

LAPPEENRANNAN TEKNILLINEN YLIOPISTO
TIETOTEKNIIKAN OSASTO

HALLINTASOVELLUS M2M-TIEDONSIIRTOON GPRS-VERKOSSA

Diplomityön aihe on hyväksytty Tietotekniikan osaston osastoneuvostossa
5.5.2004.

Työn tarkastajina toimivat professori Jari Porras ja DI Anna-Maria Kähkönen.
Ohjaajana toimi DI Anna-Maria Kähkönen.

Oulussa 24.11.2005

Teemu Toivola
Kauppurienkatu 32 A 3
90100 Oulu
+358 40 7388771

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Tietotekniikan osasto

Teemu Toivola

Hallintasovellus M2M-tiedonsiirtoon GPRS-verkossa

Diplomityö

2005

59 sivua, 20 kuvaa ja 2 taulukkoa

Tarkastajat: Professori Jari Porras
DI Anna-Maria Kähkönen

Hakusanat: M2M, konekommunikaatio, GPRS, CORBA
Keywords: M2M, machine communication, GPRS, CORBA

Pääosa matkapuhelinjärjestelmien liikenteestä on toistaiseksi puhetta. Dataliikenteen osuus on kuitenkin jatkuvasti kasvamassa uusien tekniikoiden myötävaikutuksella. Langattoman dataliikenteen kehittämiseksi tuo myös lisäpainetta internetin nopeasti yleistynyt käyttö. Eräs lupaavimmista sovellusalueista on konekommunikaatio, M2M, minkä läpilyöntiä on ennustettu jo vuosia.

Viime vuosina markkinoille tulleet M2M-laitteet tarjoavat aiemmista laitteista poiketen myös mahdollisuuden sovellusten suorittamiseen itse laitteessa ja niiden päivittämisen langattomasti. Työssä vertaillaan kolmea markkinoilla olevaa konekommunikaatioon soveltuvaa laitetta ja tarkastellaan niiden ominaisuuksia ja sovellusten päivitettävyyttä GPRS-verkkoa käyttäen. Tarkastelussa havaitaan, että teleoperaattoreiden asettamat rajoitukset ja käytössä olevat päivitysmenetelmät aiheuttavat tiettyjä ongelmia päivitysten käytettävyydelle.

Työn tuloksena kehitettiin hallintasovellusohjelmisto, mikä mahdollisti etäällä sijaitsevien M2M-laitteiden sovellusten päivittämisen helppokäyttöisen käyttöliittymän avulla. Hallintasovellusta käyttäen useita, maantieteellisesti hajallaan olevia laitteita oli mahdollista päivittää samanaikaisesti automatisoidusti. Hallintasovellus osoittautui käyttökelpoiseksi työkaluksi M2M-laitteiden sovellusten hallintaan jo, kun laitemäärä kasvoi muutamista kappaleista muutamaan kymmeneen.

ABSTRACT

Lappeenranta University of Technology
Department of Information Technology

Teemu Toivola

Control application for M2M data transfer in GPRS network

Thesis for the Degree of Master of Science in Technology

2005

59 pages, 20 figures and 2 tables

Examiners: Professor Jari Porras
M.Sc. Anna-Maria Kähkönen

Keywords: M2M, machine communication, GPRS, CORBA

Currently most of the traffic in mobile networks is voice. The percentage of data traffic is however constantly rising as new technologies offer more capacity than the previous wireless networks. Ever increasing usage of internet also motivates the developing of wireless data communication. Machine to machine communication or M2M is one of the most promising new application areas. A break through in that area has already been predicted for years.

During recent years new M2M devices that have hit the market differ from the previous products by offering features that support running software in the device itself and upgrading that software using a wireless connection. This thesis compares the features and update possibilities of three M2M devices currently on the market. The comparison shows that restrictions set by mobile service providers and available update methods cause certain problems that weaken the usability of the software update features provided by the devices.

This thesis produced a control application software that made possible to update applications to remote M2M devices using an easy-to-use user interface. The control application enabled also automatic updating of multiple devices on dispersed locations simultaneously. The control application proved to be a useful tool for controlling applications of M2M devices already when the number of devices rose from few items to tens of devices.

ALKUSANAT

Tämä työ on tehty Oulussa Sofnetix Oy:ssä Lappeenrannan teknillisen yliopiston tietotekniikan osastolle. Haluan kiittää tämän diplomityön tarkastajia, Jari Porrasta ja Anna-Maria Kähköstä, heidän opastuksesta ja neuvoista tätä työtä tehdessäni. Haluan myös kiittää työnantajaani mahdollisuudesta diplomityön tekemiseen ja miellyttävästä työilmapiiristä.

Lopuksi vielä kiitokset vanhemmilleni ja suvulle työn valmistumisen kärsivällisestä odottamisesta. Kyllä tästä viimein valmista tuli.

Oulussa, 24. marraskuuta 2005

Teemu Toivola

SISÄLLYSLUETTELO

1	JOHDANTO	4
1.1	Työn tausta	4
1.2	Tavoitteet ja rajausta.....	5
1.3	Työn rakenne	6
2	KONEKOMMUNIKAATIO.....	7
2.1	Määritelmä.....	7
2.2	Sovelluskohteet.....	8
2.3	Järjestelmän tyypillinen rakenne	9
3	TEKNOLOGIA	11
3.1	M2M-laitteet.....	11
3.1.1	Siemens TC45.....	11
3.1.2	Sony Ericsson GR47.....	13
3.1.3	Nokia 12	14
3.1.4	Ominaisuusvertailu.....	16
3.2	Langattomat päivitysmahdollisuudet.....	18
3.2.1	Piirikytkentäinen yhteys	19
3.2.2	Hypertext Transfer Protocol.....	20
3.2.3	Common Object Request Broker Architecture.....	22
3.2.4	SyncML	24
3.2.5	GPRS	25
3.2.6	Mobile IP	26
3.2.7	VPN	26
3.2.8	Yksityinen APN.....	26
4	TARPEET	27
4.1	Tiedonsiirto.....	27
4.2	Hallittavuus.....	29
5	HALLINTASOVELLUS	31
5.1	Keskeiset vaatimukset	32
5.1.1	Laitehallinta	32
5.1.2	Sovellusten hallinta.....	33
5.1.3	Vaatimukset toteutukselle.....	33
5.2	Toiminnallisuus	34
5.3	Hallintasovelluksen rakenne.....	35
5.3.1	Tietokanta	37
5.3.2	Käyttöliittymä.....	39
5.3.3	Ydin	43
6	HALLINTASOVELLUKSEN TESTAUS GPRS-VERKOSSA	46
6.1	Testiympäristö	46
6.2	Testien sisältö	48
6.3	Testauksen tuloksia.....	49
6.3.1	Operaattorit.....	49

6.3.2	CORBA	50
6.3.3	Sovellukset.....	51
6.3.4	Virhetilanteista toipuminen	53
6.3.5	Suorituskyky	54
7	YHTEENVETO.....	56
	LÄHTEET	58

LYHENTEET

AMR	Adaptive Multi-Rate
APN	Access Point Name
CORBA	Common Object Request Broker Architecture
CSD	Circuit Switched Data
EDGE	Enhanced Data rate for GSM Evolution
GIOP	General Inter-ORB Protocol
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
HTTP	Hypertext Transfer Protocol
HSCSD	High Speed Circuit Switched Data
I ² C	Inter Integrated Circuit
IMP	Information Module Profile
IP	Internet Protocol
J2ME	Java 2 Micro Edition
M2M	Machine-to-Machine
NAT	Network Address Translation
N12	Nokia 12
ORB	Object Request Broker
OTAP	Over the Air Provisioning
SMSC	Short Message Service Center
TCP	Transmission Control Protocol
UMTS	Universal Mobile Telecommunications System
VPN	Virtual Private Network
WCDMA	Wideband Code Division Multiple Access

1 JOHDANTO

1.1 Työn tausta

Matkapuhelinjärjestelmissä pääosa liikenteestä on vielä puhetta. Dataliikenteen osuuden matkapuhelinverkoissa on kuitenkin ennustettu kasvavan voimakkaasti. Osaltaan tähän on vaikuttanut 1990-luvun loppupuolella yleistyneet tekstiviestit ja 2000-luvun alussa käyttöön otettu GPRS-tekniikka, joka mahdollistaa pakettipohjaisen dataliikenteen matkapuhelinjärjestelmissä. Kolmannen sukupolven matkapuhelinjärjestelmät kuten WCDMA (UMTS) on suunniteltu nimenomaan kasvavaa datansiirtotarvetta silmällä pitäen. Internetin nopeasti yleistynyt käyttö tuo lisäpainetta langattoman dataliikenteen kehittämislle.

Matkapuhelinverkkojen kehittyminen mahdollistaa uusia sovellusalueita. Eräs lupaavimmista sovellusalueista on konekommunikaatio, M2M. M2M-laitteiden läpilyöntiä on ennustettu jo vuosia. Aiempien sukupolvien laitteet ovat kuitenkin käytännössä lähinnä vain matkapuhelinverkkoa käyttäviä modeemeja, jolloin pelkän laitteen käyttömahdollisuudet ilman erillistä sovellusta suorittavaa sulautettua laitetta ovat hyvin rajalliset.

Viime vuosina markkinoille on ilmestynyt erityisesti langattomaan konekommunikaatioon soveltuvia laiteratkaisuja. Nämä laiteratkaisut tarjoavat tavallisia matkapuhelimia vastaavat yhteyksmahdollisuudet, mutta ovat rakenteeltaan pelkistetympiä, sillä ne on etupäässä suunniteltu toimimaan itsenäisesti ilman käyttäjän ohjausta tai valvontaa. Syksyllä 2003 Nokia toi markkinoille oman M2M-moduulin, mikä mahdollistaa aiemmista malleista poiketen myös J2ME-pohjaisten sovellusten suorittamisen itse moduulissa tarjoten täten aivan uudenlaisia käyttömahdollisuuksia moduulille. J2ME-sovellusten käyttö avaa myös mahdollisuuden etäpäivittää langattomasti uusi ohjelmistoversio moduuleille. Alun perin Nokian oli tarkoitus tarjota moduulin lisäksi teleoperaattoreille yhdyskäytäväratkaisu, mikä olisi mahdollistanut keskitetyn hallinnan moduuleille. Ratkaisu kuitenkin hylättiin, minkä seurauksena moduulille tarjottiin ainoastaan hyvin pelkistetty manuaalisesti käytettävä yhden

moduulin kerrallaan päivittävä ohjelma. Konekommunikaatio-sovellukset yleensä sisältävät kymmeniä jopa satoja moduuleita, joten manuaalisesti tapahtuva moduulien yksittäispäivitys on aikaa vievää ja turhauttavaa.

Tässä työssä kehitettiin monipuolinen ja helppokäyttöinen ohjelmisto moduulien etähallintaan GPRS-verkon yli. Koska hallinta tapahtuu yleensä yrityksen sisäisestä verkosta julkisen internetin kautta matkapuhelinverkkoon, nousee ongelmaksi verkoissa olevat palomuurit.

Työn aikana kehitetty sovellus on tarkoitettu korvaamaan Nokian M2M-moduulilleen tarjoama hallintasovellus, joka oli ominaisuuksiltaan vaatimaton automatisoidun konekommunikaation vaatimuksiin nähden.

Tämä työ tehtiin Sofnetix Oy:lle syksyn 2004 aikana. Työn kirjoittamisen aikana Nokian M2M-moduulin valmistus ja markkinointi siirtyi Aplicom Oy:lle. Moduuli itsessään säilyi nimeä lukuun ottamatta samana. Työssä kuitenkin puhutaan Nokian moduulista, sillä hallintasovelluksen kehitys tehtiin sitä käyttäen.

1.2 Tavoitteet ja raja

Työn tavoitteena on selvittää M2M-laitteiden sovellusten päivittäminen ja hallinta langattomassa verkossa sekä teoreettisesti että kokeellisesti. Työ rajataan koskemaan vain GSM/GPRS verkkoa ja sen tarjoamia tietoliikennetkaisuja. Avointen (julkisen internet) ja suljettujen (yritysten sisäisten) verkkojen välillä on yleensä palomureja ja työssä tarkastellaan erilaisia tiedonsiirtotkaisuja päivitystarpeen hoitamiseksi.

Lisäksi tavoitteena on tuottaa ensimmäinen prototyyppi käyttäjäystävällisestä hallintatyökalusta, jonka avulla voidaan hoitaa automaattisesti M2M-laitteiden hallinta- ja ylläpitorutiinit kuten uusien ohjelmistoversioiden lataus ja ohjelmiston käynnistys. Usean moduulin samanaikainen päivitys tulisi olla mahdollista

automaattisesti ilman, että käyttäjän tarvitsee huolehtia esimerkiksi uudelleenyrityksistä päivityksen epäonnistuesssa. Vaatimuksena on, että hallintasovellus olisi rakenteeltaan mahdollisimman modulaarinen, jotta siihen olisi mahdollista lisätä tuki useiden eri valmistajien laitteille ilman, että käyttäjän tarvitsisi olla tietoinen valmistajakohtaisista eroista esimerkiksi yhteyden muodostukseen tai kommunikointiin liittyen. Tuotettava M2M-sovellusten hallintatyökalun prototyyppi toteutetaan erityisesti Nokia 12 –moduulille sopivaksi.

Työssä ei oteta kantaa itse sovellusten suunnitteluun, sillä laitteiden tarjoamat kehitysympäristöt ja niissä käytetyt ohjelmointikielet vaihtelevat eri valmistajien välillä. Käsiteltävät laitteet on rajattu laitteisiin, jotka tarjoavat mahdollisuuden sovelluksen suorittamiseen itse laitteessa ilman ulkopuolista prosessoria tai kytkentää ja joissa sovellus on vaihdettavissa tai päivitettävissä langattomasti. Kyseisen kaltaiset laitteet, missä laite kykenee itsenäisesti suorittamaan entistä monipuolisempia toimintoja, edustavat nykyistä M2M-laitteiden kehityssuuntaa.

1.3 Työn rakenne

Työ alkaa johdannolla konekommunikaation ja sen sovelluskohteisiin esimerkkien kanssa. Kappale 3 esittelee ja vertailee GSM-tekniikkaa käyttäviä M2M-laitteita ja käy läpi eri menetelmiä, joilla laitteisiin voidaan sovelluksia päivittää langattomasti. Kappale 4 erittelee yrityskäyttäjän kannalta konekommunikaation kohdistuvat odotukset ja tarpeet. Näiden tarpeiden perusteella määritellään hallintasovelluksen toimivuuden ja ominaisuuksien vaatimukset. Hallintasovellus on kuvattu kappaleessa 5 ja siihen liittyvä testaus kappaleessa 6.

2 KONEKOMMUNIKAATIO

2.1 Määritelmä

M2M, konekommunikaatio, käsitetään yleensä automaattisesti suoritettuna tiedonsiirtona kahden tai useamman laitteen välillä. Termi ei ota kantaa siihen, mitä siirtotietä tai protokollaa tiedonsiirtoon käytetään tai mitä tietoa siirretään. Tiedonsiirto voi olla esimerkiksi datan tai asetusten siirtämistä paikasta toiseen. M2M-termiä käytetään yleensä laitteiden, järjestelmien tai koneiden välisestä kommunikaatiosta puhuttaessa, mutta se ei sulje pois ihmisen roolia järjestelmän käyttäjänä tai sen osana. M2M laajennetaan yleensä englannin kielisiksi termeiksi ‘machine-to-machine’, ‘man-to-machine’ tai ‘machine-to-man’.

Yksinkertaistettuna M2M-tekniikan voidaan esittää muodostuvan telematiikasta ja telemetriasta. Telematiikka tarkoittaa laitteiden etähallintaa. Etähallinnan avulla laitteille voidaan asettaa uusia ohjelmaversioita tai muuttaa laitteen toimintaparametreja kuten raportointitajuuutta tai hälytysraja-arvoja. Telemetrialla taas tarkoitetaan laitteista saatavan tiedon keräämistä ja seuranta. [1]

Usein M2M-kommunikaatioon pohjautuvan järjestelmän laitteet sijaitsevat maantieteellisesti etäällä laitteita valvovasta ja hallinnoivasta tietojärjestelmästä. Laitteiden sijainnilla ei ole varsinaista merkitystä, mikäli yhteys laitteisiin muodostetaan langattoman verkon kautta. Järjestelmän kannalta ei täten ole merkitystä, onko laite kiinteästi tiettyyn paikkaan asennettu vai liikkuva alusta.

