

Jari Santikko

RADIOVERKKOTUKIASEMAHOJAIMEN OHJELMALOHKON  
MODUULITESTAUKSEN TUTKIMINEN UUDELLA TYÖKALULLA

Tietotekniikan koulutusohjelma  
2014

Santikko, Jari  
Satakunnan ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Joulukuu 2014  
Ohjaaja: Trast, Ismo  
Sivumäärä: 32  
Liitteitä:

Asiasanat: Radioverkko, Moduulitestaus, 2G, Ohjelmalohko, Tutkinta

---

Tämän opinnäytetyön tavoitteena on tutkia miten uuden työkalun avulla onnistutaan toteuttamaan radiotukiasemaohjajaimen ohjelmalohkon moduulitestaus. Tämän työkalun kaltaista työkalua ei vielä ole käytössä ja se mahdollistaisi testauksen niin sanotussa offline-tilassa eli työasemalta ei tarvitsisi luoda etäyhteyksiä testausympäristöön.

Työkalu on käytössä Tiedon asiakkaalla 3G-tasolla, mutta sitä ei ole otettu käyttöön vielä 2G tasoisen järjestelmän ohjelmalohkojen moduulitestauksessa. Työkalu löytyy Tiedon verkkolevyltä, mutta käyttöön sitä ei ole otettu vielä minkään ohjelmalohkon testauksessa. Tutkinta suoritetaan tietyn ohjelmalohkon osalta, mutta sitä voidaan soveltaa kaikkiin ohjelmalohkoihin, joita testataan testisanomilla ja sanomasekvensseillä. Työn ensimmäisessä osassa tutkitaan hieman mihin työkalu kykenee ja toisessa osassa selvitetään miten testauksessa olevaa ohjelmalohkoa tulisi testata. Lopuiksi verrattiin miten uusi työkalu soveltuu testauksessa olevan ohjelmalohkon moduulitestaukseen.

Työ suoritettiin osaksi Tieto Finland Oyj tiloissa Porissa, mutta suurimmaksi osaksi työ tehtiin loppuun itsenäisesti. Kirjallisessa osuudessa perehdyttiin ensin GSM-järjestelmään ja sitä kautta myös matkapuhelinjärjestelmiin. Tämän jälkeen selvitetiin hieman Nokian DX200 tuoteperhettä ja vielä lopuiksi tutkittiin ohjelmistokehitystä ja ohjelmisto testausta. Ohjelmistokehityksessä käytiin läpi ohjelmistokehitys V-mallin mukaisesti ja ohjelmistotestauksen osalta käytiin läpi testauksen eri vaiheet.

# RESEARCH OF A RADIO NETWORK BASE STATION CONTROLLER MODULE TESTING WITH A NEW TOOL

Santikko, Jari

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Information technology

December 2014

Supervisor: Trast, Ismo

Number of pages: 32

Appendices:

Keywords: Radio network, Module testing, 2G, Program segment, Research

---

The purpose of this thesis was to research new tool for base station controller's program block module testing. At the moment there is no tools like this to use. The tool could allow program block module testing to be performed in offline without any necessary connections to test bench.

Tool is already being use in 3G level in Tieto's customer but it hasn't been used in 2G level module testing yet. Tool is located in Tieto's net drive but it hasn't been used in testing for program block module test in any program blocks. Research is done for one certain program block which is tested with test messages and message sequences. New tool is researched and documented in the first part of the thesis. In second part module testing of the program block is documented. How well the new tool fits in module testing of program block is researched and documented in the last part of the thesis.

The research part was done partly in Tieto Finland premises in Pori. Research and the written part of the thesis was finished independently. First in the written part is introduction to GSM technology and mobile information technology. The Nokia DX 200 product family and the basics of Base Station Controller were also introduced. Software engineering and software testing is also introduced. Software development is introduced with V-model and various phases of testing are introduced.

# SISÄLLYS

LYHENTEET .....	5
1 ENSIMMÄINEN LUKU/ JOHDANTO .....	8
2 GSM .....	8
2.1 GSM teknologian historiaa .....	8
2.2 GSM järjestelmän rakenne .....	9
2.2.1 Keskusjärjestelmä .....	10
2.2.2 Tukiasemajärjestelmä .....	11
2.2.3 Käytönhallintajärjestelmä .....	12
2.2.4 Pakettikytkennäinen tiedonsiirtopalvelu .....	12
2.2.5 Mobiiliasema .....	13
2.2.6 Rajapinnat .....	13
2.3 Taajuudet .....	14
3 DX 200 TUOTEPERHE .....	15
3.1 DX 200 –tuoteperheen ominaisuuksia .....	16
3.2 DX 200 tukiasemaohjain (BSC) .....	17
4 OHJELMISTOKEHITYS JA TESTAUS .....	19
4.1 Ohjelmistokehitys .....	19
4.2 Ohjelmistotestaus .....	21
4.2.1 Moduulitestaus .....	22
4.2.2 Toimintotestaus .....	22
4.2.3 Järjestelmätestaus .....	22
5 WDOX .....	23
5.1 WDOX:n käyttöönotto .....	24
6 OHJELMALOHKON TESTAUS .....	26
7 MITEN WDOX SOVELTUU OHJELMALOHKON TESTAAMISEEN .....	28
8 LOPPUTULOS JA SEURAAVAT VAIHEET WDOX:IN KÄYTTÖÖNOTOSSA .....	28
8.1 Lopputulos .....	28
8.2 Seuraavat vaiheet käyttöönotossa .....	30
LÄHTEET .....	32
LIITTEET	

## LYHENTEET

Abis-interface	A-bis rajapinta
A-interface	A-rajapinta
Air-interface	Ilmarajapinta
AUC	Authentication Centre, autentikointikeskus
BCSU	BSC Signalling Unit, tukiasemaohjaimen merkinantoyksikkö
BSC	Base Station Controller, tukiasemaohjain
BSS	Base Station Subsystem, tukiasemajärjestelmä
BTS	Base Transceiver Station, tukiasema
CEPT	Conférence Européenne des Postes et Télécommunications
CLS	Clock and Synchronisation Unit, Kello- ja synkronointikeskus
EIR	Equipment Identity Register, laiterekisteri
ET	Exchange Terminal, keskuspääte
FSC	Fixed Network Switching Centre, kiinteän verkon matkapuhelinkeskus
GGSN	Gateway GPRS Support Node, yhdyskäytäväsolmu

GPRS	General Packet Radio Service, GSM-pakettidatapalvelu
GSM	Global System for Mobile communications, matkapuhelinjärjestelmä
GSWB	Group Switch, kytkentäkenttä
HLR	Home Location Register, kotirekisteri
ISDN	Integrated Service Digital Network, piirikytkennäinen puhelinverkkojärjestelmä
MB	Message Bus, sanomaväylä
MCMU	Marker and Cellular Management Unit
ME	Mobile Equipment, mobiililaite
MoU	Memorandum of Understanding
MS	Mobile Station, liikkuva päätelaite
MSC	Mobile services Switching Centre, matkapuhelinkeskus
NMS	Network Management Subsystem, verkonhallintajärjestelmä
NSS	Network and Switching Subsystem, keskusjärjestelmä
O&M	Operations & Maintenance interface, käyttö- ja ylläpitorajapinta
OMU	Operation and Maintenance Unit, käyttö ja ylläpitoyksikkö

PCM	Pulse Code Modulation, pulssikoodimodulaatio
PCU	Packet Control Unit, paketinhallintayksikkö
SGSN	Service GPRS Support Node, operointisolmu
SIM	Subscriber Identity Module, tilaajamoduuli
SS7	Signalling System No. 7, merkinantoprotokolla
TC	Transcoder, transkooderi
TNSDL	TeleNokia Specification and Description Language
TRX	Transceiver, lähetinvastaanotin
VLR	Visitor Location Register, vierailijarekisteri

# 1 JOHDANTO

Lopputyötä lähdettiin toteuttamaan tilanteessa, jossa haluttiin uusia nopeampia ja ketterämpiä tapoja radiotukiasemaohjaimen ohjelmalohkon moduulitestaukseen. Lisäksi haluttiin tutkia mahdollisuutta suorittaa ohjelmalohkon moduulitestausta ilman fyysistä testausympäristöä ja siihen tarvittavia nopeita ja vakaita tietoliikenneyhteyksiä. Tähän asti sellaista testaustapaa ei ole ollut mahdollista käyttää.

