa.

Edellä esitetyt resurssien valvontakomennot tekevät ainoastaan merkinnät UCDavis-tietokantaan. Jotta näistä tapahtumista saataisiin muodostettua esimerkiksi trap-viestejä, vaaditaan tiedostoon vielä lisäkomentoja. Trap-viestien tapauksessa vaaditaan IP-osoite, johon trap-viestejä lähetetään sekä viestien yhteisönimi.

Trap-toiminnallisuuden aikaansaamiseksi joudutaan määrittelemään Disman-event-MIBin (Distributed Management Event MIB) komentoja samaan tiedostoon. Tietokanta on IETF:n määrittelemä tietokanta trap-viestien luomiseen erilaisista objekteista. Disman-tietokannalla pystytään valvomaan muun muassa jonkin OID:n olemassa oloa ja ilmoittamaan sen ilmestymisestä agentin tietokantaan. Tällä tietokannalla saadaan aikaan itse valvonta, valvomalla UCDavis-MIBin lippuarvoja.

```
trapcommunity public
trapsink 192.168.23.45

rouser mibi noauth
iquerySecName mibi
```

Trap-viestien perusasetukset saadaan aikaan kahden määrittelyn avulla. Yhteisön määrittely luodaan trapcommunity-nimikkeellä sekä yhteisön nimellä, joka on tässä tapauksessa public. Trap-viestien kohdeosoitteen määrittely tapahtuu trapsink-komennolla ja kohde IP-osoitteella. Seuraava vaihe määrittelyssä on muodostaa peruskäyttäjä Disman-tietokannalle ja liittää se iquerySecName-käskyyn. Kolmas käsky määrittelee käyttäjän mibi lukuoikeudella, jonka tietokannan käyttöön ei tarvita todentamista eikä salausta. Tämän vuoksi komennon viimeinen sana on noauth.

Näiden komentojen jälkeen tiedosto on melkein valmis itse valvontaa varten. Disman-MIBin valvottavat kohteet täytyy vielä erikseen määritellä.

```
linkUpDownNotifications yes
defaultMonitors yes
monitor -o prNames prErrorFlag != 0
```

Laite, johon snmp-agentti on asennettu, voi myös valvoa laitteen liitäntöjen tilaa. Aina muutoksen sattuessa agentti lähettää trap-viestin. Ensimmäinen komento määrittää liitäntöjen trap-viestien lähetyksen. Edellä esitetty toinen komento määrittää kaikki aikaisemmin määritellyt resurssit valvonnan kohteiksi. Resursseja voidaan valvoa myös yksitellen määrittelemällä esimer-

kiksi prosessien valvontataulukko valvonnan kohteeksi. Kolmas komento on esimerkki tästä tapauksesta. [12.]

8.2.5 Trap-viestien vastaanotto

Net-SNMP-ohjelmassa määritellään trap-viestien vastaanotto snmptrapd .conf-tiedostolla. Trap-viestien vastaanotto oli automaattista ennen Net-SNMP versio 5.3. Tällöin ohjelma ei tarvinnut ollenkaan määrittelytiedostoa. Uudemmissa versioissa joudutaan määrittelemään tiedostolla snmptrapd:n toiminta. Snmptrapd.conf-tiedoston avulla pystyään kirjoittamaan tulevista trap-viesteistä tiedot muistiin ja ohjamaan ne erilliselle ohjelmalle tai laitteelle. Seuraavilla komentomäärittelyillä saadaan aikaan trap-viestien vastaanotto.

Trap-viestien vastaanoton määrittelyssä kannattaa lähteä liikkeelle siitä, mitä viestejä snmptrapd-prosessin halutaan vastaanottavan. Tällöin määritellään, millä yhteisönimellä saapuvan viestin pitää tulla, jotta se hyväksyttäisiin.

```
trapcommunity  public
snmpTrapdAddr 192.168.1.2
authcommunity read,write public 192.168.1.0/24
```

Trap-yhteisön määrittely tapahtuu trapcommunity-käskyllä. Käskyn jälkeen on vuorossa saapuvan trap-viestin yhteisönimi. Normaalisti snmptrapd-prosessi kuuntelee kaikkia IP-osoitteita. Jos tätä halutaan muuttaa, määritellään snmpTrapAddr-komennolla ja IP-osoitteella, mistä trap viestin pitää tulla, jotta se hyväksyttäisiin. Tällä komennolla voidaan määritellä monia IP-osoitteita. Kolmas käsky määrittelee samat asiat kuin kaksi ensimmäistä käskyä. Kolmannen käskyn tapauksessa on helpompi määritellä useampi laite, miltä hyväksytään trap-viestejä verkon ja verkon maskin avulla.

