

Tuukka Palko

**AUTOMATISOIDUN TESTIJÄRJESTELMÄN VERIFIOINTI
MOBIILIVERKKOJEN MITTAUKSIIN**

AUTOMATISOIDUN TESTIJÄRJESTELMÄN VERIFIOINTI MOBIILIVERKKOJEN MITTAUKSIIN

Tuukka Palko
Opinnäytetyö
Kevät 2015
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, langattomat laitteet

Tekijä: Tuukka Palko

Opinnäytetyön nimi: Automatisoidun testijärjestelmän verifiointi mobiiliverkkojen mittauksiin

Työn ohjaaja: Timo Vainio

Työn valmistumislukukausi ja -vuosi: Kevät 2015

Sivumäärä: 64

Työn tarkoituksena oli tehdä verifiointitestausta Aniten uudelle automatisoidulle testijärjestelmälle pitkin kehitystä, jotta voitaisiin varmistua itse ohjelmastosovelluksen sekä siinä käytettävien testitapausten oikeanlaisesta toiminnasta testijärjestelmään kuuluvien laitteiden kanssa. Kyseisen järjestelmän tarkoituksena on helpottaa 3GPP-matkapuhelinverkkojen infrastruktuurin ja itse testattavan laitteen testausta. Opinnäytetyön tilaajana oli Anite Telecoms Oy. Työ edellytti tietämystä eri järjestelmäosien toiminnasta ja niihin vaikuttavista tekijöistä.

Tavoitteena opinnäytetyössä oli todeta järjestelmän toiminta mahdollisimman hyväksi ja virheettömäksi sekä sillä suoritettujen mittaukset toistettaviksi. Käytännössä suoritettiin käyttöliittymän kautta tapahtuvan järjestelmän toiminnan varmistaminen ja kaikkien laitteisto-osien avulla testimittaukset.

Asiasanat: automaatio, testaus, 3GPP, LTE, testaussuunnitelma

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Wireless devices

Author: Tuukka Palko

Title of thesis: Verification of automated mobile network test system

Supervisor: Timo Vainio

Term and year when the thesis was submitted: Spring 2015 Pages: 64

The aim of the thesis was to make verification testing along with developing Anite's new automated test system. The objective was to make sure that the main software itself and test cases for it work correctly with all the devices and programs needed in a test system. The purpose of the test system is to simplify the testing of 3GPP mobile network infrastructure and device under test.

This thesis was done for Anite Telecoms. The project required knowledge on the functionality of different parts of the system and the factors affecting them.

The main objective was to verify that the system works as good as possible without remarkable errors and that it was capable of repeatable measurements. The testing was executed using live network in Anite's laboratory where I proceeded according to the test plan.

In the theory part of the thesis, I will go through the test model used in the project and automation overall, as well as the LTE mobile network since it was used in the task. I also presented different Key Performance Indicators that were used to estimate the efficiency of measurement results.

Keywords: Test plan, LTE, Mobile network, Testing, KPI, 3GPP

ALKULAUSE

Opinnäytetyö on tehty Anite Telecoms Oy:lle kevään 2015 aikana. Työn tarkoituksena oli verifioida automatisoidusti toimivan mobiiliverkkojen testaukseen luodun testijärjestelmän toiminta.

Haluan kiittää Development Manager Seppo Salosta opinnäytetyön mahdollistamisesta ja kaikkia työkavereita sekä eritoten projektiryhmää avunannosta. Haluan myös kiittää koulun puolelta opinnäytetyön ohjaajana toiminutta yliopettaja Timo Vainiota sekä kielenohjauksesta lehtori Tuula Hopeavuorta.

Oulussa 28.5.2015

Tuukka Palko

SISÄLLYS

TIIVISTELMÄ	1
ABSTRACT	2
ALKULAUSE	3
SISÄLLYS	4
SANASTO	6
1 JOHDANTO	7
2 TESTAUSMALLI	8
2.1 Testauksesta yleisesti	8
2.2 Testausmallin tyypit	9
2.2.1 Yksikkötestaus	10
2.2.3 Integroititestausta	12
2.2.4 Järjestelmätestaus	12
2.3 Verifiointi ja validointi	13
3 AUTOMAATIO TESTAUKSESSA	14
4 MATKAPUHELINVERKOT	16
4.1 Rakenne	16
4.2 LTE	18
4.2.1 Arkkitehtuuri	19
4.2.2 Tekniikat	20
5 TESTATTAVA JÄRJESTELMÄ	24
5.1 Nemo Outdoor ja Nemo Analyze	25
5.2 Suorituskykyilmaisimet	26
5.2.1 RSRP (Reference Signal Received Power)	27
5.2.2 RSRQ (Reference Signal Received Quality)	27
5.2.3 SNR (Signal to Noise ratio)	28
5.2.4 CQI (Channel Quality Indicator)	29
5.2.5 Data Throughput	29
5.3 Laitteet ja ohjelmistot	31
6 TESTAUSSUUNNITELMA	32

7 TESTAUKSEN SUORITUS	34
7.1 Yksikkötestit	34
7.2 Integrointitestit	40
7.2.1 Kytöntöjen, emulaation ja testiskriptin teko	40
7.2.2 Automaattisten toimintojen testaus mittauksen aikana	44
7.2.3 Mittauksien toistettavuus	50
7.2.4 Valmiiden testitapausten verifiointi	52
7.3 Järjestelmätestit	54
7.3.1 Kuormitustesti	54
7.3.2 Toipuvuustesti	54
8 LOPPUTULOKSET	56
9 YHTEENVETO	63
LÄHTEET	64

SANASTO

3GPP	3rd Generation Partnership Project eli organisaatio, joka luo teknisiä määrittelyjä 3G-järjestelmille
CQI	Channel Quality Indicator eli kanavan laadun ilmaisin
COM	Communication port eli sarjaportti
DUT	Device Under Test eli testauksessa käytettävä laite
eNodeB	Evolved Node B, joilla tarkoitetaan tukiasemia LTE-järjestelmässä.
GPIO	General Purpose Interface Bus
LAN	Local Area Network eli lähiverkko
LTE	Long Term Evolution eli neljännen sukupolven matkapuhelinteknologia.
MIMO	Multiple-Input and Multiple-Output eli moniantennitekniikka
OFDM	Orthogonal Frequency Division Multiplex on yksi LTE:n tekniikoista
RAN	Radio Access Network eli radioliityntäverkko
RSRP	Reference Signal Received Power eli vastaanotettu voimakkuus
UE	User Equipment eli päätelaite, jota loppukäyttäjä käyttää viestityksessä
USB	Universal Serial Bus eli USB-portti
VDT	Virtual Drive Testing eli virtuaalinen ajotestaus

1 JOHDANTO

Opinnäytetyön tarkoituksena oli verifioida automatisoidusti toimivan testijärjestelmän toiminta eli varmistaa, että käyttöliittymä vastaa sille asetettuja rakenne- ja toimintovaatimuksia kaikkien järjestelmäosien kanssa käyttäen oikeaa verkkoa. Tärkeää oli myös verifioida, että testijärjestelmäkokonaisuus pysyy käytettäessä stabiilina ja mahdollistaen toistettavat mittaustulokset. Mittauksia tehtiin LTE-teknologialla.

Järjestelmän tarkoituksena on helpottaa 3GPP-matkapuhelinverkon infrastruktuurin ja itse testattavan laitteen testausta käyttäen oikeaa tukiasemaa. Sen tärkeimpänä tavoitteena on tarjota automaattisesti tehtäviä suorituskykytestejä. Järjestelmä parantaa langattoman testauksen tehokkuutta tehtäessä testauksia laboratorio-olosuhteissa. Siinä yhdistyvät Prosim-kanavaemulaattorituotteet sekä Nemon verkon valvonta-, suorituskyvyn mittaus- ja analysointiratkaisut.

Työ toteutettiin tekemällä testaus suunnitelma, jonka mukaan työssä edettiin ja suoritettiin testit mittajärjestelmän verifiointiksi.

Opinnäytetyön mahdollisti Anite Telecoms Oy, joka on yksi johtavista testaus- ja mittausratkaisujen toimittajista kansainvälisillä langattomilla markkinoilla. Yrityksellä on yhteensä yli 500 työntekijää 14:ssä eri toimipisteessä ympäri maailman ja sen päämaja sijaitsee Englannissa.

2 TESTAUSMALLI

2.1 Testauksesta yleisesti

Omien havaintojen ja kokemusten pohjalta voisi sanoa, ettei ilman testausta mistään tuotteesta voi saada hyvää ja laadukasta, etenkään ohjelmistosta. Testauksella mahdollistetaan virheiden löytyminen ja siten asiakkaiden tyytyväisyyden kasvu, mikä taas vaikuttaa itse yrityksen talouteen ja imagoon. Testauksella näin ollen yleensäkin tarkoitetaan tuotteen taikka palvelun laadun, luotettavuuden sekä vaatimusten tarkastamista. Tärkeää on myös varmistaa, ettei järjestelmän käytöstä koidu sivuvaikutuksia eli toimintoja, joita ei saisi missään tapauksessa tapahtua. Yleisesti on huomioitava, että testaukseen kuluu suhteellisen paljon aikaa ja resursseja, vaikka se ei ajatuksena tuntuisi siltä, koska testaus kokonaisuutena pitää sisällään virheiden havaitsemista, raportoimista, niiden aiheuttajien löytämistä sekä myös niiden korjaamista. Tulee kumminkin huomioida se, ettei järjestelmän täydellistä toimintaa voida taata kaikissa käyttötilanteissa, koska ohjelmistolla tai järjestelmällä on ääretön määrä erilaisia tiloja ja suorituspolkuja.

Yleensäkin järjestelmään suunniteltu ja tehty testiajo voidaan todeta onnistuneeksi, kun siitä löytyy virhe, jota ei aiemmin olla löydetty. Tehtäessä itse testitapauksia ohjelmistolle tulee testin olla sellainen, jolla hyvällä todennäköisyydellä löydetään virhe. Testauksessa usein käytettyjä termejä on muun muassa virhe tai bugi, jolla tarkoitetaan poikkeamaa vaatimusmäärittämisistä ohjelmassa. Vika aiheutuu jonkin virheellisen toiminnon ajamisesta, häiriö taas järjestelmän ulkoisen toiminnon viasta. (4.)

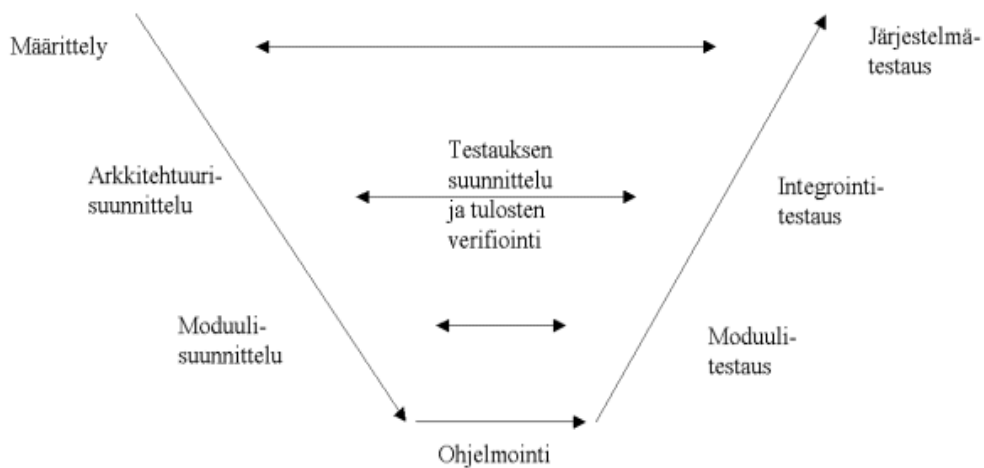
Testauksen suoritus tulee aloittaa testaussuunnitelman luomisella, johon määritellään testauksessa eteneminen vaatimusmäärittäysten perusteella. Suunnitelman pohjalta luodaan testitapauksia ja määritellään kriteerit, milloin testi voidaan todeta onnistuneeksi. Kaikki virheet kuvauksineen, jotka kehityksen aikana löydetään, niiden löytymisen ja korjauksen ajankohdat sekä kehitysversiot on syytä kirjata ylös. Tämä siksi, koska ne ovat järjestelmän

tärkeimmistä dokumentaatioista, joilla voidaan todeta milloin testauksia on suoritettu tarpeeksi. (3.)

2.2 Testausmallin tyypit

Järjestelmän kehityksessä on erittäin oleellista tehdä testausta alusta alkaen ihan yksittäisistä toiminnoista järjestelmä kokonaisuuden testaamiseen. Testaus ei useastikaan ole aivan yksinkertaista, sillä eri vaiheet ja niissä eteneminen järjestelmän kehityksessä tuovat omat haasteensa ja ongelmansa. Seuraavana esitellään testauskyyppit.

Järjestelmätestausta suoritetaan useasti V-mallin avulla, jossa testaukset suunnitellaan sitä vastaavalla kehitystasolla (kuva 1.) Malli muodostuu yksikkö-, integrointi- ja järjestelmätestaustasoista. Yleensäkin yksikkötestauksesta löydetyt virheet johtuvat ohjelmointivirheistä, integrointivaiheesta löydetyt suunnitteluvirheistä ja järjestelmävirheet pahimmassa tapauksessa alkumäärittelyvirheistä. Mallin mukaan yksikkötestaus tulee suunnitella yksikkötestausvaiheessa, integrointitestausta arkkitehtuurisuunnittelun yhteydessä ja järjestelmätestausta määrittelyvaiheessa. (1.)



KUVA 1. V-malli (5.)

2.2.1 Yksikkötestaus

Kun ohjelmansovellus halutaan mahdollisimman virheetömäksi tulee se yleisesti testata seuraavaksi esitellyillä malleilla, jotta se saadaan käytyä läpi mahdollisimman laajasti. Työssä käyttöliittymä käydään läpi musta- ja harmaalaatikkomallia käyttäen.

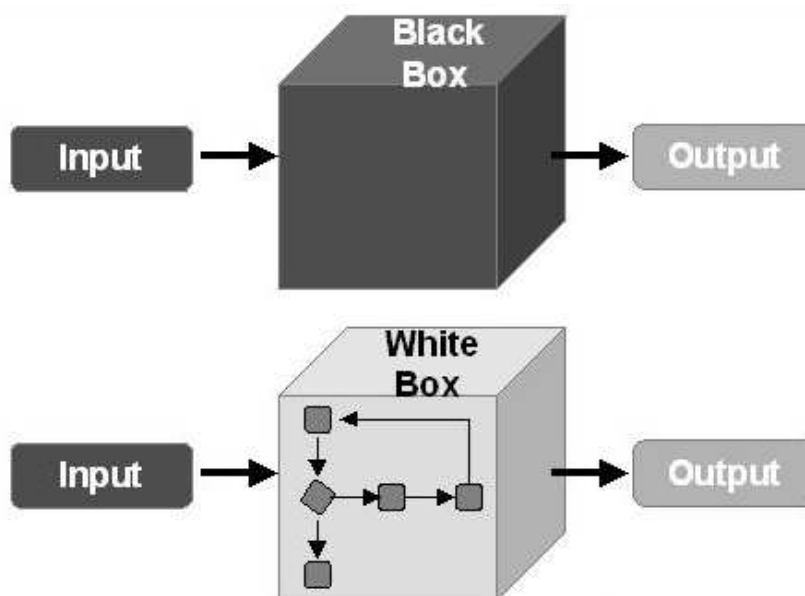
Lasilaatikkotestaus

Yksikkötestaus eli moduulitestaus suoritetaan osittain lasilaatikkomallin (white box) mukaisesti, jossa testaaja tuntee koodin. Hänen tulee tarkkailla ohjelmakoodin rakennetta sekä yksittäisten moduulien toimintaa ja etsiä niistä virheitä. Tässä mallissa käydään koodi yleensä läpi katselmoinnissa muiden työryhmän jäsenten kanssa. Ohjelmoijan tulee kiinnittää koodissa huomiota muun muassa tieto- ja ohjausrakenteisiin, rajapintoihin ja suorituspolkuihin. Testauksen yleensä suorittaa tässä mallissa siis ohjelmoija itse, joka tuntee ohjelmakoodirakenteet. (3.)

Mustalaatikkotestaus

Yksikkötestauksessa käytetään myös mustalaatikkomallia (black box) jota yleisimmin kutsutaan funktionaaliseksi testaukseksi. Testausta siinä suoritetaan suoraan sovelluksen käyttöliittymän (Graphical User Interface) kautta. Mallissa kiinnitetään huomiota vain ohjelman ulkoisiin toimintoihin ilman, että tiedetään koodista mitään, ja verrataan toimintoja määrittäisiin ja vaatimuksiin. (Kuva 2.) Käytännössä syötetään jokin arvo ja katsotaan, että ulostulosta tulee haluttu tulos. Tyypillisiä testattavia ulkoisia toimintoja ovat muun muassa erilaiset käyttöliittymäkomponenttien toiminnan testaukset ja syötetestit, kuten oikeat ja virheelliset tietoyhdistelmät, merkkikombinaatiot, tyhjät kentät ja aloitus- sekä lopetustoimintojen testaus. Myös erilaiset päivitystestaukset, esimerkiksi tietueiden lisäämiset ja poistamiset ovat yleisiä. (3.)

Tämä testauksen vaihe yleisestikin suoritetaan ennen järjestelmäosien integroimista suuremmaksi kokonaisuudeksi, koska täten kyetään tunnistamaan helpommin mistä jokin virhe on peräisin, kun muut järjestelmän osat eivät ole käytössä. Testausmallia voisi kutsua myös rajapintatestaukseksi, koska siinä usein testataan myös rajapintojen välisiä toimintoja. Myös virheen korjaus on nopeampaa taloudellisempaa, koska mitä aikaisemmin virheet löydetään ja saadaan korjatuksi, sitä vähemmän tulee kustannuksia. Mustalaatikkotestaus suositellaan tehtävän testaajalla, joka ei tunne sisäisiä koodirakenteita, ainoastaan vain sen miten tuotteen tulisi toimia ulkoisesti. (1.)



KUVA 2. Musta- ja lasilaatikon periaate (3.)

Harmaalaatikko

On myös olemassa malli, jota käytetään musta- ja lasilaatikkomallin välissä, hyödytäen molempien mallien eri osia. Tämä on nimeltään harmaalaatikko (Grey box). Siinä pyritään asettumaan loppukäyttäjän rooliin ja sitä kautta miettimään, mitä ja miten ohjelmaa kannattaa testata. Sen avulla voidaan testata erityisesti ohjelman kriittisiä kohtia antamalla esimerkiksi vääränlaisia syötteitä. (1.)

2.2.3 Integrointitestaus

Integrointitestauksen vaiheessa nimensä mukaisesti integroidaan eli yhdistetään järjestelmäkokonaisuuden osia eli moduuleita yhteen. Tämän vaiheen avulla pyritään todentamaan, että käytettävät eri järjestelmäosat toimivat yhdessä oikein ja että niiden välinen tiedonvälitys toimii. Osien integraatiota tulee verrata alussa tehtyihin vaatimus- ja teknisiin määrittelyihin, että tietyt toiminnot tapahtuvat, niin kuin oli suunniteltukin. Osittain integraatiotestausta suoritetaan myös jo yksikkötestauksen yhteydessä, koska vastaan voi tulla yksittäisten moduulien toimintaan liittyviä virhetilanteita, kun eri osat toimivat yhteistyössä. Virheen voi aiheuttaa esimerkiksi yhteensopivuusongelma käyttöliittymän ja yhdistetyn matkapuhelimen tai toisen järjestelmässä käytettävän ohjelman välillä. Voi ilmetä poikkeustilanteita, joita ei ole välttämättä suunnitteluvaiheessa otettu huomioon. (1.)

