Saimaan ammattikorkeakoulu
Tekniikka Imatra
Tietotekniikka

Mika Tiainen

# Etelä-Karjalan koulutuskuntayhtymän tietoverkon suorituskyvyn analysointi

Opinnäytetyö 2012

Tiivistelmä
Mika Tiainen
Etelä-Karjalan koulutuskuntayhtymän tietoverkon suorituskyvyn analysointi
Saimaan ammattikorkeakoulu
Tekniikka Imatra
Tietotekniikan koulutusohjelma
Opinnäytetyö, 2012
Ohjaajat: tuntiopettaja Pasi Juvonen, Saimaan ammattikorkeakoulu
IT-koordinaattori Simo Horsmanheimo, Etelä-Karjalan koulutuskuntayhtymä

Asiakkaana opinnäytetyössä oli Etelä-Karjalan koulutuskuntayhtymä. Opinnäytetyön tavoitteena oli tutkia tietoverkon suorituskykyä erilaisilla mittareilla ja selvittää mahdollisia syitä Windows-käyttöjärjestelmän kirjautumisen hitauteen.

Työ aloitettiin kehittämällä työkalu automaattiseen verkon topologiakuvan luomiseen. Tämän jälkeen valittiin yksittäisiä laitteita tarkempiin mittauksiin. Työn tuloksena syntyi joukko työkaluja verkon toiminnan seuraukseen, sekä ongelmanratkaisuun. Kirjautumisongelmaan ratkaisua ei tietoverkon osalta kuitenkaan löytynyt.

Asiasanat: tietoliikenne, kytkin, ethernet, SNMP

Abstract
Mika Tiainen
Performance Analysis of Etelä-karjalan koulutuskuntayhtymä Data Network
Saimaa University of Applied Sciences
Technology Imatra
Degree Program in Information Technology
Bachelor's Thesis, 2012
Instructors: Lecturer Pasi Juvonen, Saimaa University of Applied Sciences
IT-coordinator Simo Horsmanheimo, Etelä-Karjalan koulutuskuntayhtymä

The client in this thesis was Etelä-karjalan koulutuskuntayhtymä. Purpose of the thesis was to investigate the performance of their data network with different meters, and find out possible causes for slowness of the login process in Windows operating system.

The work started with development of a tool for creating automatic topology map of the network. After this a few devices were chosen for more in-depth analysis. As a result of the thesis several tools for monitoring and analyzing network performance were developed. Solution for the login problem was however not found.

Keywords: telecommunication, switch, ethernet, SNMP

**Sisältö**

# 1 Johdanto

Tämän opinnäytetyön tavoitteena on tutkia Etelä-Karjalan koulutuskuntayhtymän tietoverkon suorituskykyä ja tiedonsiirtokapasiteetin riittävyyttä. Lähtökohtana on käyttäjän tietokoneelle kirjautumisen hitaus Windows-käyttöjärjestelmässä.

Työn aikana on tarkoitus luoda ajantasainen ja automaattisesti päivittyvä kuva verkon topologiasta, sekä mitata verkon ja laitteiden suorituskykyä eri pisteistä. Kaikki työn toteutukseen käytetyt valmiit ohjelmistot ovat avoimen lähdekoodin ohjelmistoja, lisäksi näiden ympärille toteutetaan joitakin omia työkaluja.

Suuri osa työn tuloksista on automaattisesti päivittyviä lokeja ja kuvaajia, jotka julkaistaan WWW-sivuilla. Näin verkon tilaa pystyy helposti seuraamaan lähes reaaliaikaisesti sekä myös pidemmällä aikavälillä.

# 2 Tietoliikenneverkot

Nykyaikainen tietoliikenneverkko koostuu suuresta määrästä erilaisia, eri tasoilla toimivia, tekniikoita ja protokollia. Tässä luvussa kerrotaan lyhyesti nykyaikaisen Ethernet-verkon toiminnasta ja siihen liittyvistä tekniikoista, jotka olennaisimmin liittyivät työn toteutukseen.

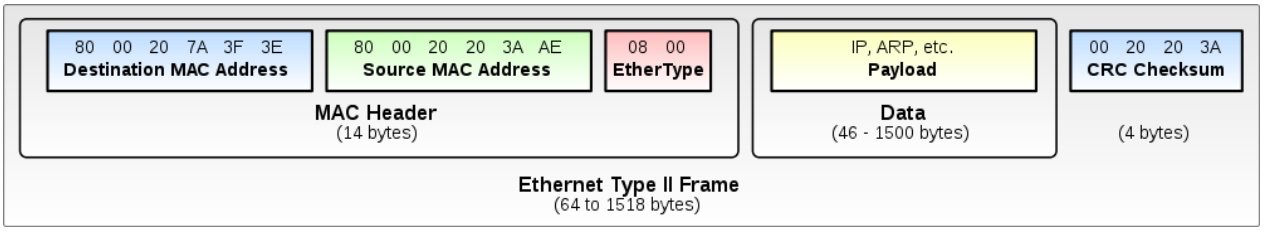
Työssä tutkittiin verkkoa pääasiassa Ethernet-tasolla. Ylemmän tason protokollien, kuten TCP ja UDP, toiminnan analysointi olisi vaatinut verkkoliikenteen kaappaamista talteen, tähän ei työn puitteissa ollut mahdollisuutta.

## 2.1 Ethernet

Ethernet on jo 1970-luvulla kehitetty pakettipohjainen verkko. IEEE 802.3 on kokoelma standardeja, jotka määrittelevät Ethernetin toiminnan. Liikennöinti Ethernet-verkossa perustuu MAC-osoitteisiin. Jokaisella verkkolaitteella on oma

uniikki osoite. Alun perin verkon toimintaan vaikutti suoraan kytkettyjen laitteiden määrä, koska toistimet (hubit) lähettivät jokaisen paketin eteenpäin jokaiselle laitteelle, lisäksi toistimet pystyivät liikennöimään vain yhteen suuntaan kerrallaan (half-duplex). Nykyaikaisessa kytketyssä verkossa laitteiden lukumäärä ei suoraan vaikuta suorituskykyyn. Kytkin tarkkailee liikennettä ja oppii näin mistä portista kukin MAC-osoite löytyy. Kun kytkin on oppinut MAC-osoitteen portin, ei se enää lähetä tähän osoitteeseen osoitettua liikennettä muihin portteihin. (Doherty ym. 2008.)

Ethernet-verkossa paketteja kutsutaan kehyksiksi. Normaalisti kehyksen koko on 64–1518 tavua (Kuva 1).



Kuva 1. Ethernet-kehys

Kehyksestä 1500 tavua on varsinaista hyötydataa, loput otsikko- ja tarkistussummakenttiä.