Trap-viestien vastaanottoon tarvitaan lisäksi määrittelyt, minne vastaanotetut viestit ohjataan. Ilman tätä näitä määrittelyjä viestejä ei löytyisi mistään vastaanoton jälkeen.

```
logOption f /etc/snmp/trap.log
```

Viestien tallentaminen voidaan tehdä moneen paikkaan logoption-komennolla. Tässä esimerkissä trap-viestit tallennetaan tiedostoon. Käskyn jälkeen on määrittelynä f (File) eli tiedosto. Tämän jälkeen määritellään hakemisto sekä tiedosto, mihin viestit ohjataan. Muita vaihtoehtoja mihin viesti voidaan ohjata on standardi virheilmoitus e (standard error stream), standardi ulos-

tulo o (standard output stream) sekä järjestelmäilmoitus s (system log). Järjestelmäilmoitusten tapauksessa vaaditaan vielä lisämäärittelyjä.

Edellä olevalla esimerkillä saadaan ohjattua trap-viestit tiedostoon. Jotta trap-viesteistä saataisiin tarpeeksi informaatiota, voidaan niiden syntaksi määritellä.

```
format1  "%#02.2h:%#02.2j TRAP%w.%q from %A\n"

format2  "%#02.2h:%#02.2j TRAP %N from %A\n"
```

Trap-viestien rakennetta voidaan rajata format-määrittelyllä. Format1-määrittely käsittelee SNMPv1 trap-viestejä ja format2 svmpv2 ja snmpv3 trap-viestejä. Edellä olevat määrittelyt eroavat ainoastaan trap-kohdan jälkeen. SNMPv1 trap-viestit sisältävät yleisen ja tarkennetun trap-tunnisteen, jotka ovat määritelty käskyssä %w.%q. SNMPv2 ja 3 määrittelyssä on tässä kohtaa trap-viestin OID-numero, joka on määritelty %N-merkinnällä.

```
doNotRetainNotificationLogs yes
ignoreAuthFailure no
donotlogtraps false
```

Tiedostossa pystytään lisäksi määrittelemään muita yleisiä trap-viestien asetuksia. Ensimmäinen edellä esitetyistä määrittelyistä kieltää trap-viestien tallentamisen NOTIFICATION-LOG-MIBiin. Toisella käskyllä voidaan muodostaa viestejä todentamisen epäonnistumisesta ja kolmannella päätetään tallennetaanko trap-viestit. Tätä komentoa voidaan käyttää hyväksi esimerkiksi, jos viestit ohjataan erilliselle ohjelmalle.

```
outputOption s
```

Trap-viestejä pystyttiin ohjamaan myös normaalille komentoriville, tällöin format-määrittelyistä ei ole hyötyä. Tähän tarkoitukseen on outputOption-komento. Tällä komennolla voidaan muuttaa trap-viestien rakennetta ennen kuin ne ohjataan eteenpäin.

Trap-viestien vastaanottoa voidaan laajentaa monella tavalla. Trap-viestit voidaan mm. välittää toiselle agentille, kirjoittaa tiedostoon tai välittää trap-viestien käsittelyohjelmalle. Trap-viestit pystytään myös ohjaamaan valmiilla ohjelmalla sähköpostiin. Edellä kuvatuilla tavoilla Net-SNMP-ohjelman toimintaa pystytään laajentamaan lähes loputtomiin. [12.]

8.2.6 Net-SNMP-ohjelman tietokannat ja uusien tietokantojen lisääminen

Net-SNMP sisältää valmiiksi muutamat tietokannat. Näillä tietokannoilla pystytään suorittamaan verkon perusvalvontaa. Näitä tietokantoja on käyty läpi jo aikaisemmissa luvuissa. Net-SNMPstä löytyvät mm. seuraavat tietokannat

- RFC1213-MIB
- UCD-SNMP-MIB
- NOTIFICATION-LOG-MIB
- HOST-RESOURCES-MIB
- DISMAN-EVENT-MIB.

Ohjelma sisältää paljon muitakin tietokantoja. Jotta nämä saataisiin käyttöön, täytyy ohjelman asennusvaiheessa sisällyttää tietokannat configure-käskyyn. Toinen vaihtoehto on tuottaa tietokantojen ohjelmakoodi ohjelman asennuksen jälkeen.