Opinnäytetyö on jaettu kolmeen osaan. Ensimmäisessä osassa (luku 2) käydään läpi GSM-tekniikkaa ja sen myötä matkapuhelinverkoja ja niiden rakennetta yleisesti. Seuraavassa osassa (luku 3) käydään lyhyesti läpi Nokian DX200 ympäristöä, jonka tukiaseman ohjelmalohkon moduulitestausta tutkitaan. Viimeisessä osassa esitellään tutkinnan alla oleva työkalu, käydään läpi, miten kyseistä ohjelmalohkoa moduulitestataan, ja lopuksi käydään läpi, miten moduulitestausta uuden työkalun avulla onnistuu.

## 2 GSM

### 2.1 GSM teknologian historiaa

1980-luvulla CEPT (European Conference of Postal and Telecommunications Administrations) perusti ryhmän, jonka tavoitteena oli saada yhteinen mobiilijärjestelmä Eurooppaan. Tähän asti Euroopassa oli käytössä monia yhteensopimattomia järjestelmiä. Mobiilijärjestelmien kasvun ja leviämisen myötä yhteisen mobiilijärjestelmän tarve oli kasvanut. Ryhmän nimeksi tuli GSM (Groupe Spécial Mobile), joka alkoi määrittelemään uutta mobiilijärjestelmää. Tästä kehitystyöstä syntyi GSM järjestelmä ja nykyään GSM lyhenne tarkoittaa ”Global System for Mobile communications”. (SYSTRA 2000, 10.)

GSM tunnetaan järjestelmänä paremmin nimellä 2G (second generation). Ensimmäinen puhelu GSM tekniikalla soitettiin vuonna 1991. Vuonna 1992 julkaistiin ensimmäinen



mäinen GSM verkko suomessa. 1990 luvun alussa muualla maailmassa huomattiin myös tarvetta yhteiselle mobiilijärjestelmälle ja siitä syystä GSM järjestelmä nousi maailmanlaajuiseksi mobiilijärjestelmäksi. USA, osa Etelä-Amerikasta ja Japani eivät ottaneet GSM järjestelmää käyttöön. PCS (Personal Communication System), joka on käytössä USA:ssa, käyttää muokattua GSM teknologiaa.

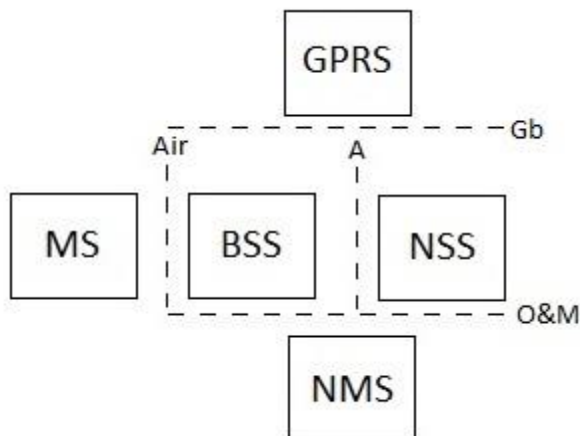
(SYSTRA 2000, 10, 12-13.)

Myöhemmin GSM järjestelmää on kehitetty. 2G:n jälkeen on kehitetty niin 3G (Euroopassa UMTS-järjestelmä) kuin 4G järjestelmät. 2G-järjestelmän päälle kehitettiin GPRS, johon viitataan kun puhutaan 2,5G teknologiasta. GPRS toi ensimmäisenä matkapuhelinverkkoihin pakettikytkentäisen ja IP-pohjaisen tietoliikennejärjestelmän. GPRS-järjestelmässä tieto kulkee samaan tapaan kuin internetissäkin, eli paketteina ja kaistan kapasiteetti riippuu pakettien määrästä. 3G mahdollisti nopeat ja monipuoliset tiedonsiirron palvelut. 3G perustu pakettikytkentäiseen nopeaan tiedonsiirtoon ja internet-yhteensopivuuteen. (Tietoliikenteen ja tietotekniikan keskusliiton www-sivut 2014)

## 2.2 GSM järjestelmän rakenne

GSM järjestelmä koostuu viidestä osasta.

- Keskusjärjestelmä (NSS)
- Tukiasemajärjestelmä (BSS)
- Verkonhallintajärjestelmä (NMS)
- Pakettikytkennäinen tiedonsiirtopalvelu (GPRS)
- Rajapinnat



Kuva 1. GSM rakenne (SYSTRA 2000, 15)

### 2.2.1 Keskusjärjestelmä

Keskusjärjestelmä (NSS, Network Switching System) koostuu matkapuhelinkeskuksesta (MSC, Mobile Services Switching centre) sekä siihen yhteydessä olevista rekistereistä. Näitä rekistereitä ovat kotirekisteri (HLR, Home Location Register), vierailijarekisteri (VLR, Visitor Location Register), Laitetunnusrekisteri (EIR, Equipment identity register) ja tunnistuskeskus (AuC, Authentication centre). Keskusjärjestelmän tehtävä on kytkeä GSM-verkon ulkopuolelta tulevat ja GSM-puhelinten väliset piirikytkentäiset yhteydet kiinteiden verkkojen määrittelyiden mukaisesti. Keskusjärjestelmän tehtävä on myös kytkeä verkon sisäiset puhelut.

Matkapuhelinkeskuksen tärkein tehtävä on kytkeä, ylläpitää ja purkaa puhelut omalla alueellaan tai kytkeä puhelu edelleen seuraavalle matkapuhelinkeskukselle. Jos yhteys ulkopuolisesta verkosta syntyy matkapuhelinkeskuksen alueelle, mutta jonka alueella vastaanottava tilaaja ei kuitenkaan ole, reitittyy puhelu oikealle matkapuhelinkeskukselle. Kuhunkin keskukseseen liittyy A-rajapinnalla yksi tai useampi tukiasemaohjain. Keskuksella voi olla myös yksi tai useampi yhteys ulkopuolisiin puhelin- tai ISDN-verkon keskuksiin.

Kotirekisteriin on tallennettu tilaaja- ja laskutustietoja. Kotirekisteristä löytyy myös tilaajan lisäpalvelut. Pysyvinä tietoina kotirekisteristä löytyy tilaajan kansainvälinen

ISDN-numero (MSISDN, Mobile Subscriber ISDN Number), kansainvälinen tilaaja-tunnus (IMSI, International Mobile Subscriber identity), GPRS tiedot, salausparametrit sekä liittymän tyyppi. Lisäksi kotirekisteriin talletetaan muuttuvina tietoina tilaajan tavoitettavuuteen liittyvät tiedot eli tieto tilaajan kytkeytymisestä verkkoon, reititystiedot sekä käyttäjän puhelunsiirrot ja muut palvelut.

Kun tilaaja siirtyy uudelle keskusalueelle, pyydetään kyseisen alueen VLR tilaajan tiedot kotirekisteristä. Näitä tietoja säilytetään VLR:ssä sen aikaa, kun tilaaja on sen alaisuudessa. VLR:ssä säilytetään tilaajan MSISDN-numeroa, IMSI-numeroa, tilaajan tilapäistä tunnusta (TMSI, Temporary Mobile Subscriber identity), roaming-numeroa (MSRN, Mobile Station Roaming Number) ja sijaintialueen tiedoa Vieraskirjelmästä löytyy myös salaustiedot ja tilaajan palvelutiedot.

Matkapuhelimilla on oma yksilöllinen laitetunnus, joka on talletettu operaattorilla laiterekisteriin (EIR, Equipment Identity Register). Laiterekisteristä verkko voi mahdollisesti tarkistaa laitteen tiedot ja tämän tiedon avulla keskus voi estää laitteen toiminnan sen verkossa esimerkiksi silloin, kun laite on merkitty varastetuksi.

Tilaajan liittyessä verkkoon, määritellään jokaiselle tilaajalle omat salaiset tunnistenumerot. Näitä tunnistenumeroita säilytetään tunnistuskeskuksessa (AuC, authentication Centre). Puhelunmuodostuksessa tunnistuskeskuksessa olevia tunnistenumeroita verrataan tilaajan lähettämiin numeroihin ja puhelu voidaan estää, mikäli tilaajalla ei ole oikeutta käyttää kyseistä verkkoa.

(Penttinen, 2006, 129-131.)

### 2.2.2 Tukiasemajärjestelmä

Tukiasemajärjestelmä koostuu tukiasemista (BTS, base transceiver station) ja tukiasemaohjaimista (BSC, base station controller). Tukiasemaohjain ohjaa tukiasemien toimintaa. Tukiasemajärjestelmästä löytyy myös transkooderi- ja nopeudensovituslaitteisto (TC, transcoder, tai TRAU, transcoder/rate adapter). Transkooderi voi sijaita fyysisesti myös matkapuhelinkeskuksen laitetilassa.

Tukiasema koostuu TRX-elementeistä tehonlähteineen ja virtalaitteineen, yhdysuotimesta, tehojakajasta, antennikaapeleista, salausrakenteista, antennista sekä mastosta ja mastovahvistimista. (Penttinen, 2006, 122.)