2.2 Sovelluskohteet

M2M-teknologia on ratkaisu langattomiin tai langallisiin tietoliikenne- ja liiketoimintasovelluksiin, joissa ihmisen läsnäolo koneen tai laitteen vieressä tehdään tarpeettomaksi. Yleensä tämä tarkoittaa toistuvien tapahtumien automatisoimista. Nimike ei ota kantaa laitteiden väliseen siirtomediaan tai käytettyyn protokollaan.

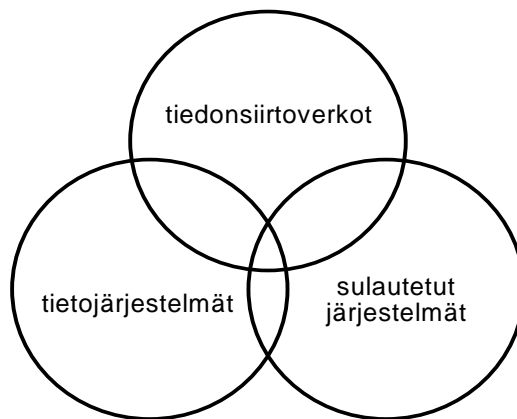
Tyypillisesti M2M-järjestelmään toteutettujen sovellutusten tarkoituksena on mitata, raportoida, monitoroida ja käsitellä tietoa sekä välittää sitä muihin tietojärjestelmiin tai vastaanottaa muista tietojärjestelmistä. M2M-järjestelmään liitetyt laitteet voivat esimerkiksi ohjata erilaisia prosesseja, kontrolloida muita laitteita, lukea sensoritietoa, monitoroida hälytysrajoja ja lähettää käsiteltyä tietoa halutussa muodossa eteenpäin. M2M-järjestelmä toimii tarvittaessa myös siltana aiemmin toteutettujen järjestelmien välillä. Tämä mahdollistaa erillisten järjestelmien kokoamisen yhdeksi yhtenäisemmäksi järjestelmäksi.

Hyvä esimerkki toteutetusta M2M-järjestelmästä on Vattenfallin toteutus, jossa asiakkaiden sähkömittareita etäluetaan M2M-laitteita käyttäen. Järjestelmässä nykyaikaiseen sähkömittariin kytketään rinnalle GSM-moduuli, jonka avulla sähkömittarin tiedot saadaan siirtymään automaattisesti säännöllisin väliajoin tietokantaan. Vattenfallin toteutuksessa laitteiden hallinta on tosin luovutettu teleoperaattorille, joten Vattenfall käytännössä vain ostaa luettujen sähkömittareiden tiedot ja teleoperaattorin velvollisuudeksi jää järjestelmän ylläpito ja hallinta. [2]

M2M-järjestelmät soveltuvat myös liikkuvan kaluston hallinnoimiseen. Teboilin käyttämä järjestelmä mahdollistaa säiliöautojen lähes reaaliaikaisen seurannan GPS-paikannuksen avulla ja tarjoaa kuljettajille järjestelmän kautta automaattisesti tiedot päivän kuormauksesta, toimituspaikoista, ajoreiteistä ja mahdollisista muutoksista. Tilausten käsittelyjärjestelmä on myös integroitu mukaan, joten tilausten tiedot päivittyvät niiden käsittelyn edistyessä. [3]

2.3 Järjestelmän tyypillinen rakenne

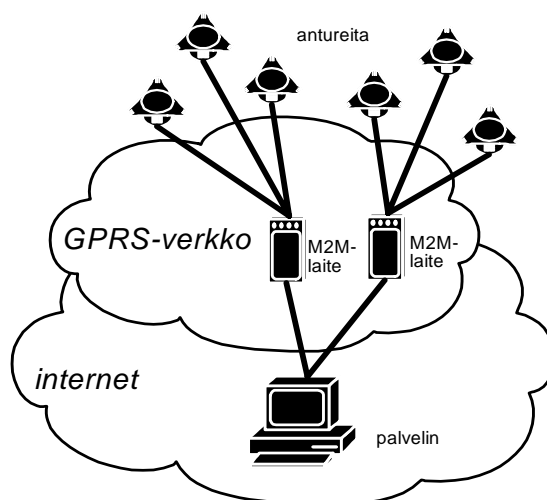
M2M-järjestelmä koostuu verkotettavista laitteista, koneista tai automaateista, eri järjestelmäkomponenttien välisistä tietoliikenneyhteyksistä, laitekannan hallinta- ja valvontajärjestelmästä sekä asiakaskohtaisesta M2M-sovelluksesta. Yleensä M2M-järjestelmä integroidaan osaksi yrityksen muita järjestelmiä tai se toimii eräänlaisena siltana eri järjestelmien välillä. M2M-ratkaisuissa yhdistyy kuvan 1 tapaan kolme eri tietotekniikan osa-alueita. [1]



Kuva 1: M2M-ratkaisujen osa-alueet.

Matkapuhelinverkkoa käytettäessä konekommunikaatio on perinteisesti toteutettu tekstiviesteillä silloin, kun siirrettävä tietomäärä on pieni eikä latenssilla ole suurta merkitystä. Tekstiviestin kokorajoituksen takia suurempia tietomääriä siirrettäessä data joudutaan jakamaan useampaan tekstiviestiin. Verkko ei kuitenkaan takaa viestien saapumista lähetyksessä eikä viive lähetyksen ja vastaanoton välillä ole ennakoitavissa. Tällöin suuremman tietomäärän kohdalla joudutaan joko turvautumaan datapuheluun tai langalliseen verkkoon. Konekommunikaation käyttökohteena ovat yleensä erilaisten teollisuuden anturitietojen siirtäminen lukulaitteelta keskitettyyn tietokantaan siirron tapahtuessa joko säännöllisin väliajoin tai ulkopuolisen pyynnön seurauksena.

Kuvassa 2 on esitetty eräs M2M-järjestelmän tyypillinen rakenne. Joukko antureita on yhdistetty M2M-laitteeseen, joka vastaa antureiden lukemisesta ja tietojen keräämisestä paikallisella tasolla. Eri paikoissa sijaitsevat moduulit puolestaan kommunikoivat GSM-verkon välityksellä internetin kautta palvelimen kanssa. Palvelimeen kerätään moduulien keräämä anturitieto, joka voidaan tällöin tallentaa keskitetysti yhteen paikkaan.



Kuva 2: Eräs M2M-järjestelmän tyypillinen rakenne.

3 TEKNOLOGIA

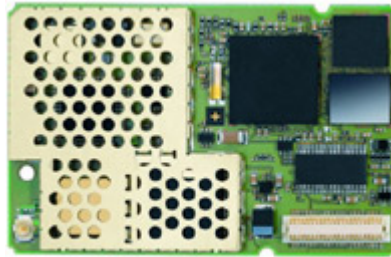
Konekommunikaatio pyrkii hyödyntämään olemassa olevia teknologioita, sillä standardeja käyttämällä laitteiden sovittaminen vanhojen järjestelmien rinnalle on helpompaa. Tämä myös helpottaa vaiheittaista siirtymistä vanhasta järjestelmästä uuteen. Teknologian osalta konekommunikaatio voidaan jakaa itse laitteisiin ja niiden käyttämiin kommunikaatiotapoihin. Kappale 3.1 keskittyy esittelemään ja vertailemaan GSM-verkossa toimivia M2M-laitteita ja kappale 3.2 selvittää näiden laitteiden mahdollisia päivitysmenetelmiä.

3.1 M2M-laitteet

Konekommunikaatioon soveltuvien laitteiden ominaisuudet riippuvat käyttötarkoituksesta ja käyttöympäristön vaatimuksista. Yksinkertaisimmillaan jopa sarjaportin avulla viestivää anturia voidaan periaatteessa kutsua M2M-laitteeksi. Tämä työ kuitenkin keskittyy sellaisiin laitteisiin, jotka tarjoavat mahdollisuuden sovellusten suorittamiseen itse laitteessa, joten seuraavissa kappaleissa on esitelty ainoastaan kyseiset vaatimukset täyttäviä laitteita.

3.1.1 Siemens TC45

Helmikuussa 2003 Siemens julkaisi ensimmäisenä M2M-laitevalmistajista moduulin, mikä mahdollisti Java-sovellusten suorittamisen itse laitteessa ilman tarvetta ulkopuoliselle suorittimelle. Aiemmat mallit olivat toimineet lähinnä tekstiviestien tai sarjaportin kautta ohjattavina GSM-modeemeina. TC45 (kuva 3) kuitenkin sisältää Java-sovellusten suorittamisen lisäksi tuen GPRS:lle ja integroidun TCP/IP-pinon, mikä mahdollistaa TCP-yhteyksien muodostamisen Java-sovelluksista käsin.



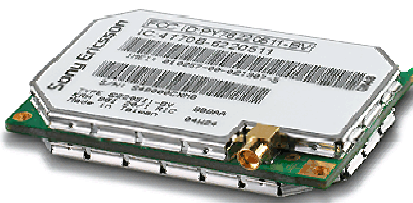
Kuva 3: Siemens TC45.

TC45:ssä on varattuna sovelluksille 300 kilotavua muistia, jota sovellukset voivat käyttää joko tallennustilana tai ajonaikaiseen suoritukseen. Moduuli kuitenkin rajoittaa käynnissä olevien sovellusten määrän yhteen. Moduulille siirrettävien sovellusten tulee olla Javan IMP 1.0:n ja TC45:n määritysten mukaisia. Sovellusten siirtäminen laitteelle tapahtuu Sunin OTAP-menettelyä muistuttavalla tavalla, missä moduuli hakee määrätyn sovelluksen HTTP-palvelimelta. [4] Päivitysmenetelmä on tarkemmin esitelty kappaleessa 3.2.2. Päivitysmenetelmä mahdollistaa sovelluksen siirtämisen, poistamisen, pysäyttämisen ja käynnistämisen. Kaikki muu mahdollinen sovelluksen toimintaan liittyvä asetusten määrittäminen pitää olla toteutettuna itse sovelluksessa. Sovellusten määrää rajoittaa vain moduulin vapaana olevan muistin määrä, sillä sovellukset erotellaan toisistaan tiedostonimen perusteella. Moduuli mahdollistaa tarvittaessa sovelluksen automaattisen käynnistyksen heti moduulin käynnistyttyä. [5]

Moduuli tarjoaa sovelluksille TCP-yhteyksien lisäksi myös mahdollisuuden tavallisten matkapuhelinverkkojen palveluiden, kuten tekstiviestien ja datapuheluiden, käyttöön ja useamman I/O-portin, joiden kautta sovellus voi kommunikoida muiden moduuliin kytkettyjen laitteiden kanssa. I/O-portteja voidaan käyttää niputettuna tavallisena sarjaporttina, mikä esimerkiksi mahdollistaa verkkoyhteyden vanhemmalle ulkopuoliselle anturille. Moduulin hallitseminen Java-sovellusten lisäksi on myös mahdollista käyttämällä AT-komentoja sarjaportin kautta. [6]

3.1.2 Sony Ericsson GR47

Sony Ericsson julkaisi oman sovellusten suorittamiseen pystyvän M2M-moduulinsa maaliskuussa 2003 toisena laitevalmistajana Siemensin jälkeen. [7] Ominaisuuksien puolesta GR47 (kuva 4) on hyvin pitkälti vastaava kuin TC45. Sovelluksille on tarjolla TCP/IP-pino GPRS-yhteyttä hyödyntäen ja laite tarjoaa useita liitännätapoja ulkopuolisille laitteille. Sovellusrajapinnasta löytyy myös valmiita kirjastoja lisälaitteiden, kuten numeronäppäimistön ja nestekidenäytön, hallintaan.



Kuva 4: Sony Ericsson GR47.

Muihin M2M-moduuleihin verrattuna GR47:n poikkeavuus on siinä käytetty ohjelmointikieli. Javan sijasta Sony Ericsson on valinnut moduuliin heidän oman syntaksiltaan C:n kaltaisen tulkatun skriptikielen. Skriptikielen käyttäminen toisaalta selittää moduulin vähäisen muistimäärän muihin vastaaviin laitteisiin verrattuna, sillä sovellus vie tilaa vain koodirivien verran. Tällä tavoin 44 kilotavua riittää monimutkaisemmallekin ohjelmalle. Moduuli tosin pystyy säilömään kerralla vain kaksi sovellusta ja niitäkin kutsutaan muistipaikkojen 0 ja 1 avulla. Vain toinen sovelluksista voi olla käynnissä. Tilarajoitusten takia tulkki kuitenkin tarjoaa valmiit kirjastot yleisimmille lisälaitteille, kuten MBus- ja I²C-protokollia käyttäville laitteille. [8]

Moduulin tulkki toimii kahdessa tilassa, joiden välillä sovellus voi itse tarvittaessa siirtyä. Perustilassa sovelluksella on käytettävissä kaikki moduulin tarjoamat tietoliikenneyhteydet, mutta saatavilla oleva käskykanta on rajoittunut. Tulkin suoritusnopeus ei myöskään ole ennakoitavissa, vaan se on riippuvainen verkkorajapinnan käytöstä. Mikäli moduuli joutuu keskustelemaan verkon kanssa,

hidastuu sovellusten suorittaminen tänä aikana. Tästä johtuen tulkki tarjoaa sovelluksille toisen tilan, jota käytettäessä koko tulkin käskykanta on käytettävissä, mutta verkkoyhteys on tänä aikana sammutettuna. Tässä tilassa ollessaan sovelluksen vasteajat ovat kuitenkin huomattavasti helpommin ennustettavissa, joten se soveltuu paremmin esimerkiksi reaaliaikaiseen mittaamiseen. Sovellus voi tallentaa haluamansa tiedot moduulin muistiin, suorituksen jälkeen siirtyä perustilaan ja lähettää kerätyt tiedot verkossa olevalle palvelimelle.

GR47:n sovellukset ovat päivitettävissä joko sarjaportin kautta tehtävällä siirrolla tai datapuhelulla. Päivitys käynnistetään AT-komennolla, minkä jälkeen siirto tapahtuu Xmodem-protokollaa käyttäen. Xmodem-protokolla mahdollistaa binääridatan siirtämisen terminaalikyhteyden yli. Siirretty sovellus korvaa pysäytettynä olleen sovelluksen ja on käynnistettävissä toisella AT-komennolla onnistuneen siirron jälkeen. [9]

3.1.3 Nokia 12

Kesäkuussa 2003 Nokia julkaisi Sony Ericssonin jälkeen Nokia 12 -nimisen M2M-moduulin (kuva 5). [10] Liitännämahdollisuuksiltaan Nokia 12 on Siemensin ja Sony Ericssonin laitteita vastaava, tarjoten useita analogisia ja digitaalisia I/O-portteja lisälaitteiden kanssa kommunikointiin. Moduuli pystyy niin ikään suorittamaan Java-sovelluksia Siemensin TC45:n tapaan. Java-sovellukset eivät tosin ole moduulien välillä täysin yhteensopivia, vaikka ne käyttävätkin samaa Java IMP 1.0 -määrittystä. [11]



Kuva 5: Nokia 12.

Nokia 12 tarjoaa sovellusten tallennustilaa 1024 kilotavua, eikä moduuliin siirrettyjen sovellusten määrällä ole vapaan tilan lisäksi muita rajoituksia. Jokainen sovelluspaketti voi sisältää useamman sovelluksen, mikä mahdollistaa sovellusten ryhmittelyn ja yhteisten kirjastojen käyttämisen. Sovelluksille on tarjolla TCP/IP-pino, mikä mahdollistaa muun muassa TCP-yhteyksien avaamisen GPRS-yhteyttä käyttäen. Nokian tarjoama Java-tulkki sisältää myös integroituna kirjaston, minkä avulla sovellus voi käyttää sarjaporttiin liitettyä GPS-laitetta paikannukseen.

Sovellusten päivittäminen on mahdollista sarjaporttiyhteydellä tai langattomasti GPRS- ja CSD-yhteyksien avulla. Langattomassa päivityksessä moduuli herätetään tekstiviestillä, minkä jälkeen päivitys ja muu laitteen hallinta on mahdollista toteuttaa CORBA-yhteyttä käyttäen. Päivitysmenetelmä on esitelty tarkemmin kappaleessa 3.2.3. Muista laitteista poiketen Nokian toteutus mahdollistaa myös laitteen asetusten muokkaamisen päivitysyhteyden avulla. Moduuli tarjoaa myös yksinkertaisen tekstiviestipohjaisen käyttöliittymän, mikäli Java-sovelluksen käyttämistä ei nähdä tarpeelliseksi.