Integrointivaiheessa tulisi alussa kiinnittää huomiota kaikista virhealttiimpiin kohtiin, jotta niistä löytyvät virheet saataisiin poistetuksi. Koska usein testataan suurta järjestelmää, joka koostuu useista osista, tulee sitä läpikäydessä miettiä mihin osiin kiinnitetään eniten huomiota ja käytetään resursseja sekä milloin voidaan todeta jokaisen moduulin osalta, että ne toimivat yhdessä niin hyvin, että voidaan siirtyä tarkastelemaan kokonaisuutta.

2.2.4 Järjestelmätestaus

Tämän testaustyypin tarkoituksena on verrata järjestelmäkokonaisuutta alussa tehtyihin määrittelydokumentaatioihin. Tärkeää osaa tässä vaiheessa näyttelevät ei-toiminnalliset testit. Tämmöisiä ovat suorituskäyttöön liittyvät testit, kuten erityyppiset järjestelmän kuormitustestit, että miten se kestää käsitellä suuriakin määriä dataa. Käytännössä järjestelmä vietään sen ääri rajoille. Sen testaus etenkin automatisoidussa järjestelmässä on erityisen tärkeää, koska automaatiolla juuri pyritään massasuoritukseen, jota yksittäinen henkilö ei kykene normaalisti suorittamaan nopeassa ajassa. Järjestelmästä tulee varmistaa yleensä, että miten se toipuu siihen kohdistuneista äkillisistä

ongelmista, kuten tärkeän osakokonaisuuden irrottamisesta suorituksen aikana. Käytettävyys on myös tärkeällä sijalla, koska loppukäyttäjä haluaa mahdollisimman helppokäytettävän tuotteen. Täten sen läpikäyminen on myös oleellista.

2.3 Verifiointi ja validointi

Kun järjestelmän avulla tehtävät toiminnot halutaan varmistaa, ei tule unohtaa verifiointia ja validointia. Verifiointi tarkoittaa järjestelmän vaatimusten toteamista oikein kehitetyiksi ja validointi sitä, että järjestelmä vastaa loppukäyttäjän vaatimuksia. (6.) Tätä tarkastelua tulee tehdä testauksen ohella V-mallin mukaisesti pitkin kehitystä.

Järjestelmän käytön verifiointissa tulee pitkin matkaa tehdä regressiotestausta (regression testing). Siinä aikaisemmin virheen antanutta testiä ajetaan uudelleen ja varmistetaan, ettei se enää esiinny uudestaan, kun rakenteeseen on tehty muutoksia. (5, s. 49.)

3 AUTOMAATIO TESTAUKSESSA

Puhuttaessa automaatiosta tarkoitetaan täällä itsestään toimivaa tapahtumaa, ilman että ihminen ohjaa taikka suorittaa sitä. Testauksen tarkoituksena yleensäkin on varmistaa tuotteen laatu, esimerkiksi löytämällä mahdolliset virheet ohjelmistosta. (6.) Kun testaus tehdään automatisoidusti, tarkoitetaan sillä, että testit, jotka testaaja olisi itse tehnyt, suoritetaan nyt koneen tekemänä. (7.) Tämä edellyttää sitä, että itsekseen toimiva testijärjestelmä kykenee toistamaan testitapaukset samanlaisina. Suurimpana ja tärkeimpänä hyötynä on se, että kyetään ajamaan suuria testimääriä lyhyemmässä ajassa ja siten myös pienentämään kustannuksia ja lisäämään testauskattavuutta.

Testaus voidaan jakaa kahteen osaan: ihmisen suorittama testaus ja koneen suorittama testaus. Koska testauksella pyritään matkimaan asiakkaan käyttötapoja, on erittäin oleellista, että ihminen suorittaa testejä. Pääasiassa siksi, koska ihminen kykenee käyttämään tuotetta oikein, virheellisesti ja arvaamattomasti. Vain ihminen pystyy täten jäljittelemään ihmisten käyttötapoja käytettäessä tuotetta ja tällä tavalla loppukäyttäjän tyytyväisyys paranee. (8.)

Testiautomaation hyödyllisyys tulee esille erityisesti silloin, kun halutaan tehdä useita toistoja testeille. Kun halutaan tehdä toistoja, ihminen ei ole siihen hyvä vaihtoehto, koska se on loppupeleissä hyvin yksitoikkoista toimintaan. Esimerkiksi jos tehdään yli päivän mittaisia toistoja niin täten ihminen nopeasti tylsistyy, varsinkin jos motivaatio ei ole kohdallaan. Samalla uhaksi myös tulee testaajan tekemien virheiden lisääntyminen. Siksi tällaisissa tilanteissa on hyvä käyttää testiautomaatiota eli konetta, joka ei kyllästy toistoihin toisensa perään ja joka mahdollistaa myös testienteon nopeuden. Toistettavuudella täten pyritään varmistamaan, että järjestelmä tekee samat asiat joka kerta luotettavasti samanlailla. (8.)

Kun tehdään toistoja mittauksille, tulee usein esille R&R-niminen menetelmä, joka muodostuu sanoista repeatability (toistettavuus) ja reproducibility (uusittavuus). Menetelmän yksinkertaisemman version, 1 Gage-tyypin avulla

pyritään havaitsemaan mittausvirheitä järjestelmillä mitatuista tuloksista suorittamalla useita toistoja ja arvioimaan mittausten poikkeamia (bias). Kun tehdään mittauksia, niin tulee huomioida, että mittauksissa ilmenee aina vaihtelua. Mikäli sitä esiintyy liikaa suhteessa toleranssiin, on mittaus tällöin liian vaihteleva eli epävakaata verrattuna siihen, mihin se on tarkoitettu. Mikäli menetelmää halutaan käyttää laajemmasti, voidaan käyttää mittausten suorituksessa useita tekijöitä eli operaattoreita, joiden tuloksien toistettavuuksia verrata. (28.)

Automatisoitu testaus mahdollistaa myös suorituskyvyn ja kestävyuden varmentamisen. Suorituskyvyllä tarkoitetaan täten sitä, miten järjestelmä suoriutuu erilaisissa tilanteissa. Voidaan esimerkiksi kuormittaa järjestelmää järjettömän suurilla datamäärillä tai pitämällä sitä käynnissä erittäin pitkiä aikoja yhtäjaksoisesti ja katsomalla onko sillä haitallista vaikutusta mihinkään järjestelmäosaan tai ylipäätensä lopputuloksiin.

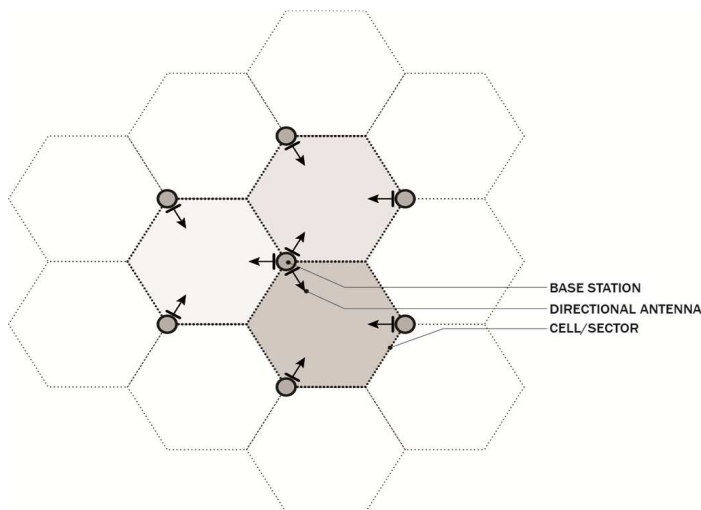
Työssä verifioitava järjestelmä rakentuu pääsovelluksesta, jonka avulla automatisoidusti ohjailaan Nemo Outdoor -työkalua itse mittausten suorittamisessa, Prosim-kanavaemulaattoria emulaation ajamisessa sekä Nemo Analyze-työkalua tulosten analysoinnissa. Automaation hyöty järjestelmissä perustuu siihen, että pysytään kasaamaan suuria testimääriä muutamalla painalluksella ja ajamaan ne ohjelmoidusti itseksensä alusta loppuun. Manuaalisesti tämä olisi suhteellisen pitkäkestoinen projekti, koska eri laitteista ja ohjelmista pitäisi käynnistää ja sulkea erikseen niiden toimintoja. Myöskin virheiden syntyminen olisi todennäköisempää.

Järjestelmä mahdollistaa ajokertojen eli iteraatioiden määrittelemisen, jolloin pystytään tekemään helposti toistoja yksittäisille testeille. Vaikka automaatio tekeekin suurimman osan työstä, ei sillä kyetä kumminkaan kaikkea tekemään, sillä jonkun tulee tutkia ja arvioida testeistä saadut tulokset, että ovatko ne riittävän hyviä verrattuna odotettuihin arvoihin. Mikäli mittauksissa tulee virheitä tai muita omituisuuksia, tarvitaan ihmistä paikantamaan, raportoimaan ja korjaamaan ne.

4 MATKAPUHELINVERKOT

4.1 Rakenne

Ympäristöä, jossa päätelaitteet kommunikoivat keskenään kutsutaan matkapuhelinverkoksi. Verkko on rakenteeltaan solumainen (kuva 3), minkä takia siitä usein käytetään myös nimitystä soluverkko, jossa jokaisessa solussa toimii omia tukiasemia ja muita runkoverkon osia, joilla toiminta-alue kyetään kattamaan. Radioyhteys solujen sisällä liikkuviin päätelaitteisiin eli matkapuhelimiin luodaan juurikin tukiasemilla. (9, s. 19.) Solujen kokoon ja muotoon vaikuttavat lähettimen teho, taajuusalue, maastontyyppi sekä tukiaseman ja antennien välillä käytetyt suuntakuviot (10.)



KUVA 3. Soluverkko (14.)

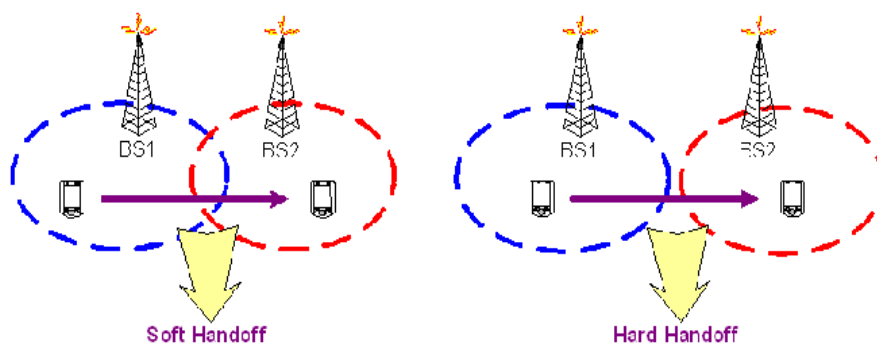
Matkapuhelinverkko muodostetaan kolmesta komponentista: päätelaitteesta (UE), runkoverkosta (CN) sekä radioliityntäverkosta (RAN). Päätelaitteella tarkoitetaan käytännössä matkapuhelinta, jonka avulla itse kommunikointi tietoliikenneyhteyden yli tapahtuu. Radioliityntäverkot koostuvat tukiasemista, tukiasemaohjaimista ja niiden välisistä rajapinnoista. Solujen sisäinen viestintä,

määritetään tukiasemilla, kuten signaalin koodaus ja modulaatio.

Matkapuhelimen ja radioliityntäverkon välillä tapahtuva radioliikenne kulkee kahta kanavaa pitkin, matkapuhelimesta tukiasemaan (ylälinkki) ja tukiasemasta matkapuhelimeen (alalinkki). Runkoverkko toimittaa liikenteen radioliityntäverkon ja internetin tai julkisen matkapuhelinverkon välillä. Sen tehtävänä on muun muassa datan autentikointi eli todennus sekä ohjaus. (9, s. 21.)

Yksi erittäin tyypillinen tapahtuma soluverkossa on handover eli yhteysvastuun vaihto tai tukiaseman vaihdoksikin kutsuttu. (Kuva 4.) Sitä käytetään, kun käytetyn solun kanavien määrä maksimoituu tai kun uuden solun signaali on voimakkaampi.

Kun puhelunsiirto tapahtuu solusta toiseen ja varataan uusi taajuus ja aikaväli, kutsutaan tätä kovaksi yhteysvastuun vaihdoksi (hard handover). Kun uuden solun yhteyden tarkistus on suoritettu onnistuneesti, katkaistaan yhteys edelliseen soluun. On olemassa myös pehmeäksi kutsuttua yhteyden vaihtoa (soft handover). Siinä hyödynnetään solujen päällekkäisyyttä, jolloin saadaan käyttöön enemmän kanavia alueella. Päätelaitteet kykenevät olemaan kytköksissä useaan soluun, jossa itse solun vaihto saadaan aikaiseksi käyttämällä aina yhtä aktiivisena olevaa yhteyttä. (12.)



KUVA 4. Eri tukiasemavaihdot (11.)

Matkapuhelinverkoja on kehitetty vuosien mittaan käyttäen erityyppisiä tekniikoita. Ne on jaoteltu niiden perusteella useisiin generaatioihin. Verkkojen kehitys alkoi aikoinaan analogisista NMT:stä ja ARP:stä, joissa alussa pyrittiin

siirtämään vain puhetta käyttäen piirikytkentäistä yhteyttä. Heikkoa siitä teki se, että tilaajien välillä olevat yhteys oli varattu koko puhelun keston ajan. Sitten huomattiin, että matkapuhelinverkossa tulee kyetä tiedonsiirtoon, kuten internetin käytössä, jossa on käytössä pakettikytkentäinen yhteys. Sitä mukaan yhteyksien laatua alettiin parannella. GSM-verkossa pakettikytkentäistä yhteyttä alettiin käyttämään ensimmäisenä. Tekniikkoina siinä käytettiin GPRS- (General Packet Radio Service) ja EDGE-tekniikkaa (Enhanced Data rates for Global Evolution). Kun sen nopeus ei enää tahtonut riittää, oli pakko taas kehittää jotain uutta, jolloin syntyi UMTS-verkoissa käytössä oleva rajapinta W-CDMA (13, s. 298–300.) Tänä päivänä suosituimpia tekniikoita ovat toisen sukupolven GSM-, kolmannen sukupolven W-CDMA- sekä neljännen sukupolven LTE-teknologiat.

Käytettävät matkapuhelinjärjestelmät voidaan jaotella näin ollen neljään sukupolveen, ja viideskin on jo tulossa, jolla tulisi päästä taas entistä suurempiin nopeuksiin. Työssä käytetään LTE-teknologiaa, jota yleisesti kutsutaan myös 4G:ksi ja täten siitä kerrotaan seuraavaksi.

4.2 LTE

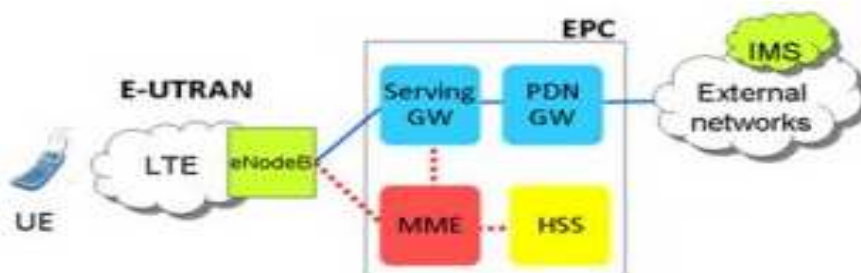
LTE-teknologia (Long Term Evolution) yhdistetään yleisesti neljännen sukupolven matkapuhelinverkoksi eli 4G:ksi. LTE on 3GPP-standardointijärjestön kehittämä teknologia. Järjestö luo siihen jatkuvasti kehittyneempiä standarijulkaisuja. Kehittyneempää LTE-versiota kutsutaan yleisesti LTE Advancedksi, joka mahdollistaa paremman suorituskyvyn spektritehokkuudelle. 3GPP-järjestö koostuu kuudesta eri tietoliikennealan järjestöstä. Näitä ovat ETSI, ATIS, ARIB, CCSA, TTA ja TTC. Verrattuna aikaisempiin matkapuhelinverkkoihin, kuten UMTS ja GSM, mahdollistaa LTE suuremmat nopeudet ja kapasiteetin datansiirrossa sekä pienemmän viivehajonnan. Datanopeuteen vaikuttaa monet seikat kuten käytettävät antennimallit, MIMO ja myöskin tietyillä ajoilla käytettävät modulointimenetelmät. (15.)

4.2.1 Arkkitehtuuri

LTE-verkossa käytetään UMTS-verkosta kehittynyttä rajapintaa, jota kutsutaan E-UTRAN:ksi (Evolved UTRAN). Se muodostuu tukiasemista (eNodeB).

LTE:hen on kehitetty runkoverkko (Evolved Packet Core) uudelleen, toisin kuin esimerkiksi UMTS-verkossa, jossa ei ollut paljon eroavaisuuksia verrattuna aikaisempaan GSM-verkkoon. Kun E-UTRAN-rajapinta ja runkoverkko (EPC) yhdistetään muodostuu EPS (Evolved Packet System), jolla mahdollistuu pakettien siirto päätelaitteen ja operaattorin kesken. (15, s. 2.)

LTE-verkon rakennetta muutettiin huomattavasti selkeämmäksi aiempiin tekniikoihin nähden (kuva 5.) Tämä mahdollistettiin poistamalla verkon välisolmuja. Radioverkko-ohjaimen (RNC) käyttö erotellaan tukiaseman ja liikkuvuudenhallintayksikön (MME) sekä yhdyskäytävän (S-GW) kesken. Runkoverkon oleelliset osat muodostetaan liikkuvuudenhallintayksiköstä ja palvelevasta yhdyskäytävästä. MME:n tehtävänä on ohjata päätelaitteiden ja istuntojen erilaisia toimintoja, kuten paikannusta ja tietoturva sekä palvelun laadun valvomista. Se on yhteydessä HSS-palvelimeen, joka sisältää kaikki tiedot päätelaitteista. E-UTRAN:n ja runkoverkon välillä tapahtuvista toimista huolehtii S-GW. Tuon yhdyskäytävän tehtäviin kuuluu muun muassa päätekäyttäjän tietojen reititys ja puskurointi katkottomasti solunvaihdon tapahtuessa. LTE-verkon arkkitehtuuriin kuuluu myös PDN, mikä on samankaltainen yhdyskäytävä, kuin S-GW:kin mutta sitä käytetään EPC:n puolella LTE-verkon ja muiden pakettiverkkojen pakettiliikenteen hallinnointiin. Arkkitehtuuriin kuuluu myös PCRF, joka huolehtii käytettävien päätelaitteiden oikeuksista käyttäjä palveluita. (16, s. 7–8.)



