

Sami Väisänen

**Verkkoyhteyden hallintaohjelmisto**

Insinööri  
Kajaanin ammattikorkeakoulu  
Tekniikan ala  
Tietotekniikka  
Kevät 2013



Koulutusala Tekniikan ja liikenteen ala	Koulutusohjelma Tietotekniikka
Tekijä(t) Sami Väisänen	
Työn nimi Verkkoyhteyden hallintaohjelmisto	
Vaihtoehtoiset ammattiopinnot	Ohjaaja(t) Arto Partanen
	Toimeksiantaja Ari Särkiniemi Sunit Oy
Aika Kevät 2013	Sivumäärä ja liitteet 27
<p>Insinööriyön tavoitteena oli suunnitella ja toteuttaa ohjelmistosovellus, joka lukee sarjaportin välityksellä modeemien laitetietoja, säätää modeemiin yhteysasetuksia, muodostaa Internet-yhteyden, sekä valvoo yhteyden tilaa pitämällä siitä lokia. Verkkoyhteyden hallintaohjelman on siis tarkoitus sisältää samanlaisia ominaisuuksia, joita esimerkiksi nettitikuissa käytetään.</p> <p>Toimeksiantona saadun ohjelman oli oltava mahdollisimman selkeä sekä käytettävyydeltään että dokumentoinniltaan. Toteutus tuli tehdä rakenteeltaan modulaarisesti, jotta uusien ominaisuuksien, kuten esimerkiksi uusimpien verkkotekniikoiden lisääminen myöhempisiin versioihin olisi mahdollisimman vaivatonta. Myös yksi tutkimusongelma oli ennakoita, kuinka ohjelmisto käyttäytyy poikkeavissa yhteystilanteissa, kuten satunnaisten katkeamisten aikana.</p> <p>Ohjelman taustalla suoritettavat rajapintojen kommunikoinnit, laitteiden tunnistamiset ja ohjauskomentojen parsiminen kirjoitettiin C/C++ kielellä omaksi dll-tiedostoksi. Dll:n lukuapuna käytettiin erillistä xml-tiedostoa, joka sisältää ohjaukseen perustuvia modeemitietoja.</p> <p>Ohjelman ulospäin näkyvä käyttöliittymä kirjoitettiin C#:lla. Käyttöliittymässä ohjataan dll-tiedostoa, ja sillä ilmaistaan käytettävä modeemi ja puhelinoperaattori, sekä tietoa Internet-yhteyden tilasta ja signaalinvoimakkuudesta. Ohjelma kerää myös taustalla lokitietoa. Ohjelma tallentaa tarpeellisia asetuksia erilliseen kokoonpanotiedostoon, josta ne myös ladataan seuraavalla kerralla käynnistäessä.</p> <p>Lopputuloksena toteutettiin verkkoyhteyden hallintaohjelmiston yksinkertainen runko. Ohjelmaa on mahdollista laajentaa useilla jatkokehitysmahdollisuuksilla.</p>	
Kieli	Suomi
Asiasanat	Internet-yhteys, modeemi, ohjelmisto, sarjaliikenne
Säilytyspaikka	<input checked="" type="checkbox"/> Verkkokirjasto Theseus <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto



School Kajaani University Of Applied Sciences	Degree Programme Information Technology
Author(s) Sami Väisänen	
Title Connection Manager Software Project	
Optional Professional Studies	Instructor(s) Arto Partanen
	Commissioned by Ari Särkiniemi Sunit Oy
Date Spring 2013	Total Number of Pages and Appendices 27
<p>The connection manager is a software application that is intended to create an Internet connection, to maintain the connection on and to keep the statistic log about connection quality. The program is divided into two parts: the user interface and the dll-library.</p> <p>The modular code program needs to be as user friendly and simple to enlarge in the future as possible. Also, one of the research problems was to predict how the program will behave during abnormal contact situations, such as random disconnections.</p> <p>The main program part communicates with modems via the RS232 serial port. The handling of different interfaces, identifications and parsing modem commands was implemented in the C / C + + language as a separate dll file. Dll also takes advantage of a separate xml file that contains the control information based on the modems.</p> <p>The user interface idea is to control the dll-file, and it displays out the modem information and the telephone operator, as well as information about the Internet connection status and about signal strength. The manager also collects background information into the log file. The program stores the necessary configuration settings into a separate inifile, and loads the settings from it on startup. The user interface part was written in C #.</p> <p>A simple connection manager software frame was the end result of the work. There are still possibilities to enlarge the project.</p>	
Language of Thesis	Finnish, English
Keywords	Internet-connection, modem, serial communication, software
Deposited at	<input checked="" type="checkbox"/> Electronic library Theseus <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

## ALKUSANAT

Tämä insinöörityö on tehty opinnäytetyönä Kajaanin ammattikorkeakoulun tietotekniikan koulutusohjelman neljäntenä vuonna. Työn toimeksiantaja on kajaanilaisen IT-yritys Sunit Oy:n tuotekehitysyksikkö. Insinöörityön tavoitteena on ollut suunnitella ja toteuttaa verkko-yhteyden hallinnan ohjelmiston pohja, jota on mahdollisimman luontevaa lähteä kehittämään laajemmaksi.

Haluan kiittää Sunit Oy:n tuotekehitysyksikköä insinöörityön toimeksiannosta, kaikkia työn parissa opastaneita sekä ongelmatilanteissa eteenpäin auttaneita. Erityiskiitokset haluan välittää Sunit Oy:n yhteyshenkilöille Ari Särkiniemelle ja Jussi Rytilahdelle sekä ammattikorkeakoulun valvojalle Arto Partaselle.

Kajaanissa 12.4.2013

Sami Väisänen

## SISÄLLYS

1 JOHDANTO	1
2 VERKKOHALLINTAOHJELMISTON TEORIA	2
2.1 Internet-yhteyden muodostamisen edellytykset	2
2.1.1 Modeemit	4
2.1.2 AT-komennot	5
2.1.3 RS232-sarjaliikenne	7
2.2 Ohjelmallisen toteutuksen vaatimukset	9
2.2.1 Modulaarinen ohjelmarakenne	9
2.2.2 DLL-tiedosto ohjelman suorituksen taustalla	10
2.2.3 XML-tiedostojen hyödyntäminen tietokantana	11
3 TYÖVAIHEET	12
3.1 Suunnittelu	12
3.1.1 Perehtyminen ja tiedonkeruuvaihe	12
3.1.2 Vaatimusmäärittelyn laatiminen	13
3.1.3 Ohjelmarakenteiden kuvaaminen lohko- ja vuokaavioilla	14
3.2 Toteutus	15
3.2.1 DLL-tiedoston ohjelmallinen toteutus	15
3.2.2 Käyttöliittymän ohjelmallinen toteutus	18
3.3 Testaus	23
3.3.1 Laitteistojen tunnistus ja toiminta	23
3.3.2 Yhteyden tilan mittaus	25
3.3.3 Lokinpidon todennus	25
4 YHTEENVETO	26
LÄHTEET	27

## SYMBOLILUETTELO

API	<i>Application programming interface</i> . Ohjelmointirajapinta, jonka välityksellä ohjelman osat ovat välityksessä toisiinsa.
APN	<i>Access Point Name</i> . Osoitin, jolla Internet-yhteys avataan mobiililaajakaistaisissa sovelluksissa.
AT-komento	<i>Attention</i> -komento. AT-alkuinen komentosarja, jolla ohjataan mo-deemeja sekä kysellään tietoja.
DLL	<i>Dynamic Link Library</i> . Tyypillisesti erillinen tiedosto, joka sisältää suoritettavia funktioita tai resursseja.
GPRS	<i>General Packet Radio Service</i> . Pakettikytkentäinen tiedonsiirtopalvelu, jota käytetään langattoman Internet-yhteyden muodostamisessa.
Ini-tiedosto	<i>Initialization file</i> . Kokoonpanotiedosto, joka siirtää ja tallentaa nimi-arvo-parametreja ohjelmalle.
RAS	<i>Remote Access Service</i> . Yhdistämiseen ja soittamiseen tarvittava palvelu.
RS232	<i>Recommended Standard 232</i> . Yleinen sarjaporttiväylä esimerkiksi mo-deemin ja PC:n välillä.
XML	<i>Extensible Markup Language</i> . Xml-kieli on rakenteellinen kuvauskieli, joka auttaa jäsentämään laajoja tietomääriä selkeästi.
XPath	Xml-kielessä dokumentin sijainnin ilmaisin, joka tarjoaa mahdollisuuden kysellä dokumentin tiedostopolun osia tietyillä valintakriteereillä.

## 1 JOHDANTO

Connection Manager on ohjelmistosovellus, jonka tarkoitus on ohjauskäskyjä antamalla selvittää modeemien laitetietoja, muodostaa yhteys Internetiin ja valvoa yhteyden tilaa. Yhteyden tilasta kerätään myös tietoa lokiin. Ohjelmisto pitää siis sisällään samanlaisia ominaisuuksia, mitä esimerkiksi nettitikuissa ja muissa mobiililaajakaistaisissa sovelluksissa käytetään.

Insinööriyön aihe saatiin vuonna 2012 toimeksiantona kajaanilaisen IT-yrityksen Sunit Oy:n tuotekehitysyksiköltä. Aikaisempi verkkoyhteyden hallintaohjelmisto oli osoittautunut työlläksi päivittää, joten oleellista oli saada vaihtoehtoisia ratkaisuja tähän. Toimeksiantona saadun ohjelman oli oltava selkeä käytöltään ja dokumentoinniltaan. Koska vastaavat ohjelmistosovellukset kehittyvät edelleen, toteutus tuli tehdä rakenteeltaan mahdollisimman modulaarisesti, jotta uusien ominaisuuksien, kuten esimerkiksi uusimpien verkkotekniikoiden lisääminen jatkokehitysversioihin olisi mahdollisimman vaivatonta. Myös yksi tutkimusongelma oli ennakoida, kuinka ohjelma käyttäytyy poikkeavissa yhteystilanteissa, kuten satunnaisten katkeamisten aikana.

Ohjelman taustalla suoritettavat rajapintojen kommunikoinnit, tunnistamiset ja komentojen tiedostosta parsiminen kirjoitettiin C/C++ kielellä omaksi dll-tiedostoksi. Dll:n lukuapuna käytettiin erillistä xml-tiedostoa, joka sisältää ohjaukseen perustuvia modeemikäskyjä.