Nokia on julkaissut moduulista myös parannelun N12i -mallin. Muutokset kuitenkin koskevat lähinnä laitteen verkkorajapintaa ja käyttäjälle tämä näkyy hieman nopeampana GPRS- ja EDGE-yhteytenä. Ainoana lisäominaisuutena on tuki AMR-audiokoodaukselle. Muiden ominaisuuksien osalta uusi malli vastaa täysin vanhempaa N12-mallia.

3.1.4 Ominaisuusvertailu

Taulukkoon 1 on kerätty edellä esiteltyjen moduulien eroavat ominaisuudet vertailun helpottamiseksi. Nokian moduulista on taulukossa esitettynä vain vanhempi N12-malli.

Taulukko 1: Moduulien ominaisuusvertailu.

	Siemens TC45	Sony Ericsson GR47	Nokia 12
Moduulin koko	53 x 34 x 3.5 mm	50 x 33 x 7.2 mm	45 x 36 x 9 mm
Paino	10 g	18.5 g	15 g
Sovelluskieli	Java IMP 1.0	tulkattu C	Java IMP 1.0
Muistin koko	300 kt	44 kt	1024 kt
GPRS-luokka	8	8	6
CSD	•	•	•
HSCSD		•	•
SMS	•	•	•
GPRS	•	•	•
EDGE			•

Fyysisen koon ja painon suhteen moduuleissa ei ole merkittäviä eroja. Siemensin laite on kuitenkin selvästi kevyempi Sony Ericssoniin verrattuna. Painolla ei kuitenkaan ole yleensä suurta merkitystä M2M-sovelluksissa. Langattomien yhteyksien osalta Nokia 12 tarjoaa parhaimmat yhteismahdollisuudet, sillä se on ainoa EDGE:n hallitseva moduuli. Kaikki moduulit ovat niin sanottuja kaksitaajuuksilaitteita, joten moduulin samaa versiota ei ole mahdollista käyttää eurooppalaisissa ja amerikkalaisissa verkoissa. Tästä syystä jokainen valmistaja tarjoaa laitteesta kahta eri versiota käytetyn verkon taajuualueesta riippuen.

Nokia on muista valmistajista poiketen valinnut moduulin käyttämään GPRS-luokkaa 6. Tämä ei kuitenkaan vaikuta merkittävästi moduulin tiedonsiirtonopeuteen. Luokkien 6 ja 8 välinen ero on siinä, että luokka 8 käyttää vastaanottamiseen neljää aikaviipaletta ja lähettämiseen yhtä. Vastaavasti luokka 6 mahdollistaa lähetyksessä ja vastaanotossa käytettävän aikaviipalemäärän

valitsemisen kahdeksi tai kolmeksi. Molemmissa luokissa on kuitenkin maksimissaan viisi aikaviipaleta samanaikaisesti käytössä. Konekommunikaatiossa nopeammalle lähetysnopeudelle voi tulla käyttöä, mikäli moduulilta siirretään usein suuria datamääriä verkon yli.

Merkittävin ero moduulien välillä ilmenee sovelluskieltä vertailtaessa. Nokia ja Siemens käyttävät matkapuhelimissa jo pitempään käytettyä J2ME-toteutusta, kun taas Sony Ericsson on valinnut moduuliinsa oman kappaleessa 3.1.2 esitellyn C:n kaltaisen skriptikielen. Vaikka Nokia ja Siemens käyttävätkin moduuleissaan samaa sovelluskieltä, eivät toiselle moduulille kirjoitetut sovellukset ole keskenään siirrettäviä. Käytännössä tämä tarkoittaa, että Nokia 12 sovellusta ei voi suoraan siirtää Siemens moduuliin tai päinvastoin. Syynä tähän on moduulien alustakohtaisten kirjastojen eroavaisuudet. Sony Ericssonin moduulissaan käyttämä skriptikieli puolestaan pakottaa kirjoittamaan sovellukset uudestaan, mikäli sovellus halutaan siirtää moduulista toiseen. Tässä mielessä Sony Ericssonin valitsema lähestymistapa on muita huonompi.

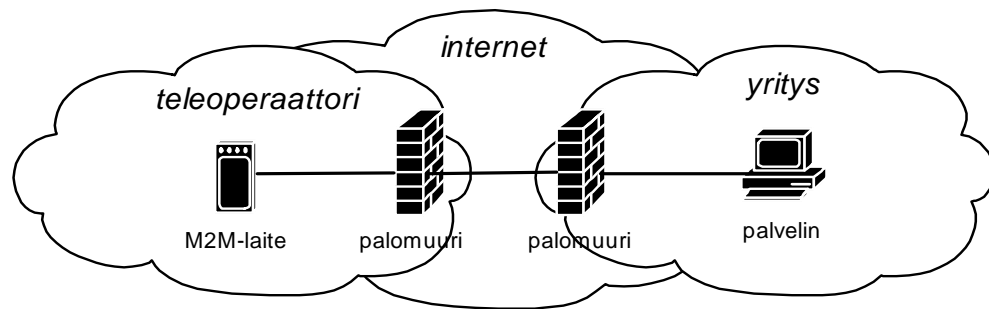
Moduulit eroavat merkittävästi myös käytettävissä olevan muistimäärän osalta. Erot ovat osittain selitettävissä sovelluskielten eroilla. Sony Ericssonin skriptikieli ei edellytä sovelluksen kääntämistä ja paketoimista kirjastoineen, joten yksittäisen sovelluksen koko on pienempi kuin Java-pohjaisilla moduuleilla. Oletettavasti tästä syystä myös moduulin muistimäärä on muiden valmistajien moduuleja selvästi pienempi. Tulkatussa skriptikielessä ohjelmakoodin kommentointi kuluttaa laitteen rajallista muistia, joten pahimmassa tapauksessa sovelluksesta joudutaan ylläpitämään kehitysvaiheessa kommentoimatonta versiota kommentoidun rinnalla laitteen muistinkäytön minimoimiseksi. Eri moduuleiden tulkkien toteutuksesta johtuen sovellusten suoritusnopeutta on kuitenkin hankala vertailla. Nokian moduulin muita suurempi muisti tarjoaa enemmän säilytystilaa sovellusten väliaikaistiedostoille ja useamman sovelluksen säilöminen itse moduulissa saattaa olla joissakin tilanteissa hyödyksi.

3.2 Langattomat päivitysmahdollisuudet

Ylläpidettävyyden kannalta laitteen ohjelmiston tulee olla päivitettävissä, jotta koko laitetta ei tarvitse korvata ohjelmistovikojen ilmetessä. Perinteisesti laitteiden päivitys on hoidettu sarjakaapeliyhteydellä suoraan laitteen kanssa. Kuitenkin kun laitteiden määrä kasvaa, muodostuu kaikkien laitteiden päivittämisestä aikaa vievä operaatio. Erityisesti tämä pätee, mikäli jokaisen laitteen luona joudutaan fyysisesti käymään erikseen. Lisäksi osa laitteista saattaa myös sijaita sellaisissa paikoissa, joihin pääseminen edellyttää erityisjärjestelyjä.

Nykyaikaiset M2M-laitteet sisältävät usein liittynän ulkoiseen langattomaan verkkoon, esimerkiksi matkapuhelinverkkoon. Ulkoista langatonta verkkoa käytetään laitteen toimintaan liittyvään seurantaan ja ohjaukseen. Tämän lisäksi verkkoa on mahdollista käyttää myös ohjelmistopäivityksen aikana. Tällöin laitteet on mahdollista päivittää automatisoidusti ilman fyysistä käyntiä laitteiden luona. Tämä säästää ajan lisäksi myös kustannuksia ja madaltaa kynnystä päivitysten suorittamiselle.

Yhteys matkapuhelinverkosta internetin kautta yrityksen verkkoon saattaa kuitenkin muodostua ongelmaksi langattoman päivityksen kannalta. Kuvassa 6 on esitetty pelkistetty kaavio tyypillisen verkon rakenteesta teleoperaattorin ja M2M-laitetta hallinnoivan yrityksen välillä. Koska yhteys kulkee osittain internetissä, päätyvät sekä teleoperaattorit että yritykset usein käyttämään palomuuria internetin ja oman verkkonsa välillä. Vaikka tämän kaltainen ratkaisu on perusteltua verkon tietoturvan kannalta, aiheuttaa se kuitenkin ongelmia, kun yrityksen verkosta halutaan saada yhteys teleoperaattorin verkossa olevaan M2M-laitteeseen. M2M-laitteen on mahdollista päivittää niin halutessaan tietoja suoraan yrityksen palvelimelle, sillä yritys voi muuttaa oman palomuurinsa asetuksia siten, että yhteys laitteelta palvelimelle sallitaan. Yrityksellä ei kuitenkaan ole mahdollisuutta vaikuttaa teleoperaattorin palomuurin toimintaan, ellei teleoperaattori tarjoa sellaisia palveluvaihtoehtoja, joissa palomuurin rajoituksia on lievennetty.



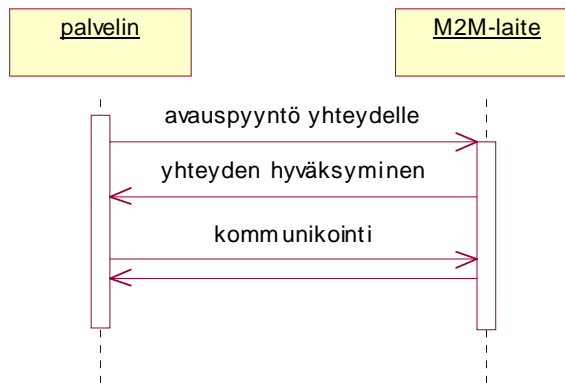
Kuva 6: Verkon rakenne.

Edellä kuvatun ongelman kiertämiseksi ovat eri valmistajat päätyneet M2M-laitteissaan toisistaan poikkeaviin ratkaisuihin, jotka esitellään seuraavissa kappaleissa. Olemassa olevien ratkaisujen lisäksi esitellään myös muutamia muita vaihtoehtoja, jotka saattaisivat olla käyttökelpoisia kahden palomuurin asettamien rajoitusten ohittamiseksi.

3.2.1 Piirikytkentäinen yhteys

Helpoin tapa laitteen päivittämiselle on käyttää piirikytkentäistä CSD-yhteyttä. Tämä edellyttää, että palvelin tietää kunkin päivitettävän laitteen puhelinnumeron, ja että palvelimelle on toteutettu tarvittava liitäntä CSD-yhteyttä varten.

CSD-yhteyden toiminta on esitetty kuvassa 7. Palvelin avaa yhteyden laitteelle, minkä jälkeen laite tunnistautuu palvelimelle ja on valmis vastaanottamaan komentoja. Vastaavalla tavalla päivitys on myös mahdollista suorittaa HSCSD-yhteydellä, mikäli tiedonsiirrosta halutaan nopeampi ja päivitettävä laite sitä tukee.



Kuva 7: CSD-yhteyden toiminta.

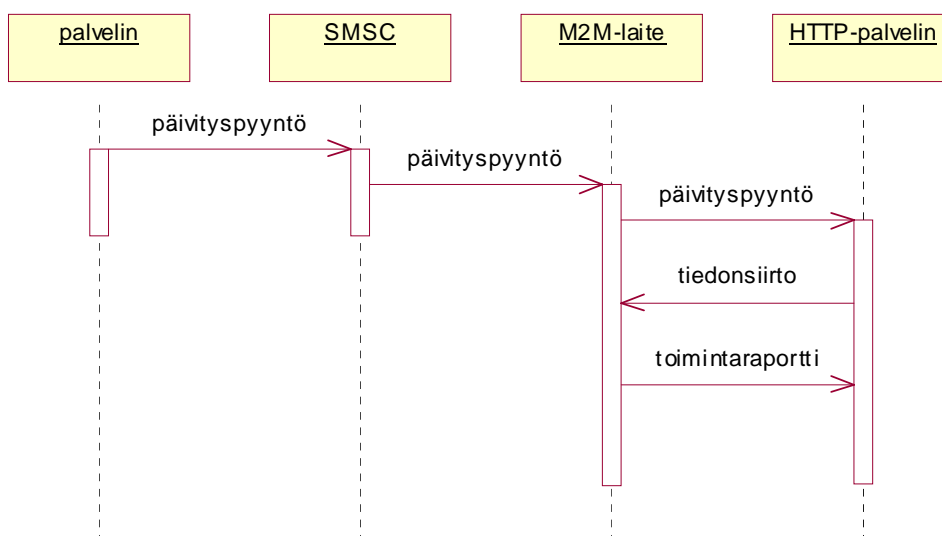
Laitteeseen muodostettava CSD-yhteys ei kulje internetin kautta, joten IP-verkon puolella olevat palomuurit eivät aiheuta ongelmia. Palvelin tarvitsee kuitenkin jokaista samanaikaisesti muodostettua yhteyttä varten oman yhteyskohtaisen modeemin. Tästä seuraa, että käytävissä oleva laitteisto rajoittaa samanaikaisesti päivitettävien laitteiden määrää ja vaikuttaa ison laitemäärän päivitysnopeuteen.

3.2.2 Hypertext Transfer Protocol

Sun Microsystemsin suosittelema tapa J2ME-pohjaisten sovellusten päivittämiseen perustuu HTTP-protokollan käyttämiseen laitteen ja palvelimen välillä. Päivitysmenetelmä on alun perin tarkoitettu matkapuhelinten Java-sovellusten päivittämiselle, mutta muutamat M2M-laittevalmistajat ovat soveltaneet sitä myös M2M-sovellusten siirrossa.

HTTP-yhteyden avulla suoritettu päivitys on esitetty kuvassa 8. Palvelin aloittaa päivitystapahtuman lähettämällä SMSC:n kautta päivityspyynnön laitteelle. Laite vastaanottaa tekstiviestin matkapuhelinverkon kautta, joten IP-verkon puolella olevat palomuurit eivät pääse vaikuttamaan viestin kulkuun. Palvelimen puolestaan tarvitsee tietää ainoastaan laitteen puhelinnumero. Laitteen vastaanottama viesti sisältää tiedot siitä, mitä laitteen tulisi tehdä. Sunin OTAP-menettely mahdollistaa sovelluksen asentamisen, poistamisen ja käynnistämisen. Mikäli kyseessä on päivitys, saa laite viestin mukana HTTP-osoitteen, josta

sovellus on haettavissa. Osoitteen lisäksi on mahdollista siirtää myös käyttäjätunnus ja salasana, joiden avulla laite voi tunnistautua palvelimelle. GPRS-yhteyden avulla laite pystyy hakemaan sovelluksen annetusta osoitteesta ja tiedonsiirron päätyttyä se ilmoittaa palvelimelle siirron onnistumisesta tai epäonnistumisesta POST-metodilla.



Kuva 8: HTTP-pohjainen päivitys.

Yksinkertaisuutensa ansiosta HTTP-pohjainen päivitys on ohjelmiston kannalta kevyt toteuttaa. Se on kuitenkin etupäässä tarkoitettu matkapuhelimien sovellusten päivittämiselle, eikä mahdollista M2M-laitteiden kanssa muuta tiedonsiirtoa sovelluksen lisäksi päivityksen yhteydessä. Palvelimen vastuulle jää muun muassa laitteeseen asennettujen sovellusten listan ylläpitäminen, sillä protokolla ei mahdollista listan pyytämistä laitteelta. Ainoa tieto minkä palvelin saa päivityksen yhteydessä on laitteen HTTP-palvelimelle palauttama raportointikoodi, mistä voidaan tulkita laitteen tila ja päivityksen onnistuminen.

Päivityksen aloittamisessa käytetty tekstiviesti voi myös muodostua tietyssä tilanteessa ongelmalliseksi. Kuormittamattomassa matkapuhelinverkossa tekstiviestin kulkeminen laitteelta toiselle on lähes välitöntä. Kuormitetussa verkossa viesti saattaa kuitenkin viipyä matkalla huomattavasti pidempään.