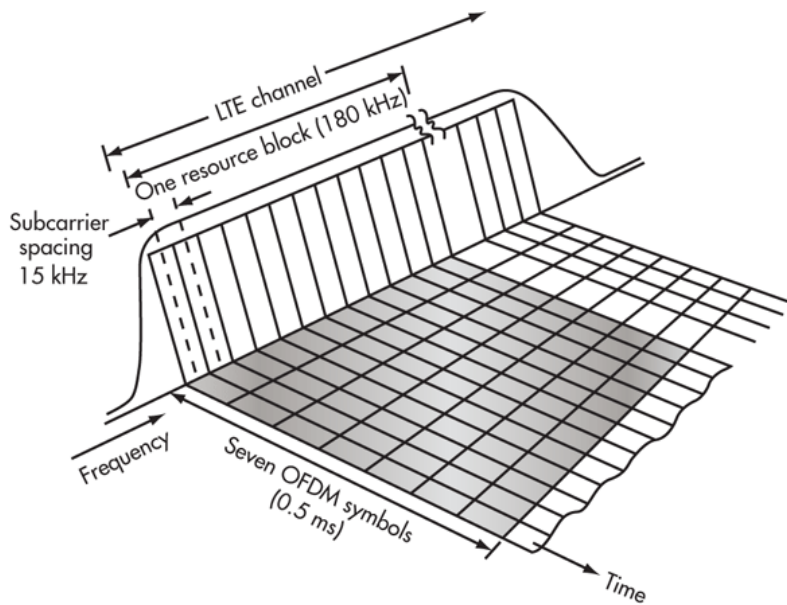
4.2.2 Tekniikat

LTE-verkko sisältää paljon erilaisia tekniikoita, mutta merkittävimpinä tekniikoina pidetään MIMO:a ja OFDMA:ta.

OFDM (Orthogonal Frequency Division Multiplex)

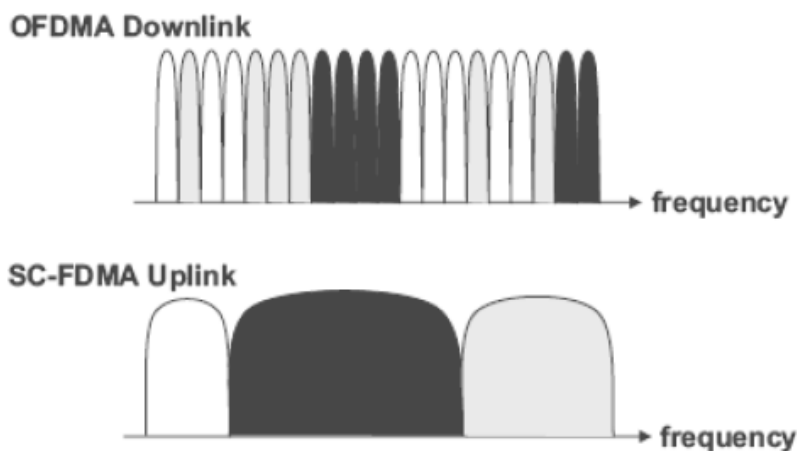
LTE:ssä käytetään OFDM-tekniikkaa, joka mahdollistaa suuremmat datansiirtonopeudet sekä hyvän sietokyvyn signaalille haitallisille tekijöille, kuten monitie-etenemisille. Alalinkin- ja ylälinkin suuntaan tapahtuvassa datansiirrossa käytetään omia modulointitapoja. Lähetteen teho ilmaistaan PAPR:lla (Peak to Average Power Ratio), jolla saadaan ilmoitetuksi tehohuiput signaalitehoon verrattuna.

Alalinkin suuntaan eli tukiasemasta päätelaitteelle käytetään OFDMA-modulointitapaa, jossa käytetään datansiirtoon monia kapeita kantoaaltoja samanaikaisesti taajuuden perusteella ilman että ne häiritsevät toisiaan. Kyseinen tapa tukee molempia taajuus- ja aikajakoista dupleksisutta. Itse modulointitekniikoina LTE:ssä käytetään QPSK-, 16QAM- ja 64QAM-tekniikoita. QPSK pitää sisällään yhtä symbolia kohti 2 bittiä, 16QAM 4 bittiä ja 64QAM 6 bittiä. Aika-taajuusakselilla voidaan havainnollistaa OFDMA:n toimintaa (kuva 6.) Siinä symboleista on koottu resurssilohko (resource block), jossa yksi lohko ilmoitetaan 180 kHz taajuusalueella, jossa seitsemän symbolia vastaa 0,5 millisenkuntia aikatasossa. Aikatasossa lähetetyt symbolit erotetaan suojavälillä eli syklisellä etuliitteellä, jossa OFDM-symbolia edeltää vastaavan symbolin loppupään osa. Tämä mahdollistaa häiriöiden syntymisen ehkäisyn niiden välillä. Sitä suurempi bittinopeus saavutetaan, mitä enemmän käyttäjä saa sille varattuja lohkon osia ja mitä suurempaa modulointitekniikkaa käytetään. (17, s. 15–17.)



KUVA 6. OFDMA:n toiminta (19.)

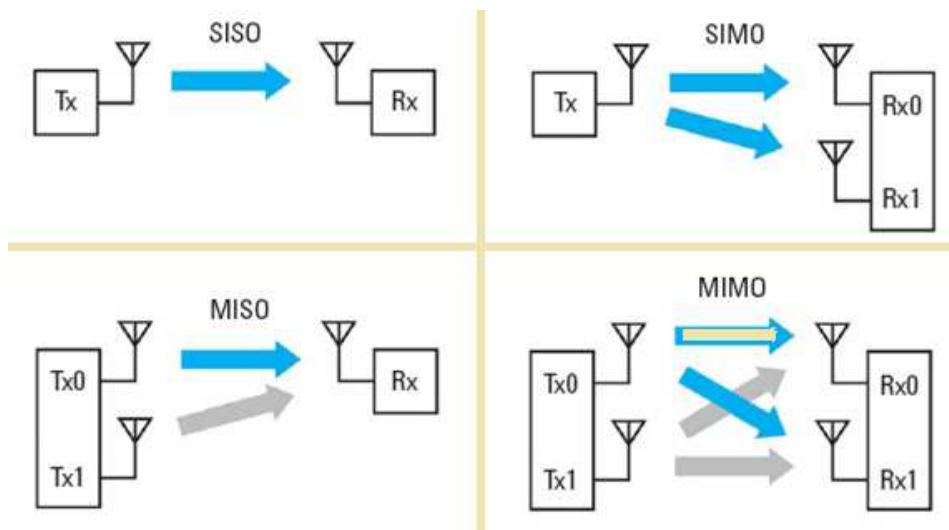
SC-FDMA (Single Carrier Frequency-division multiple access)-tekniikkaa käytetään LTE:ssä päätelaitteelta tukiasemalle päin tapahtuvassa datansiirrossa. Tiedonsiirto tapahtuu käyttäjien kesken ajan perusteella, siten, että kaikilla on käytössä sama taajuus. Siinä jokaiselle symbolille varataan kokonainen taajuuskaista.(3, s. 18) SC-FDMA-tekniikassa PAPR-arvot ovat OFDMA:han verrattuna pienempiä jokaisen modulaation kohdalla (kuva 7), eli lähetin kykenee siirtämään tietoa kovemalla teholla.(15, s. 3.)



KUVA 7. OFDMA:n ja SC-FDMA erot (20.)

MIMO (Multiple Input Multiple Output)

MIMO-tekniikassa lähetykseen ja vastaanottoon käytettyjä antennia on useita, täten sitä kutsutaan moniantennitekniikaksi. Lähetyksmenetelmiä on useita erilaisia (kuva 8). Sitä alettiin ensimmäisenä käyttää LTE-teknologiassa, jossa mahdollistui neljän antennin samanaikainen käyttö. LTE-Advanced, joka on parannus LTE-standardiin, mahdollistaa jopa kahdeksan antennin käytön. Lähetyksantennien määrällä on merkitystä, kun halutaan saavuttaa tietty tiedonsiirtonopeus tietyllä kaistanleveydellä. Tekniikalla kyetään näin saavuttamaan suuremmat tiedonsiirtonopeudet sekä tiedonsiirron luotettavuuden parantaminen.



KUVA 8. MIMO-tekniikan eri antennilähetyksmenetelmät (21.)

Shannon-Hartleyn teoreeman mukaan kaistanleveys ja signaali-kohinasuhde määrittelevät radiokanavassa käytettävän kapasiteetin. Kaistanleveyteen voidaan vaikuttaa muun muassa säätämällä lähetystehoa ja käyttämällä suurempaa moduloitintapaa, mikä toisaalta taas edellyttää laadukkaampaa SNR:ää. (9, s. 37–38.)

Suureman tiedonsiirtonopeuden saavuttamiseksi käytetään tilallista limitystä (spatial multiplexing). Tällöin signaali kulkee jokaisessa antenniparissa yksilöllisesti eri datavirtoina. Lähetyspäässä ositetaan siirrettävä tieto, minkä jälkeen se lähetetään rinnakkain useaa antennia käyttäen. Kun signaalit pääsevät vastaanottopäähän, yhdistetään ne taas omaksi datavirrakseen. (18, s. 15–16.)

Luotettavuuden parantamiseksi käytetään MIMO:ssa aika-tilakoodausta (diversity coding). Sen avulla kyetään vähentämään haittoja huonoissa olosuhteissa, jotka aiheutuvat häipymisestä. Kyseisessä koodauksessa käytetään useaa antennia lähettämään sama signaali yhdenaikaisesti. (18, s. 16.)

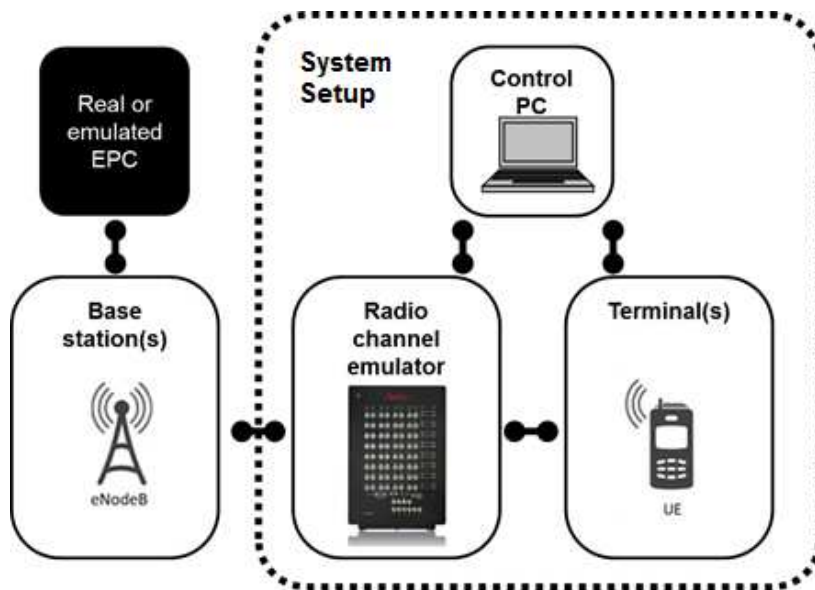
5 TESTATTAVA JÄRJESTELMÄ

3GPP-matkapuhelinverkkojen ja niissä käytettävien matkapuhelimien testaukseen kehitetty testijärjestelmä, jossa mahdollistuu oikeiden tukiasemien käyttö. Se mahdollistaa suorituskykytestienteon automatisoidusti tietokoneelta käsin, parantaen testienteon tehokkuutta laboratorio-olosuhteissa.

Järjestelmästä kehitetään samanaikaisesti kahta erilaista kokonaisuutta, joista tässä työssä testataan ainostaan oikeaa verkkoa hyödyntävää osaa, toisessa laajemmassa osassa on käytössä Anite 9000-verkkoemulaattori.

Järjestelmä muodostuu jo olemassa olevista Aniten työkaluista (Nemo Outdoor ja Nemo Analyze), joista on yhdistelemällä muodostettu uusi automatisoitu suorituskykytestaukseen tarkoitettu testityökalu. Toimivaan järjestelmäkokonaisuuteen tarvitaan lisäksi itse testattavaa matkapuhelinta verkon mittauksia varten sekä radiokanavaemulaattoria. (Kuva 9.) Prosimilla kyetään määrittelemään ajettava emulaatio eli virtuaalinen ajoreitti, joka sisältää halutunlaisen etenemisympäristön radioaallolle tukiaseman ja päätelaitteen välille. Testijärjestelmä ohjaa sitä ATE-komennoilla. Emulaattoreilla kyetään jäljittelemään radiokanavan ominaisuuksia luotettavasti ja toistettavasti. Tällaisia ovat muun muassa monitie-eteneminen, etenemisvaimennus, viivehaje, Dopplerhaje, polarisaatio ja korrelaatio. Emulaatioon määritetään useita eri parametreja, kuten taajuus, tehotasot ja kanavamallit. (29.)

Järjestelmän avulla kyetään suorittamaan erilaisia tuotekehitykseen, laadunvarmistukseen ja suorituskykyyn liittyviä mittaustestejä käyttäen oikeaa verkkoa. Tässä versiossa on käytössä ainoastaan LTE- ja UMTS-teknologiat, mutta tulevaisuudessa julkaisuissa järjestelmä tulee tukemaan useita radioteknologioita, kuten GSM, CDMA2000 ja TD-SCDMA. Järjestelmä mahdollistaa päätelaitteelle erilaiset verkon KPI-mittaukset, kuten FTP-, SFTP-, HTTP-, HTTPS-datansiirto- ja ping-testit.

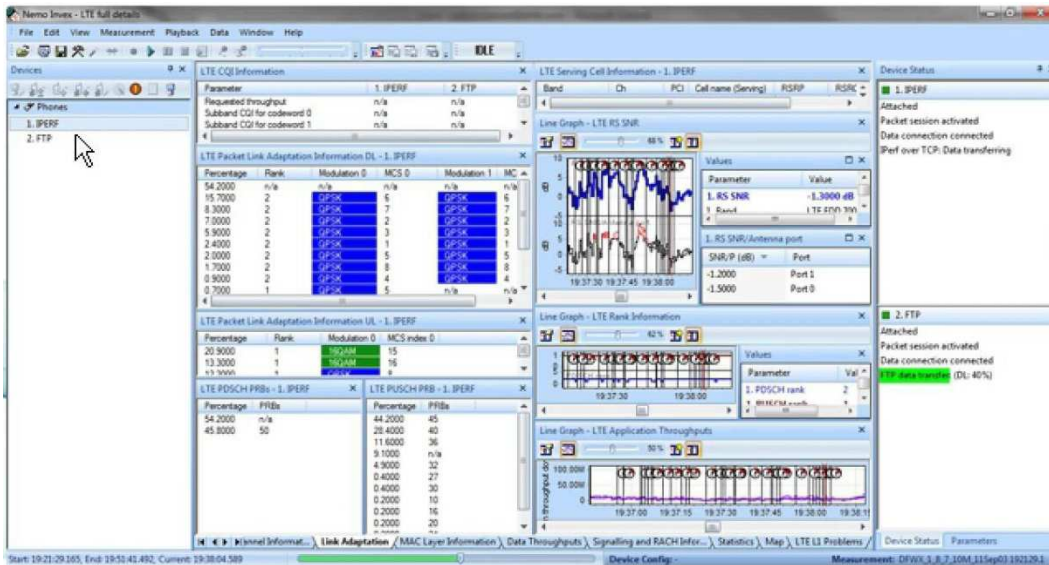


KUVA 9. Järjestelmän muodostuminen

Seuraavaksi esiteltynä järjestelmän sisältämiä työkaluja käytetään langattomien verkkojen mittauksissa tarkkailuun, suorituskyvyn mittaukseen, analysointiin ja raportointiin.

5.1 Nemo Outdoor ja Nemo Analyze

Nemo Outdoor on tietokoneella mobiiliverkkojen mittauksiin käytettävä ohjelmisto, joka mahdollistaa eri verkkoparametrien, signaloinnin ja dataliikenteen tarkkailun ja nauhoituksen päätelaitteen ja tukiaseman välillä. Se sopii käytettäväksi kaikille langattomille teknologioille mahdollistaen myös eri protokollien käytön ja eri parametrien reaaliaikaisen seurannan erilaisten diagrammien avulla (kuva 10). Sen avulla saadaan tallennettua mitatut langattoman verkonajotestitulokset sopivaan muotoon, mahdollistaen nopean ja helpon vianetsinnän ja analysoinnin. Ohjelman käyttö perustuu siinä ajettaviin Nemo-skripteihin, joiden avulla voidaan ohjailta ja tarkkailla matkapuhelimen käyttäytymistä erilaisissa tapahtumissa, kuten FTP-tiedonsiirrossa tai äänipuheluissa. Skripteissä voidaan määrittellä mitä ja milloin mitataan. Ohjelma sisältää valmiiksi jo suuren määrän valmiita testiskriptejä, mutta niitä voi halutessaan myös itse helposti luoda. (27.)



KUVA 10. Näkymä Nemo Outdoorista (27.)

Nemo Analyze:lla kyetään tarkastelemaan ajomittaustuloksia paljon monipuolisemmin, mitä Nemo Outdoorilla. Analyze:lla nimensä mukaisesti siis tehdään mittaustulosten käsittelyä ja niiden perusteella yhteenvetoja. Mittaustulosten vertailu on helppoa, koska samalle näytölle siinä pystytään asettamaan usea mittaus kerrallaan. Se on täten myöskin kätevä työkalu vianetsintään ja karttakuvan tarkkailuun.

5.2 Suorituskykyilmaisimet

Jotta testisyteemiä voidaan kehittää ja testata, tulee tietää sen sisältämistä verkon suorituskykymittareista ja niiden ominaisuuksista. On olemassa erilaisia tarkastelunäkökuja arvioitaessa matkapuhelinverkkoja. Työssä kiinnitetään pääosin huomiota mahdollisimman stabiileina pysyviin KPI-arvoihin, kun testataan testijärjestelmän toimintaa. Tarkoituksena on todeta suoritettujen automatisoitujen toimintojen antavan mahdollisimman toistettavia tuloksia kerta toisensa jälkeen.

5.2.1 RSRP (Reference Signal Received Power)

Yhtenä tärkeänä KPI:na (Key Performance Indicator) LTE-verkon mittauksissa on RSRP. Sitä hyödynnetään muun muassa solujen valikoinneissa, uudelleenvalikoineissa ja tukiasemavaihtoissa. Se on käsitteellisesti verrannollinen UMTS-tekniikan RSCP:n kanssa. RSRP:llä tarkoitetaan vastaanotettua signaalin voimakkuutta mitattaessa verkkoa päätelaitteella. Voimakkuus sisältää viitetietoja, joita päätelaite ja matkapuhelinverkko tarvitsevat voimakkuuden määrittämiseen. Voimakkuus ilmoitetaan desibeleinä suhteessa milliwatteihin (dBm) ja sen raportointialueena käytetään -40 – -140 dBm, mutta täytyy huomioida ettei sillä nähdä tarkasti signaalin laatua. Käyttökelpoiset arvot tyypillisesti ovat -75 – -120 dBm:n alueelta. (23.) RSRP määritellään lineaarisesti keskiarvoksi, kun verrataan tehon suhdetta resurssilohkoista (Resource Block). Käytettävä taajuuskaista kertoo resurssilohkojen lukumäärän, jonka signaali sisältää. (22, s. 28.) Voimakkuuteen voi vaikuttaa muun muassa matkapuhelimen ja antennin etäisyys toisistaan, käytettyjen antennikaapeleiden aiheuttamat häviöt ja antennivahvistukset.

