Metropolia Ammattikorkeakoulu
Tietotekniikan koulutusohjelma

Aki Ikonen

Verkon suorituskyvyn mittaaminen

Insinöörityö 18.4.2010

Ohjaaja: yliopettaja Matti Puska
Ohjaava opettaja: yliopettaja Matti Puska

Metropolia Ammattikorkeakoulu **Insinöörityön tiivistelmä**

Tekijä Otsikko Sivumäärä Aika	Aki Ikonen Verkon suorituskyvyn mittaaminen 32 sivua 18.4.2010
Koulutusohjelma	tietotekniikka
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	yliopettaja Matti Puska yliopettaja Matti Puska

Tässä insinöörityössä tutustuttiin verkon suorituskyvyn valvontaan pienikokoisessa laboratorio-verkossa tehdyillä mittauksilla.

Teoriaosuudessa käytiin läpi merkittäviä asioita verkon valvonnasta sekä suorituskyvystä ja sen mittauksesta.

Käytännön osuudessa tehtiin mittauksia pienikokoisessa verkossa käyttäen kolmea eri ilmaisohjelmaa, jotka soveltuivat suorituskyvyn mittaamiseen. Käytännön osuus antaa mittaustuloksien lisäksi kuvan, jonka avulla mittauksia pystyy toistamaan.

Mittaustuloksia onnistuttiin tekemään ilmaisohjelmilla, mutta ohjelmia täytyy olla monia saadakseen suorituskyvyn mittaustuloksia riittävän monesta osa-alueesta. Tutkimustuloksista voi kuitenkin päätellä, että ilmaisohjelmat eivät mahdollisesti ole paras ratkaisu suorituskyvyn valvontaan – ainakaan yrityksissä, joissa verkon koko on suuri.

Hakusanat	suorituskyky, verkonhallinta, vasteaika

Helsinki Metropolia University of Applied Sciences Abstract

Author	Aki Ikonen
Title	Network performance monitoring
Number of Pages	32
Date	18 April 2010
Degree Programme	Information Technology
Degree	Bachelor of Engineering
Instructor	Matti Puska, Principal Lecturer
Supervisor	Matti Puska, Principal Lecturer

This thesis introduces network performance testing in small scale laboratory network.

In the theoretical part of the thesis the most important details are about performance measuring and network monitoring.

In the practical part of the work there are three different freeware or open source softwares used, to measure and monitor the laboratory network. Therefore, practical part is giving the results of the measurements and tests and at the same time provides instructions on how to repeat the tests that have been executed in the project.

From the outcome of the tests it becomes obvious that the free/open source software, although being able to provide a decent amount of information and data, wouldn't be the ideal option for network monitoring; at least not in companies where the size of the network is significantly large.

Keywords	performance, network monitoring, response time

Sisällys

Lyhenteet

HTTP	Hypertext Transfer Protocol; protokolla, jota käytetään web-selaamisessa
IP	Internet Protocol; Internetissä käytettävä pakettimuoto
ITU-T	International Telecommunications Union – Telecommunications Standardization Sector; kansainvälinen televiestintäliitto
LAN	Local Area Network; lähiverkko
MTBF	Mean Time Between Failures; aika, joka kuluu virheiden välillä
MTTR	Mean Time To Repair; aika, joka kuluu ennen kuin tilanne on korjattu
PHP	Hypertext Preprocessor; ohjelmointikieli, jota käytetään web-pohjaisissa palveluissa
QoS	Quality of Service; palvelutason laatu
RTT	Round Trip Time; aika, joka kuluu paketin lähettämisestä paketin takaisin saapumiseen
SLA	Service Level Agreement; palvelunlaatusopimus
TCP	Transmission Control Protocol; protokolla, joka on yleisesti käytössä verkkoliikenteessä
UDP	User Datagram Protocol; protokolla, joka on myös yleisesti käytössä verkkoliikenteessä
VoIP	Voice over IP; kansainvälinen verkkopuheluiden nimike
WFQ	Weighted Fair Queuing; jonotusjärjestelmä, mikä on reitittimissä yleensä vakiona

1 Johdanto

Verkonhallinnasta on tullut elintärkeä osa erilaisten organisaatioiden IT-järjestelmien toiminnanohjausta. Nykyinen taloustilanne on kuitenkin pakottanut yritykset miettimään tarkemmin IT-investointejaan, ja kustannustehokkuudesta on muodostunut merkittävä tekijä, joka rajoittaa verkonhallinan työkalujen valinnanvaraa. Tämä tutkielma lähestyy aihetta suorituskyvyn kautta.

Ottaen huomioon vallitsevan tilanteen, jossa kustannustehokkuus on keskeisintä, tämä tutkimus pyrkii selvittämään, pystytäänkö ilmaisohjelmilla saavuttamaan hyväksyttävä tulos suorituskyvyn mittaamisessa ja täten etsii vastausta oheiseen tutkimuskysymykseen: *Voidaanko verkon suorityskykyä mitata luotettavasti ilmaiseksi saatavilla olevilla ohjemilla?*

Työn tarkoituksena oli mitata verkon suorituskykyä ilmaiseksi saatavilla olevilla ohjelmilla. Ohjelmia löytyi tähän käyttötarkoitukseen muutamia, mutta verkon suorituskyvyn mittaukseen tarvitaan vähintään kolmea osa-aluetta: kaistanleveyden mittaamista, vasteajan mittaamista sekä palvelutason täyttymisen valvontaa. Näitä kaikkia kolmea varten ei yhtä ilmaisohjelmaa löytynyt. Työtä varten jouduttiin käyttämään kolmea eri ohjelmaa saamaan mittaustuloksia kaikista näistä osa-alueista. Kukin testi kesti kaksitoista tuntia, ja niistä saadulla datalla saattoi tehdä riittävän luotettavia päätelmiä.

Päädyin valitsemaan verkon suorituskyvyn valvonnan mittaamisen siksi, että aihe itsessään oli mielenkiintoinen ja siitä teki haastavan se, ettei saatavilla ollut valmiina yhtä ohjelmaa, jolla kaikki mittaukset ja päätelmät olisi mahdollista tehdä. Käytännön osalta tutkielman antina on tarjota verkonhallinnan käytön empiirisesti tutkittu ja toimivaksi varmistettu menetelmä verkon suorituskyvyn mittaamiseen ilmaiseksi saatavilla olevilla ohjelmilla. Tutkimustilanne on toistettavissa missä tahansa seuraamalla tässä tutkimusraportissa esiteltyjä askeleita.

Insinöörityön kaltaisen tutkielman tulisi täyttää tutkimuksen osalta *validiteetin* (pätevyys) ja *reliabiliteetin* (pysyvyys) asettamat vaatimukset. Ensiksi mainitulla

viitataan tutkimustulosten tarkkuuteen ja mittareiden kykyyn mitata tarkoitettua asiaa. Jälkimmäisellä vuorostaan tarkoitetaan tutkimusasetelman johdonmukaisuutta, tarkkuuta ja sattumanvaraisten virheiden minimoimista. [12] Tähän pyrittiin esimerkiksi tutkimustilanteen yhteydessä valitsemalla mahdollisimman yksinkertainen laitteisto ja selostamalla tutkimuksen eteneminen seikkaperäisesti.

2 Verkon suorituskyky

2.1 Suorituskyvystä tarkemmin

Suorituskyvyn kertomiseen ei ole yksittäistä skalaarisuuretta, joka kuvaisi koko suorituskykyä. Käyttäjän kannalta ”suorituskyky” tuo ensimmäiseksi mieleen kysymyksen, kuinka nopeasti laite vastaa pyyntöihin, kun taas toiselle tulee mieleen käytännön läheisempi vastaus: ”Kuinka montaa asiakasta voidaan kerralla palvella?”. Syvemmälle katsottaessa pelkkä vastausaika tai palveltavien asiakkaiden määrä ei riitä vastaukseksi, mikäli palvelu ei ole jatkuvasti toiminnossa tai sen toimivuudesta ei ole takeita. Toisesta näkökulmasta katsottuna jotkin toiset laatuominaisuudet palvelussa voivat olla suorituskykyä tärkeämpiä sen toimivuuden kannalta. [3.]