## 2.2 Spanning tree protocol

Spanning tree protocol (STP) on Ethernetin rinnalla OSI-mallin tasolla kaksi toimiva protokolla, jonka tarkoituksena on estää silmukoiden syntyminen verkkoon sekä mahdollistaa varayhteyksien rakentaminen. Silmukka Ethernet-verkossa, jossa STP ei ole käytössä, aiheuttaa yleislähetysmyrskyn (broadcast storm). Myrsky aiheutuu, koska yleislähetyspaketit lähetetään aina jokaiseen kytkinporttiin, paitsi siihen josta se vastaanotettiin Jos jokin portti on kytketty suoraan takaisin kytkimeen, kiertää paketti sieltä takaisin ja lähetetään välittömästi uudestaan. Tämä kertautuu loputtomiin ja generoi niin paljon liikennettä kuin kyseessä oleva laite pystyy lähettämään ja mahdollisesti kaataa tai jumiuttaa verkkolaitteita. (Doherty ym. 2008.)

STP:n toimintaperiaate on yksinkertainen:

1. Yksi osallistuvista laitteista valitaan juureksi (root bridge).
2. Jokaiselle laitteiden väliselle linkille asetetaan hinta (cost), joka useimmiten oletuksena valitaan linkin nopeuden mukaan: mitä nopeampi linkki, sitä pienempi hinta.
3. Jokainen laite laskee portin, josta on ”halvin” reitti juureen. Muut osallistuviin laitteisiin kytketyt portit tiputetaan pois käytöstä.
4. Jos jokin osallistuvien laitteiden välisistä yhteyksistä katkeaa, laskevat kaikki muut laitteet uuden ”halvimman” portin, ottaen mahdollisesti käyttöön, jonkin aiemmin käytöstä poistetun varayhteyden.

STP ei perusversiossaan mahdollista kuormantasausta, vaan käytössä on ainoastaan yksi linkki kerrallaan. Kehittyneemmät versiot kuten PVST (Per-VLAN Spanning Tree) ja MSTP (Multiple Spanning Tree Protocol) mahdollistavat useamman rinnakkaisen puun muodostamisen, jolloin jokaiselle VLAN:lle voidaan laskea oma optimaalinen reitti.

## 2.3 VLAN

Virtuaalilähiverkko. Tekniika (Virtual LAN, VLAN, IEEE 802.1Q) mahdollistaa fyysisen verkon jakamisen useisiin loogisiin verkkoihin, jotka toimivat täysin erillään näkemättä toistensa liikennettä. Ilman virtuaalilähiverkkoja kaikki yleislähetyspaketit päätyvät jokaiselle verkkoon kytketylle laitteelle, riippumatta siitä ovatko ne esimerkiksi samassa IP-aliverkossa.

VLAN:ien toiminta perustuu Ethernet-kehykseen lisättävään neljän tavun otsikkoon, joka sisältään VLAN-numeron (tagin). Tagin perusteella kytkimet tietävät mihin VLAN:n paketti tulee lähettää. Standardi mahdollistaa 4094 VLAN:ia. Käytännössä yksittäisessä laitteessa käytössä oleva maksimimäärä on huomattavasti pienempi.

# 3 Verkon toiminnan ja suorituskyvyn seuranta

Verkon toimintaa voidaan mitata useilla tavoilla, useilla eri tasoilla. Tässä työssä on keskitytty tutkimaan kytkinten suorituskykyä, mahdollisia virheitä tiedonsiirrossa sekä verkkoviivettä (latenssia) IP-tasolla.

Ennen varsinaisen suorituskyvyn seurantaa täytyi verkon topologiasta saada selkeä kuva. Tähän tarkoitukseen kehitettiin oma ohjelma, joka hakee topologiatiedot suoraan verkkolaitteista.

## 3.1 Kytkimen suorituskyky

Kytkinten suorituskyky ilmoitetaan yleensä paketteina sekunnissa (pps), itse siirretyn tiedon määrä riippuu siis pakettien koosta. Ethernet-kehyksen minimikoko on 72 tavua, jonka lisäksi kehysten lähetysväli (interframe gap) on 12 tavua, tästä saadaan minimiksi 84 tavua. Maksimikoko on 1526 + 12 = 1538 tavua. Jos käytetään VLAN:eja, kasvaa sekä minimi että maksimi vielä VLAN-tagille varatut neljä tavua (88 ja 1542). Kaavassa 3.1 on laskettu gigabit ethernetin teoreettinen maksimipakettimäärä. (Cisco Systems 2010.)

$$\frac{1\,000\,000\,000\ \frac{b}{s}}{84\ t \times 8\ \frac{b}{t}} = 1\,488\,096\ \frac{p}{s} \qquad (3.1)$$

Tästä saatava tulos tarkoittaisi siis tiedonsiirtoa 1Gb/s, 84 tavun paloissa. Koska gigabit ethernet on aina Full Duplex, voidaan luku vielä kertoa kahdella, jos oletetaan että tietoa siirretään molempiin suuntiin yhtä aikaa (kaava 3.2).

$$1\,488\,096\ \frac{p}{s} \times 2 = 2\,976\,192\ \frac{p}{s} \qquad (3.2)$$

Vanhimmat tutkitussa verkossa käytössä olevat kytkimet ovat HP:n Procurve 2524 –mallia. HP ilmoittaa kyseisen kytkimen suorituskyvyksi 6,6 miljoonaa pakettia sekunnissa ja tiedonsiirtokapasiteetiksi 9,6Gb/s. (Hewlett-Packard Deve-

lopment Company 2010) Kytkimissä on 24 Fast Ethernet –porttia ja kaksi paikkaa, joihin voidaan asettaa myös Gigabit Ethernet –moduulit.

Kaavassa 3.3 on laskettu siirrettävän tiedon määrä silloin, jos kaikki portit liikennöisivät molempiin suuntiin täydellä vauhdilla.

$$\left(24 \times 100\ \frac{Mb}{s} + 2 \times 1\ \frac{Gb}{s}\right) \times 2 = 8{,}8\ \frac{Gb}{s} \tag{3.3}$$

Kaavasta 3.3 nähdään että jos pakettia sekunnissa suorituskyky riittää, pystyy kytkin siirtämään tietoa kaikkien porttien välillä täydellä vauhdilla.

Siirrettävän tiedon määrä minimikokoisilla paketeilla valmistajan ilmoittamalla maksimi sekuntivauhdilla selviää kaavan 3.4 avulla.

$$6\ 600\ 000\ \frac{p}{s} \times 84\ t\ \times 8\ \frac{b}{t} = 4{,}4\ \frac{Gb}{s} \tag{3.4}$$

Kaavasta 3.4 nähdään, että teoriassa kytkimen suorituskyky voisi jossain tilanteessa olla rajoittava tekijä. Käytännössä tutkitussa verkossa ei normaalitilanteessa liikkunut sellaisia tietomääriä, joissa nämä rajat tulisivat vastaan.