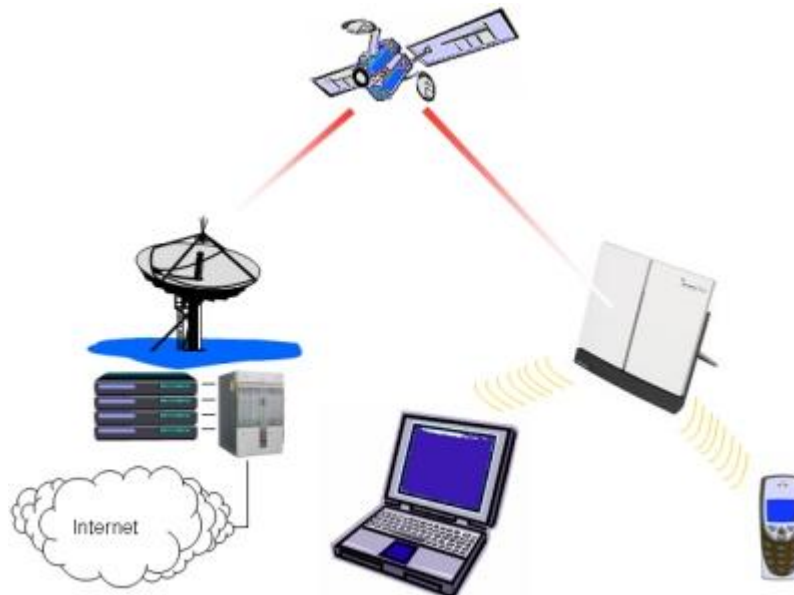
Ulospäin näytettävässä käyttöliittymässä ohjataan dll-tiedostoa ja sillä ilmaistaan käytettävä modeemi ja operaattori sekä tietoa Internet-yhteyden tilasta ja signaalin voimakkuudesta. Ohjelma kerää taustalla myös yhteydenkulusta lokitietoa. Ohjelma tallentaa tarpeellisia asetuksia erilliseen kokoonpanotiedostoon, josta ne myös ladataan seuraavalla kerralla käynnistäessä. Ohjelmiston käyttöliittymä kirjoitettiin C#:lla.

Insinööriyön kirjallisessa osiossa selostetaan Connection Managerin taustalla vaadittavaa teoriaa, suunnittelun ja toteutuksen vaiheita sekä niistä saatuja tuloksia. Asiakkaan toiveesta ohjelman lähdekoodit ja vuokaaviot on salattu.

## 2 VERKKOHALLINTAOHJELMISTON TEORIA

### 2.1 Internet-yhteyden muodostamisen edellytykset

Kiinteän laajakaistayhteyden korvaavana vaihtoehtona voidaan yhä suuremmissa mahdollisuuksissa käyttää liikkuvaa eli mobiililaajakaistaista yhteyttä. Mobiililaajakaistaisen Internet-teknologian suurimpina hyötyinä voidaan pitää sen liikutettavuutta sekä joustavuutta. Sen haittapuolena on puolestaan yhteyden tilan satunnainen epävakaus. Esimerkiksi tiheään asutulla alueella verkko voi ruuhkautua kun taas harvemmin asutulla alueella on vaikeuksia löytää kantoaaltoa. Kuva 1 esittää liikkuvan laajakaistan teknologisia toiminnallisuuksia.



Kuva 1. Mobiililaajakaistaisen teknologian toiminta.[1]

Internet-yhteys mobiililaajakaista-modeemilla avataan yhteysosoitteen eli APN:n kautta. APN voidaan määrittellä joko modeemilaitteen tiedonsiirron asetusprofiiliin tai tietokoneen modeemiasetuksiin. Yhteyden todentamiseen käyttäjä tarvitsee myös soitettavan puhelinnumeron, käyttäjänimen ja salasanan. Useissa palveluntarjoajan tapauksissa soitettava numero on muotoa \*99# ja jälkimmäiset profiilikentät jätetään tyhjiksi.

Kun yhteys toiseen modeemiin on saatu soittamalla, pelkkä kantoaallon kuuluminen ei vielä riitä. Tietokoneen hallinnan palveluille on annettava tästä tieto ja palvelut toimivat omien tehtäviensä mukaan Internet-yhteyden muodostamiseksi. Palvelut-työkalun avulla voidaan



hallita paikallisten tietokoneiden ja etätietokoneiden palveluita. Yksi tällainen yhdistyspalvelu on RAS, eli *Remote Access Service*. Palvelun voi joko käynnistää, pysäyttää, keskeyttää tai käskä jatkamaan käyttöä. [2] Kun Internet-yhteys muodostetaan yhteysosoitteen kautta, käyttäjä saa TCP/IP:n kautta käyttöönsä joko julkisen tai muunnetun IP-osoitteen. Mobiililaajakaistais-ten sovellusten yhdeksi huomioiseikaksi muodostuu se, ettei tämä IP-osoite pysy samana, vaan yhteyden käyttämä osoite myös vapautuu toisten käyttäjien käytettäväksi yhteyden päät-tyessä. [3.]

Liikkuvassa laajakaistayhteydessä verkonhallintaohjelmistossa käytetään eri verkkotekniikoi-ta. Pakettikytkentä on tiedonsiirtomenetelmä, jossa data jaetaan paketeiksi tiedonsiirtoa var-ten. Pakettikytkentäisessä tiedonsiirrossa verkosta ei varata kiinteää päästä päähän kulkevaa reittiyhteyttä tai siirtokapasiteettia tiedonsiirtoa varten. Verkon laitteet välittävät paketit ot-sikkokentässä olevan osoitteen perusteella käytettävän verkon yli. Näin peräkkäisten paketti-en on mahdollista siirtyä eri reittejä pitkin ja jopa järjestystä vaihtaen. Tunnetuin pakettikytkennän sovellus on Internetin perustana toimiva IP-protokolla. Tällä hetkellä mobiiliverkois-sa käytettävistä pakettikytkentäsovelluksista yleisin tyyppi on GPRS. [4.]

OSI-malli kuvaa tiedonsiirtoprotokollien yhdistelmää seitsemässä kerroksessa. Kukin kerrok-sista käyttää yhtä alemman kerroksen palvelua ja tarjoaa palveluja yhtä kerrosta ylemmäs. OSI-malli on käsitteellisesti ehjä ja kansainvälinen ISO-standardi. Sekä TCP/IP että OSI-malli koostuvat kerroksista, ja osa näiden kerroksista on myös samanlaisia. OSI pitää raja-pinnat, palvelut ja protokollat erillään. TCP/IP taas ei tee näille selkeitä eroja, mutta eroja on jälkeinpäin istutettu malliin. OSI on kerrosmalli, jossa alemmat kerrokset on piilotettu ylemmiltä, kun taas TCP/IP muodostaa hierarkian, jossa sekä verkkokerros että kuljetusker-ros ovat näkyvissä sovelluksille. [5.]

TCP/IP-ohjelmistoissa on usein mukana työkaluja yhteyden testaamista varten. Ping on työ-kalu, joka lähettää ICMP-protokollan mukaisia vastauspyyntöviestejä halutuille laitteille ja kerää saaduista vastauksista erilaista tietoa. Jokaisen laitteen, joka toteuttaa IP-protokollan, on vastattava vastauspyyntöviesteihin. [6.]

### 2.1.1 Modeemit

Modeemi on laite, joka siirtää tietokoneiden käsittelemää tietoa puhelinlinjaa pitkin linjan toiseen päähän. Modeemi muuntaa siirrettävän tiedon ääneksi, jota vastaanottava modeemi kuuntelee ja muuntaa sen linjan toisessa päässä takaisin dataksi. Näin tieto liikkuu modeemeihin kiinnittyneiden laitteiden välillä.

Vaikka liikkuvien laajakaistojen sovellukset ovat kehittyneet huimasti viimeisen kymmenen vuoden aikana, esiintyy niiden yhteyksissä silti edelleen enemmän epävarmuuksia kiinteisiin laajakaistoihin nähden. Tutuin ongelmatilanne käyttäjälle on yhteyden äkillinen hidastuminen tai katkeaminen. Puhelinlinjoilla kuuluvat kohinat haittaavat tätä tiedonsiirtoa ja aiheuttavat esimerkiksi ei-toivottuja merkkejä tai taukoja yhteyteen. Modeemi voi siis tulkita häiriön välitettäväksi dataksi. Puhelinlinjan häiriöiden määrää voidaan edelleen pitää hyvin paikallisena suureena. Nykyaikaiset virheen korjaavat modeemit osaavat tosin peittää linjan häiriöitä käyttäjän näkyvistä.

Modeemeita on kahta päätyyppiä: ulkoinen modeemi liitetään mikeroon erillisellä kaapelilla, korttimodeemi taas asennetaan koneen sisälle sopivaan lisälaittekorttipaikkaan. Valinta modeemityyppien välillä on sopivuuskysymys: ulkoinen modeemi tarvitsee pöytätilaa, maksaa hieman enemmän, mutta se on toisaalta helpompi vaihtaa toiseen koneeseen. Myös merkkivaloista on joskus apua diagnostiikan kannalta. Korttimodeemin etuna taas on sen piilotettavuus, mutta se on huonompi siirreltävyydessä. Tiedonsiirto-ominaisuuksiltaan modeemityypeillä ei ole eroa.[7.]

Laittevalmistajilla on tarjota myös tuotekehityskäyttöön soveltuvia modeemialustoja. Tähän tarkoitukseen ovat valmistaneet modeemeja esimerkiksi Motorola ja Telit. Kehitysalustoissa on yleensä liityntämahdollisuuksia RS232-portin tai USB:n kautta. Telit-alustassa on hieman vähemmän Internet-yhteyteen etua tuovia ominaisuuksia kuin Motorolan alustassa, mutta se soveltuu GPS-testaukseen sekä modeemin tunnistamisvaiheeseen hyvin yksinkertaisten ominaisuuksiensa vuoksi. Molemmat modeemityypit ovat ulkoisia. [8.]