Suorituskyvyn suureet voidaan jakaa kahteen eri luokkaan: ”käyttäjille näkyviin palvelukykysuureisiin” ja näkyviä suureita selittäviin ”sisäisiin suorituskykysuureisiin”. Ensimmäisiin kuuluu esimerkiksi vasteaika ja suoritusteho ja toisiin muiden muassa komponenttien käyttöasteet ja järjestelmän sisäisien jonojen pituudet. [3.]

Käyttäjän kannalta suorituskyky tarkoittaa järjestelmän kykyä palvella. Käyttäjä huomaa suorituskyvyn laitteen nopeudesta vastata pyyntöihin, mikäli palvelu on saatavilla. Mikäli palvelu ei ole lainkaan saatavilla, suorituskyvyn arvo on nolla.

Suorituskyvyn havaitseminen eri osa-alueissa on huomattavampaa kuin toisissa. Esimerkiksi VoIP (Voice over IP) -puheluissa, IPTV:ssa (Internet Protocol Television) tai videota toistettaessa suorituskyvyn täytyy olla riittävää, jotta ääni tai video ei pysähdy sitä toistettaessa. Puolestaan tiedonsiirrossa tai verkkosivujen latautumisen nopeudessa ei suorituskyvyn tarvitse olla täydellistä. Tätä varten verkkolaitteissa on mahdollista määritellä QoS-palvelu (Quality of Service), jolla voidaan määrittää tosiaikainen liikenne kulkemaan verkossa korkeammalla prioriteetilla kuin esimerkiksi verkkoselaaminen. Kuva 1 esittää tosiaikaisten verkkosovellusten SLA (Service Level Agreement) -rajoja.

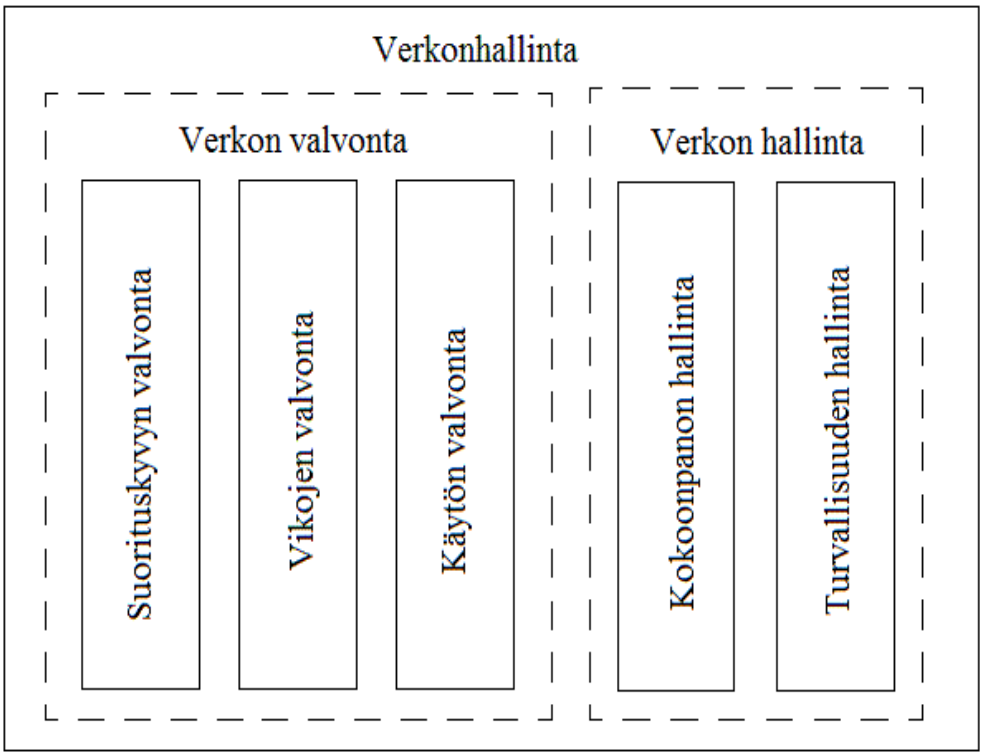
Traffic Type	Maximum Packet Loss	Maximum One-Way Latency	Max. Jitter
VoIP	1%	200 ms	30 ms
Videoconferencing	1%	200 ms	30 ms
IPTV broadcast video	0.001%	250 ms	50 ms

Kuva 1. SLA-arvot VoIP:lle, videoneuvottelulle ja IPTV:lle. [4]

2.2 Verkonhallinta

Verkonhallinnan tärkeys kasvaa verkon koon kasvaessa. Tietoverkot tarjoavat monenlaisia palveluita ja protokollia, jotka kuljettavat monentyyppistä dataa verkossa. Nykyiset yritykset ovat yhä enemmän riippuvaisia Internetin ja Intranetin toiminnasta – mikäli verkko lakkaa toimimasta, työt keskeytyvät mitä suurimmalla todennäköisyydellä. Verkkojen topologiat kasvavat jo pienissäkin yrityksissä sellaisiin mittasuhteisiin, että verkkoa ei pysty enää ihmisvoimin valvomaan. Tämän takia on ihmisten avuksi on kehitetty ja jatkuvasti kehitetään lisää uusia työkaluja, joiden avulla verkonhallinta on helpompi toteuttaa. [14.]

Verkonhallinta koostuu kahdesta osa-alueesta: verkon valvonnasta ja verkon hallinnasta. Verkon valvonta on lukuprosessi, jossa mitataan verkon eri osa-alueiden arvoja tiettyinä hetkinä. Sen tärkein tehtävä on analysoida verkkoa ja antaa tietoa verkon määrittelyistä. Verkon valvonnan tärkein tavoite on parantaa järjestelmän luotettavuutta. Verkon hallinta puolestaan on kirjoitusprosessi, jonka avaintehtävänä on laitteiden ja komponenttien asetusten ylläpitäminen. Kuva 2 esittää verkonhallinnan karkeaa jakoa hallintaan ja valvontaan. [2.]



Kuva 2. Verkonhallinnan osa-alueet. [2]

2.3 Suorituskyvyn valvonta

Verkon valvonta vaatii erityyppisiä informaatioita verkosta, jotta tuloksista voidaan tehdä päätelmiä. Nämä tyypit voidaan jakaa kolmeen eri osa-alueeseen:

- staattinen informaatio, joka kertoo verkon muun muassa verkon topologian, esimerkiksi kytkinten määrän verkossa
- dynaaminen informaatio, joka on koko ajan muuttuvaa, esimerkiksi verkossa liikkuvan datan hetkellinen määrä
- tilastollinen informaatio, joka yleensä muodostuu dynaamisen informaation perusteella, kuten tietyn ajanjakson arvo liikkuneesta datasta kahden reitittimen väliltä. [2.]

Verkonhallinnan tärkeänä osa-alueena on suorituskyvyn valvonta, koska verkon tai muunkaan järjestelmän hallinta ei onnistuisi ilman kuormituksen valvontaa. Suorituskyvyn valvonnalla on tavoitteena ylläpitää tietoa verkon suorituskykyyn vaikuttavista suureista, joiden heikentymisestä tulisi saada hälytys, jonka avulla tilanteeseen pystyttäisiin vaikuttamaan mahdollisimman nopeasti. Suorituskyvyn valvonnan haasteena on mittareiden valinta, koska osa mittareista on laitevalmistaja-kohtaisia eivätkä tue toisen valmistajan verkkolaitteita. [2.] Tämän takia tämä työ painottuu valvontaohjelmiin, jotka eivät ole valmistaja riippuvaisia.