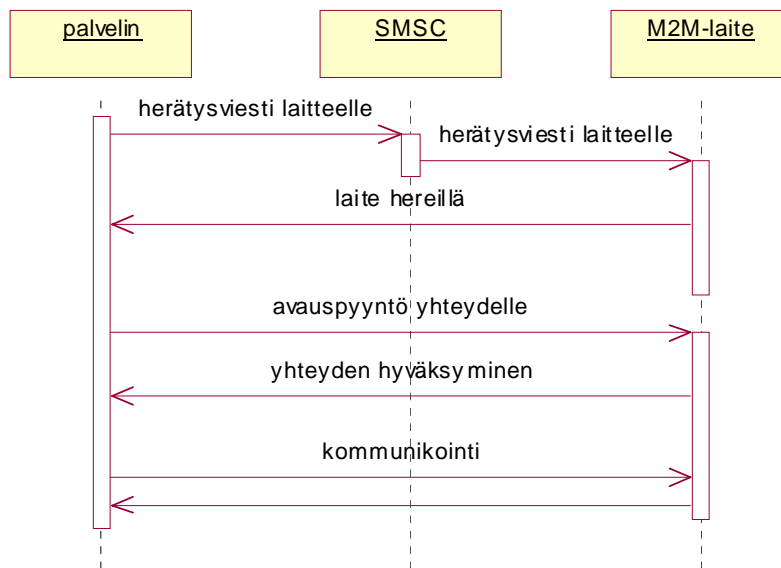
Verkko ei myöskään takaa viestin varmaa perillemenoaa. Lähettäjä ei kuitenkaan saa varmaa kuittausta viestin toimitustilasta. Tämä aiheuttaa sen ongelman, että laitteen päivitettävyys on riippuvainen matkapuhelinverkon kuormituksesta, eikä laitteen vastausaika ole täysin ennustettavissa. Päivitystä suorittava ohjelmisto ei täten voi luottaa siihen, että yksittäinen lähetetty päivityspyyntö päättyy täydellä varmuudella riittävän nopeasti päivitettävälle laitteelle.

Tiedonkulun helpottamiseksi HTTP-palvelin on useimmiten integroitu päivityspyynnön lähettäneeseen palvelimeen. Tällöin palvelin voi kirjata tietokantaansa laitteen ilmoittaman raportointikoodin ja poistaa sovelluksen HTTP-palvelimelta päivityksen päätyttyä. Näin menettelemällä voidaan välttää mahdollisuus, että ulkopuolinen liikennettä seurannut taho saisi haettua siirretyn sovelluksen HTTP-palvelimelta päivityksen jälkeen.

3.2.3 Common Object Request Broker Architecture

Nokia on N12-moduulissaan muista valmistajista poiketen päätenyt käyttämään CORBA-yhteyttä sovellusten päivittämiseen. Sovelluspäivitysten lisäksi CORBA mahdollistaa myös laitteen muiden tietojen helpon käsittelyn ja sille löytyy tuki useista ohjelmointiympäristöistä. [12]

CORBA-yhteyden avulla suoritettu kommunikointi laitteen kanssa on esitetty kuvassa 9. Yhteyden avaus aloitetaan lähettämällä palvelimelta viestikeskuksen kautta tekstiviesti moduulille. Tämän binäärimuotoisen tekstiviestin avulla moduuli voidaan käskää avaamaan joko GPRS- tai CSD-yhteys valitulle palvelimelle. Moduuli tarkistaa viestin mukana tulleiden asetusten vastaavuuden omassa muistissaan oleviin asetuksiin ja avaa yhteyden ainoastaan siinä tilanteessa, että asetukset vastaavat ja viestin lähettänyt numero on sama kuin moduulille ennestään määritetty numero. Vastauksena moduuli lähettää GIOP-vastauksen palvelimelle, josta palvelin voi tunnistaa moduulin ja sen osoitteen. [13]



Kuva 9: CORBA-pohjainen päivitys.

Palvelin avaa uuden yhteyden moduulille tunnistettuaan, että kuittausviesti tuli samalta moduulilta, jolle yhteyspyyntö lähetettiin. Uuden yhteyden avaaminen aiheuttaa kuitenkin Nokian päivitystoteutuksen ongelman. Mikäli moduulin käyttämällä teleoperaattorilla on palomuri verkkonsa suojana kuvan 6 tapaan, ei palvelin pysty muodostamaan yhteyttä moduulille. Teleoperaattori saattaa myös estää moduulia saamasta julkista IP-osoitetta, jolloin palvelimen ei ole mahdollista löytää oikeaa reittiä operaattorin verkkoon.

Olettaen, että teleoperaattori sallii yhteyden avaamisen palvelimelta moduulille, voi palvelin seuraavaksi lähettää moduulille yhteyspyynnön. Yhteyden muodostuttua palvelin voi kommunikoida CORBA:n avulla moduulin kanssa. CORBA:n käytön ansiosta palvelin pystyy sovelluspäivitysten lisäksi tarkkailemaan moduulin tilaa ja vaikuttamaan sen toimintaan. Esimerkiksi moduulin yhteysasetuksia voidaan muuttaa.

Edellä mainitun palomuurin tunnistaminen palvelimelta käsin on ongelmallista. Palvelin ei välttämättä pysty IP-osoitteen perusteella päättelemään, onko moduulin ilmoittama osoite julkinen. Mahdollisen palomuurin olemassaolo puolestaan selviää vain, jos moduulille lähetettyyn yhteyspyyntöön ei saada

vastausta. GPRS-verkossa vastauksen saaminen saattaa kuitenkin olosuhteista riippuen kestää, joten välitöntä tapaa palomuurin olemassaolon toteamiseen ei ole. Ongelman kiertämiseksi Nokian moduulin pitäisi tukea CORBA:n tarjoamaan kahdensuuntaista menettelytapaa, mikä mahdollistaisi moduulilta palvelimen suuntaan avatun yhteyden käyttämisen myös moduulille tarkoitettujen komentojen välittämiseen. Nokian moduulista kyseinen ominaisuus on kuitenkin jätetty pois, joten ainoaksi vaihtoehdoksi jää sellaisten teleoperaattoreiden käyttäminen, joilla ei ole palomuuria estämässä verkon ulkopuolisia yhteyksiä tai teleoperaattorilta oman APN:n vuokraaminen konekommunikaatiota varten.

CORBA-päivityksen toisena ongelmana on myös kappaleessa 3.2.2 esitetty epävarmuus yhteydenavauksen alussa käytetyn tekstiviestin perillemenosta. Vaikka päivitysmenetelmä käyttääkin toisenlaista viestimuotoa, ei sillä ole vaikutusta viestin perillemenon varmuuteen.

3.2.4 SyncML

SyncML on erityisesti matkapuhelimissa käytetty protokolla tietojen päivittämiseen tietokoneen ja puhelimen välillä. Sen käyttö rajoittuu kuitenkin usein lähinnä puhelimen kalenteritietojen ja puhelinmuistion päivittämiseen. SyncML mahdollistaa muutosten tekemisen erikseen sekä puhelimeen että tietokoneeseen, minkä jälkeen tiedot voidaan yhä saada synkronoitua. Protokolla kuitenkin mahdollistaa myös ainoastaan yhteen suuntaan tapahtuvan synkronoinnin. SyncML käyttää tiedon kuvaamiseen XML-pohjaisia viestejä, joten siirretyn tiedon rakenteella ei ole merkitystä protokollan kannalta. Yhdensuuntaista synkronointia käyttämällä myös konekommunikaatioon tarkoitettujen laitteiden ohjelmistopäivitykset olisi mahdollista toteuttaa standardoidulla tavalla. SyncML ei kuitenkaan määrittele tapaa laitteen herättämiselle etäkäyttötapauksissa. Operaation aloittaminen edellyttäisi laitteen herättämistä jollakin tavalla, esimerkiksi tekstiviestillä. [14]

3.2.5 GPRS

M2M-laitteet eivät yleensä ole jatkuvasti tilassa, jossa GPRS-yhteys on aktiivinen ja verkko käytettävissä. Tämän ja GPRS-verkon rajoitusten takia yhteyden avaaminen laitteelle ei ole suoraan mahdollista. Jotta yhteys olisi mahdollinen, pitää laite jollain menetelmällä käskä ää avaamaan GPRS-yhteys ja ilmoittamaan oma osoitteesä palvelimelle. Normaalisti tämä toteutetaan joko määrittelemällä laite avaamaan yhteys tietyin aikaväleihin tai lähettämällä laitteelle komento, esimerkiksi tekstiviesti, jolla laitetta käsketään ottamaan yhteyttä palvelimelle. Tekstiviesti ei kuitenkaan ole herätysmenetelmänä täysin luotettava, sillä palvelin ei saa tietoa tekstiviestin etenemisestä ja päättymisestä itse laitteelle. Mikäli laite ei vastaa heräteviestiin, ei voida olla varmoja, johtuuko ongelma viestin hukkumisesta matkalla vai toimintahäiriöstä itse laitteessa.

Vaihtoehto tekstiviestin tilalle on datapuhelun soittaminen laitteelle niin, että puhelun saapuessa laite kuitenkin sulkee puhelun siihen vastaamatta. Soittajan numeron perusteella laite tunnistaa soittajan ja numeron ollessa ennalta määritettyjen asetusten mukainen avaisi GPRS-yhteyden ja ottaisi yhteyttä palvelimelle. Soiton katkeaminen kertoisi palvelimelle laitteen reagoineen yhteydenavaukseen. Esitetty herätysmenetelmä on kestoiltaan niin lyhyt, että palvelimelle riittäisi yksi puhelinlinja useamman samanaikaisen yhteyden avaamiseen. Soiton katkaiseminen ennen siihen vastaamista minimoi myös herätystavasta aiheutuvat kulut, sillä operaattorit eivät laskuta vastaamatta jätetyistä puheluita.

Laitteen herättäminen päivitystä varten ei ole kaikissa käyttötarkoituksissa tarpeellinen. Sellaiset laitteet, jotka ovat muutenkin usein yhteydessä palvelimeen voivat tietyin aikaväleihin tarkistaa palvelimelta myös mahdollisen päivityksen tarpeellisuuden. Tällä tavoin palvelimen ei tarvitse erikseen avata yhteyttä laitteelle. Haittapuolena tosin on, että laitteiden päivitys voi kestää pitkään. Eri laitteiden päivitys saattaa tapahtua eri aikoina riippuen tahdistaa, jolla laitteet tarkistavat päivityksiensä saatavuutta. Vastaavalla menettelyllä laitteille on mahdollista päivittää ohjelmistopäivityksen yhteydessä myös muitakin asetuksia.

3.2.6 Mobile IP

Mobile IP on protokolla, joka mahdollistaa laitteen liikkumisen yrityksen verkon ulkopuolella siten, että yhteys laitteeseen voidaan ylläpitää laitteen sijainnista riippumatta. Konekommunikaatiossa kyseinen menettely auttaisi huomattavasti ohittamaan verkkojen asettamia rajoitteita. Mobile IP kuitenkin edellyttää, että sekä yrityksen oma verkko että vierailtava verkko tukevat protokollaa. Myös NAT:lla varustetussa verkossa vierailu on mahdollista, mutta tämäkin vaatii reitittimeltä erityisiä asetuksia. Operaattoreiden kannalta mobile IP on yhä kehitysvaiheessa, minkä takia mobile IP -pohjaiset verkot eivät ole vielä yleistyneet. Tämän vuoksi käyttömahdollisuudet konekommunikaatiossa ovat vielä varsin rajalliset. [15]

3.2.7 VPN

Langattomien verkkojen kautta kommunikoitaessa tietoturva voi muodostua ongelmaksi varsinkin, jos ohjelmistopäivitysten joutuminen vääriin käsiin halutaan varmuudella estää. Mikäli laitteen resurssit riittävät, on VPN tämän kaltaisessa tilanteessa järkevin ratkaisu. VPN myös mahdollistaisi kappaleessa 3.2.3 esitetyn kommunikointiongelman kiertämisen, sillä kun VPN-yhteys laitteen ja palvelimen välille saadaan muodostettua, on liikennöinti molempiin suuntiin mahdollista ilman operaattorin asettamia rajoituksia. Käytännössä kuitenkin nykyiset resursseiltaan vaatimattomat M2M-laitteet eivät sovellu VPN-yhteyksien muodostamiseen.

3.2.8 Yksityinen APN

Yksinkertaisin menetelmä laitteen kanssa kommunikointiin on kuitenkin suoraan yhteyden avaaminen laitteelle. Normaalisti GPRS-verkossa tämä ei kuitenkaan onnistu ilman operaattorin apua. Riippuen käytettyjen laitteiden määrästä voi kuitenkin olla kannattavaa ostaa operaattorilta käytettäväksi oma APN, jolloin operaattorin asettaman palomuurin toimintaan voidaan vaikuttaa. Operaattorin halukkuus kyseiseen järjestelyyn kuitenkin vaikuttaa vaihtoehdon toteutuskelpoisuuteen.

4 TARPEET

Yritysten hallintasovellukselle asettamat tarpeet voidaan karkeasti jakaa kahteen osa-alueeseen. Toisaalta tarpeet muodostuvat tiedonsiirron sujuvuudesta, toisaalta myös laitemäärän hallittavuudesta.

4.1 Tiedonsiirto

Langattomassa konekommunikaatiossa itse kommunikaatio näyttelee tärkeää osaa. Yhteyden laatu, luotettavuus ja siirtonopeus vaikuttavat M2M-ratkaisujen käyttökelpoisuuteen. Teleoperaattoreiden verkkoja käytettäessä yhteyden toimivuuteen ei usein voida vaikuttaa, joten sovelluksen vastuulle jää virhetilanteiden tunnistaminen ja niistä toipuminen. Luotettavan yhteyden varmistaminen voi myös edellyttää toissijaisen yhteyden käyttöön varautumista.

Kappaleessa 3.1 esitellyt laitteet käyttävät kaikki GSM-verkkoa siirtomedianaan. Käyttäjän kannalta tämä tarkoittaa sitä, että verkon laatuun voi vaikuttaa vain teleoperaattorivalinnan perusteella. Teleoperaattoreiden välisten erojen selvittäminen voi kuitenkin olla ongelmallista. Moni teleoperaattori jättää lähes poikkeuksetta kertomatta verkostaan konekommunikaation kannalta oleellisia tietoja, kuten mahdollisen palomuurin vaikutuksen tiedonsiirtoon GPRS-yhteyttä käytettäessä.

GPRS-yhteyden luotettavuus riippuu monesta asiasta, eikä käyttäjällä ole kovinkaan suuria mahdollisuuksia yhteyden luotettavuuden parantamiseksi. Luotettavuuteen vaikuttavat laitteen ja lähimmän tukiaseman sijainti, joiden perusteella signaalin kuuluvuus määräytyy. Pelkästään hyvä kenttä ei kuitenkaan riitä, sillä myös muut samaa tukiasemaa käyttävät laitteet vaikuttavat sen suorituskykyyn ja tarjolla olevaan kapasiteettiin. Siirtonopeuden kannalta GPRS-yhteys on hidas lähiverkko- tai sarjaporttisyhteyteen verrattuna, mutta M2M-tiedonsiirtoihin riittävä. Käytetty GPRS-luokka on riippuvainen M2M-laitteen radorajapinnan ominaisuuksista. Tällä ei kuitenkaan ole pienten tiedostojen osalta suurta merkitystä tiedonsiirron keston kannalta, sillä yhteyden avaaminen usein

vie suurimman osan ajasta. GPRS-verkolle ominainen noin sekunnin viive jokaisen paketin tiedonsiirrossa ei tosin ole ongelma konekommunikaation kannalta.

Tietoturva ei GPRS-verkon puolella ole ongelma, sillä ilmarajapinnassa on käytössä monitasoinen GSM:ltä peritty salaus [16]. Yhteys laitteelta palvelimelle kulkee kuitenkin GPRS-verkon lisäksi yleensä myös internetin kautta julkisia yhteyksiä pitkin. Usein nämä yhteydet ovat salaamattomia M2M-laitteiden rajallisen laskentakapasiteetin vuoksi. Teoriassa tämä mahdollistaa tilanteen, jossa ulkopuolisella on mahdollista päästä käsiksi laitteen ja palvelimen välillä siirrettyyn tietoon. M2M-laite itsessään on yleensä suhteellisen hyvin valmistajan puolesta suojattu luvattomia päivityksiä vastaan. Varsinaiset heikkoudet sijaitsevat usein joko laitteessa ajettavassa sovelluksessa, palvelimen ohjelmistossa tai järjestelmän käyttäjässä [17].

