jamk.fi

JYVSECTEC RGCE -ympäristön palveluiden monitorointi Open Source -tuotteella

Sakari Remonen

Opinnäytetyö
Huhtikuu 2015

Tietotekniikan koulutusohjelma
Tekniikan ja liikenteen ala

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Remonen, Sakari	Opinnäytetyö	13.04.2015
	Sivumäärä 74	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi
JYVSECTEC RGCE -ympäristön palveluiden monitorointi Open Source -tuotteella

Koulutusohjelma
Tietotekniikan (tietoverkot) koulutusohjelma

Työn ohjaaja(t)
Mika Rantonen
Antti Häkkinen

Toimeksiantaja(t)
Jyvsectec
Marko Vatanen

Tiivistelmä

Opinnäytetyön toimeksiantajana toimi Jyväskylän Ammattikorkeakoulun IT-instituutin kanssa yhteistyötä tekevä Jyvsectec. Opinnäytetyön tavoitteena oli toteuttaa toimeksiantajalle RGCE-verkon monitorointipalvelu, jolla voidaan monitoroida verkon kriittisiä palveluita. Tarkoituksena oli myös etsiä ja vertailla erilaisia ilmaisia tai Open Source -tuotteita, jotka sopivat kyseiseen tarkoitukseen. Opinnäytetyö toteutettiin toimeksiantajan virtuaaliseen tuotantojärjestelmään virtuaalisena Linux CentOS -palvelimena.

Verkon monitoroinnilla on tarkoitus helpottaa verkon ylläpitäjiä hallitsemaan ja monitoroimaan verkossa tapahtuvia ongelmia ja täten ennaltaehkäistä niitä. Verkon monitorointi toteutettiin tuotteella OpenNMS käyttäen valvontamekanismeina SNMP -protokollaa ja -agentteja.

Työ aloitettiin yhteistyössä toimeksiantajan kanssa määrittelemällä monitorointiohjelmiston vaatimukset. Toteutettava työkalu valittiin toimeksiantajan kanssa yhteistyössä, kun tuotteita ja niiden ominaisuuksia oli vertailtu.

Työn tuloksena saatiin Jyvsectecin RGCE -verkkoon toimiva ja helposti muokattavissa oleva monitorointipalvelin. Palvelin siirrettiin lopuksi toimeksiantajan tuotantojärjestelmään ja otettiin käyttöön.

Avainsanat (asiasanat)

Verkonvalvonta, SNMP, JYVSECTEC, RGCE, raportointi, monitorointi, OpenNMS

Muut tiedot

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

Description

Author(s)	Type of publication	Date
Remonen, Sakari	Bachelor's thesis	13.04.215
		Language of publication: Finnish
	Number of pages 74	Permission for web publication: x

Title of publication
Monitoring of JYVSECTEC RGCE-network services with Open Source product

Degree programme
Information Technology

Tutor(s)
Mika Rantonen
Antti Häkkinen

Assigned by
Jyvsectec
Marko Vatanen

Abstract

This bachelor's thesis was assigned by Jyvsectec (Jyväskylä Security Technology). Jyvsectec works with JAMK University Of Applied Sciences and its institute of ICT. The main goal of this thesis was to implement a network monitoring software to the client's RGCE network with an Open Source product and to find a suitable free network monitoring software to monitor the network's critical services such as NTP, DNS and Web services.

This thesis was implemented to the client's virtual production system as a virtual CentOS Linux server.

The purpose of network monitoring is to facilitate network administrators to manage and monitor network problems that occur and thus to prevent them. Network monitoring was carried out using the Open Source product OpenNMS with SNMP protocol and agents.

The project started in collaboration with the client by specifying the requirements for the monitoring software, after which the software was chosen in collaboration with the client as the different Open Source products were first tested and compared.

This thesis resulted in an implementation of an Open Source OpenNMS network manager solution to the client's RGCE network, which is easy to customize and maintain by the administrators.

Keywords/tags (subjects)
Network monitoring, SNMP, JYVSECTEC, RGCE, reporting, OpenNMS

Miscellaneous

SISÄLTÖ

KUVIOT

TAULUKOT

Lyhenteet

ACK	Acknowledge
CMIP	Common Management Information Protocol
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
GPG	GNU Privacy Guard
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Organization for Standardization
NTP	Network Time Protocol
JAMK	Jyväskylän ammattikorkeakoulu
JDK	Java Development Kit
LAN	Local Area Network
MIB	Management Information Base
OID	Object Identifier
PDF	Portable Document Format

RGCE	Realistic Global Cyber Environment
SSH	Secure Shell
SGMP	Simple Gateway Management Protocol
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VOIP	Voice Over Internet Protocol
VPN	Virtual Private Network

1 Työn kuvaus

1.1 Jyväskylän ammattikorkeakoulu

Tämä opinnäytetyö tehtiin Jyväskylän ammattikorkeakoululle, joka on hyvin vetovoimainen sekä kansainvälinen korkeakoulu. Jyväskylän ammattikorkeakouluun kuuluu opettajakorkeakoulu, hyvinvointiyksikkö, liiketoimintayksikkö ja teknologiayksikkö. Toimipisteitä sijaitsee eri puolilla Jyväskylää ja Tarvaalassa Saarijärvellä. (Tutustu ja menesty 2014.)

Aktiivisia opiskelijoita Jyväskylän ammattikorkeakoulussa on noin 8500. JAMK tarjoaa korkeakoulututkintoon johtavaa koulutusta, ammatillista opettajakoulutusta, avoimia ammattikorkeakouluopintoja, täydennyskoulutusta sekä oppisopimusmahdollisuutta täydennyskoulutukseen nuorille sekä aikuisille. (Mt.)

JAMKin asema on hyvin vahva Jyväskylän seudun ja Keski-Suomen kehittäjien joukossa. JAMKilla on hyvät suhteet Keski-Suomen alueen yrityksiin sekä yhteisöihin. Jyväskylän ammattikorkeakoulun tunnistaa helposti seuraavasta kuviossa 1 näkyvästä logosta. (Mt.)

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

Kuvio 1. JAMK logo (Jyväskylän ammattikorkeakoulu 2015.)

1.2 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi Jyvsectec (Jyväskylä Security Technology), joka on Jyväskylässä toimiva kyberturvallisuuteen keskittyvä tutkimus-, koulutus- ja kehityskeskus. Jyvsectec kuuluu Jyväskylän ammattikorkeakoulun IT-instituuttiin. Jyvsectecillä on käytössä kyberturvallisuuden kehitysympäristö RGCE (Realistic Global Cyber Environment), jossa voidaan tuottaa erilaisia palveluita yhteistyöverkoston käyttöön. (Jyvsectec 2014.)

Jyvsectecin toimintaan kuuluvat muun muassa:

- Kehitysympäristön (RGCE) sekä tilannekeskuksen kehittäminen ja ylläpitäminen.
- Kehityspalvelut liittyen kyberharjoituksiin, tilannekuvaan, koulutukseen ja tuotteisiin.
- Eri kyberturvallisuus toimijoiden väliset kansainväliset ja alueelliset yhteistyömuodot.
- Puolueeton keskus avoimelle tiedonvaihdolle.
- Kansainvälisten kyberturvallisuus toimijoiden tietoisuus Keski-Suomen tarjoamista kehitys mahdollisuuksista.

Uudet kansalliset ja kansainväliset tutkimus- ja kehitysprojektit. (Mt.)

Jyvsectecin tunnistaa kuviossa 2 nähtävästä Jyvsectecin logosta.

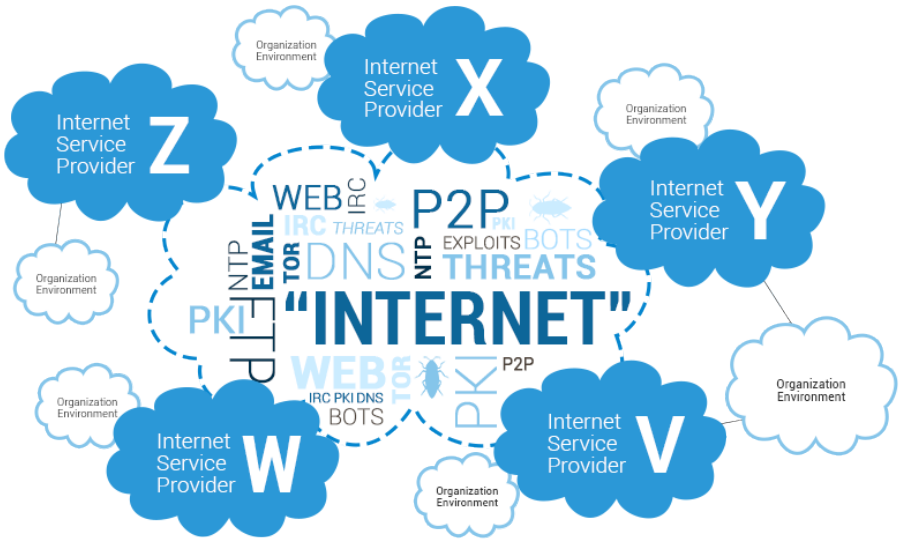
Kuvio 2. Jyvsectec logo (Jyvsectec 2014.)

1.3 RGCE

Jyvsectecin suunnittelema ja ylläpitämä kyberturvallisuuden kehitysympäristö eli RGCE (Realistic Global Cyber Environment) on suljettu tietoverkko, joka nimensä mukaan on ”Internet” -tasoinen verkko, jossa voidaan toteuttaa erilaisia verkon palveluita ja hyökkäyksiä eriytetyssä ympäristössä. RGCE:n päätoiminnallisuuksia ovat muun muassa:

- Vastaa oikeaa Internet-ympäristöä.
- Eriytetty tietoverkko mahdollistaa hyökkäysten, tunnettujen haavoittuvuuksien ja oikeiden haittaohjelmien käyttämisen.
- Verkko sisältää automaattisesti generoitua käyttäjätoiminnallisuutta ja näin vastaa aitoa verkkoliikennettä.
- Mahdollistaa realististen organisaatioympäristöjen rakentamisen. (RGCE 2014.)

Kuviossa 3 on esitetty RGCE-verkon looginen rakenne. Verkko sisältää erilaisia verkon peruspalveluja, kuten NTP (Network Time Protocol), DNS (Domain Name System), Web-palvelut, Email-palvelut yms.



Kuvio 3. RGCE-verkko (RGCE 2014.)

RGCE-verkko on suunniteltu vastaamaan oikeaa Internet ympäristöä, jossa on oikeita julkisia IP-osoitteita ja maantieteellisiä sijainteja sisältäen eritasoisia (Tier 1, Tier 2 ja Tier 3) operaattorireitityksiä sekä Internetin ydinpalveluita, kuten nimipalvelut ja sertifikaatit. Tämä mahdollistaa realistisen operaattoritasoisen organisaatioympäristön rakentamisen verkkoon. RGCE on eriytetty verkko, joka mahdollistaa riskittömän ympäristön hyökkäyksien ja haavoittuvuuksien testaamiselle ilman, että sillä vaarannetaan tai saastutetaan oikeita julkisia tuotantoympäristöä tai järjestelmiä. (RGCE 2014.)

1.4 Tavoitteet ja tehtävät

Tämän opinnäytetyön tavoitteena oli toteuttaa Jyvsectecin RGCE -tietoverkkoon verkon monitorointipalvelu Open Source -tuotteella. Monitoroitavia kohteita olivat RGCE-verkon tietyt palvelimet ja palvelut. Näitä olivat muun muassa Web-, DNS-,Email-, ja NTP-palvelimet. Työssä ei siis ollut tarkoituksena monitoroida verkon aktiivilaitteita vaan verkon palveluita. Työssä määritettiin myös valvottaville kohteille sovitut valvonta-aikavälit, hälytysraja-arvot yms. jotka sovittiin yhdessä toimeksiantajan kanssa. Monitorointipalvelin toteutettiin virtuaalisena palvelimena Jyvsectecin vCloud -järjestelmään.

Tehtävänä oli myös vertailla eri Open Source -monitorointituotteita ja toteuttaa näistä yksi, joka sopii parhaiten palveluiden valvontaan. Työkalu valittiin yhteistyössä Jyvsectecin kanssa. Kokonaisuudessaan opinnäytetyön tavoitteena oli oppia rakentamaan verkonmonitorointipalvelu verkkoon, ymmärtämään sen asettamat vaatimukset ja täydentää omaa tietotaitoa tietotekniikka-alalta.

2 Tietoverkkojen teoria

2.1 OSI-malli

Tärkein nykypäivänä käytettävä tietoverkkomalli on ISO organisaation (International Organization for Standardization) kehittämä Open Systems Interconnection eli OSI -referenssimalli. Tämä malli jakaa tietoverkon viestinnät eri tasoihin ja esittää, miten jokaista kerrosta käytetään kommunikointiprosessissa. Jokainen kerros lisää tietoa dataan lähetysprosessin yhteydessä ja samalla tavalla käyttäen ja poistaen tätä tietoa vastaanottoprosessissa. (Sosinsky 2009.)

OSI-malli määrittää seitsemän eri tasoa käyttäen numeroita 1-7 seuraavassa järjestyksessä: fyysinen, siirtoyhteys-, verkko-, kuljetus-, istunto-, esitystapa- ja sovelluskerros. Ensimmäiset neljä kerrosta ovat laitteistoon liittyviä ja kolme viimeistä ovat lähinnä ohjelmistoon. (Mt.)

OSI-malli määrittää taulukossa 1 näkyvät kerrokset.

Taulukko 1. OSI-malli (Sosinsky 2009)

Kerros	Tuettu liikenne tyyppi	Toiminto
7. Sovellus	Data	Sovelluskerros hallinnoi verkkoyhteyttä sovelluksen ja verkon välissä.
6. Esitystapa	Data	Esitystapakerroksessa tieto muotoillaan muotoon, joka voidaan käsitellä vastaanottavassa järjestelmässä.
5. Istunto	Data	Istuntokerros luo uniikin yhteyden lähettävän ja vastaanottavan järjestelmän välille ja varmistaa että tiedot on siirretty oikein.
4. Kuljetus	Segmentit tai datagrammit	Kuljetuskerros hallinnoi tiedon lähetyksen ja vastaanoton eheyden.
3. Verkko	Paketit	Verkkokerros kontrolloi osoitteita joita käytetään tiedonsiirrossa. Esimerkiksi IP-osoitteet.
2. Siirtoyhteys	Kehykset	Siirtoyhteys kerros hallinnoi laite osoitteita. Esimerkiksi MAC osoitteet.
1. Fyysinen	Bitit	Fyysinenkerros määrittää siirtomedian, esimerkiksi kupari tai valokuitu.

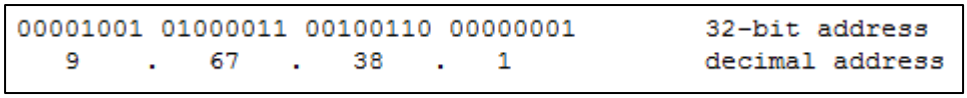
2.2 IP-osoitteet

IP eli Internet Protocol on suunniteltu käytettäväksi pakettikytkentäisissä tietoverkkojärjestelmissä. IP -protokolla tarjoaa tavan tietolohkojen siirtämiseen lähteestä kohteeseen, jossa lähde ja kohde on identifioitu IP-osoitteella. (RFC 791 1981.)

Koska IPv4 -osoitteet ovat loppumassa, kehitettiin IP -protokollalle laajennuksena IPv6-protokolla. Tässä käsitellään kuitenkin vain IPv4-protokollan mukaisia IP-osoitteita.

IP käyttää osoitteita ilmoittamaan lähteen ja kohteen tietoverkkojärjestelmissä. Jokainen osoite koostuu 32 bitistä, jotka on jaoteltu neljään pisteellä eroteltuun desimaalilukuun. Jokainen desimaalinumero edustaa 8-bittistä osaa osoitteesta. (Cepeda 2000.)

Kuviossa 4 on esitetty esimerkki IP-osoitteen rakenteesta.



Kuvio 4. IP-Osoite (Cepeda 2000)

Jokainen osoite voidaan jakaa kahteen eri loogiseen osaan:

- Verkko-osoite, joka kertoo, missä osiossa koko verkkoa järjestelmä sijaitsee.
- Päätelaiteosoite, jonka avulla tunnistetaan tietty järjestelmä tai laite verkon sisällä.

IP-osoitteita on määritetty yksityiseen käyttöön, joita ei mainosteta julkisessa verkossa (Internet). Näitä privaattialueita on A, B ja C.

- A luokka, johon kuuluvat IP-osoitteet alueelta 10.0.0.0 – 10.255.255.255 sisältäen 16 777 216 mahdollista osoitetta.
- B luokka, johon kuuluvat IP-osoitteet alueelta 172.16.0.0 – 172.31.255.255 sisältäen 1 048 576 mahdollista osoitetta.

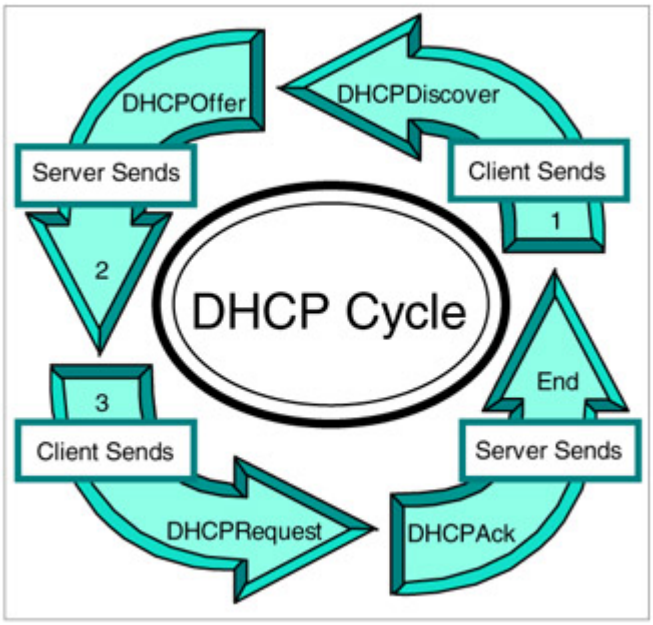
- C luokka, johon kuuluvat IP-osoitteet alueelta 192.168.0.0 – 192.168.255.255 sisältäen 65 536 mahdollista osoitetta.

(Cepeda 2000.)

2.3 DHCP

Tietoverkot ovat jatkuvasti muutoksen alla. Uusia laitteita liitetään ja vanhoja poistetaan, uusia toimipisteitä lisätään ja uusia työntekijöitä tulee lisää. Kaikkia näitä muutoksia tapahtuu joka päivä. Kaikkien muutoksien ylläpito voi olla suuri työ ilman järjestelmää, joka vastaa automaattisesti muutoksien vaatimuksiin. TCP/IP -tietoverkoissa jokaisella laitteella täytyy olla IP-osoite, verkkomaski ja yhdyskäytävän osoite kommunikoidakseen verkossa. Myös vähintään yksi nimipalvelin (DNS) on hyvä olla käytössä. IP-osoitteiden määrittäminen käsin on järkevää ainoastaan verkoissa, joissa verkko-osoitetta tarvitsevien laitteiden määrä jää pieneksi. DHCP palvelimen käyttö mahdollistaa TCP/IP-verkon luotettavamman, joustavamman ja ylläpidollisesti kevyemmän käytön. DHCP jakaa dynaamisesti IP-osoitteita laitteille määritetystä alueesta, eikä näin vaadi käyttäjän manuaalisesti määrittämään jokaiselle laitteelle erikseen IP-osoitetta. (Cepeda 2000.)

Kuviossa 5 on kuvattu DHCP-palvelun kautta tarjottava IP-osoitteen jakoprosessi. Prosessi alkaa, kun asiakas lähettää ”DHCPDiscover” -viestin. Tämän jälkeen DHCP-palvelin vastaa viestiin ”DHCPOffer” -viestillä, jonka jälkeen asiakas lähettää ”DHCPRequest” -pyynnön, johon palvelin vastaa myönnettäessä IP-osoitetta ”DHCPAck” –viestillä. Tässä vaiheessa prosessi on valmis ja asiakaslaitteelle on jaettu IP-osoite automaattisesti.



Kuvio 5. Automaattisen IP-osoitteen jakamisen prosessi (Cepeda 2000)

2.4 DNS

DNS eli Domain Name System on palvelu, joka muuttaa IP-osoitteet nimeksi. Kun IP-osoitteet on muutettu selkokielisempään muotoon, on palvelua helpompi ja käyttäjäystävällisempi käyttää. Esimerkiksi internet-sivusto google.com toimii IP-osoitteessa 216.58.209.132, mutta käyttäjälle se näkyy DNS-palvelun avulla nimellä google.com.

Internet ja tietoverkot toimivat siten, että kaikille päätelaitteille on asetettu IP-osoite (päätelaite, palvelin, reititin jne.). Ilman mahdollisuutta käyttää mitään palvelua selkokielisellä nimellä meidän täytyisi tietää jokaisen palvelun IP-osoite. Kun maailmassa on satoja miljoonia päätelaitteita ja Internetsivustoja, on mahdoton tehtävä ylläpitää kaikkia palveluiden osoitteita muistissa pelkän IP-osoitteen avulla. Tämän ongelman ratkaisuksi kehitettiin nimipalvelimia, jotka sijaitsevat tunnetussa paikassa, ja jotka ylläpitävät palveluiden IP-osoitteita esimerkiksi sivuston www.google.com löytämiseksi. (Aitchison 2011.)