5.2.2 RSRQ (Reference Signal Received Quality)

Toinen tärkeä parametri, jota päätelaite mittaa referenssisignaalista (RS) on RSRQ, jota käytetään, kun halutaan tietää enemmän signaalin laadusta. RSRP ei täten anna niin kattavaa laatumietoa. Mitä parempi RSRQ-arvo on, sitä parempi solun signaali on. Saadun laatumietoon pohjalta voidaan arvioida häiriötasoa. Parametri on käsitteellisesti verrannollinen UMTS-tekniikan E_c/I_0 :lle, mutta erona niissä on, että RSRQ:ssa otetaan huomioon resurssilohkot taajuuskaistalla. Sen mittaaminen on mahdollista ainoastaan, kun päätelaitteessa on datayhteys päällä ja näin ollen kyseinen KPI riippuu lähetetystä tiedosta ja verkon kuormituksesta. (22, s. 40.) RSRQ:ta kuvataan kaavalla 1.

$$N * \frac{RSSP}{RSSI}$$

KAAVA 1

N = resurssilohkojen määrä

$RSSI$ = kaikkien vastaanotettujen signaalien voimakkuutta

5.2.3 SNR (Signal to Noise ratio)

Signaali-kohinasuhteella, jota myös usein kutsutaan hyötysuhteeksi, tarkoitetaan kohina- sekä hyötysignaalin tehon suhdetta. Mikäli kohinaa havaitaan paljon taikka signaali on muuten heikko, aiheuttaa se pienen kohinasuhteen. Pieni kohinasuhde taas tuo vastaanottimelle omat ongelmansa, kuten että sen on vaikea erotella signaalin sisältämiä tietoja sen kohinasta. Kohinasignaaliin lasketaan mukaan kaikki ei-toivotut signaalit, häiriöt mukaan lukien. Sitä parempi yhteyden laatu saadaan, mitä parempaan signaali-kohinasuhteeseen päästään ja sillä on myös vaikutusta tiedonsiirtonopeuteen. (25.) SNR ilmaistaan desibeleinä ja sen hyvyysrajana tyypillisesti pidetään nollaa dB:ä suurempia arvoja (kuva 11). SNR määritellään seuraavalla kaavalla 2. (25.)

$$(S/N)_{dB} = 10 \log_{10} \frac{\text{Signaaliteho}}{\text{Kohinateho}}$$

KAAVA 2

Useasti SNR:stä puhuttaessa tulee vastaan nimi SINR (Signal to Interference & noise ratio). Sillä tarkoitetaan signaali-häiriösuhdetta. SNR:n verrattuna se on laajempi, sillä siihen lasketaan mukaan nimensä mukaisesti myös häiriöt.

		RSRP (dBm)	RSRQ (dB)	SINR (dB)
RF Conditions	Excellent	≥ -80	≥ -10	≥ 20
	Good	-80 to -90	-10 to -15	13 to 20
	Mid Cell	-90 to -100	-15 to -20	0 to 13
	Cell Edge	≤ -100	≤ -20	≤ 0

KUVA 11. KPI-arvoalueita RF-oloissa (23.)

5.2.4 CQI (Channel Quality Indicator)

CQI:lla ilmaistaan käytettävän kanavan laatua, joka on erityisen tärkeää suorituskyvyn kannalta. Se ottaa huomioon kohinat sekä vastaanottavan laitteen ominaisuudet.

CQI ilmoittaa suurimmat käytetyt modulaatiomallit sekä lohkojen virhesuhteet (BLER) kanavassa. Mitä suurempaa modulointitapaa siis käytetään, sitä suurempi on CQI-arvo. Se saadaan lasketuksi vertaamalla signaalin suhdetta kohinaan ja häiriöihin. Kanavan laatu ilmoitetaan yksinkertaisessa muodossa 0-15 välillä. Jos arvo on 0 tarkoitetaan sillä, ettei matkapuhelin ole vastaanottanut LTE signaaleita ja että kanava on toimintakyvytön. Puhelin voi käyttää kahdenlaista metodia lähetettäessä CQI-arvoja tukiasemalle ylälinkin kautta. Lähetys voidaan tehdä jaksottain, käyttäen PUCCH- tai PUSCH-kanavia tai jaksottomasti PUSCH-kanavalla. Jos tukiasema haluaa laatutietoja tietyltä ajalta, tapahtuu se jaksottomasti. Alalinkille on vain yksi metodi, PDCCH (Physical Downlink Control Channel). Laatuindeksi ei ole kuitenkaan suoriltaan riippuvainen mitatun signaali-kohinasuhteen kanssa. (24.)

5.2.5 Data Throughput

Kyseisellä termillä tarkoitetaan suuretta, joka ilmoittaa käytettävän yhteyden datansiirtonopeuden. Kyseinen siirtonopeus saadaan lasketuksi jakamalla verkonläpi kulkevan tiedoston koko, sen lähetykseen kuluneella ajalla.

Mittayksikkönä käytetään tyypillisesti megabittiä sekunnissa (Mbit/s), mutta kilobittiä ja bittiä sekunnissa ovat myös varsin yleisiä. (26.) Maksiminopeus saadaan täten lasketuksi TCP-protokollan:n avulla kaavalla 3.

$$\text{Throughput} \leq \frac{RWIN}{RTT}$$

KAAVA 3

RWIN = TCP:n ikkunakokoa

RTT (Round-trip delay time) = aika, joka kuluu paketin lähetyksestä sen kuittaukseen asti. Mikäli kuittaus on onnistunut, kutsutaan tätä nimellä ACK (Acknowledgment) ja mikäli se oli epäonnistunut niin NACK:ksi (Negative acknowledgment). (26.)

5.3 Laitteet ja ohjelmistot

Työssä käytettiin useita erilaisia laitteita ja ohjelmia.

Laitteet ja tarvikkeet:

- Anite Prosim F8-radiokanavaemulaattori
- RF-suojattu kotelo (Shield Box)
- Erilaisia matkapuhelimia:
 - Samsung Galaxy S4 LTE-A
 - Samsung Galaxy Alpha
- RF-antennikaapeleita

Ohjelmistot:

- Nemo Outdoor
- Nemo Analyze
- Microsoft Excel
- Minitab 17

6 TESTAUSSUUNNITELMA

Työ aloitettiin testaussuunnitelman tekemisellä. Testauksen tavoitteena oli osoittaa, että testijärjestelmä täyttäisi sille asetetut vaatimukset ja toiminnot sekä mahdolliset puutteet ja virheet löytyisivät, joiden johdosta tietokoneelta ajettava järjestelmä ei toimi, toimii väärin taikka ei vastaa sille asetettuja määrittelyjä.

Tein suunnitelman siitä, miten testauksessa edettäisiin työryhmässä tehtyjen vaatimusmäärittelyjen pohjalta, joka sisälsi tarkat tiedot siitä, mitä sovelluksen tulee osata tehdä ja miten. Suunnitelma tehtiin ajatellen testauksen kokonaisuutta, että miten siinä kannattaisi edetä, koska testaus on olennainen osa järjestelmän kehityksessä.

Testauksen työvaiheisiin kuuluivat testauksen suunnittelu, testiympäristön luominen, toiminnallisten testien ja mittausten suorittaminen, testitulosten tarkastelu sekä raportointi. Automatisoidun järjestelmän testaus suoritettiin osana yleisesti käytettyä ja aikeisemmin esiteltyä prosessimallia, V-mallia, jota käytetään paljon erilaisissa projekteissa. Testauskokonaisuuden vaiheet ja testaukset jakautuvat taulukko 1 mukaisesti:

TAULUKKO 1. Testattavat asiat

<p>Yksikkötestit</p> <p>Verifioidaan pääsovelluksen yleinen ilme ja rakenne sekä testataan käyttöliittymäkomponentit, että ne tekevät vain niille kuuluvat toiminnot.</p>	<ul style="list-style-type: none">- Testataan projektin ja kampanjan luomisesta:<ul style="list-style-type: none">• Sovelluksen tietoisesti antamat virheilmoitukset (tallentaminen)• uudelleen nimeämiset• sarakkeiden leveydet ja skaalautuminen• kytketyn puhelimen löytyminen sekä diagnostiikka-moodissa, että verkonjaolla.• tietojen pysyminen synkronissa eri
--	---

	<p>projekteja ja kampanjoita vaihdeltaessa</p> <ul style="list-style-type: none"> • testikuvauslaatikon oikea toiminta • luotujen tiedostojen ilmestyminen/poistuminen kohdekansiosta
<p>Integrointitestit</p> <p>Kasataan testilaitteisto laboratorioon ja luodaan automaatiolla toimiva skripti ja emulaatio Propsimiin.</p>	<ul style="list-style-type: none"> - Varmistetaan, että automatiset toiminnot toimivat mittauksia suoritettaessa: <ul style="list-style-type: none"> • järjestelmäosat vaihtuvat käyttöliittymässä oikea-aikaisesti • mittausarvot vaihtuvat Outdoori:n mukaan luotettavasti • mitatut tulokset näkyvät oikein ja, että niitä kyetään käsittelemään Analyze:ssä • suoritetaan mittauksia kahdella eri matkapuhelimella ja todetaan, että järjestelmä kykenee toistettaviin ja stabiileihin tuloksiin. <ul style="list-style-type: none"> ○ verrataan mittauksiin kuluneita aikoja sekä suorituskykyilmaisimista signaalin voimakkuutta ja signaali-kohinasuhdetta. • verifioidaan ne valmiit testitapaukset, jotka ajan puitteissa keretään toistamaan EXFO:lta vuokratussa Atlas-laboratoriossa, jossa mahdollistuu useamman solun käyttö LTE-teknologialla.
<p>Järjestelmätestit</p> <p>Kuormitustesti</p>	<ul style="list-style-type: none"> - Tarkastellaan testijärjestelmän suorituskykyä kuormittamalla sitä eli asetetaan järjestelmä sen ääriarajoille. Tarkastellaan suuria määriä mitattuja RSRP- ja CQI-arvoja.
<p>Toipumistesti</p>	<ul style="list-style-type: none"> - Aiheutetaan tahallisia virheitä järjestelmän käytössä irroittamalla eri yksiköitä kesken mittauksen ja varmistetaan hallittu käyttäytyminen.

7 TESTAUKSEN SUORITUS

Sovelluksen testauksessa ja läpikäynnissä etenemismallina käytettiin V-mallia, jossa edettiin yksikkötestauksesta aina järjestelmätestaukseen asti. Sille tehtiin regressiotestausta pitkin kehitystä, jotta voitiin todeta eri vaiheista löytyneiden vikojen korjautuneen. Perustoimintojen varmistamisen ja määrittäisiin vertaamisen yhteydessä kiinnitettiin myös huomiota sen käytettävyyteen, mittausten stabiilisuuteen ja toistettavuuteen sekä järjestelmän suorituskykyyn.

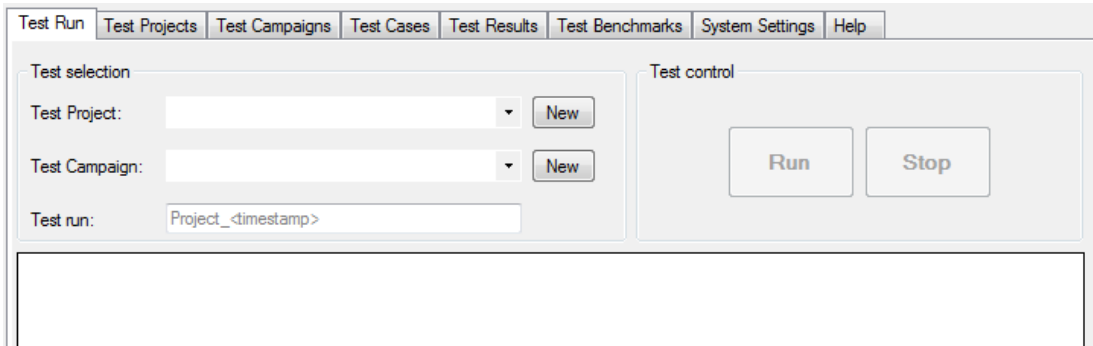
7.1 Yksikkötestit

Koodiperäisen yksikkötestauksen teki pääosin ohjelmoijat itse, mutta omalta osalta verifioin sekä testaajan, että loppukäyttäjän näkökulmasta sovelluksen ulkoista perustoiminnallisuutta ja käyttöliittymäkomponentteja, että ne toimivat oikein tekivät vain niille määrättyt asiat. Käyttöliittymä käytiin läpi mustalaatikkomallin avulla. Varmistettiin myös, että tietoisten virhetilanteiden antamat ilmoitukset toimivat. Testivaiheessa ei ajettu vielä mitään tai käytetty muita järjestelmän osia.

Testaus aloitettiin käyttöliittymän yleisilmeen läpikäynnillä ja komponenttien testauksella. Testijärjestelmän C#-ohjelmointikielellä luotu käyttöliittymä muodostuu useista erilaisista komponenteista eli niin sanotuista rakennuspalasista. Jokaisella komponentilla on tietty luokka käyttöliittymäkirjastossa, jonka mukaan ne toimivat. Niiden tarkoituksena on yleenäkin suorittaa tai näyttää tietty toiminto ja antaa joissakin tapauksissa käyttäjän muokata niitä. Tällaisia ovat välilehdet (tab), painonapit (button), staattiset sekä muutettavat tekstikentät (textbox), listauskentät (listbox), ilmoitusikkunat sekä alavetovalikot (dropdown list).

Käytiin läpi kaikki kyseiset elementit sekä niiden väliset yhteydet ja todettiin, että ne toimivat virheettömästi ja siten mahdollistavat sovelluksen sujuvan käytön, kun mitään ei vielä ajettu. Verrattiin komponenttien toimintaa niiden

alkumäärittelyihin ja spesifikaatioihin. Tuli varmistaa, ettei ohjelmassa tapahtunut niin sanottuja ristiriitaisia tapahtumia kenttien välillä, jotka yleensä aiheuttavat kehitysvaiheessa virheitä. Testattiin, että annetusta syötteestä tulee odotettu tulos ja ettei mihinkään staattisiin kenttiin voitu syöttää tai poistaa mitään arvoja.

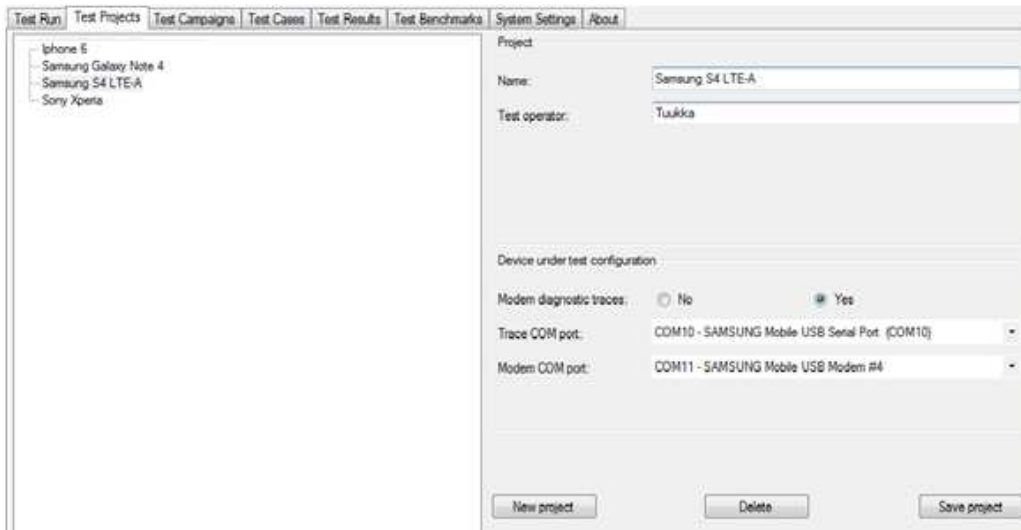


KUVA 12. Test Run-välilehti

Sovelluksen avauduttua, alunäkymäksi tuli Test Run-välilehti, jossa itse testiajojen valitseminen ja hallinnointi tapahtui (kuva 12.) Käytettävyyden kannalta oli varsin järkevää, että Test Run-välilehti oli alunäkymänä, koska mikäli käyttäjä on jo aikaisemmin ajanut testejä valmiiksi määritetyllä puhelimella, nopeuttaa se entisestään sovelluksen käyttöä, etenkin mikäli halutaan ajaa sama testikampanja uudestaan. Kyseinen välilehti muodostuu määritellyistä projekteista (puhelin, yhteystapa ja testaaja) ja kampanjoista (testitapaukset, emulaatiot ja raporttimallit), joita mitattaessa halutaan käyttää. Test Run-, Test Results- ja Test Benchmarks-välilehtiä tarkasteltiin tarkemmin toiminnallisen testausvaiheen yhteydessä, koska niiden toiminnan verifioimisessa vaadittiin testiajosta syntyviä toimintoja ja tuloksia. Test Run-välilehdessä sekä projektin, että kampanjan tekoon oli valikoiden vieressä painonapit, joista täytyi pystyä onnistuneesti siirtymään niiden luontiin tarkoitetuille välilehdille suoraan, joten ne käytiin testataten ensin läpi.

1) Test Project-välilehden testaus

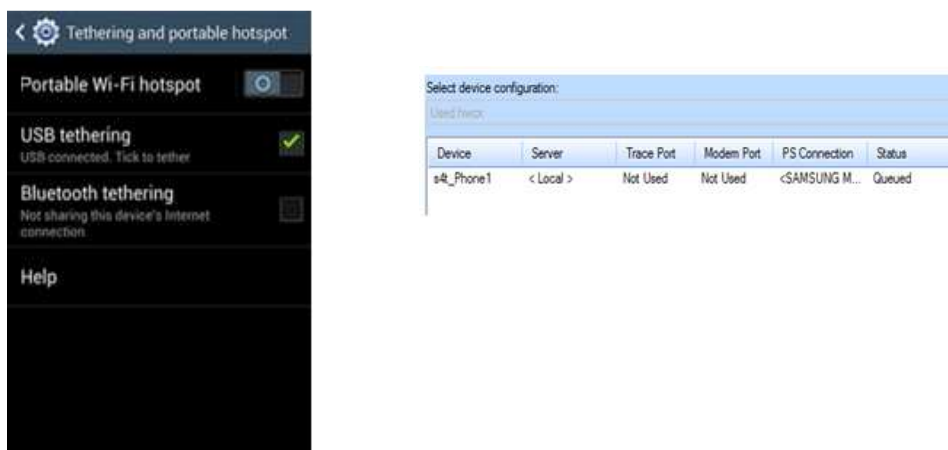
Painettiin projektin kohdalta New-nappia, jolloin käyttäjä siirtyi Test Project-välilehteen (kuva 13.) Kyseisessä välilehdessä on tärkeää, että käyttäjä kykenee helposti luomaan projekteja, esimerkiksi jos hän haluaa vertailla testauksia eri puhelimilla tai eri yhteystavoilla.



KUVA 13. Test Project-välilehti

Siksi oli tärkeää varmistaa projektin- sekä testaajan muutettavista nimeämiskentistä, että pystyttiin syöttämään erilaisia merkkikombinaatioita ja erikoismerkkejä (kuva 15.) Mikäli jokin merkki ei ollut sallittu niin tuli ohjelman antaa siitä tietoinen info-ikkuna, virheilmoituksen sijasta (kuva 14) muutoin tulosteiden tuli tulla näkyviin. Myöskin tuli varmistaa, ettei ohjelma antanut virhettä tyhjästä tai kohtuuttoman pitkistä nimeämistä, vaikkakin se on erittäin epätodennäköistä, että käyttäjä käyttäisi esimerkiksi 50 merkin mittaisia nimiä, mutta toisaalta ohjelman täytyi olla mahdollisimman virheetön. Kaikkia virheitä tuli pitää kriittisinä, koska yksikin virhe voi vaikuttaa jossaki muussa ohjelman osassa aiheuttaen uusia virheitä.

project-nappia, joka päivittää asetukset tai käynnistettiin ohjelma uudestaan. Nyt valikoissa tuli näkyä kytketyt diagnostiikka trace- ja modeemiportit ja ne vielä tarkistettiin oikeiksi Windowsin laitehallinnasta. Katsottiin luodusta projektitiedostosta, että se sisälsi asetetut portit ja että ne eivät olleet muuttuneet vääriksi. Tätä testattiin vielä useilla eri valmistajien puhelimilla ja todettiin valikot toimiviksi. Tuli ottaa huomioon, että niiden käyttöön vaadittiin puhelinkohtaiset ajurit ja diagnostiikat, joita kaikki puhelimet eivät tukeneet. Tehtiin vielä muuten sama homma, mutta ei määritelty COM-portteja vaan Application Tester-osan avulla itse verkkolaitteena toimiva puhelin, jonka verkkoasetuksiin asetettiin USB tethering-verkonjako päälle, jolloin puhelin toimi modeemina tietokoneelle (kuva 16.) Tässä vaiheessa käytettiin simuloitua propsimia, jotta voitiin todeta Nemo Outdoorin ottavan käyttöön oikeat portit mittauksiin.