Virhetilanteita kommunikaation aikana voivat aiheuttaa sovelluksen virheet sekä verkon häiriöt. Virheiden näkyminen laitteen käytön aikana ei ole suotavaa, joten laitteen tulee kyetä toipumaan virheistä omatoimisesti ja mahdollisimman nopeasti. Sovellusvirheet ovat korjattavissa sovellusta päivittämällä. Valmistajan tarjoaman sovellusten kohdalla päivitettävyyden riippuu valmistajan laitteelle tarjoamasta tuesta, mikä usein vaihtelee valmistajakohtaisesti ja on myös riippuvainen laitteen iästä.

Langaton verkkoyhteys ei koskaan ole yhtä toimintavarma kuin langallinen. Langattomuuden etuna on laitteiden sijoittaminen sellaisiin paikkoihin, minne langallisen yhteyden järjestäminen olisi muuten ongelmallista. Laitteiden ei välttämättä myöskään tarvitse olla helposti fyysisesti käsiteltävissä etäpäivitysmahdollisuuden ansiosta. Toisaalta laitteet vaativat mahdollisuuden kahdensuuntaiselle yhteydelle, mikä rajoittaa käytettävissä olevien langattomien verkkoratkaisujen määrää.

Langattomassa yhteydessä satunnaiset verkkoyhteyden katkot saattavat aiheuttaa ongelmia kommunikoinnin kannalta. Katkojen esiintymistiheyteen vaikuttavat laitteen antennin vastaanottokyvyn lisäksi käytetyn tukiaseman muutkin käyttäjät, joten kaikkien virhemahdollisuuksien poissulkeminen julkisessa verkossa on mahdotonta. Varsinkin liikkuvan laitteen saattaa olla hankalaa ylläpitää jatkuvaa yhteyttä palvelimelle yhteyden laadun vaihdellessa eri tukiasemien välillä. Tällöin käytetyn sovelluksen tulee pystyä reagoimaan mahdollisiin tiedonsiirron aikana tapahtuviin verkkoyhteyden katkoihin ja kyetä joko jatkamaan siirtoa yhteyden palaututtua tai muodostamaan uusi yhteys tilanteen niin vaatiessa. GPRS-yhteyden vikasietoisuus auttaa kuitenkin usein yhteyden säilymisessä katkon yli. [18]

4.2 Hallittavuus

Laitteiden määrän kasvaessa muodostuu kokonaisuuden hallinnasta helposti ongelma. Muutaman laitteen hallinta ei tuota ongelmia, sillä laitteet ovat järkevästi yksilöitävissä ja täten myös helposti muistettavissa. Jos käsiteltävänä kuitenkin on satoja tai jopa tuhansia laitteita, muuttuu laitemäärä yksilöistä massaksi, jonka hallitseminen ilman apukeinoja on hankalaa, ellei peräti mahdotonta.

Esimerkiksi Vattenfallin suunnitelma [3] asentaa vuoden 2005 toukokuusta lähtien jopa 10 000 - 15 000 langattomasti luettavaa sähkömittaria kuukaudessa edellyttäen hyvin skaalautuvaa hallintamenetelmää, jotta kukin laite voitaisiin lukea kerran vuorokaudessa. Vattenfallin ratkaisu ongelmaan on luovuttaa laitteiden hallinta ja vastuu kommunikoinnin toiminnasta operaattorille. Teleoperaattori puolestaan toimittaa Vattenfallille vain mittareiden lukemat, jolloin Vattenfall välttyy laitteiden hallinnan järjestämiseltä. Tämän kaltainen järjestely kuitenkin sitoo yrityksen tiettyyn operaattoriin ja hankaloittaa operaattorin vaihtamista, mikäli siihen myöhemmin ilmenee syytä.

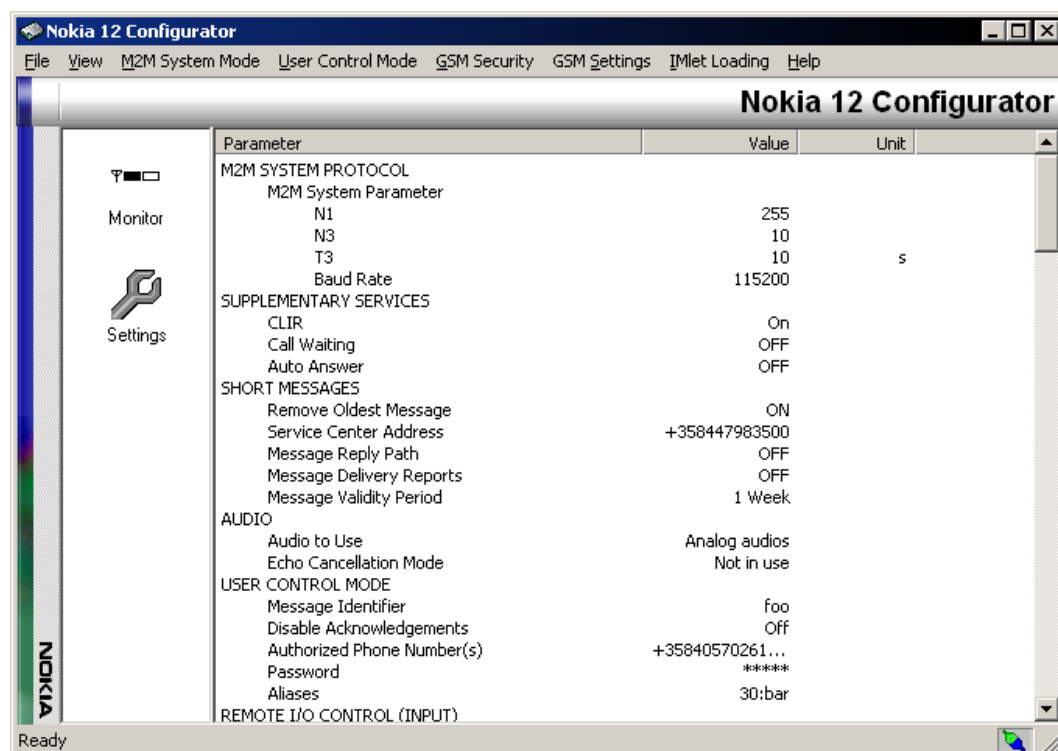
Ihanteellisessa tilanteessa yrityksellä on itsellään laitekanta omassa hallinnassa ja päivitettävissä omien tarpeiden mukaan. Tämä mahdollistaa myös teleoperaattorista riippumattoman toiminnan. Hallinnan kannalta laitteita tulee voida ryhmitellä halutun kriteerin perusteella, sillä tuhansia laitteita pitkän listan selaaminen ei ole tehokasta. Koska laitteet toimivat usein tiedonkeruulaitteina, tulee niiden keräämiin tietoihin päästä helposti käsiksi. Yritysten kannalta ongelmana kuitenkin on, ettei laitevalmistajilla ole tarjota sopivia valmiita työkaluja laitteiden hallintaan. Tästä on seurannut nykyinen tilanne, jossa yritykset lykkäävät suunnitelmiaan langattomaan viestintään siirtymisestä odotellessaan valmiiden ratkaisujen tarjonnan paranemista.

5 HALLINTASOVELLUS

M2M-laitteiden määrän kasvaessa muodostuu selvä tarve sovellukselle, joka mahdollistaa laitteiden keskitetyn hallinnan ja päivitysten suorittamisen. Moni laitevalmistaja tarjoaa vain yksinkertaisen laitteen asetusten muuttamiseen tarkoitetun ohjelman. Näiden ohjelmien käyttökelpoisuutta rajoittaa niiden kyky käsitellä vain yhtä laitetta kerralla. Tämän lisäksi ohjelmat vaativat jatkuvasti ihmisen läsnäoloa suorituksen aikana, eivätkä edes mahdollista useamman laitteen automaattista peräkkäistä päivitystä.

Nokian ratkaisu oman laitteensa hallittavuusongelmalle oli Nokia M2M Gateway -nimellä tunnettu kokonaisuus. Ratkaisun toiminta perustui teleoperaattorin verkkoon sijoitettuun ohjelmistopohjaiseen reitittimeen. Reititin toimi yhdyskäytävänä yrityksen oman verkon ja teleoperaattorin GSM-verkossa olevien laitteiden välillä. Yrityksen kannalta tämä mahdollisti teleoperaattorin mahdollisen palomuurin rajoitusten ohittamisen. Ratkaisu myös paransi hallintayhteyksien tietoturvaa mahdollistamalla VPN-yhteyden yhdyskäytävänä toimivan reitittimen ja yrityksen verkon välillä. Nokia kuitenkin lopetti yhdyskäytäväratkaisun tarjoamisen keväällä 2004. [19]

Nokian lopetetun yhdyskäytäväratkaisun jälkeen ainoaksi tavaksi hallita laitteita jäi Nokian tarjoama yksinkertainen Windows-pohjainen sovellus (kuva 10). Sovelluksen kautta on mahdollista hallita yhtä yksittäistä laitetta kerrallaan sarjakaapelin tai TCP-yhteyden avulla. Sovellus ei kuitenkaan tarjoa mahdollisuutta herättää GSM-verkossa olevia laitteita, joten yhteydet pitää muodostaa manuaalisesti muita keinoja käyttäen kuten soittamalla datapuhelu halutulle laitteelle.



Kuva 10: Nokia 12 Configurator.

5.1 Keskeiset vaatimukset

Työssä kehitettiin Nokian sovelluksen korvaava laitteiden ja sovellusten hallintaohjelmisto. Aluksi ohjelmistoa varten laadittiin vaatimusmäärittely, johon vaatimukset yksilöitiin. Kullekin vaatimukselle annettiin tunniste, lyhyt kuvaus sekä määriteltiin vaatimus joko pakolliseksi tai vapaaehtoiseksi. Pakolliset vaatimukset määriteltiin tarkastelemalla Nokian sovelluksen laitehallinnan kannalta oleellisia ominaisuuksia. [20]

5.1.1 Laitehallinta

Vaatimuksena oli tukea useamman eri laitevalmistajan laitteista. Laitemäärän tuli olla skaalautua muutamasta laitteesta jopa satoihin. Hallintasovellukselta edellytettiin tukea useamman laitteen samanaikaiselle päivittämiselle. Kyseisellä menettelyllä oli tavoitteena lyhentää päivitysprosessiin kuluva aikaa laitemäärän

kasvaessa suureksi. Laitemäärän hallinnan helpottamiseksi laitteita tuli myös olla mahdollista käsitellä ryhminä.

Jokaiselle laitteelle haluttiin erillinen loki, jonka kautta laitteelle suoritettavat toimenpiteet ja mahdolliset päivitysprosessissa tapahtuneet virheet olisivat selkeästi näkyvissä. Laitteen tilan tarkistaminen ja uudelleenkäynnistäminen tuli niin ikään olla mahdollista. Hallintasovellukselta vaadittiin myös vikasietoisuutta verkkohäiriöiden ja käyttäjien virheellisten toimenpiteiden varalta.

5.1.2 Sovellusten hallinta

Sovellusten hallinnan tavoitteena on ylläpitää tietoa siitä, mikä sovellus missäkin laitteessa on asennettuna ja onko kyseinen sovellus käynnistettynä. Vaatimuksen täyttämiseksi hallintasovelluksen piti mahdollistaa sovellusten lähettäminen yksittäisille laitteille tai laiteryhmillä. Sovellusten käynnistäminen ja sammuttaminen tuli olla mahdollista joko siirron yhteydessä tai erillisenä toimenpiteenä. Myös sovelluksen operatiivinen tila piti kyetä tarkistamaan. Lisäksi haluttiin mahdollisuus ajastaa sovelluksiin liittyviä muutoksia.

5.1.3 Vaatimukset toteutukselle

Toteutuksesta haluttiin mahdollisimman modulaarinen ja toteutusalueesta riippumaton, jotta sen laajentaminen myöhemmin ei vaatisi isoja rakenteellisia muutoksia. Laajentamistarpeina nähtiin tuki uusille moduuleille, eri operaattoreille ja eri ajoympäristöille. Hallinnasta haluttiin keskitetty, mutta kuitenkin siten, että järjestelmän käyttäminen useammasta paikasta olisi mahdollista.

Tiedonsiirron luotettavuuteen haluttiin kiinnittää erityistä huomiota. Ensimmäiseksi tuettavaksi laitteeksi valittiin Nokia N12, mistä johtuen tiedonsiirtona tulotisiin käyttämään CORBA-yhteyttä moduulin ja palvelimen välillä.

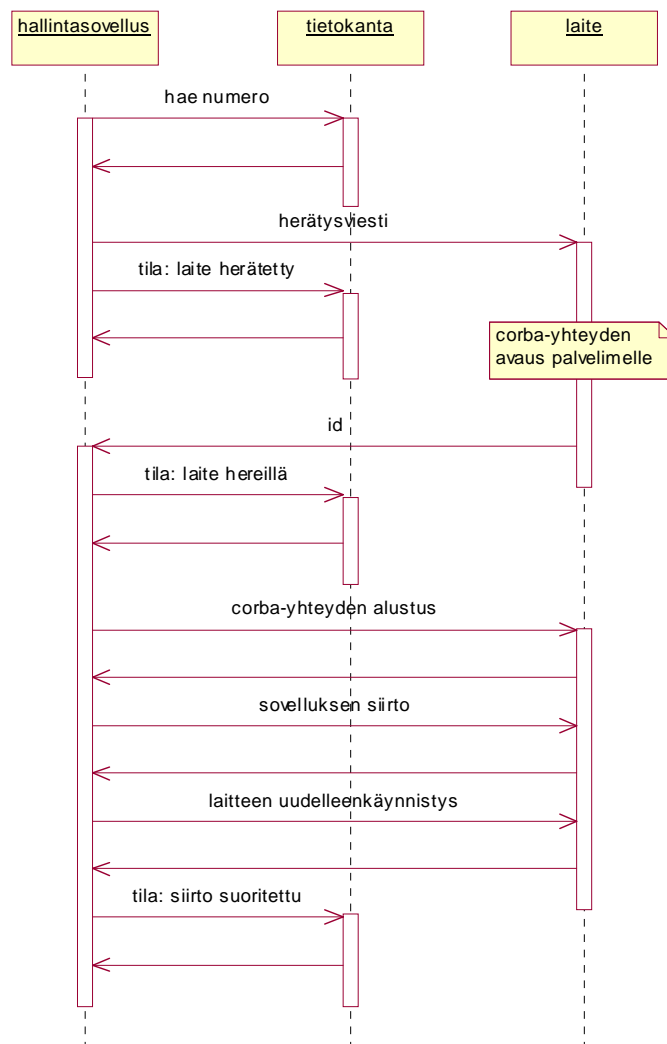
Käyttöliittymän osalta vaadittiin selkeää ja helppokäyttöistä tarpeeseen mukautettavaa käyttöliittymää. Siltä vaadittiin modulaarisuutta, mikä mahdollistaisi uusien ominaisuuksien lisäämisen uusien käyttötarpeiden ilmetessä.

5.2 Toiminnallisuus

Vaatimusmäärittelyn jälkeen tehtiin toiminnallisuuskuvaus, joka toteutettiin oliopohjaisesti UML:n avulla. Apuna käytettiin Rational Rose -työkalua. Toiminnallisuus mallinnettiin viestikaavioiden avulla, jotta sovelluksen eri osien välisestä viestiliikenteestä saatiin selvä kuva ja rajapinnat sovellusten eri osien välillä määriteltyä.

Kuvassa 11 on esitettyä esimerkkinä tilanne, jossa käyttäjä on valinnut käyttöliittymän kautta laitteelle sovelluksen lähetettäväksi ja hallintajärjestelmä alkaa suorittaa sovelluksen siirtoa laitteelle. Kuvan selkeyttämiseksi hallintasovelluksen eri osa on kuvattu yhtenä kokonaisuutena. Viestikaavion alkutilana on käynnissä oleva hallintasovellus, jolla on tarvittavat verkkoyhteydet käytettävissä. Hallintasovellus hakee käyttöliittymältä saadun komennon perusteella tietokannasta laitteen numeron ja lähettää laitteelle herätysviestin, minkä jälkeen laite merkitään tietokantaan herätetyksi ja vastausta odottavaksi. Herätysviestin mukana siirtyy tunniste, minkä perusteella laite voidaan myöhemmin varmentaa oikeaksi.