Tukiasemaohjain eli BSC hoitaa oman alueensa radioresurssien hallinnan (RRM, radio resource management). MSC hoitaa siten puhelujen kytkennän BSC:n kautta MS:lle. Yhteydenaikaisista radiopinnan tapahtumista huolehtii BSC. Jokaisen tukiasemaohjaimen alueella on monta tukiasemaa (BTS, Base Transceiver Station), jotka muodostavat sijaintialueita. Sijaintialue voi olla myös kokonaisuuksia tai osia useammasta BSC:n alaisuudessa olevasta alueesta. MS:n siirtyessä sijaintialueelta toiselle, ilmoittaa MS siitä tukiasemaohjaimelle. Tukiasema ilmoittaa keskuksen vierailijarekisteriin MS:n uuden sijaintialueen, jonka jälkeen verkko osaa reitittää MS:lle saapuvat puhelut oikeaan paikkaan.

Transkooderit hoitavat puheen koodauksen ja dekodauksen sekä datan nopeussovituksen GSM-verkon ja ulkopuolisen verkon välillä. GPRS ympäristössä transkooderia vastaa paketinhallintayksikkö (PCU, Packet Control Unit).

### 2.2.3 Käytönhallintajärjestelmä

Käytönhallinnalla on useita tehtäviä. Käytönhallinta on yhteydessä joko BSS:n tai NSS:n kanssa. Tärkeimmät tehtävät ovat verkon käyttö ja kunnossapito, tilaajien hallinta sekä viestien hallinta. Näitä voidaan hallita yhden tai useampien käyttö- ja kunnossapitokeskusten kautta. Niiden kautta voidaan GSM-verkkoelementtien osien ohjelmistoja asentaa tai parametreja muuttaa. Niiden kautta voidaan myös valvoa verkkoa ja sen tapahtumia. (Penttinen, 2006, 134.)

### 2.2.4 Pakettikytkennäinen tiedonsiirtopalvelu

GPRS ei ole GSM:n erillinen järjestelmä vaan datapalvelu. GPRS:n avulla GSM-verkosta saadaan luotua pakettikytkentäiset yhteydet Internet-palveluihin. GPRS käyttää samoja resursseja kuin tavalliset GSM-yhteydet. Yhteiselle radorajapinnalle

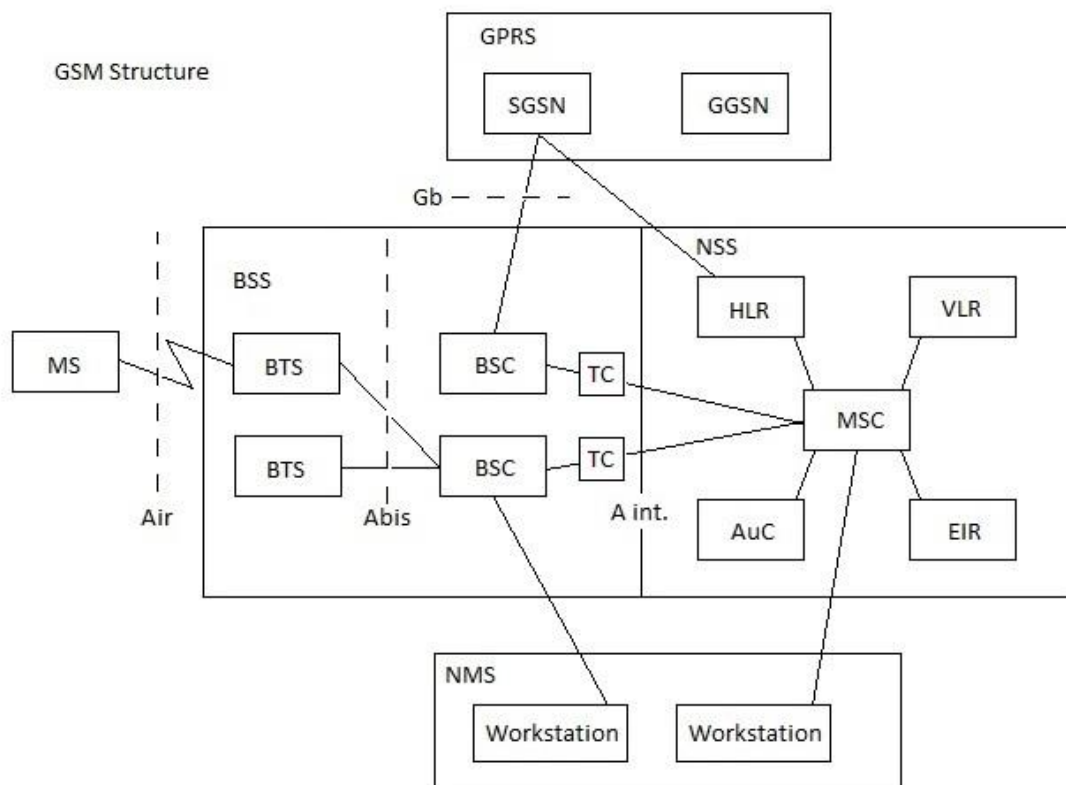
voidaan kytkeä puheluita, piirikytkentäisen datan käyttäjiä ja GPRS:n käyttäjiä. Siitä huolimatta että GPRS käyttää samaa radorajapintaa, ohjataan GPRS-yhteydet erilliseen GPRS-runkoverkkoon. Tätä varten BSC:ssä on erillinen paketiinohjausyksikkö (PCU, PAcKet Control Unit), jolla kytkeydytään GPRS-runkoverkon elementtiin. Tämän elementin nimi on SGSN (Serving GPRS Support Node) ja se huolehtii GPRS-päätelaitteiden liikkuvuuden hallinnasta ja radorajapinnan salauksesta. GPRS-runkoverkon ja Internetin välissä on GGSN (Gateway GPRS Support Node), joka hoitaa käytännössä reitittimen tehtäviä verkkojen välissä. GPRS-laitteen liikkueessa verkon alueella, GGSN osaa ohjata datapaketit oikealle SGSN:lle. (Penttinen, 2006, 158-161.)

### 2.2.5 Mobiiliasema

MS eli mobiiliasema on käyttäjälle näkyvä osa GSM-verkosta. Käytännössä MS on päätelaite eli ME (Mobile Equipment), kuten matkapuhelin. Päätelaitteessa on talleassa laitteen IMEI (International Mobile Equipment Identity), jonka verkko voi tarvittaessa kysyä laitteelta. Lisäksi tarvitaan aina SIM-kortti, jonka avulla tilaaja tunnistetaan verkosta. SIM-kortissa on tallennettuna tilaajan IMSI (International Mobile Subscriber Identity), jonka avulla tilaaja tunnistetaan verkossa. SIM-kortilla on myös tilaajan salaustiedot. Ilman SIM-korttia ei puheluita pysty muodostamaan muihin kuin hätänumeroihin. (Svein Willassenin www-sivut 2014)

### 2.2.6 Rajapinnat

Aluksi GSM-verkon määrittelyihin piti saada useita avoimia rajapintoja, jotta operaattorit voisivat käyttää eri valmistajien tuotteita verkkojen rakentamiseen. Nykyään määrittelyissä on vain kaksi täysin avointa rajapintaa ja muutama rajapinta, jotka eivät ole avoimia. Kaksi avointa rajapintaa ovat ilma-rajapinta(Air), joka toimii MS:n ja BTS:n välillä ja toinen avoin rajapinta on A-rajapinta(A int.), joka toimii BSC:n ja MSC:n välillä. Ei-avoimista rajapinnoista esimerkkinä on Abis-rajapinta, joka on BTS:n ja BSC:n välillä. Koska Abis-rajapinta ei ole täysin avoin, eri valmistajien BTS:ää ja BSC:tä ei aina voi käyttää yhdessä. (SYSTRA 2000, 14)



Kuva 2. Tarkempi kuvaus GSM-järjestelmän rakenteesta

### 2.3 Taajuudet

GSM tekniikka käyttää eri taajuusalueita. Alkuperäisen määrittelyn 900 MHz:n taajuus on yksi näistä. Nykyään taajuuksia on määritelty lisää.

Taulukko 1. GSM:n taajuusalueet. (Penttinen, 2006, 136.)