### 2.1.2 AT-komennot

AT-komentosarja on standardoitu protokolla, jolla modeemien kanssa kommunikoidaan. AT-komentoja on ollut käytössä jo 70-luvulta saakka, kun Hayes-yhtiö käytti niitä Smartmodemin ohjauksen kehitysratkaisuna lankaverkon analogisissa ja ISDN-modeemeissa. Tästä juontaa myös AT-komentojen toinen nimitys Hayes-komentosarja. Kiinteiden laajakaistayhteyksien läpimurtovaiheessa komennot kuitenkin jäivät tilapäisesti vähemmälle käytölle. Nyt, kun mobiililaajakaistaisten sovellusten, kuten nettitikkujen ja matkapuhelimien Internet-yhteydet ovat yleistyneet, AT-komentojen hyödyntäminen on jälleen koettu tarpeelliseksi ja monipuoliseksi keinoksi ohjata modeemilaitteita. [9.]

Kaikki AT-komennot alkavat ”AT”-prefiksillä ja päättyvät Enter-merkkiin. Tätä voi hyödyntää myös ilmaisemaan viestin tunnistettavuutta sarjaliikenteen kautta ohjelmoimisessa. Ainoa poikkeus AT-alkuisuudesta on +++-komento, jota käytetään yhteyden katkaisemiseen. Yksi komento voi olla maksimissaan neljäkymmentä merkkiä pitkä. Modeemin oman tulkinnan kannalta ei ole väliä, onko annettu AT-käskey kirjoitettu pienillä vai isoilla kirjaimilla, mutta kirjainkokojen keskenään sekoittamista ei suositella käytettäväksi saman käskyn yhteydessä. Lähes kaikki modeemit käyttävät samoja komentoja, ja niitä on myös taulukoitu. Kuitenkin on kannattavaa tarkistaa valmistajakohtaisesti modeemin datalehdiltä komentojen mahdollisia poikkeuksia. Esimerkiksi pakkauksen ja vuonvalvonnan asetuksissa on jo modeemikohtaisia eroja.

Yksinkertainen esimerkki AT-komennosta on, kun halutaan testata, mikä on tietokoneeseen kytketyn sarjakaapelin toisessa päässä käytössä olevan modeemin valmistaja. Tämän kokeilun voi testata syöttämällä vaikkapa HyperTerminal tai TeraTerm-ohjelman kautta modeemiin merkit AT+GMI, jolloin ohjelman ja modeemin välinen dialogi näyttää seuraavanlaiselta:

AT+GMI

Telit

OK

Ylimmällä rivillä siis annettiin heräte kysymällä modeemin valmistajaa. Keskimmaisella rivillä modeemi vastasi Telit ja viimeisellä rivillä modeemi kuittasi, että AT-komento oli kunnossa ja se suoritettiin oikein.

Tyypillisesti modeemin takaisin antaman ilmoituksen eli vasteen toteutumisen tunnuksena voi olla joko OK tai ERROR. AT-komento voi myös pitää sisällään laajempia syöttöparametreja, kuten verkkotyypin tunnuslukuja, ja nämä täytyy myös pystyä erottelemaan kun herätettä tai takaisin palaavan viestin vastetta tulkitaan. Taulukossa 1 on esitetty myös muita tyypillisimpiä vasteita, joita modeemilta saadaan ulos tiettyä herätettä vastaan. [10.]

Taulukko 1. AT-komentojen yleisimpiä vasteita. [7]

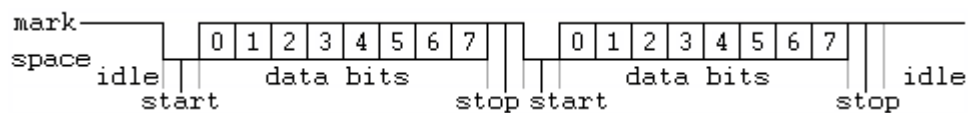
Vaste	Vasteen merkitys
OK	Komento oli kunnossa ja se suoritettiin.
ERROR	Komennossa oli jotain vikaa, eikä sitä voitu suorittaa.
BUSY	Puhelinnumero, johon soitettiin, on varattu.
CONNECT	Yhteys toiseen modeemiin on saatu. Modeemit viheltävät toisilleen Carrier-kantaaltoa, josta ne tietävät olevansa yhteydessä toisiinsa.
NO CARRIER	Toisessa päässä oleva modeemi hukkui kuulumattomiin tai sen kantaalto ei kuulu. Modeemi antaa tämän ilmoituksen kun yhteys on purettu tai katkennut.
NO DIALTONE	Modeemi ei kuule keskusääntä, kun se nostaa luurin soittaakseen.

### 2.1.3 RS232-sarjaliikenne

RS232 on yksi yleisimmistä ja yksinkertaisimmista sarjaliikenneväylistä ja alun perin tarkoitettu juuri keskustietokoneen ja pääteidenväliseen liityntään modeemien ja puhelinverkon välityksellä. Tyypillisiä RS-liitännän lähetysopeuksia ovat 1200, 2400, 4800, 9600 ja 19200 bps. Onnistuneen tiedonsiirron aikaansaamiseksi sekä lähettäjän että vastaanottajan tiedonsiirto-parametrien tulee olla samat.

Asynkronisen ja synkronisen tiedonsiirron ero on siinä, että asynkronisessa siirrossa jokaiseen lähetettävään merkkiin koodataan aloitusbitti, lopetusbitti sekä mahdollisia tarkistusbittejä – tämä myös syö osan tiedonsiirtokapasiteetista. Synkronisessa tiedonsiirrossa nopeus taas saadaan tahdistussignaalista, jolloin siirtokapasiteettiakin saadaan käyttöön lähes täysin. Synkronisessa siirrossa siirrettävien merkkien välillä ei ole muita merkkejä. [11.]

Kuvassa 2 on esitetty bitit Start, kahdeksan databittä ja Stop. Kun liikennettä ei ole, linja lepää Mark-tilassa, eli hiljaisessa Idle-tilassa 1. Liikenne alkaa Start-bitillä ja linja muuttuu Space-tilaan eli 0-tilaan. Databitit kulkevat Start-bitin jälkeen vähiten merkitsevistä bitistä eniten merkitsevään, tosin kuvassa 2 ne on esitetty binääriluvun kirjoitusjärjestyksen kannalta takaperoisessa järjestyksessä. Tämän jälkeen vuorossa on Stop-bitti. RS232-linjalla Mark-tila kulkee alhaisena jännitetasona, mikä tarkoittaa negatiivista jännitettä, jossa polariteetti on toisin päin kuin sarjaliikennepiirillä tavallisesti. [12.]



Kuva 2. Asynkronisen liikenteen peruseriaate. [12]

Ohjelmoidessa RS232-liikennettä vaaditaan hieman perehtymistä sarjaliikenteen asynkroniseen toimintaan. Ohjelmallisena toteutuksena sarjaportille yleensä syötetään merkkijonona käskyjä, joista sarjakaapelin päässä oleva laite suorittaa protokollansa mukaisesti toimintoja. Jos liikenne on kaksisuuntaista, laite voi antaa myös palautuksena vasteita sekä luettujen databittien määrän.

Kun sarjaportin kautta lähetetään toistuvasti samaa tietoa, viestin pituus on yleensä helppo määrittellä etukäteen. Vastaanotto puolen ohjelmoiminen on vaikeampaa, koska aina ei voida tarkkaan voida ennustaa, kuinka pitkä viestin sanoman pituus on, sekä kulkeeko kaikki tar-

peellinen tieto välttämättä edes kerralla yhtenä pakettina. Joskus voi myös tulla tilanne, ettei viestiä ruuhkautuksen takia ehditä saamaan perille. Tällöin viestiä lukevalle funktiolle on annettava mahdollisuus jäädä odottamaan tai aloittamaan tiedon lähettäminen uudelleen. Sen vuoksi vastaanotettavalle viestille on myös ohjelmallisesti määriteltävä tunnistettava alku ja loppu.

Yksi tapa tällaisessa sarjaliikenteen takaisinluvussa on käyttää hyväksi tilapäistä rengaspuskuria, joka palauttaa merkkijonoarvoja tietyn mittaisina. Rengaskuvaus viittaa tapaan, kuinka tiedon tallennus- ja lukupaikka kiertää puskuria. Tallennettaessa puskuriin tietoa kirjoituspaikka etenee kohti puskurin loppua. Kun kirjoituspaikka ylittää puskurin koon, se siirretään takaisin alkuun. Tiedon lukukohta liikkuu samalla tavalla eteenpäin, palaten alkuun puskurin lopussa. Näin taustalla koko ajan toimiva puskuri myös varastoi väliaikaisesti ylimääräisen tiedon, jota ei vielä ole ehditty käsittelemään. Rengaspuskuria voidaan käyttää esimerkiksi muuntamaan tietovirtaa joka saadaan pakettiverkon yli vaihtelevalla nopeudella muotoon, josta kiinteällä nopeudella sitä kuluttava käsittelijä saa aina samalla viiveellä seuraavan alkion käsittelyyn.

## 2.2 Ohjelmallisen toteutuksen vaatimukset

### 2.2.1 Modulaarinen ohjelmarakenne

Moduulilla tarkoitetaan ohjelman itsenäistä osaa, jolla on omat syöttötiedot, tulostiedot sekä omat toiminnalliset funktionsa. Laajempi ohjelma voidaan koota moduuleista, jolloin puhutaan modulaarisesta ohjelmasta ja modulaarisesta ohjelmoinnista. Tarve modulaarisuuteen syntyy yleensä, kun samanlaiset ohjelman osat toistuvat ohjelmien eri kohdissa. Ohjelmamoduulin sidokset muuhun ohjelmaan tulisi välittyä määriteltyjen parametrien kautta. Tällöin ohjelmamoduuleja tai sen osia on helppo yhdistää tai kopioida ohjelmakoodin muihin paikkoihin, tai muihin laajempiin ohjelmiin. Tämä lisää pitkällä tähtäimellä myös ohjelmien luotettavuutta ja siirrettävyyttä, koska vaikeasti etsittäviä ohjelmointivirheitäkin syntyy vähemmän. [13.] [14.]

Myös versiopäivitys on yksi syy, miksi moduuliratkaisuja suositaan. Varsinkin pitkien aikavälien käyttöön suunniteltujen ohjelmien on oltava mahdollisimman luontevasti laajennettavissa. Toimintojen hajauttamiset on tehtävä järkevästi omiin aliohjelmiin ja liian raskasta koodaamista, kuten toistuvien funktioiden erikseen kirjoittamista on vältettävä. Helpompaa jatkon kannalta on, kun järjestää tietojen syöttöä ja lukemista erillisten tiedostojen välillä.