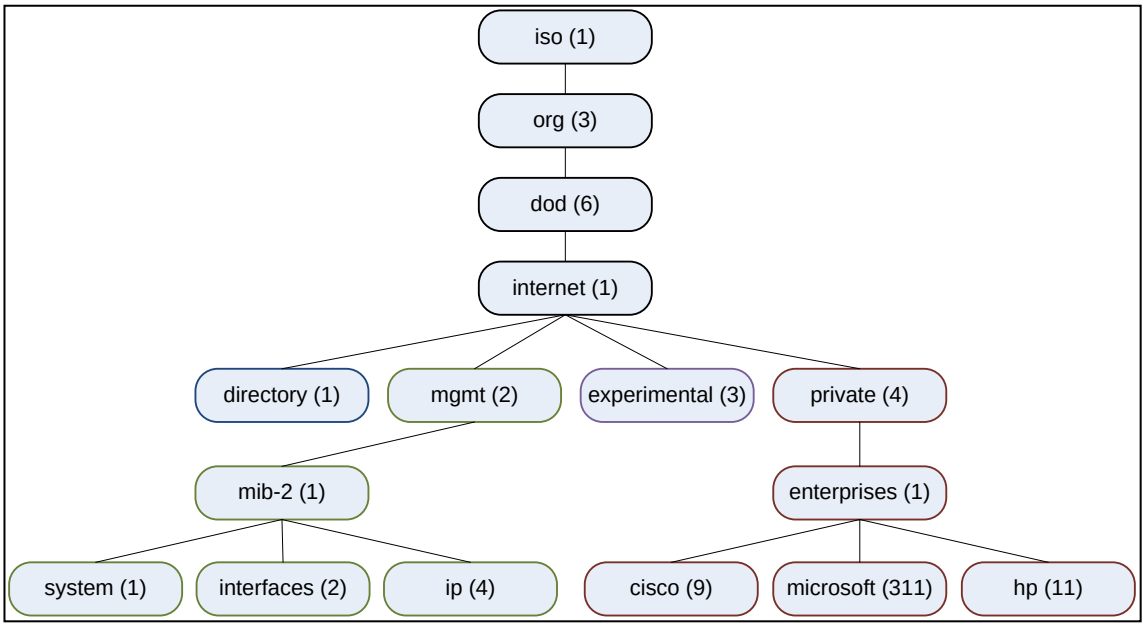
Mitään omia erillisiä suorituskykymittauksia ei suoritettu jo senkin takia, että yllä esitettyihin tiedonsiirtomääriin pääseminen, ilman erillistä kytkinten testaamiseen tarkoitettua mittalaitteistoa ei helposti onnistu.

## 3.2 SNMP

SNMP (Simple Network Management Protocol) on verkkolaitteiden valvontaan ja etähallintaan kehitetty protokolla. SNMP:stä on julkaistu kolme eri versiota, joista nykyisin käytössä ovat kaksi ja kolme. Versio kolme lisäsi lähinnä tietoturvaominaisuuksia, kuten liikenteen salauksen. Protokolla perustuu yksittäisiin muuttujiin, jotka on organisoitu puumaiseen hierarkiaan. Hierarkia ja tietueiden sisältö kuvataan MIB-tiedostojen avulla. Yksittäistä tietuetta kutsutaan nimella OID (Object identifier).

Hierarkia on jaettu standardoituun ja valmistajakohtaisiin. Standardoitujen tietueiden sisältö on sama laitteesta ja valmistajasta riippumatta. Esimerkiksi `.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0` joka sisältää vapaamuotoisen kuvauksen laitteesta.

Valmistajakohtaiselle tiedolle on varattu enterprises-hierarkia, josta valmistaja saa oman haaran. Jokainen laitevalmistaja voi vapaasti määritellä oman haaransa alla sijaitsevat tietueet ja julkaista rakenteen kuvaavat MIB-tiedostot. Kuvassa 2 on kuvattu pieni osa SNMP-hierarkiaa.



Kuva 2. SNMP-hierarkia

Yleisin SNMP:n käyttötarkoitus verkkolaitteiden yhteydessä on liikennemäärien kerääminen. Se on kuitenkin vain pieni osa siitä tiedosta, joka SNMP:n avulla on laitteista saatavilla. Lähes kaikki tieto, jota tämän työn yhteydessä syntyneet skriptit käyttävät, on kerätty SNMP:tä käyttäen.

## 3.3 LLDP

LLDP (Link Layer Discovery Protocol) on valmistajariippumaton standardoitu protokolla, jonka avulla verkkolaitteet "mainostavat" tietoja itsestään toisille verkkolaitteille. Aiemmin laitevalmistajilla oli omia vastaavia protokollia, tunnetuimpana CDP (Cisco Discovery Protocol), jota nimestä huolimatta tukivat myös

useiden muiden valmistajien laitteet. Uudemmat Ciscon laitteet käyttävät molempia rinnakkain.

Käytännössä LLDP on ethernet-kehys, jonka laitteet lähettävät tietyin väliajoin kaikkiin portteihin. Kehys sisältää tietoja lähettävästä laitteesta (nimi, portti, IP-osoite, yms.). Vastaanottaessaan LLDP-kehyksen laite tallentaa sen sisältämät tiedot kehyksessä määritellyksi ajaksi (HP-kytkimien lähettämissä paketeissa oletuksena 2 minuuttia).

Tiedot jotka HP-kytkin on naapureista kerännyt, kytkimen hallinnasta katsottuna. Selviävät seuraavista taulukoista 1 ja 2.

```
XXX# sh lldp info remote-device

 LLDP Remote Devices Information

  LocalPort | ChassisId                 PortId PortDescr SysName
  --------- + ------------------------- ------ --------- ----------------------
  22        | 00 16 b9 c7 a4 90         1      Port #1   XXX
  48        | 00 18 71 2b bb 00         50     50        XXX
  49        | 00 14 38 36 dd 00         50     50        XXX
  50        | 00 1f fe 17 e4 80         49     49        XXX
  51        | 00 1f fe 15 c7 80         52     52        XXX
  52        | 00 1f fe b3 d9 c0         28     28        XXX
```

Taulukko 1. Kytkimen naapurit

```
XXX# sh lldp info remote-device 52

 LLDP Remote Device Information Detail

  Local Port   : 52
  ChassisType  : mac-address
  ChassisId    : 00 1f fe b3 d9 c0
  PortType     : local
  PortId       : 28
  SysName      : XXX
  System Descr : ProCurve J9085A Switch 2610-24, revision R.11.30, ROM R.1...
  PortDescr    : 28

  System Capabilities Supported  : bridge, router
  System Capabilities Enabled    : bridge

  Remote Management Address
     Type    : ipv4
     Address : XXX
```

Taulukko 2. Kytkimen yksittäisen naapurin tiedot

LLDP:n avulla kerätyistä tiedoista pystyy selvittämään verkon topologian. Tätä ominaisuutta on käytetty hyväksi työn aikana luoduissa verkkokuvissa.