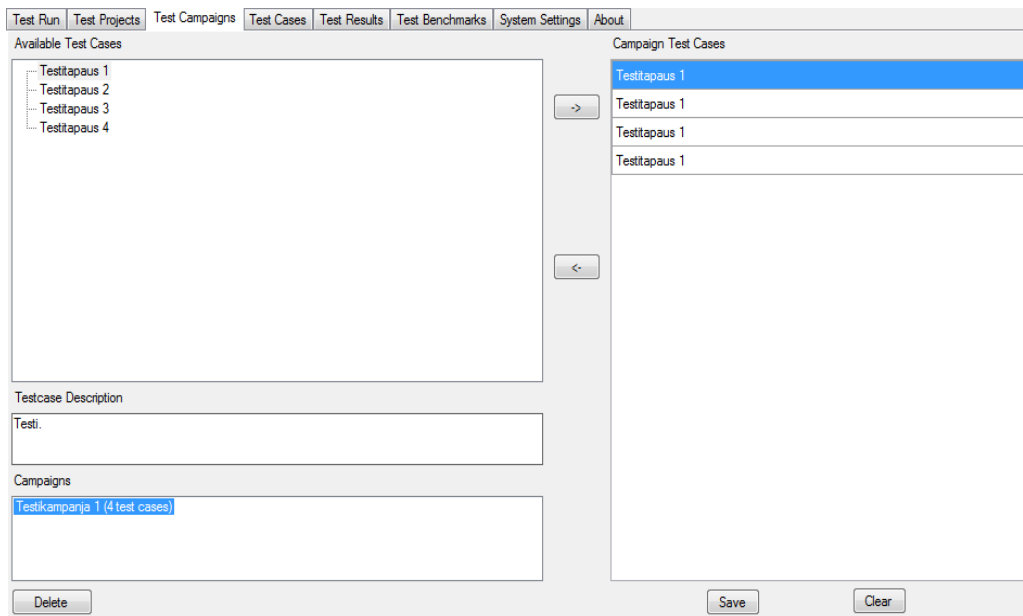
KUVA 16. Verkonjaon käyttöönoton verifiointi järjestelmässä

2) Testikampanja-välilehden testaus

Siirryttiin takaisin Test Run-välilehteen ja painettiin New-nappia kampanjan kohdalta, jolloin käyttäjä siirtyi kampanjanluonti-välilehteen (kuva 17.) Kampanjat ovat isompia kokonaisuuksia, jotka kootaan sovellukseen

määritellyistä testitapauksista. Testitapaukset taas sisältävät käytettävän verkkoteknologian, solumäärän, mitattavat KPI:t, ajettavan emulaation polun, raporttimallin sekä itse testitapaus-skriptin toimintavaiheet.

Vasemmanpuoleisessa ylemmässä listauskentässä tuli näkyä nuo sovellukseen määritetyt testitapaukset, joita käyttäjän tuli kyetä valitsemaan tuplaklikkauksen tai nuolinappien avulla Campaign creation-listaan. Valittiin haluttu määrä testitapauksia, tämän jälkeen painettiin Clear-painiketta ja todettiin, että kampanjalista pystyttiin tyhjentämään. Valittiin testit uudestaan ja painettiin Save, minkä jälkeen tuli sovelluksen kysyä nimeä kampanjalle, jolloin suoritettiin sama toimenpide, kuin Test Projects-osioissakin tekstikentille. Tallentuneen kampanjan näin ollen täytyi tallentua sille tarkoitettuun Campaigns-listaan, josta voitiin nähdä testit ja niiden lukumäärät. Luotiin useita erilaisia kampanjoita ja vaihdeltiin niitä vuorotellen ja varmistettiin, että tietty kampanja sisälsi aiemmin määritellyt testit ja että ne eivät muuttuneet missään vaiheessa vaan pysyivät luotettavasti oikeina. Testikampanjoiden poistaminen testattiin vastaavasti, kuin testiprojektienkin. Testcase Description-laatikon tiedot todettiin myös toimiviksi, lisäämällä ja poistamalla testitapaukseen kuvausteksti sekä skriptieditorilla, että suoraan XML-koodiin muokkaamalla. Kampanja-välilehti itsessään on käytettävyydeltään erittäin yksinkertainen ja helposti omaksuttava, josta on helposti nähtävillä valitut ja valittavana olevat testit ja niiden kuvaukset.



7.2 Integroitit

Ensimmäisessä testausvaiheessa luotiin testikytkennät, emulaatio ja testitapaus-skriptit, jotta voitiin suorittaa dynaaminen tarkastelu eli testataan sovelluksen käyttäytymistä suorituksen aikana käyttäen kaikkia järjestelmäosia. Myöskin verkonmittausarvoja tuli pitää silmällä, että sovellus kykenee pitämään arvot hyvissä rajoissa. Läpi käytäessä pidettiin sovelluksen loki-ikkunaa auki, josta tarkkailtiin eri toimintojen onnistumisia. Testattavassa versiossa keskitytään datan tiedonsiirto-mittausten testaukseen, kuten FTP- ja ping-testeihin, joissa tarkkaillaan lähetysten onnistumisten lisäksi myös signaalin ja kanavan laatua ja vakautta. Seuraavassa vaiheessa varmistetaan, että valmiiden testitapausten mittauksia kyetään toistamaan, jossa skriptien lisäksi tulee myös valmiit emulaatiot ja raporttimallit. Työssä mittauksiin käytetään ainoastaan LTE-verkkoa, jota laboratorioissa pystytään ainoastaan mittaamaan.

7.2.1 Kytkentöjen, emulaation ja testiskriptin teko

Testaus aloitettiin testikytkentöjen tekemisellä laboratorioon (kuva 18.) Yhdistettiin live-testiverkon tukiasemasta labraan tulevat RF-antennikaapelit Prosim F8-radiokanavaemulaattorin sisääntuloihin. Aniten tiloissa käytettävässä testiverkossa käytettiin LTE:n taajuuskaistaa 1800 ja se mahdollisti vain yhden solun käytön. Kanavana toimi 1300. Yksisoluisessa verkossa täten käytetään vain yhtä taajuusaluetta, jossa useat käyttäjät liikkuvat vapaasti ollen yhteydessä tukiasemaan, joka on taas yhteydessä runkoverkkoon (EPC), josta taas ollaan yhteydessä internetiin. Prosimilla luotiin tulevalle signaalille halutunlainen kanavamalli. Tukiaseman signaalin laadussa saattoi tapahtua välillä pientä heittoa muun muassa sään ja antennin suuntauksen takia, mutta emulaatiota kyettiin tarvittaessa säätämään Prosimista sopivalle tasolle. Testauksessa käytettävä matkapuhelin asetettiin

eristävään laatikkoon (shield box), josta se yhdistettiin myös Propsimiin. Puhelin täytyi yhdistää USB-kaapelilla tietokoneeseen, jotta sen diagnostiikka saatiin käyttöön ja asetettiin puhelin käyttämään LTE-verkkoa. Propsim kytkettiin vielä tietokoneeseen LAN- tai GPIB-kaapelilla, jotta testijärjestelmä kykeni ohjailemaan sitä testiajojen aikana ATE-komentojen välityksellä.

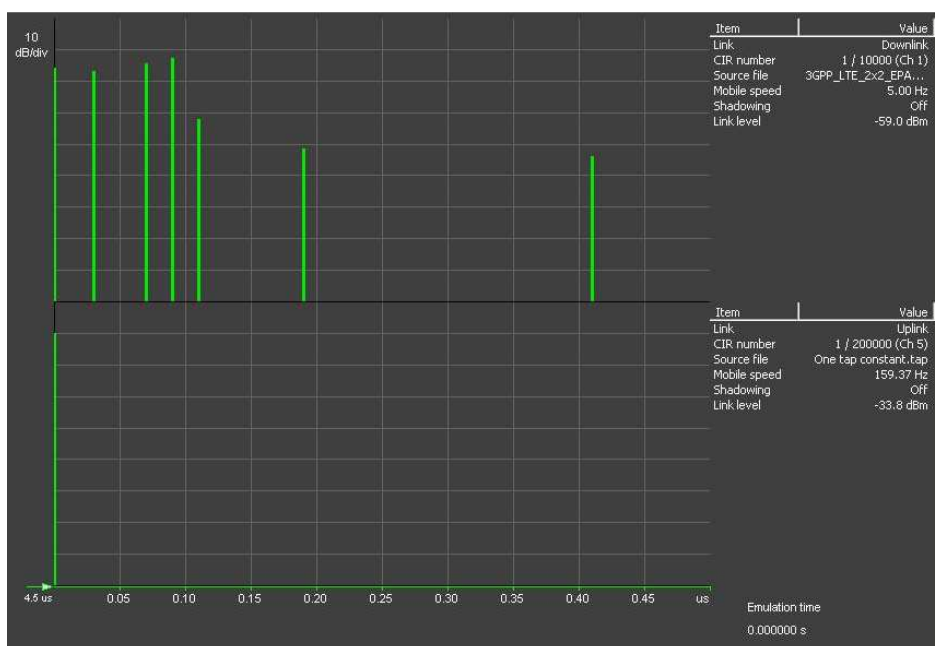
Tietokoneessa pääsovelluksen lisäksi tuli olla asennettuna Nemo Outdoor ja Nemo Analyze, joiden käyttö vaati tietokoneen USB-porttiin liitettävän muistitikun, joka sisälsi käyttölisenssit. Myös pääsovelluksen käyttöön piti olla toimiva lisenssi sovelluksessa, joka testattiin aiemmin sekä käytettäviin matkapuhelimiin tuli olla asennettuna vaadittavat ajurit. Puhelimeissa käytettiin TeliaSoneran micro- ja nano-SIM-kortteja, joka mahdollistivat datansiirtotestien suorituksen LTE-verkon välityksellä.



KUVA 18. Aniten laboratoriossa käytettävän yksisoluisen verkon laitteisto

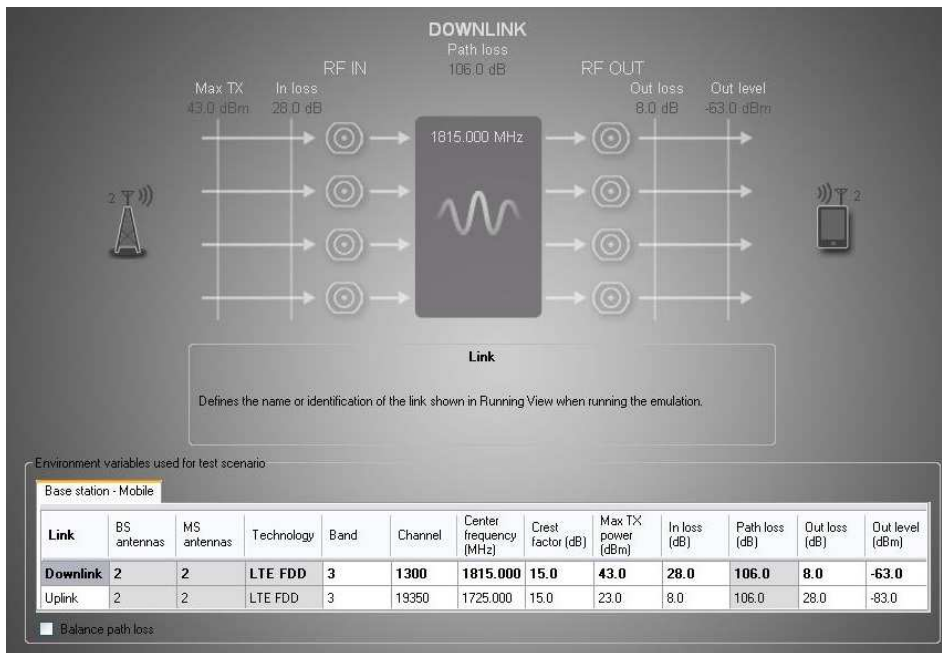
Testien ajamiseksi luotiin ensin sopiva emulaatio, jota Propsimissa ajettaisiin toimintojen varmistamiseksi. Emulaatiolle valittiin topologiaksi neljä loogista

kanavaa sisältävä 2X2 MIMO, jossa lähetykseen ja vastaanottoon käytetään kahta antennia. Valittiin käytettäväksi tekniikaksi FDD eli taajuusjakoinen tekniikka, jossa lähetteet liikkuvat pareittain ala- ja ylälinkissä erillisillä taajuuskaistoilla. Tukiasemasta matkapuhelimeen päin asetettiin keskitaajuudeksi 1815 MHz ja matkapuhelimesta tukiasemaan päin 1720 MHz. Tukiasema siis vastaa matkapuhelimelle menevästä ja sieltä tulevasta tietoliikenteestä sekä myöskin kertoo matkapuhelimelle tärkeitä komentoja esimerkiksi kanavanvaihtoista. Kanavamalliksi alalinkille valittiin yleisesti LTE-kanavamallina käytetty ja pienen viivehajonnan omaava EPA 5 Hz low correlation-malli, joka sisälsi 7 tappia eli polkua. Mallissa solujen koot ovat varsin pieniä, minkä johdosta sitä käytetään kaupunkialueilla. Ylälinkille taas valittiin 1 tappinen vakiona pysyvä malli, jossa ei esiinny lainkaan nopeaa häipymistä. Sitä käytettäessä voidaan luoda ideaalinen kanava. (Kuva 19.)



KUVA 19. Emulaation CIR-näkymän tapit

Jotta signaalitaso saatiin hyväksi, asetettiin ensin sopiva Crest factor-arvo, joka yleisesti on 15 dBm, mikä kertoo signaalin huippuarvon ja tehollisarvon suhteen. Annettiin myös sopivat signaalin sisä- ja ulostulotasot sekä kaistanleveydeksi 40 MHz. Lopuksi emulaatio generoitiin käyttökelpoiseksi (kuva 20.)



KUVA 20. Emulaation luomisnäkyminen Prosimissa

Tämän jälkeen luotiin testitapaus-skripti testattavaan järjestelmään automaattisten testiajojen toimintojen tarkastelua varten. Testitapausten teossa käytettiin siihen kehitettyä skriptieditoria sekä välillä muutettiin myös arvoja suoraan koodiin. Ensimmäiseksi testiksi luotiin datansiirtotesti, käyttäen FTP-protokollaa, jossa vastaanotettiin zip-tiedosto määritellystä osoitteesta LTE-verkossa. Toiseksi Ping-testi, jossa otettiin yhteys host-osoitteeseen 150 kertaa. Testeihin valittiin tarkkailtavaksi KPI-arvoiksi signaalin voimakkuus, signaalikohinasuhde sekä tiedonsiirtonopeus. Skriptit luotiin ja tallennettiin järjestelmän omaan .vdttest-tiedostomuotoon. Testitapauksiin valittiin aiemmin luotu emulaatio, solujen määräksi yksi, mittausajaksi 50 sekuntia sekä raporttimalleiksi VDT_FTP ja VDT_ICMP_Ping, joilla luodaan mittaustulosten tarkastelua varten lopuksi raportti Nemo Analyzessä. Raporttiin valittiin ilmoitettavaksi keskiarvoinen signaalin voimakkuus, SNR ja keskiarvoinen alalinkin latausnopeus. Molempien automatisoitujen testien vaiheet asetettiin toimimaan seuraavanlaisesti:

- Etsitään käytettävä emulaatio Prosimista ja käynnistetään se

- Avataan Nemo Outdoor ja etsitään käytettävän matkapuhelimen verkonsovitin- tai COM- portit
- Etsitään käytettävä Outdoor-skripti hakemistosta
- Käynnistetään mittaus ja sen jälkeen skripti Outdoorissa
- Käynnistetään mittaukseen määritelty mittausaika testijärjestelmässä
- Suoritetaan mittaus määriteltyjen kertojen ja ajan mukaan
- Suljetaan Outdoor-skripti
- Lopetetaan mittaus Outdoorissa
- Lopetetaan emulaation ajaminen Prosimissa
- Luodaan Analyze:ssä Excel-tulostiedosto

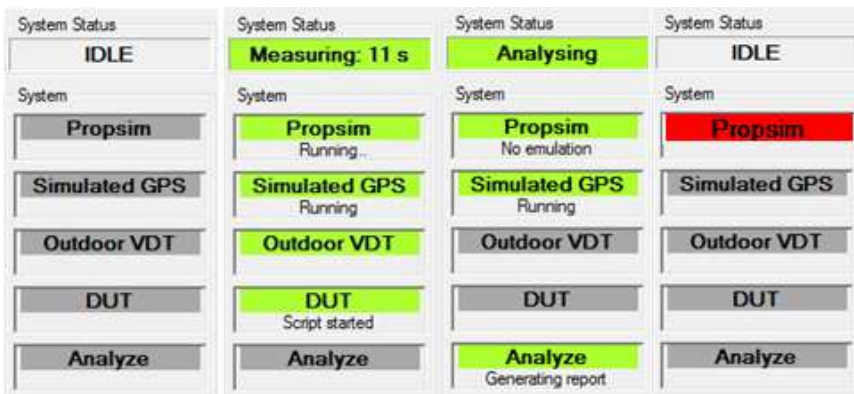
7.2.2 Automaattisten toimintojen testaus mittauksen aikana

Testattiin, että järjestelmä suorittaa toiminnot äskeisessä järjestyksessä ongelmitta. Kun kytkennät oli tehty, avattiin pääsovellus ja siirryttiin Test Project-välilehteen, jossa luotiin projekti ja kampanja. Kytkeyn Samsung Galaxy S4-puhelimen COM-portit löytyivät, jolloin ne valittiin valikoihin. Test Campaign-välilehdestä valittiin erikseen aikaisemmin tehdyt testitapaukset ja luotiin niistä omat kampanjat. Kun projektit ja kampanjat oli luotu, valittiin ne Test Run-välilehdessä ja annettiin ajettavalle testille nimi. Testattiin, että nimen pystyi myös jättää laittamatta jolloin testin tuli tallentua aikaleiman (timestamp) perusteella.

Testimittauksen hallinnointi tapahtui Run- ja Stop-napeilla, joiden käyttö tuli olla estetty kun vaadittavia asetuksia ei oltu annettu ja vastaavasti sallittu kun asetukset oli annettu. Ennen mittauksen käynnistystä täytyi muistaa muuttaa Nemo Outdoorissa puhelimen asetuksiin käytettäväksi vain LTE-verkkoa, ettei puhelin vaihtanut verkkoteknologiaa kesken mittauksen. Kun kaikki valinnat oli tehty painettiin Run-nappi, jolloin kampanjan ajo käynnistyi.