2.4 Verkon suorituskyvyn mittaus

Verkon suorituskyvyn mittaamiseen vaaditaan vähintään kolmesta neljään osa-aluetta. Verkon suorituskyvyn kannalta merkittävät tekijät ovat:

- saatavuus (Availability)
- vasteajat (Response time)
- virheettömyys (Accuracy) ja datan eheys
- käyttöaste (Utilization) [1].

Tässä työssä mitattiin myös verkon läpäisykykyä (Engl. throughput) näiden lisäksi.

Edellä mainitusta listasta suorituskykysuureet voidaan jakaa kahteen luokkaan, jotka ovat palvelukykysuureet – suureet, jotka näkyvät käyttäjälle ja joita edustavat tässä listassa saatavuus sekä vasteajat sekä sisäisiin suorituskykysuureisiin, joita tässä listassa edustavat virheettömyys ja käyttöaste. [3.]

Saatavuus kertoo laitteiden mahdollisista ”alas menoista” eli vikatiloista, jolloin laite ei ole käytettävissä. Saatavuus oli mittauksia tehdessä sataprosenttinen, mutta käytännössä siihen vaikuttavia asioita voivat olla esimerkiksi sähkökatkokset, laitteiden vioittuminen tai inhimilliset verkonvalvojien tekemät virheet. Stallingsin [13] mukaan saatavuutta pystyy määrittelemään kaavalla 1:

$$Saatavuus = \frac{MTBF}{(MTBF + MTTR)}, \quad (1)$$

jossa MTBF (Mean Time Before Failure) on aika ennen ongelmaa ja MTTR (Mean Time To Repair) on aika, joka ongelman korjaukseen kuluu. [1.]

Vasteajalla tarkoitetaan aikaa, joka järjestelmältä kuluu annetun pyynnön suorittamiseen. Vuorovaikutteisessa käytössä vasteaika voidaan määritellä ajaksi, joka kuluu viimeisen syötteen antamisesta vastauksen saamiseen järjestelmältä. Vasteaikaa palvelimelta ladattavalta web-sivulta kuvaisi seuraavanlainen tapahtumaketju:

- viive päätteeltä päätepalvelimeen eli päätelaitteen linjanopeudesta koostuva viive
- viive, joka syntyy päätepalvelimen odottaessa vuoroaan lähettää tietoa verkkoon
- verkon aiheuttama viive, joka kuluu tiedon siirtymisessä palvelimesta päätteeseen
- palvelupyynnön suoritukseen kuluva aika palvelimella
- viive palautumisesta palvelimelta takaisin verkkoon
- verkon aiheuttama viive
- viive päätepalvelimesta päätteelle eli linjanopeudesta aiheutuva viive [2.].

Datan eheys on teoriassa merkittävä asia, mutta tämän työn mittauksissa ei datan eheyteen varsinaisesti oteta kantaa, koska sen oletetaan olevan paras mahdollinen. Datan eheyttä mitataan prosentuaalisesti. Esimerkiksi jos lähetetyistä sadasta paketista kaksi pakettia päätyy bittivirheeseen, on datan eheys 98 prosenttia. Mikäli datan eheys laskee ja paketit päätyvät bittivirheisiin, tämä aiheuttaa paketin uudelleen lähettämisen, mikä tässä tapauksessa tarkoittaa suorituskyvyn alenemista virheellisen paketin kohdalta 50 %:iin normaalista. Virheiden määrää pystytään prosentuaalisesti laskemaan kaavalla 2:

$$Virheiden\ lukumäärä = \frac{(\Delta ifInErrors)\times 100}{(\Delta ifInUcastPkts + \Delta ifInNUcastPkts)}, \quad (2)$$

Virheettömyydelle on olemassa puolestaan seuraavanlainen kaava 3:

$$Virheettömyys = \frac{100-(\Delta ifInErrors)\times 100}{(\Delta ifInUcastPkts + \Delta ifinNUcastPkts)}, \quad (3)$$

joissa, $\Delta ifInErrors$ on kahden SNMP (Simple Network Management Protocol) ifInErrors-objektin tilan kyselykierron ero, joka kuvaa sisääntulevien virheellisten pakettien määrää. $\Delta ifInUcastPkts$ on kahden SNMP ifInUcastPkts -objektin tilan kyselykierron ero, joka kuvaa sisääntulevien täsmälähetys-pakettien määrää. $\Delta ifInNUcastPkts$ puolestaan on kahden SNMP ifInNUcastPkts-objektin tilan kyselykierron ero, joka kuvaa sisääntulevien ryhmälähetys- ja yleislähetys-pakettien määrää. [1.]

Käyttöaste mittaa laitteen käytössä oloaikaa tietyn hetken aikana. Esimerkiksi jos verkossa tehdään läpäisykyvyn suorituskyvyn mittausta, on käyttöaste kyseisellä hetkellä sadassa prosentissa verkon sarjalinkin osalta, koska verkko on täydessä rasituksessa niiden osalta. Käyttöastetta mittaamalla ja valvomalla pystytään löytämään verkon ”pullonkauloja”, joihin olisi syytä keskittyä. Esimerkiksi jos käyttöaste jonkin linkin osalta on keskimäärin 1 %, ei se todennäköisesti tule olemaan verkon ”pullonkaulana” missään vaiheessa. Mutta mikäli linkin keskimääräinen käyttöaste on 10 % tai enemmän, on linkki vähintäänkin lyhytaikaisesti täydessä rasituksessa, joka

tällöin rajoittaa muun verkon siirtonopeutta. [5.] Mittaamalla käyttöaste esimerkiksi prosessorista, verkon liitännöistä tai jonoista, on verkon ongelman solmupiste helpompi löytää. Käyttöasteen mittaamiseen käytetään seuraavanlaisia kaavoja:

$$\frac{(\Delta ifInOctets + \Delta ifOutOctets) \times 8 \times 100}{(sekunttejen\ määrä\ \Delta{:}n\ aikana) \times ifSpeed}, \tag{4}$$

Kaavaa 4 käytetään vuorosuuntaisen (half duplex) käyttöasteen mittaukseen.

$$\frac{\max(\Delta ifInOctets, \Delta ifOutOctets) \times 8 \times 100}{(sekunttien\ määrä\ \Delta{:}n\ aikana) \times ifSpeed}, \tag{5}$$

Kaavaa 5 käytetään laskettaessa käyttöastetta kaksisuuntaisesta (full duplex) -yhteydestä. Tarkemman tuloksen kaksisuuntaisesta -käyttöasteesta saa kuitenkin laskemalla sisääntulo- ja ulostulokäyttöasteen erikseen seuraavanlaisesti:

$$Sisääntulo\ Käyttöaste = \frac{\Delta ifInOctets \times 8 \times 100}{(sekunttien\ määrä\ \Delta{:}n\ aikana) \times ifSpeed}, \tag{6}$$

ja

$$Ulostulon\ käyttöaste = \frac{\Delta ifOutOctets \times 8 \times 100}{(sekunttien\ määrä\ \Delta{:}n\ aikana) \times ifSpeed}, \tag{7}$$

näissä kaavoissa $\Delta ifInOctets$ on kahden SNMP $ifInOctets$ -objektin tilan kyselykierron ero, mikä kuvaa sisääntulevan liikenteen oktettien määrää. $\Delta ifOutOctets$ on kahden SNMP $ifOutOctets$ -objektin tilan kyselykierron ero, joka kuvaa ulosmenevän liikenteen oktettien määrää. $ifSpeed$ on SNMP ifSpeed-objektin määritetty liitännän nopeus. [1.]