GSM-versio	Taajuuskaista	Taajuudet/Uplink	Taajuudet/Downlink
P-GSM 900	25 MHz	890-915 MHz	935-960 MHz
E-GSM 900	10 MHz	880-890 MHz	925-935 MHz
R-GSM 900	4 MHz	876-880 MHz	921 – 925 MHz
GSM 1800	75 MHz	1710-1785 MHz	1805-1880 MHz
GSM 1900	PCS 1900-määrittelyiden mukainen	riippuu alueesta	riippuu alueesta

Oheisessa taulukossa on eri GSM tekniikoiden tukemia taajuusalueita. Lisäksi Amerikassa on käytössä myös 800 ja 850 MHz:n taajuusalueet. Käytännössä kaikki GSM-versiot perustuvat samoihin spesifikaatioihin ja erot tulevat esille pääasiassa tehotasoissa, kanavien lukumäärässä ja taajuusalueissa. GSM verkoille on määritelty nimet GSM 800, GSM 850, GSM 900, GSM 1800 ja GSM 1900 MoU:n (Memorandum of Understanding) toimesta.

Kanavaväli eli taajuus on 200 kHz ja taajuusalueiden alussa ja lopussa on yhden kanavan suojaetäisyys. Esimerkiksi P-GSM:n ensimmäinen taajuus on 890,2 MHz. Lähetys- ja vastaanottotaajuudet on jaettu TDMA-kehysten (Time Division Multiple Access) mukaisesti kahdeksaan aikaväliin. GSM-järjestelmässä on käytössä TDMA:n ja FDMA:n (Frequency Division Multiple Access) yhdistelmä, jotta voidaan toimia eri taajuuksilla.

### 3 DX 200 TUOTEPERHE

DX 200 on Nokian tuoteperhe. Sitä käytetään yleisnimityksenä verkkoelementeille, joilla muodostetaan tietoliikenneverkko. DX 200 –järjestelmien kehitys alkoi 1970-luvulla ja ensimmäiset toimitukset asiakkaille tapahtuivat vuonna 1980. Ensimmäinen verkkoelementti oli kiinteän verkon puhelinkeskus FSC (Fixed Network Switching Centre), mutta nykyään sen rinnalla on lukuisia eri verkkoelementtejä. Näistä verkkoelementeistä keskeisimmät ovat matkapuhelinkeskus MSC (Mobile Switching Centre), matkapuhelinverkon tilaajien kotirekisteri HLR (Home Location Register) ja matkapuhelinverkon tukiasemaohjain BSC (Base Station Controller).

Jokaisen tuoteperheen tuotteen perustana käytetään periaatteessa samanlaista vi-  
kasietoista DX 200 –laitteistoa, jonka päälle kootaan tietyn verkkoelementin tarvitta-  
va ohjelmisto. (DX-aapinen, 2)

### 3.1 DX 200 –tuoteperheen ominaisuuksia

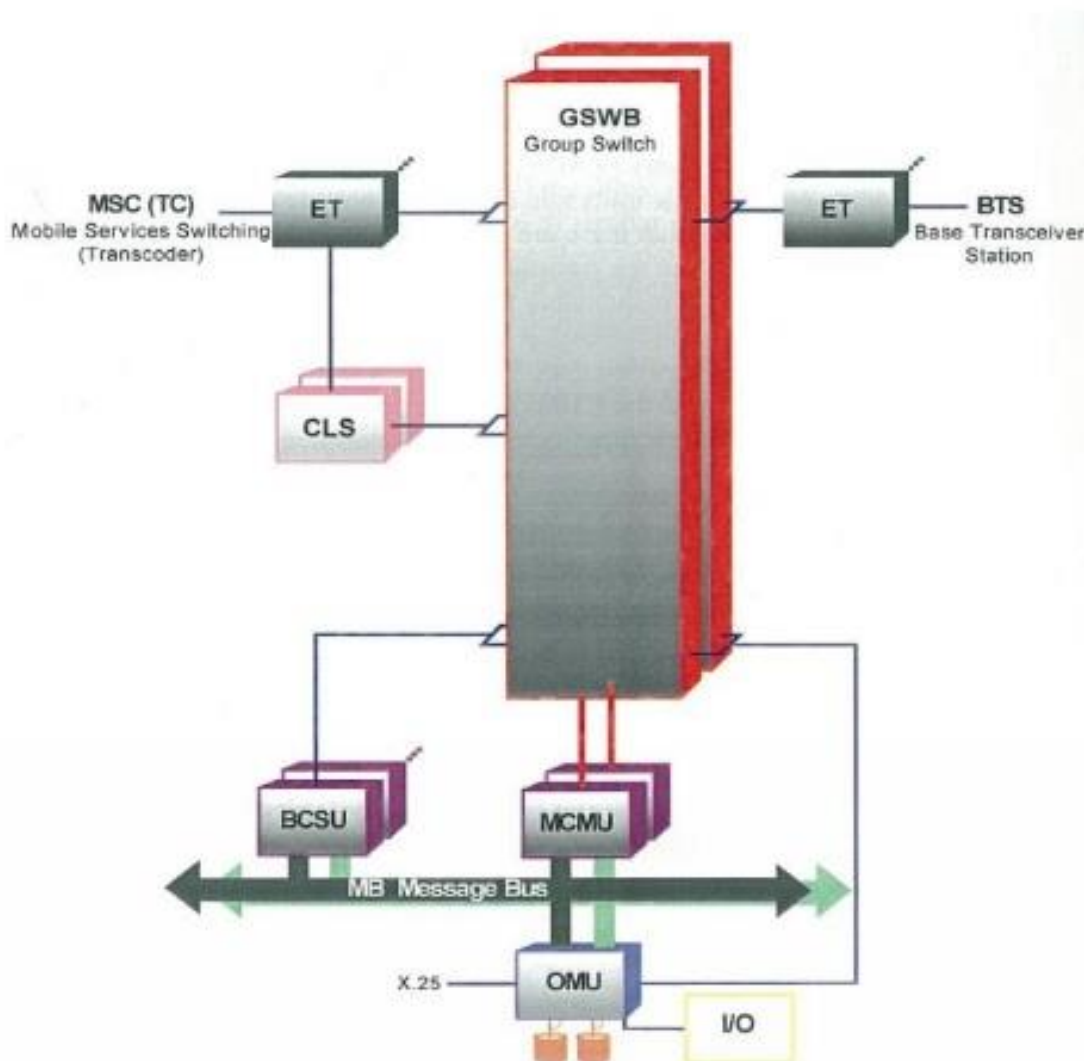
DX 200 –järjestelmä on hajautettu ja löyhästi kytketty moniprosessijärjestelmä. Sen arkkitehtuuriset pohjautuvat pitkälti siihen tarpeeseen, mihin keskus on tarkoitettu. Se on periaatteessa yleiskäyttöinen tietokonejärjestelmä, joka vasta ohjelmistolla saadaan palvelemaan tiettyä tarkoitusta. Lähemmin tarkasteltuna siitä voidaan kuitenkin erottaa laitteistokomponentteja, jotka ovat välttämättömiä puhelinkeskukselle.

Järjestelmä on yleisesti ymmärrettynä reaaliaikainen. Järjestelmässä reaaliaikaisuus toteutetaan kolmella perusmekanismilla. Ensimmäinen näistä on se, että se perustuu keskusmuistipohjaisuuteen. Tämä tarkoittaa sitä, että kaikki tarvittavat tiedot sijaitsevat tietokoneen omassa muistissa, eikä esimerkiksi hitaalla ulkoisella tietovälineellä. Toinen asia, joka mahdollistaa reaaliaikaisuuden, on rinnakkaisuus. DX 200:n järjestelmässä on niin laitteisto- kuin ohjelmistotasolla monistettuja komponentteja, jotka suorittavat samoja tehtäviä. Viimeinen reaaliaikaisuuden mahdollistava piirre on riittävän nopeat tiedonsiirtomenetelmät. DX 200 –järjestelmässä olevat prosessit kommunikoivat keskenään nopean sanomaväylän kautta.

DX 200 –järjestelmä on myös vikasietoinen. Järjestelmän vikasietoisuus perustuu pitkälti varmennukseen. Jos joku keskuksen yksikkö vikaantuu, tulee keskuksen pysyä jatkamaan toimintaansa. DX 200 –järjestelmässä varmennustapoja on kahden laisia. Ensimmäinen varmennustapa on 2N-varmennus. 2N-varmennuksella yksiköllä on tasan yksi varayksikkö. Eli kun toiminnassa oleva yksikkö vikaantuu, otetaan varayksikkö käyttöön. Varayksikkö toimii yleensä synkronisesti aktiivisen yksikön rinnalla ja kykenee näin ottamaan milloin tahansa aktiivisen yksikön tehtävät itselleen. Kun varayksikkö ottaa aktiivisen yksikön tehtävät vastaan, kutsutaan tätä tapahtumaa puolenvaihdoksi. Toisena varmennustapana on niin sanottu N+1 varmennus. N+1 varmennuksessa on yhteinen varayksikkö, joka voi ottaa aktiivisen yksikön tehtävät itselleen, tai sitten aktiivisen yksikön tehtävät voidaan ohjata jollekin toiselle aktiiviselle yksikölle, jolle tehtävät voidaan jakaa. (DX-aapinen, 48-79)



### 3.2 DX 200 tukiasemaohjain (BSC)



Kuva 6. DX 200 BSC arkkitehtuuri (SYSTRA 2000, 184).