Myös dokumentoinnin on oltava selkeää laajennuksen kannalta. Lohkokaaviot, luokkakaaviot ja vuokaaviot selkeyttävät hyvin ohjelman toimintaa graafisena kuvaustapana. Kaavioita voi piirtää esimerkiksi Microsoft Visiolla, Prosalla tai Dia:lla. Myös ohjelmakoodin kommentointiin on olemassa nopeasti ohjelmoijaa auttavia työkaluja. Esimerkkinä tästä on Doxygen-dokumentointijärjestelmä, joka kuvaa web-näkymäisesti erikseen kaikki oleelliset aliohjelmat ja parametrit.

### 2.2.2 DLL-tiedosto ohjelman suorituksen taustalla

Dll-tiedosto tulee sanoista *digital link library*. Kyseessä on kirjasto, jonka tarkoitus on linkittää erillisenä osiona kaikki käytettävät aliohjelmat, joita ohjataan kutsumalla niitä toisella moduulilla. Tämä on myös sikäli kätevä tapa, jos dll halutaan integroida toiseen ohjelmaan, on helpompaa linkittää tiedosto kun monta riviä erillistä koodia. Dll-tiedostot ovat käyttökelpoisimpia laajoissa sovelluksissa, koska silloin voidaan toteuttaa yhteisiä toimintoja sovelluskonaisuuksissa. Linkitettävät kirjastot helpottavat ohjelmiston päivitystä, koska päivitys on monesti jonkin Dll:n korvaaminen uudemmalla versiolla.

Dll:n eroaa tavallisesta Windows-ohjelmasta siinä, etteivät ne ole suoraan suoritettavissa olevia kuten exe-tiedostot, eivätkä dll:t yleensä ota viestejä vastaan. Dll:t ovat erillisiä tiedostoja, jotka sisältävät tavallisesti funktioita tai resursseja. Ne suoritetaan vasta kun jokin toinen moduuli kutsuu kirjaston funktiota. Useassa tapauksessa tämä moduuli on käyttöliittymä tai järjestelmän ylempi osa. Itse dll sisältämä ohjelmakoodi ei välttämättä poikkea paljonkaan muista Visual Studion projektikoodista, vaan dll-lopputuloksena syntyy vasta käännösvaiheessa. Staattinen linkitys luodaan ohjelman käännösvaiheessa kun objektimoduuleita, kirjastoja tai resurssitiedostoja linkitetään exe-tiedostoon. Dynaaminen linkitys tapahtuu vasta ohjelman suorituksen aikana. Tällöin funktiokutsu linkitetään kirjastomodulissa sijaitsevaan funktioon.

Dll-tiedoston hakujärjestys toteutuu seuraavanlaisesti: Etusijalla on hakemisto, joka sisältää sovelluksen exe-tiedoston. Sen jälkeen on prosessin oletushakemisto eli current directory, sekä Windows-järjestelmähakemisto, kuten system32. Kolmanneksi haetaan Windows-hakemisto sekä polkuasetuksissa olevat hakemistot. Sovellusta ei voida käynnistää, mikäli näitä löydy. Implisiittinen linkitys taas ilmaisee projektiin tietoja riippuvuussuhteista ja Dll:n riippuvuutta johonkin edellä mainituista hakemistoista. [15.]



### 2.2.3 XML-tiedostojen hyödyntäminen tietokantana

XML tulee sanoista *Extensible Markup Language*. Xml-kieli on rakenteellinen kuvauskieli, joka auttaa jäsentämään laajoja tietomääriä selkeästi. XML on myös standardi, jossa tiedon merkitys on kuvattavissa tiedon sekaan. Xml-kieltä käytetään sekä formaattina tiedonvälitykseen järjestelmien välillä että formaattina dokumenttien tallentamiseen. Kuvassa 3 on esitetty xml-tiedoston Modem-elementin sisältämää alitietoa attribuuttiarvoina.

```

<?xml version="1.0" encoding="UTF-8"?>
- <xml>
  - <Modem>
    <Manufacturer value="Motorola"/>
    <Model value="H24"/>
    <MaxSSBars value="5"/>
    <SupportMultichanneling value="1"/>
    <Support3G value="1"/>
  + <Commands>
  </Modem>
+ <Modem>
+ <Modem>
</xml>

```

Kuva 3. Esimerkki xml-tiedoston osasta.

XML on tekstimuotoista ja muistuttaa html-kieltä, jolla www-sivut kirjoitetaan, ja ne kummatkin ovat SGML-kielen yksinkertaistettuja osajoukkoja. Xml-kieli ei kuitenkaan ole tarkoitettu sivunkuvauskieleksi kuten html, vaan sillä kuvataan tiedon rakenne ilman ennalta määrättyjä koodeja. Kielellä voidaan myös muodostaa uusia koodeja, joiden avulla voidaan luoda dokumentteja hyvinkin monipuolisiin käyttötarkoituksiin. [16.] [17.]

Xml-tiedoston rakenne koostuu riveistä, joita kutsutaan nodeiksi eli solmuiksi, ja ne sisältävät omia alisolmuja, joita kutsutaan elementeiksi ja niiden lainausmerkeissä olevaa tekstisisältöä attribuuteiksi. Kuvausrakennetta voidaan myös ajatella eräänlaisena sukupuuna.

Xml-dokumenttien käsittelyyn on tarjolla myös paljon erilaisia työkaluja. Yksi näistä on XPath, joka ilmaisee tiedostopolun sijainnin ja tarjoaa mahdollisuuden kysellä xml-dokumentista osia tietyillä valintakriteereillä. [18]

### 3 TYÖVAIHEET

#### 3.1 Suunnittelu

##### 3.1.1 Perehtyminen ja tiedonkeruuvaihe

Insinööriyön aihe Connection Manager-ohjelmistosta otettiin puheeksi ensimmäisen kerran jo vuoden 2012 keväällä. Ensimmäinen palaveri työtä koskien pidettiin vuoden 2012 elokuun lopulla. Tässä vaiheessa käytiin läpi, millaisia Connection Manager-ohjelmia on jo olemassa, ideoitiin mitä ominaisuuksia tulevassa työssä olisi hyvä olla olemassa ja millaisilla resursseilla ne olisi mahdollista toteuttaa. Toki myös vilkaistiin mitä toimintoja tilaajan ensimmäisessä managerissa oli, sekä kuinka se oli toteutettu ja mistä ominaisuuksista halutaan uusia ratkaisuja.

Ohjelmakoodin kirjoittamiseen ja testaukseen saatiin tilaajalta kannettava HP-tietokone sekä tarvittavaa testausohjelmistoa. Managerin testaustietokoneena oli tarkoitus käyttää Sunit fD-mallia, jossa on valmiina tiettyjä ominaispalveluita yhteyden muodostamiseen. Testausmodeemiksi saatiin käyttöön myös Motorola H24-modeemintestausalusta, joka tukee RS232-yhteyttä. Kajaanin ammattikorkeakoululta saatiin apukoekäyttöön myös Telit GM862-testausalusta. Liikkuvien laajakaistojen modeemit tarvitsevat yhteyden muodostamiseen myös datayhteyttä tukevan SIM-kortin. Korttiratkaisuksi valittiin yksityiskäytössä oleva nettitikun SIM-kortti.

Alkusyksyllä 2012 etsittiin tietoa muun muassa yleisimmistä AT-komennoista ja niiden toimintoista. Annettujen modeemien spesifikaatioita tutkittiin ja tarkistettiin, mitä ominaisuuksia niistä löytyy. Motorolan H24 modeemin testialustaa testattiin myös jo tuolloin luomalla yhteys USB:n välityksellä. Yksinkertainen kokeiluesimerkki sillä oli aluksi antaa modeemille käskyjä HyperTerminal-ohjelmalla ja katsoa millaisessa muodossa saadaan takaisin vasteita.

Samalla tutkittiin, millaisia ominaisuuksia omassa käytössä olevissa nettitikuissa olevissa Connection Managereissa oli tarjota käyttöön. Pohdittiin myös, onko kaikkia vastaavia ominaisuuksia tarpeellista tai aikataulujen puitteissa edes mahdollista sisällyttää insinööriyön versioon.

### 3.1.2 Vaatimusmäärittelyn laatiminen

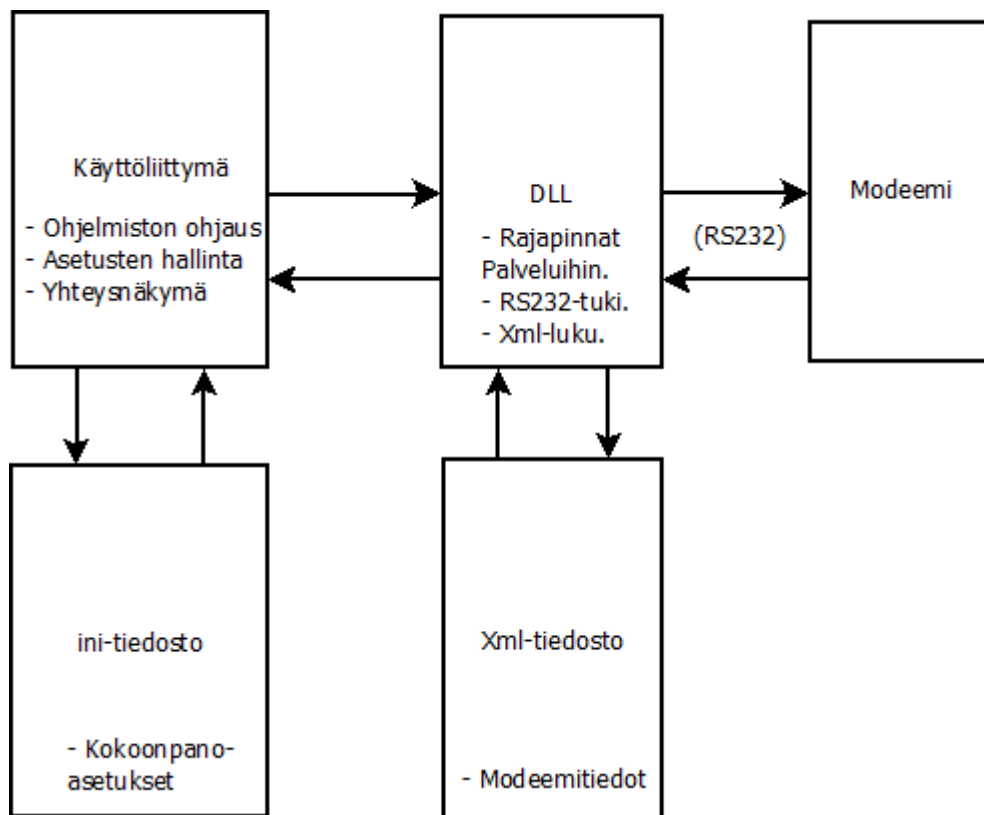
Kun ensimmäinen ideointikeskustelu työstä oli pidetty, tuli laatia suunnitelma siitä, millaista projektia lähdetään toteuttamaan. Vaatimusmäärittelmä toimii projektisuunnitelman tavoitteiden selkeyttämisen tukena ja se helpottaa myös lopputestausvaiheessa tarkastellessa tuloksia, kun verrataan, ovatko ne sitä, mitä niiden alun perin on vaadittu olevan. Vaatimusmäärittelmä määrittelee myös suoritettavien toimintojen tärkeysjärjestyksen ja tekee jo siten palveluksen työvaiheiden suunnitteluun. Esimerkiksi käyttöliittymänäkymässä infoikkunan olemassaolon puuttuminen ei välttämättä ole niin suuri vahinko verrattuna siihen, että ohjelma ei tue RS232-liikennettä. Vaatimusmäärittelmä antaa myös suunnittelijalle suojaa, ettei kesken projektin ilmaannu uusia toteutusideoita - poikkeuksena toki toteutusratkaisut hyvine perusteluineen.