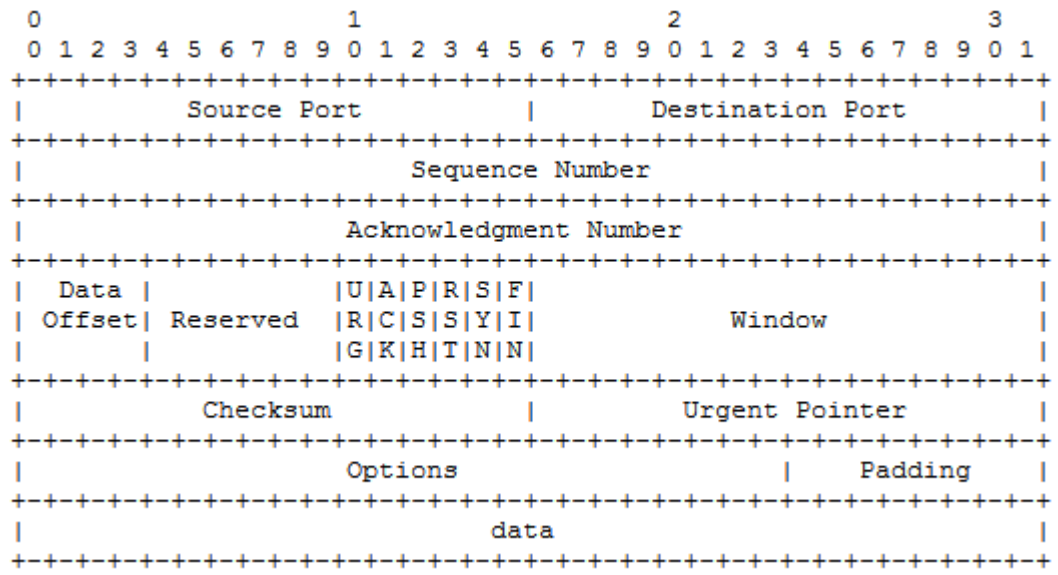
2.5 TCP

TCP eli Transmission Control Protocol on laajimmin käytetty kuljetusprotokolla tietoverkoissa nykypäivänä. TCP toimii OSI-mallin neljännessä-, eli kuljetuskerroksessa. TCP on Etelä-Kalifornian yliopiston määrittelemä standardi RFC 793. TCP tarjoaa valvontamekanismeja, jotka hallitsevat tietoa, jota viesti sisältää. Näin TCP varmistaa, että tieto lähetetään hallittavissa olevina segmentteinä. TCP varmistaa myös, että tieto saapuu yhtenäisenä ja että uudelleen koottu tieto on eheä kopio tiedosta, joka on lähetetty. TCP sisältää joukon ohjauskäskyjä, jotka voivat muuttaa siirrettyä tietomäärää yksittäisissä paketeissa ja lisäksi tahdin, jolla paketteja lähetetään. Sovelluksia, jotka käyttävät TCP:tä ovat selaimet, Web-palvelimet, sähköpostiohjelmat sekä tiedonsiirtosovellukset. (Sosinsky 2009.)

TCP kehitettiin ratkaisemaan luotettavan viestinnän ongelman epäluotettavissa verkoissa. Kun tietoa pitää jaksottaa IP-paketeiksi ja siirtää niitä joukkona IP-pyyntöjä, täytyy käytössä olla mekanismi, jolla hallita IP-tietovirtaa. TCP mahdollistaa sen, että sovellus antaa yksittäisen tietokäskyn, jonka jälkeen TCP hoitaa tiedonsiirron yksityiskohdat. (Mt.)

TCP-segmentit lähetetään Internet-datagrammeina. IP-otsikko kuljettaa monta tietokenttää sisältäen lähde- ja kohdeisännän osoitteet. TCP-otsikko seuraa IP-otsikkoa toimittamalla tietoa liittyen TCP-protokollaan. Tämä jakaminen sallii muidenkin isäntätason protokollien olemassaolon kun TCP. (RFC 793 1981.)

Kuviossa 6 on kuvattu TCP-protokollan otsikkomuoto.



Kuvio 6. TCP otsikkomuoto (RFC 793 1981)

Tärkeimmät kentät TCP-viestissä on seuraavat:

Source port:

16 bittiä. Sisältää lähdeportin numeron.

Destination port:

16 bittiä. Sisältää kohdeportin numeron.

Sequence Number:

32 bittiä. Sisältää sekvenssinumeron ensimmäisestä dataoktetista tässä segmentissä.

Acknowledgement Number:

32 bittiä. Jos Ack-hallintabitti on määritetty, tähän kenttään tulee seuraavan sekvenssin numero, jota segmentin lähettäjä odottaa vastaanottavansa. Kun yhteys on luotu, tämä lähetetään aina.

Data Offset:

4 bittiä. Numeromäärä 32-bittisiä sanoja TCP -otsikossa. Tämä kertoo, mistä varsinainen hyötydata alkaa. TCP -otsikko on 32 bittiä pitkä integraalinumero.

Reserved:

6 bittiä. Varattu tulevaisuuden varalle. Täytyy olla nolla.

Control bits:

6 bittiä.

URG: “Urgent Pointer field significant” Kiireellinen pointteri.

ACK: “Acknowledgment field significant” Kuittaus.

PSH: “Push Function” Push toiminto.

RST: “Reset the connection” Uudelleen luo yhteyden.

SYN: ”Synchronize sequence numbers” Sykronoi sekvenssinumerot

FIN: “No more data from sender” Lopettaa tiedonsiirron.

Window:

16 bittiä. Tieto-oktettien määrä alkaen siitä, joka esitettiin ”Acknowledgement Number” kentässä, jonka tämän segmentin lähettäjä on valmis hyväksymään.

Checksum:

16 bittiä. Checksum kenttä on tiedon eheyden varmistamiskenttä.

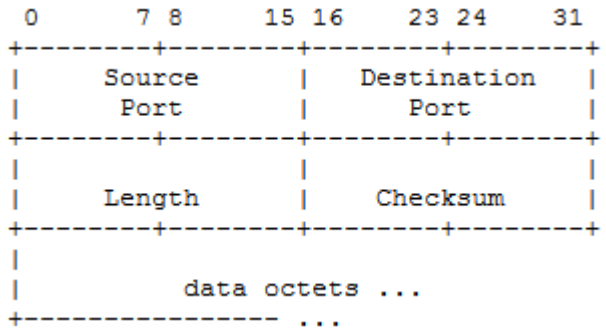
2.6 UDP

UDP eli User Datagram Protocol on kuljetuskerroksen protokolla, joka on määritetty RFC 768 -standardissa. UDP luo tilattoman yhteyden kahden isäntälaitteen välille IP-verkossa. UDP luo lyhyen tiedonsiirtomuodon, jota kutsutaan datagrammiksi ja yhteyttä kutsutaan ”datagram socket” -nimellä. Virtuaalinen yhteys, joka luodaan käyttäen samaa mallia portista erityyppisten tiedon lähetykseen isäntälaitteiden välillä. Tilaton viittaa siirtomekanismiin, joka ei yritä varmistaa lähetettyjen tietojen oikeellisuutta. Vastaanottava järjestelmä uudelleenrakentaa tiedon vastaanotetuista datagrammeista huolimatta siitä, ovatko ne oikeassa järjestyksessä tai onko niiden järjestys täydellinen. (Sosinsky 2009.)

Koska UDP ei ylläpidä otsikkotietoja TCP-protokollan tavoin, tarkoittaa tämä, että UDP siirtää tietoa paljon nopeammin kuin TCP. Tämä tekee UDP:stä paremman valinnan, kun luotettava tiedonsiirto ei ole tarpeen, esimerkiksi kun viesti on lyhyt tai viesti sisältää paljon tarpeettomia tai valinnaisia tietoja. Esimerkiksi nimipalvelut käyttävät UDP:ta,

koska niiden viestit ovat lyhyitä. Ääni, musiikki ja video sovellukset käyttävät UDP:ta, koska jos kehys putoaa elokuvasta tai lyhyt katkos ilmenee VOIP puhelussa, käyttökokemus ei kärsi niin paljon kuin esimerkiksi tiedonsiirrossa. Lähes kaikki suoratoistomediasovellukset käyttävät UDP:ta niiden kuljetusprotokollana IP-verkoissa. (Mt.)

Kuviossa 7 on kuvattu UDP:n otsikkomuoto.



Kuvio 7. UDP-protokollan otsikkomuoto (Postel 1980)

Kentät UDP otsikossa:

Source port, lähdeportti:

Valinnainen kenttä. Tarvittaessa kertoo lähettäjän portin, ja näin voidaan olettaa portti, johon vastata, jos muu porttitieto puuttuu.

Destination port, kohdeportti:

Esittää kohdeportin. Portilla on merkitys erityisesti IP-kohdeosoitteeseen.

Length, pituus:

Esittää tämän datagrammin pituuden oktetteina sisältäen tämän otsikon ja datan.

Tarkistussumma, checksum:

16-bittinen tarkistussumma, joka varmistaa tiedon eheyden.

3 Verkon monitorointi

3.1 Yleistä

Verkon monitoroinnilla tarkoitetaan järjestelmää, joka monitoroi tietoverkossa olevia palvelimia ja palveluita jatkuvasti tunnistaen hitaat tai kokonaan sammuneet palvelut. Monitoroinnilla valvotaan myös verkon eri komponenttien käyttöä ja suorituskykyä ilmoittaen niistä verkon ylläpitäjille erilaisilla hälytysmenetelmillä, kuten esimerkiksi sähköpostilla. Tietoverkon monitorointi on erittäin tärkeässä roolissa tietoverkoissa, jotta verkon ylläpitäjät huomaisivat verkossa tapahtuvat ruuhkatilanteet, hidastumiset ja palveluiden kaatumiset. (Downin 2013.)

Tietoverkon monitorointi on kriittinen IT-toiminto, joka voi säästää rahaa tietoverkon suorituskyvyssä, työntekijän tuottavuudessa ja infrastruktuurin kustannuksien ylityksessä. Tietoverkon monitorointipalvelu monitoroi verkkoa sen ongelmilta, ja ennalta ehkäisevästi tunnistaa erilaisia ongelmia kuten vähäisen levytilan, haittaohjelmat ja vanhenevan laitteiston. Monitorointi voi näin vähentää näiden ongelmien vaikutusta yrityksen tuottavuuteen. (Mt.)

Verkon monitoroinnin etuja:

- **Luotettavuus:** Verkon monitorointi seuraa kriittisiä palveluita ja ilmoittaa verkon ylläpitäjille, ennen kuin ongelma rupeaa vaikuttamaan loppukäyttäjiin. Esimerkiksi monitorointiohjelmisto voi ilmoittaa jos palvelin kaatuu, jos palvelu lakkaa vastaamasta tai jos levytila on loppumassa. Tämä varmistaa ennakoivan lähestymistavan ongelmiin verrattuna siihen, että loppukäyttäjät ilmoittaisivat ongelmasta vasta, kun se on jo tapahtunut.

- **Tieto verkosta:** Verkon monitorointipalvelu varoittaa verkon ylläpitäjiä suorituskykyongelmista tai häiriötilanteista lähettämällä erilaisia hälytyksiä tietokoneille tai puhelimiin. Tämä mahdollistaa IT-järjestelmänvalvojia olemaan aina tietoisia ongelmista riippumatta siitä, missä he sijaitsevat.

- **Kapasiteetti:** Kun tietää ja ymmärtää verkon laitteiden kapasiteetin ja käytön, voi ennalta tunnistaa kohteita, jotka tarvitsee lisää levytilaa ja lisätä sitä hallitusti.

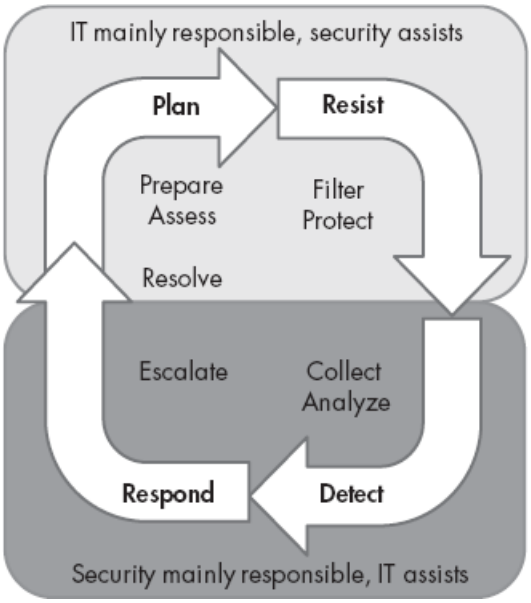
- **Vianmääritys:** Verkon monitoroinnin avulla voi nopeasti tunnistaa laitteen, joka aiheuttaa ongelman täten lyhentäen aikaa, joka kuluisi muuten vianmääritykseen ja menisi hukkaan.

- **Trendien seuraaminen:** Ongelmat, jotka ilmenevät ajoittain tai tiettyinä ruuhka-aikoina voi olla vaikeasti havaittavissa, mutta jatkuva verkonvalvonta raportoi verkossa tapahtuvista trendeistä ja ruuhkatilanteista, jotta ylläpitäjät ymmärtävät verkon keskeiset trendit ja niiden vaikutuksen suorituskykyyn.

- **Päivitysten ja muutosten suunnittelu:** Jos tietty laite käy jatkuvasti suorituskyvyltään maksimissaan, tulee asiaan puuttua. Verkon monitoroinnilla voidaan seurata laitteiden suorituskykyä ja näin suunnitella etukäteen laitteisiin tulevat muutokset.

- **Näytä muille mitä tapahtuu:** Raportit ja statistiikat verkon terveydestä ja aktiivisuudesta ovat hyviä työkaluja osoittamaan noudatetaanko tiettyä palvelutasosopimusta tai kertomaan miksi tietty laite täytyy korjata tai vaihtaa. (Mt.)

Verkonhallinnassa voidaan käyttää hyvänä apuvälineenä kuviossa 8 näkyvää prosessikuvaa. Kuviossa 8 esitetään, että verkonhallinnassa on tärkeää pitää prosessi jatkuvana. Suunnitellaan, kehitetään vastustus, huomioidaan ja vastataan. Suunnitteluvaiheessa valmistellaan ja ratkaistaan ongelmat, vastustusvaiheessa suodatetaan ja suojataan tietoa, tunnistusvaiheessa keretään tietoa ja analysoidaan sitä ja vastausvaiheessa toteutetaan vastustus, jonka jälkeen prosessi toistaa itseään kokoajan kehittyvänä.



Kuvio 8. Verkon monitoroinnin prosessikuvaus (Bejtlich 2013)

Verkonvalvonnasta on olemassa kaksi pääasiallista lähestymistapaa, joita käytetään yleisesti monitoroidakseen verkon suorituskykyä:

- Passiivinen monitorointi: Passiivisella verkonmonitoroinnilla tarkoitetaan sitä, että verkon laitteet tallentavat tietoa verkon liikenteestä, jotka antavat viitteitä tietystä verkkolaitteesta. Ajastettua kyselyä käytetään tyypillisesti noutamaan tämä tieto raportointiin ja analysointiin. Passiivinen verkonmonitorointi ei vaadi mitään lisäliikennettä verkkoon, jotta sitä voidaan käyttää mittaus-tarkoituksissa. Tämä nostaa kuitenkin kysymyksen kuinka useasti tätä tietoa tulisi noutaa? (Farrel, A 2009)

- Aktiivinen monitorointi: Toisin kuin passiivinen monitorointi; aktiivinen monitorointi sisältää lisäliikennettä verkkoon. Synteettiset testi virrat, jotka käsittävät ”koetin” -paketit lähetetään verkossa yksinomaan luonnehtimaan verkon suorituskykyä; vastaanotettujen virtojen analyysia käytetään tämän kuvaamiseen. Ideaalitilanteessa olisi parasta mitata verkossa tapahtuvaa viivettä, viiveenvaihtelua ja suorituskykyä kun se tapahtuu, mutta tämä ei yleensä ole mahdollista käytännössä. (Mt.)

Verkonvalvonnassa puhutaan yleisesti myös proaktiivisesta ja reaktiivisesta monitoroinnista. Vaikka monitoroinnin toteutus millä vain tavalla on parempi kuin ei monitorointia lainkaan, monet yritykset epäonnistuvat implementoimaan ratkaisun, joka on proaktiivinen. Proaktiivisella monitoroinnilla tarkoitetaan järjestelmää, joka hälyttää ylläpitäjiä ennen kuin järjestelmässä tapahtuu vikatilanne. Tämä on ideaaliratkaisu verkonmonitorointiin, jolla pyritään estämään verkossa tapahtuvat vikatilanteet. Reaktiivisella monitoroinnilla taas tarkoitetaan hälytyksiä, jotka aktivoituvat vasta, kun vikatilanne on tapahtunut. Esimerkkinä tästä voi olla, että järjestelmä kaatuu ja vasta sen jälkeen ylläpitäjiä hälytetään. Hälytys voi tulla jopa vasta puhelinsoittona loppukäyttäjältä tietohallintoon. (Dragich 2012.)

3.2 Teoriaa SNMP-protokollasta

SNMP (Simple Network Management Protocol) on IETF:n vuonna 1988 kehittämä protokolla, jonka avullahallitaan Internetissä ja muissa kytketyissä tietoverkoissa olevia elementtejä. SNMP on johdettu sen edeltäjistään SGMP (Simple Gateway Management Protocol) ja se oli tarkoitus korvata objekti-orientoituneella CMIP-protokollalla. Koska CMIP oli kuitenkin monimutkainen ja sen kehityksessä kesti pitkään, monet laitevalmistajat olivat jo ottaneet SNMP:n käyttöön eivätkä halunneet enää siirtyä siitä CMIP:iin, joka johti laajaan SNMP:n käyttöön. SNMP on vieläkin kaikkein yleisin viestintäalan hallintaprotokolla. (Sathyan 2010.)

SNMP on sovellustason hallintaprotokolla, joka käyttää UDP:tä OSI-mallin kuljetuskerroksessa. Sen toiminta perustuu manager/agentti-malliin sisältäen managerin, jolla on hallintatoiminnot sekä agentin, joka on verkkoelementissä tehtävänään lähettää ja vastaanottaa viestejä managerilta sekä tallentaa niitä hallintatietokantaan. SNMP-protokollassa manageri ja agentti käyttävät hallintatietokantaa (MIB, Management Information Base) ja suhteellisen pientä joukkoa komentoja, joilla vaihtaa tietoja. MIB on kokoelma tiedoista, jotka on järjestelty hierarkisesti puuksi. Se käyttää numeerista merkkiä tai OID-tunnistetta (Object Identifier) erottaakseen kunkin muuttujan MIB-tietokannassa ja SNMP-viesteissä. (Mt.)

Taulukossa 2 on esitelty eri SNMP-viestit:

Taulukko 2. SNMP -viestit

Viesti	Selitys
Get [GetRequest]	Tämä on pyyntö managerilta agentille, jotta saadaan arvo tietystä OID:ista.
GetNext [GetNextRequest]	Tämä on pyyntö managerilta agentille, jotta saadaan arvo seuraavasta MIB instanssista. GetNext-viestiä käytetään siirtymiseen MIB-puussa.
Response [GetResponse]	Agentti vastaa tiedolla managerille komentoihin Get, GetNext tai Set käyttäen "response" -komentoa.
Set [SetRequest]	Tämä on pyyntö managerilta agentille asettaa arvo tiettyyn MIB-instanssiin.
Trap	Tämä on asynkrooninen viesti agentilta managerille ilmoittaakseen tietyn tapahtuman esiintymisen.

SNMP:n edut:

- Helppokäyttöinen agentin ja managerin implementaatio vain muutamalla komennolla.
- Helppoa lisätä ohjelman muuttujat monitorointiin.
- Laajennettavuus pienellä vaivalla tulevaisuuden tarpeisiin yksinkertaisuuden vuoksi.
- SNMP on laajalti käytössä ja monet laitevalmistajat tukevat sitä.
- Yksinkertainen ottaa käyttöön ja käyttää. (Mt.)

SNMP:n heikkoudet:

- Ensimmäisissä SNMP versioissa oli tietoturva-aukkoja ja SNMP-protokollan toinen versio korjaa joitain ongelmia liittyen tiedon yksityisyyteen, autentikointiin ja kulunvalvontaan.
- Tietoa ei voida esittää yksityiskohtaisemmin ja organisoidusti, joka täyttäisi seuraavan sukupolven tietoverkkojen vaatimukset SNMP:n yksinkertaisen suunnittelun vuoksi. (Mt.)

3.3 SNMPv2

Suurena heikkoutena SNMP:n ensimmäisessä versiossa oli tietoturvan puute. Päätavoite SNMPv2:ssa oli voittaa tämä turvallisuuskysymys. SNMPv1 ja SNMPv3 ovat IETF:n standardoimia, mutta SNMPv2 ei ole täydellisesti standardoitu kenenkään toimesta ja siitä on saatavilla vain luonnosversioita. Vaikka SNMPv2:ssa oli parannettu turvallisuutta verrattuna SNMPv1:een, vasta SNMPv3 standardoitiin, ja se tarjoaa kaupallisen tason hallinnan ja turvallisuuden kehysmallin. (Sathyan 2010.)