# 4 Ohjelmistot

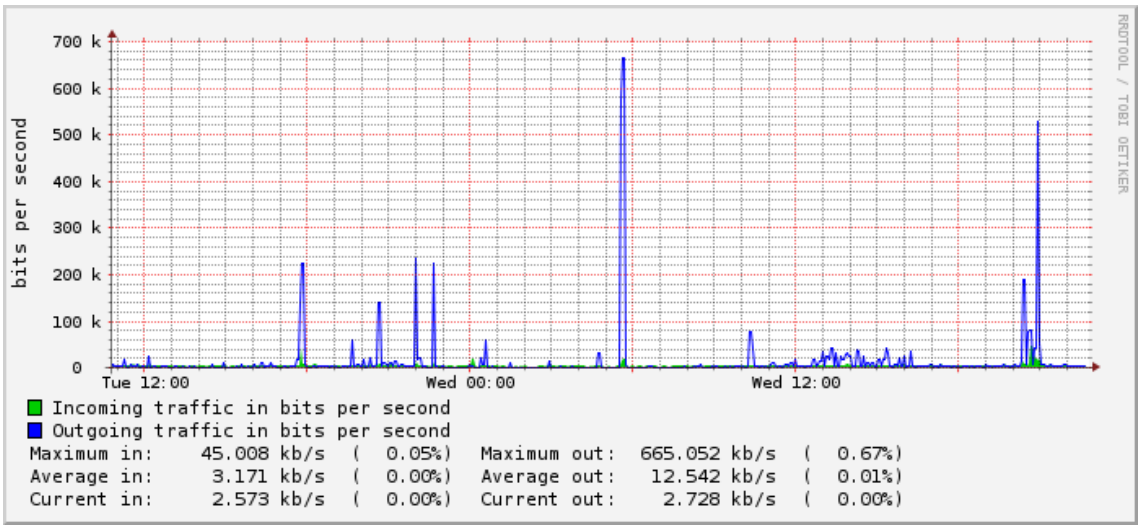
Työn toteutuksessa käytettiin apuna lukuisia valmiita avoimen lähdekoodin ohjelmistoja sekä toteutettiin joitakin omia työn helpottamiseksi. Tässä luvussa esitellään lyhyesti tärkeimmät näistä.

Erikseen mainittujen lisäksi työn ohessa syntyi esimerkiksi työkalu konfiguraatiomuutosten tekemiseen useille HP-kytkimille kerralla (samat komennot ajetaan jokaisella laitteella), sekä useita PHP:lla toteutettuja WWW-sivuja erilaisten kerättyjen tietojen, kuten lokitapahtumien, esittämiseen.

## 4.1 MRTG

The Multi Router Traffic Grapher (http://oss.oetiker.ch/mrtg/) on, kuten nimestä voi päätellä, verkon liikennemäärien tallennukseen ja graafien julkaisuun tarkoitettu ohjelmisto. Käytännössä sillä voi tallentaa myös mitä tahansa muita numeerisesti mitattavia arvoja.

Tässä työssä tarkkailtiin liikennemäärien lisäksi myös kytkinten prosessorikuormaa. Kerätyt tiedot julkaistaan graafisina kuvaajina WWW-sivulla. Esimerkki kuvassa 3.

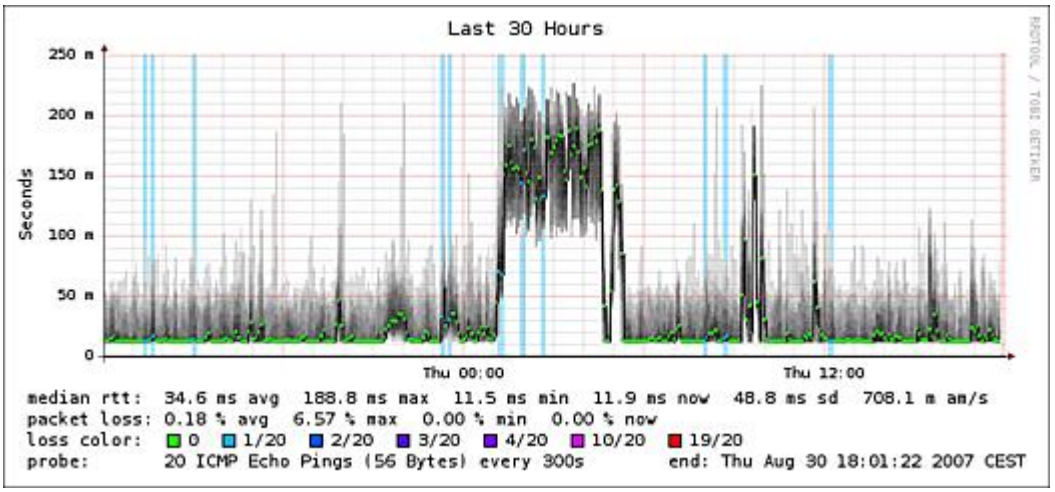


Kuva 3. Esimerkki MRTG:n tuottamasta kuvaajasta

Kuvaajista pystyy nopeasti havaitsemaan normaalista poikkeavia liikennemääriä. Tarvittaessa niiden avulla pystyy myös luomaan hälytyksiä valvontajärjestelmään.

## 4.2 SmokePing

SmokePing (http://oss.oetiker.ch/smokeping/) mittaa verkon latenssia. Yleisimmin mittaukseen käytetään yksinkertaisesti Ping-työkalua, joka lähettää ICMP echo request-paketin, johon vastapuoli vastaa echo reply-paketilla. Paketin lähetyksestä vastauspaketin saapumiseen kulunut aika on vasteaika (RTT, Round-Trip Time). Vasteajan lisäksi mitataan myös mahdollisia hukkuneita paketteja (Packet loss), mikä on yleensä merkki joko verkon tai vastapään laitteen ylikuormituksesta. Kuvassa 4 on esimerkki vasteaikakuvaajasta.




Kuva 4 Esimerkki SmokePing:n tuottamasta kuvaajasta

Halutessa voi myös käyttää erilaisia sovellustason mittareita, eli mitata esimerkiksi HTTP-palvelimen vasteaikaa, lähettämällä HTTP-kyselyn. Myös näistä pystyy tarvittaessa luomaan automaattisia hälytyksiä.

## 4.3 Netmap

Jo työn alkuvaiheessa kävi selväksi, että tutkimuksen kohteena olleen verkon topologiasta oli tarve saada ajan tasalla oleva kokonaiskuva. Tätä varten tehtiin oma ohjelmansa, joka kerää topologiatiedot suoraan verkkolaitteista ja luo niiden pohjalta verkkokuvan Graphviz-ohjelman avulla (http://www.graphviz.org). Ohjelma kerää myös runsaasti muuta tietoa laitteista, kuten malli- ja sarjanumerot, ohjelmistoversion, muistinmäärän sekä porttimäärän.