Vaatimusmäärittelmään listattiin ominaisuuksia, muun muassa mitä tietoja Connection Managerin käyttöliittymä tulee näyttämään käyttäjälle, millä laitteilla käyttöjärjestelmällä ohjelman tulee toimia, millaisiin rajapintoihin se on yhteydessä ja mihin asetuksiin käyttäjä voi itse vaikuttaa. Tässä tapauksessa ohjelma esimerkiksi vaati toimiakseen tietokoneen, jonka käyttämä käyttöjärjestelmä on Windows XP tai Windows 7.

Koska työ on laaja ja oltiin tekemässä vasta ohjelmiston pohjaa, tärkeää oli jo alkuvaiheessa myös päättää, mitä laajan kokonaisuuden ulkopuolelle voidaan rajata ensimmäisessä versiossa. Vaatimukset priorisoitiin asteikolla 1-5. Mitä korkeamman numeron prioriteetti sai, sitä vähemmän kiireellinen sen toteutuminen oli. Esimerkiksi rajattiin, että tämä insinööriyön versio tukee vain modeemiyhteyksiä sarjaportin kautta. Vastaavasti WLAN-tuen osalta päätettiin, että se voidaan ottaa esille vasta myöhemmin mahdollisessa jatkokehitysvaiheessa.

### 3.1.3 Ohjelmarakenteiden kuvaaminen lohko- ja vuokaavioilla

Kun vaatimusmääritelmä oli hyväksytty, seuraava vaihe oli hahmotella kuvaamalla ohjelman toteutus lohkokaaevioina ja tarkentaa kaavioita vähitellen pienemmiksi paloiksi. Kaaevio kuvien piirtämiseen käytettiin Dia-piirtoohjelmaa. Aluksi piirrettiin kokonaisuus ja ohjelman osien väliset riippuvuudet lohkokaaeviotasolla ja myöhemmin osat pilkottiin erillisiksi vuokaavio-osiksi. Kuva 4 yksinkertaistaa, mitä osia kokonaisuuteen kuuluu. Tilailajan pyynnöstä tarkemmat vuokaaviot on julistettu salaisiksi.



Kuva 4. Ohjelmiston yleisosat.

Päätettiin, että käytännössä kaikki ohjelman suorituksen taustalla tapahtuva työnteko hajauteetaan omaan dll-tiedostoon. Dll vastaa sarjaportin liikennöinnistä, RAS-kommunikoinnista, xml-tiedostosta haettavien tietojen parsimisesta ja tietojen välittämisestä käyttöliittymän näkymään. Ohjelman dll-tiedoston kirjoituskieleksi valittiin C/C++ ja sitä ohjaavan käyttöliittymän kirjoituskieleksi C#.

## 3.2 Toteutus

### 3.2.1 DLL-tiedoston ohjelmallinen toteutus

Dll-tiedostoon luotiin ensimmäiseksi API-rajapinta, jonka kautta funktiot viedään tunnistettavasti käyttöliittymälle. Yhteyden toiminnallisuutta voitiin testata aluksi sijoittamalla ohjelmakoodiin vaikkapa erilliseen Debug-tekstitiedostoon päätyviä tulostuksia sekä käyttämällä tiedostojen välisiä MessageBox-ilmoituksia.

Seuraavaksi tärkein osa-alue, mikä tuli toteuttaa heti alussa, oli sarjaliikenteen hallinta ohjelmallisesti. Ohjelmoidessa RS232:sta, sen fyysiseen puoleen ei tässä työssä ollut oleellista ottaa kantaa muuten kuin tarkastamalla, että porttia avatessa yhteysasetukset ovat samat kuin Ohjauspaneelin Laitehallinta-näkymässä. Tämän jälkeen tarpeellista ei ole tehdä muuta kuin kytkeä sarjakaapeli modeemiin ja käytettävän COM-portin välille.

Sarjaportinkäsittelyyn hyödynnettiin erästä aikaisemmin töissä toteutettua ohjelman osaa, joka lähettää RS232-kaapelia pitkin tavujonoina ohjauskomentoja. Tämä dll:n osa myös tunnistaa kaikki käytössä olevat portit, sekä sulkee ja avaa ne eri tilanteissa käskettäessä. Lähetysosaa muutettiin ainoastaan lähettämään nyt tavujonojen sijaan myös merkkijonotietoa, joka alkaa AT-prefiksillä ja päättyy Enter-merkkiin. Merkkien lähettämisessä laitteelle ei oikeastaan ilmennyt ongelmia.

Vastaanottoapuoleen tuli kuitenkin tehdä muutoksia, jotta portista voitaisiin lukea tietoa merkkijonona, koska esimerkiksi modeemivaste täytyy vielä lähettää myös käyttöliittymälle ymmärrettävässä tekstimuodossa. Vasteen alusta täytyi myös parsia muun muassa ylimääräisiä rivinvaihtoja. Taustalla täytyi myös olla esillä globaali taulukko, josta oikeaksi parsittu tieto siirretään käyttöliittymälle. Tiedon vastaanottoapuolen käsittely-osuus muodostui lopulta yhdeksi haasteellisimmista työvaiheista, koska vastetieto ei aina palautunut perille halutunlaisessa muodossa tai sen kokonaisuudesta saattoi puuttua osia. Yksi vaihtoehto on tyhjentää portinlukupuskuri heti viestin ja bittimäärän lukemisen jälkeen. Yksinkertaisten komentojen osalta tämä oli vielä toimiva, mutta jos sanoma on pitempi kuin kahdeksan bittiä, tai se sisältää useampia muuttujia, täytyy prosessoimaton tieto viedä johonkin talteen. Tätä varten täytyi luoda rengaspuskuri, jossa tieto kiertää väliaikaisesti kehää täyttymiseensä saakka. Ideana on samalla välttää myös yhteentörmäyksiä.

Seuraavaksi dll:ään kirjoitettiin rajapinta tietokoneen RAS-palveluun ja sen liityntäfunktioihin. Tässä kohtaa oleellinen tutkimusongelma oli selvittää, kuinka toimitaan tilanteissa, joissa yhteyden muodostaminen ei onnistukaan tai yhteys katkeaa kesken käytön, ja kuinka tällaisesta saadaan tieto toimenpiteitä varten. Yksi ratkaisu tähän on Ping-komennon käyttö. Kirjoitettua Ping-ohjelmaa käytettiin tässä kohtaa tietyn www-osoitteen etsintään ja tarkistetaan, häviääkö verkossa paketteja ja minkälaisia viiveitä verkko aiheuttaa eri laitteiden välille. Mikäli huomattava määrä paketteja häviää, verkossa on mahdollisesti jotain vikaa. Ping-ohjelman lisäksi verkkoyhteyden tilaa voitiin ennakoitusti tutkia myös oman AT-komennon avulla. [20.]

Rakennettavan dll-tiedoston oli tarkoitus olla helposti integroitavissa tuleviin muihinkin ohjelmiin, joten funktioiden liiallista kovakoodaamista tuli harkitusti välttää. Koska yleisimmät AT-komennot ovat kaikissa modeemeissa lähes samat, aluksi jokainen komento kirjoitettiin omaksi funktiokseen. Esimerkiksi käskyt, jotka antavat vasteena tunnistamansa modeemin perustiedot, on itse kirjoitettu.

Tammikuun loppupuolella käydyssä välipalaverissa ehdotettiin at-komentojen hakemiseksi tietokantamaista toimintamallia, tässä tapauksessa xml-tiedostosta lukemista. Tilaaajalta saadusta config.xml nimisestä tiedostosta löytyisi jatkossa yleisimpien modeemien tiedot. Mikäli ohjelmistoa on tarkoitus päivittää jatkossa, on huomattavasti helpompaa lisätä uusia solmuja xml-tiedostoon, sen sijaan, että kirjoittaisi koodiin uusia funktioita ja niissä välittyviä parametreja. Xml-formaatista lukeminen ei oikeastaan paljoa eroa mistään muustakaan tekstitiedoston lukemisesta. Tiedon lukeminen voidaan toteuttaa lähes millä tahansa ohjelmamuodolla, kunhan ymmärtää parsimisen kohteen. Tärkeintä on olla tietoinen xml-tiedoston solmujen juurista, sekä tarkistaa, että tiedoston kirjoitusasu on validi.