Järjestelmäosien tilojen vaihtuminen

Tärkeää osaa ajossa näyttelee eri staukset eli laitteiden suoritustilat, joiden tulee kyetä automaattisesti vaihtumaan skriptin mukaan (kuva 21.) Ennen kun mittaus käynnistettiin, tuli System status-laatikon kenttien olla harmaassa idle-tilassa, millä ilmoitettiin, ettei mitään suoritettu. Ajon käynnistyttyä muuttui System status Running-tilaan ja Prosim-kenttä vihreäksi tarkoittaen, että se oli kytketty ja, että ajettava emulaatio oli löytynyt. Tämä todettiin vielä erikseen Prosimista, että emulaatio tosiaan oli käynnistynyt ja, että ATE monitor oli auennut, josta tarkkailtiin, että käytettävien signaalien tehotasot pysyivät hyvinä eikä esiintynyt liian suuria tehotasoja suhteessa asetettuun arvoon. Vastaavasti sen tuli ilmoittaa, ettei emulaatiota löydetty mikäli se uupui, joten se testattiin muuttamalla vääräksi emulaation polkua skriptiin. Tämän jälkeen käynnistyi Outdoor, josta sovellus haki ja varmisti käytettävän matkapuhelimen sekä siinä käytettävät portit ja diagnostiikan, yhdisti sen LTE-verkkoon ja käynnisti ajettavan Outdoor-skriptin. Outdoor VDT-kentän tuli myös tällöin vaihtua käyttöliittymässä aktiiviseksi. Kun itse Outdoor-skriptin ajo alkoi, muuttui DUT:n tila vihreäksi kertoen, että sillä suoritettiin mittausta. Tällöin katsottiin Outdoorin omalta näytöltä, että pakettidatayhteys saatiin käynnistettyä ja, että Outdoor-skriptin toimintajärjestys toimi systemaattisesti. Varmistettiin, että VDT-skriptiin määritetty 50 sekunnin mittausaika oli oikea ja että se lähti kulumaan mittauksen alkaessa. Koko prosessin ajan tarkkailtiin sovelluksen loki-ikkunaa, että se näytti oikeat asiat oikeaan aikaan (kuva 22.) Stop-napin toiminta testattiin lopettamalla testaus eri vaiheissa mittauksen suoritusta ja täten toteamalla hallittu lopetus.



KUVA 21. System status-paneelin tilojen vaihtelut

Mikäli jokin edellä mainituista kentistä muuttui harmaaksi, ei kyseistä testijärjestelmäosaa suoritettu tai sen suoritus oli lopetettu. Järjestelmäosan virheestä täytyi ilmestyä punainen kenttä sekä statukseen että lokiin. Onnistuneen mittauksen päätteeksi tuli kaikkien kenttien muuttua passiivisiksi lukuunottamatta Analyze VDT:tä joka muuttui aktiiviseksi sen ajaksi, kun se muodosti mitatuista tuloksista Excel-raportin. Tarkastettiin, että raportti sisälsi kaikki mittaukseen valitut KPI-arvot, niiden tuloskäyrät sekä karttakuvaan ajetusta reitistä. Kun mittaus päättyi käyttöliittymään tuli tulla siitä ilmoitus.

```

14:58:24.7035 INFO Closing connections to devices succeeded!
14:58:27.1747 INFO Analysis finished!
14:58:39.3530 INFO Checking devices and paths:
14:58:39.8970 INFO Prosim: Anite Telecoms,Prosim FS8,880012,3.5 v0.0
14:58:39.8970 INFO Outdoor UDI v7.5.0.5
14:58:39.8970 INFO Analyze v7.30.0.0
14:58:39.9200 INFO Sequencer: Running
14:58:39.9620 INFO Prosim: Connection established!
14:58:39.9620 INFO Outdoor UDI is running!
14:58:41.0221 INFO Outdoor UDI: Connection established!
14:58:41.0541 INFO Loading configuration file to Outdoor UDI.
14:58:45.5736 INFO TESTCASE "UDI-RM_D_A_TM_001" Skipped
14:58:45.5906 INFO Sequencer: Stopped
14:58:45.5906 INFO Closing connections to devices...
14:58:45.5906 INFO AnalyzeJobTask: Finishing job worker...
14:58:45.8406 INFO Closing connections to devices succeeded!
14:58:48.0568 INFO Analysis finished!
14:59:00.1378 INFO Checking devices and paths:
14:59:00.6589 INFO Prosim: Anite Telecoms,Prosim FS8,880012,3.5 v0.0
14:59:00.6589 INFO Outdoor UDI v7.5.0.5
14:59:00.6589 INFO Analyze v7.30.0.0
14:59:00.6869 INFO Sequencer: Running
14:59:00.7359 INFO Prosim: Connection established!
14:59:00.7359 INFO Outdoor UDI is running!
14:59:09.8010 INFO Outdoor UDI: Connection established!

```

KUVA 22. Toimintojen tarkkailu loki-ikkunasta

Mittausarvojen vaihtuminen Outdoorin mukaan

Käynnistettiin testit vuorotellen uudestaan ja varmistettiin, että testeihin valitut verknomittausarvot näkyivät ja vaihtuivat Outdoorin mukaan sovelluksessa

oikein ja vakaina. Varmistettiin myös että järjestelmä kykeni pitämään arvot odotetuissa rajoissa. (Kuva 11.) Testijärjestelmä perustuu juuri KPI:den mittaukseen, jolla saadaan tietoa verkon tilasta matkapuhelimen ja tukiaseman välillä. Testitapauksiin oli lisättyä pari KPI:ta lisää, jotta voitiin tarkastella, että järjestelmä kykeni ilmoittamaan muitakin tärkeitä LTE-verkon tilan tarkasteluarvoja mittauksen aikana. Tällöin valittuna oli RSRP, RSRQ, RSSI, SNR, tiedonsiirtonopeus ja CQI. Kun mittaukset olivat käynnissä verrattiin Outdoorin mittaamia arvoja ja pääsovellukseen annettuja arvoja omilta näytöiltään (kuva 23.) Mittausarvojen tuli olla täysin samat koska arvot käyttöliittymään tulee Outdoorista. Jos niissä oli pienikin ero, ei järjestelmän voitu katsoa olevan luotettava. LTE-verkon suorituskykyilmaisimien rajat pysyivät useaan kertaan tehdyissä mittauksissa stabiileina ja hyvissä rajoissa ilman suuria heittäilyitä, kuten kuvasta 23 voidaan nähdä. Kyseisen mittauskerran RSRP-arvot pyörivät -88 dBm:n ympärillä, mikä kuvaa hyvää signaalin voimakkuutta, kun verrataan sivulla 25 olevaan taulukkoon, jossa RSRP-arvot välillä -80 – -90 dBm ovat yleisesti hyviä. RSRQ-arvot liikkuvat -7 dB paikkeilla, mikä osoittaa todella hyvää signaalin laatua. Hyvänä signaalikohinasuhde arvoina pidettiin 0 dB:ä suurempia arvoja ja koska mittaukset liikkuvat 5dB:n ympärillä, niin senkin voidaan todeta olevan hyvä. Myöskin CQI eli kanavanlaatuarvot liikkuvat 10 ympärillä eli varsin suurina ja siten hyvinä, mikä selittyy sillä, että mittattaessa Outdoor käytti 64QAM-modulaatiota suurimman osan ajasta. Mittauksen päätyttyä tarkastettiin luotua tulostiedostoa, että se sisälsi kaikki mitatut arvot valituista KPI:sta listattuna. Tulostiedostosta tuli myös käydä ilmi käytetty emulaatio, Outdoor-skripti sekä mittausaika. Tarkasteltiin vielä tulostiedostojen kokoja, ettei esimerkiksi edellisten mittausten tuloksia kirjattu seuraavaan mittaustiedostoon, mikä voisi vaikuttaa tiedostokokoihin ja siten aiheuttaa ongelmia tietokoneen muistin käytössä. Lopuksi voitiin todeta mittausarvojen vaihtuvan luotettavasti Outdoorin ja Prosimin asetusten mukaisesti ja pysyvän vakaina.

Nemo Outdoor							Käyttöliittymä	
LTE Serving Cell Information - 1. Samsung S4 LTE-A, Phone1							Runtime KPI values	
Band	Ch	PCI	Cell name (Serving)	RSRP	RSRQ	RSSI	APP TP	20 Mbit/s
LTE FDD 1800	1300	11E	m/a	-89.9375	-7.3750	-62.5625	CGI	8
							RSRP	-89.9375 dBm
							RSRQ	-7.3750 dB
							RSSI	-62.5625 dBm
							SNR	5.4000 dB

KUVA 23. Vaihtuvien arvojen oikea-aikaisen vaihtumisen testaus.

Testituloksien oikea näkyminen

Seuraavaksi varmistettiin, että mittauksista tulleet tulokset ilmestyivät Test Results-välilehteen (kuva 24.) Kun haluttiin tarkastella mitattua kampanjaa, tuli alasvetovalikoihin määritellä käytetty projekti ja kampanja, jolloin sen sisältämät testit listattiin aikajärjestyksessä listbox:iin. Tulos KPI-arvotaulukosta testattiin, että mitatut arvot vaihtuivat kussakin testitapauksessa kun niitä vaihdeltiin listboxista käsin. Testeistä täytyi myös pystyä avaamaan niiden kohdekansio sekä tulosraportti niille tarkoitetuilta napeilla. Testattiin kunkin mittauksen onnistumisen oikea näkyminen:

- Ajettiin 15 testiä, josta ajettiin onnistuneesti 10, luotiin kaksi virhettä ja kaksi testin välistäjättämistä sekä loppuun keskeytys. Varmistettiin Test Results-kansiosta, että onnistumisen tila vastasi ajon päätteeksi annettua Completed-, Error- ja Cancelled-ilmoitusta. Välistäjättämissä ei tullut näkyä tuloslistauksessa. Muutettiin tilamääriä ja toistettiin vielä useamman kerran, jotta voitiin todeta valikoiden vaatimusten mukainen toiminta. Kyseinen välilehti oli yksi sovelluksen tärkeimmistä, jonka oli toimittava luotettavasti.
- Testattiin, että FTP-tiedonsiirron suoritus oli onnistunut katsomalla ensin Outdoorista, että mittaukset suoritettiin samanaikaisesti määriteltyjen kertojen verran ja että pakettien vastaanotto onnistui. Mittauksien päätyttyä katsottiin latauskansiosta, että määritellystä osoitteesta vastaanotettu tiedosto oli ilmestynyt kohdekansioon. Ping-

testissä todettiin, että ICMP pingaus kohde host-osoitteeseen onnistui Outdoor-skriptiin määriteltyjen toistokertojen mukaan. Pakettikooksi asetettiin 32 tavua. Verrattiin usean mittauksen RTT-arvoja eli aikoja, joka kului paketin siirtoon ja niiden tuli olla lähes samat. Molemmista testeistä syntyneitä Analyze-raporttien tuloksia verrattiin keskenään ja todettiin, että FTP-tiedonsiirto- sekä pingauskertojen onnistumiset vastasivat mittaustapahtumia.

The screenshot shows a software interface for test results. At the top, there are tabs for 'Test Run', 'Test Projects', 'Test Campaigns', 'Test Cases', 'Test Results', 'Test Benchmarks', 'System Settings', and 'About'. Below the tabs, there is a 'Result Selection' section with dropdown menus for 'Test Project: Samsung Galaxy S4', 'Test Campaign: 15xFTP', and 'Test Run: 15_2015-04-29 09:43:45'. To the right, there is a 'Campaign status' summary box showing: Duration 00:16:36, Unselected 0, Cancelled 1, Error 2, Measured 0, and Completed 10/13. Below this is a 'Result KPI Values' box showing: Avg Application thro: 7,129, Avg servig cell RSR: -86,21, Cell reselection: 0, and File transfer success: 100,00%. The main part of the interface is a table with columns: Test Case Name, Start time, Status, Duration, Folder, and Report. The table contains 14 rows of test cases, all named 'Nemo_FTP'. Most are 'Completed' (green), but two are 'Error' (red) and one is 'Cancelled' (grey).

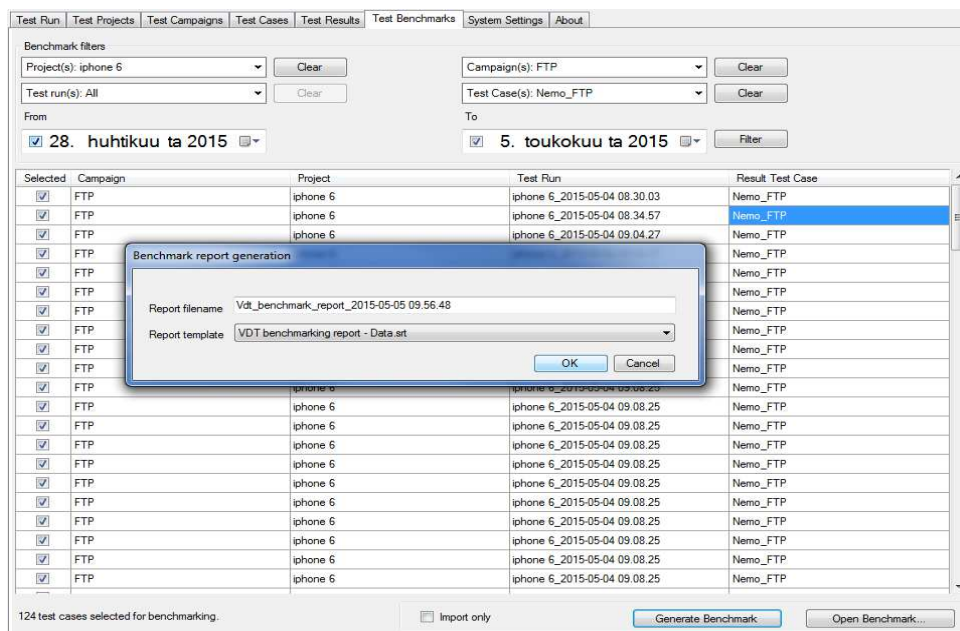
Test Case Name	Start time	Status	Duration	Folder	Report
Nemo_FTP	29.4.2015 9:44	Completed	00:01:20		
Nemo_FTP	29.4.2015 9:45	Completed	00:01:23		
Nemo_FTP	29.4.2015 9:47	Completed	00:01:12		
Nemo_FTP	29.4.2015 9:48	Completed	00:01:24		
Nemo_FTP	29.4.2015 9:49	Error	00:00:13		
Nemo_FTP	29.4.2015 9:49	Completed	00:01:21		
Nemo_FTP	29.4.2015 9:51	Completed	00:01:14		
Nemo_FTP	29.4.2015 9:52	Completed	00:01:24		
Nemo_FTP	29.4.2015 9:54	Completed	00:01:23		
Nemo_FTP	29.4.2015 9:55	Completed	00:01:26		
Nemo_FTP	29.4.2015 9:57	Completed	00:01:19		
Nemo_FTP	29.4.2015 9:58	Error	00:01:17		
Nemo_FTP	29.4.2015 9:59	Cancelled	00:00:16		

KUVA 24. Test Results

Test Benchmarks-välilehden tarkoituksena on kyetä valitsemaan yksittäisiä testejä kaikista ajetuista testeistä eri kampanjoista ja vertailemaan niitä. Oleellisia komponentteja, joiden välilehdessä tuli toimia, oli alasvetovalikot, joilla kyetään määrittelemään tietty testi tarkasti. Pystytään esimerkiksi etsimään kaikki tietyn testaajan ajamat testit tietyllä puhelimella päivän ajalta tai monen testaajan tulokset ja vertaamaan niitä Analyzessä. Ikkunasta varmistettiin, että valikot näyttivät oikeiden päivien testit ja että kalenterin tiedot olivat käytetyn tietokoneen mukaiset. Myös testien aukaisu- ja alasvetovalikoiden tyhjennyspainikkeet testattiin toimiviksi.

Ajettiin joka päivä viikon ajan muutama testikampanja. Testeihin luotiin erilaisia tuloksia, kuten aiemmin tehtiin eli luotiin virheitä, testien välistäjättämissä ja eri

mittausaikoja, jotta testauskattavuutta saatiin laajemmaksi. Tämän jälkeen avattiin Test Benchmarks ja varmistettiin, että tietyinä päivinä tiettyyn kellonaikaan tapahtunut testi näkyi taulukossa oikein määrittämällä käytetty puhelin ja suorituspäivät. Kun Benchmarks-lehdessä tietyt testit oli valittu ja painettiin Run, todettiin, että käyttäjä kykeni valitsemaan haluamansa raporttimallin tuloksien tarkastelua varten Analyze:ssä (kuva 25.) Jätettiin mitattaessa pari testiä analysoimatta ja testattiin, että ne pystyi analysoimaan jälkikäteen.



KUVA 25. Test Benchmarks

7.2.3 Mittauksien toistettavuus

Testattiin miten mittajärjestelmä kykeni toistojen tekemiseen yksisoluisessa live-testiverkossa samasta testikampanjasta ja ettei mittausaikojen välillä esiintynyt suurta vaihtelua, kun niiden tulisi pysyä lähes samoina. Koska testijärjestelmä kehitettiin toimimaan automatisoidusti, jossa testikampanjoiden mittausoiminnot tapahtuvat skriptien mukaan, oli tärkeää, että testattiin mittauksiin kulunut aika, ettei mittaamisessa tapahdu suuria heittäilyitä, jotka vaikuttaisivat merkittävästi

lopputulokseen. Mittausvaihtelujen mittausten välillä tuli olla pieniä, kun niitä verrattiin määriteltyyn aikaan. Mittaukset suoritettiin kahdella eri puhelimella (Samsung Galaxy S4 LTE-A ja Samsung Galaxy Alpha), joiden tuloksia myös verrattiin. Puhelimien molemmille antennipaikoille oli kytketty SMA-antennikaapelit (kuva 26.) Mittaukset suoritettiin laboratorion live-testiverkossa käyttäen oikeaa tukiasemaa. Tuli tarkkailla mittausajan lisäksi jälleen suorituskykyilmaisimien arvoalueita, joita mittaukset eivät missään tapauksessa saaneet alittaa tai ylittää ja niiden tuli pysyä vakaina. Tehotasoissa ei saanut esiintyä merkittäviä eroja, kuten hetkellisiä piikkejä tai katkoksia, lukuunottamatta 10 sekunnin odotusaikaa, joka asetettiin Outdoor-skriptiin. Tuli kumminkin huomioda, ettei yksisoluisessa live-verkossa, jossa usea käyttäjä liikkuu vapaasti yhdellä taajuusalueella käyttäen eri tiedonsiirtonopeuksia voida saada täysin toistettavia KPI-tuloksia. Tämä siksi, koska radioyhteys muuttuu hiukan kaiken aikaa tukiaseman ja puhelimen välillä, kun tukiasema tekee tiedonsiirtokanavien perusteella skedulointia eli päätöksiä kuka lähettää dataa ja milloin. Arvojen tuli liikkua kumminkin samalla tasolla.



KUVA 26. Testauksessa käytetyt puhelimet

Molemmilla puhelimilla mitattiin 40 kertaa 50 sekunnin kestävää FTP-testiä, jossa zip-tiedosto vastaanotettiin määritellystä osoitteesta LTE-verkossa. Tämä toistettiin vielä toisen kerran, jotta saatiin vertailutuloksia. Eli automaatio oli

yhtäjaksoisesti päällä noin 66 minuuttia. Mittaukset vietiin Minitab-ohjelmistoon, jossa Gage R&R-toiminnon avulla kyettiin laskemaan toistettavuuden mittausrheprosentti. Tulokset näkyvät lopputulokset-osiossa.

7.2.4 Valmiiden testitapausten verifiointi

Testitapauksia järjestelmän mukana tuli useita, joissa osassa käytettiin eri emulaatioita, verkkoteknologioita sekä solu- ja antennimääriä. Testattava versio erosi rinnalla kehitettävästä vastaavasta sovelluksesta, jossa käytettiin simuloitua verkkoa. Kytkennät tehtiin testien tietojen perusteella ja käyttöliittymän toimintojen tuli vastata aiemmin läpikäytyä vaikkakin skriptit ja mitattavat KPI:t vaihtelivatkin. Testitapaukset pääosin verifioi eri testaja Tanskassa, mutta Oulussa tarkoituksena oli testata kolme ensimmäistä testitapausta, jotka teknologiakylässä sijaitsevassa EXFO:lta vuokratussa Atlas-laboratoriossa oli mahdollista mitata. Siellä pystyttiin suorittamaan usean solun mittauksia LTE-tekniikalla, muttei lainkaan W-CDMA:lla, joka sulki pari testitapausta pois. Mittauksissa käytettiin siis oikeita tukiasemia. Ajan puitteissa keskimäärin kumminkin mittaamaan vain kaksi testiä, joista ensimmäisestä saatiin tulokset työhön. Automatisoidut testit testattiin useaan kertaan verraten lopputuloksien olevan samanlaiset käyttäen samoja kytkentöjä ja asetuksia. Tuloksista muodostettiin vertailukäyrät.