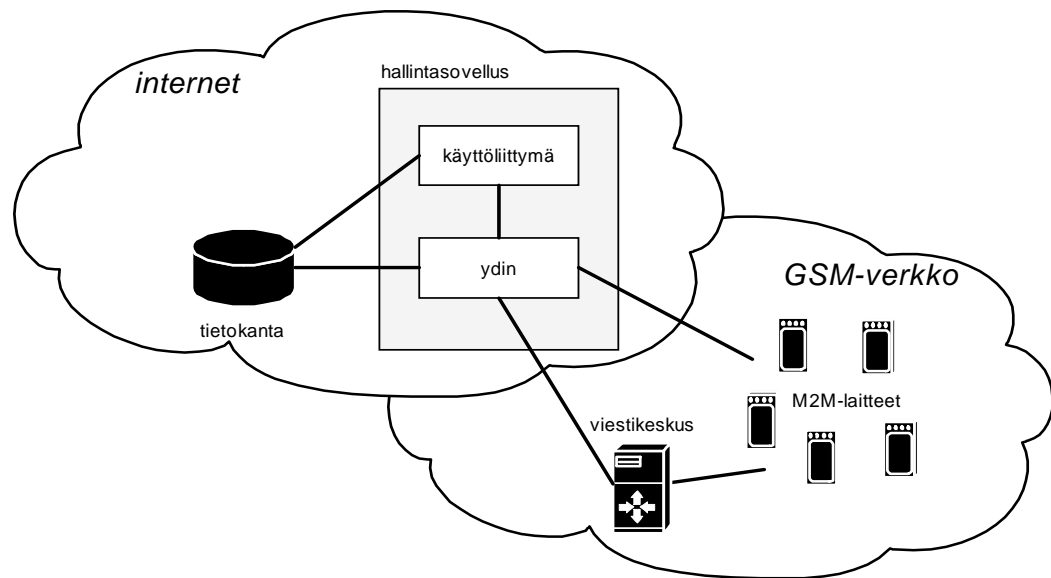
Herätysviestin saatuaan laite kirjautuu verkkoon ja avaa CORBA-yhteyden palvelimelle käyttäen herätysviestin mukana tullutta tunnistetta. Hallintasovellus tarkistaa tunnisteen tietokannasta saaden vastauksena siirtoon tarvittavat tiedot ja merkitsee laitteen aktiiviseksi. Tämän jälkeen avataan uusi CORBA-yhteys laitteelle, siirretään sovellus ja käsketään laitetta käynnistämään itsensä uudestaan. Lopuksi tietokantaan merkitään sovelluksen siirron onnistuneen ja laitteen olevan jälleen valmis vastaanottamaan uusia komentoja.



Kuva 11: Viestikaavio sovelluksen siirtämisestä laitteelle.

5.3 Hallintasovelluksen rakenne

M2M-hallintajärjestelmä tarvitsee toimiakseen kuvassa 12 esitetyt komponentit. Työssä kehitetyn hallintasovelluksen keskeisimmät osat ovat käyttöliittymä ja ydin.



Kuva 12: M2M-hallintajärjestelmän komponentit.

Käyttöliittymänä toimii HTTP-palvelimella toteutettu sivukokonaisuus. Jokainen sivu generoidaan PHP-kieltä käyttäen latauskohtaisesti, jolloin käyttäjän kannalta tilanne näyttää reaaliaikaiselta. Käyttöliittymällä on suora yhteys tietokantapalvelimeen ja varsinaiseen hallintasovelluksen ytimeen. Tietokantayhteyden avulla käyttöliittymän kautta voidaan suoraan muokata laitteiden tietoja ja määrittää laitteille suoritettavia komentosarjoja. Ytimelle annetulla käskyllä puolestaan saadaan haluttujen toimintojen suoritus käynnistymään.

Tietokanta tarjoaa hallintasovellukselle keskitetyn paikan, jonne tietoa voidaan varastoida. Se tarjoaa myös mahdollisuuden hakea esimerkiksi laitelistoja halutuilla suodatuksilla siten, että ydin saa vastauksena valmiin laitelistan sen tarvitsemassa lopullisessa muodossa. Tietokanta on eriytetty muusta ytimeestä, jotta se olisi tarvittaessa korvattavissa, mikäli halutaan käyttää muuta tietokantaratkaisua tai mikäli asiakkaalla on jo ennestään jokin tietokantaratkaisu asennettuna.

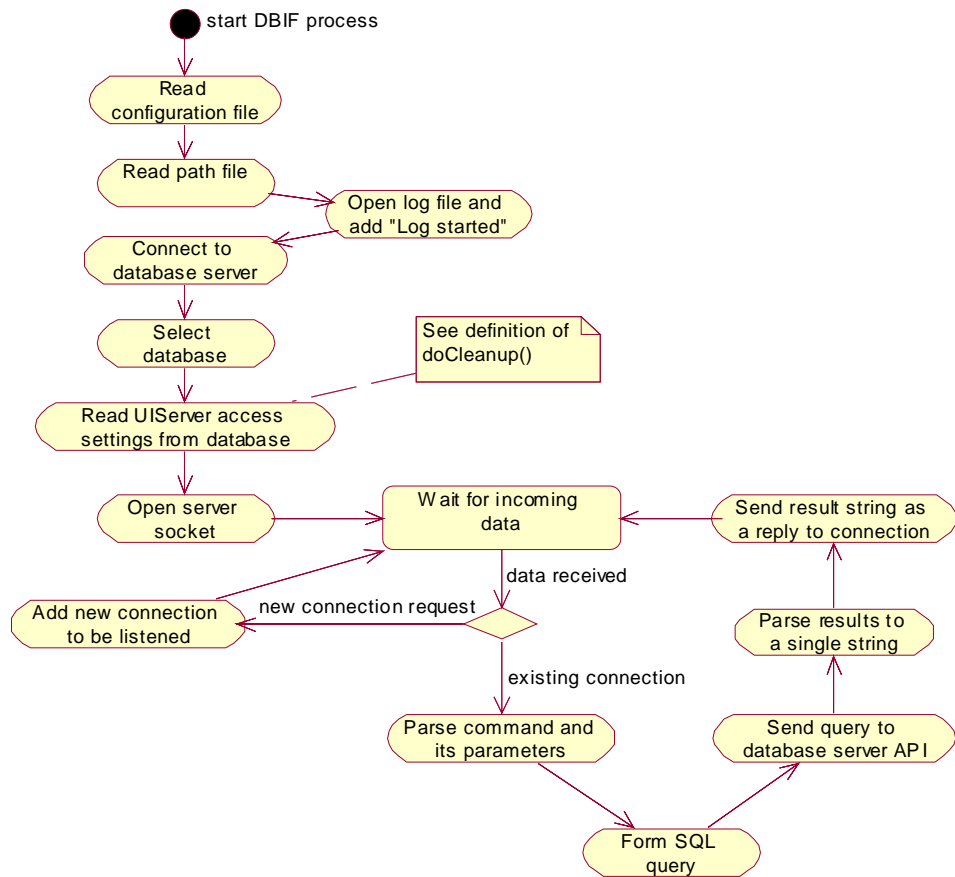
Viestikeskus mahdollistaa laitteiden herättämiseen tarvittavien tekstiviestien lähettämisen hallintasovelluksen kautta automatisoidusti. Viestikeskuksena

voidaan käyttää joko operaattorin tarjoamaa palvelua tai modeemina esiintyvää matkapuhelinta. Jälkimmäinen vaihtoehto soveltuu erityisesti sellaisissa tapauksissa, joissa lähetettävien viestien määrä ei ole kovinkaan suuri. Viestikeskusta hallitaan suoraan hallintasovelluksen ytimestä, joten se ei tarvitse yhteyttä tietokantaan.

Hallintasovelluksen ydin vastaa eri toiminnallisuuksien toteuttamisesta. Se toimii rajapintana tietokannan, käyttöliittymän ja laitteiden välillä. Sen vastuulla on kommunikointi laitteiden kanssa käyttöliittymän kautta saatujen ohjeiden mukaan.

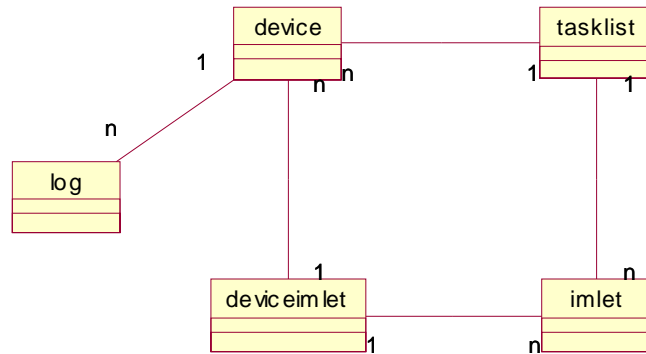
5.3.1 Tietokanta

Tietokannan tarkoituksena on hallintasovelluksen käyttämien ja tuottamien tietojen varastoiminen ja tarjoaminen hallintasovelluksen muille osille. Tietokannan tiedot tarjotaan muille osille erillisen tietokantarajapinnan avulla. Kuvassa 13 on esitetty tietokantarajapinnan vuokaavio, mikä selventää kuinka prosessi käynnistyessään lukee tarvittavat asetukset ja sen jälkeen siirtyy käsittelemään sille tulevat yhteydet ja yhteyksien kautta lähetetyt komennot.



Kuva 13: Vuokaavio tietokantarajapinnan prosessista.

Hallintasovelluksen käyttämän tietokannan rakenne (kuva 14) koostuu viidestä taulusta, joita käytetään log-taulua lukuun ottamatta laitteiden ja niiden sovellusten hallintaa. Log-taulun tarkoituksena on tarjota käyttäjälle tietoa hallintasovelluksen toiminnasta, laitteelle suoritetuista toimenpiteistä ja virhetilanteessa antaa tietoa, minkä perusteella vian syytä voidaan lähteä selvittämään.



Kuva 14: Tietokannan rakenne.

Device-taulu sisältää listan laitteista mukaan lukien laitteiden tiedot, kuten puhelinnumeron ja laitteen mahdollisen nimen. Imlet-taulu puolestaan sisältää hallintasovellukseen syötetyt M2M-sovellukset. Koska jokaiseen laitteeseen on mahdollista tallentaa useampi sovellus, joudutaan tietokannassa käyttämään yhtä taulua kahden edellä mainitun taulun välillä. Tämä deviceimlet-nimellä oleva taulu sisältää listan siitä, mikä sovellus on asennettuna missäkin laitteessa. Tieto käynnissä olevasta sovelluksesta kuitenkin löytyy device-aulusta, sillä laitteet kykenevät suorittamaan ainoastaan yhtä sovellusta kerrallaan.

Tasklist-taulun käyttötarkoituksena on laitteille suoritettavien komentojen säilyminen sinä aikana kun laite on joko jonossa tai ottamassa yhteyttä. Se myös mahdollistaa useamman komennon ketjuttamisen samalle laitteelle, mikäli kyseiselle toiminnalle on tarvetta.

5.3.2 Käyttöliittymä

Käyttöliittymä toteutettiin selainpohjaisena, jotta järjestelmän käyttäminen olisi mahdollisimman alustariippumatonta. Yhteensopivuutta eri selainten välillä pyrittiin varmistamaan olemalla käyttämättä selainriippuvaisia ominaisuuksia. Tämän takia käyttöliittymä ei vaadi selaimelta esimerkiksi JavaScript- tai Java-tukea. Kyseisten ominaisuuksia jättäminen pois mahdollistaa käyttöliittymän toimivuuden myös PDA-laitteiden ja tekstipohjaisten terminaalien selaimilla.

Käyttöliittymä on jaettu kolmeen pääosaan; laitteet, sovellukset ja asetukset. Näiden lisäksi näkyvillä on myös muutama vähemmän oleellinen osio, joiden kautta järjestelmän toiminnasta saadaan tarvittaessa tietoa.

Hallintasovelluksen käyttöönottovaiheessa käyttöliittymän käyttö aloitetaan yleensä asetussivulta (kuva 15). Tämän sivun kautta määritetään hallintasovelluksen käyttämät portit, moneenko laitteeseen ollaan maksimissa samanaikaisesti yhteydessä, mitä viestikeskusta käyttäen heräteviestit lähetetään ja millä tavoin laitteiden kanssa kommunikoidaan. Laitteiden yhteysasetusten tulee vastata käyttöliittymän kautta asetettuja arvoja. Laite ei reagoi lähetettyihin heräteviesteihin, mikäli yksikin heräteviestin mukana tulevista yhteysasetuksista on eri kuin laitteen muistista löytyvät asetukset.

The screenshot shows the 'Settings' page of the IMlet Manager. The navigation bar at the top includes 'Home', 'Status', 'Device Manager', 'IMlet Manager', and 'Settings'. The main content area is divided into three sections:

- Engine**: Contains a table for server configurations and a field for simultaneous connections.

	ip	port
UI server	127.0.0.1	31020
DB server	127.0.0.1	31010
IMlet server	127.0.0.1	6667
Simultaneous connections	1	
- SMS Gateway**: Contains fields for ip / port (127.0.0.1, 13131) and user / password (foo, ***).
- N12 Communication**: Contains a bearer selection (GPRS selected, CSD), a gateway name (internet), and an M3 server ip / port (123.123.123.123, 6667).

A 'Save' button is located at the bottom of the settings area.

Kuva 15: Asetukset.

Kuva 16 esittää sovellusten hallintaan käytettyä sivua. Sivun mahdollistaa sovellusten tietojen tarkastelun ja sovellusten syöttämisen hallintasovelluksen tietokantaan. Näkyvänä tietona on myös niiden laitteiden määrä, joihin sovellus on asennettu. Ainoastaan tietokannassa olevat sovellukset ovat valittavissa laitteille asennettaviksi. Tällä rajoituksella voidaan varmistaa, että laitteeseen lähetetystä sovelluksesta on tarvittaessa saatavilla paikallinen kopio. Samasta syystä tietokannassa olevaa sovellusta ei voi poistaa sen ollessa asennettuna johonkin laitteeseen. Hallintasovellus tarkistaa syötettävän sovelluksen tiedot ennen sen lisäämistä tietokantaan. Tässä vaiheessa sovelluksesta puretaan muut tietokannan tarvitsemat tiedot ja samalla rakenteeltaan virheelliset sovellukset voidaan myös hylätä.

The screenshot shows the 'IMlet Manager' interface. At the top, there is a navigation bar with links for 'Home', 'Status', 'Device Manager', 'IMlet Manager', and 'Settings'. Below this is a table listing installed applications. The table has columns for 'name', 'suite name', 'version', 'info', 'date', 'size', 'use', and 'action'. Below the table is a section titled 'Add IMlet' with input fields for 'Name:', 'Suite name:', 'Version:', 'Info:', and 'File:'. There is a 'Browse...' button next to the 'File:' field. At the bottom of the form, there is a 'Submit' button and a checked checkbox labeled 'Autodetect Name and Suite name'.

name	suitename	version	info	date	size	use	action
Identification	Identification	0.9	test release	19.09.2005 10:03:18	1.12 kB	2	edit / delete
Location	Location	2.1	-	19.09.2005 10:02:48	55.27 kB	2	edit / delete
Network	Network	1.3	improved version	19.09.2005 10:03:58	84.59 kB	1	edit / delete
Network	Network	1.2	-	19.09.2005 10:03:51	121.84 kB	0	edit / delete

Add IMlet

Name:

Suite name:

Version:

Info:

File:

Autodetect Name and Suite name

Kuva 16: Sovellukset.

Kuva 17 esittää laitehallintaan tarkoitetun osion ensimmäistä sivua. Sivun kautta voidaan syöttää uusia laitteita tietokantaan ja valita laiteryhmiä päivitettäväksi. Sivun ilmoittaa myös laitteiden reaaliaikaisen tilan, minkä avulla on mahdollista seurata laitteiden päivitystä, jos sellainen on meneillään.