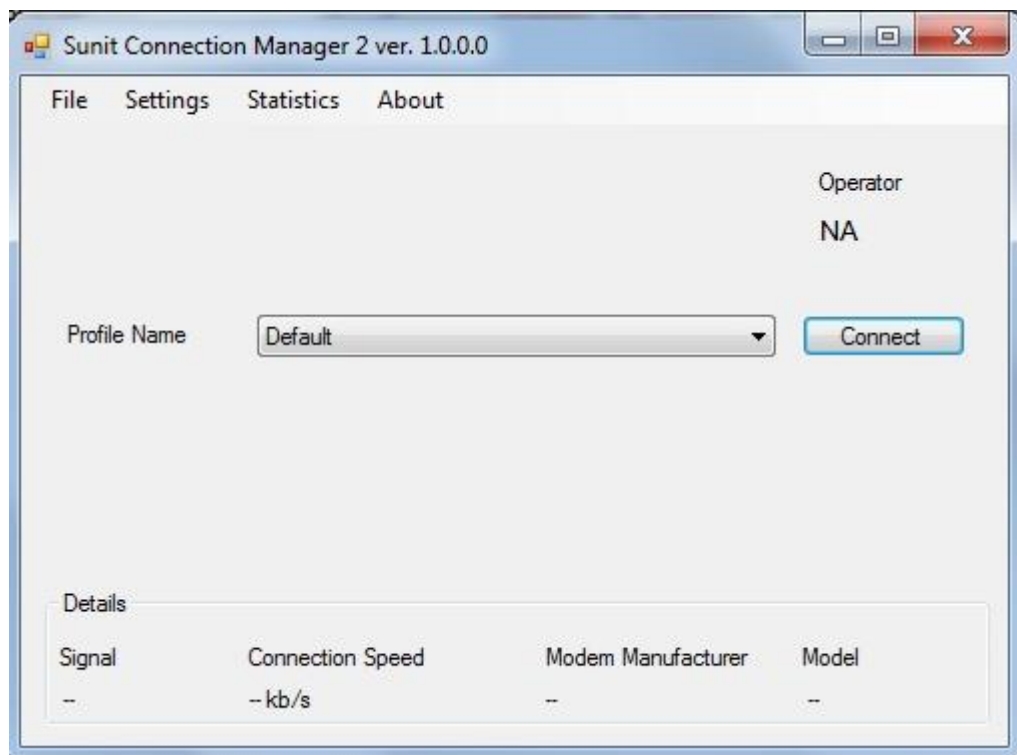
Koodiin kirjoitettiin yksinkertainen xml-lukija, joka lataa config.xml-tiedoston, valitsee XPathia hyödyntäen tarvittavat solmut ja viittaa solmujen poluissa elementin attribuuttiarvoihin. Kun solmut on paikannettu, tapahtuu attribuuttiarvon parsiminen merkkijonoksi, joka voidaan lähettää käskynä modemille. Lukijaa laajennettiin myös siten, että se on staattisesti ohjattavissa kuvan 6 *Device*-parametrin tilanvaihdoksilla. Kun valitaan esimerkiksi Motorolan modeemi, lukuohjelma valitsee vain tämän modeemin alaiset asetukset ja käskyt.

Ohjelman xml-lukuosalla ei ollut vaikeuksia löytää tiedostoa ja lukea sieltä valittuja solmuja, mutta ongelmia ilmeni aluksi hieman attribuutin arvon palauttamisen osalta. Koska mukana oli ylimääräisiä symboleja, jotka haittaavat modeemin käskyn ymmärtämistä, täytyi tehdä vielä erillinen ohjelman osa, joka erottelee epäolennaiset merkit luetun attribuuttirivin alusta. Koska kyseessä on at-komento, loppuun tulee lisätä mukaan vielä Enter-merkki. Lopulta tiedostosta haetun tiedon oli mahdollista kulkea ymmärrettävästi sarjaportin läpi modeemille.

### 3.2.2 Käyttöliittymän ohjelmallinen toteutus

Käyttöliittymän kirjoituskieleksi valittiin C#, koska se on kätevä visuaalisen ilmeen luomiseksi, sekä C#-kirjastoilla saa nopeasti toteutettua laajojakin komentojen ohjauskäsittelyjä. Ensimmäinen vaihe oli luoda liittymän ja dll-rajapinnan välille yhteys. Tässä käyttöliittymä pitää sisällään aliohjelmakutsuja ja niiden referenssiparametreja, joita hallitsemalla dll-tiedoston käskyjä suoritetaan. Ensimmäiset funktiot, jotka toteutettiin, olivat sarjaporttien avaaminen ja sulkeminen. Kun näiden toimivuus todettiin, muita aliohjelmiä voitiin alkaa lisäämään yksi toisensa jälkeen.

Käyttöliittymän ohjaaminen toteutettiin tapahtumilla eli eventeillä. Klassinen esimerkki tapahtumasta on komentoja antavien nappien ja pudotuslaatikkojen luominen. Käyttöliittymän etusivun näkyvässä voidaan suoraan valita käytettävä profiili ja yhdistää. Näkymän yläosan työkaluvalikosta löytyy pääsy muokattaviin asetuksiin, yhteyden statistiikkoihin, infoon sekä ohjelman sulkemiseen. Käyttöliittymän perusnäkyvästä tuli kuvan 5 mukainen.



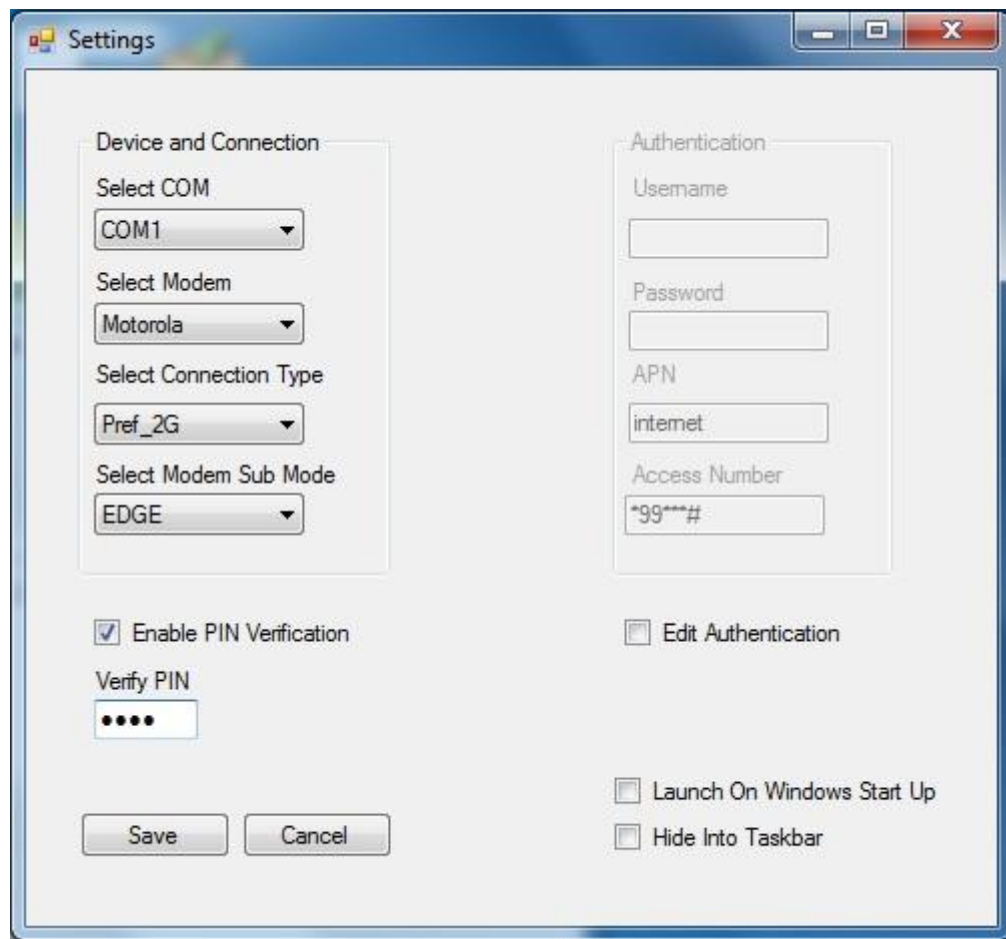
Kuva 5. Käyttöliittymän perusnäkyvä

Heti kun ohjelma avataan perusnäkyväseen, avauksen yhteydessä lähetetään dll-tiedostolle aliohjelmakutsu, jossa tiedustellaan, tunnistaako Manager modeemia. Jos kuvan 5 tekstikenttien



*Modem Manufacturer*, *Model* ja *Operator* alaisuuteen tulostuu ohjelman avauksen aikana valmistajakohtaista tekstiä, tämä on merkki siitä, että yhteys ohjelman ja modeemin välillä on olemassa. Modeemiyhteyden toimiessa voidaan klikata *Connect*-nappia ja yhdistää. Yhteyden muodostuessa ohjelma antaa merkin yhteyden muodostumisesta, jonka perusteella voidaan aloittaa yhteyden tilan tarkkailu. Tekstikenttien *Signal*-ja *Connection Speed*-alatekstit ilmoittavat signaalin voimakkuuden ja yhteysnopeuden. Mikäli yhdistäminen ei onnistu, täytyy käydä tarkistamassa asetukset.

Todella tärkeä käyttöliittymän osa on asetukset eli *Settings*. Asetukset avataan perusnäkyvän yläosan välilehdeltä omaksi lomakenäkymäksi, ja kaikki mitä tässä ikkunassa käytetään, tallentuu kokoonpanoasetuksina Ini-tiedostoon. Käyttöliittymän asetuskäytöstä tuli kuvan 6 mukainen.



Kuva 6. Käyttöliittymän asetusten perusnäkyvä

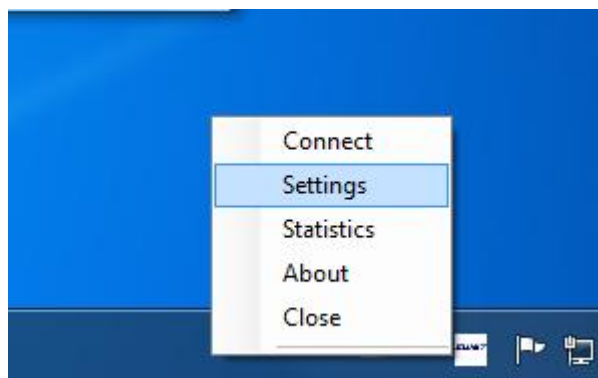
Asetusten muutoksiin vaikutetaan poistumalla joko Save–tai Cancel-nappia painamalla ja asetukset tallentuvat kokoonpanotiedostoon.