Edut SNMPv2:ssa verrattuna SNMPv1:een ovat:

- Parempi tapahtumienvalvonta käyttäen ”inform”-viestiä. SNMPv1 -agentin generoima ”trap” -viesti yksinkertaisesti kerättiin managerin päässä, kun taas SNMPv2:ssa ”inform”-viesti agentilta on hyväksyttävä vastauksella managerilta. ”Inform”-tapahtumaviesti lähetetään uudelleen, jos manager ei vastaa viestiin.
- Parannettu virheenkäsittely. Uusia virhe- ja poikkeusmenetelmiä.
- Laajennetut tietotyypit, kuten BITS, uSigned32, Counter64 ja monia muita.
- Standardoitu multiprotokollatuki.
- Parannettu turvallisuus.
- Suurempi tiedonhaku mahdollista ”GetBulk”-komennon käyttämisen ansiosta, johtaen parempaan tehokkuuteen ja suorituskykyyn. (Mt.)

Komennot ”Get”, ”GetNext” ja ”Set”, jotka ovat käytössä SNMP:n ensimmäisessä versiossa ovat myös käytössä SNMPv2:ssa ja ne toimivat samalla tavalla. (Sathyan, J 2010).

SNMPv2 määrittää kaksi uuttaa protokolla komentoa:

- GetBulk: Tätä komentoa käytetään noudettaessa isoa määrää tietoa.
- Inform: Tätä komentoa voidaan verrata ”trap”-komentoon, mutta vahvistettuna ”trap”-viestinä. (Mt.)

3.4 SNMPv3

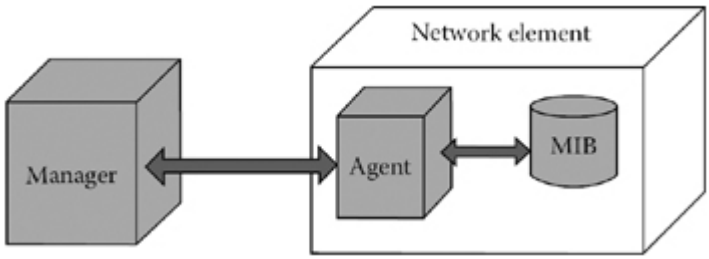
SNMPv3 on määritetty useassa eri standardissa. Näitä ovat muun muassa RFC 3411, RFC 3412, RFC 3413 ja RFC 3414. SNMPv3 lisää tietoturvallisuutta ja etähallinta- mahdollisuuksia edellisiin versioihin verrattuna. SNMPv3-arkkitehtuuriin esiteltiin käyttäjäpohjainen turvallisuusmalli viestien turvallisuudelle ja näkymään perustuva kulunvalvontamalli. SNMPv3 tukee myös erilaisten turvallisuus-, kulunvalvonta-, ja viestien prosessointimallien samanaikaista käyttöä. SNMPv3:ssa on mahdollisuus dynaamisesti määrittää SNMP-agentti käyttämään SNMP:n ”Get”-komentoa MIB-objektiin vastaamaan agentin määrityksiä. SNMPv3 tarjoaa kolmetasoisen turvallisuusmallin, jossa korkein taso tarjoaa todennuksen ja yksityisyyden, toinen taso todennuksen ilman yksityisyyttä ja alimmalla tasolla ei käytetä kumpaakaan. (Kundu, Lavlu 2009.)

3.5 MIB-tietokanta

MIB eli Management Information Base on kokoelma hallittavista objekteista. Tämä on virtuaalinen tietokanta, joka sisältää muuttujia, jotka vastaavat hallittavia resursseja. MIB-tietokantaa valvotaan päivittämällä tietoa MIB-tauluun. Verkonvalvoja eli ”manager” pyytää tietoa MIB-taulusta agentin kautta. Agentti hakee hallintatiedon ja vastaa managerille. Verkonvalvoja voi käyttää MIB-taulukkoa monitoroidakseen verkkoa ja se voi myös päivittää sitä. (Sathyan 2010.)

Varsinainen MIB-määritelmä on vain tiedosto, joka antaa tietoa tietyssä muodossa ja MIB-instanssi sisältää varsinaiset muuttujat, jotka liittyvät hallittuun objektiin. MIB ilmentyy agentissa, mutta muuttujien varsinaiset kopiot sijaitsevat verkkoelementissä, jossa agentti toimii. Manageri yleensä pitää kuitenkin paikallista kopiota MIB-muuttujista helpon käytettävyyden, suorituskyvyn sekä tiedon tulkinnan helpottamisen vuoksi. Sopiva synkronointimekanismi on yleensä tarpeen varmistaakseen, että managerilla oleva kopio on päivitetty ajoittain vastaamaan todellista tietoa käyttäen agenttia. (Mt.)

Kuviossa 9 on kuvattu kuinka manageri ja agentti hakevat ja käyttävät tietoa verkkoelementin MIB-taulusta.

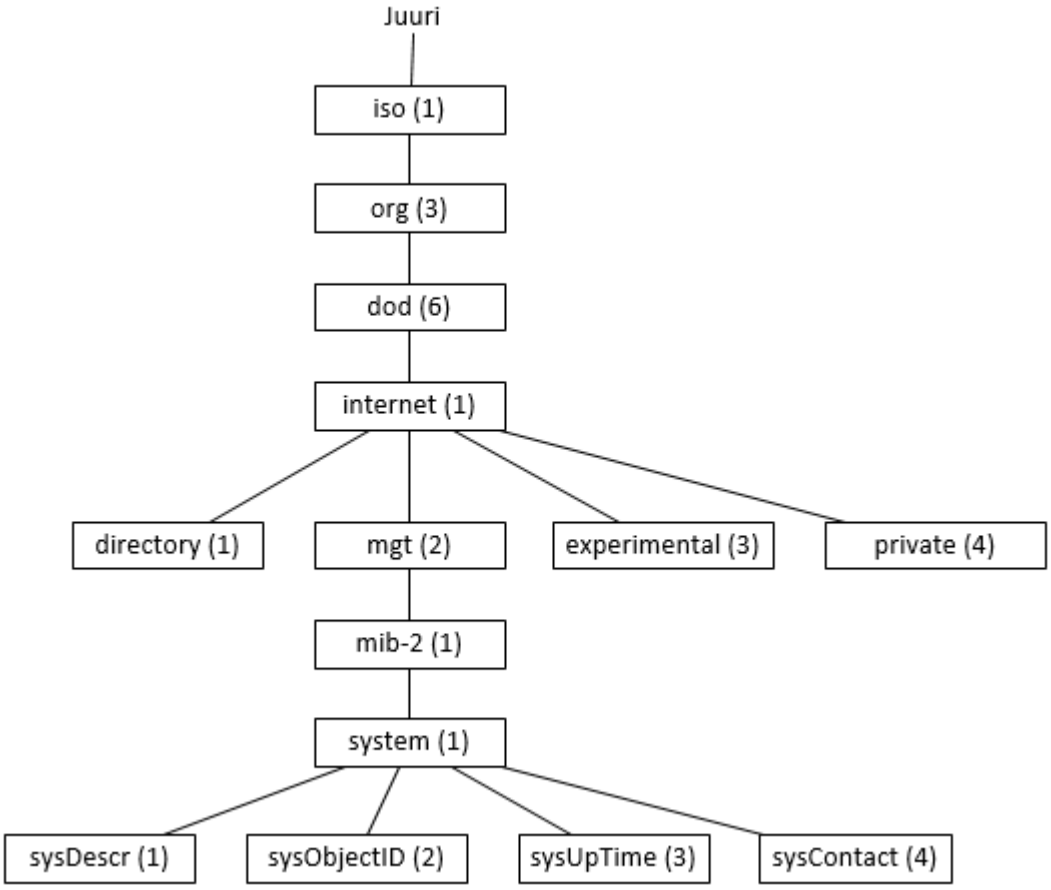


Kuvio 9. Manager- ja MIB-komponentin välinen vuorovaikutus (Mt.)

3.6 OID

Kaksi ratkaisevaa SNMP:n mallia ovat OID:it (Object Identifier) ja MIB-taulut. SNMP toimii kyselemällä Objekteja. Objekti on yksinkertaisesti jotain, mitä voidaan kerätä tietona verkkolaitteesta. Esimerkiksi objekti voi olla tietyn rajapinnan tila. Kysyttäessä rajapinnan tilaa vastaus palauttaa muuttujan; rajapinta voi olla ylhäällä tai alhaalla. SNMP tunnistaa objektit OID:ien avulla. (Leskiw 2015.)

OID:it on jäsennelty erittäin selkeästi ja ne on rakennettu hierarkkiseen puumuotoon samankaltaisesti kuin kansiorakenne tietokoneella. Toisin kuin kansiot, kaikki SNMP-objektit on kuitenkin numeroitu. Kuten kuviossa 10 esitetystä mallista voidaan nähdä, ylin kohta ”iso” on merkitty numerolla ”1”. Seuraava taso ”org” on puolestaan merkitty numerolla ”3”, koska se on kolmas objekti ISO:n alla. OID:it on aina kirjoitettu numeeriseen muotoon, joten ensimmäiset kolme tasoa tässä esimerkissä on kirjoitettu ”1.3.1” ei ”iso\org\dod”. (Mt.)



Kuvio 10. Esimerkki OID-puusta

Edellä olevassa kuviossa 10 voidaan nyt hakea esimerkiksi kenttä ”sysUpTime” OID:illä ”.1.3.6.1.2.1.1.3.0”. Seuraavassa kuviossa 11 tämä haku on tehty onnistuneesti.

```
-bash-4.1# snmpget -v 2c -c opennms localhost .1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (120374964) 13 days, 22:22:29.6
4
```

Kuvio 11. SNMP OID "sysUpTime"

Vastaavalla tavalla voitaisiin noutaa myös esimerkiksi “sysContact”-kenttä, jonka OID on ”.1.3.6.1.2.1.1.4.0”. Tähän kenttään määritettiin SNMP-toteutuskohdassa arvo ”Sakari Remonen G2590”.

4 Tuotteet

4.1 Vaatimukset

4.1.1 Toimeksiantajan vaatimukset

Toimeksiantajan Jyvsectec vaatimuksena oli, että työkalu, jolla verkonmonitorointi toteutetaan, pitää soveltua erityisesti:

- Verkon eri palveluiden ja palvelimien valvontaan, kuten esimerkiksi aikapalvelut (NTP), nimipalvelut (DNS) ja Web-palvelut. Työssä ei ole siis tarkoitus valvoa verkon aktiivilaitteita.

- Työkalun tulee olla Open Source -tuote tai muuten täysin ilmainen ilman määrällisiä tai ajallisia rajoitteita.

- Ohjelmistolla voidaan kerätä monitoroitavia kohteita ryhmiin, jotta niiden selaaminen on helppoa ja jotta verkosta saadaan heti selkeä kokonaiskuva.

- Hälytyksiä tulee voida ilmaista eri tasoisina riippuen ongelman vakavuudesta. Hälytyksistä pitää tulla selkeästi esille, onko jokin verkon palvelu kokonaan pois päältä vai hälytetäänkö vain pienestä ongelmasta.

- Muokattava käyttäjän rajapinta, jotta verkonvalvojat näkevät heti, jos verkossa esiintyy hälytyksiä.

- Valvottavia kohteita on helppo lisätä.

Kun tuotteita on vertailtu ja karsittu pois, asennetaan niistä kolme, jotta voidaan käytännössä vertailla eri tuotteiden eroa. Näistä kuitenkin vertailun jälkeen toteutetaan lopulta vain yksi. Palvelimet asennetaan CentOS Linux -alustoille Jyvsectecin vCloud -järjestelmään virtuaalipalvelimina. Monitorointi voi tapahtua agenteilla tai esimerkiksi

SNMP- protokollaa käyttäen. Monitorointipalvelin asennetaan verkkoon siten, että sieltä on pääsy verkon eri palveluihin ilman, että verkon rakennetta tarvitsee muuttaa.

Verkon palveluiden monitoroinnissa ei riitä, että esimerkiksi DNS palvelinta monitoroidaan vain ICMP echo -kyselyllä, joka kertoo onko palvelin pystyssä, mutta ei kerro varsinaista DNS-palvelun tilaa. On myös tärkeää saada esimerkiksi Web palvelimista tieto, millä viiveellä sivustot vastaavat.

4.1.2 RGCE:n valvottavat palvelut

Kun verkonmonitorointityökalu on asennettu, antaa toimeksiantaja tarkemman listan valvottavista kohteista.

Taulukossa 3 esitellään RGCE verkossa valvottavat palvelut.

Taulukko 3. Valvottavat palvelut

Nimi	Tyyppi	IP-osoite	Valvottavat palvelut	Tekniikka
NTP_B	NTP (Ntpd)	80.12.71.2	NTP, ICMP	SNMP
ns1.netfun.fi	Nimipalvelin (Bind)	192.49.144.4	DNS, ICMP, SNMP, SSH	Agent,SNMP
www.reuters.com	Verkkosivu (Wordpress)	144.243.214.28	HTTP, ICMP, SNMP, SSH	Agent,SNMP
mail.netfun.fi	Sähköposti (Postfix+Dovecot)	192.49.144.7	HTTP, HTTPS, ICMP, IMAP, POP3, SMTP, SNMP, SSH	Agent,SNMP

4.2 Open Source -tuotteet

Open Source verkonmonitorointituotteita löytyi hyvin monta, mutta koska rajoituksena oli täysi ilmaisuus ja tuotteen tarkoituksena on monitoroida vain verkon palveluita, jäi vertailun jälkeen työkaluista vain muutama varteenotettava vaihtoehto jäljelle. Soveltuvimmat kolme tuotetta olivat OpenNMS, Zabbix ja Nagios core. Cacti pääsi myös hyvin

lähelle parhaimmistoa, mutta se keskittyi enemmän erilaisten grafiikoiden piirtämiseen verkosta, mikä ei tässä työssä ollut niin tärkeää.

OpenNMS on Java-pohjainen täysin ilmainen GPL-linsenssillä toimiva verkon monitorointiohjelmisto. OpenNMS oli ensimmäinen avoimen lähdekoodin verkonmonitorointiohjelmisto ja sen kehitystyö aloitettiin jo vuonna 1999. Tuotteen Java-pohjaisuus arvelutti Javan tunnettujen tietoturvaongelmien vuoksi, mutta eduksi OpenNMS:ssä voidaan nähdä käytettävyys ja selkeä palveluiden kategorisointi. Myös palveluiden ja palvelimien käytettävyydet näkyvät heti selkeästi OpenNMS:n etusivulla, joka oli tärkeää tässä työssä. Agentiton toimintatapa on tuettu, mutta myös OpenNMS perustuu hyvin pitkälle manager-agentti-toimintatapaan. Työkalussa on myös valmiiksi sisäänrakennettu SNMP-tuki. Web käyttöliittymällä saa OpenNMS:ään täyden hallinnan.

Zabbix on täysin ilmainen, GPL-lisenssillä toimiva yritystasoinen verkon monitorointiohjelmisto, joka on saatavilla Red Hat, CentOs, Debian ja Ubuntu Linux -alustoille 32- ja 64-bittisenä versiona. Zabbix on kirjoitettu C-kielellä. Zabbix:in toiminta perustuu hyvin pitkälle manager-agentti-periaatteeseen. Valvottaville palvelimille on siis asennettava Zabbix-agentti. Zabbix kuitenkin tukee myös agentitonta monitorointia, ja sisältää ”Auto-discovery”-ominaisuuden. Työkalussa on myös valmiiksi sisäänrakennettu SNMP-tuki. Valintaan vaikuttivat positiivisesti Zabbix:en muokattava näkymä, valvottavien kohteiden ryhmittely ja hälytyksien eri tasojen helppo määritys. Zabbix:ia hallitaan täysin Web-käyttöliittymällä. Myös aikaisempi henkilökohtainen kokemus puolsi valintaa.

Nagios Core on täysin ilmainen GPL-lisenssillä toimivan monitorointiohjelmisto, joka on kirjoitettu C-kielellä. Nagioksesta on olemassa myös maksullinen kaupallinen versio, mutta se karsiutui maksullisuutensa vuoksi heti pois. Nagios Core:lle on saatavilla monta erilaista käyttäjärajapintaa ”front-end” ja monia liitännäisiä ”plug-in”, joilla voidaan laajentaa sen toimintaa.

Tämän työn toteutukseen valittiin yhdessä toimeksiantajan kanssa OpenNMS sen parhaimman soveltuvuuden vuoksi. OpenNMS on hyvin automatisoitu järjestelmä, jolla voidaan helposti monitoroida verkon palveluita. OpenNMS on käyttövalmis paketti valvo-

maan verkon eri palveluita ilman, että agentteja tarvitsee verkkoon asentaa. OpenNMS ei myöskään tarvitse paljon eri lisäosia toimiakseen, vaan se on käyttövalmis heti perus-paketin asennuksen jälkeen. Eri tuotteita testattiin käytännössä ennen lopullisen valinnan tekemistä.

Taulukossa 4 on esitelty vertailtujen tuotteiden eroja, sekä niiden soveltuvuus verkon-monitorointiin.

Taulukko 4. Open-Source tuotteiden vertailu

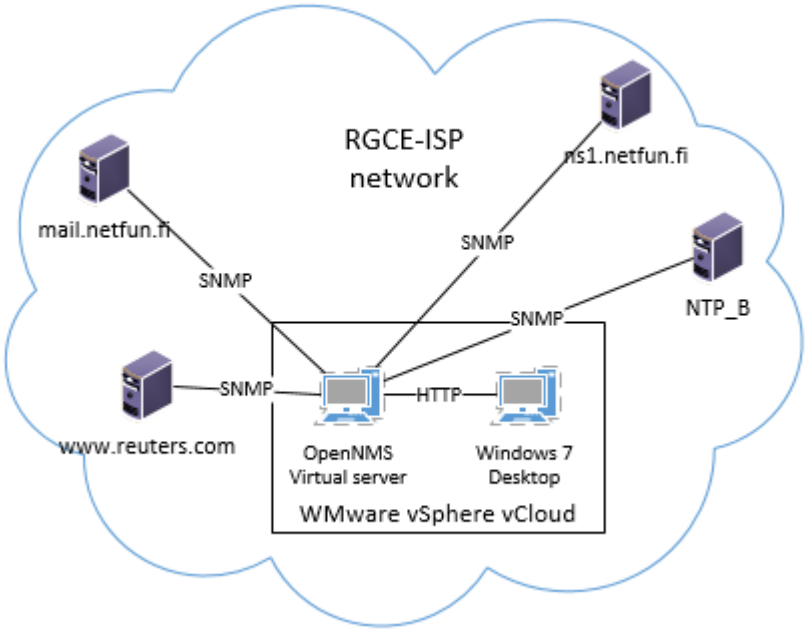
Ohjelmisto	Lisenssi	Agentiton	Web Käyttö-liittymä	Auto Discove-ry	SNMP	Soveltuvuus
OpenNMS	GPL	Kyllä	Täysi kontrolli	Kyllä	Kyllä	Palveluiden valvonta. All-in-one solution "out-of-the-box"
Zabbix	GPL	Tuettu	Täysi kontrolli	Kyllä	Kyllä	Palvelimien ja palvelui-den valvonta. Perustuen agentin käyttöön.
Nagios core	GPL	Tuettu	Asennettavissa	Asennettavissa	Asennettavissa	Laajan tietoverkon val-vonta. Muokattava, pal-jon eri lisäosia.
Cacti	GPL	Kyllä	Täysi kontrolli	Asennettavissa	Kyllä	Verkon palveluiden ja laitteiden käytöstä grafii-koita.

5 Toteutus

5.1 Ympäristön esittely

Työ toteutettiin Jyvsectecin vCloud -virtuaaliympäristöön, joka on rakennettu käyttäen WMware vSphere -ohjelmistoa. Palvelin asennettiin virtuaalisena palvelimena RGCE-ISP -verkkoon. RGCE-ISP-verkko on rakennettu vastaamaan oikeaa palveluntarjoajan verkkoa, jossa on käytössä paljon erilaisia palveluita, kuten esimerkiksi nimi-, aika- ja sähköpostipalvelut. Palvelimen käyttöjärjestelmänä käytetään Linux CentOS 6.6 -jakelupaketin 64-bittistä versiota, jossa ei ole ollenkaan graafista käyttöliittymää, joten sen toiminnot tehtiin komentoriviltä. OpenNMS-palvelinta voitiin hallita myös virtuaaliselta Windows 7 -työpöydältä selaimen kautta.

Kuviossa 12 on kuvattu monitorointipalvelin osana RGCE-ISP-verkkoa.



Kuvio 12. RGCE-ISP looginen

Monitorointipalvelin kerää verkon palveluista tietoa agentteja ja SNMP protokollaa käyttäen.

OpenNMS palvelin sai DHCP palvelulta IP-osoitteen 194.85.166.147, joten sitä voitiin hallita Windows 7 virtuaalikoneella selaimen kautta menemällä osoitteeseen 194.85.166.147:8980/opennms/.