Home Status Device Manager IMlet Manager Settings										
Edit mode: disabled										
	name	number	location	updated	connected	status	running imlet	imlets	action	
<input type="checkbox"/>	AGP-123	36173052	60,20° N 23,01° E	19.09.2005 10:00:18	26.01.2005 13:46:51	ok	Location	2	details / status / delete	
<input type="checkbox"/>	BAR-322	56678342	69,05° N 20,83° E	19.09.2005 10:00:26	26.01.2005 13:48:34	ok	Location	2	details / status / delete	
<input type="checkbox"/>	FOO-478	39852956	60,15° N 24,95° E	19.09.2005 10:00:05	26.01.2005 13:48:34	empty	none	0	details / status / delete	
<input type="checkbox"/>	RGB-508	76529314	66,48° N 25,67° E	25.08.2005 15:23:23	26.01.2005 13:48:34	ok	Network	1	details / status / delete	
<input type="checkbox"/>	TLA-995	12345678	65,02° N 25,50° E	19.09.2005 09:59:51	21.01.2005 09:32:25	empty	none	0	details / status / delete	
<input type="checkbox"/>	VGO-971	56782345	61,17° N 24,42° E	19.09.2005 09:59:59	25.01.2005 17:27:57	empty	none	0	details / status / delete	

Show selected

Select by group

All devices

Select by imlet

Identification (1.0) ... 2

Add Device

Name:

Number:

Location:

Group:

Kuva 17: Laitteet.

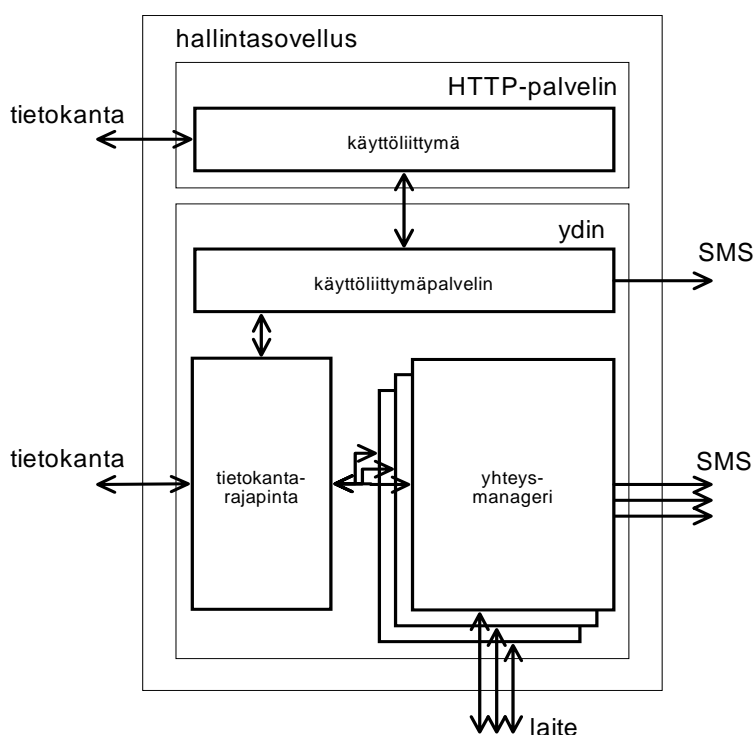
Laittehallintasivun kautta on mahdollista päästä käsiksi yksittäisten laitteiden asetuksiin (kuva 18). Sivun mahdollistaa laitteen tietojen päivittämisen, sovellusten siirtämisen ja niiden tilan muuttamisen. Laitteen tapahtumaloki on myös nähtävissä tätä kautta ja se päivittyy reaaliaikaisesti laitteen tilan muuttuessa. Käyttöliittymässä on myös varattu paikka laitteen sijainnille. Laitteen paikkatieto on mahdollista selvittää, mikäli siihen on kytketty GPS-moduuli.

Home Status Device Manager IMlet Manager Settings					
Details: BAR-322					
Edit mode: disabled					
name	number	location	status	action	
BAR-322	56678342	69,05° N 20,83° E	ok	<input type="button" value="Save"/>	<input type="button" value="Status"/> <input type="button" value="Reboot"/>
updated	connected				
19.09.2005 10:00:26	26.01.2005 13:48:34				
Transfer imlet					
Network (1.3) <input type="button" value="Select"/>		<input type="checkbox"/> Start imlet after transfer			
<input type="button" value="Transfer"/>					
IMlet list					
name	version	added	active	action	
Identification	0.9	19.09.2005 10:00:26	no	start / delete	
Location	2.1	25.08.2005 15:23:11	yes	stop / delete	

Kuva 18: Laittehallinta.

5.3.3 Ydin

Hallintasovelluksen ydin koostuu kolmesta prosessista (kuva 19), joiden kautta ydin on yhteydessä kaikkiin muihin hallintasovelluksen osiin. Tietokantarajapinta vastaa ytimen yhteyksistä tietokantaan, käyttöliittymäpalvelin hoitaa käyttöliittymän viestien tulkitsemisen ja yhteysmanageri toimii välikappaleena laitteiden kanssa kommunikointiin.



Kuva 19: Ytimen rakenne.

Tietokantarajapinta tarjoaa ytimen muille osille yhteyden tietokantaan. Se on toteutettu siten, että tietokannan tyyppi voidaan tarvittaessa vaihtaa toiseen, jolloin pelkästään tietokantaa käsittelevä rajapinta joudutaan toteuttamaan uudestaan. Tällä tavalla erityyppisten tietokantojen tukeminen on mahdollista pienellä vaivalla. Tietokantarajapinta tarjoaa tietokantayhteyden käyttöliittymäpalvelimelle ja yhteysmanagerille. Varsinainen käyttöliittymä puolestaan käyttää suoraan yhteyttä tietokantaan.

Käyttöliittymäpalvelin vastaa ytimen ja käyttöliittymän välisestä kommunikoinnista. Se vastaanottaa käyttöliittymän lähettämät päivityspyynnöt ja asettaa ne ylläpitämäänsä jonoon, mikäli asetettu maksimimäärä laitteita on jo aktiivisena. Käyttöliittymäpalvelin lähettää tarpeen mukaan uusille laitteille herätysviestejä, kun aktiivisena olevat laitteet kuittaavat omat tehtävänsä suoritetuiksi. Palvelimen toimintoihin kuuluu myös vastaamattomien laitteiden palauttaminen takaisin jonon perälle uutta yhteysyritystä varten ja virhetilanteeseen päätyneiden laitteiden siivoaminen pois jonosta. Se myös ilmoittaa käyttöliittymälle jokaisella sivunlatauksella ytimen toimintatilan, jolloin käyttöliittymä voi tarvittaessa ilmoittaa käyttäjälle, mikäli ytimen toiminnassa on ilmennyt ongelmia.

Yhteysmanagerin vastuulla on kommunikointi laitteiden kanssa alkaen laitteen tunnistamisesta, M2M-sovelluksen siirtämiseen ja tiedonsiirtoprosessin päättämiseen. Yhteysmanageri luo itsestään uuden prosessin tunnistetun laitteen otettua yhteyden. Laitteet tunnistetaan herätysviestin mukana lähetetystä numerosarjasta, joka vastaa tietokannassa laitteen tunnistenumeroa. Yhteysmanageri tarkistaa tietokantaliittymän kautta onko ilmoitetulla tunnistenumerolla oleva laite merkitty päivitettäväksi ja käynnistää laitetta varten uuden prosessin tarkistuksen onnistuessa. Tunnistenumeron yhteydessä yhteysmanageri saa tietokannasta tiedon laitteelle tehtävistä toimenpiteistä.

Siirrettävä sovellus haetaan toimenpidelistan mukana tulevaa sovellustunnistetta käyttäen tietokannasta, mikäli laitteelle on tarkoitus sovellus siirtää. Tämän jälkeen yhteysmanageri avaa laitteen suuntaan oman CORBA-yhteyden tiedonsiirtoa varten ja alkaa suorittaa toimenpidelistan mukaisia komentoja. Mahdollinen sovellus siirretään laitteelle laitteen valitsemaa lohkokokoa käyttäen. Virhetilanteet voidaan tunnistaa CORBA-rajapinnan avulla, jolloin virheen laadusta riippuen yhteysmanageri voi yrittää toimenpidettä uudestaan tai merkitä toimenpiteen epäonnistuneeksi, mikäli korjausyritykset eivät tuota toivottua tulosta.

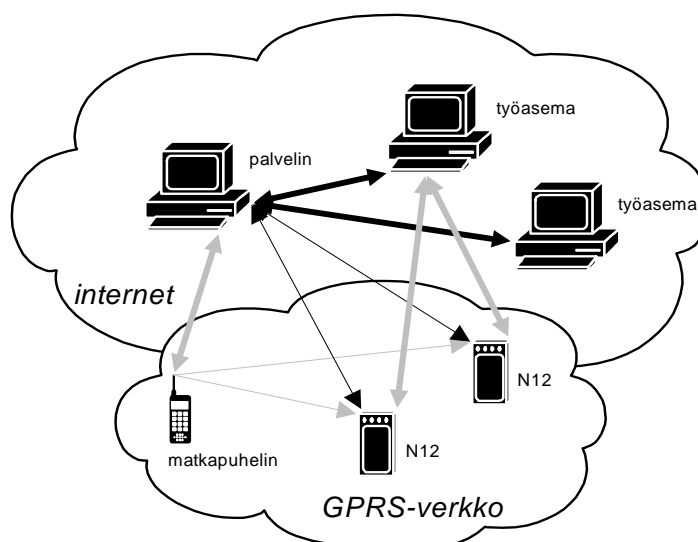
Yhteysmanageri lisää tietokantaan laitteelle lokimerkintöjä valittujen toimenpiteiden edistyessä. Niitä käyttäen käyttäjän on mahdollista seurata päivitystapahtuman etenemistä käyttöliittymän kautta lähes reaaliaikaisesti. Valittujen toimenpiteiden päätyttyä laitteen uusi tila päivitetään tietokantaan. Tarvittaessa yhteysmanageri pystyy myös lähettämään laitteelle uuden herätysviestin, mikäli yhteys laitteeseen menetetään kesken toimenpiteiden suorittamisen tai toimenpiteet edellyttävät laitteen käynnistämistä uudelleen eri toimenpiteiden välissä. Yhteysmanageri ei kuitenkaan missään vaiheessa aktiivisesti etsi laitteita verkosta, vaan odottaa laitteiden yhteydenottoja.

6 HALLINTASOVELLUKSEN TESTAUS GPRS-VERKOSSA

6.1 Testiympäristö

Hallintasovelluksen testiympäristö rakentui melko pitkälti tavallisen PC-työaseman ympärille. Työasemalta oli pääsy matkapuhelinverkkoon sarjaporttiin liitetyn matkapuhelimen avulla. Hallintasovellus käytti puhelinta vain tekstiviestien lähettämiseen, mutta se mahdollisti tarvittaessa myös datapuheluiden soittamisen muille laitteille. Työaseman lisäksi käytössä oli kaksi muuta PC-konetta, joiden käyttötarkoitus rajoittui lähinnä hallintasovelluksen testaamiseen verkon yli ja moduulien asetusten muuttamiseen Windows-pohjaisilla sovelluksilla.

Testiympäristön kokoonpano on esitetty kuvassa 20. Kuvassa esiintyvät nuolet esittävät eri laitteiden välisiä yhteyksiä. Paksu harmaa viiva kuvastaa sarjaporttiyhteyttä, ohut harmaa viiva tekstiviestein suoritettua viestintää, paksu musta viiva TCP-yhteyttä ja ohut musta viiva GPRS-verkon yli luotua TCP-yhteyttä. Nuolien suunnat ilmaisevat liikenteen kulkusuunnan.



Kuva 20: Testiympäristö.

PC-työaseman käyttöjärjestelmäksi valittiin Debian GNU/Linux [21] helpon päivitettävyyden ja siirrettävyyden takia. Myös pakettihallinnan mahdollistama helppo ohjelmien ja kirjastojen asentaminen ja poistaminen oli yhtenä perusteena valinnalle. HTTP-palvelimena on käytössä Apache [22] SSL- ja PHP-moduulien kanssa. Kyseiset moduulit mahdollistavat dynaamisesti muuttuvan sisällön tarjoamisen sekä salatun että salaamattoman yhteyden kautta. Tietokantapalvelimena toimii MySQL [23], jonka hallinta on mahdollista suoraan PHP-tulkin sekä komentorivipohjaisten ohjelmien kautta. Tietokannan avulla kaikki oleellinen tieto voidaan tallentaa yhteen paikkaan, josta sen hakeminen esimerkiksi PHP-pohjaisen sivun kautta on käyttäjän kannalta mahdollisimman huomaamatonta.

Sarjaportissa sijaitsevan matkapuhelimen hallintaan käytetään Kannel-projektin [24] tarjoamaa WAP- ja SMS-yhdyskäytäväratkaisua. Kannel tarjoaa muille ohjelmille rajapinnan, jonka kautta halutussa muodossa olevia tekstiviestejä voidaan lähettää riippumatta tavasta, jolla laitteisto on yhteydessä matkapuhelinverkon viestikeskukseen. Testiympäristössä käytetty matkapuhelin edustaa yksinkertaisinta ja samalla testausmittakaavassa halvinta ratkaisua. Kannel pystyy kuitenkin myös keskustelemaan verkon yli suoraan operaattorin viestikeskuksen kanssa, mikäli kyseinen mahdollisuus on olemassa, sillä Kannel tukee kaikkien yleisimpiä viestikeskusten kanssa käytettyjä protokollia. Suoraan viestikeskuksen kanssa kommunikointi edellyttää kuitenkin sopimuksen tekemistä operaattorin kanssa, eikä kyseinen sopimus välttämättä ole hinnalta sopiva, mikäli tekstiviestejä lähetetään vain satunnaisesti. Toisaalta suora viestikeskuksen käyttäminen mahdollistaa tekstiviestin lähetys- ja toimitustietojen luotettavan saatavuuden.

Moduuleina hallintasovelluksen kehityksessä ja testauksessa on käytetty Nokia 12 ja Siemens TC45 moduuleja. Nokia 12 on kiinnitetty 1CQ-testilevyyn, minkä avulla moduulin tilan seuraaminen on yksinkertaisempaa testilevyn valokoodien ansiosta. Testilevyn kautta on myös mahdollista käyttää moduulin tarjoamia sarjaportteja, joiden kautta sovellusten siirto itse moduuliin on myös mahdollista.

Testilevy sisältää niin ikään moduulin tarvitseman virtaliitännän ja korttilukijan SIM-kortille. Siemens TC45 on puolestaan kiinni Tactus-kotelossa [25], mikä tarjoaa moduulille vastaavat liitännät kuin Nokian 1CQ-testilevy.

6.2 Testien sisältö

Testauksessa pyrittiin löytämään mahdollisia laitteiston ja hallintasovelluksen toimintaa liittyviä ongelmatilanteita ja havainnoimaan laitteiden käyttäytymisen säännöllisyyttä GPRS-verkossa. Käytetty Linux-alusta oletettiin toimivaksi, joten siihen liittyviä testejä ei suoritettu. Moduuleissa käytettiin kahden eri operaattorin liittymiä liittymä- ja operaattorikohtaisten erojen havainnoimiseksi. Aluksi testit keskittyivät CORBA-yhteyden toimivuuteen yksittäisen moduulin kanssa. Tämän jälkeen testausta laajennettiin kattamaan useamman moduulin samanaikainen päivitys.

Rajallisen moduulimäärän takia useamman laitteen päivittämisessä jouduttiin rajoittumaan kahdella moduulilla testaamiseen. Tämä tapahtui joko päivittämällä molempia moduuleja samaan aikaan tai päivittämällä kahta moduulia vuorotellen. Nokian rajapintatoteutuksesta johtuen vuorottelua oli mahdollista toistaa useampaan kertaan peräkkäin, mikä mahdollisti hallintasovelluksessa pidemmän päivitysjonon testaamisen.

Testauksen osana tarkkailtiin myös moduulin vikasietoisuutta joko heikentämällä moduulille tarjolla olevaa GSM-kenttää tai vaikuttamalla laitteen toimintaan kesken siirron. Näin pyrittiin simuloimaan mahdolliseen liikkuvaan laitteeseen liittyviä ongelmia, kuten tukiasemalta toiseen siirtymisiä ja mahdollisia yhteyden ajoittaisia katkeamisia laitteen siirtyessä katvealueille.

Nokian ja Siemensin moduulien päivitysmenetelmien erojen takia testauksessa keskityttiin etupäässä Nokian moduulin toiminnallisuuden tarkkailemiseen. CORBA-rajapinnan kautta Nokian moduuli kykeni tarjoamaan monipuolisempia virheilmoituksia vikatilanteessa, joten sen avulla moduulin käyttäytymistä oli

Siemensin moduuliin verrattuna helpompi seurata. Kumpikaan moduuleista ei tarjonnut erillistä liityntää, jonka kautta moduulin tilaa olisi voinut seurata reaaliaikaisesti samaan aikaan kun moduulille suoritettiin toimenpiteitä. Nokian moduulille tosin oli tarjolla sovellus, minkä avulla tilaseurantaa vastaava toiminnallisuus oli saatavilla. Kyseistä sovellusta ei kuitenkaan käytetty testauksen aikana, sillä sen havaittiin kuormittavan moduulia ja sovellusluonteen takia estävän toisen sovelluksen käynnistämisen tilalle seurannan katkeamatta.