Netmap:n toiminta perustuu verkkolaitteiden LLDP:n avulla keräämiin tietoihin. Ohjelmalle annetaan yhden verkkolaitteen IP-osoite, sekä SNMP-community, jota verkossa käytetään. Tämän jälkeen ohjelma noutaa annetun laitteen tiedot, joihin sisältyy tähän kytkettyjen muiden laitteiden IP-osoitteet. Kyselyä jatketaan rekursiivisesti eteenpäin, kunnes kaikki verkosta löytyvät laitteet on läpikäyty.

Kaikki ohjelman keräämät tiedot tallennetaan YAML-muodossa tiedostoon, jota muut sovellukset voivat käyttää. Laitteiden välisistä linkkitiedoista muodostaan Graphviz-ohjelmaa varten oma kuvaustiedosto, josta Graphviz muodostaa varsinaisen verkkokuvan.

# 5 Työn kulku

Työn lähtötilanteessa tutkittavan verkon suorituskyvystä ei kerätty minkäänlaisia pysyviä tilastoja, ainoastaan hetkellisiä liikennemääriä tutkittiin tarvittaessa suoraan yksittäisten laitteiden hallintaliittymän kautta. Tällä tavoin ei kuitenkaan pystytty tekemään minkäänlaista ongelmien analysointia jälkikäteen.

Myöskään kattavaa ajantasaista kokonaiskuvaa verkon topologiasta ei ollut olemassa, joten työn aluksi lähdettiin kehittämään järjestelmää jolla verkon topologiasta saataisiin generoitua automaattisesti päivittyvä kuva. Tämä helpotti huomattavasti verkon rakenteen hahmottamista.

## 5.1 Verkon topologia

Kaikki verkossa käytössä olevat laitteet tukivat jo lähtötilanteessa LLDP-protokollaa, ja se on myös oletuksena kaikissa porteissa käytössä, joten muutoksia laitteiden konfiguraatioon ei tässä vaiheessa tarvinnut tehdä. Myöhemmin tosin ilmeni, että yhden käytössä olevan laitemallin vanhassa ohjelmistoversiossa oli virhe, jonka takia LLDP-viestejä ei vastaanotettu porteista, joissa oli määriteltynä ainoastaan tunnuksellisia virtuaalilähiverkkoja (tagged VLAN). Tämä ongelma ratkesi päivittämällä kyseisten laitteiden ohjelmisto.

SNMP, jota tietojen keräämiseen laitteilta käytettiin, on periaatteessa varsin yksinkertainen. Yksi tietue (OID) sisältää yhden arvon, joka voi olla esimerkiksi luku- tai tekstimuotoinen. Tämä kuitenkin aiheuttaa sen, että saadakseen haettua esimerkiksi yksittäisen kytkinportin kohdalta kerätyt LLDP-tiedot voit joutua hakemaan kymmeniä yksittäisiä tietueita. Perl-ohjelmointikielelle, jolla Netmap toteutettiin, on onneksi saatavilla SNMP::Info –niminen kirjasto (http://snmp-info.sourceforge.net), joka toteuttaa hankalimman osuuden, eli tiedon keräämisen, ja tarjoaa käytettäväksi valmiit tietorakenteet. Lisäksi kirjasto abstraktoi valmistajakohtaiset erot tiedonhaussa. Tällä hetkellä tästä ei käytännössä ole hyötyä, koska kaikki verkkolaitteet ovat saman valmistajan, mutta mikäli verkkoon tulee muiden valmistajien laitteita, pitäisi tiedonhaun niistä onnistua yhtälailla, mikäli vain vastaavat ominaisuudet laitteista löytyvät.

Periaatteena oli kerätä kerralla mahdollisimman paljon tietoa riippumatta siitä, onko niille välttämättä välitöntä tarvetta. Tärkeimpänä ovat topologiatiedot, eli laitekohtaisesti portit, ja niiden takaa löytyvät laitteet. Muita kerättäviä tietoja ovat esimerkiksi: laitteen valmistaja, nimi, sijainti, malli, sarjanumero, muistin määrä, porttien lukumäärä, ohjelmistoversio sekä virtuaalilähiverkkojen nimet ja numerot.

Kerätyt tiedot tallennetaan kahdessa eli muodossa:

- YAML (Yet Another Markup Language, http://www.yaml.org) –muotoiseen tiedostoon, josta tiedot saadaan luettua eri ohjelmointikielillä helposti käsiteltäväksi tietorakenteeksi.
- Topologiatiedot erikseen toiseen tiedostoon Graphviz-ohjelman käyttämässä muodossa, josta muodostetaan verkkokuva.

Taulukossa 3 esimerkki yhden laitteen tallennettavista tiedoista YAML-muodossa:

```
ips:
  XXX.XXX.XXX.XXX:
    contact: ''
    ipforward: 0
    location: 'XXX'
    max_vlans: 253
    mem_total: 15973944
    model: 2626-CR
    name: XXX
    ports: 26
    serial: 'XXX'
    vendor: hp
    ver_os: H.10.83
    ver_rom: H.08.02
    vlans:
      1: DEFAULT_VLAN
      20: XXX
links:
  XXX.XXX.XXX:XXX:
    25:
      duplex: full
      name: 'Port 25'
      neighbor: XXX.XXX.XXX.XXX
      neighbor_port: 8
      speed: '1.0 Gbps'
      type: gigabitEthernetT
      untagged_vlan: 1
      vlan_ids:
        - 1
```

Taulukko 3. Esimerkki YAML-muotoisesta tietorakenteesta

Näiden tietojen avulla tehtiin muun muassa WWW-sivu, jossa on listattu kaikki verkosta löytyneet laitteet ja niiden tiedot (malli, sarjanumero, ohjelmistoversio, jne.).

Taulukossa 4 esimerkki Graphviz-määritystiedoston osa.