DX 200 tukiasemaohjain on rakennettu samalla lailla kuin kaikki muutkin DX 200 tuotteet. Tukiasemaohjain koostuu useasta tietokoneyksiköstä, joilla on kaikilla on omat tehtävänsä ja tietokoneyksiköstä.

Tukiasemaohjain koostuu seuraavista yksiköistä:

- KytKentä Kenttä GSWB  
KytKentä Kentän tehtävänä on kytkeä puhelut ja datayhteydet

- Keskuspääte ET  
ET:n tehtävä sovittaa PCM-yhteydet sopiviksi MSC:n ja BTS:n välillä.
- Kello- ja synkronointiyksikkö CLS  
CLS tehtävä on luoda synkronointisignaaleja BSC:n eri yksiköille ja näin tahdistaa niitä.
- Signaalointiyksikkö BSCU  
BCSU:n tehtävänä on hoitaa SS7 signallointi MSC:n ja BSC:n välillä sekä hoitaa signallointi BSC:n ja BTS:n välillä.
- Kytkentämatriisi- ja radioressienhallintayksikkö MCMU  
MCMU:n tehtävä on hallita ja valvoa GSWB:tä ja samalla se vastaa soluista ja radiokanavista.
- Käyttö ja ylläpitoyksikkö OMU  
OMU toimii käyttäjän päätteenä tukiasemaohjaimen. OMU:n avulla käyttäjä voi suorittaa tarvittavat huoltotoimenpiteet ja operointitehtävät.
- Sanomaväylä MB  
Sanomaväylän tehtävä on siirtää sanomia eri yksiköiden välillä BSC:n sisäisesti.
- Pakettiohjainyksikkö PCU  
PCU on lisäelementti, jolla voidaan kytkeä BSC GPRS-runkoverkkoon.

(SYSTRA 2000, 182-185)

## 4 OHJELMISTOKEHITYS JA TESTAUS

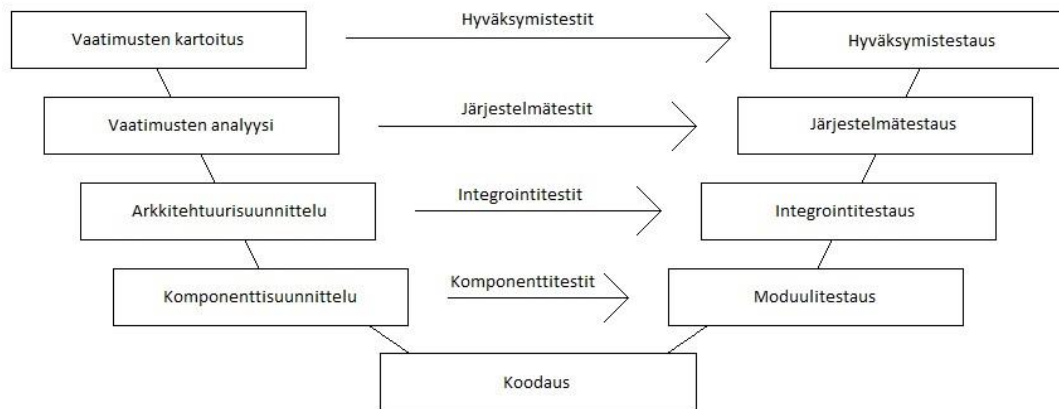
### 4.1 Ohjelmistokehitys

Ohjelmistokehitystä voidaan toteuttaa monilla eri toimintatavoilla. Yksi näistä tavoista on niin kutsuttu V-malli.

V-malli on prosessimalli, jota voidaan käyttää projektien suunnittelussa ja toteutuksessa. V-mallin avulla projektia on helpompi hallita ja sen avulla saadaan hyvin määriteltyä projektin päämäärät, tehtävät ja aikataulun. V-malli on vesiputousmallia kehittyneempi lineaarinen malli, joka määrittelee puhtaan vaihejaon muiden prosessien tapaan. V-mallissa varsinainen työ jaetaan kahteen eri osaan. Ensimmäiseen osaan kuuluu suunnittelu ja toteutus ja jälkimmäiseen osaan kuuluu testaus. V-mallin mukaan jokaista suunnittelu- ja toteutusvaihetta vastaa testausvaihe. Ideana on se, että samalla kun suunnitellaan jokin tietty kokonaisuus, suunnitellaan samalla myös miten se tullaan myöhemmin testaamaan. Koska jokaisessa suunnittelu- ja toteutusvaiheessa suunnitellaan myös testaus, pysyy testaus ajan tasalla toteutuksen kanssa. V-mallin vaihejako vastaa käytännössä perinteistä vesiputouksen jakoa.

V-mallin vaihejako:

- Vaatimusten kartoitus
- Vaatimusten analyysi
- Arkkitehtuurisuunnittelu
- Komponenttisuunnittelu
- Koodaus
- Moduulitestaus
- Integroititestaus
- Järjestelmättestaus
- Hyväksymistestaus



Kuva 3. V-mallin kaaviokuva

Samoin kuten vesiputousmalli, perustuu V-malli myös dokumentointiin. Puhtaimmillaan V-mallissa dokumentit tarkastellaan jokaisen vaiheen jälkeen. Näin dokumentit pysyvät ajan tasalla. Hyvä dokumentaatio ja etenkin testausdokumentaatio mahdollistaa myös toimivan regressiotestauksen eli vanhojen testien suorittamisen uudelleen.

V-mallilla on monia etuja. V-malli on helppo omaksua ja siihen löytyy laaja teoria- ja työkalutuki. Tämän takia malli ei vaadi käyttäjiltä mitään erityistaitoja. V-mallia voidaan helposti muokata erilaisia projekteja ja organisaatioita varten ja sen avulla voidaan helposti määritellä mitä tehdään missäkin vaiheessa projektia ja mitä kukin tekee. V-mallissa testausta suunnitellaan koko kehitystyön ajan, joten jos projektilla tulee kiire aikataulun tai budjetin ylittyessä, ei tuote jää täysin testaamatta.

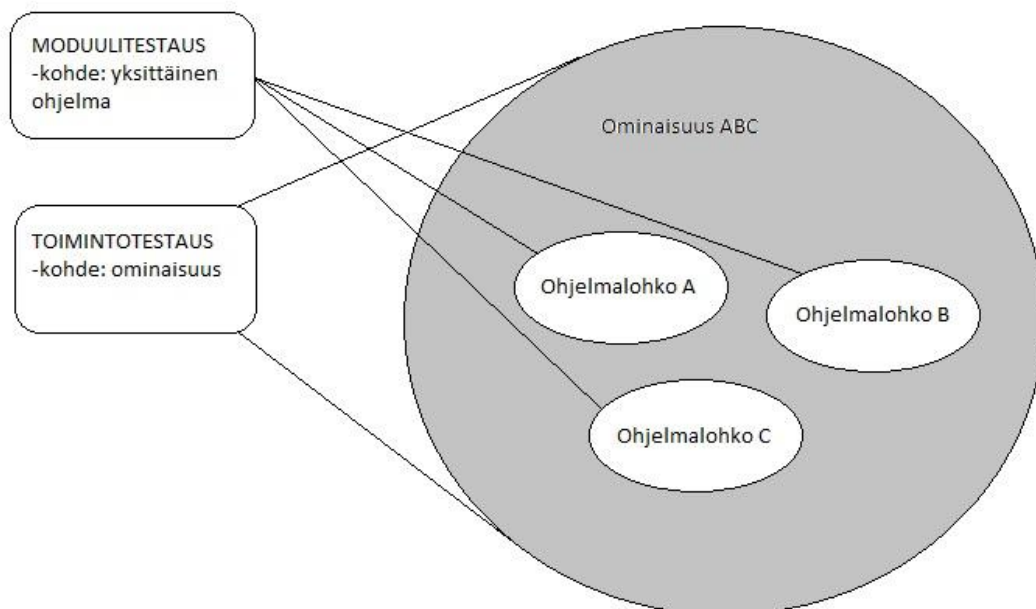
Koska V-malli on melko lähellä vesiputousmallia, kärsii se samoista ongelmista. Projektin aikana työn vaatimukset eivät saa muuttua. Jos tätä ei saada toteutettua, ei V-mallia kannata myöskään käyttää. Vaatimuksien muuttuminen aiheuttaa lisätyötä ja jokainen vaihe täytyy tarkastaa uudestaan muuttuneilta osin. V-mallista johtuen asiakas näkee valmista vasta, kun projekti on päättymässä. Jos projekti lähtee väärään suuntaan, on sen kurssia vaikea muuttaa ja aikataulusta jäljessä olevaa projektia on vaikea saada takaisin aikatauluun. Lisäksi projektit, joissa käytetään V-mallia, ovat yleensä kalliita kattavan dokumentoinnin ansioista. Dokumentointiin ja sen katselmointiin kuluu aikaa ja näin ollen tuottaa lisää työtunteja. (Helsingin yliopiston www-sivut 2014)

## 4.2 Ohjelmistotestaus

Ohjelman testaaminen on suunnitelmallista virheiden etsimistä ohjelmaa tai sen osaa suorittamalla. Virheellä tarkoitetaan ohjelman toimimista vastoin sen määrittelyissä määriteltyä toimimista.