Jokaisesta testitapauksesta tuli kiinnittää huomiota seuraaviin asioihin:

1. Katsotaan testiskriptistä sen toimintajärjestys, minkä jälkeen siitä luodaan kampanja ja ajetaan se. Verrataan skriptiin määritettyjä toimintoja sovelluksen suorittamiin toimintoihin ajon aikana ja todetaan ne oikein suoritetuiksi.

2. Ajetaan testikampanja sille tarkoitetulla kytkennällä ja asetuksilla ja varmistetaan, että testien suoritusajat pysyvät stabiileina, edellyttäen että kytkennöissä ei tapahdu muutoksia. Testikampanja muodostetaan suuresta määrästä testitapauksia, jotta saadaan mahdollisimman paljon vertailutuloksia. Suoritetaan testit eri puhelimilla, verrataan tuloksia. Niiden välillä ei tulisi olla merkittäviä eroja.
3. Tehdään useita mittauksia ja tarkkaillaan mittauksen aikana sekä lopussa, kuhunkin mittaukseen valittuja KPI-arvoja. Varmistetaan, että järjestelmä kykenee pitämään ne sallituissa rajoissa mahdollisimman stabiileina loppuun asti ilman virheitä tai kaatumisia. Verrataan tuloksia avaamalla ne erikseen Nemo Outdooriin sekä Analyzeen ja asettamalla tuloskäyriä limittäin, jotta voidaan havaita, ettei automaattiset toiminnot vaikuta mitattuihin arvoihin. KPI-arvojen kun tulee olla keskimäärin samanlaisia manuaalisesti ja automaattisesti mitattuna.

1) Testitapaus 1

Ensimmäinen FTP-testi oli varsin yksinkertainen, jossa taajuuskaistana käytettiin band1:stä ja kytkennäksi 2x2 MIMO sekä emulaationa käytettiin 3GPP:n määrittelemää EVA 70Hz-viivehajontamallia jossa käytetään keskisuurta viivelevitystä. Testissä varmistettiin, että tiedonsiirtonopeus laski sitä mukaan, kun skriptiin määritelty signaali-kohinasuhde asetettiin laskemaan 30 sekunnin välein. Mittauksen aikana SNR-arvojen tuli alkaa 30 desibelistä ja laskea aina -5 dB asti.

2) Testitapaus 2

Toisena testinä oli myös FTP-testi, jossa käytettiin samaa EVA-kanavamallia, mutta testissä tehtiin handover 30 sekunnin sykleillä, jossa tehoalue oli -80 – -110 dBm. Testissä mitattiin handoverien selkeää onnistumista ja tiedonsiirtonopeutta. Kytkenä käytettiin kahta 2x2 MIMO LTE taajuuskaista 1:stä sekä yhtä vastaavaa taajuuskaista 2:lla.

7.3 Järjestelmätestit

7.3.1 Kuormitustesti

Testin tarkoituksena oli varmistaa, että automatisoidusti toimiva testijärjestelmä kestää suuriakin testimääräajoja ja myös pitkäkestoisia semmosia eli saada testijärjestelmä sen ääri rajoille. Testattiin, ettei mittausajankesto vaikuta tuloksiin. Samalla myös tarkasteltiin mitattuja tuloksia ja varmistettiin, ettei suorituskyky heikentynyt ajan myötä tai aiheuttanut muita mittausvirheitä, kun järjestelmä automaattisesti käynnistää ja sulkee mittauksia ja laitteita.

Testikytkentänä käytettiin aikaisemmin käytettyä 2X2 MIMO:a. Testitapauksiksi valittiin myös aikaisemmin itse luotu FTP-skripti, mutta skriptien XML-tiedostoihin muutettiin ajoaika 60 sekunnin mittaiseksi. Avattiin pääsovellus ja siirryttiin Test Campaign-välilehteen, jossa valittiin testejä 300 kappaletta ja käynnistettiin ajo. Tarkasteltiin mittausaikojen lisäksi suorituskykyilmaisimista RSRP:n ja CQI-arvojen stabiilisuutta, jotka ovat tärkeitä puhelimen, tukiaseman ja verkon välisen yhteyden suorituskyvyn kannalta. Keskiarvoistetun signaalin voimakkuuden ei tulisi heitellä montaa dBm:ää ja kanavan laadun tulisi pysyä 0-15 välillä. Tuloksista muodostettiin lopuksi normaalijakauma-käyrät suorituskyvyn havainnointiin.

7.3.2 Toipuvuustesti

Testien tarkoituksena oli varmistaa, ettei testijärjestelmä kaadu fataalien virheiden ilmetessä, vaan osaa toimia hallitusti eli käytännössä irrotetaan moduuleita yksitellen suorituksen aikana. Toipuvuuden testaus on erittäin tärkeä osa verifiointia, koska järjestelmä muodostuu monista eri moduuleista, joita kaikkia tarvitaan onnistuneen mittauksen suorittamiseksi.

- 1) Käynnistettiin ajo ja annettiin mittauksen olla päällä jonkin aikaa, kunnes irroitettiin mittauksessa käytettävä puhelin testijärjestelmästä kesken

mittauksen. Tällöin sovelluksen loki-ikkunaan tuli ilmestyä punainen ilmoitus, ettei matkapuhelinta löytynyt ja loput testitapaukset mitä kampanjasta oli jäljellä tuli sivuuttaa asettaen statukseen Error-tekstin.

- 2) Tehtiin muuten sama operaatio, mitä äsken suoritettiin mutta tällä kertaa irroitettiin Prosim testijärjestelmästä kesken mittauksen, jolloin käytettävän skenaarion käyttöön saatiin virhe. Käyttöliittymään tuli ilmestyä teksti, ettei Prosimia oltu kytketty, päättäen mittauksen hallitusti.

- 3) Irroitettiin Nemo Outdoorissa ja Analyzessä vaadittava lisenssitikku kesken mittauksen. Kun lisenssitikku irroitettiin tietyn testitapauksen suoritusajana, mittausajan päätyttyä tuli sovelluksen asettaa mittaus virheelliseksi, perustuen Outdoorissa/Analyzessä tulleeseen virheeseen, minkä jälkeen päättää mittaus hallitusti. Testattiin myös, että tikun hetkellinen irroitus ei aiheuttanut kaatumista.

- 4) Varmistettiin, ettei sovellus sekoja mittaustulosten tallentamisen kanssa, kun se suljetaan Task Managerista kesken mittauksen. Siihen asti, mihin mittauksissa oltiin päästy, tuli olla tallennettuna onnistuneesti kohdekansioon. Mittauksen ei tullut jatkua siitä mihin jäätiin, kun ohjelma käynnistettiin uudestaan vaan alkaa niin sanotusti tyhjältä pöydältä.

8 LOPPUTULOKSET

Toistettavuuden tulokset

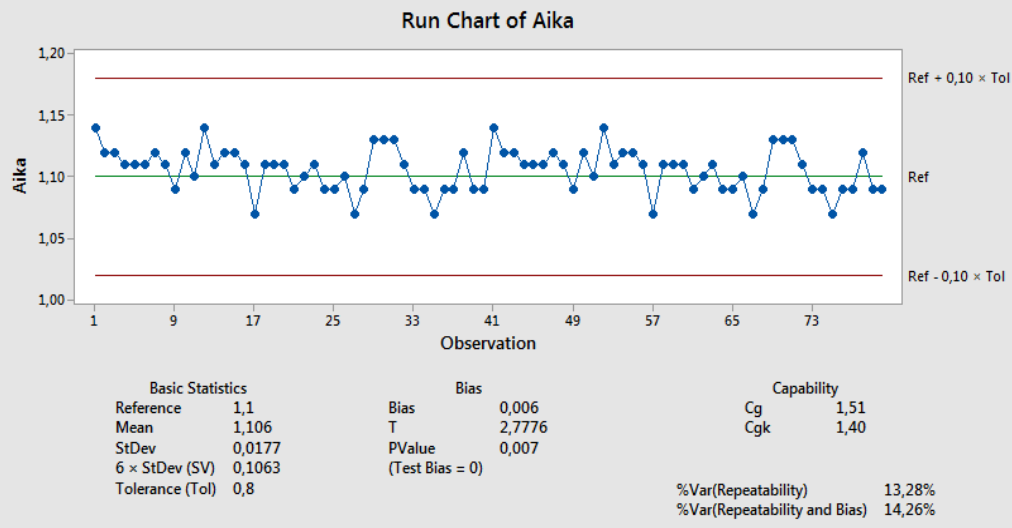
Testijärjestelmällä suoritetuista ajoista saadut mittausajat pysyivät hyvinä ja stabiileina, jolloin voidaan järjestelmän todeta olevan kykenevä toistettaviin testeihin. Kun verrataan eri puhelimilla mitattuja aikakäyriä (kuva 27 ja kuva 28), huomataan, että molemmilla puhelimilla mitatut toistotulokset pysyivät hyvin lähellä toisiaan. Mittauksissa esiintyi vain muutaman sekunnin heittelyitä, minkä ei katsottu olevan suuri. Sekuntien heittelyt selittyvät sillä, että välillä tietokoneella saattaa kestää hiukan kauemmin etsiä tai käynnistää automaattisesti laitteita ruuhkaisemman suorituskyvyn kannalta. Oleellisinta oli, etteivät ajat heitelleet suuremmin.

Samsung Galaxy S4 LTE-A:lla mitatuille arvoille muodostettiin ensin Minitabilla graafi, josta voidaan nähdä, että 80 toiston mittaus tuloksien välillä tapahtui varsin pientä heittelyä 1,07–1,14 minuutin välillä (kuva 27.) Referenssiarvoksi asetettiin 1,1 minuuttia. Poikkeamia (bias) ilmeni odotetusti, joka voidaan nähdä vertaamalla mittauspisteitä vihreeseen referenssiiviivaan. C_g vertaa tulosten jakaumaa toleranssista, jossa tavoite arvona oli 1,33:a suuremmat luvut. Mittauksessa saatiin arvoksi 1,51, joka täten yltää vaadittuun. Minitab antoi 13,28 prosentin mittausvirheen tehdyille toistoille, mikä osoittaa todella hyvää mittajärjestelmän suorituskykyä vaatimuksiin nähden. Mittausjärjestelmän toistokyky voidaan yleensäkin todeta hyväksi, kun mittausprosentti on 20 prosentin luokkaa tai pienempi, mutta mikäli prosenttiosuus olisi ollut yli 30, ei toisto- ja suorituskyky olisi ollut riittävä.

Type 1 Gage Study for Aika

Gage name: Samsung Galaxy S4
Date of study: 10.4.2015

Reported by: Tuukka Palko
Tolerance: 0,8
Misc: 80 mittausta



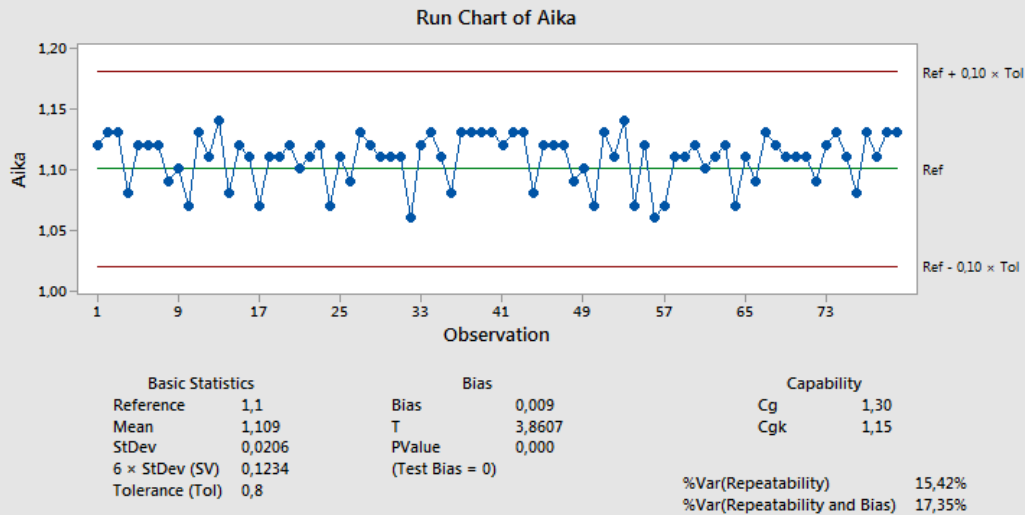
KUVA 27. Samsung Galaxy S4 LTE-A:n mittausaikojen vaihtelevuus.

Samsung Galaxy Alpha:lla mitatut saman testin tulokset eivät liiemmin eronneet edellisestä ja nekin voitiin todeta hyvin toistetuiksi (kuva 28.) Siinä 80 toistokerran välillä heittäilyä tapahtui 1,06—1,13 minuutin välillä. Referenssi- ja toleranssiarvoiksi oli valittu samat kuin aiemminkin. C_g -arvoksi saatiin 1,30, mikä on vielä hyvä tulos, vaikka jäätiin 0,03 tavoitearvosta. Toistettavuuden mittausvirheelle saatiin tulokseksi 15,42%, mikä on hyvä. Voidaan täten todeta, että saman testin suoritus aika pysyi eri puhelimilla mitattuna vakaana.

Type 1 Gage Study for Aika

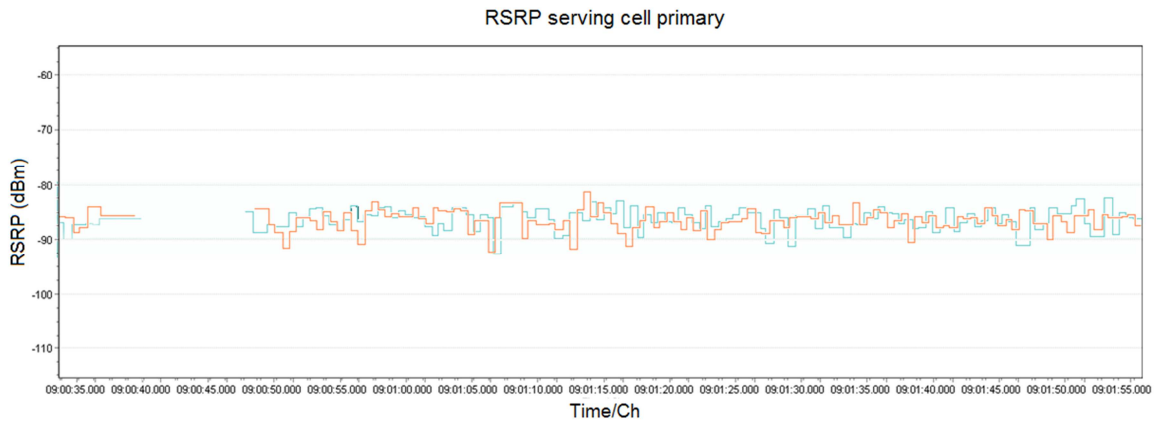
Gage name: Samsung Galaxy Alpha
Date of study: 10.4.2015

Reported by: Tuukka Palko
Tolerance: 0,8
Misc: 80 mittausta

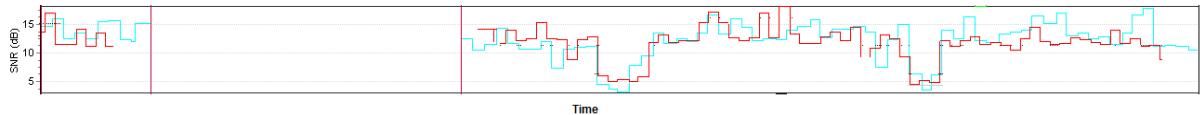


KUVA 28. Samsung Galaxy Alpha:n mittausaikojen vaihtelevuus.

Verrattiin suorituskykyilmaisimien stabiilisuutta RSRP- ja SNR-käyrien vertailulla eri aikoina mitatuista samoista testeistä ja katsottiin, että mittaus pysyi samalla tasolla kun Outdoor-skriptin alkupuolelle asetettiin 10 sekunnin odotusaika. Tasot pysyivät useaan kertaan tehdyissä mittauksissa samoilla tasoilla, kuten kuvissa 29 ja 30 olevista kahden mittauksen limityksistä voidaan nähdä. Signaalin voimakkuudessa ei tapahtunut suurempia muutoksia, vain muutaman dBm:n heittelyitä, eikä signaalin sisääntulon herkkyydessäkään tapahtunut merkittäviä muutoksia.



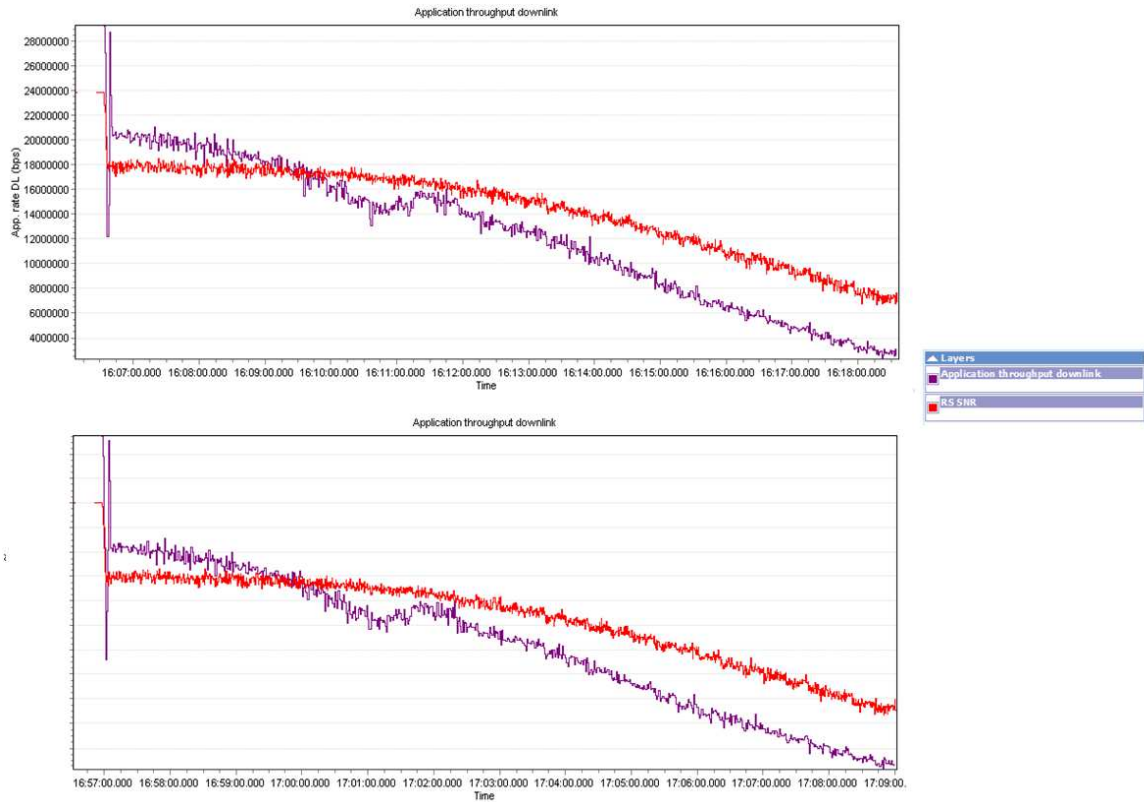
KUVA 29. Kahden mittauksen RSRP-käyrien vaihtelevuus.