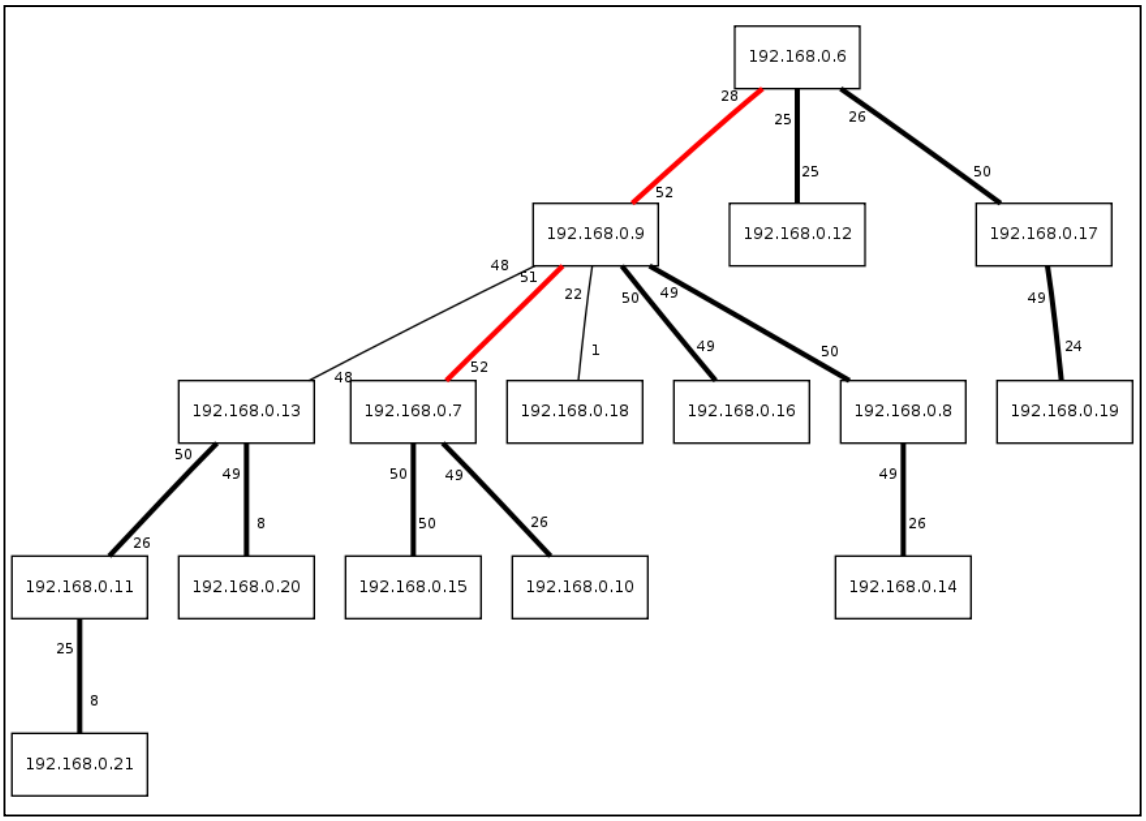
```
graph verkko {
graph [ ratio = 0.71, overlap = prism, esep = "+1", splines = true
];
node [ shape = box, fontname = "sans-serif", fontsize = 10 ];
edge [ labeldistance = 2, fontname = "sans-serif", fontsize = 8 ];

"XXX.XXX.XXX.XXX" [
    label = "XXX.XXX.XXX.XXX\nXXX\nXXX"
];
"YYY.YYY.YYY.YYY" [
    label = "YYY.YYY.YYY.YYY\nYYY\nYYY"
];

"XXX.XXX.XXX.XXX" -- "YYY.YYY.YYY.YYY" [
    taillabel = "15", headlabel = "8",
    color = black, penwidth = 1
];
"YYY.YYY.YYY.YYY" -- "ZZZ.ZZZ.ZZZ.ZZZ" [
    taillabel = "50", headlabel = "26",
    color = black, penwidth = 3
];
}
```

Taulukko 4. Esimerkki Graphviz-määritystiedostosta

Näistä tiedostoista syntyy Graphviz-ohjelmalla kuvassa 5 esitetyn kaltainen topologiakuva.



Kuva 5. Esimerkki Graphviz:llä tehdystä verkkotopologiakuvasta

Värit kuvaavat linkin tyyppiä (musta kuparikaapeli, punainen monimuoto valokuitu). Viivan paksuus kuvaa linkin nopeutta (100 Mbps / 1 Gbps). Todellisissa kuvissa on IP-osoitteiden lisäksi laitteen nimi ja sijainti.

## 5.2 Verkon toiminnan seuranta

Sen jälkeen kun verkon topologiasta oli saatu kokonaiskuva, valittiin joukko laitteita, joiden liikenteen määrää ruvettiin seuraamaan tarkemmin. Tietojen keräämiseen käytettiin MRTG-ohjelmistoa. Ohjelmalle tehtiin valmis pohja, jonka perusteella konfiguraatiotiedosto luotiin automaattisesti listaamalla niiden laitteiden IP-osoitteet, joista tiedot haluttiin.

Porttikohtaisten liikennemäärien lisäksi laitteista tallennettiin prosessorikuorma, jotta nähtäisiin, kasvaako se missään tilanteissa lähelle maksimia.

SmokePing-ohjelmistoa käytettiin verkkoviiveen mittaamiseen esimerkiksi levypalvelimille. Kasvaneet viiveet voivat kertoa ongelmista verkossa tai palvelimen

ylikuormituksesta. Sovellustasolla seurantaa ei tehty, vaan yksinkertaisesti pingaamalla palvelimia.

## 5.3 Lokit

Kaikilta kytkimiltä kerättiin lokit keskitetysti yhteen paikka. Tavoitteena oli automatisoida lokien analysointi mahdollisimman pitkälle, niin että pystyttäisiin esimerkiksi lähettämään automaattisia hälytyksiä tapahtumien perusteella. Tältä osin työ jäi kuitenkin kesken, johtuen lokien analysoinnin hankaluudesta. Esimerkiksi virheiden määrää pitäisi seurata porttikohtaisesti pidemmällä aikavälillä, koska yksittäisiä virhemerkintöjä tulee siellä täällä useita kertoja tunnissa.

Runkoverkon seurantaa varten tehtiin erillinen WWW-sivu, joka suodattaa näkyviin ainoastaan runkoverkkoon kuuluvien porttien (portit haetaan Netmap:n keräämistä tiedoista) merkinnät. Näissä ei pitäisi virhemerkintöjä tulla ollenkaan.

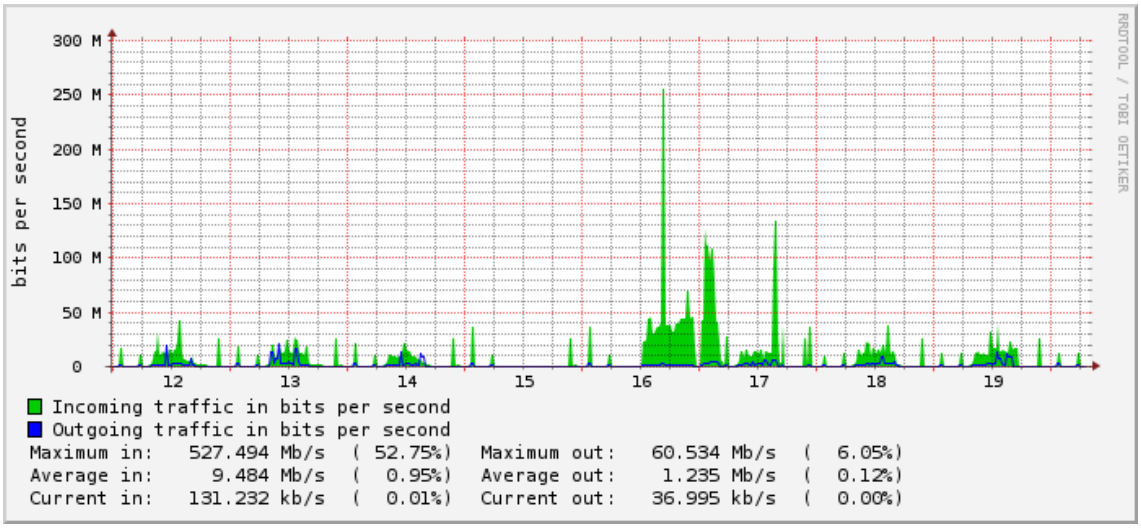
# 6 Tulokset

Lähtökohtana oli löytää verkosta mahdollisia ongelmia, jotka voisivat aiheuttaa hitauden käyttäjän sisäänkirjautumisessa Windows-työasemalle. Oletuksena oli, että hitaus johtuu kelluvan käyttäjäprofiilin latauksesta levypalvelimelta ja sen aiheuttamasta verkkokuormasta.