Dokumenttien katselmuksia voidaan käytännössä sanoa testauksen aikaisimmaksi muodoksi. Tuotetun koodin ja dokumenttien katselmuksilla voidaan löytää monia vikoja jo ennen varsinaista testausta. Testaus jatkaa siitä mihin katselmuksella, etenkin koodin katselmus, päättyi. Testauksella on tarkoitus löytää mahdollisia vikoja tuotteesta, jotta ne voitaisiin korjata ja näin ollen parantaa tuotteen laatua.

Testaus jaetaan kolmeen osaan. Ensimmäisenä tulee moduulitestaus, tämän jälkeen aloitetaan toimintotestaus (integraatiotestaus) ja lopuksi järjestelmätestaus. Ohjelmistosuunnittelijan vastuulla on yleensä suorittaa moduulitestaus tuottamalleen koodille/omalle ohjelmalohkolleen. Toimintotestauksen ja järjestelmätestauksen suorittaa erillinen testaukseen keskittynyt henkilö tai organisaation osa. (DX aapinen 37.)



Kuva 4. Moduulitestaus ja toimintotestaus

#### 4.2.1 Moduulitestausta

Moduulitestausta tarkoitetaan ohjelmalohkon sisäistä testaamista. Moduulitestausta testataan ohjelmalohkon toimintaa ja ohjelman toimintaa sen spesifikaatiota vasten. Testauksella yritetään löytää mahdolliset virheet toteutuksesta ja varmistaa ohjelman oikea toiminta. Moduulitestausta voidaan suorittaa niin sanottuna lasilaatikkotestausta. Lasilaatikkotestausta suunnittelija valitsee ja laatii testit tuntien ohjelman toiminnan. Näin ollen suunnittelija voi testata ohjelman toimintaa täsmällisesti vain niiltä osin kuin se on tarpeellista. Moduulitestausta voidaan suorittaa pöytätestauksena paperilla tai laboratoriossa sijaitsevalla testikeskuksella. Yleensä moduulitestausta suoritetaan laboratoriossa sijaitsevalla testikeskuksella etäyhteyden avulla. (DX Aapinen, 38.)

#### 4.2.2 Toimintotestausta

Toimintotestausta testataan jotakin ohjelman toimintaa tarkemmin. Yleensä jonkin tietyn ohjelman ominaisuuden toimiminen vaatii useiden ohjelmien yhteistoimintaa, näin ollen toimintotestausta testataan ohjelmien yhteistoimintaa. (DX Aapinen, 38.)

#### 4.2.3 Järjestelmätestausta

Järjestelmätestauksessa testataan koko järjestelmän ohjelmiston, laitteiston ja dokumentaation yhteensopivuus ja toiminnallisuus. Järjestelmätestauksessa testataan erikseen erilaisilla kokoonpanoilla. Kun edelliset testaukset keskittyvät toiminnalliseen testaukseen, keskitytään järjestelmätestauksessa myös muihin kuin toiminnalliseen puoleen.

Järjestelmätestauksessa toiminnallisessa puolessa testataan muun muassa järjestelmän asentaminen, käynnistäminen, alustus ja tiedostojen ja tietokantojen konvertointi.

Ei-toiminnalliseen testaukseen sisältyy muuan muassa suorituskyvyn testaaminen jossa järjestelmän toimintaa testataan järjestelmän maksimi ja normaali kuormitustilanteissa. Ei-toiminnalliseen testaukseen sisältyy myös erilaiset stressitestit ja voilyymitestit. Myös dokumentaation käyttökelpoisuutta voidaan testata näissä testeissä. (DX Aapinen, 38-39.)

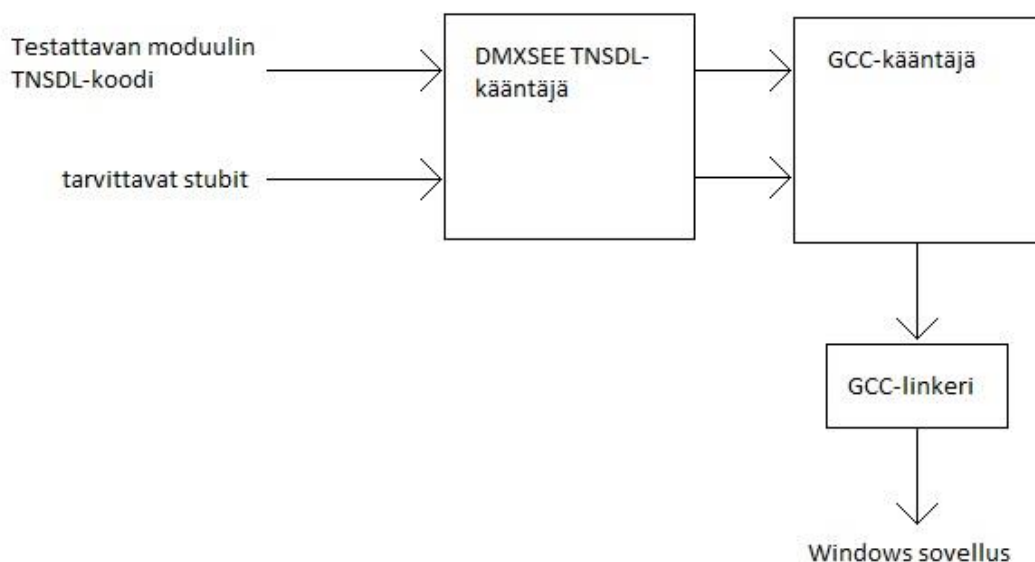
## 5 WDOX

WDOX on yksinkertainen tapa simuloida ohjelmalohkon toimintaa omalla koneella. Sen kanssa ei tarvita erillistä hardwarea, vaan WDOX pyörii lähes millä tahansa normaalilla tietokoneella. WDOX:lla ohjelmalohkosta muodostetaan yksi erillinen sovellus, jolla voidaan testata ohjelmalohkon toimintaa.

WDOX:a voidaan käyttää normaalisti erillisen testausohjelman kanssa. Testausohjelmassa ei tarvitse tehdä muuta kuin asettaa oikea IP-osoite ja muuttaa yhteystyypiksi RDOX. Tämän jälkeen testausohjelmaa voidaan käyttää testaamiseen samalla tavalla kuin normaalin testiympäristön kanssa testattaessa.

Kun Windows sovellusta luodaan, tarvitaan kolme asiaa. Sama kääntöympäristö, jota käytetään normaalin imagen kääntämisessä, GCC kääntäjä ja linkkeri ja viimeisenä tarvittavat ”stubit”, joilla simuloidaan tarvittavia kirjastokutsuja joita WDOX ei vielä itsestään tue. GCC kääntäjä ja linkkerinä voidaan käyttää esimerkiksi TDM-GCC 4.6.1, joka on vapaasti saatavilla verkossa.

Ensimmäiseksi Windows sovellusta luodessa tehdään normaali TNSDL kääntö. Tämän jälkeen C-kielen kääntö GCC-kääntäjällä. Lopuksi oma koodi linkataan sovellukseen GCC-linkerin avulla.



Kuva 5. Windows sovelluksen luonnin kaaviokuva

### 5.1 WDOX:n käyttöönotto

WDOX:n käyttö on sen käyttöönoton jälkeen helppoa. Testattavan Windows-sovelluksen luominen voidaan lisätä käytössä olevaan kääntötiedostoon ja sovellus voidaan halutessa kääntää säätämällä kääntötiedoston alussa olevan muuttujan alustusta "no" arvosta "yes" arvoon.