5.2 OpenNMS

5.2.1 Asentaminen

OpenNMS:stä päätettiin asentaa versio 15.0.1-1, koska se oli uusin OpenNMS-versio CentOS 6.x -käyttöjärjestelmälle. OpenNMS-ohjelmiston asentaminen aloitettiin asentamalla “opennms-repo” .rpm paketti. Tämä paketti sisälsi tiedon, jonka YUM-asentaja tarvitsee, jotta se voi asentaa kaikki tarvittavat OpenNMS-paketin osat.

Asennus aloitettiin komennolla:

```
# rpm -Uvh http://yum.opennms.org/repofiles/opennms-repo-stable-
rhel6.noarch.rpm
```

Edellä olevassa komennossa käytetyt parametrit määritellään seuraavasti:

- ”rpm” on RPM muotoisten pakettien manageri.
- -U tarkoittaa ”upgrade”, eli päivittää/lisätä.
- -v tarkoittaa ”verify”, eli tarkistaa, että paketti on eheä.
- -h tarkoittaa ”hash”, joka kirjoittaa 50 kappaletta risuaitamerkkiä kun RPM paketti on purettu asentamisen selkeyttämiseksi

OpenNMS tarvitsee toimiakseen SQL-tietokantaohjelmiston. Seuraavaksi asennettiin PostgreSQL tietokantaohjelmisto seuraavalla komennolla:

```
# yum install postgresql postgresql-server
```

Seuraavaksi käynnistettiin “postgresql”-palvelu. Tämä tapahtui komennoilla:

```
# /sbin/service postgresql initdb
```

```
# /sbin/service postgresql start
```

Edellä olevissa komennoissa ensimmäinen komento alustaa tietokannan, ja toinen komento käynnistää itse “postgresql” -palvelun.

Varmistettiin, että tietokantapalvelu käynnistyy automaattisesti aina käynnistyksen yhteydessä. Tämä tehtiin tekemällä muutos ”chkconfig”-tiedostoon seuraavalla komennolla:

```
# /sbin/chkconfig postgresql on
```

Oletuksena, PostgreSQL sallii yhdistämisen sen tietokantaan vain, jos on kirjauduttu paikallisena käyttäjänä, joka vastaa PostgreSQL-käyttäjää. Koska OpenNMS toimii pääkäyttäjällä ”root”, se ei voi yhdistää tunnuksella ”postgres” tai ”opennms” oletuksena, joten se täytyi muuttaa asetuksiin, jotta se sallitaan. Tämä tehtiin muuttamalla tietokannan ”pg_hba.conf”-tiedoston asetuksia seuraavalla tavalla.

Oletuksena pg_hba.conf tiedoston sisältö on seuraavanlainen:

```
local   all         all                               ident sameuser
host    all         all         127.0.0.1/32          ident sameuser
host    all         all         ::1/128               ident sameuser
```

Se muokattiin korvaamalla ”ident”- ja ”sameuser” –määritykset ”trust” -määrityksellä, jonka jälkeen tiedoston rakenne oli seuraavanlainen:

```
local   all         all                               trust
host    all         all         127.0.0.1/32          trust
host    all         all         ::1/128               trust
```

Tämän jälkeen palvelu piti vielä käynnistää uudelleen jotta muutokset tulevat voimaan. Tämä tehtiin komennolla:

```
# /sbin/service postgresql restart
```

Nyt kaikilla paikallisilla käyttäjillä on oikeus tietokantaan. Tämä ei ole hyvä tietoturvan kannalta, koska jokaisella, jolla on tunnus paikallisesti palvelimeen, on myös täysi oikeus tietokantaan.

Seuraavaksi OpenNMS tarvitsee toimiakseen Javan kehitys kirjaston. OpenNMS tukee vain JDK 7 versiota. Uusin JDK versio oli 7u75, joka ladattiin seuraavalla komennolla:

```
# wget --no-cookies --no-check-certificate --header "Cookie:
gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-
securebackup-cookie" "http://download.oracle.com/otn-
pub/java/jdk/7u75-b13/jdk-7u75-linux-x64.rpm"
```

Tämän jälkeen paketti asennettiin komennolla:

```
# rpm -ivh jdk-7u45-linux-i586.rpm
```

Edellä olevassa komennossa parametri “-i” tarkoittaa “install”, eli RPM paketin asentamista.

Kun kaikki OpenNMS:n vaatimat paketit oli asennettu, voitiin aloittaa itse OpenNMS:n asentaminen. OpenNMS-ohjelmisto ei ole yksi paketti, vaan yhdistelmä useasta komponentista. CentOS:sin YUM-pakettimanageri latasi ja asensi kaikki nämä komponentit ja niiden riippuvuudet automaattisesti, ellei niitä jo ollut asennettu palvelimelle.

OpenNMS pakettien asentaminen tehtiin komennolla:

```
# yum -y install opennms
```

Edellä olevassa komennossa parametri ”-y” tarkoittaa, että YUM vastaa jokaiseen mahdolliseen kysymykseen ”yes”. Tämä nopeuttaa asentamista.

Seuraavaksi kerrottiin OpenNMS:lle mitä Java-asennusta haluttiin käyttää. Tämä tehtiin seuraavalla komennolla:

```
# /opt/opennms/bin/runjava -S /usr/java/latest/bin/java
```

Seuraavaksi luotiin tietokanta OpenNMS:lle. Tämä tehtiin komennolla:

```
# /opt/opennms/bin/install -dis
```

Edellä olevassa komennossa parametrit määritetään seuraavasti:

- -d päivittää tietokannan
- -i lisää kaiken oletustiedon, joka kuuluu tietokantaan

- -s luo tallennetut menettelyt, jota OpenNMS käyttää tietynlaisen tiedon saatavuuteen

Tämän jälkeen OpenNMS antoi virheilmoitusta GPG avaimesta (GNU Privacy Guard). Tämä korjattiin tuomalla GPG-avain uusiksi järjestelmään. Tämä tehtiin seuraavalla komennollla:

```
#rpm --importhttp://yum.opennms.org/OPENNMS-GPG-KEY
```

Tässä vaiheessa OpenNMS ja kaikki sen vaatimat paketit oli asennettu. Seuraavaksi poistettiin vielä käytöstä ”iptables” palvelu, joka toimii CentOS:ssisa palomuurina. Tämä tehtiin siksi, ettei palomuuri estäisi palvelun toimimista testiympäristössä. Tämä tehtiin seuraavilla komennoilla:

```
# /etc/init.d/iptables save
# /etc/init.d/iptables stop
# chkconfig iptables off
# /etc/init.d/ip6tables stop
# chkconfig ip6tables off
```

Nyt OpenNMS palvelu voidaan käynnistää komennolla:

```
# /sbin/service opennms start
```

Tämän jälkeen OpenNMS palveluun pääsee käsiksi selaimella menemällä osoitteeseen http://194.85.166.147:8980/opennms/

Oletus käyttäjätunnuksena oli ”admin” ja salasanana ”admin”. Salasana vaihdettiin heti OpenNMS:n graafisen käyttöliittymän kautta arvoon ”root66”.

Ensimmäisen käynnistyksen yhteydessä OpenNMS antoi virheilmoituksen:

```
# Starting OpenNMS: Started OpenNMS, but it stopped running: for
details see /opt/opennms/logs/daemon/output.log
```

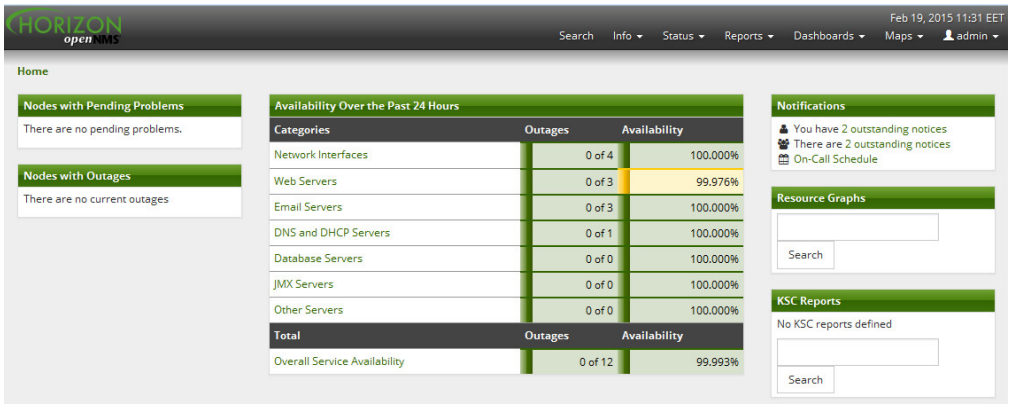
Tiedostossa output.log oli seuraava virheilmoitus:

```
Exception thrown by the agent : java.net.MalformedURLException: Lo-
cal host name unknown: java.net.UnknownHostException:
opennms1.localdomain: opennms1.localdomain
```

Tämä johtui siitä, että koneen nimi vaihdettiin heti alussa ”localhost”:ista ”opennms1”:ksi, jonka seurauksena OpenNMS ei osannut tehdä nimiselvitystä ”opennms1” kohteelle. Muutos piti tehdä /etc/hosts -tiedostoon, jonne lisättiin localhost -kohtaan ”opennms1”. Tämän jälkeen OpenNMS käynnistyi virheettömästi.

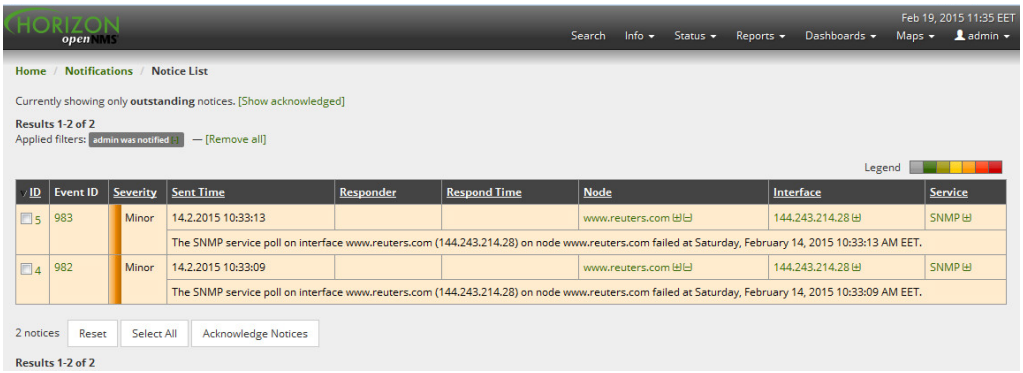
5.2.2 Ympäristö ja näkymät

Kun OpenNMS:sään on kirjauduttu pääkäyttäjänä, on näkymä kuten kuviossa 13. Tässä näkymässä nähdään heti palveluiden tila ja kokonaiskäytettävyyden prosentti. Jos jokin palvelu on kokonaan pois päältä, näkyisi se punaisena kenttänä ”Outages”-kohdassa. Vasemmalla reunalla nähdään päällä olevat hälytykset, joita kuviossa 13 ei ole yhtään. Oikeassa reunassa taas nähdään ilmoitukset, joita tässä esimerkissä on kaksi kappaletta. Ilmoituksia klikkaamalla pääsee tutkimaan tarkemmin valitun ilmoituksen tietoja.



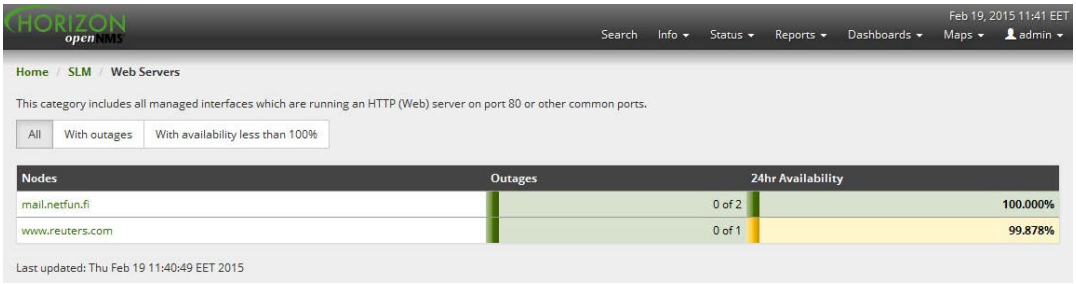
Kuvio 13. OpenNMS admin -etusivu

Kuviossa 14 on avattu ilmoitukset näkymä klikkaamalla ilmoitusta etusivulla. Näkymässä nähdään ilmoituksen vakavuus (minor, major, outage) ja myös värikoodattu vakavuustaso (punainen vakavin). Ilmoitukset täytyy ruksittaa ja huomioida, jotta ne poistuvat etusivun näkymästä. Tämä tapahtuu laittamalla ruksi ilmoituksen eteen ja klikkaamalla ”acknowledge notices”.



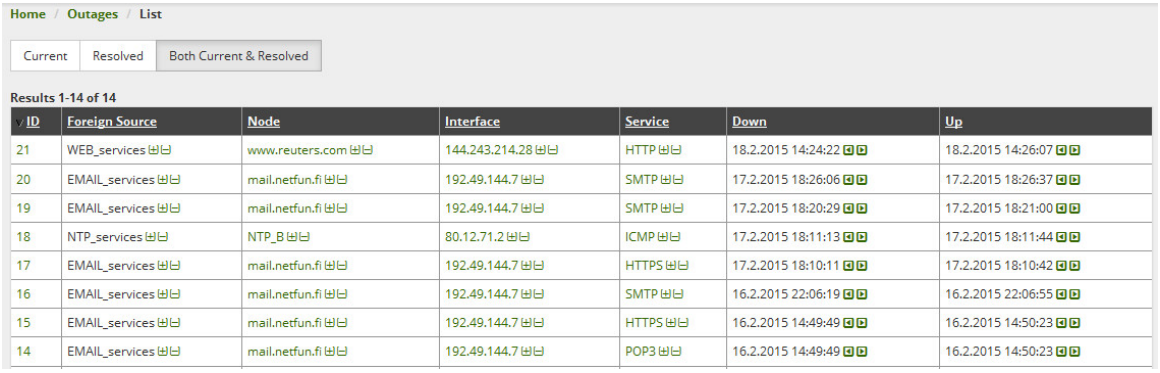
Kuvio 14. OpenNMS ilmoitukset

Kun kuviossa 13 näkyvältä etusivulta klikkaa jotain palvelu tyyppiä, esimerkiksi “Web servers”, aukeaa kuviossa 15 oleva näkymä, joka listaa kaikki siihen kategoriaan sopivat palvelimet, ja kertoo niiden yksittäisten palveluiden tilan ja kokonaiskäytettävyys prosentin.



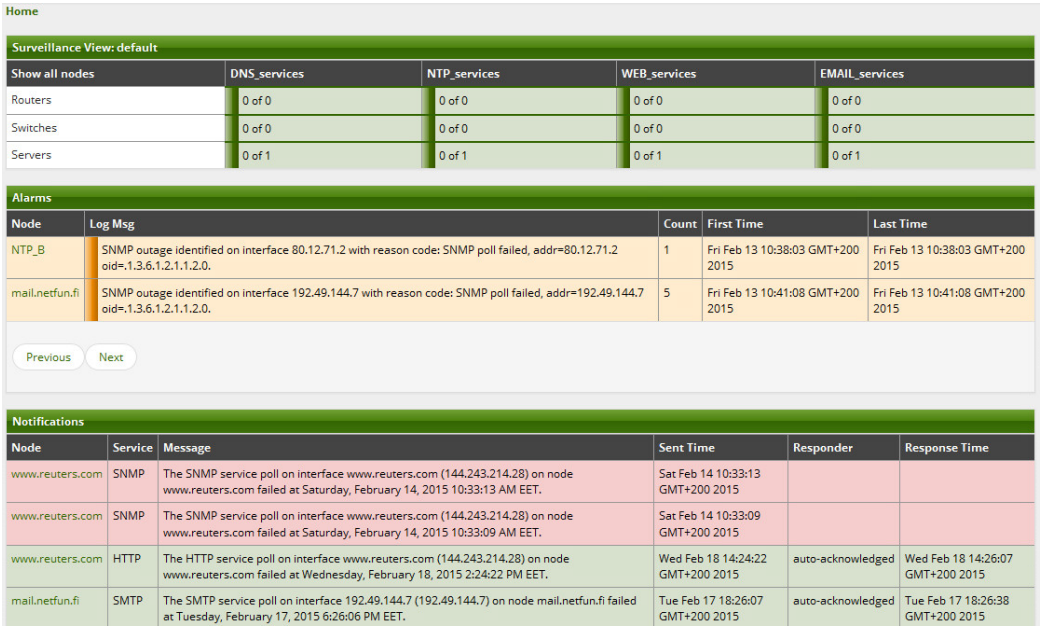
Kuvio 15. OpenNMS Web Servers

Klikkaamalla “Outages”-kohtaa ja sen jälkeen “Both Current & Resolved” päästään kuviossa 16 esitettyyn näkymään, joka kertoo järjestelmien kaatumiset ja niiden palautumisien ajankohdat.



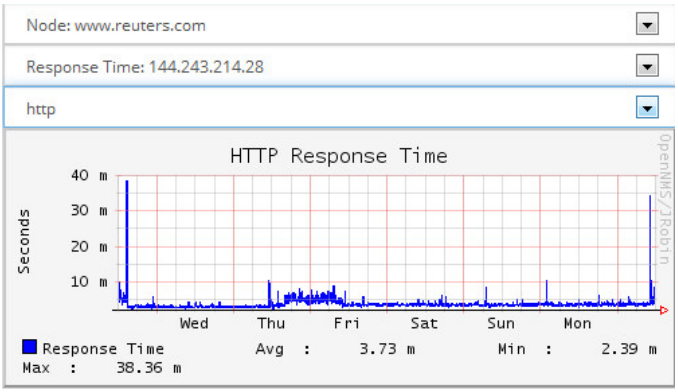
Kuvio 16. OpenNMS outages näkymä

Paras ja kätevin näkymä on “Dashboard”, joka kertoo kaikki hälytykset, huomautukset, katkokset ja viiveet palveluissa yhdellä sivulla. Kuviossa 17 on kuvattu OpenNMS:n ”Dashboard”-sivu. Jotta laitteet näkyvät Dashboard-sivulla, tulee niiden kuulua ainakin kahteen valvontakategoriaan, esimerkiksi ”Servers” ja ”DNS_Services”. Tämän sivun ulkoasua voi muokata tekemällä muutokset tiedostoon ”surveillance-views.xml” OpenNMS-palvelimella, jossa määritetään muun muassa sivulla näkyvät sarakkeet.



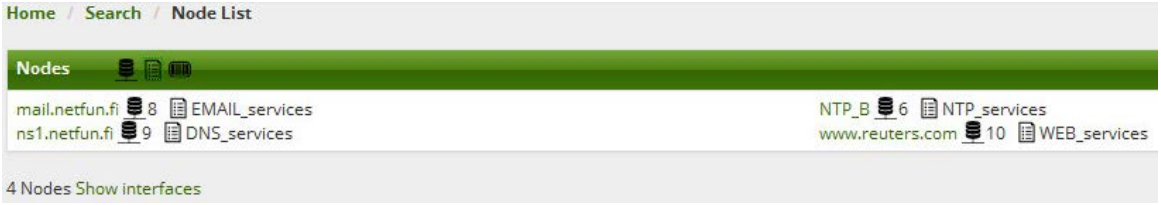
Kuvio 17. OpenNMS Dashboard näkymä

Dashboardilla on alareunassa myös pika linkki “resource graphs”-näkymään, joka kertoo erilaisina grafiikkoina vasteaikoja, prosessorikuormaa, ym. tietoa palveluista. Seuraavassa kuviossa 18 on esimerkkinä palvelimen ”www.reuters.com”-palvelun HTTP-vasteaika. Tämä ei ole ICMP:n viive, vaan ”HTTP Get” komennolla saatu vastausviive, joka kertoo sivuston realistisen HTTP-vastausajan.



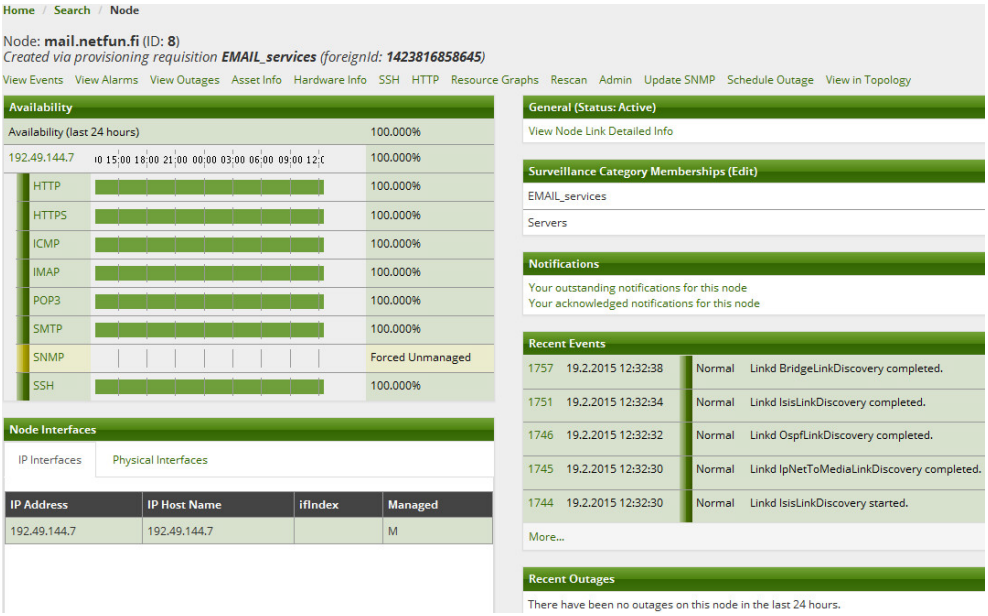
Kuvio 18. OpenNMS HTTP response time

Node List -näkymässä nähdään kaikki laitteet, joita on lisätty OpenNMS:sään. Laitteiden nimien perässä nähdään myös ryhmä, johon kyseinen laite kuuluu. Seuraavassa kuviossa 19 näkyy neljä laitetta, jotka on lisätty järjestelmään.