Jo työn alussa ajatus siitä, että profiilien lataus aiheuttaisi niin paljon liikennettä, että se tukkisi modernin verkon, vaikutti mahdottomalta. Profiilin maksimikoko on rajoitettu varsin pieneksi, joten edes suuren määrän yhtä aikaa kirjautuvia käyttäjiä ei pitäisi aiheuttaa kuin korkeintaan hetkellisiä piikkejä liikenteessä.

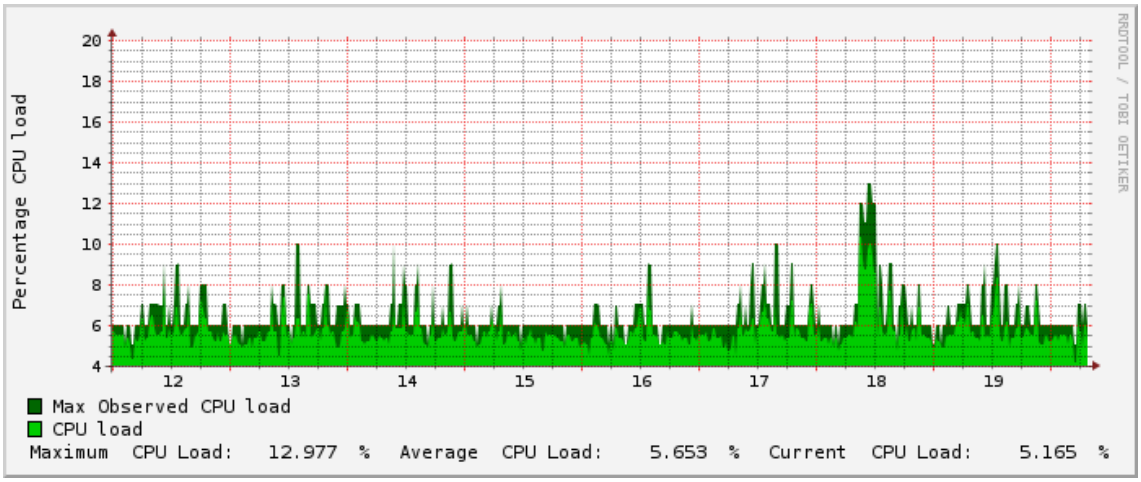
Käytännössä seurannan aikana liikennemäärät eivät edes hetkellisesti nousseet linkkien maksimiin. Ainoa tällainen tilanne oli tutkimuksen alussa, mikä johtui silmukasta verkossa ja aiheutti koko verkon täydellisen jumiutumisen. Tämän jälkeen koko verkossa otettiin käyttöön spanning tree-protokolla eikä vastaavaa enää ilmennyt. Keskimäärin liikennemäärät olivat alle puolet linkkien kapasitee-

tista. Hetkellistä suurempaa liikennettä aiheuttivat muun muassa varmuuskopioiden ottaminen, mutta tämä tapahtuu viikonloppu- tai yöaikaan, jolloin sillä ei ole vaikutusta normaaliin käyttöön.



Kuva 6. Liikennemäärä viikon ajalta

Kuvasta 6 näkyy, ettei liikenne missään vaiheessa viikkoa nouse lähellekään linkin maksimikapasiteettia, normaalisti pysytään alle kymmenessä prosentissa maksimista. Kuvassa 7 kuvataan saman laitteen prosessorikuorma samalta ajanjaksolta.

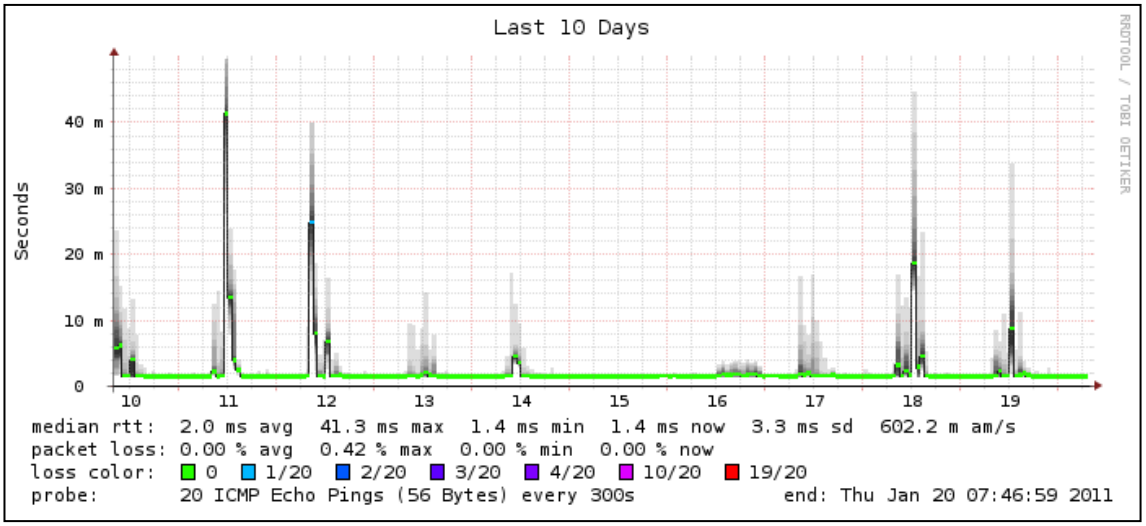


Kuva 7. Kytkimen prosessorikuorma viikon ajalta

Kuten kuvasta 7 näkyy, eivät nämäkään arvot nouse kuin hetkellisesti yli kymmeneen prosenttiin. Käytännössä liikennemäärä ei edes suoraan vaikuta prosessorikuormitukseen, enemmän vaikuttaa liikenteen tyyppi, esimerkiksi pie-

nemmistä paketeista koostuva liikenne kuormittaa enemmän kuin vastaava liikenne, joka koostuu suuremmista paketeista. Lisäksi tietenkin esimerkiksi kytkimen palomuuriominaisuuksien tai liikenteen priorisoinnin käyttö vaikuttaa prosessorikuormaan.