Läpäisykyvyn mittaukseen käytetään kaavaa 8:

$$Maksimaalinen\ läpäisykyky = \frac{TCP - ikkuna\ koko}{RTT} \quad (8)$$

Tässä työssä läpäisykyvyn laskemista ei kuitenkaan tehty manuaalisesti laskemalla, vaan sen laskemiseen hyödynnettiin Iperf-työkalua.

2.5 VoIP:n suorituskyvyn määrittäminen

VoIP-puheluiden laatua mitataan MOS (Mean Opinion Score) -arvolla. Tämän laskemiseen on olemassa ITU-T:n (International Telecommunications Union – Telecommunications Standardization Sector) määrittämä E-malli. E-mallin perusteella on tehty myös työkaluja, jotka auttavat suunnittelemaan korkeatasoisia ääniohjelmia piiri- ja pakettikytkentäisissä verkoissa. Työkalu arvioi suhteellista häiriötä äänenlaadussa verrattaessa eri verkkolaitteita ja verkon topologioita. [6]

Merkittäviä tekijöitä äänenlaadussa ovat

- viive
- pakettihävikki
- kaiku
- taustakohina
- koodekin valinta [6].

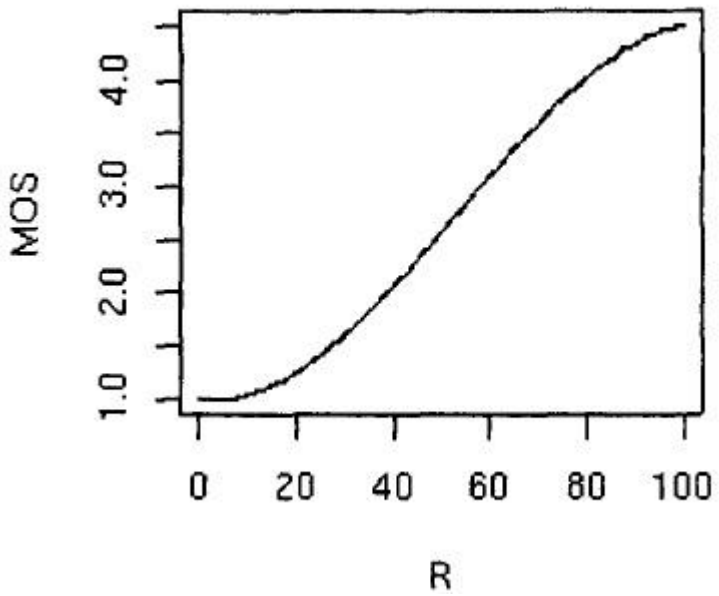
E-mallin pääperiaate on laskea R-tekijän arvo, mikä voi saada arvon 0–100, jossa 0 vastaa huonointa mahdollista ja 100 parasta mahdollista arvoa. Tämä R-tekijä määrittää MOS-arvon, jonka asteikko on

- heikko
- alhainen
- keskitaso
- korkea
- paras [6].

MOS määrittyy R arvojen perusteella kaavan 9 mukaan seuraavasti [6]:

$$R < 0 : MOS = 1$$
$$R > 100 : MOS = 4.5$$
$$0 < R < 100 : MOS = 1 + 0.035\,R + 7x10^{6}R(R-60)(100-R) \quad (9)$$

Kuvassa 3 kaava on sijoitettuna kuvaajaan:



Kuva 3. R-tekijän suhteen MOS:iin. [6]

Taulukko 1 osoittaa miten R-tekijän arvot on yleisesti luokiteltu. Tästä voidaan huomata, että R-tekijän arvot alle 60:ssä vastaavat heikkoa äänenlaatua.

Taulukko 1. R-kuvaajan taso arvot ja määritetyt MOS arvot. [6]

R-tekijä	Äänenlaatu	MOS
90 < R < 100	Paras	4.34-4.50
80 < R < 90	Korkea	4.03-4.34
70 < R < 80	Keskitaso	3.60-4.03
60 < R < 70	Alhainen	3.10-3.60
50 < R < 60	Heikko	2.58-3.10

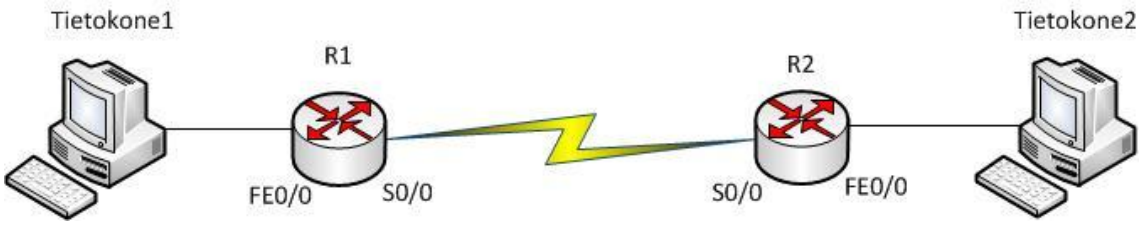
Tässä työssä ei käytännön osuudessa kuitenkaan VoIP-puheluiden laatua testattu sen tarkemmin, mutta edellämainitut tekijät määrittävät teoreettisesti VoIP-puheluiden laadun.

3 Mittauslaitteisto

Mittaustavoitteena oli saada tuloksia aikaisemmin mainituista suorituskykyyn liittyvistä tekijöistä, kuten vasteajasta, käytettävyydestä sekä verkon läpäisykyvystä. Näiden suureiden mittaamiseen jouduttiin käyttämään kolmea eri työkalua, koska ilmaisohjelmista ei löytynyt yhtä tiettyä ohjelmaa, jolla mittaukset olisi pystynyt tekemään. Mittauksia varten pyrittiin rakentamaan mahdollisimman yksinkertainen verkko selkeästi tulkittavien mittaustuloksien saamiseksi.

Käytössäni mittauksia varten oli kaksi Pentium 4-pöytäkonetta, joissa molemmissa oli Windows XP, jossa käytettiin VMware ohjelmaa virtuaalikoneiden pyörittämistä varten. Virtuaalikoneiden käyttöjärjestelmiksi otettiin Linuxin Ubuntu-käyttöjärjestelmän PasTmon -monitorointiohjelmaa varten. PasTmon ei tarvitse erityisesti Ubuntua kaikista eri Linux-versioista, mutta valitsin Ubuntun, koska se oli kaikista versioista tutuin itselleni. PRTG (Paessler Router Traffic Grapher) -ohjelma pyörii ainostaan Windows-pohjalla ja tämä käyttöjärjestelmä oli molemmissa koneissa pääkäyttöjärjestelmänä.

Verkkolaitteina käytössä oli kaksi Ciscon 2611XM-reititintä, joista ensimmäisen reitittimen (R1) FastEthernet-portti oli kytkettynä ensimmäiseen pöytäkoneeseen (Tietokone1), ja toisen reitittimen (R2) FastEthernet-portti oli kytkettynä toiseen pöytäkoneeseen (Tietokone2). Reitittimet olivat kytkettyinä toisiinsa sarjalinkin kautta, jonka nopeudeksi oli määritetty 128 kbit/s, jotta mittaustulokset olisivat helpompia tulkita. Päädyin tekemään mahdollisimman yksinkertaisen verkkotopologian, koska se myös helpottaa tuloksien tulkitsemista ja mahdolliset ongelmat on helpompi löytää yksinkertaisesta verkosta. Kuva 4 havainnoillistaa verkon topologiaa:



Kuva 4. Käytetyn verkon topologia.

4 Ohjelmisto

Ohjelmistotyötä varten tuli olla Open Source eli vapaan lähdekoodin omaava ohjelma tai vaihtoehtoisesti esittely- tai vapaasti kokeiltavissa olevia-versioita maksullisista ohjelmista, koska tarkoituksena oli saada mittaukset tehtyä ilman ylimääräisiä kustannuksia. Tavoitteena oli löytää ohjelma tai ohjelmia, joilla pystytään tekemään verkon suorituskyvyn mittauksia eri osa-alueilla. Ilmaisohjelmista ei kuitenkaan löytynyt yhtä ohjelmaa, jolla olisi pystynyt tekemään mittauksia jokaiselta osa-alueelta. Valinnat lopullisista ohjelmista työtä varten tehtiin ohjelmien ominaisuuksien perusteella.