Kuvio 19. OpenNMS Node List

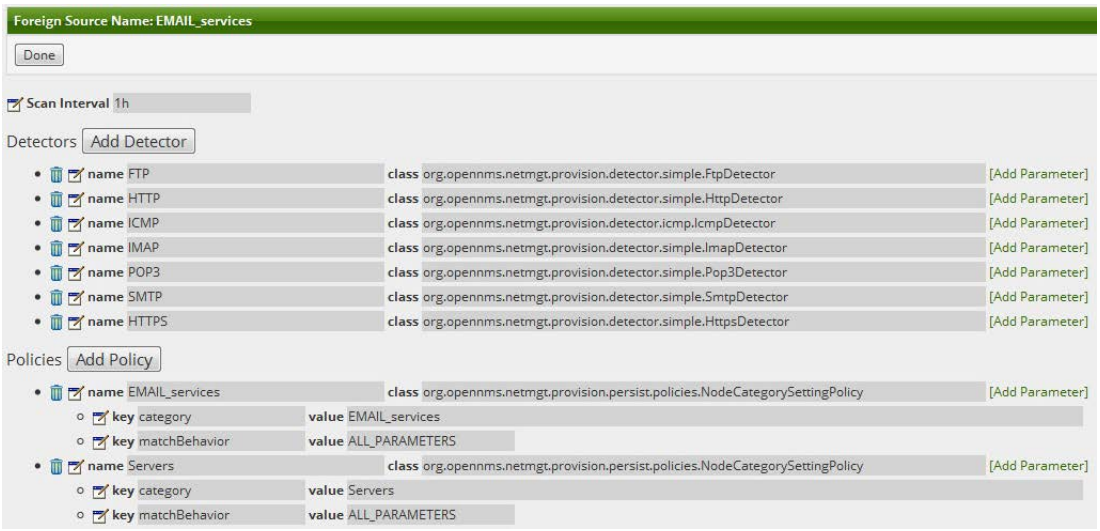
Klikkaamalla jotain laitetta, saadaan laitteesta ja sen monitoroiduista palveluista lisää ja tarkempaa tietoa. Seuraavassa kuviossa 20 on avattu mail.netfun.fi -sähköpostipalvelin. Palvelimen palveluista saadaan vielä yksityiskohtaisempaa tietoa klikkaamalla niitä, tai yläreunassa olevia linkkejä. Näkymän oikeassa reunassa näkyy myös, mihin valvontakategorioihin palvelin kuuluu. Tässä esimerkissä mail.netfun.fi -palvelin kuuluu ”EMAIL_services”- ja ”Servers”-valvontakategorioihin.



Kuvio 20. OpenNMS node

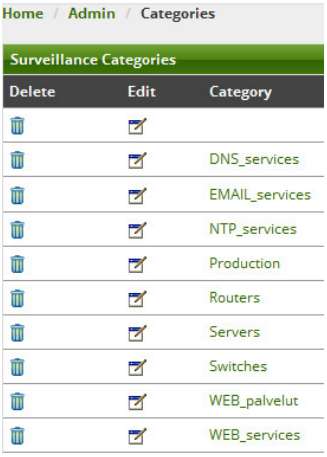
5.2.3 Valvontakategoriat

Jotta OpenNMS on kätevää käyttää, kannattaa sinne luoda erilaisia valvontakategorioita eli ”Surveillance Categories” ja ryhmäsäännöksiä eli “Provisioning Requisitions”. Nämä ryhmät määrittävät mitä palveluita skannataan laitteista, jotka kuuluvat kyseiseen ryhmään. Esimerkkinä seuraavassa kuviossa 21 on ”EMAIL_services” ryhmäsäännös, johon on määritetty valvottaviksi palveluiksi FTP, HTTP, ICMP, IMAP, POP3, SMTP ja HTTPS. Valvonnan aikaväliksi on määritetty tässä yksi tunti. Tämä aikaväli määrittää sen, kuinka usein uusia palveluita laitteista etsitään, ei jo löytyneiden palveluiden viiveaikoja. Kuviossa 21 nähdään myös, että kyseinen ryhmäsäännös on lisätty myös valvontakategorioihin ”Servers” ja ”EMAIL_services”.



Kuvio 21. OpenNMS EMAIL_services

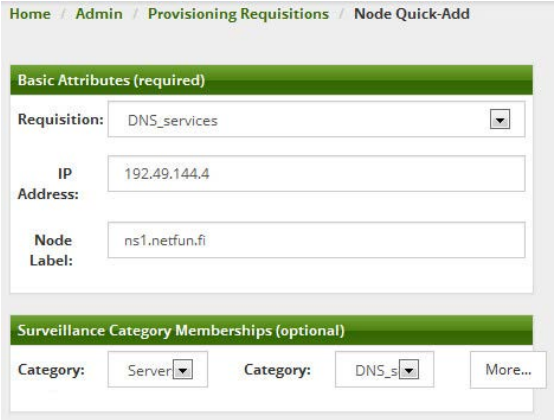
Valvontakategoriat on esitelty kuviossa 22. Näiden lisääminen on ehdotonta, jotta edellä luodut valvontaryhmät voidaan linkittää niihin, ja jotta uudet laitteet voidaan lisätä näihin ryhmiin kätevästi.



Kuvio 22. OpenNMS-valvontakategoriat

5.2.4 Monitoroitavien kohteiden lisääminen ja ryhmittely

Valvottavien laitteiden lisääminen on helppoa, kun valvontakategoriat ja ryhmät on luotu. Laitteiden lisääminen onnistuu ”Node Quick-Add”-sivulta, johon tarvitaan vain laitteen IP-osoite ja nimi, sekä mihin ryhmiin kohde halutaan lisätä. Kuviossa 23 on esimerkki ns1.netfun.fi -palvelimen lisäämisestä.

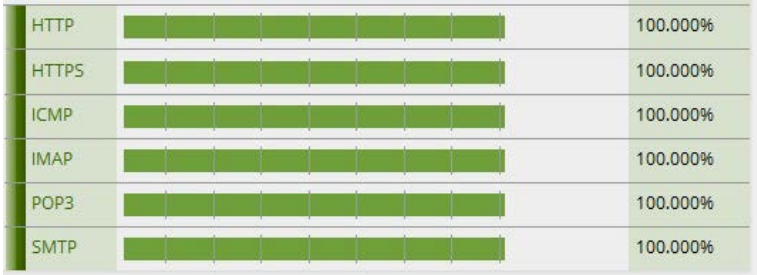


Kuvio 23. OpenNMS Node Quick-Add

Tämän jälkeen laitteesta skannataan kaikki palvelut, jotka on määritetty siihen kuuluvaan ryhmään. Nyt laite on lisätty onnistuneesti monitoroitavaksi OpenNMS:sään.

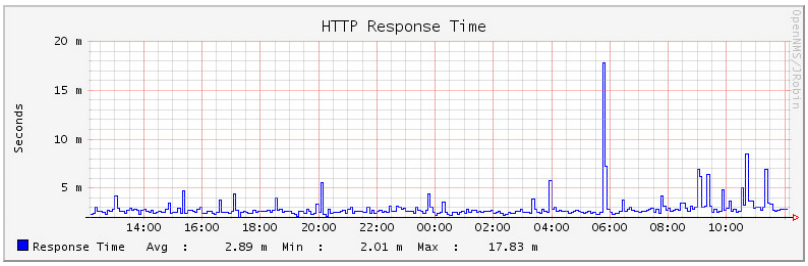
5.2.5 Palveluiden monitorointi

OpenNMS monitoroi verkon eri palveluita ja tallentaa niiden vasteajat ”.jrb”-muotoisiksi tiedostoiksi, joista OpenNMS osaa näyttää käyttäjälle palveluiden vasteajat grafiikkana. Laitteen monitoroidut palvelut nähdään laitteen sivustolta vihreänä merkittyinä. Kuviossa 24 nähdään mail.netfun.fi -laitteen monitoroitavat palvelut.



Kuvio 24. OpenNMS mail.netfun.fi –laitteen monitoroitavat palvelut

OpenNMS tallentaa kaikista monitoroitavista palveluista vasteajat automaattisesti kansioon /var/opennms/rrd/response/. Grafiikat vasteajoista Web käyttöliittymästä löytyy kohdasta ”Resource Graphs – Response Time”. Esimerkkinä tästä on monitoroitu palvelimella mail.netfun.fi toimivaa HTTP-palvelua, jonka vasteajat nähdään kuviossa 25.



Kuvio 25. OpenNMS HTTP-vasteaika

Seuraavaksi haluttiin todentaa mail.netfun.fi -palvelimelta, että OpenNMS tekee oikeaa HTTP-GET-kyselyä saadakseen palvelusta luotettavan vasteajan. Tämä tapahtui avaamalla ”HTTP access log” komennolla:

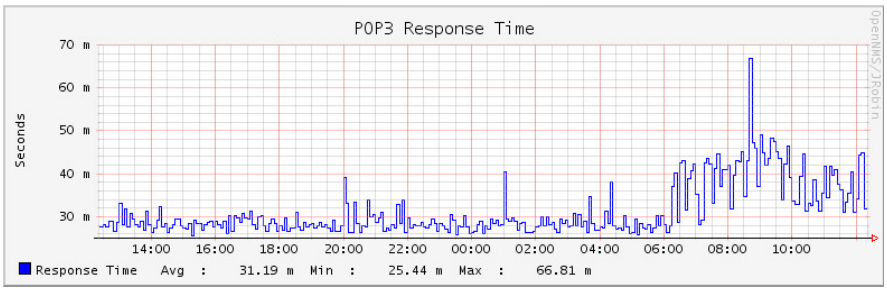
```
# tail /var/log/httpd/access_log
```

Joka tulosti seuraavaa:

```
194.85.166.147 - - [12/Mar/2015:06:40:58 -0400] "GET / HTTP/1.1"
200 38888 "-" "OpenNMS HttpMonitor"
194.85.166.147 - - [12/Mar/2015:06:45:58 -0400] "GET / HTTP/1.1"
200 38888 "-" "OpenNMS HttpMonitor"
194.85.166.147 - - [12/Mar/2015:06:50:58 -0400] "GET / HTTP/1.1"
200 38888 "-" "OpenNMS HttpMonitor"
194.85.166.147 - - [12/Mar/2015:06:55:58 -0400] "GET / HTTP/1.1"
200 38888 "-" "OpenNMS HttpMonitor"
194.85.166.147 - - [12/Mar/2015:07:00:58 -0400] "GET / HTTP/1.1"
200 38888 "-" "OpenNMS HttpMonitor"
194.85.166.147 - - [12/Mar/2015:07:05:58 -0400] "GET / HTTP/1.1"
200 38888 "-" "OpenNMS HttpMonitor"
194.85.166.147 - - [12/Mar/2015:07:10:58 -0400] "GET / HTTP/1.1"
200 38888 "-" "OpenNMS HttpMonitor"
194.85.166.147 - - [12/Mar/2015:07:15:58 -0400] "GET / HTTP/1.1"
200 38888 "-" "OpenNMS HttpMonitor"
194.85.166.147 - - [12/Mar/2015:07:20:58 -0400] "GET / HTTP/1.1"
200 38888 "-" "OpenNMS HttpMonitor"
```

Edellä olevasta tulosteesta nähdään, että OpenNMS:n IP-osoitteesta 194.85.166.147 tulee palvelimelle mail.netfun.fi HTTP-GET kyselyä joka viiden minuutin välein, eli tämä vasteaika kertoo sivuston realistisen viiveen, eikä pelkästään ICMP-kyselyllä saatua viivettä. Näin HTTP vasteaikagrafiikka on luotettavaa tietoa.

Toinen esimerkki on kuviossa 26 kuvattu mail.netfun.fi sähköpostipalvelimella toimivan POP3 -palvelun vasteaika.



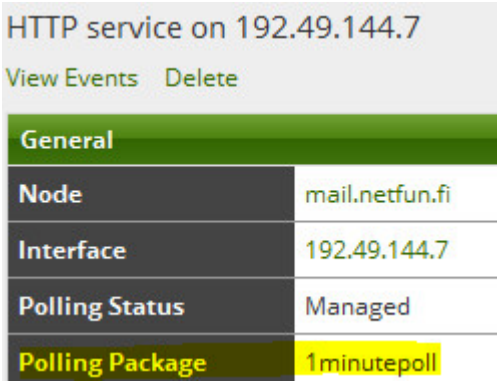
Kuvio 26. OpenNMS POP3 vasteaika

Kohdassa 5.2.8 on käsitelty hälytyksien luonti, kun palvelun vasteaika ylittää halutun raja-arvon.

OpenNMS tekee oletusarvoisesti kiertokyselyitä palveluihin viiden minuutin välein. Tätä asetusta haluttiin muuttaa tietyille laitteille ja halutuille palveluille. OpenNMS:n kiertokyselyiden määritykset on tehty tiedostoon ”/opt/opennms/etc/poller-configuration.xml”, jonne lisättiin uusi kiertokyselypaketti, joka määrittää laitteen ja siihen kohdistuvat palveluiden kiertokyselyjen väliajan. Esimerkkinä haluttiin toteuttaa mail.netfun.fi -palvelimelle HTTP-palvelu monitorointiin joka minuutti oletuksen joka viiden minuutti sijaan. Tämä toteutettiin lisäämällä tiedostoon seuraavat rivit:

```
<package name="1minutepoll" remote="true">
  <filter>IPADDR != '0.0.0.0'</filter>
  <include-range begin="192.49.144.7" end="192.49.144.7"/>
  <rrd step = "300">
    <rra>RRA:AVERAGE:0.5:1:2016</rra>
    <rra>RRA:AVERAGE:0.5:12:1488</rra>
    <rra>RRA:AVERAGE:0.5:288:366</rra>
    <rra>RRA:MIN:0.5:288:366</rra>
    <rra>RRA:MAX:0.5:288:366</rra>
  </rrd>
  <service name="HTTP" interval="60000" user-defined="true" sta-
tus="on">
    <parameter key="retry" value="1"/>
    <parameter key="timeout" value="3000"/>
    <parameter key="port" value="80"/>
    <parameter key="url" value="/"/>
    <parameter key="rrd-repository" val-
ue="/var/log/opennms/rrd/response"/>
    <parameter key="ds-name" value="http"/>
    <parameter key="thresholding-enabled" value="true"/>
  </service>
  <downtime interval="30000" begin="0" end="300000"/>
  <downtime interval="300000" begin="300000" end="43200000"/>
  <downtime interval="600000" begin="43200000" end="432000000"/>
  <downtime begin="432000000" delete="true"/>
</package>
```

Edellä olevilla komennoilla määritettiin uusi kiertokysely paketti, joka koskee kohdetta 192.49.144.7, jonka palvelua HTTP monitoroidaan joka 60000ms, eli joka minuutti. Nyt OpenNMS näyttää laitteen mail.netfun.fi HTTP-palvelun kohdassa ”Polling Package” olevan juuri määritetty ”1minutepoll”. Tämä näkymä on kuvattu kuviossa 27.



Kuvio 27. OpenNMS Polling Package

Nyt HTTP-logia katsottaessa mail.netfun.fi -palvelimelta huomataan, että OpenNMS hakee HTTP-GET kyselyllä HTTP-palvelun vasteajan joka minuutti. Tämä näkymä on kuvattu kuviossa 28.

```
-bash-4.1# tail /var/log/httpd/access_log
194.85.166.147 - - [27/Mar/2015:13:30:05 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
194.85.166.147 - - [27/Mar/2015:13:31:05 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
194.85.166.147 - - [27/Mar/2015:13:32:05 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
194.85.166.147 - - [27/Mar/2015:13:33:05 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
194.85.166.147 - - [27/Mar/2015:13:34:05 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
194.85.166.147 - - [27/Mar/2015:13:35:05 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
194.85.166.147 - - [27/Mar/2015:13:36:05 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
194.85.166.147 - - [27/Mar/2015:13:37:06 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
194.85.166.147 - - [27/Mar/2015:13:38:06 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
194.85.166.147 - - [27/Mar/2015:13:39:06 +0000] "GET / HTTP/1.1" 200 528 "-" "Op
enNMS HttpMonitor"
```

Kuvio 28. OpenNMS mail.netfun.fi HTTP-GET 1min

5.2.6 SNMP-agentin määrittäminen

Jotta OpenNMS osaa hakea lisää tietoa laitteesta, kannattaa laitteessa olla asennettuna SNMP-agentti nimeltä “net-snmp”. Täten saadaan laitteesta paljon lisätietoa, kuten prosessorikuormat, rajapintojen liikennemäärät, TCP-yhteyksien määrä ym.

Net-snmp-ohjelman asennus tapahtuu CentOS Linux -palvelimelle seuraavasti:

```
# yum install net-snmp net-snmp-utils
```

Seuraavaksi ”net-snmp”-työkalu täytyy vielä määrittää oikein. Määrittäminen tapahtuu tiedostoon /etc/snmp/snmpd.conf komennolla:

```
# nano /etc/snmp/snmpd.conf
```

Seuraavassa kuviossa 29 näkyy valmiiksi määritetty snmpd.conf -tiedosto. Muutokset on korostettu punaisilla merkinnöillä.

```
  GNU nano 2.0.9          File: /etc/snmp/snmpd.conf

com2sec local  localhost       public
com2sec mynetwork 194.85.166.128/28 public

####
# Second, map the security name into a group name:

#       groupName      securityModel securityName
group   MyRWGroup v2c           local
group   MyROGroup v2c           mynetwork

####
# Third, create a view for us to let the group have rights to:

# Make at least  snmpwalk -v 1 localhost -c public system fast again.
#       name           incl/excl     subtree         mask(optional)
view    systemview    included   .1.3.6.1.2.1.1
view    systemview    included   .1.3.6.1.2.1.25.1.1
view    all           included   .1                      80
####
# Finally, grant the group read-only access to the systemview view.

#       group          context sec.model sec.level prefix read   write  notif
access  MyROGroup ""      any       noauth    exact  all none none
access  MyRWGroup ""      any       noauth    exact  all all  none

syslocation Jyvsectec
syscontact Sakari Remonen
```

Kuvio 29. Net-SNMP-työkalun määritys

Edellä olevassa kuviossa 29 määritetään ryhmä ”mynetwork” ja sen verkko 194.85.166.128/28, joka vastaa verkkoa, josta SNMP saa hakea tietoa ja josta OpenNMS

toimii. Yhteisön eli ”Community” nimeksi on määritetty ”public”. Tämä tieto on erittäin tärkeä, koska tämän perusteella laitteet hakevat SNMP-tietoa.

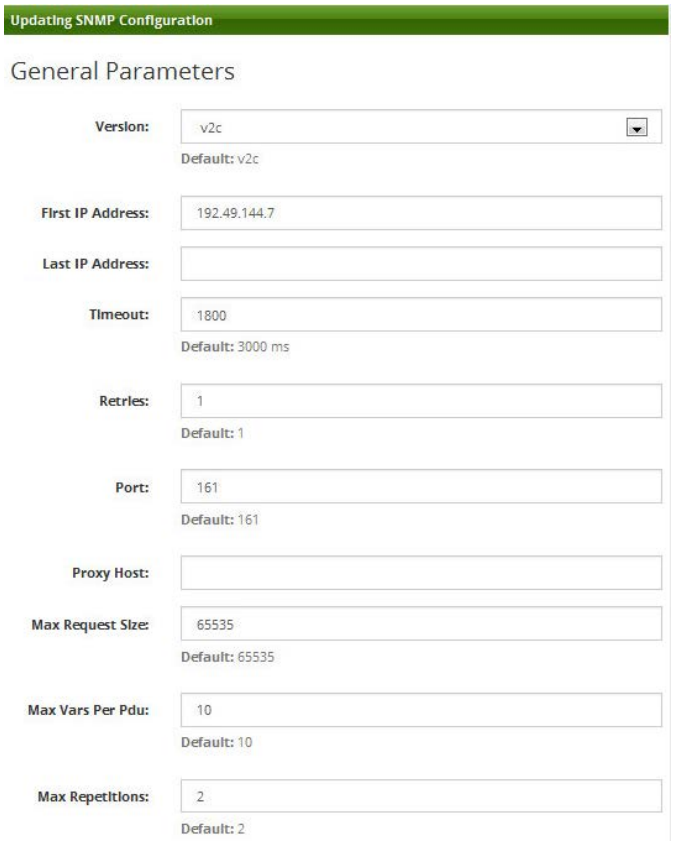
Seuraavaksi on määritetty ryhmät ”MyRWGroup” ja ”MyROGroup”, jotka on määritetty käyttämään SNMPv2c-versiota. Tämän jälkeen on annettu oikeudet lisäämällä ”view”-kohtaan ”all” ja juurihakemistoksi ”.1”. Tämä tarkoittaa sitä, että laitteesta voidaan hakea kaikki SNMP-tieto juuren alta.