Myöskään SmokePingin verkkoviivekuvaajista ei ilmene mitään normaalista poikkeavaa. Paketteja ei matkalla häviä. Viive nousee hieman päiväsaikaan, todennäköisesti johtuen koneiden kuormituksesta, mutta ei siinä määrin, että se selittäisi kirjautumisen hitautta.



Kuva 8. Levypalvelimen vasteaika

Kuvasta 8 näkyy, että vasteajat kyllä nousevat selvästi päiväsaikaan, mutta se todennäköisesti johtuu siitä, että koneen kuorma kasvaa, eikä kone kerkeä vastaamaan yhtä ripeästi. Joissain tilanteissa näkyy myös minimaalista pakettihävikkiä (0,42 prosenttia maksimissaan), tämäkin voi johtua koneen kuormituksesta. Varsinaiset palvelutehtävät priorisoidaan korkeammalle, kuin ICMP ping-paketteihin vastaaminen, jolloin kuorman noustessa riittävästi saatetaan jättää kokonaan vastaamatta.

Lokeissa näkyy satunnaisesti virheitä porteissa, joista muutamia tutkittiin tarkemmin. Virheet olivat kuitenkin vähäisiä, yleensä kerran tai kaksi päivässä yksittäisen portin kohdalla. Useimmiten yksittäisiä virheitä vaikuttaisi tulevan silloin, kun työasema käynnistetään tai sammutetaan, ei varsinaisen käytön aikana.

Ratkaisua alkuperäiseen ongelmaan ei tietoliikenneverkon puolelta löytynyt, joitakin ideoita jatkoon kuitenkin syntyi. Lisäksi työn tuloksena syntyi verkko-ongelmien selvittämiseen useita työkaluja, jotka toivottavasti auttavat jatkotutkimuksissa.

# 7 Yhteenveto ja pohdinta

Työn aloittaminen sujui helposti, kiitos aikaisemman kokemuksen verkkolaitteiden parissa. Tiedonkeruuta haastavampaa oli kerätyn tiedon analysointi ja tulosten esittäminen. Tämä oli työssä eniten aikaa vienyt osuus ja myös se jossa jäi eniten kehitettävää. Miten löytää olennainen osa, ja saada se esitettyä ymmärrettävällä tavalla? Pyrkimyksenä oli myös tehdä järjestelmästä mahdollisimman pitkälle automatisoitu ja vähäistä ylläpitoa vaativa. Käyttäjille kirjoitettiin erillinen käyttöohje, jossa kuvattiin ylläpitoa vaativat osat ja se, miten eri työkalut liittyvät toisiinsa.

Alkuperäisistä tutkimusmenetelmistä kokeilematta jäi itse verkkoliikenteen analysointi. Työaseman verkkoliikennettä kirjautumisen aikana voisi tutkia käyttämällä kytkimistä löytyvää portin peilausta (port mirror). Tämä ominaisuus mahdollistaa kaiken portissa kulkevan liikenteen lähettämisen toiseen porttiin. Peiliporttiin voi kytkeä toisen tietokoneen ja käyttää esimerkiksi Wireshark (http://www.wireshark.org) –ohjelmaa liikenteen analysointiin. Näin näkisi mitä työasema käytännössä liikennöi verkkoon kirjautuessa, ja myös mahdolliset viiveet palvelimen vastauksissa tulisivat ilmi.

Verkon ruuhkautumisen syitä selvittäessä kannattaan huomioida käytettävän sovellustason protokollan vaikutus suorituskykyyn. Windows-ympäristössä tiedostojen jakoon käytetään Microsoftin kehittämää SMB-protokollaa (Server Message Block), kirjautumisen yhteydessä käyttäjäprofiilin synkronointi levypalvelimelta tapahtuu SMB-protokollalla. Protokollan alkuperäisessä versiossa on tunnettuja suorituskykyongelmia. Microsoft on julkaissut SMB-protokollasta version 2 (SMB2), joka parantaa suorituskykyä ja vähentää edestakaista liikennettä asiakkaan ja palvelimen välillä. (Barreto 2008.)

Tämän uuden version käyttö saattaa nopeuttaa profiilin synkronointia kirjautumisen yhteydessä. SMB2 edellyttää kuitenkin käyttöjärjestelmäpäivitystä sekä asiakkaalle että palvelimelle. Asiakkaan tulee olla Windows Vista tai uudempi, palvelimen Windows Server 2008 tai uudempi. Tämän takia tätä ei pystytty työn aikana testaamaan käytännössä.

Työtä tehdessä pääsin perehtymään siihen, mitkä tekijät nykyaikaisten kytkinten suorituskykyyn vaikuttavat ja mitkä tekijät aiheuttavat mahdollisia ongelmia. Ominaisuuksia toiminnan seurantaan löytyy paljon, mutta laitteissa itsessään ei ole kapasiteettia säilyttää tietoa pidemmältä ajanjaksolta. Käytännössä tiedot täytyy kerätä ja tallentaa erilliselle palvelimelle sekä suorittaa analysointi siellä. Tähän tarkoitukseen löytyy paljon kaupallisia sovelluksia, mutta varsin pitkälle pääsee myös työssä käytetyillä avoimen lähdekoodin sovelluksilla. Useimmat ominaisuudet perustuvat avoimiin standardeihin, mikä mahdollistaa näiden, sekä työn tuloksena syntyneiden työkalujen toteuttamisen.

Vaikka alkuperäistä ongelmaa ei työn aikana saatu ratkaistua, suljettiin kuitenkin useita mahdollisia tekijöitä pois. Sovellustason suorituskykyyn vaikuttaa moni muukin asia kuin verkkolaitteet. Käytetty protokolla ei välttämättä skaalaudu nykyisiin vaatimuksiin, laitteista riippumatta. Tutkimuksen ohessa syntyneet työkalut ovat kuitenkin hyödyllisiä monenlaisten verkko-ongelmien selvittämisessä.

## Kuvat



## Taulukot

## Lähteet

Barreto, J 2008. SMB2, a complete redesign of the main remote file protocol for Windows. http://blogs.technet.com/b/josebda/archive/2008/12/05/smb2-a-complete-redesign-of-the-main-remote-file-protocol-for-windows.aspx Luettu 15.1.2011

Cisco Systems Inc. 2010. Bandwidth, Packets Per Second, and Other Network Performance Metrics. http://www.cisco.com/web/about/security/intelligence/network_performance_metrics.html#2 Luettu 7.10.2010

Doherty, J. & Anderson, N. & Della Maggiora, P. 2008. Cisco Networking Simplified, Second Edition. Indianapolis, USA. Cisco Press

Hewlett-Packard Development Company 2010. HP ProCurve Switch 2500 Series. http://h10144.www1.hp.com/products/switches/switch2524-2512/overview.htm#J4813A Luettu 7.10.2010