Tällä hetkellä ohjelmalohko käännetään kääntötiedostolla, jossa määritellään muun muassa millä kääntäjäversiolla koodia käännetään. WDOX käyttö vaatii hieman kääntötiedoston muokkaamista. Kääntötiedostoon kannattaakin luoda muuttuja nimeltä "WDOX" ja alustaa se arvoon "no". Kääntötiedostoon lisätään if-lauseke IF (WDOX="yes") ennen niitä kohtia, jotka vaativat muutoksia WDOX sovellusta käytettäessä ja tiedoston sama osa muuttamattomana else-haaraan. Näin kääntötiedoston alussa "WDOX" arvoa alustettaessa voidaan määritellä käytetäänkö WDOX:a vai ei.



Esimerkki if-rakenteesta, jota kääntötiedostossa voidaan käyttää WDOX:a lisätessä kääntötiedostoon:

```
if (WDOX="yes")
```

```
tähän WDOX-käännön vaatima versio
```

```
else
```

```
tähän kääntötiedoston alkuperäinen versio
```

```
end if
```

WDOX:n käyttöönotossa voi vastaan tulla monenlaisia ongelmia. Yksi näistä on kääntäjäversion vaihtuminen. Tällaiset tilanteet pitää ennen WDOX:n käyttöönottoa tutkia ja selvittää, jotta koodi ja sen testaus olisi edelleen johdonmukaista ja varmaa.

WDOX vaatii omien tiedostojen lataamista kääntökansioon. Nämä tiedostot voidaan sijoittaa vaikka verkkolevyille, jonka jälkeen voidaan kääntötiedoston oikeaan kohtaan lisätä niiden hakeminen verkkolevyiltä kääntökansioon. Jos WDOX:n tiedostoja ei haluta verkkolevyille, voidaan ne sijoittaa myös paikallisesti koneelle. Tällöin tiedostojen täytyy sijaita jokaisella työasemalla saman tiedostopolun takana, muuten WDOX kääntöä tehdessä täytyy oikea polku käydä aina asettamassa, jotta tiedostot saadaan kopioitua kääntökansioon.

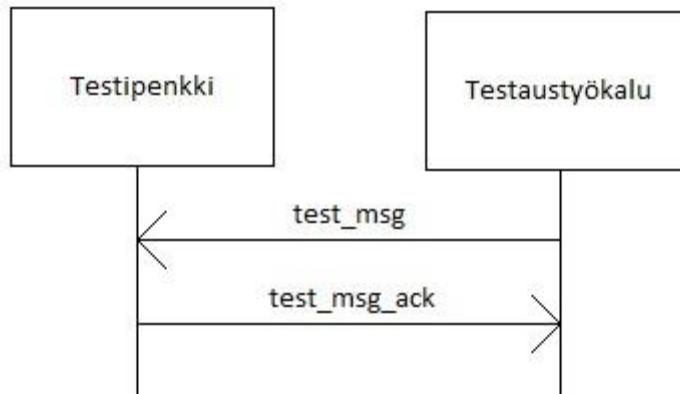
WDOX:n tiedostoista löytyy WDOX.xml tiedosto, jossa määritellään WDOX:n asetukset. Tähän asetetaan arvot sen mukaan, minkä alustan ohjelmalohkoa testataan. Tiedostossa määritellään myös kerätyn tietokonekokonaisuuden pituus ja se, näytetäänkö ajastimia vai ei. Tähän asetetaan myös testattavasta ohjelmalohkosta tiedot. Tiedostossa määritellään myös tiedot rajapinnasta. Jos halutaan ajaa useampia WDOX sovelluksia samalla koneella, pitää rajapinnan tietoja aina muuttaa. Nämä tiedot voidaan keran laittaa kohdalleen, jonka jälkeen tiedosto voidaan tallettaa sellaiseen paikkaan, josta juuri tätä ohjelmalohkoa testatessa saadaan valmis versio eikä itse testaajan tarvitse käyttää aikaa määrittelyihin.

## 6 OHJELMALOHKON TESTAUS

Ohjelmalohkosta käännetään image-tiedosto, joka ladataan testipenkkiin ja sen jälkeen testipenkki käynnistetään uudelleen. Käynnistyksen jälkeen yleensä varmistetaan, että uuden imagen lataus onnistui ja käytössä tosiaan on nyt uusi muokattu versio ohjelmalohkosta. Tämän jälkeen testit voi aloittaa. Testit suoritetaan erillisellä testausohjelmalla. WDOX käyttö sallisi ohjelmalohkon testaaminen ilman testipenkiä. Ohjelmalohkosta käännetään WDOX-sovellus, jota voidaan käyttää testipenkin tapaan testausohjelman kanssa.

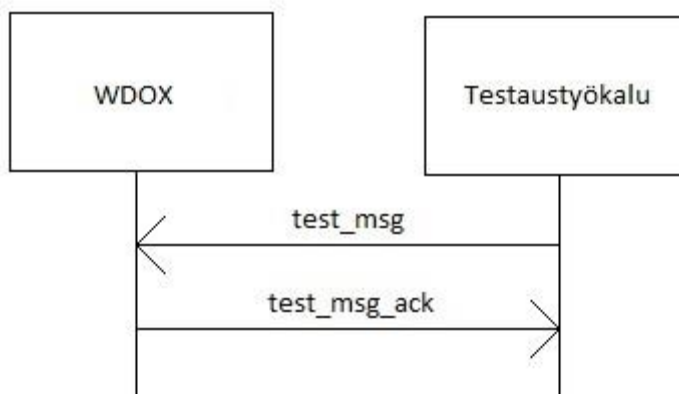
Ohjelmalohkoa testataan niin sanotuilla sanomasekvensseillä. Eli ohjelmalohkolle lähetetään työkalun avulla testisanomia ja ohjelmalohkon toimintaa tarkastellaan sen lähettämällä vastaussanomilla. Jokaisessa sanomassa on kenttiä, joiden arvot määrittelemällä oikein saadaan ohjelmalohko vastaamaan halutulla tavalla. Työkalulla voidaan asettaa haluttu vastaussanoma, jonka poiketessa vastaanotetusta sanomasta työkalu ilmoittaa virheen. Jos vastaussanomaa ei vastaanoteta ollenkaan, jää työkalu usein odottamaan tätä sanomaa ja testin suorittaminen jumiutuu. Tällöin kyseessä on yleisimmin myös virhe.

Komponenttisuunnitteluvaiheessa on ennalta suunniteltu erilaiset skenaariot, jotka testaamalla saadaan toteutus (koodi) testattua mahdollisimman kattavasti. Moduulitestausvaiheessa nämä skenaariot mallinnetaan käytössä olevalla työkalulla testeiksi, jotka ajetaan läpi. Läpi ajaminen tarkoittaa, että työkalu lähettää testissä määritellyt sanomat ja komennot tukiasemaohjaimelle siinä järjestyksessä kuin ne on testissä määriteltä. Työkalu seuraa testin kulkua ja ilmoittaa virheellä, jos vastaussanomassa on eri arvot kuin mitä testaaja on ennalta määrittänyt testiin. Tällöin ohjelmalohkon toiminnassa saattaa olla virhe tai testin suunnittelija ei ole ottanut jotakin yksityiskohtaa huomioon. Tukiasemaohjain ei myöskään välttämättä vastaa ollenkaan johonkin sanomaan tai vastaa liian hitaasti, jolloin jokin ajastin saattaa laueta. Näissäkin tapauksissa kyseessä on virhe tai testin suunnittelija ei ole ottanut jotakin yksityiskohtaa huomioon. Yhdellä testillä voidaan käydä useampia skenaarioita läpi. Työkalulla voidaan ottaa myös tukiasemaohjaimesta tulosteita. Tutkinnan alla olevaa ohjelmalohkoa testatessa yleensä otetaan tulostuksena MCMU:sta loki, jonka avulla nähdään missä kohtaa koodia mahdollinen virhe on tapahtunut.



Kuva 6. Esimerkki ohjelmalohkon testauksen sanomasekvenssistä.

WDOX-sovelluksen kanssa testatessa testit suoritetaan samaan tapaan kuin testipenkin kanssa. Testausohjelmassa mikään muu ei muutu, kuin että yhteystyyppinä on telnet –yhteys TCP/IP –yhteyden sijaan ja IP –osoitteeksi asetetaan 127.0.0.1. Tämän jälkeen testaustyökalu saa yhteyden WDOX –sovellukseen. Testien ajossa eroa ei huoma.



kuva 7. Testauksen sanomasekvenssi WDOX:n kanssa.

## 7 MITEN WDOX SOVELTUU OHJELMALOHKON TESTAAMISEEN

WDOX:lla käytännössä simuloidaan testiympäristöä. Siihen yhdistetään testiohjelmalla samalla lailla kuin testiympäristöön. Sanomat lähetetään samaan tapaan kuin testiympäristöön. WDOX lähettää myös vastaussanomat samalla lailla kuin testiympäristö. Vastaussanomaa voidaan verrata testiohjelmalla samalla lailla kuin normaalisti testausympäristöltä saatua sanomaa. Tämä helpottaa paljon testien tekemistä ja vanhoja testejä voidaan käyttää WDOX:lla testatessa.