Lisäksi on määritetty kohdat ”syslocation” ja ”syscontact”, joihin voi kirjoittaa laitteesta tarkempaa tietoa. Tämäkin tieto näkyy SNMP-kyselyssä ja OpenNMS:ssä.

Tämän jälkeen snmpd-palvelu vielä käynnistetään ja määritetään se käynnistymään automaattisesti aina käynnistyksen yhteydessä seuraavilla komennoilla:

```
# service snmpd start
# chkconfig snmpd on
```

Tämän jälkeen OpenNMS:sään on määritettävä SNMP-kohteen IP-osoite, yhteisön nimi ja SNMP-versio, jotta tietoja voidaan noutaa. Asetus löytyy kohdasta Admin -> Configure OpenNMS -> Configure SNMP Community Names by IP. Kuviossa 30 ja 31 on kuvattu näkymä, johon on määritetty vastaavat SNMP-asetukset, jotka juuri aikeisemmin määritettiin palvelimeen.



Kuvio 30. OpenNMS SNMP-parametrien määrittäminen

Kuvio 31. OpenNMS SNMP-yhteisö

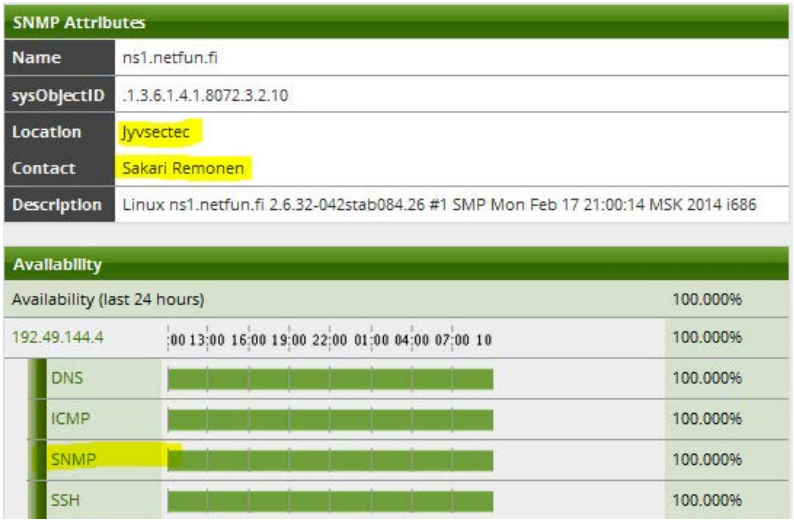
Tämä tekee muutokset tiedostoon /opt/opennms/etc/snmp-config.xml, johon vastaavat muutokset voi tehdä myös käsin ilman graafista ympäristöä. ”snmp-config.xml” tiedoston sisältö näyttää nyt tältä:

```
<snmp-config xmlns=”http://xmlns.opennms.org/xsd/config/snmp” version=”v2c” read-community=”public” timeout=”1800” retry=”1”>

  <specific>144.243.214.28</specific>

  <specific>192.49.144.4</specific>

  <specific>192.49.144.7</specific>

</snmp-config>
```

Tämän jälkeen OpenNMS-palvelin kannattaa vielä uudelleen käynnistää seuraavalla komennolla:

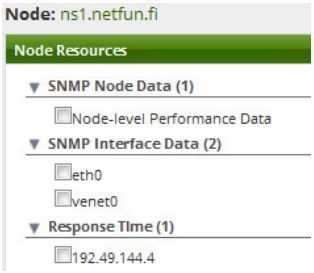
```
# shutdown –r now
```

Seuraavassa kuviossa 32 on ns1.netfun.fi -palvelin, jossa nähdään, että SNMP-palvelu on ylhäällä. Nähdään myös, että SNMP on hakenut määrittämämme sijainnin ”Jyvsectec” sekä kontaktin ”Sakari Remonen”.



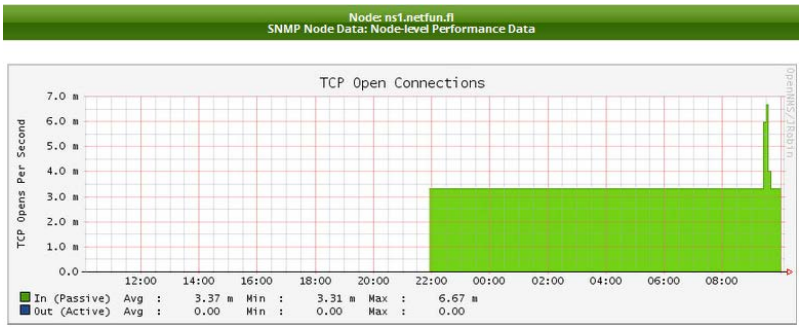
Kuvio 32. OpenNMS SNMP palvelu

Klikkaamalla laitteen “resource graphs”-kohtaa, löytyy nyt lisää tietoa, mitä on kerätty käyttäen SNMP-protokollaa. Tämä näkymä on kuvattu kuviossa 33.

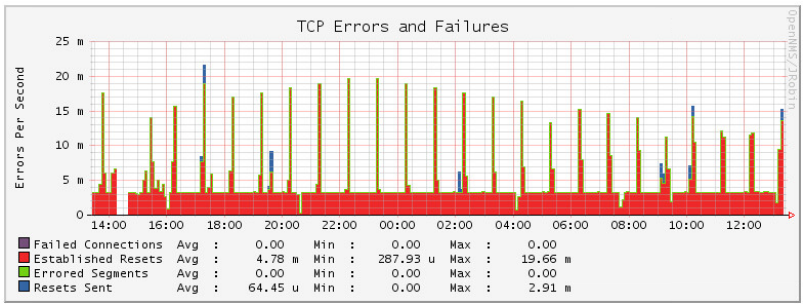


Kuvio 33. OpenNMS SNMP graphs

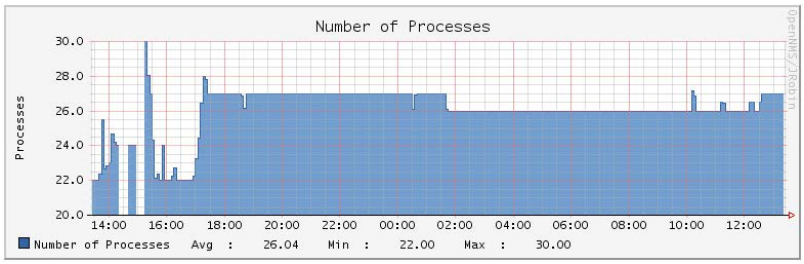
Klikkaamalla ”Node-level Performance Data” saadaan auki lisää SNMP:n keräämää tietoa. Kuvioissa 34, 35, 36 ja 37 nähdään muutama SNMP:n keräämä tieto ns1.netfun.fi -palvelimesta. Kuviossa nähdään muun muassa TCP-yhteyden tietoja, prosessien määrä ja järjestelmän muistin käyttö.



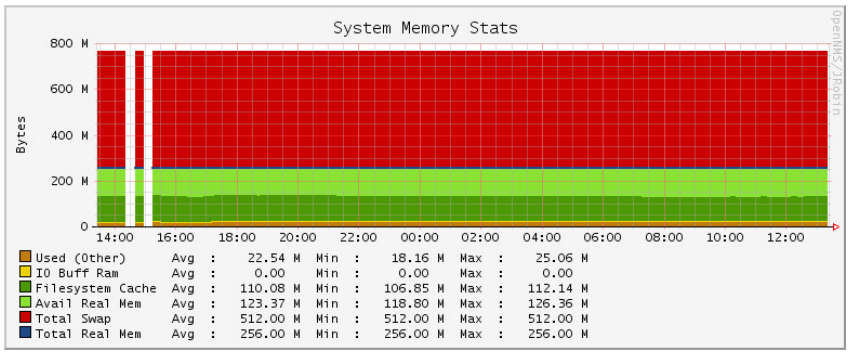
Kuvio 34. OpenNMS SNMP TCP Connections



Kuvio 35. OpenNMS SNMP TCP errors



Kuvio 36. OpenNMS SNMP Number of Processes



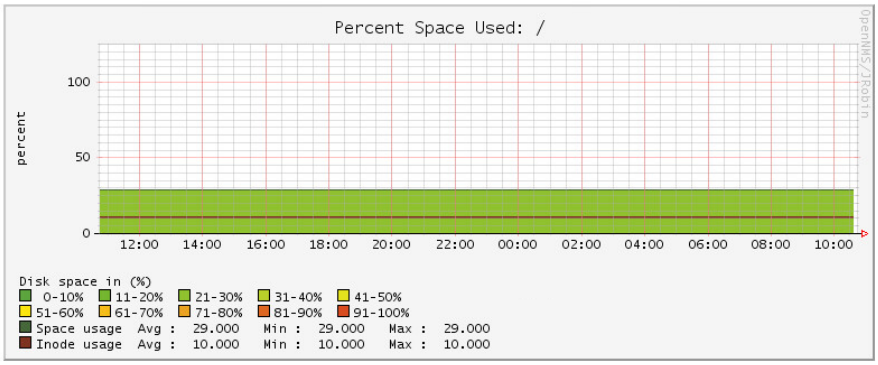
Kuvio 37. OpenNMS SNMP System Memory Stats

Palvelimelle mail.netfun.fi määritettiin net-SNMP antamaan vielä tietoja levytilan käytöstä. Tämä tapahtui lisäämällä snmpd.conf -tiedostoon seuraava rivi:

```
includeAll Disks 10%
```

```
disk / 10000
```

Edellä olevat komennot snmpd.conf -tiedostossa määrittävät sen, että SNMP-agentti antaa tietoa kaikista levyistä juuresta mitattuna. 10%- ja 10000-parametrit määrittävät sen, että levyissä tulee olla ainakin 10MB käytettynä, jotta sitä monitoroidaan. Seuraavassa kuviossa 38 nähdään palvelimen mail.netfun.fi levytilan käyttö.

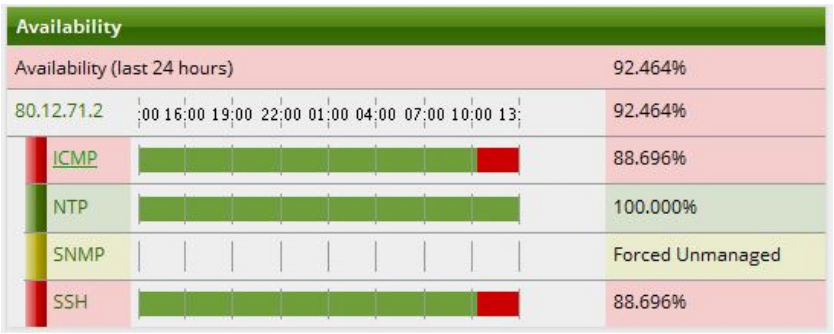


Kuvio 38. OpenNMS SNMP mail.netfun.fi levytilan käyttö

Tämän jälkeen vaihdettiin vielä “community” nimi nimeksi “opennms”, koska ei haluttu pitää oletusta “public” käytössä.

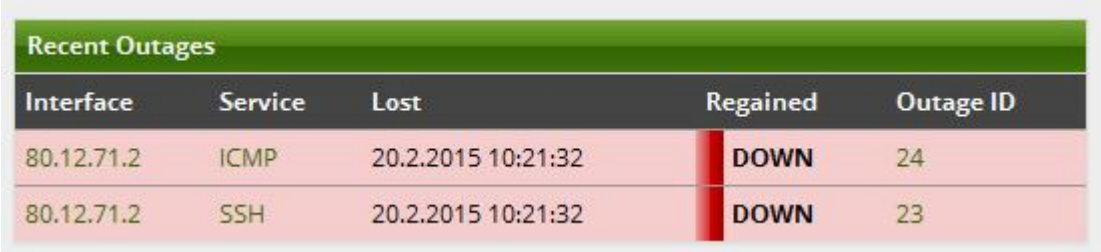
5.2.7 Hälytykset

Kun jokin palvelu on hidas tai poistuu kokonaan toimimasta, hälyttää OpenNMS siitä automaattisesti. Kun laitteen näkymän avaa, ne palvelut jotka ovat sammuneet tai ei jostain muusta syystä vastaa, näkyvät punaisena korostettuna. Esimerkkinä tästä on laite, joka ei vastaa ICMP-kyselyyn eikä myöskään SSH-palvelu vastaa (ks. kuvio 39). NTP-palvelu kuitenkin toimii.



Kuvio 39. OpenNMS-hälytys 1.

Samalla sivulla oikeassa reunassa on myös erikseen kohta, jossa näkyvät laitteen kaikki edellisetkin häiriöt. Kuviossa 40 on esimerkki NTP_B palvelimen ”Recent Outages”-kohdasta, joka ilmoittaa, että palvelut ICMP ja SSH ovat lakenneet vastaamasta samaan aikaan pävämäärällä 20.2.2015 kellonajalla 10:21:32.



Recent Outages

Interface	Service	Lost	Regained	Outage ID
80.12.71.2	ICMP	20.2.2015 10:21:32	DOWN	24
80.12.71.2	SSH	20.2.2015 10:21:32	DOWN	23

Kuvio 40. OpenNMS hälytys 2.

Hälytykset näkyvät myös suoraan etusivulla, jossa ilmoitetaan “Nodes with Pending Problems”. Kuviossa 41 on kuvattu etusivun näkymä, kun NTP_B-palvelin on kaatunut. Klikkaamalla hälytystä saa ongelmasta lisätietoa.

Nodes with Pending Problems

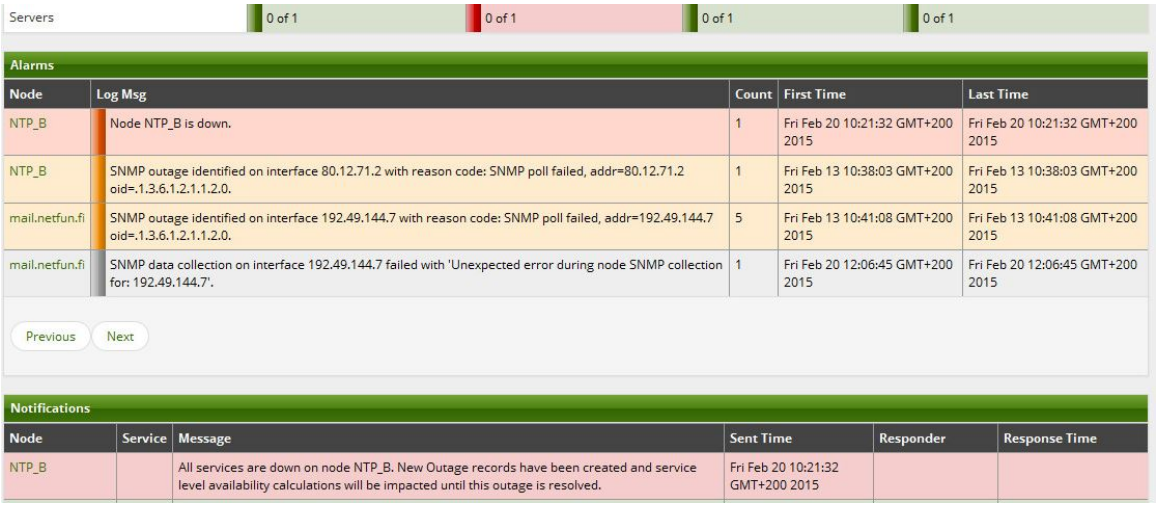
NTP_B has 1 alarm (3 hours)

Availability Over the Past 24 Hours

Categories	Outages	Availability
Network Interfaces	1 of 7	98.627%

Kuvio 41. OpenNMS hälytys 3.

Hälytykset näkyvät tietysti myös Dashboard näkymässä. Kuviossa 42 esitetty Dashboard näkymä ongelman tapahtuessa. Kuviosta 42 nähdään, että OpenNMS antaa ilmoituksen ”Node NTP_B is down”, ja ajan milloin näin on viimeksi tapahtunut.



Servers | 0 of 1 | 0 of 1 | 0 of 1 | 0 of 1

Alarms

Node	Log Msg	Count	First Time	Last Time
NTP_B	Node NTP_B is down.	1	Fri Feb 20 10:21:32 GMT+200 2015	Fri Feb 20 10:21:32 GMT+200 2015
NTP_B	SNMP outage identified on interface 80.12.71.2 with reason code: SNMP poll failed, addr=80.12.71.2 oid=.1.3.6.1.2.1.1.2.0.	1	Fri Feb 13 10:38:03 GMT+200 2015	Fri Feb 13 10:38:03 GMT+200 2015
mail.netfun.fi	SNMP outage identified on interface 192.49.144.7 with reason code: SNMP poll failed, addr=192.49.144.7 oid=.1.3.6.1.2.1.1.2.0.	5	Fri Feb 13 10:41:08 GMT+200 2015	Fri Feb 13 10:41:08 GMT+200 2015
mail.netfun.fi	SNMP data collection on interface 192.49.144.7 failed with 'Unexpected error during node SNMP collection for: 192.49.144.7'.	1	Fri Feb 20 12:06:45 GMT+200 2015	Fri Feb 20 12:06:45 GMT+200 2015

Previous Next

Notifications

Node	Service	Message	Sent Time	Responder	Response Time
NTP_B		All services are down on node NTP_B. New Outage records have been created and service level availability calculations will be impacted until this outage is resolved.	Fri Feb 20 10:21:32 GMT+200 2015		

Kuvio 42. OpenNMS hälytys 4.

Kaikki hälytykset saa myös listattua kohdasta ”Status – Outages – All outages”, josta valitaan ”Both current & Resolved”.

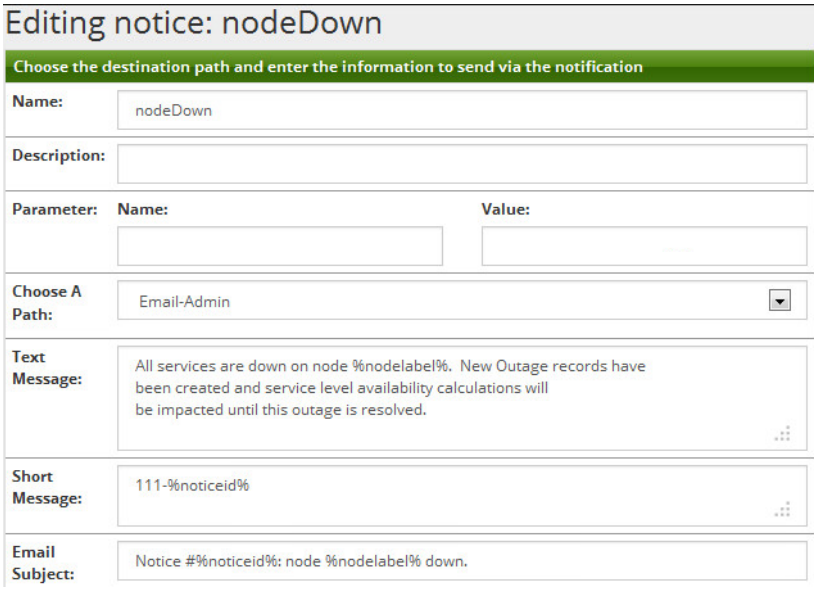
Seuraavat hälytykset on määritetty oletuksena OpenNMS:sään. Käyttäjä voi halutessaan lisätä myös omia hälytyksiä tietyillä raja-arvoilla. Kuviossa 43 on OpenNMS:sään oletuksena määritetyt hälytykset, joissa on muun muassa ”nodeDown” ja ”High Threshold”, jotka tarkoittavat sitä, että OpenNMS hälyttää, kun näihin sopiva tapahtuma tapahtuu.