Työn aikana jouduttiin kokeilemaan erilaisia ohjelmia ja niiden soveltuvuutta verkon suorituskyvyn mittaamiseen. Ensimmäisiä ohjelmistoja, joita valitsin niiden luvattujen ominaisuuksien perusteella, olivat Zenoss ja Xymon, mutta näiden ohjelmien ominaisuudet eivät suoranaisesti olleet käytännöllisiä tätä työtä varten. Nämä ohjelmat olisivat sopivampia isompaan verkkoon ja valvomaan, että verkon laitteet ovat toiminnassa. Ohjelmissa oli tätä varten erilaisia mahdollisia hälytyksiä, joiden avulla laitteiden mahdollinen alasmeno voitaisiin tunnistaa ja reagoida siihen mahdollisimman nopeasti. Nämä tiedot eivät varsinaisesti auttaneet tämän työn tavoitteissa, vaikka saatavuus on yksi suorituskyvyn elementeistä.

Vasteajan suorituskyvyn mittaamiseen päädyttiin lopulta PasTmon v0.13b Open Source -ohjelmaan eli vapaan lähdekoodin ohjelmaan, joka on passiivinen verkkoliikenteen valvontaohjelmisto. Tämä ohjelma on Linux-alustoilla toimiva ja vaatii monia kirjastoja lähdekoodin lisäksi. Passiivinen pakettien ”haistelu” tapahtuu libpcap-kirjaston avulla. Ohjelmassa on lisäosa, joka lähettää viiden minuutin välein pakattua tietoa PostgreSQL-tietokantaan ja PasTmonin PHP (Hypertext Preprocessor) -koodi käy Apachen avulla lukemassa tietoja SQL-kannasta. PasTmon valittiin työhön, koska se oli hyvä ominaisuuksiltaan vasteajan mittaukseen. PasTmon -ohjelmisto on vähemmän kokeneelle Linux-käyttäjälle haastava asentaa, mutta ohjelman tekijä ja siihen liittyvät foorumit auttavat ongelmissa.

Paessler Router Traffic Grapher (PRTG) on puolestaan aktiivinen monitorointityökalu. Aktiivisen monitoroinnin erona passiiviseen on se, että aktiivinen monitorointityökalu luo liikennettä tuloksien saamista varten. Ohjelma valittiin laitteen käytettävyysmittauksia varten. PRTG valittiin työhön aluksi sen lupaamien ominaisuuksien pohjalta, vaikka se ei kaikkia ominaisuuksia kuitenkaan täyttänyt loppujen lopuksi. Ohjelma on maksullinen, mutta siitä on saatavilla ilmainen kokeiluversio täysillä ominaisuuksilla 30 päivän ajalle tai vaihtoehtoisesti ohjelma rajoitettuna vain kymmeneen tunnistimeen, ilman aikarajoitusta.

Ohjelman ominaisuuksiin kuului esimerkiksi VoIP-simulointi kahden PRTG -ohjelman välilllä. Kokemukseni testien jälkeen oli, että käytettävissä oleva versio ei kuitenkaan toiminut toivotulla tavalla. Ohjelman mukaan liikennettä oli, mutta reitittimien lokien mukaan liikennettä ei kulkenut laisinkaan. Wiresharkin avulla tämä varmistui, kun Wireshark ei näyttänyt verkossa menevän yhtään VoIP-pakettia.

Ohjelmasta pystyttiin kuitenkin hyödyntämään ohjelman omaa *ping*-työkalua, jonka avulla saatiin käytettävyysmittauksia tehtyä. Ohjelmassa voi asettaa hälytysrajoja tiettyjen vaatimuksien ylityttyä, jolloin ohjelma kirjaa lokiin virheilmoituksen. *Ping*-paketteja lähetettiin VoIP-liikennettä simuloiden, niin että kaistanleveys lähetetyistä *ping*-paketeista oli hyvin lähellä normaalia VoIP-liikennettä. Huomioon vasteajoista täytyy kuitenkin ottaa se, että VoIP-liikenne on yksisuuntaista ja *ping*-työkalun kanssa paketit myös palaavat, jolloin vasteaika pitää jakaa kahdella vastatakseen VoIP-pakettien reaalista viivettä.

Testiohjelmiksi monitorointityökalujen ohella täytyi käyttää ohjelmia, jotka luovat verkkoon liikennettä, jotta monitorointituloksiin saadaan tuloksia. PRTG-ohjelma monitoroinnin ohella hoiti myös liikenteen generointia, joten tämä ohjelma kuuluu myös testausohjelmiin.

Iperf on yksinkertainen palvelin-asiakasohjelma, jossa ohjelma pitää olla kahdella eri koneella ja toisessa se asetetaan palvelintilaan ja toisessa asiakastilaan. Oletusasetuksina Iperf lähettää UDP-paketteja (User Datagram Protocol) asiakaspäästä palvelinpäähän

luodessaan liikennettä. Iperfissä on myös mahdollista käyttää TCP (Transmission Control Protocol) -paketteja UDP-pakettien sijaan. Tässä insinöörityössä hyödynnettiin Iperf-ohjelmaa mittaamaan kaistan läpäisykykyä sekä simuloimaan verkossa liikennettä. Iperf-ohjelma valittiin liikenteen generointiohjelmaksi, koska sillä on helppo tehdä haluttavan verran liikennettä jatkuvalla toistolla. Tässä työssä ohjelmalla simuloitiin 64 kbit/s liikennettä, joka vastaa puolta sarjalinkin kapasiteetista. Myös liikenteen simulointi tapahtuu UDP-pakettien lähetyksellä.

5 Verkon suorituskyvyn hallinta

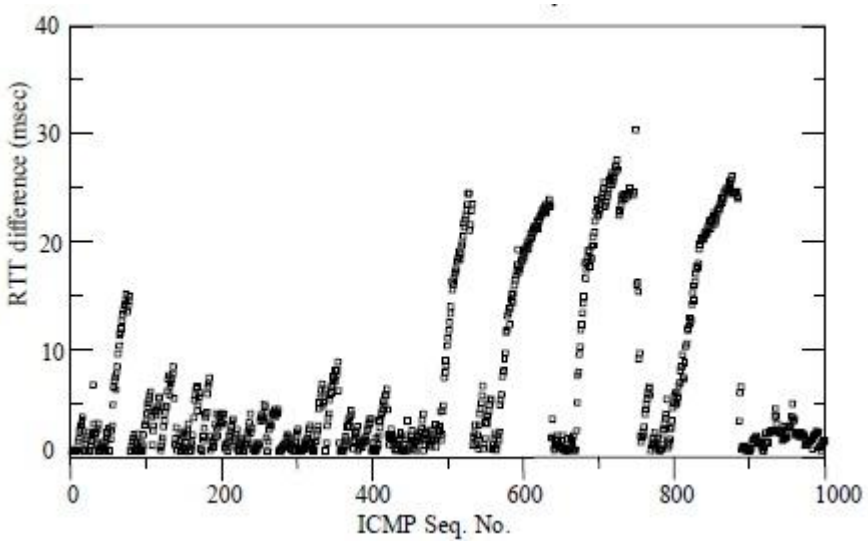
Verkkoliikenteen termeissä *verkon suorituskyvyn hallinta* on määrittely- ja mittaustulos verkon liikenteestä, jonka avulla saadaan jatkuva ja ennustettava tulos verkon palvelutasosta. Suorituskyvyn hallinta sisältää verkon aktiviteettien valvonnan ja verkon suunnittelun tai määrittelyiden muuttamista, verkon suorituskyvyn ja liikenteen hallinnan parantamiseksi. Verkon suorituskyvyn mittaaminen auttaa verkon valvojaa huomaamaan seuraavia seikkoja:

- normaalia verkon suorituskykyä, jotta huono suorituskyky pystyttäisiin tunnistamaan helpommin
- viimeisimmät tai mahdolliset rasitusongelmat
- hitaat vasteajat
- ohjelmatason, palvelimien sekä verkon ylhäälläoloaikaa
- verkon optimi siirtoaikoja [1.].