Tietokantojen toiminnallisuus koostuu kahdesta osasta: itse tietokanta tiedostosta sekä tietokannan tietojen hakuohjelmasta. Tietokanta tiedosto määrittelee objektit sekä objektien tyypit. Tämä tiedosto on puhdas tekstitiedosto, jonka voi luoda normaalilla tekstieditorilla. Tietokanta tiedoston nimi on yleensä standardin mukainen esimerkiksi DISMAN-EVENT-MIB.txt. Agentti ei välttämättä tarvitse tekstimuotoista tiedostoa. Jos tiedostoa ei ole, agentti esittää kaikki tietokannan tiedot numeromuodossa.

Tietokannan toiminnallisuus voidaan saada aikaan muutamalla tavalla, agentti voi esimerkiksi kysyä tietoja erilliseltä agentilta. Hakemisen voi toteuttaa myös ohjelmakoodilla. Ohjelmakoodin tapauksessa mahdollisia ohjelmointikieliä ovat C++, Perl tai Python.

Tietojenhaku voidaan siis toteuttaa esimerkiksi C++-ohjelmointikielellä. Seuraavaksi tarkennetaan hieman miten C++-koodista saadaan luotua tietokanta. Ensimmäinen vaihe C++-ohjelman luomisessa on luoda objekti.c-tiedosto, joka on samanniminen tiedosto kuin tietokannan objekti. Objekti.c tiedostoon määritellään mistä itse tieto haetaan ja liitetään se tietokannan objektin nimeen. Lisäksi tarvitaan objekti.h-tiedosto, jossa luodaan .c-tiedostoon luotujen objektien määrittelyt.

Ohjelman luomiseen on myös apuohjelma, joka luo pohjan C++-koodille. Ohjelman nimi on mib2c, ja se tarvitsee tekstimuotoisen mib-tiedoston, josta

se kehittää pohjan C++-koodille. Tämä ohjelma luo siis vain pohjan, johon ohjelmoijan täytyy kirjoittaa itse toiminnallisuus.

C++-ohjelman luomisen jälkeen se pitää vielä muuttaa agentin hyväksymään muotoon. Tämä saadaan aikaan käyttämällä valmista make-ohjelmaa. Edellä esiteltyjen tiedostojen lisäksi tarvitaan Makefile-tiedosto. Makefile sisältää tiedoston muuttamiseen tarvittavia määrittelyjä. Ohjelma ottaa sisään Makefile, objekti.c- ja objekti.h-tiedostot ja muodostaa näistä objekti.so-tiedoston. Make voi tarvita lisäkirjastoja ohjelman muuntamiseen. Nämä ylimääräiset kirjastot löytyvät helpoiten snmpd:n otsikkotiedoista. Kun objekti.so on saatu luotua, se täytyy liittää vielä snmpd.conf tiedostoon.

```
dlmod objekti /usr/share/snmp/objekti.so
```

Luotu C++-kielinen ohjelma on nyt saatu muutettua .so-muotoon. Tämän tiedoston voi liittää agentin snmpd.conf-tiedostoon, joka tapahtuu dlmod-komennolla. Komennon perään määritellään tietokannan objektin nimi, hakemisto ja ohjelman nimi. Näiden toimenpiteiden jälkeen agentti käynnistetään uudelleen, jonka jälkeen uusi objekti on käytössä. [12.]

8.2.7 Net-SNMP:n käyttöönotto

Net-SNMP-ohjelman käyttöönotossa tuli esille monia ongelmia. Laitteesta, johon ohjelma asennetaan, on hyvä tarkistaa palomuuriasetukset sekä reititystaulukko. Kun nämä ovat kunnossa, pitäisi ohjelman käyttöönoton olla yksinkertaista.

Tämän jälkeen ohjelmaan täytyy määritellä agentin toiminta. Jos ohjelman halutaan vastaanottavan myös trap-viestejä, täytyy myös trap-määrittelyt lisätä. Kun ohjelman määrittelytiedostot snmpd.conf ja snmptrapd.conf ovat oikeassa hakemistossa, täytyy prosessit käynnistää uudelleen. Tämän jälkeen ohjelma on käyttökunnossa.

Joissakin Linux-käyttöjärjestelmissä Net-SNMP-ohjelman snmpd-tiedostossa on määritelty ylimääräinen IP-osoite. Tällöin ohjelma ei vastaanota viestejä kuin kyseisestä IP-osoitteesta. Tämä IP-osoite täytyy poistaa tai vaihtaa, jos agentin halutaan vastaanottavan SNMP-viestejä muista IP-osoitteista.