Event Notifications

Actions			Notification	Event	UEI
Edit	Delete	Off On	High Threshold	OpenNMS-defined threshold event: highThresholdExceeded	uei.opennms.org/threshold/highThresholdExceeded
Edit	Delete	Off On	High Threshold Rearmed	OpenNMS-defined threshold event: highThresholdRearmed	uei.opennms.org/threshold/highThresholdRearmed
Edit	Delete	Off On	Low Threshold	OpenNMS-defined threshold event: lowThresholdExceeded	uei.opennms.org/threshold/lowThresholdExceeded
Edit	Delete	Off On	Low Threshold Rearmed	OpenNMS-defined threshold event: lowThresholdRearmed	uei.opennms.org/threshold/lowThresholdRearmed
Edit	Delete	Off On	interfaceDeleted	OpenNMS-defined node event: interfaceDeleted	uei.opennms.org/nodes/interfaceDeleted
Edit	Delete	Off On	interfaceDown	OpenNMS-defined node event: interfaceDown	uei.opennms.org/nodes/interfaceDown
Edit	Delete	Off On	nodeAdded	OpenNMS-defined node event: nodeAdded	uei.opennms.org/nodes/nodeAdded
Edit	Delete	Off On	nodeDown	OpenNMS-defined node event: nodeDown	uei.opennms.org/nodes/nodeDown
Edit	Delete	Off On	nodeLostService	OpenNMS-defined node event: nodeLostService	uei.opennms.org/nodes/nodeLostService

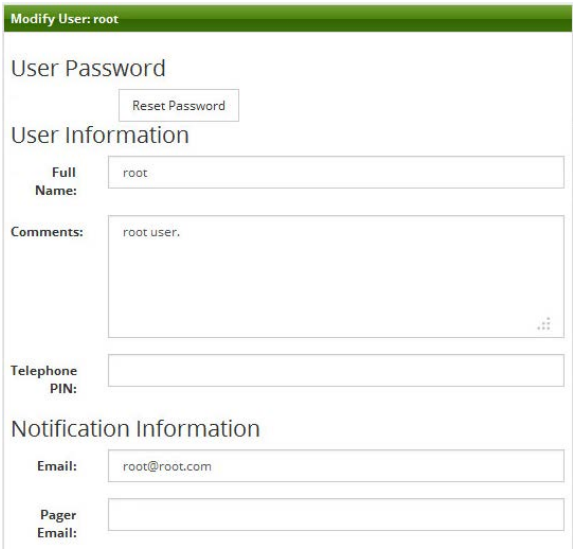
Kuvio 43. OpenNMS-hälytykset

Hälytyksiä pystyy lisäämään, poistamaan ja muokkaamaan. Esimerkkinä kuviossa 44 hälytykselle ”nodeDown” voidaan määrittää kenelle ja minkälaisen sähköpostin järjestelmä lähettää ylläpitäjille, kun ongelma ”nodeDown” tapahtuu. Kyseisessä esimerkissä on määritetty OpenNMS lähettämään sähköpostia järjestelmän ylläpitäjälle.



Kuvio 44. OpenNMS "nodeDown"-hälytys

Sähköposti osoitteet, johon OpenNMS lähettää hälytykset on määritetty jokaisen käyttäjän ominaisuuksiin. Esimerkkinä käyttäjän ”root” sähköposti-osoitteen määritys on esitetty kuviossa 45.



Kuvio 45. OpenNMS-käyttäjätiedot

Käyttäjälle voi myös määrittää työskentely ajat eli ”Duty Schedule”, jonka perusteella OpenNMS osaa valita kenelle lähettää hälytys tiettyinä aikoina.

5.2.8 Hälytyksien määritys

OpenNMS:n haluttiin hälyttävän, kun esimerkiksi HTTP-palvelu vastaa hitaasti. OpenNMS:sään määritettiin käsin nämä hälytys raja-arvot. Esimerkiksi kun HTTP-palvelun vasteaika ylittää yhden sekunnin, tulee järjestelmän ylläpitäjälle hälytys tästä sähköpostilla.

Hälytyksien luominen aloitetaan luomalla hälytyspaketti tiedostoon /opt/opennms/etc/threshd-configuration.xml, johon lisätään seuraavat rivit:

```
<package name="http-latency">
    <filter>IPADDR != '0.0.0.0'</filter>
    <include-range begin="1.1.1.1" end="254.254.254.254"/>
    <include-range begin="::1"
end="ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff"/>
    <service name="HTTP" interval="300000" user-defined="true"
status="on">
        <parameter key="thresholding-group" value="http-latency"/>
    </service>
</package>
```

Edellä olevilla riveillä määritetään mitä laitteita kyseinen hälytysarvo koskee. Itse varsinaiset raja-arvot määritetään vasta seuraavassa kohdassa. Edellä olevassa komennossa myös muuttujan “thresholding-group” arvo tulee olla sama kuin kohta luotavassa hälytysryhmässä.

Seuraavaksi luodaan hälytys raja-arvot tiedostoon /opt/opennms/etc/thresholds.xml, jonne kirjoitetaan seuraavat rivit:

```
<group name="http-latency" rrdReposito-
ry="/opt/opennms/share/rrd/response/">
       <threshold type="high" ds-type="if" value="1000.0"
           rearm="500.0" trigger="1" triggered-
UEI=”threshold/http/latency/exceeded” ds-name="http"/>
   </group>
```

Edellä olevilla riveillä määritetään seuraavaa:

- Hälytysryhmän nimi on ”http-latency”.

- Tallennetut viivetiedot eli ”rrdRepository”-kansio on määritetty ”/opt/opennms/share/rrd/response/”.

- Raja-arvon tyypiksi on määritetty ”high” mikä tarkoittaa, että hälytys laukeaa, kun tämä kyseinen arvo ylittyy.

- ”ds-type” on määritetty ”if” eli rajapinnaksi tässä tapauksessa.

- Itse raja-arvo määritetään ”value”-kohdassa, jossa on millisekuntien määrä, tässä tapauksessa 1000, eli 1 sekunti.

- ”rearm”-arvo kertoo milloin hälytys poistuu. Tässä tapauksessa näin tehdään, kun viive tippuu takaisin puoleen sekuntiin.

- ”trigger”-arvo kertoo, kuinka monta näytettä tulee olla ”rearm”-kentän vastineita, jotta hälytys poistuu.

- ”triggeredUEI” on käyttäjän itse määrittämä kenttä, joka kertoo hierarkisesti tapahtumasta. Tämän lisääminen auttaa jälkikäteen luomaan kyseisestä raja-arvosta hälytystapahtuman.

- ”ds-name”-kenttään määritetään vielä palvelu, jonka viivettä monitoroidaan. Tässä tapauksessa monitoroidaan HTTP-palvelun viivettä.

Vielä pitää käydä lisäämässä tiedostoon /opt/opennms/etc/poller-configuration.xml oikean palvelun alle rivi:

```
<parameter key="thresholding-enabled" value="true"/>
```

Seuraavaksi määritetään vielä OpenNMS lähettämään kyseisestä hälytyksestä sähköpostia ylläpitäjälle. Tämä tapahtuu graafisen käyttöliittymän kautta kohdasta ”Configure OpenNMS -> Manage thresholds”, jossa nähdään nyt juuri luotu ”http-latency”-ryhmä. Kuviossa 46 on kuvattu ”threshold configuration”-näkymä. Painamalla kohtaa ”Edit”, päästään muokkaamaan kyseistä ryhmää.

Threshold Configuration

Name	RRD Repository	
cisco	/opt/opennms/share/rrd/snmp/	Edit
coffee	/opt/opennms/share/rrd/snmp/	Edit
hrstorage	/opt/opennms/share/rrd/snmp/	Edit
http-latency	/opt/opennms/share/rrd/response/	Edit

Kuvio 46. OpenNMS thresholds

Seuraavaksi avautuu kuviossa 47 näkyvä sivusto, josta klikkaamalla kohtaa ”Triggered UEI” päästään luomaan kyseisestä raja-arvosta hälytystapahtuma.

Edit group http-latency

Basic Thresholds

Type	Description	Datasource	Datasource type	Datasource label	Value	Re-arm	Trigger	Triggered UEI	Re-armed UEI		
high		http	if		1000.0	500.0	1	threshold/http/latency/exceeded	threshold/http/latency/rearmed	Edit	Delete

Kuvio 47. Raja-arvojen muokkaaminen

Nyt kyseiseen hälytykseen voidaan muokata lähetettävä viesti ja kohde, kun raja-arvo ylittyy. Kuviossa 48 on luotu sähköpostiviesti käyttäjälle ”Admin”, jonka otsikkona on ”HTTP service response time slow on node %nodelabel& / %interfaceresolve%”, jossa otsikossa on muuttujat ”%nodelabel%” ja ”%interfaceresolve%”, jotka kertovat kyseisen laitteen IP-osoitteen ja nimen.

Editing notice: HTTP service response time slow

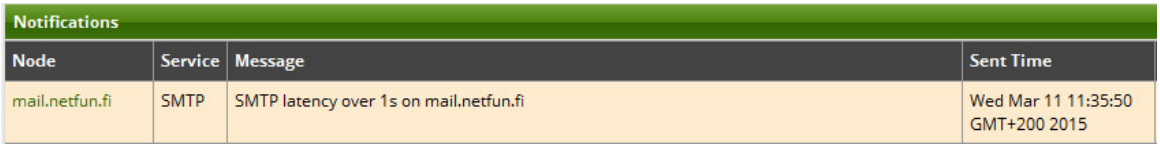
Choose the destination path and enter the information to send via the notification

Name:	HTTP service response time slow	
Description:	HTTP service response time slow	
Parameter:	Name:	Value:
Choose A Path:	Email-Admin	
Text Message:	HTTP service in %nodelabel% is answering slow (over 1s).	
Short Message:	Notice %noticeid% HTTP service slow in %nodelabel%	
Email Subject:	Notice #%noticeid% HTTP service slow in %nodelabel%	

Kuvio 48. OpenNMS thresholds hälytys

Nyt OpenNMS on konfiguroitu hälyttämään, kun HTTP-palvelu vastaa yli yhden sekunnin vasteajalla.

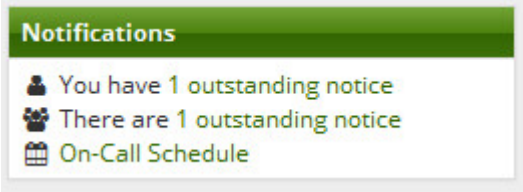
Seuraavissa kuviossa 49, 50 ja 51 on kuvattu ilmoitus, kun SMTP-palvelun viive on noussut yli raja-arvon. Hälytys näkyy etusivulla, notifications- ja dashboard -näkymässä.



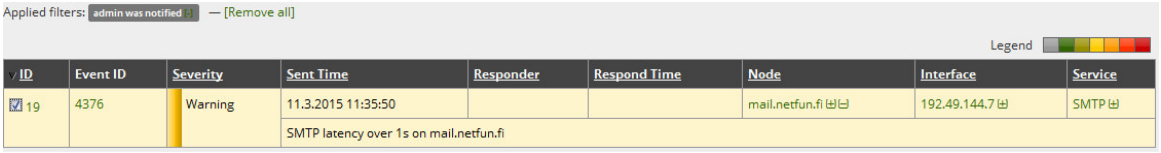
Notifications

Node	Service	Message	Sent Time
mail.netfun.fi	SMTP	SMTP latency over 1s on mail.netfun.fi	Wed Mar 11 11:35:50 GMT+200 2015

Kuvio 49. OpenNMS hälytys SNMP hidas 1.



Kuvio 50. OpenNMS hälytys SNMP hidas 2.



Kuvio 51. OpenNMS hälytys SNMP hidas 3.

Kuviosta 51 huomataan myös, että hälytyksen vakavuustaso on “Warning”, koska palvelu toimii, eikä ole kuin hidastunut. Huomataan myös, että hälytyksessä ilmoitetaan ”Admin was notified”, eli järjestelmä ilmoittaa, että myös sähköposti on lähetetty ylläpitäjälle.

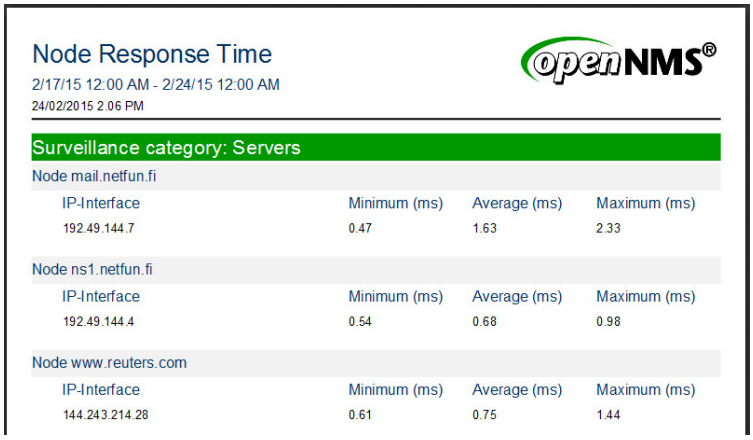
5.2.9 Raportit

OpenNMS osaa luoda automaattisesti .PDF muotoisia (Portable Document Format) raportteja laitteista ja niiden palveluista sekä lähettää näitä automatisoidusti halutuille kohteille. Kohdasta ”Reports – Database reports – List reports” nähdään saatavilla olevat raportit. Kuviossa 52 on kuvattu kaikki oletuksena käytössä olevat OpenNMS-raportit. Nämä voidaan tulostaa heti, lähettää sähköpostiin tai ajastaa ja automatisoida tämä toiminto toistuvaksi.

Local Report Repository	
Name	**Description**
Default calendar report	standard opennms report in calendar format
Default classic report	standard opennms report in tabular format
Early morning report	Global overview of outages, notifications and events in last 24 hours
Response Time Summary for node	Response Time by node across one or more surveillance categories
Availability by node	Availability by node across one or more surveillance categories
Availability Summary -Default configuration for past 7 Days	Availability summary across one or more surveillance categories
Response time by node	Response time by node across one or more surveillance categories
Serial Interface Utilization Summary	Serial Interface Utilization Summary
Total Bytes Transferred by Interface	Total Bytes Transferred by Interface
Average and Peak Traffic rates for Nodes by Interface	Average and Peak Traffic rates for Nodes by Interface

Kuvio 52. OpenNMS raportit

Kuviossa 53 on kuvattu .PDF muotoinen raportti kategorian ”Servers” vastaus ajasta. Raportista nähdään eri laitteiden minimi-, keskiarvo- ja maksimi vasteaika.



Node Response Time

2/17/15 12:00 AM - 2/24/15 12:00 AM

24/02/2015 2.06 PM

Surveillance category: Servers

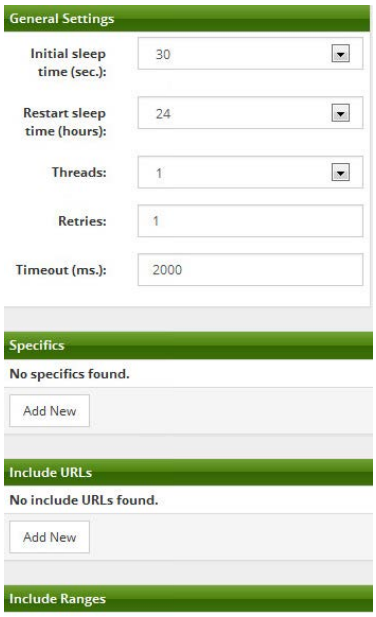
Node mail.netfun.fi			
IP-Interface	Minimum (ms)	Average (ms)	Maximum (ms)
192.49.144.7	0.47	1.63	2.33
Node ns1.netfun.fi			
IP-Interface	Minimum (ms)	Average (ms)	Maximum (ms)
192.49.144.4	0.54	0.68	0.98
Node www.reuters.com			
IP-Interface	Minimum (ms)	Average (ms)	Maximum (ms)
144.243.214.28	0.61	0.75	1.44

Kuvio 53. OpenNMS-raportti

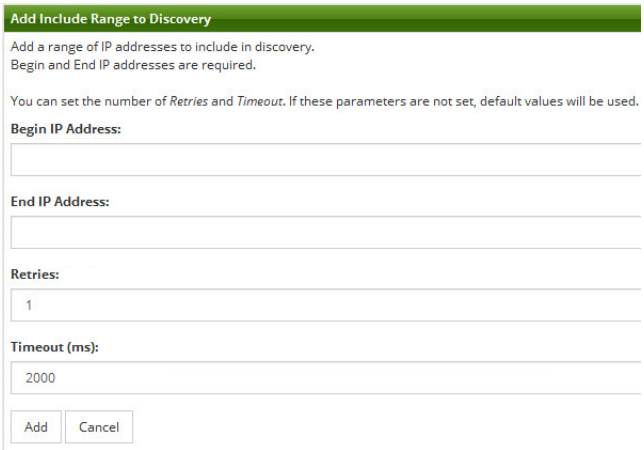
5.2.10 Discovery

OpenNMS tukee auto-discovery-ominaisuutta, joka osaa etsiä verkosta laitteita sekä palveluita automaattisesti. OpenNMS:n discovery-ominaisuus toimii käyttämällä ICMP Echo-kyselyitä määritettyihin IP-alueisiin. Kohteet jotka vastaavat kyselyyn, OpenNMS luo ”newSuspectEvent”-tapahtuman kohteesta, ja lähtee tutkimaan kohteen palveluita tarkemmin, jonka seurauksena uusi laite luodaan automaattisesti järjestelmään.

Discovery-ominaisuus löytyy kohdasta ”Configure OpenNMS – Configure Discovery”, johon voidaan määrittää muun muassa IP-alue, johon Discovery-toiminne halutaan kohdentaa, aikakatkaisu (timeout) ja uudelleenyrityksien määrä (retries). Halutessaan voi myös määrittää IP-alueen, jota ei halua tutkittavan. Kuvioissa 54 ja 55 on kuvattu Discoveryn asetuksia.

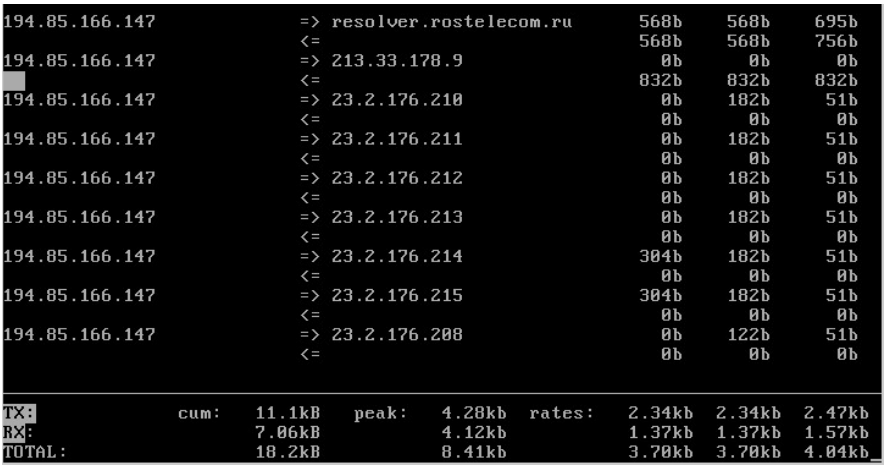


Kuvio 54. OpenNMS Discovery



Kuvio 55. OpenNMS Discovery IP-alueet

Tämän määrityksen jälkeen OpenNMS alkaa automaattisesti lähettämään ICMP Echo-kyselyitä määritettyihin IP-alueisiin. Kuviossa 56 on kuvattu OpenNMS:n palvelimen verkkoliikenne discovery-määrityksen jälkeen. Kuviossa nähdään, että palvelin lähettää kyselyitä järjestyksessä annettuihin verkkoalueisiin.



Kuvio 56. OpenNMS ICMP discovery

Kuviossa 57 on kuvattu OpenNMS:n ilmoitukset, kun OpenNMS:n discovery löytää verkosta monitoroitavia kohteita. Kuviossa 57 nähdään, että muun muassa laite nimeltä ”mail.telia.se” on löytynyt.

ID	Event ID	Severity	Sent Time	Responder	Respond Time	Node
28	4773	Warning	13.3.2015 8:35:04			mail.telia.se
			OpenNMS has discovered a new node named mail.telia.se. Please be advised.			
27	4767	Warning	13.3.2015 8:35:01			resolver.telia.se
			OpenNMS has discovered a new node named resolver.telia.se. Please be advised.			
26	4764	Warning	13.3.2015 8:35:00			ns1.telia.se
			OpenNMS has discovered a new node named ns1.telia.se. Please be advised.			
25	4761	Warning	13.3.2015 8:34:58			ntp.telia.se
			OpenNMS has discovered a new node named ntp.telia.se. Please be advised.			

Kuvio 57. OpenNMS discovery-toiminteen tulokset

5.2.11 Prosessit

Toimiakseen OpenNMS vaatii paljon taustapalveluita. Kaikki OpenNMS:n taustapalvelut saadaan näkyviin kirjoittamalla seuraava komento palvelimelle:

```
#service opennms –v status
```

Seuraavassa kuviossa 58 on avattu OpenNMS-palvelimen kaikki OpenNMS-palvelut.

Kuvio 58. OpenNMS palvelut

Seuraavassa on listaus tärkeimmistä palveluista ja niiden tehtävistä:

OpenNMS.Pollerd

- Käsittelee kaikki palveluiden kiertokyselyt ja vasteaikojen nauhoittamisen.

OpenNMS.Collectd

- Kerää ja tallentaa tietoa eri lähteistä, sisältäen SNMP, JMX, http ja NSClientin.

OpenNMS.Discovery

- Mahdollistaa tavan jolla OpenNMS osaa etsiä verkosta laitteita. Voi olla joko automaattinen tai manuaalinen.

OpenNMS.Eventd

- Vastaanottaa ja kirjoittaa kaikki tapahtuma tiedot.

OpenNMS.Trapd

- Käsittelee SNMP trap tietojen prosessoinnin. Trapd palvelu mahdollistaa OpenNMS:n vastaanottaa SNMP trap-viestejä ja tehdä niistä tapahtumia ja niin edelleen hälytyksiksi tai ilmoituksiksi.

OpenNMS.Actiond

- Käytetään generoimaan Java-toimenpiteitä perustuen vastaanotettuihin tapahtumiin.

OpenNMS.Notifd

- Luo ilmoitusviestit perustuen tapahtumiin.

OpenNMS.Scriptd

- Vastaavanlainen kuin Actiond, generoi ulkoisia toimenpiteitä perustuen vastaanotettuihin tapahtumiin.