Ylävasemmalla, asetusten *Device and Connection*-osiossa vaikutetaan laitetietoihin ja yhteysasetuksiin. *Select COM*-kohdalla voidaan vaikuttaa suoraan modeemiin kytkettävän COM-portin valintaan. Pudotuslista tarjoaa käytettäväksi kaikkia tietokoneesta löytyviä portteja. Näkyville jätettävä portti jää käyttöön, ja se avataan välittömästi, kun näkymästä poistutaan. *Select Modem*-pudotuslaatikolla taas vaikutetaan käytössä olevan modeemin valintaan. Valitsemisen seurauksena haetaan kyseiselle modeemille oikeat arvot xml-tiedostosta. Alla olevilla *Connection Type*-ja *Sub Mode*-pudotuslaatikoilla voidaan samalla periaatteella vaikuttaa vielä signaalin voimakkuuteen ja verkkotekniikkaan.

Oikealla näkyvässä *Authentication*-osiossa voidaan käyttäjälle antaa mahdollisuus muokata oleellinen osa profiilia syöttämällä itse APN-tiedot ja palvelimen valintanumero. Jos halutaan vaihtaa käyttäjänimi ja salasana, myös niiden syöttö onnistuu tätä kautta.

*Enable PIN Verification*-laatikko on toteutettu sallimaan perinteinen PIN-koodin kysely, ja se sallii syöttää alapuolen tekstikenttään PIN-koodin salasananäkymässä. PIN-koodin syöttäminen voidaan myös asettaa vain yhteen kertaan, jolloin se pysyy tallessa seuraavilla kerroilla.

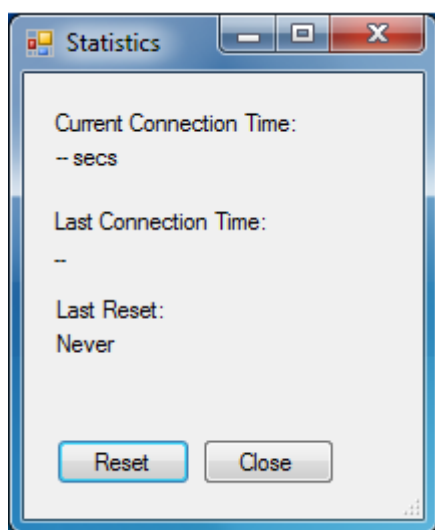
Käyttäjä voi vapaaehtoisesti valita lisäominaisuutena myös sen, että ohjelma käynnistyy seuraavilla kerroilla automaattisesti, kun tietokone käynnistetään uudelleen. Toinen lisäominaisuus on ohjelmanpaneelin piilottaminen käytön ajaksi taustalle. Jos ohjelman haluaa taas auki, se onnistuu klikkaamalla tehtäväpalkin ikonia. Tehtäväpalkkivalikon näkymä on esitetty kuvassa 7.



Kuva 7. Tehtäväpalkkivalikko

Aluksi asetusten tallennus toteutettiin rekisterillä. Rekisterissä funktioille määriteltiin tiloja, jotka voivat olla joko tosia tai epätosia. Yksinkertaisessa sovelluksessa tämä olisi oikein riittävä toteuttamistapa. Mutta koska ohjelmiston tulee voida hallita myös erilaisin kokoonpanoasetuksiin tallennettuja profileja, rekisteri ei välttämättä ole siihen luontevin vaihtoehto. Myöhemmin rekisterin käyttö päätettiin korvata vaihtamalla asetusten tallentaminen ja lukeminen omaan ini-tiedostoon, joka käyttää omia nimi- ja arvo-parametreja. Ini-tiedostoon on helppo päivittää toimivat asetukset ja tiedoston korvausratkaisu on myös kätevä tilanteissa, joissa on tarvetta saada etäapua. Kun ohjelma seuraavan kerran käynnistetään, asetukset latautuvat samoina millaisiksi ne viimeksi jätettiin.

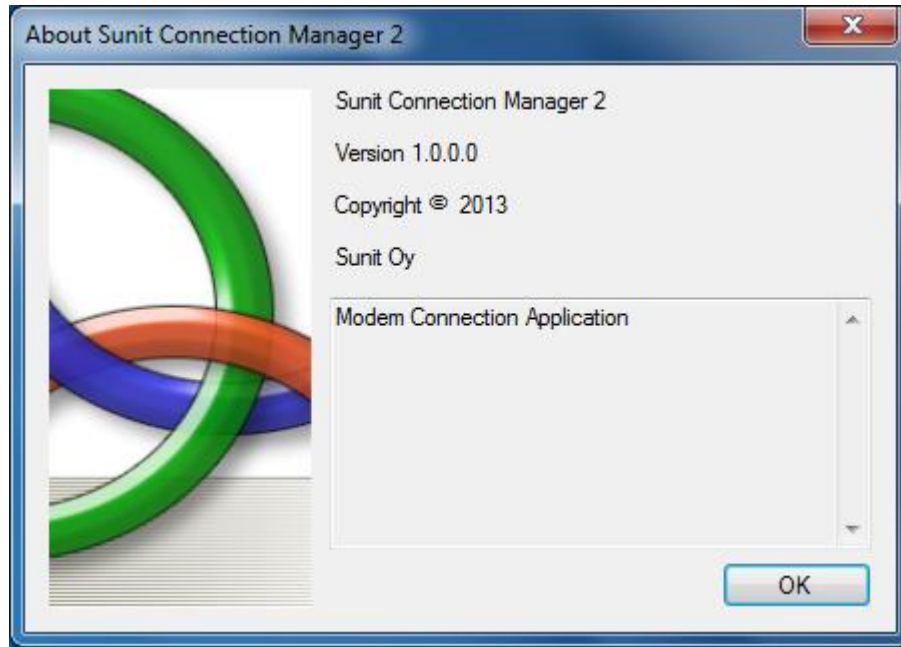
Statistiikkaikkuna toteutettiin omana taulukoituna näkymänä. Näkymään kerätään tiedot nykyisen ja edellisen yhteyden muodostamisten ja katkaisujen ajankohdista sekä ilmoitukset vikatilanteista. Käytössä on staattinen ajastin, joka laskee taustalla olevan yhteyden käynnissä oloa yhden sekunnin välein. Statistiikkanäkymästä tuli kuvan 8 mukainen:



Kuva 8. Statistiikkaikkunan perusnäky

Tapahtumaloki pitää myös lukua poikkeustilanteista ja se jättää muistiin päivätyn merkinnän mahdollisten katkosten ilmetessä. Mikäli yhteyttä ei saada ollenkaan, tähän näkymään tulostuu myös oma AT-komennon kautta saatava virhetieto. Loki tallentuu myös omaan tekstitiedostoon ja statistiikkaikkunassa on myös *Reset*-painike, josta statistiikka on myös mahdollista tyhjentää halutessaan.

Aloituskäytännöstä pääsee vielä pieneen infoikkunaan, jota klikkaamalla ohjelma näyttää omalla About-laatikolla pienen infopakettin ohjelmistosta, versiotiedot sekä hyperlinkin valmistajan kotisivuille. Infoikkunasta tuli kuvan 9 mukainen:



Kuva 9. Infoikkunan perusnäky

### 3.3 Testaus

#### 3.3.1 Laitteistojen tunnistus ja toiminta

Ohjelmaa testattiin vaihe kerrallaan. Testauksessa tietokoneena käytettiin tilaajan omaa fD-mallin ajoneuvoyksikköä, jossa käyttöjärjestelmänä on Windows 7. Ohjelmaa testattiin myös Windows XP-käyttöjärjestelmässä kannettavalla HP-koneella, jolla se myös kirjoitettiin. Havaittiin, että käyttöliittymä ja dll-kirjaston ohjelmat käynnistyvät ja ne ovat toisistaan riippuvaiset onnistuneen toiminnallisuuden edellyttämiseksi.

Ohjelma tunnisti oikean sarjaportin välityksellä Telit GM862-modeemin ja antoi siitä vastee-na seuraavia tietoja, sekä mahdollisuuden suorittaa perustoimintoja:

- Modeemin valmistaja.
- Modeemin malli.
- Vikakoodien tarkkuuden määrittäminen.
- Modeemin nollaus.
- GPS-tieto.

Aluksi tunnistaminen tapahtui muutamalla itse kirjoitetulla yksittäisellä komennolla. Kun useamman modeemitiedon sai samanaikaisesti tulostettua näkyviin, ohjelmaa kokeiltiin myös nettitikussa käytettävällä SIM-kortilla. Kävi ilmi, että ohjelma antoi myös määritellä yhteyteen vaikuttavia asetuksia.

- PIN-koodin tunnistaminen.
- PIN-koodin kyselyn ohittaminen.
- APN-asetukset.
- Operaattoritiedot, maa- ja verkkokoodit.
- Verkkoon rekisteröitymisen tila.

Ohjelmalla yritettiin luoda Internet-yhteys soittamalla valittuun numeroon \*99\*\*\*# ja saada signaalista seuraavia tietoja:

- Yhteyden muodostamisen tunnus CONNECT toiseen modeemiin.
- Signaalin voimakkuus.
- Yhteyden nopeus.
- Yhteyden olemassaolon kesto.

Telit-alusta antoi soiton vasteena NO CARRIER-ilmoituksen, mikä tarkoittaa, että modeemi ei löytänyt kantoaaltoa. On epävarmaa pääseekö Kajaanin ammattikorkeakoulun GM862-alustalla ylipäänsä Internetiin. Ammattikorkeakoulun henkilökunnan mukaan voi olla, ettei kyseisessä testausalustassa tueta vanhan mallin heikoimpia yhteystyyppejä, kuten gsm-yhteyksiä.

Motorola-alustaan ei saatu yhteyttä sarjaportin välityksellä. Internet-yhteys oli kuitenkin mahdollista luoda tällä modeemilla sen USB-portin kautta. Vian epäillään liittyvän baudinopeuden eroavaisuuksiin ja modeemille täytyy yrittää tehdä palautus oletuksiin.

Ohjelma onnistui avaamaan xml-tiedoston ja löytämään tarvittavat elementtien attribuutit valitun modeemin perusteella. Arvojen parsiminen lainausmerkkien ja viittauksen keskeltä tosin tuotti aluksi hieman ongelmia. Kun tarkoitus oli esimerkiksi tulostaa pelkästään attribuutin arvo *Telit*, ohjelma tulosti mukaan myös sen attribuuttiviittauksen *"value = "Telit"*.