9 VALVONTAJÄRJESTELMÄN TESTAAMINEN

9.1 Ensimmäinen testiympäristö

Ensimmäinen testiympäristö luotiin ohjelmistojen testausta varten. Testiympäristöön kuului työasema, sulautettujen järjestelmien tietokone sekä hallittava kytkin. Työaseman tarkoituksena oli selvittää tietoja muilta testijärjestelmän laitteilta. Kytkin ja tietokone sisälsivät valvonnan kannalta oleelliset ohjelmat. Testiympäristössä testattiin miten tietojen haku onnistuu muilta laitteilta sekä selvitettiin miten työaseman ohjelmaa pystyttiin laajentamaan.

Testiympäristö saatiin toimimaan oletetulla tavalla. Informaatio saatiin kulkemaan laitteilta työasemalle sekä työaseman ohjelmaa pystyttiin laajentamaan. Seuraava vaihe oli testata itse hallintaohjelmaa.

9.2 OPC-rajapinta

OPC-rajapinta (Object Linking and Embedding for Process Control) on yksi yleisimmistä rajapinnoista automaatiotekniikan puolella. Sen yleistymistä voidaan selittää sen standardoitumisella sekä yksinkertaisuudella.

OPC on Microsoftin COM- (Component Object Model) ja DCOM-arkkitehtuureihin (Distributed Component Object Model) perustuva sarja standardeja. OLE-rajapintaa, joka on OPC:n aiempi kehitysmuoto, käytetään vielä nykyään Windows-sovelluksissa esimerkiksi Word-tekstinkäsittely- ja Excel-taulukkolaskentaohjelmien välillä. OPC:n tarkoituksena on tarjota standardi rajapinta eri valmistajien välille. OPC tarjoaa standardin määrän objekteja, liitäntöjä ja tekniikoita automaatiotekniikan tarpeisiin. Koska OPC on kehitetty COM- ja DCOM- arkkitehtuurien pohjalta, siihen on helppo kehittää ohjelmistoja. Tämä sai aikaan sen, että OPC-tuotteet ovat yleistyneet. [11.]

Tämän työn testauksessa OPC-rajapintaa käytetään tiedon välitykseen IP-verkon ja automaatiotekniikkaan kehitetyn valvontasovelluksen välillä. Käytössä on kaksi osapuolta: palvelin ja palvelimen asiakas. Palvelimen tehtävä on hankkia tietoa IP-verkosta ja tarjota sitä OPC-rajapinnan yli asiakkaalle. Asiakkaan tehtävänä on vastaanottaa tietoa palvelimelta sekä välittää sitä valvontasovellukselle. Yhteys pystyy toimimaan myös toiseen suuntaan eli valvontasovellukselta IP-verkkoon.

9.3 Toinen testiympäristö

Toisen testiympäristön tarkoituksena oli muodostaa päästä päähän -ratkaisu. Testiympäristössä oli mukana automaatiotekniikkaan kehitetty valvontasovellus, työasema sekä OPC-ohjelma. Testijärjestelmä toimi seuraavalla tavalla: työasema sisälsi informaatiota, jota haluttiin toimittaa valvontasovellukselle. Ensin informaatio siirrettiin OPC-palvelimelle, joka toimitti informaation edelleen OPC-asiakkaalle ja sitä kautta valvontasovellukseen.

Informaatio saatiin kulkemaan sille tarkoitetulla tavalla, jolloin päästä päähän -ratkaisu saatiin aikaan. Tässä vaiheessa tuli esille myös jatkokehitysmahdollisuuksia. Testituloksista ja jatkokehitysmahdollisuuksista kerrotaan lisää liitteessä, joka on luottamuksellinen.

10 YHTEENVETO JA JOHTOPÄÄTÖKSET

Insinöörityössä luotiin verkonvalvontajärjestelmä. Tämä verkonvalvontajärjestelmä soveltuu eri käyttökohteisiin. Järjestelmällä voidaan valvoa monia erityyppisiä laitteita SNMP-protokollan avulla. Työssä selvitettiin, miten Net-SNMP-ohjelmalla saadaan aikaan toimiva yhden laitteen kokoonpano. Toinen selvitettävä asia oli Windows-käyttöjärjestelmän toimivuus valvontajärjestelmässä. Lisäksi selvitettiin, miten kyseiset laitteet saadaan toimimaan verkonvalvontajärjestelmän osana sekä tarjoamaan valvonnan kannalta oleellista informaatiota.