5.2.12 SNMP tiedon keräys

OpenNMS kerää verkon laitteista tietoa SNMP-protokollaa käyttäen ja tallentaa niistä tiedot ja vasteajat palvelimelle ”.jrb”-muotoisiksi tiedostoiksi kansioon /var/opennms/rrd/snmp/, josta niistä voidaan tehdä erilaisia grafiikoita OpenNMS:n graafiseen ympäristöön. Tiedon kerääminen on määritetty tiedostossa /opt/opennms/etc/datacollection-config.xml, joka edelleen viittaa eri tiedonkeruuryhmiin eli ”dataCollectionGroup”:eihin. Tiedonkeruuryhmät ja niiden tarkemmat määritykset löytyvät kansiosta /opt/opennms/etc/datacollection/. Yksi tärkeimmistä tiedostoista on netsnmp.xml, johon on määritetty kaikki SNMP-agentilla haettavat tiedot, ja johon voi myös itse lisätä halutessaan lisää kerättävää tietoa. Esimerkkinä kohdassa 5.2.11 luotu käyttäjän itsemäärittämä SNMP kohde, jota halutaan kerätä.

5.2.13 Käyttäjän itse määrittämä SNMP tiedon keräys

Tavoitteena oli saada kerättyä sähköpostipalvelimelta mail.netfun.fi sähköpostijonon koko lukumääränä OpenNMS-palvelimelle grafiikkana. Palvelimelta saatiin näkyviin sähköpostijonon koko komennolla:

```
# mailq
```

Joka tulostaa vastaukseksi sähköpostijonon koon seuraavanlaisesti:

```
-- 23 Kbytes in 6 Requests.
```

Seuraavaksi täytyi tehdä skripti, joka poistaa kaiken muun paitsi luvun “6” tulosteesta. Tämä tehtiin luomalla seuraavanlainen skripti tiedostoksi /bin/mailqstats.sh

```
#!/bin/bash
mailq | tail -n 1 | awk '{if (NF > 4) {print $5} else {print 0}}'
```

Seuraavaksi tämä juuri luotu skripti täytyi määrittää /etc/snmp/snmpd.conf tiedostoon. Tämä tapahtui lisäämällä seuraava komento tiedostoon:

```
extend mailqstats /root/bin/mailqstats.sh
```

Kun snmpd on tämän jälkeen uudelleenkäynnistetty, voidaan juuri luotua uutta SNMP tietoa kokeilla. Tämä voidaan tehdä komennolla:

```
snmpwalk –v2c -c public localhost .1.3.6.1.4.1.8072.1.3.2
```

Yllä esitetty komento tuottaa onnistuessaan seuraavanlaisen tulosteen:

```
NET-SNMP-EXTEND-MIB::nsExtendNumEntries.0 = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendCommand."mailqstats" = STRING:
/bin/mailqstats.sh
NET-SNMP-EXTEND-MIB::nsExtendArgs."mailqstats" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendInput."mailqstats" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendCacheTime."mailqstats" = INTEGER: 5
NET-SNMP-EXTEND-MIB::nsExtendExecType."mailqstats" = INTEGER: ex-
ec(1)
NET-SNMP-EXTEND-MIB::nsExtendRunType."mailqstats" = INTEGER: run-
on-read(1)
NET-SNMP-EXTEND-MIB::nsExtendStorage."mailqstats" = INTEGER: per-
manent(4)
NET-SNMP-EXTEND-MIB::nsExtendStatus."mailqstats" = INTEGER: ac-
tive(1)
NET-SNMP-EXTEND-MIB::nsExtendOutput1Line."mailqstats" = STRING: 6
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."mailqstats" = STRING: 6
NET-SNMP-EXTEND-MIB::nsExtendOutNumLines."mailqstats" = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendResult."mailqstats" = INTEGER: 0
NET-SNMP-EXTEND-MIB::nsExtendOutLine."mailqstats".1 = STRING: 6
```

Tuloksien OID käyttää “extend”-funktion nimeä “mailqstats” ja on näin käännettynä ASCII arvoihin “109.97.105.108.113.115.116.97.116.115”.

Seuraavaksi tulee määrittää OpenNMS-palvelin noutamaan tämä tieto. Tämä määritys tehtiin tiedostoon /opt/opennms/etc/datacollection/netsnmp.xml, johon lisättiin seuraavat rivit:

```
<group name="mailq-stats" ifType="ignore">
   <mibObj
oid=".1.3.6.1.4.1.8072.1.3.2.4.1.2.10.109.97.105.108.113.115.116.9
7.116.115"
      instance="1" alias="mailqsize" type="octetstring" />
</group>
```

Kohtaan <*systemdef*> pitää vielä lisätä seuraava rivi, jotta OpenNMS sisällyttää ”mailqstats”-ryhmän kerättävien listaan.

```
<systemDef name="Net-SNMP">
  <sysoidMask>.1.3.6.1.4.1.8072.3.</sysoidMask>
  <collect>
     <includeGroup>mailq-stats</includeGroup>
   </collect>
</systemDef>
```

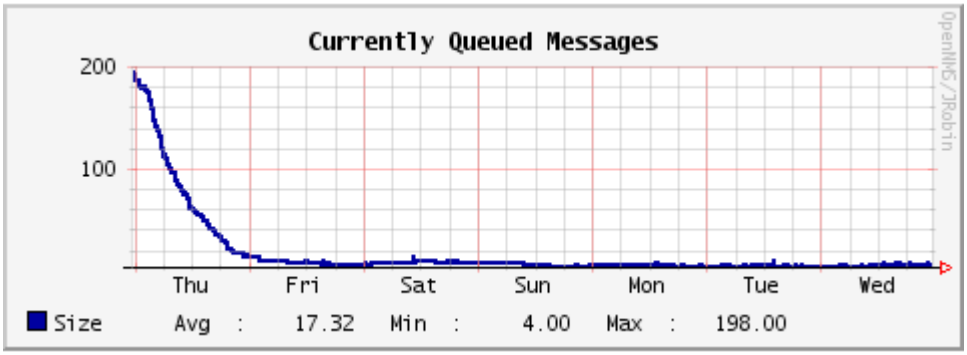
Mikäli SNMP-tiedon keräys onnistuu, luo OpenNMS-palvelin siitä tiedoston mailqsize.jrb-palvelimelle. Seuraavana vaiheena on luoda kerätystä tiedosta grafiikka. Tämä tapahtuu lisäämällä seuraavat rivit tiedostoon /opt/opennms/etc/snmp-graph.properties.d/netsnmp-graph.properties

```
report.netsnmp.mailq.name=Current Mail Queue Size
report.netsnmp.mailq.columns=mailqsize
report.netsnmp.mailq.type=nodeSnmp
report.netsnmp.mailq.command=--title="Currently Queued Messages" \
 DEF:queue={rrd1}:mailqsize:AVERAGE \
 LINE2:queue#0000A0:"Size    " \
 GPRINT:queue:AVERAGE:"Avg  \\: %8.2lf " \
 GPRINT:queue:MIN:"Min  \\: %8.2lf " \
 GPRINT:queue:MAX:"Max  \\: %8.2lf \\n"
```

Täytyy muistaa myös lisätä juuri luotu “netsnmp.mailq“-grafiikka tiedoston yläosassa olevaan ”reports=”-kohtaan seuraavalla tavalla:

```
reports=netsnmp.mailq, \
```

Nyt käyttäjän määrittämä SNMP-tiedonkeruu on valmis, ja grafiikka sähköpostijonosta pitäisi näkyä laitteen mail.netfun.fi alla kohdassa ”Node-level Performace Data”, kuten kuviossa 59 on esitetty.



Kuvio 59. SNMP sähköpostijonon koko (Tarus 2009)

Valitettavasti tässä työssä en onnistunut jostain syystä tätä tietoa keräämään oikein, koska “.jrb”-tiedostoa ei OpenNMS luonut palvelimelle. Yksi mahdollinen syy voi olla,

että sähköpostijono oli kokoajan tyhjä, eli kyselyyn tulostui ”0”, jonka vuoksi tiedostoa ei luoda.

Kuitenkin SNMP-haku onnistuu palvelimelta, kun sitä kokeilee komennolla:

```
# /opt/opennms/bin/snmp-request –c opennms –v 2c 192.49.144.7
.1.3.6.1.4.1.8072.1.3.2.4.1.2.10.109.97.105.108.113.115.116.97.116.
115
```

Tämä tulostaa oikein sähköpostijonon koon. On syytä huomioida, että edellisessä komennossa SNMP community -arvo on “opennms”, koska tämä vaihdettiin työn loppupuolella.

SNMP-tiedonkeruu määrityksen voi myös tehdä graafisen ympäristön kautta kohdasta ”Manage SNMP Collections and Data Collection Groups”, joka muokkaa tiedonkeräämisessä käytettäviä ”.xml”-tiedostoja vastaavalla tavalla kuin juuri tehty menetelmä.

5.2.14 Uuden monitoroitavan palvelun lisääminen

OpenNMS:n pystyy asettamaan monitoroimaan myös käyttäjän itse määrittämiä palveluita. OpenNMS ei automaattisesti monitoroi aikapalvelua “NTP”, joten tämä täytyi itse lisätä monitorointiin. Uuden palvelun monitorointiin lisääminen aloitettiin muokkaamalla tiedostoa /opt/opennms/etc/capsd-configuration.xml, johon lisättiin seuraavat rivit:

```
 <protocol-plugin protocol="NTP"
class-name="org.opennms.netmgt.capsd.NtpPlugin" scan="on" user-
defined="false">
 <property key="port" value="123"/>
 <property key="timeout" value="3000"/>
 <property key="retry" value="2"/>
 </protocol-plugin>
```

Edellä olevilla määrityksillä määritetään protokolla nimeltä ”NTP”, joka käyttää porttia 123. Tällä määrityksellä OpenNMS tunnistaa NTP-palvelun, jos se on käytössä laitteessa.

Seuraavaksi OpenNMS määritettiin tekemään kiertokyselyitä NTP-palveluun, jotta palvelusta saadaan monitoroitavaa tietoa, kuten esimerkiksi sen vasteaika, sekä nähdään onko palvelu toimiva vai ei. Tämä määritys tehtiin tiedostoon /opt/opennms/etc/poller-configuration.xml, johon lisättiin rivit:

```
 <service name="NTP" interval="300000" user-defined="false"
status="on">
 <parameter key="retry" value="3"/>
 <parameter key="timeout" value="5000"/>
 <parameter key="port" value="123"/>
 <parameter key="rrd-repository"
value="/var/opennms/rrd/response"/>
 <parameter key="ds-name" value="ntp"/>
 </service>

<monitor service="NTP"
class-name="org.opennms.netmgt.poller.NtpMonitor" />
```

Edellä olevilla määrityksillä OpenNMS tekee kiertokyselyitä NTP-palveluun joka viides minuutti, jonka jälkeen OpenNMS tallentaa palvelun vaste-ajat /var/opennms/rrd/response/ -kansioon. Nyt palvelu NTP on lisätty monitoroitaviin palveluihin ja OpenNMS tunnistaa NTP monitoroitavaksi palveluksi kuten kuviossa 60 on esitetty.

Kuvio 60. OpenNMS NTP-palvelun monitorointi

5.2.15 Muut toiminnot

OpenNMS sisältää myös paljon muita toimintoja, joita ei tässä työssä ollut tarvetta käyttää. Muutamia käteviä lisätoimintoja OpenNMS:ssä on:

- Hajautettu monitorointi; sallii usean OpenNMS-palvelimen asentamisen eripuolille verkkoa.

- Kartat; sallii käyttäjän luoda täydellisen verkkotopologian verkostaan. Myös geograafisen kartan luonti on mahdollista.

- Ajoitetut katkokset; voidaan määrittää laitteelle tai palvelulle ajoitettu katkos, jolloin OpenNMS ei hälytä, kun palvelu poistuu käytöstä.

- HTTP Collector; pystytään keräämään suorituskyvyn mittatietoja käyttämällä HTTP-protokollaa.

- Path Outages; kun OpenNMS:sään on määritetty paljon aktiivilaitteita, osaa OpenNMS kertoa missä verkkopoluissa on ollut ongelmia.

5.3 Lopullinen toteutus

Kun OpenNMS-palvelin oli asennettu, määritetty ja testattu onnistuneesti muutamalla testipalvelimella, lisättiin yhdessä toimeksiantajan kanssa järjestelmään paljon RGCE-verkon palvelimia ja laitteita ”auto-discovery”-ominaisuudella. Discovery sivulta OpenNMS määritettiin skannaamaan haluttuja verkkoalueita kolmen vuorokauden välein uusista laitteista. Palvelin siirrettiin myös lopulliseen paikkaan ja sille määritettiin nimi ja verkko-osoite RGCE-verkon juuri-DNS-palveluun. Loppujen lopuksi OpenNMS saatiin asennettua JYVSECTEC:in tuotantojärjestelmään monitoroimaan miltei kaikkia RGCE–verkon palveluita.

6 Pohdinta

Onnistuin mielestäni hyvin rakentamaan toimeksiantajalle toimeksiantajan vaatimuksiin perustuvan verkon monitorointi palvelun. Vaatimuksena ollut täysi ilmaisuus avasi paljon eri vaihtoehtoja toteutettavalle tuotteelle, mutta käytännön testauksen jälkeen valinta oli selkeä. Sain mielestäni toteutettua hyvin selkeän ja helposti ylläpidettävän verkon kriittisten palveluiden monitorointiohjelmiston, jota on helppo jatkossa toimeksiantajan muokata omaan haluamaansa suuntaan.

OpenNMS on ensimmäinen kehitetty suuri avoimen lähdekoodin verkon monitorointiohjelmisto, ja täten sitä on osattu kehittää eteenpäin oikeaan suuntaan. Vaikka OpenNMS on alun perin suunniteltu todella suuriin verkkoympäristöihin, toimi se mielestäni erittäin hyvin myös pienemmässä ympäristössä. Yllätyin, miten kätevästi OpenNMS osaa etsiä automaattisesti palvelimista palveluita ilman, että agentteja tarvitsee palvelimiin implementoida. Dashboard-näkymä on mielestäni erittäin kätevä, koska siinä näkyy heti kuva verkosta ja sen hälytyksistä. Käyttöliittymäkin on pienen opettelun jälkeen todella kätevä ja sivuilla navigointi onnistuu kätevästi, kun melkein jokaiselta sivulta on pääsy jokaiselle sivulle. Eri valvontaryhmien luonti ja niihin eri palveluiden määrittäminenkin on tehty helpoksi. Kun ryhmät on luotu, on uusia valvottavia kohteita helppo lisätä automaattisesti monitorointiin. OpenNMS tukee myös ”Auto discovery”-ominaisuutta, mutta sitä en tässä työssä käyttänyt RGCE-verkon suuren laitemäärän vuoksi. ”Auto Discovery”-ominaisuuteen voi kuitenkin määrittää esimerkiksi IP-alueita joita skannataan, jotta järjestelmä ei skannaa koko verkkoa ja näin kuormita liikaa OpenNMS-järjestelmää.

Myös huonoja puolia OpenNMS-järjestelmästä löytyy. OpenNMS toimii käyttäen Javan versiota 7, ja on täten kohtalaisen hidasta käyttää. Lisäki huomioiden Javan tunnetut tietoturvariskit, tämän käyttö epäilytti. Lisäksi Javan uusin versio 8 ei ole vielä edes tuettu. Myös tietyt yksinkertaiset asiat on tehty OpenNMS:ssä vaikeaksi. Esimerkkinä tästä on valvontakategorioiden eli ”provisioning requisitions”-nimien muuttaminen. Tein aluksi ryhmät suomenkielisenä, jonka jälkeen sain toimeksiantajalta tiedon, että rakenne-

taan järjestelmä englanninkieliseksi. Jo tehtyjen suomenkielisten ryhmien nimien muuttaminen ei ollutkaan niin yksinkertaista kuin voisi luulla. Tähän kului aikaa, kun täytyi käsin käydä muuttamassa jokainen ryhmän nimi jokaiseen ”.xml”-tiedostoon ja SQL-tietokantaan. Olisikohan ollut OpenNMS:n kehittäjille niin vaikeaa lisätä vain ”edit”-nappula nimen viereen.

Pitkään päänvaivaa aiheutti SNMP:llä kerättävä tieto mail.netfun.fi -palvelimen sähköpostijonosta, jota en saanut millään toimimaan, vaikka tein sen täysin Internetistä löytyneen ohjeen mukaisesti, ja ”snmpwalk”-komento onnistui palvelimella. Jostain syystä OpenNMS ei ”.jrb”-tiedostoa luonut, joka viittaa siihen, että SNMP-haku ei onnistunut. Vaikka SNMP-request onnistuikin OpenNMS-palvelimelta tulostamaan jonon koon oikein.

Kokonaisuudessaan OpenNMS oli erittäin helppo asentaa, ja sen omilta wiki-sivuilta löytyi hyvät ohjeet ympäristön muokkaamiseen. Täysin valmis tuote OpenNMS ei kuitenkaan mielestäni ole. Paljon löytyy pientä viilaamista, jota täytyy käydä välillä tekemässä palvelimen komentoriviltä (CLI), ilman, että muutoksia on mahdollista tehdä graafisen Web-käyttöliittymän kautta.

Kokonaisuudessaan onnistuin työssäni mielestäni hyvin toteuttamaan vaatimuksia vastaavan tuotteen. Onnistuin myös ajallisesti hyvin, koska aloitin työn vasta Helmikuussa ja aikaa oli vain Huhtikuuhun. Opin paljon Linux-palvelimista ja etenkin SNMPv2c-protokollan asettamista haasteista. Oli mielenkiintoista nähdä, miten paljon lisää tietoa voi kaivaa palvelimista asentamalla niihin SNMP-agentin toimimaan.

Lähteet

Aitchison, R. 2011. Pro DNS and BIND 10. Viitattu 28.1.2015. http://www.jamk.fi/kirjasto, Oppaat, e-aineistot, Tietotekniikka, Books24x7.

Bejtlich, R. 2013. The Practice of Network Security Monitoring: Understanding Incident Detection and Response. Viitattu 28.1.2015. http://www.jamk.fi/kirjasto, Oppaat, e-aineistot, Tietotekniikka, Books24x7.

Blumenthal, U. & Wijnen, B. 2002. RFC 3414. User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). Internet RFC Archives.

Case, J. Harrington, D., Presuhn, R. & Wijnen, B. 2002. RFC 3412. Message Processing and Dispatching for the Simple Network Management Protocol (SNMP). Internet RFC Archives.

Cepeda, O. 2000. Beyond DHCP. Up and Running with DHCP. Viitattu 28.1.2015. http://www.jamk.fi/kirjasto, Oppaat, e-aineistot, Tietotekniikka, Books24x7.

Downin, M. 2013. The Importance of Network Monitoring. Viitattu 28.1.2015. http://www.animate.com/the-importance-of-network-monitoring/

Dragich, L. 2012. Event Management: Reactive, Proactive or Predictive? Viitattu 10.3.2015. http://www.apmdigest.com/event-management-reactive-proactive-or-predictive

Farrel, A. 2009. Network Management: Know It All. Viitattu 10.3.2015. http://www.jamk.fi/kirjasto, Oppaat, e-aineistot, Tietotekniikka, Books24x7.

Harrington, D.,Presuhn, R. & Wijnen, B. 2002. RFC 3411. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. Internet RFC Archives.

Jyvsectec. 2014. Jyvsectecin kotisivut. Viitattu 28.1.2015. http://jyvsectec.fi/jyvsectec/

Kundu, D. & Lavlu S. 2009. Comparison of SNMP Versions and Security. Viitattu 28.1.2015. http://www.jamk.fi/kirjasto, Oppaat, e-aineistot, Tietotekniikka, Books24x7.

Leskiw, A. 2015. OIDs and MIBs. Viitattu 18.2.2015. http://www.networkmanagementsoftware.com/snmp-tutorial-part-2-rounding-out-the-basics

Levi, D., Meyer, P. & Stewart, B. 2002. RFC 3413. Simple Network Management Protocol (SNMP) Applications. Internet RFC Archives.

OpenNMS. 2014. Installation:Yum. Viitattu 3.2.2015. http://www.opennms.org/wiki/Installation:Yum

Postel, J. 1980. RFC 768. User Datagram Protocol. Internet RFC Archives.

RFC 791. 1981. Internet Protocol. Internet RFC Archives. University of Southern California.

RFC 793. 1981. Transmission Control Protocol. Internet RFC Archives. University of Southern California.

RGCE. 2014. Jyvsectecin kotisivut. Viitattu 28.1.2015. http://jyvsectec.fi/rgce/

Sathyan, J. 2010. Overview of SNMP (Simple Network Management Protocol). Viitattu 28.1.2015. http://www.jamk.fi/kirjasto, Oppaat, e-aineistot, Tietotekniikka, Books24x7.

Sosinsky, B. 2009. Networkin Bible. Viitattu 28.1.2015. http://www.jamk.fi/kirjasto, Oppaat, e-aineistot, Tietotekniikka, Books24x7.

Tarus. 2009. The Many Uses of Net-SNMP. Viitattu 3.3.2015. http://www.adventuresinoss.com/?p=1147

Tutustu ja menesty. 2014. Jyväskylän ammattikorkeakoulun kotisivut. Viitattu 28.1.2015. http://www.jamk.fi/fi/Tietoa-JAMKista/Tutustu-JAMKiin/