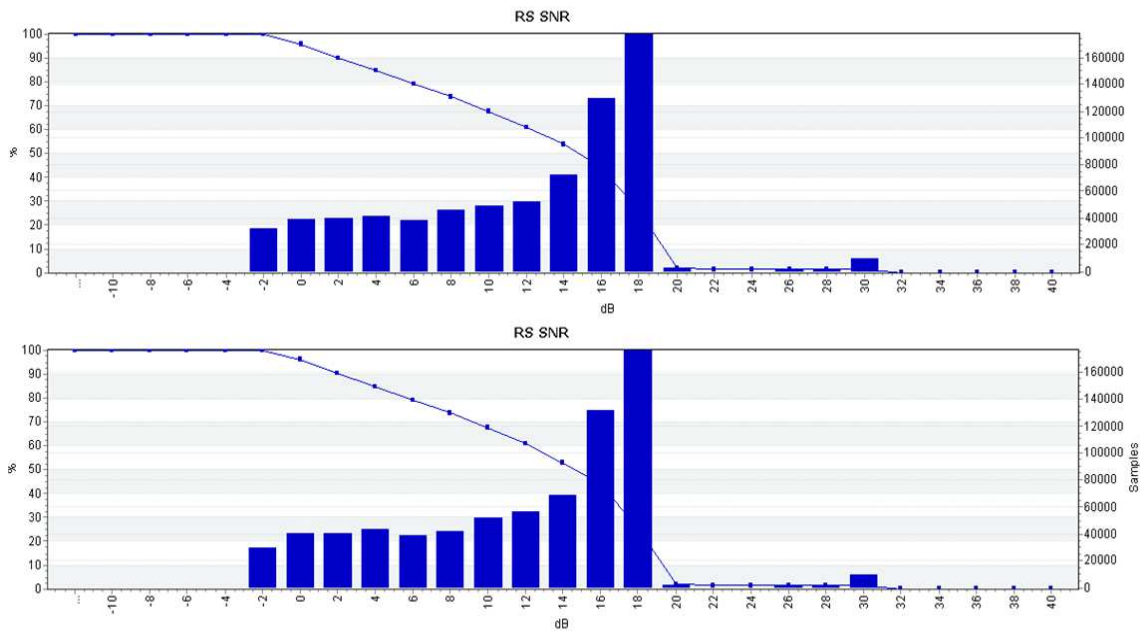
KUVA 30. Kahden mittauksen SNR-käyrien vaihtelevuus.

Testitapauksien tulokset

Ensimmäisessä testissä Excel-tulostiedostoon piirrettyjen graafien avulla voidaan todeta, että tiedonsiirtonopeus laski odotetusti signaali-kohinasuhteen mukaan ja se pystyttiin toistamaan ilman suurempia vaihteluita. Kuten kahdesta eri aikoihin suoritetusta mittauksesta (kuva 31 ja kuva 32) voidaan nähdä, nopeus lähtee vähitellen laskemaan 21 Mbit/s:sta aina 4 Mbit/s:n asti, kun signaali-kohinasuhdetta pienennettiin.



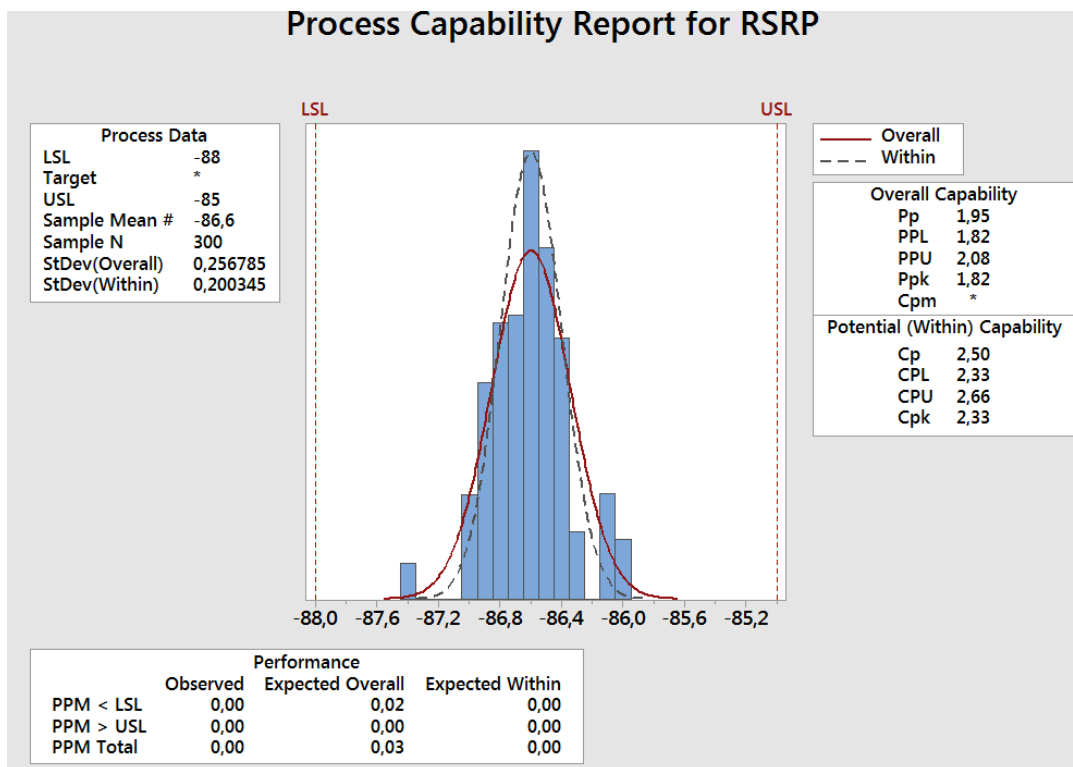
KUVA 31. Ensimmäisen testitapauksen tiedonsiirtonopeuden vertailu.



KUVA 32. Ensimmäisen testitapauksen SNR vertailu.

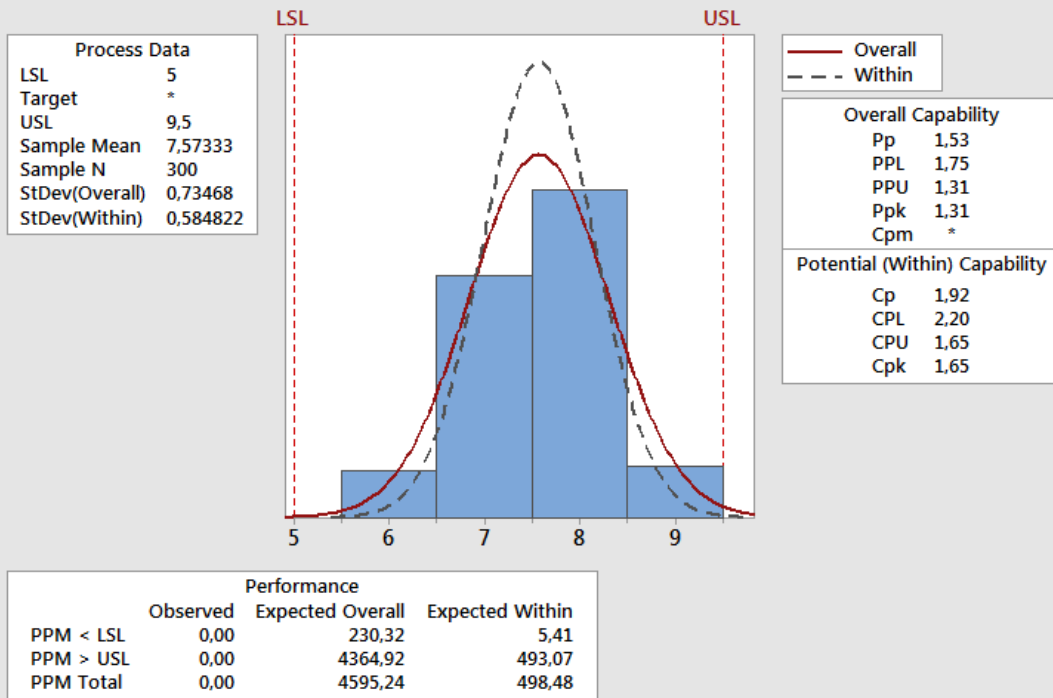
Suorituskyvyn mittaustulokset

Järjestelmätestausvaiheessa suoritettu suorituskykymittauksista luotiin Minitab-ohjelmistolla Gaussin käyrät 300 mittauksesta, joista voitiin huomata matkapuhelimella verkosta mitatun signaalin voimakkuuden ja kanavanlaatu-tulosarvojen jakautuneen normaalijakauman mukaisesti, eli että arvot olivat jakautuneet suhteellisen tasaisesti keskiarvorajan molemmin puolin. Täten voidaan todeta testijärjestelmän olevan kykenevä pysymään riittävän stabiilina pitkiä aikoja, mahdollistaen suurien testimäärien ajon ja siten hyvän suorituskyvyn. C_{pk} kuvaa suorituskykyä ja mitä suurempi se on sitä parempi suorituskyky on. Mikäli se olisi alle yhden ei se olisi ollut riittävä. Kuten kuvista 33 ja 34 voidaan nähdä signaalin voimakkuuden mittauksissa se oli 2,33 ja kanavan laadulla 1,65.



KUVA 33. Suorituskykymittaus RSRP:lle

Process Capability Report for CQI



KUVA 34. Suorituskykymittaus CQI:lle

9 YHTEENVETO

Työn aiheena oli verifioida automatisoidun mobiiliverkkojen testaukseen kehitetyn järjestelmän toiminta, joka tapahtui graafisen käyttöliittymän kautta. Läpikäynnin, testien ja mittausten perusteella voidaan todeta järjestelmän toimivan hyvin.

Testijärjestelmä rakentui monesta eri osasta, joista Prosimin ja Nemo Outdoorin käyttö olivat minulle suhteellisen tuttuja. Vastaan tuli pitkin matkaa erilaisia ongelmia, jotka saatiin ratkaistuksi pienen mietinnän jälkeen.

Virheitä käyttöliittymästä löytyi yllättävän paljon. Niiden syntymisen syyt ja aiheuttajat saatiin löydettyä ja raportoitua ohjelmoijien tietoisuuteen. Käyttöliittymän verifiointitestauksen jälkeen sen toiminta oli luotettavaa. Mittauksia tehdessä suurimpana ongelmana oli viimeisimpien Outdoor- ja Analyze- versioiden yhteensopivuusongelmat käyttöliittymän kanssa, mikä hidasti välillä testimittausten tekemistä. Loppujen lopuksi mittaukset kumminkin saatiin tehtyä onnistuneesti ja mittaustulosten pohjalta voidaankin todeta järjestelmän olevan kykenevä toistettaviin ja stabiileina pysyviin tuloksiin. Valmiista testitapauksista kerkesin verifioida vain kaksi, joista loppujen testaus siirtyy tuonnemmaksi.

Ainahan tietenkin omassa työskentelyssä jää parannettavaa, etenkin kun on iso kokonaisuus. Olisi voinut joiltakin osin esimerkiksi tehdä mittauksia laajemminkin ja tarkemmin mutta testausaikaa ei ollut loputtomasti. Työtä kyetään jatkamaan moneltakin kantilta järjestelmän kehittyessä. Voidaan suorittaa mittauksia eri verkkoteknologioilla ja tekemään muitakin KPI-mittauksia.

LÄHTEET

1. Rauhala, Eija 2010. Ohjelmistotestauksen suunnittelu - Case: A-lehdet Oy:n laskujen tulostusohjelma. Haaga-Helia ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö. Saatavissa: http://www.theseus.fi/bitstream/handle/10024/23687/Rauhala_Eija.pdf?sequence=1 Hakupäivä: 12.3.2015.
2. White-Box unit testing-A bottom-up approach of software testing, 2011. Software testing genius. Saatavissa: <http://www.softwaretestinggenius.com/white-box-unit-testing-a-bottom-up-approach-of-software-testing>. Hakupäivä 10.3.2015.
3. Paakki, Juha, 2003. Ohjelmistotuotanto. Tietojenkäsittelytieteen laitos. Saatavissa: <http://www.cs.helsinki.fi/u/paakki/ohtuk03-luento12-bw.pdf> Hakupäivä 19.3.2015.
4. Vakevainen, Petri 2001. Testaussuunnitelma. SoberIT. Saatavissa: <http://www.soberit.hut.fi/tik-76.115/00-01/palautukset/groups/Osprey/t4/testaussuunnitelma.html>. Hakupäivä 17.3.2015.
5. Lintula, Heli 2004. Vaatimusten validointi ja verifiointi. Kuopion yliopisto. Tietojenkäsittelytieteen laitos. Pro gradu -tutkielma. Saatavissa: <http://www.cs.uku.fi/tutkimus/Teho/helingu.pdf>. Hakupäivä 15.2.2015.
6. Katara, Mika – Vuori, Matti – Jääskeläinen, Antti 2014. Ohjelmistojen testaus. Tampereen teknillinen yliopisto Saatavissa: <http://www.cs.tut.fi/~otekn/materiaali/testauskalvot-otekn.pdf>. Hakupäivä 19.3.2015.

7. Hietamies, Otto 2013. Automatisoitu testausjärjestelmä sulautettujen ohjelmistojen regressiotestaukseen. Tampereen teknillinen yliopisto. Signaalinkäsittelyn ja tietoliikennetekniikan koulutusohjelma. Diplomityö. Saatavissa:
<http://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/21923/Hietamies.pdf?sequence=1&isAllowed=y> Hakupäivä: 22.3.2015.
8. Laatu ja testaus: Automatisointi, 2012. TestausOSY. Saatavissa:
<http://testausosy.fi/wp-content/uploads/2012/11/LT-Vol1Ed2.pdf>.
Hakupäivä 22.3.2015.
9. Kullas, Tomi – Smedberg, Olli 2013. Matkapuhelinverkkojen neljäs sukupolvi: 4G LTE. Tampereen ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö. Saatavissa:
https://www.theseus.fi/bitstream/handle/10024/68640/Kullas_Tomi_Smedberg_Olli.pdf?sequence=1 Hakupäivä: 23.3.2015.
10. Matkapuhelinverkko, 2014. STUK. Saatavissa:
http://www.stuk.fi/sateilyn-hyodyntaminen/matkapuhelimet-ja-tukiasemat/fi_FI/matkapuhelinverkko/ Hakupäivä 16.3.2015.
11. Maniezzo, Daniela, 2004. Horizontal and Vertical Handoff. Saatavissa:
<http://www.cs.ucla.edu/ST/research.html>. Hakupäivä 22.4.2015.
12. Nurminen, Jukka, 2010. Soluverkot. Aalto-yliopisto, teknillinen korkeakoulu. Saatavissa:<http://www.cse.tkk.fi/fi/opinnot/T-110.2100/2010/luennot-files/Solukoverkot%20%28luonnos%29.pdf>.
Hakupäivä 22.4.2015.
13. Penttinen, J, 2001. GSM-tekniikka: Järjestelmän toiminta ja kehitys kohti UMTS-aikakautta. Porvoo: WSOY.

14. Chen, Michael, 2011. Signal Space. Saatavissa:
<http://urbanomnibus.net/2011/07/signal-space>. Hakupäivä 22.4.2015.
15. Penttinen J–Penttinen P, 2011. LTE osana varmentavaa viestintäjärjestelmää. Viestimies 3, 43–47.
16. Brigatti, K 2011. Nukkuvan solun havaitseminen itseorganisoituissa LTE-radioverkoissa. Jyväskylän yliopisto, Tekniikan laitos. Pro gradu - tutkielma. Saatavissa:
<https://jyx.jyu.fi/dspace/bitstream/handle/123456789/37429/URN%3aNB%3afi%3ajyu-201202211242.pdf?sequence=1> Hakupäivä: 15.3.2015.
17. Pietiläinen, Kimmo 2010. LTE tulevaisuuden mobiiliverkkoratkaisuna. Lappeenrannan yliopisto. Tietotekniikan osasto. Kandidaattityö. Saatavissa: <http://www.doria.fi/bitstream/handle/10024/69647/nbnfi-fe201105171574.pdf?sequence=3> Hakupäivä: 15.3.2015.
18. Vartiamäki, Ville 2012. 3GPP Long Term Evolution. Tampereen ammattikorkeakoulu. Tekniikan koulutusohjelma. Opinnäytetyö. Saatavissa:
http://www.theseus.fi/bitstream/handle/10024/43296/Vartiamaki_Ville.pdf?sequence=1 Hakupäivä: 13.3.2014.
19. Frenzel, Lou, 2013. Fundamentals of communications access technologies. Electronic design. Saatavissa:
<http://electronicdesign.com/communications/fundamentals-communications-access-technologies-fdma-tdma-cdma-ofdma-and-sdma>. Hakupäivä 15.3.2015.
20. Teletopix, 2011. Multicarrier technologies in LTE. Saatavissa:
<http://www.teletopix.org/4g-lte/multicarrier-technologies-in-lte/>. Hakupäivä 15.3.2015.

21. Teletopix, 2014. LTE MIMO types of inputs and outputs. Saatavissa: <http://www.teletopix.org/4g-lte/lte-mimo-4g-lte/lte-mimo-types-of-inputs-and-outputs/>. Hakupäivä 15.3.2015.
22. Sainju, Prabhat Man 2012. LTE performance analysis on 800 and 1800 MHz bands. Tampereen yliopisto. Tekniikan osasto. Opinnäytetyö. Saatavissa: <http://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/21173/sainju.pdf?sequence=1&isAllowed=y> Hakupäivä: 13.3.2015.
23. La Rocca, Maurizio, 2015. RSRP and RSRQ measurement in LTE. Saatavissa: <http://www.laroccasolutions.com/training/78-rsrp-and-rsrq-measurement-in-lte>. Hakupäivä 3.3.2015.
24. Brügger, Thomas. R&S TS8980 test system analyzes LTE quality indicators: CQI, PMI and RI. Saatavissa: http://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_news_from_rs/203/NEWS_203_english_TS8980.pdf. Hakupäivä 1.4.2015.
25. Kurki, Jouko, 2010. Tiedonsiirron perusteet. Aalto-yliopisto, teknillinen korkeakoulu. Saatavissa: http://www.cse.hut.fi/fi/opinnot/T-110.2100/2010/luennot-files/Tiedonsiirron_2.pdf. Hakupäivä 2.3.2015.
26. Measuring network throughput, 2015. Wikipedia. Saatavissa: http://en.wikipedia.org/wiki/Measuring_network_throughput. Hakupäivä 2.3.2015.
27. Nemo Outdoor – the ultimate drive test tool for wireless networks, 2015. Anite. Saatavissa: <http://www.anite.com/businesses/network->

- [testing/products/nemo-outdoor-ultimate-drive-test-tool-wireless-networks#.VRxJBeG1doN](#). Hakupäivä 3.3.2015.
28. Karjalainen, Tanja 2014. Informaation luotettavuus on usein kehityksen este. Quality knowhow Karjalainen Oy. Saatavissa: <http://www.qk-karjalainen.fi/fi/artikkelit/mittaussysteemin-vaihtelu/> Hakupäivä 5.4.2015.
29. Prosim channel emulation solutions, 2013. Anite. Saatavissa: http://www.anite.com/prosim#.VM_DxS61d6I Hakupäivä 5.4.2015.
30. The evolved packet core, 2015. 3GPP. Saatavissa: <http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core> Hakupäivä 22.5.2015.