5.1 Aktiivinen ja passiivinen monitorointi

Yleisesti suorituskyvyn vertailu verkoissa on tehty joko passiivisella tai aktiivisella monitoroinnilla. Aktiivisessa vertailussa ohjelma tuottaa paketteja verkkoihin, joiden avulla se tekee mittaustuloksia. Esimerkiksi *ping* lähettää verkkoon ICMP (Internet Control Messages Protocol) -paketteja ja antaa niiden saapumiseen menneen ajan perusteella RTT (Round Trip Time) -tuloksen [8]. Passiivisen monitoroinnin periaatteena on puolestaan ”haistelu”, joka valvoo verkon normaalia liikennettä ja tekee sen perusteella mittaustuloksia, eli passiivinen monitorointi ei luo verkkoon minkäänlaista uutta liikennettä.

Seuraavanlaisen tutkimuksen mukaan aktiivinen ja passiivinen monitorointi eivät eroa tuloksien perusteella niin paljoa kuin voisi olettaa. On kuitenkin tärkeää käyttää parasta mahdollista mittausteknologiaa riippumatta siitä, kuinka paketit on luotu.
Kuva 5 esittää eroa RTT:n mittauksessa, jossa on käytetty sekä passiivista ja aktiivista monitorointia [7].



Kuva 5. Aktiivinen ja passiivinen monitorointi. [7]

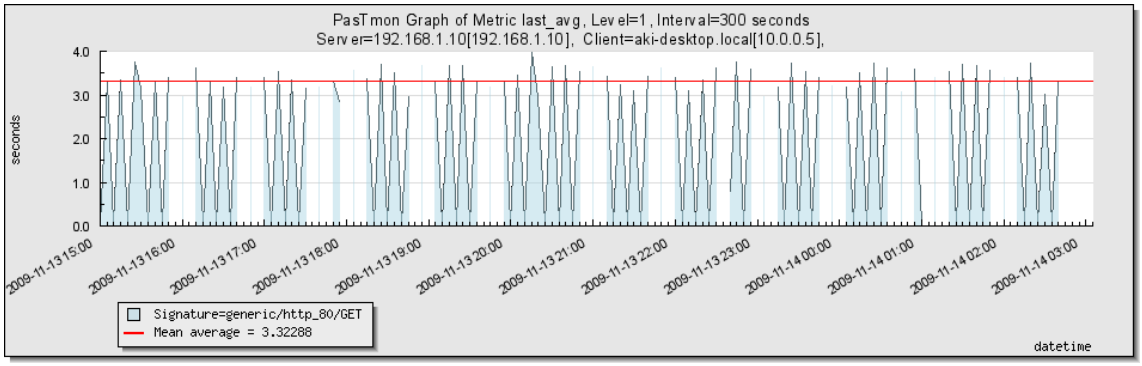
Kuvassa oikean puolen mittaustuloksen tehnyt kone lähettää *ping*-työkalulla ICMP-paketteja ja mittaa niistä RTT:tä ja vasemmalla puolella kuvasta on kone, joka ”haistelee” passiivisesti liikennettä. Kuvasta voi havaita, että aktiivisen ja passiivisen mittaustuloksen eroavaisuus voi olla jopa 30 ms ja tulokset ovat harvemmin lähempänä toisiaan kuin 5 ms. [7]

6 Testituloksia

Testejä tehtiin kahdentoista tunnin mittausjaksoissa. Kaksitoista tuntia on tosin lyhyehkö aika verkon suorituskyvyn mittaustesteihin, sillä poikkeavuudet tuloksissa saattavat olla harvassa eikä näin lyhyt mittausjakso tämän takia niitä tuo aina esille. Tällä tavalla saatiin kuitenkin ns. ihanteelliset olosuhteet, joissa datan eheys ei kärsinyt ja tuloksista saatiin helpommin tulkittavia.

6.1 Vasteajan mittaus

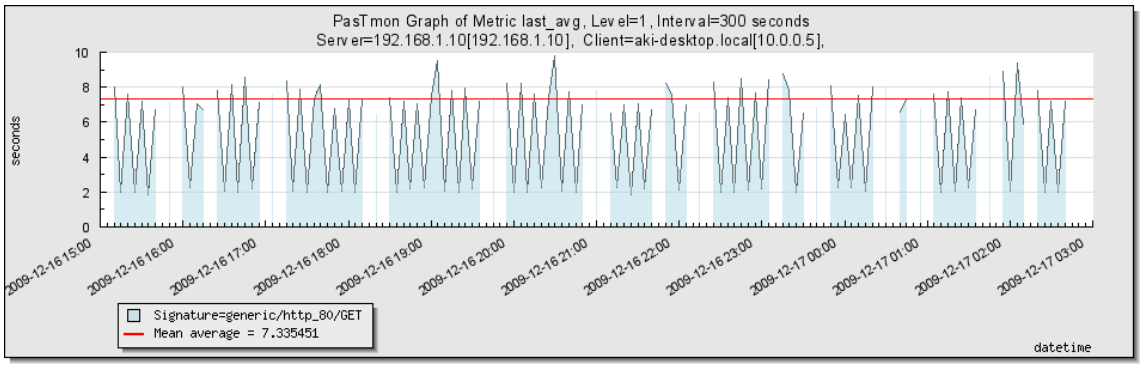
Ensimmäisenä testattiin verkon vasteaikaa PasTmon-ohjelmalla, jolla saatiin kuvan 6 mukaisia tuloksia:



Kuva 6. PasTmon mittaus 1.

Testissä tietokone1:ltä ladattiin 22 kt:n kokoinen tiedosto tietokone2:lle käyttäen HTTP (Hypertext Transfer Protocol) -protokolaa. Tietokone2:lla oli PasTmon-ohjelma monitoroimassa liikennettä. Verkon topologia oli edellä mainitunlainen, eli ”pullonkaulana” verkossa oli sarjalinkin välinen 128 kbit/s. Tuloksen hitaalle vasteajalle löytyy selitys aikaisemmista tutkimuksista (Effects of Internet Performance on Web Response time [9]), jonka mukaan HTTP:n vasteaika on vähintään kaksinkertainen minimaaliseen *ping*-ajan RTT:hen. Tämä selittyy sillä, että minimaalinen TCP-tapahtuma, kuten HTTP-GET, vaatii kaksi RTT:tä, joten tulos vastaa odotettua [10].

Toisena vasteajan testinä oli muuten samanlainen testi, mutta lisäksi verkkoa rasitettiin 64 kbit/s Iperf-ohjelman avulla. Siitä saatiin kuvan 7 mukaiset tulokset



Kuva 7. PasTmon mittaus 2.

Kuvasta voi selkeästi päätellä, että vasteaika on suunilleen tuplaantunut edellisestä, joten tulokset olivat odotetunlaiset.