Testausohjelman WDOX:n yhteistoimintaa testattiin lyhyellä sanomasekvenssillä. Tässä sekvenssissä lähetettiin ennalta oikeaksi todettu testisanoma WDOX:lle ja vastaus sanomaa verrattiin testityökalulla ennalta määrättyyn sanomaan. Testiohjelma lähetti normaalisti sanoman WDOX:lle ja WDOX vastasi sanomaan odotetusti. Mitään ongelmia ei ilmennyt. Sanomat lähtivät myös hyvin nopeaan, viitaten siihen että WDOX:lla testaaminen olisi nopeampaa kuin normaalilla testiympäristöllä. Nopeamman testaamisen toteamiseen tarvitaan kuitenkin enemmän pidempiä testejä ja useampia testien ajokertoja.

Yksi asia, mitä WDOX:lla ei kuitenkaan voi testata, on puolenvaihdot. WDOX:lla simuloidaan vain käynnissä olevaa järjestelmää ja puolenvaihtoa ei voi WDOX:lla simuloida. Näin ollen testitapaukset, joissa vaaditaan puolenvaihto, ei voida WDOX:lla testata vaan niiden testaukseen vaaditaan vielä normaalia testiympäristöä.

## 8 LOPPUTULOS JA SEURAAVAT VAIHEET WDOX:IN KÄYTTÖÖNOTOSSA

### 8.1 Lopputulos

Ennen WDOX:n käyttöönottoa pitää selvittää sen käyttöönotossa ilmenevät ongelmatilanteet. Käyttöönotossa ilmenevät virheet tulee tutkia ja selvittää huolellisesti, jotta koodin ja sen testauksen varmuus ja johdonmukaisuus voidaan säilyttää.

Testityökalun ja WDOX:n yhteistoiminta toimi hyvin. Sanomien lähetys ja vastaanotto testattiin testillä, johon asetettiin ennalta oikeiksi todettu sanoma ja siihen vastausanoma. WDOX-sovellusta testattaessa sanomat lähetettiin ja vastaanotettiin nopeaan. Testityökalussa ei tarvinnut tehdä suuria muutoksia. Testiympäristön valinnassa ei tarvinnut kuin asettaa testausympäristöksi RDOX, yhteystyypiksi telnet ja IP-osoitteeksi 127.0.0.1 ja porteiksi WDOX.xml tiedostossa määritellyt arvot. Tämän jälkeen testiohjelmiston ja WDOX-sovelluksen välillä toimi moitteettomasti. WDOX-sovelluksen kanssa voi testata myös etäyhteyden avulla käynnistämällä WDOX-sovelluksen erillisessä koneessa. Tällöin IP-osoitteena käytetään sen koneen IP-osoitetta, jossa WDOX:a käytetään.

Yhtenä testaamisen ongelmana tähän mennessä on ollut hitaat yhteydet. WDOX mahdollistaa moduulitestaamisen ilman näitä yhteyksiä. Hitaiden yhteyksien ansiosta testaaminen nopeutuu ja yhteyksien aiheuttamia ongelmia ei enää ole.

WDOX:n käyttö on helppoa. Sen jälkeen kun kääntötiedostoon on lisätty WDOX:n vaatimat muutokset on itse sen käyttö helppoa. Käytännössä kääntötiedoston alussa esitellään muuttuja, jonka arvoa vaihtamalla voidaan päättää halutaanko kääntää normaalin testipenkkiin ladattavan imagen sijaan WDOX-sovellus. Muuttuja voi olla esimerkiksi nimeltä ”WDOX” ja se olisi normaalisti arvossa ”no”. Kun muuttujan arvoksi muutetaan ”yes”, käännetään WDOX-sovellus. WDOX:n tarvittavat tiedostot voidaan tallettaa verkkolevyille ja kääntötiedosto voi hoitaa näiden kopioimisen kääntöhakemistoon.

WDOX:lla ei voi testata puolenvaihtoa. Jos testitapauksessa tarvitaan puolenvaihto, tarvitaan testaukseen edelleen testipenkkiä.

WDOX käyttäminen moduulitestauksessa voi nopeuttaa testien kulkua. Tutkinnassa oli havaittavissa testien nopeampi ajo verrattuna testipenkissä testaamiseen jo pelkää sen takia, ettei WDOX:a tarvitse käynnistellä uudestaan ennen testauksen aloittamista. WDOX:lla testaaminen mahdollistaa myös ”offline”-tilassa testaamisen, joka on hyvä lisä moduulitestaukseen. WDOX ei kuitenkaan yksin korvaa testipenkkiä

moduulitestauksessa jo sen takia, ettei sillä voi simuloida puolenvaihtoa. WDOX olisi kuitenkin hyvä lisätyökalu moduulitestaukseen.

## 8.2 Seuraavat vaiheet käyttöönotossa

Ennen kuin WDOX voidaan ottaa kunnolla käyttöön, pitää vielä seuraavia vaiheita suorittaa.

Koodi ei kääntynyt uudella kääntäjäversiolla, jota WDOX käyttää. Koodin ongelmalliset osiot laitettiin tutkinnan ajaksi kommentteihin ja niihin ei sen enempää puututtu. Ennen kuin WDOX otetaan käyttöön, pitää ongelmat koodin käännössä selvittää.

Kääntötiedosto pitää muokata kunnolliseksi. Tutkinnassa käytössä oli tilapäinen versio kääntötiedostosta. WDOX pitää lisätä kääntötiedostoon niin, että se on helppo ottaa käyttöön ja ottaa pois käytöstä. Kääntötiedosto pitää myös testata ennen käyttöönottoa.

Ohjelmalohkoille pitää tehdä omat spesifiset WDOX.xml tiedostot. WDOX.xml tiedostossa määritellään ohjelmalohkoon liittyviä asetuksia ja ohjelmalohkon tietoja. Näin ollen jokaiselle ohjelmalohkolle olisi hyvä luoda omat tiedostot, jotka säilötään tiettyyn paikkaan. Jokaisen ohjelmalohkon oma kääntötiedosto noutaisi oman WDOX.xml tiedostonsa. Säilö voi olla esimerkiksi verkkolevyllä.

WDOX käyttöönotto vaatii myös hyvät ohjeet. Tähän asti testaus on suoritettu testipenkillä ja testausohjelmalla. Kokemusta WDOX:sta ei ole, joten hyvät tarkat käyttöohjeet olisi hyvä laatia ennen kuin WDOX on käytössä. Tämä nopeuttaisi käyttöönottoa ja auttaisi käyttäjiä WDOX:n käytössä. Ennen käyttöönottoa kannattaa myös järjestää koulutus liittyen WDOX:n käyttöön.

Kaikki vanhat testit on hyvä vielä ajaa WDOX:n kanssa ja normaalin testiympäristön kanssa, jotta saadaan varmistettua että testit toimivat kummassakin samalla tavalla.

Viimeinen vaihe on tutustua ja ottaa käyttöön WDOXMT työkalu. WDOXMT:n avulla saadaan testi suoritettua suoraan WDOX-sovelluksella. Käännön yhteydessä WDOX-sovellukseen lisätään jo testit, jotka halutaan suorittaa, ja ne suoritetaan automaattisesti WDOX-sovellusta käynnistettäessä. Tässä työssä ei WDOXMT:tä tutkittu, joten jos se nähdään käytännölliseksi, voidaan sekin ottaa käyttöön. WDOX:a voidaan tosin käyttää hyvin ilmankin WDOXMT:tä.

## LÄHTEET

Helsingin yliopiston www-sivut. 2014. Viitattu 13.11.2014.

[http://www.cs.helsinki.fi/u/taina/opol/k-2009/pdf/luku-6\\_2.pdf](http://www.cs.helsinki.fi/u/taina/opol/k-2009/pdf/luku-6_2.pdf)

Nokia Networks Oy. 2000, SYSTRA GSM System Training.

Penttinen, J. 2006. Tietoliikenneverkot – Perusverkot ja GSM. Sanoma Pro Oy.

Silander, S. 1999. DX-AAPINEN Johdatus DX 200 ohjelmistotyöhön. Helsinki: Nokia Telecommunications Oy.

Svein Willassenin www-sivut. 2014. Viitattu 13.11.2014

<http://www.willassen.no/msl/node4.html>

Tietoliikenteen ja tietotekniikan keskusliiton www-sivut. 2014. Viitattu 13.11.2014

[http://www.ficom.fi/tietoa/tietoa\\_4\\_1.html?Id=1034921940.html](http://www.ficom.fi/tietoa/tietoa_4_1.html?Id=1034921940.html)