Windows-käyttöjärjestelmien tapauksessa nousi esille SNMP:n versiot. Koska käyttöjärjestelmä tukee ainoastaan SNMP-versioita 1 ja 2, on tietoturva huomion arvoinen asia. Yleensä Windows-laitteita käytetään kuitenkin sisäverkossa, jolloin liikutaan turvallisemmalla puolella. Tämän vuoksi tietoturva ei yleensä ole este SNMP-toiminnallisuuden käyttöönotolle.

Net-SNMP-ohjelma on hyvä huomioida valittaessa SNMP-ohjelmistoja Linux-pohjaisiin käyttöjärjestelmiin. Ohjelman avulla saadaan helposti aikaan monipuolisia ominaisuuksia, joita pystytään hyödyntämään verkonvalvonnassa. Ohjelma havaittiin hyväksi vaihtoehdoksi Linux-järjestelmiin, sen helpon laajennettavuuden vuoksi. Net-SNMP-ohjelman avulla pystytään saamaan aikaan esimerkiksi tiedon keskittäminen. Tällä tekniikalla voidaan valvoa haluttuja objekteja aliagenttien tietokannoista. Samalla valvova agentti pitää huolta aliagenttien toiminnasta ja seuraa yhteyksien toimivuutta. Tämä toiminto korostuu erityisesti valvottaessa sellaisia laitteita, jotka tukevat vain SNMP:n perustoiminnallisuutta.

Työn tuloksina saadut testijärjestelmät olivat onnistuneita ratkaisuja. Näiden järjestelmien pohjalta pystytään saamaan aikaan Netcontrol Oy:n tarkoituksiin sopivia valvontajärjestelmiä. Valvontajärjestelmän kehitysehdotukset puuttuvat tästä julkisesta insinöörityöstä. Tarkemmat kuvaukset jatkokehitysmahdollisuuksista löytyvät liitteestä, joka on luottamuksellinen.

LÄHTEET

[1] Stallings, William. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison Wesley Longman, Inc. 1999.

[2] Cisco Systems Inc. *AES and 3-DES Encryption Support for SNMP Version 3*. [Verkkodokumentti]. 2007. [Viitattu 1.4.2008]. Saatavissa: http://www.cisco.com/en/US/docs/ios/12_4t/12_4t2/snmpv3ae.pdf

[3] OID Repository. [Verkkodokumentti]. 2008. [Viitattu 1.4.2008]. Saatavissa: http://www.oid-info.com/

[4] Internet Engineering Task Force. *Structure and Identification of Management Information for TCP/IP-based internets*. [Verkkodokumentti]. 1991. [Viitattu 1.4.2008]. Saatavissa: http://tools.ietf.org/html/rfc1065

[5] Internet Engineering Task Force. *Management Information Base for Network Management of TCP/IP-based internets*. [Verkkodokumentti]. 1990. [Viitattu 1.4.2008]. Saatavissa: http://tools.ietf.org/html/rfc1156

[6] Internet Engineering Task Force. *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*. [Verkkodokumentti]. 1991. [Viitattu 1.4.2008]. Saatavissa: http://tools.ietf.org/html/rfc1213

[7] Internet Engineering Task Force. *Structure of Management Information Version 2 (SMIv2)*. [Verkkodokumentti]. 1991. [Viitattu 1.4.2008]. Saatavissa: http://tools.ietf.org/html/rfc2578

[8] Internet Engineering Task Force. *Remote Network Monitoring Management Information Base*. [Verkkodokumentti]. 2000. [Viitattu 1.4.2008]. Saatavissa: http://tools.ietf.org/html/rfc2819

[9] Internet Engineering Task Force. *The ESP Triple DES Transform*. [Verkkodokumentti]. 1995. [Viitattu 1.4.2008]. Saatavissa: http://tools.ietf.org/html/rfc1851

[10] National Institute of Standards and Technology. *Approved Algorithms*. [Verkkodokumentti]. 2007. [Viitattu 1.4.2008]. Saatavissa: http://csrc.nist.gov/groups/ST/toolkit/block_ciphers.html

[11] OPC Foundation. *What is OPC?* [Verkkodokumentti]. 2007. [Viitattu 1.4.2008]. Saatavissa: http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp

[12] Net-SNMP. [Verkkodokumentti]. 2007. [Viitattu 2.4.2008]. Saatavissa: http://www.net-snmp.org