6.2 Verkon läpäisykyvyn mittaus

Iperf on hyvin yksinkertainen ohjelma käyttää. Sen käynnistämiseen palvelinpäähän tulee syöttää seuraava komento:

iperf –s

Tämän jälkeen ohjelma jää odottamaan asiakaskoneelta tulevaa yhteyttä, joka asetettiin seuraavanlaisesti:

iperf –c [kohde IP-osoite]

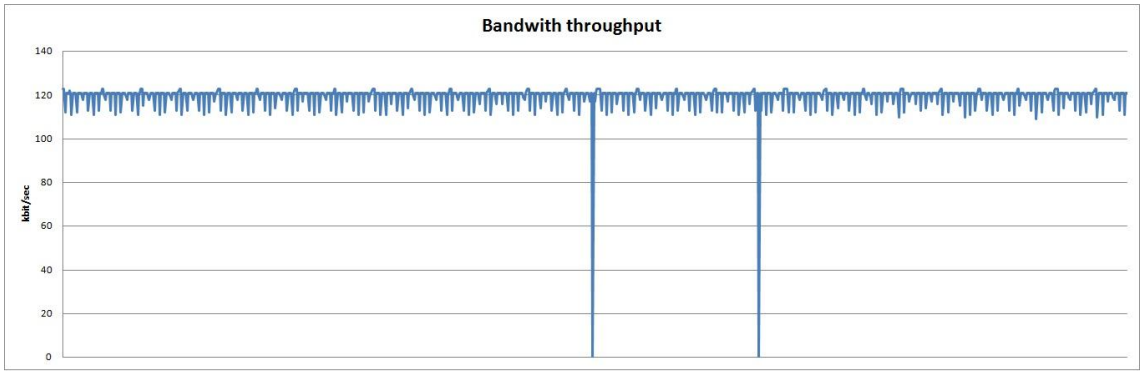
Näillä parametreillä iperf alkaa tehdä verkon läpäisykyvyn mittausta.

Seuraavana testinä verkolle tehtiin läpäisykykytesti, joka mittaa verkon todellista siirtonopeutta. Tätä testiäkin tehtiin kahdentoista tunnin verran Iperf-ohjelmalla. Lyhyehkö mittaustulos ajanjaksolta näyttää seuraavanlaiselta:

```
[1844] local 10.0.0.2 port 5001 connected with 192.168.1.2 port 2338
[ ID] Interval       Transfer     Bandwidth
[1844]  0.0-11.2 sec   152 KBytes   111 Kbits/sec
[1840] local 10.0.0.2 port 5001 connected with 192.168.1.2 port 2340
[ ID] Interval       Transfer     Bandwidth
[1840]  0.0-11.4 sec   168 KBytes   121 Kbits/sec
[1848] local 10.0.0.2 port 5001 connected with 192.168.1.2 port 2341
[ ID] Interval       Transfer     Bandwidth
[1848]  0.0-11.3 sec   168 KBytes   122 Kbits/sec
[1844] local 10.0.0.2 port 5001 connected with 192.168.1.2 port 2342
[ ID] Interval       Transfer     Bandwidth
[1844]  0.0-11.2 sec   168 KBytes   123 Kbits/sec
[1840] local 10.0.0.2 port 5001 connected with 192.168.1.2 port 2344
[ ID] Interval       Transfer     Bandwidth
[1840]  0.0-11.2 sec   168 KBytes   123 Kbits/sec
[1860] local 10.0.0.2 port 5001 connected with 192.168.1.2 port 2347
[ ID] Interval       Transfer     Bandwidth
[1860]  0.0-12.1 sec   168 KBytes   113 Kbits/sec
[1844] local 10.0.0.2 port 5001 connected with 192.168.1.2 port 2348
[ ID] Interval       Transfer     Bandwidth
[1844]  0.0-11.3 sec   168 KBytes   121 Kbits/sec
[1840] local 10.0.0.2 port 5001 connected with 192.168.1.2 port 2350
[ ID] Interval       Transfer     Bandwidth
[1840]  0.0-11.4 sec   168 KBytes   121 Kbits/sec
```

Testi osoitti, että verkon siirtonopeus vaihteli välillä 111 kbit/s - 123 kbit/s, mikä oli oletettu tulos verkon teoreettisen maksiminopeuden 128 kbit/s kannalta.

Kahdentoista tunnin mittaustuloksien numeraaliset arvot syötettynä Exceliin esitetään kuvassa 8.

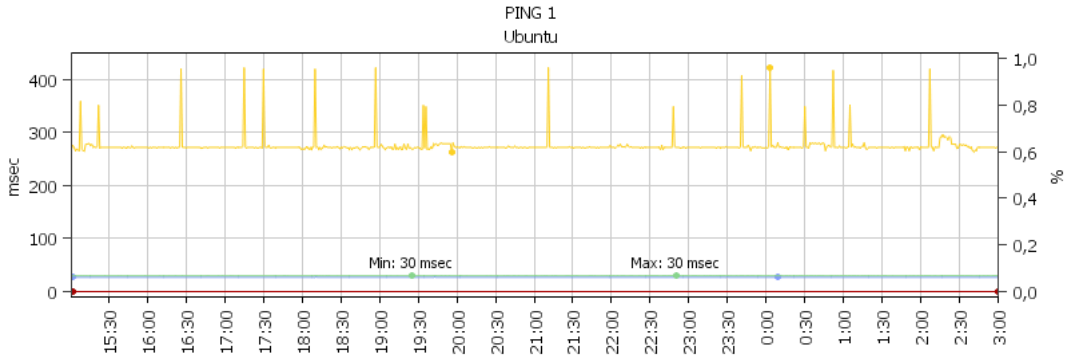


Kuva 8. Verkon läpäisykyky kahdentoista tunnin ajalta

Kuvasta voidaan havaita, kuinka tasaisena verkon läpäisykyky mittauksen aikana pysyi, koska verkossa ei ollut mitään muuta liikennettä samaan aikaan. Kahdessa eri kohdassa tulos kuitenkin putosi 0 kbit/s:iin, jonka aiheutti mittauspakettien hävikki eli packet loss.

6.3 Käytettävyysmittaus

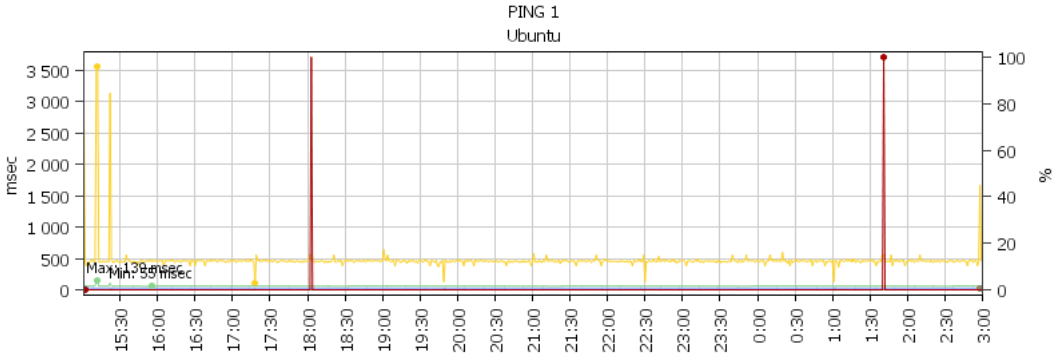
Seuraavana vuorossa oli käytettävyysmittaukset PRTG-monitorointityökalulla. Testissä simuloitiin VoIP-liikennettä ICMP-pakettien avulla. Simulointi tapahtui niin, että ohjelma lähetti dataa noin 90 kbit/s tietokone2:lta tietokone1:lle, jotta kaistan kuormitus olisi lähellä normaalia VoIP-puheluista syntyvää liikennettä. Yleisimmin VoIP-puheluissa käytetään G7.11-koodekia, joka luo liikennettä Ethernet-lähiverkossa noin 87,2 kbit/s normaaliasetuksilla. [11] Tulokset olivat kuvan 9 laisia.



Kuva 9. PRTG-mittaus 1.