Koska solmun mukana oli ylimääräisiä symboleja, jotka häiritsevät modeemin käskyn ymmärtämistä, täytyi tehdä vielä erillinen ohjelman osa, joka erottelee epäolennaiset merkit luetun attribuuttirivin alusta, ja kun kyseessä on at-komento, lisätään loppuun mukaan vielä Enter-merkin tunnus. Lopulta tiedostosta haetun arvon oli mahdollista kulkea ymmärrettävästi sarjaportin läpi modeemille asti.

Kokoonpanoasetusten tallennus ja lataaminen onnistui jokaisella managerin käyttökerralla. Ini-tiedosto luotiin ja ohjelma osasi päivittää siihen muutokset ohjelman sulkiessa. Ohjelma reagoi asetuksiin oikein myös silloin jos ini-tiedosto avattiin sen kohdekansiossa itse ja siihen kirjoitettiin haluttuja tilan arvoja.

### 3.3.2 Yhteyden tilan mittaus

Signaalin voimakkuutta mittaavat muuttujat, jotka ovat riippuvaisia vasteiden tiedoista. Koska kantoaalto-ongelmien vuoksi yhteyttä ei saatu muodostettua, tämä testausalue jäi vajaaksi. On kuitenkin jo esiselvitysvaiheessa ollut tiedossa, kuinka signaalin voimakkuuden kyselykäskyjen tulisi toimia ja millaisia niiden vasteiden tulee olla.

Hieman ei-toivotuksi tilanteeksi muodostui myös esiselvitysvaiheessa se, ettei esimerkiksi 2G:tä voi vaihtaa 3G:ksi yhteyden olemassaolon aikana ilman yhteyden väliaikaista katkaisua. Pientä epätarkkuutta saattoi myös jo silloin ilmetä, koska näiden toimintojen on oltava suorituksessa jatkuvasti - samalla kun samaa reittiä voi kulkea muitakin tietoja. Tähän tulee kehittää jatkokehitysvaiheessa suurempaa huomiota.

### 3.3.3 Lokinpidon todennus

Jo ohjelman kirjoittamisen aikana koodin kriittisiin osa-alueisiin sijoitettiin indikaattoreita, jotka antavat ilmoituksen mahdollisten virheiden ilmestymisestä. Niistä ennakoituja kirjoittamisen aikana testaamia tietoja pystyi käyttämään myös jo rakentamisen aikana hyväksi.

Jos koodin muuttuja, joka on riippuvainen yhdistämisen vasteesta, sai tilan tosi, alkoi yhteyden ajan laskeminen ja lokiin kirjoitus. Loki ilmoittaa päivämäärän, koska tapahtuma on luotu ja koska se lopetettiin. Vaikka yhteyttä ei saatu todennettua, virheilmoitus kantoaallon saamattomuudesta ja merkintä yrityksen kellonajasta näkyy myös lokissa. Laskurit sai myös tyhjennettyä ja näyttämään ajankohdan koska viimeisin nollaus on tapahtunut.

#### 4 YHTEENVETO

Verkkoyhteydenhallinta-ohjelmisto toteutettiin usean eri osa-alueen summana. Haastava työ vaati perehtymistä tietoliikennetekniikkaan, liikkuvan laajakaistan periaatteellisiin toimintoihin sekä usean eri ohjelmointikielen keskinäisten riippuvuuksien hallintaan. Lopputuloksena saatiin toteutettua ohjelmisto, joka tunnistaa sarjaportin välityksellä modeemeja ja kykenee kontrolloimaan modeemin asetuksia. Se myös pyrki luomaan yksinkertaisen modeemiyhteyden, jonka tilaa tarkkaillaan. Ohjelmisto käyttää hyväkseen xml-lukua ja siinä on rajapinta hallinnan RAS-palveluihin. Se myös tallentaa profiiliasetuksia kokoonpanotiedostoon. Ohjelma toimii vaatimusten mukaisesti Windows XP- ja Windows 7-käyttöjärjestelmissä.

Käyttöliittymän rakentamisen kanssa ei oikeastaan ilmennyt suuria ongelmia, joissa lopputulos olisi poikennut alkuperäisestä suunnitelmasta. Sen sijaan ylivoimaisesti ongelmallisin osuus tuli dll-tiedoston sarjaliikenteessä, kun oli kuljetettava merkkijonotietoja eteenpäin sijainnista toiseen parametrien arvosisältöä menettämättä saati niiden yhteentörmäyksiä välttämättä. Haastava vaihe oli myös usean at-komentojen samanaikainen liikennöinti sekä yhteyden vikatilanteiden analysointi. Puskuri täytyi aluksi tyhjentää jokaisen lukukerran jälkeen.

Yhteyden tarkempaan mittaamiseen olisi jälkikäteen ajateltuna voitu varata enemmän tutkimisaikaa ja sarjaportin rengaspuskuriongelman venymisen vuoksi aivan kaikkia vaatimusmääritelmään listattuja kohtia ei saatu laaditun aikatauluun päätökseen asti toteutettua. Projekti aloitettiin jo alkusyksyllä 2012, mutta aikaa kului odotettua enemmän portin vastaanoton käsitteilyn parissa. Yhteyden muodostumista ei ole ohjelmistolla vielä todennettu, mutta edellytykset tälle ovat olemassa ja ne pyritään viemään loppuun.

Solmutietojen lukeminen xml-tiedostosta osoittautui toimivaksi ja itse tiedoston sisältöäkin on vielä tarkoitus laajentaa. Koska testattavaksi laadittu xml-tiedosto on vielä keskeneräinen, osa at-käskyistä on toistaiseksi vielä jätetty kovakoodatun funktiokutsun varaan. Ohjelmassa on myös jo valmiina parametrit, joiden avulla voidaan vaikuttaa tiedoston solmupolun valintaan.

Mahdollisessa seuraavan version kehityksessä ohjelmaan voisi täydentää lisäämällä muun muassa LAN-, WLAN- ja USB-tuen, Proxy-tuen. GPS-paikannuksen kehittäminen ja tuki useammalle kuin yhdelle SIM-kortille ovat myös mahdollisuuksia. Lisäksi статистиikkaa voisi laajentaa mittaamaan esimerkiksi lähetetyn ja vastaanotetun pakettidatan määrää.



## LÄHTEET

- 1 Kuva. Internet to reach its final destination of mobile broadband. Päivitetty 11.4.2013[WWW-dokumentti].  
<http://www.ecotarget.com/technology/internet-technology/evolution-of-internet-to-reach-its-final-destination-of-mobile-broadband>
- 2 Windows XP:n Tietokoneen hallinta -työkalun käyttäminen. Päivitetty 10.3.2013 [WWW-dokumentti].  
<http://support.microsoft.com/kb/308423/fi>.
- 3 MMS-\_ja\_internetasetukset. Päivitetty 26.3.2013[WWW-dokumentti].  
[http://www5.sonera.fi/ohjeet/MMS-\\_ja\\_internetasetukset](http://www5.sonera.fi/ohjeet/MMS-_ja_internetasetukset).
- 4 Pakettikytkentä. Päivitetty 7.4.2013[WWW-dokumentti].  
<http://fi.wikipedia.org/wiki/Pakettikytkent%C3%A4>.
- 5 OSI ja TCP/IP –malli. Päivitetty 12.4.2013[WWW-dokumentti].  
<http://www.krimaka.net/tietotekniikka/verkko-ja-ethernet/osi-ja-tcp-ip-mallit.html>
- 6 Verkonhallintatyökaluja. Päivitetty 12.4.2013 [WWW-dokumentti].  
<http://www.netlab.tkk.fi/julkaisut/tyot/diplomityot/611/tyokalut.html>.
- 7 Modeeminkäytön perusasiat. Päivitetty 19.11.2012[WWW-dokumentti].  
<http://www.helsinki.fi/atk/oppaat/mod/modeemi.html>.
- 8 Matkapuhelin modeemina. Päivitetty 5.4.2013 [WWW-dokumentti].  
[http://linux.fi/wiki/Matkapuhelin\\_modeemina](http://linux.fi/wiki/Matkapuhelin_modeemina).

- 9 Jukka Heino. Tietoliikennetekniikan perusteet. 2009 (Luentomoniste)
- 10 Parhaat AT-komennot. Päivitetty  
1.9.2012 [WWW-dokumentti].  
<http://www.siptune.net/tiki-index.php?page=Parhaat+AT-komennot>.
- 11 Hayes modem AT command set. Päivitetty  
16.3.2013[WWW-dokumentti].  
<http://www.lammertbies.nl/comm/info/hayes-at-commands.html>.
- 12 Kuva. Asynkroninen tiedonsiirto. Päivitetty  
16.3.2013 [ WWW-dokumentti].  
[http://fi.wikipedia.org/wiki/Asynkroninen\\_](http://fi.wikipedia.org/wiki/Asynkroninen_).
- 13 Moduuli. Päivitetty  
16.3.2013 [WWW-dokumentti].  
[http://fi.wikipedia.org/wiki/Moduuli\\_%28ohjelmointi%29](http://fi.wikipedia.org/wiki/Moduuli_%28ohjelmointi%29).
- 14 Modulaarinen ohjelmointi. Päivitetty  
16.3.2013 [WWW-dokumentti].  
[http://cs.stadia.fi/~silas/ohjelmointi/c\\_opas-Modulaar.html](http://cs.stadia.fi/~silas/ohjelmointi/c_opas-Modulaar.html).
- 15 Ari Vesanen, Tietojenkäsittelytieteiden laitos, Oulun yliopisto Windows-ohjelmointi 2008 – 2009, DLL:t  
3.3.2013[pdf-dokumentti].  
[http://www.tol.oulu.fi/users/ari.vesanen/windowsohjelmointi/Luennot/kalvot/WO\\_DLL.pdf](http://www.tol.oulu.fi/users/ari.vesanen/windowsohjelmointi/Luennot/kalvot/WO_DLL.pdf).
- 16 XML. Päivitetty  
28.1.2013 [WWW-dokumentti].  
<http://fi.wikipedia.org/wiki/XML>.
- 17 XML Tutorial. Päivitetty  
14.2.2013 [WWW-dokumentti].  
<http://www.w3schools.com/xml/>.

18 XML Path Language (XPath) Version 1.0. Päivitetty  
4.3.2013 [WWW-dokumentti].  
<http://www.w3.org/TR/xpath/>