Kuvasta voidaan havaita, että vasteaika kohoaa mittauksien aikana kymmenen kertaa yli 400 ms:n, mikä tarkoittaa, että yhdensuuntainen vasteaika ylittää 200 ms:n, joka on yli SLA-rajojen. SLA-raja-arvoksi on valittu 200 ms, kuten kuvan 1 (s.9) lähteessä [4] mainitaan. Sininen viiva kuvassa kertoo alhaisimman *ping*in arvon mittaustilanteessa.

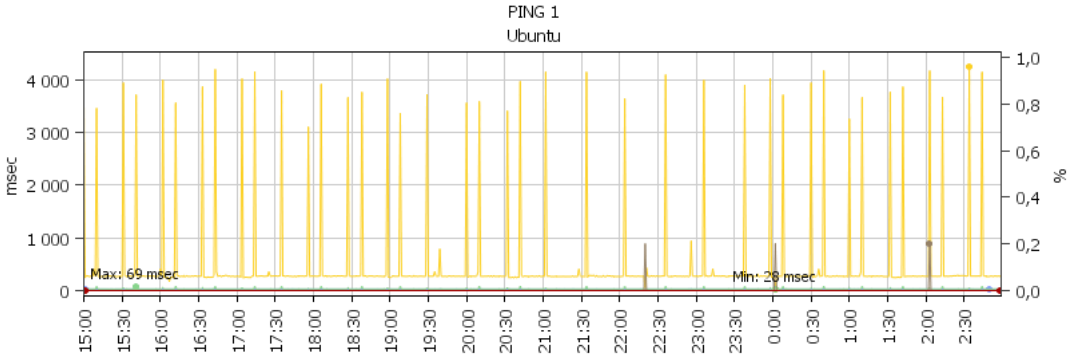
Kuva 10 havainnollistaa VoIP-liikennettä, kun kaistaa rasitetaan samanaikaisesti 64 kbit/s Iperfin avulla.



Kuva 10. PRTG-mittaus 2.

Voidaan havaita, että RTT verkossa on jatkuvasti 500 ms:n luokkaa, mikä on siis kelvotonta verkkopuheluille. Testien aikana on tullut myös pakettihävikkiä, jota punaiset viivat kuvaavat.

PRTG:llä tehtiin myös testi, jossa simuloitiin verkkopuhelua ja samanaikaisesti verkkoa rasitettiin vasteaikamittauksissa käytetyllä *wget*-komennolla. *Wget*-komennon avulla Ubuntu-koneelta lähetettiin pyyntö ladata 22 kt:n kokoinen tiedosto verkosta viidensadan sekunnin välein. Testitulokset on esitelty kuvassa 11.



Kuva 11. PRTG-mittaus 3.

Kuvasta voi selkeästi havaita, että verkkopuheluiden viive nousee niin suureksi, että puheluita ei samanaikaisesti verkossa pystytä toteuttamaan. Tähän syynä on myös se, että reitittimessä on oletusasetuksena WFQ (Weighted Fair Queuing), joka tarkoittaa

että kaikki liikenne verkossa kulkee samalla arvolla. Mikäli reitittimissä olisi määritelty QoS-jonotusmetodeja, HTTP-liikenteen olisi voinut määrittää menemään alemmalla arvolla kuin VoIP-puheluiden (tässä tapauksessa ICMP-liikenteen). Tämän avulla VoIP-liikenteen vasteajat olisivat voineet pysyä siedettävinä.

7 Yhteenveto

Yleisesti ottaen tutkimus täytti *validiteetin* vähintäänkin kohtuullisesti johtuen siitä, että työkalut mittaustuloksien saamiseen valittiin hyvin tarkasti ja huolellisesti, jotta tulokset olisivat riittäviä. *Reliabiliteetin* osalta puolestaan työssä olisi voinut olla parantamisen varaa, koska esimerkiksi VoIP-mittausta ei pystytty täysin simuloimaan reaalisten olosuhteiden mukaisesti.

Käytännön osalta työn tavoitteena oli tarjota empiirisesti tutkittu ja toimivaksi varmistettu menetelmä verkon suorituskyvyn mittaamiseen ilmaiseksi saatavilla ohjelmilla, miltä osin työ onnistui täyttämään vaatimukset vähintäänkin tyydyttävästi.

Jatkotutkimuksien osalta olisi hyvä saada käytettävyysmittauksista tarkempia mittaustuloksia, joihon mahdollisesti jo PRTG:n uusimmilla versiolla pystytäisiinkin vaikuttamaan. Tutkimuksia olisi mielenkiintoista päästä tekemään myös kaupallisilla ohjelmilla, jotta voisi verrata niiden hyötyä ilmaisohjelmiin.

Yleisesti ottaen verkon valvonta ilmaisohjelmilla on hyvin haasteellista, koska riittäviin mittaustuloksiin tarvitsee montaa erillistä ohjelmaa. Yrityksen kannalta ei ole kovinkaan järkevää säästää näissä kustannuksissa ja valita käyttöön haasteellisia ilmaisohjelmia. Järkevämpää puolestaan olisi valita maksullisista ohjelmista sentyyppinen ohjelma, jolla pystytään täyttämään yrityksen verkon suorituskyvyn valvonnan vaatimukset kokonaisuudessaan.

Lähteet

1 Elliott, Cristopher., Maggiora, Paul., Phelps, Kent., Elliott, Chritopher., Thompson, James. 2000. Cisco Press: Performance and Fault Management.

2 Verkonhallinnan toteuttaminen. (WWW-dokumentti.) http://www.netlab.tkk.fi/julkaisut/tyot/diplomityot/611/verkonhall_toteutus.html. Luettu 13.3.2010.

3 Alanko, Timo. 2005. Johdanto: Mitä on suorituskyky (WWW-dokumentti.) http://www.cs.helsinki.fi/u/alanko/ska/tekstit/1.%20Johdanto.pdf. Luettu 13.3.2010.

4 Performance Monitoring and Measurement (WWW-dokumentti.) http://www.download-it.org/free_files/Pages%20from%20Chapter%2013%20Performance%20Monitoring%20and%20Measurement-da37d1f2e5f05b006a8d290286152821.pdf. Luettu 13.3.2010.

5 Lowekamp, Bruce. 2003. Combining Active and Passive Network Measurements to Build Scalable Monitoring Systems on the Grid. College of William and Mary.

6 Cole, R.G. & Rosenbluth, J.H. 2001. Voice Over IP Performance Monitoring. AT&T Laboratories

7 Cleary, John., Donnely, Stehpen., Graham, Ian., McGregor Anthony. & Pearson Murray. 2000 Design Principles for Accurate Passive Measurement.

8 Postel, J. 1981. RFC792 - Internet Control Message Protocol. (WWW-dokumentti.) http://www.faqs.org/rfcs/rfc792.html. Luettu 30.3.2010

9 Cottrell & Halperin. 1997. Effects of Internet Performance on Web Response time (WWW-dokumentti.) http://www.slac.stanford.edu/comp/net/wan-mon/ping/correlation.html. Luettu 30.3.2010

10 Matthews, Warren & Cottrell, Les. 2000. The PingER Project: Active Internet Performance Monitoring for the HENP Community. Stanford Linear Accelerator Center

11 Voice Over IP – Per Call Bandwidth Consumption (WWW-dokumentti.) http://www.cisco.com/application/pdf/paws/7934/bwidth_consume.pdf. Luettu 10.4.2010

12 Opinnäytetyön tukimateriaali (WWW-dokumentti.). Kajaanin AMK. http://193.167.122.14/Opari/ontTukiLuotettavuus.aspx. Luettu 11.4.2010

13 Stallings, William. 1996. SNMP, SNMPv2, and RMON, Second Edition. Reading, MA: Addison-Wesley.

14 Haikonen, Jarmo., Hlinovsky, Jan. & Paju, Antti. 2000. Verkonhallinta. (WWW-dokumentti.) http://www.netlab.tkk.fi/opetus/s38118/s00/tyot/47/. Luettu 13.4.2010.