Sovelluspakettien koot vaihtelivat muutamasta kilotavusta hieman yli sataan kilotavuun. Isoimmat sovelluspaketit sisälsivät useamman sovelluksen, joiden avulla paketin kokoa saatiin kasvatettua ja tällä tavoin siirtoon kuluva-aikaa pidennettyä. Testeissä käytetyt M2M-sovellukset sisälsivät toimivia, virheellisesti käyttäytyviä ja käynnistymiskyvyttömiä sovelluksia. Niiden avulla tarkkailtiin moduulin käyttäytymistä virhetilanteissa. Sovelluspakettien siirtämisen onnistumista testattiin myös vaihtelemalla verkon saatavuutta ja sen toimivuutta.

6.3 Testauksen tuloksia

6.3.1 Operaattorit

Testaus oli alun perin tarkoitus suorittaa kahden eri operaattorin liittymillä. Testauksen alkuvaiheissa kuitenkin jo ilmeni, että Soneran APN sisältää dokumentoimattoman palomuurin, mikä estää TCP-yhteyksien yhdistämisen silloin, kun yhteyttä yritetään avata internetin suunnalta moduulille päin. GPRS-verkon suunnalta internetiin puolestaan TCP-yhteyden avaamiselle ei ollut asetettu esteitä. Tästä rajoituksesta seurasi, että GPRS-yhteyden avulla laitteen päivittäminen Soneran verkossa ei ollut mahdollista, sillä CORBA edellyttää toisen TCP-yhteyden avaamista moduulin kanssa kommunikointia varten. CORBA:n protokollamääritysten uudemmat versiot sisältävät tuen kaksisuuntaisen TCP-yhteyden käyttämiselle, mutta tätä ominaisuutta ei ollut Nokia moduulissa toteutettu, vaikka sen ilmoittama CORBA:n versio näin antoi olettaa.

Soneran liittymien sijaan siirryttiin käyttämään DNA:n liittymiä, joissa ei ilmennyt rajoituksia yhteyksien avaamisissa niin kauan kuin kohdeportti pidettiin suurempana kuin 1024. Testauksen kannalta tämä ei muodostunut ongelmaksi, sillä yhteysportti oli sekä moduulissa että palvelimessa vapaasti määritettävissä. Ongelmana oli kuitenkin edelleen Nokian moduulin CORBA-toteutus.

6.3.2 CORBA

Moduulin dokumentaation mukaan M2M-sovelluksen siirron aluksi moduuli asetetaan ensin vastaanottavaan tilaan ja sovellusta lähetetään moduulille sen ilmoittaman kokoisina paketteina kunnes koko tiedosto on siirretty. Testikäyttöön hankitussa moduulissa kuitenkin ilmeni, että se oli oletusasetuksiltaan tarkoitettu käytettäväksi Nokian lopetetun M2M Gateway -kokonaisuuden kanssa. Moduuli ei tässä tilassa ollessaan tunnistanut kaikkia lähetettyjä CORBA-komentoja ja kieltäytyi siirtämästä sovelluksia halutulla tavalla. Tämän ongelman ehkäisemiseksi hallintasovellus ohjelmoitiin tarkistamaan moduulin käyttämä tila ja muuttamaan sitä ennen sovelluksen siirron aloittamista. Haittapuolena tällä menettelyllä oli kuitenkin se, että tilan vaihtamiseksi moduuli jouduttiin käynnistämään uudelleen ja tämä samalla tarkoitti uuden yhteyden avaamista moduulin käynnistyttyä. Moduuli ei kuitenkaan ollut valmis käsittelemään herätysviestejä heti uudelleenkäynnistyksen jälkeen, joten viestin lähettäminen toteutettiin hieman viivästetysti halutun uuden yhteyden aikaansaamiseksi.

Yhteyttä avattaessa moduulille lähetettävä viesti käytti GIOP:n versiota 1.2. Yhteyden muodostuttua moduuli ei kuitenkaan osannut käsitellä kaikkia versioon 1.2 sisältyviä komentoja. Tämän ongelman kiertämiseksi Linux-puolella käytetty CORBA-rajapinta OmniORB käskettiin käyttämään korkeintaan versiota 1.0 ja olemaan tekemättä ylimääräisiä tarkistuksia, täten minimoiden käytettyjen komentojen määrä. Näiden muutosten jälkeen moduulin kanssa kommunikointi eteni M2M-sovelluksen siirtoon asti ongelmitta.

Hallintasovelluksen toteutuksen ja testauksen aikana moduulin CORBA-rajapinnassa ilmeni muitakin ongelmia. Dokumentaation perusteella epäonnistuneiden CORBA-komentojen piti tuottaa poikkeuksia, joista olisi ollut mahdollista selvittää virheen syy. Käytännössä kuitenkin ilmeni, että moduulin CORBA-toteutus ei ainakaan OmniORB:n kanssa antanut dokumentaation mukaisia poikkeuksia virhetilanteissa, jolloin virhetilanteiden syyt jouduttiin päättämään tapahtumakohdan perusteella. Sovellusten pysäyttämisen yhteydessä moduuli saattoi pysäytyksen kuitaamisen sijaan katkaista yhteyden, jolloin yhteys jouduttiin avaamaan uudestaan, mikäli moduulille haluttiin vielä tehdä jotain pysäyttämisen jälkeen. Sarjakaapeliyhteyttä tarkistuksen apuna käyttäen sovellukset kuitenkin aina pysähtyivät, vaikka yhteys pysäyttämisen jälkeen saattoikin sovelluksen koosta riippuen katketa.

6.3.3 Sovellukset

Toimivien sovellusten lisäksi moduulia testattiin myös toimimattomilla tai muuten rikkiäisillä sovelluksilla. Moduuli kykeni tunnistamaan väärällä tavalla paketoitua sovelluspaketteja ja ilmoittamaan siirron yhteydessä virheestä, mutta virheellisesti toimivat sovellukset ilmenivät ongelmallisiksi. Yhtenä testitapauksena oli sovellus, joka pyrki varaamaan käyttöönsä isomman muistialueen kuin moduulilla oli tarjota. Lopputuloksena oli sovelluksen pysähtyminen heti käynnistyksen jälkeen. Vaikka moduulin sovelluksia käynnistävä funktio vaikutti tekevän jotain tarkistuksia käynnistyksen jälkeen, ei se kuitenkaan tunnistanut, ettei käynnistetty sovellus todellisuudessa jäänyt käyntiin. Varmistuksen saamiseksi jokaisen siirretyn sovelluksen tila jouduttiin tarkistamaan käynnistämisen jälkeen erikseen.

Sovelluksella oli myös mahdollista saada moduuli käyttökeltomaan tilaan useiksi minuuteiksi. Moduuliin ei ollut mahdollista saada yhteyttä, mikäli sille lähetetty sovellus oli ohjelmoitu käynnistämään moduuli uudestaan heti sovelluksen käynnistyttyä. Moduulin käynnistymisen yhteydessä ei ole riittävän pitkää viivettä ennen sovelluksen käynnistämistä, joten moduulille ei ollut

mahdollista lähettää komentoja niin kauan kun sovellus käynnistyi heti moduulin käynnistymisen jälkeen. Moduulin turvamekanismi kuitenkin poisti sovelluksen käynnistettävien sovellusten listalta sovelluksen käynnistettyä moduulia riittävän moneen kertaan uudestaan, minkä jälkeen moduuliin oli jälleen mahdollista saada yhteys.

Sovellusten sammuttamisen pitäisi olla yksinkertainen tapahtuma, mutta senkin kanssa ilmeni satunnaisesti paljon muistia käsittelevien sovellusten kanssa ongelmia. Joidenkin sovellusten kanssa moduuli saattoi sammuttamisen jälkeen katkaista verkkoyhteyden, vaikka niin ei pitäisi tapahtua. Tästä seurasi, ettei sammuttamisen onnistumisesta kertovaa paluuviestiä vastaanotettu. Uuden yhteyden avaamisen jälkeen sovellus oli kuitenkin joka kerta sammunut. Sovellus ei sammutuksen yhteydessä pysty enää suorittamaan komentoja, joten sovelluksella ei pitäisi olla mahdollisuutta katkaista verkkoyhteyksiä sammumisen aikana. Sammumisen tarkkailua varten luodut sovellukset eivät pystyneet sammumisprosessiin vaikuttamaan. Sen perusteella ei siten ollut pääteltävissä, mistä yhteyden katkeaminen johtui, varsinkin kun ongelma ei esiintynyt samaa sovellusta käytettäessä jokaisella kerralla.

Sovellukset eivät saa moduulilta tietoa päivityksissä käytetyistä CORBA-yhteyksistä, vaikka yhteys käyttää samaa verkkoyhteyttä, mikä sovelluksillekin on tarjolla. Tämä aiheutti tilanteen, jossa sovelluksen oli toisinaan mahdollista häiritä moduulin hallinnointia hallintasovelluksen kautta. Yksinkertaisimmillaan riitti, että sovellus käski moduulia kirjautumaan ulos verkosta ja etsimään kuuluvuusalueella olevia operaattoreita, minkä seurauksena auki oleva CORBA-yhteys katkesi. Hallintasovelluksella siirrettäväksi tarkoitettujen sovellusten pitäisi täten ottaa huomioon mahdollisten päivitysyhteyksien mahdollisuus. Koska moduuli ei kuitenkaan välitä sovellukselle tietoa muista yhteyksistä, voi sovellus lähinnä pyrkiä välttämään verkkoyhteyksiin vaikuttavien toimintojen käyttämistä.

6.3.4 Virhetilanteista toipuminen

Moduulin tiedonsiirtoyhteyden toimivuutta testattiin vaihtelemalla tarjolla olevaa GSM-kenttää. Käytännössä tämä toteutettiin lyhentämällä moduulin antennin pituutta pala kerrallaan, kunnes moduulilla ei enää ollut verkkoyhteyttä käytettävissä. Toimistoympäristössä lyhytkin antenni riitti riittävän kentän aikaansaamiseen. Koko antennin poistaminen kuitenkin esti verkkoyhteyden saamisen.

Yhteyden vikasietoisuutta testattiin poistamalla antenni kesken siirron ja kytkemällä antenni tietyn ajan kuluttua takaisin, jolloin siirron jatkumisesta voitiin todeta yhteyden palautuneet. Katkon kestoa pidennettiin viiden sekunnin välein kunnes päästiin tilanteeseen, jossa tiedonsiirto ei antennin palauttamisen jälkeen enää jatkanut. Järjestely toistettiin useampaan kertaan tulosten varmistamiseksi. Lopputuloksena havaittiin, että yhteyden palautumistodennäköisyys heikkenee sen jälkeen kun katko oli kestänyt yli 15 sekuntia. Myös yhteyden palautumisnopeus vaihteli muutamasta sekunnista kymmeneen sekunteihin tämän jälkeen, mikä saattaa johtua moduulin verkkoja hakevien toimintojen ajoituksista. Yli 30 sekunnin kestäneistä katkoksista palautuminen oli niin vaihtelevaa, ettei siitä voinut tehdä tehdyllä testimäärällä luotettavia johtopäätöksiä. Tämän seurauksena käytettyyn CORBA-toteutukseen asetettiin rajoitus, joka katkaisi yhteyden, jos aktiivisuutta ei havaittu 30 sekuntiin.

CORBA-rajapinnan ongelmia lukuun ottamatta moduuli suoriutui hyvin muista häiriötilanteista. Mikäli moduulista katkaistiin virrat kesken tiedonsiirron, kykeni se yhä käynnistämään itsensä toimintakuntoon virran palattua. Tässä tilanteessa siirrosta kesken ollut tiedosto poistui moduulin muistista. Moduuli ei myöskään jäänyt täysin käyttökelvottomaan kuntoon, mikäli siihen lähetettiin ja käynnistettiin sovellus, joka käynnisti moduulia kerta toisensa jälkeen uudelleen. Moduuli esti sovelluksen käynnistämisen keskimäärin kymmenen minuutin jälkeen ja palasi tämän jälkeen normaalin toimintakuntoon, jolloin moduuliin saatiin taas yhteys ja virheellisesti toimiva sovellus voitiin poistaa.

Moduulin hallinnoimista ja sovellusten siirtämistä käytettiin sekä CSD- että GPRS-yhteyden avulla. CSD-yhteyttä käytettäessä ilmeni, että moduuli ei yritä avata yhteyttä uudestaan, mikäli CSD:n käyttämä soittosarja on varattu. Moduuli ei niin tapahtuessa myöskään lähetä virheviestiä, joten hallintasovelluksen tulee kyseisessä tapauksessa lähettää moduulille uusi herätysviesti. Ongelmaksi muodostuu kuitenkin helposti tekstiviestin perillemenossa tapahtuva mahdollinen viive, jolloin ei voi varmuudella sanoa, johtuuko moduulin vastauksen puute siitä, että soittosarja on ollut varattuna vai siitä, ettei tekstiviesti vielä ole saavuttanut moduulia.

6.3.5 Suorituskyky

Hallintasovelluksen tietokannan lokiin kirjautui automaattisesti merkintä hallintasovelluksen lähettäessä yhteyden avaavan tekstiviestin ja toinen merkintä moduulin ottaessa yhteyttä palvelimeen. Lokista poimittiin satunnaisesti valittuna kuukauden ajalta 60 molempaa yhteystyyppiä edustavaa merkintää. Koska lokimerkinnät sisälsivät myös aikaleiman, oli aikaleimojen erotuksesta mahdollista laskea se aika, mikä moduulilta kului yhteydenottoon (taulukko 2). GPRS- ja CSD-yhteydet oli eroteltavissa lokiin kirjatun yhteyden IP-osoitteen perusteella. DNA:n GPRS-yhteydet käyttivät aina omaa IP-osoitealuettaan, joten niiden erottelu Soneran CSD-soittosarjan IP-osoitealueesta oli mahdollista.

Taulukko 2: Yhteydenottoon kuluva aika

	minimi	maksimi	keskiarvo
GPRS	18 s	65 s	20,6 s
CSD	33 s	99 s	35,7 s

Tuloksista oli havaittavissa, että GPRS-yhteyttä käytettäessä laite vastaa herätysviestiin keskimäärin 15 sekuntia CSD-yhteyttä nopeammin. CSD-yhteyttä käytettäessä moduuli joutuu soittamaan modeemisoittosarjaan ja käsittelemään vastapään kanssa ennen kuin laite pystyy avaamaan yhteyden vastataksaan.

palvelimelle, jolloin aikaa kuluu GPRS-yhteyttä enemmän. Testisarjassa ilmeni molemmilla yhteystyypeillä muutamia selvästi hitaampia yhteydenottoja. Nämä poikkeamat johtuvat oletettavasti herätysviestin toimittamisessa tapahtuneesta viiveestä operaattorin verkossa. Laitteiden päivittäminen ei kuitenkaan ole aikakriittistä, joten variaatiot yhteydenoton kestossa eivät ole haitaksi hallintasovelluksen toiminnalle.

7 YHTEENVETO

Työn aihetta valitessa vallinnut käsitys GPRS-verkossa tapahtuvan konekommunikaatio tiedonsiirron yksinkertaisuudesta osoittautui pian työn alkamisen jälkeen harhaksi. GPRS-verkossa itsessään ilmeni eri operaattoreiden asettamia rajoituksia, jotka häiritsivät tiedonsiirtoa, eivätkä itse laitteetkaan vaikuttaneet aivan loppuun asti ajatelluilta tuotteilta. Tässä mielessä ei siis ole ihme, ettei konekommunikaatio ole saavuttanut sitä suurta suosiota, mitä sille monet tahot ovat ennustaneet.

Nykyisellään konekommunikaatio tarkoittaa lähinnä langallisen yhteyden korvaamista langattomalla. Laitteet ja käytettävissä olevat verkkoyhteydet rajoittavat järjestelmien mahdollisuuksia. Langattomien laitteiden etuna on nopea asennettavuus ja siirrettävyys. Laitteiden kehittyessä ja käyttäjien löytäessä uusia käyttötarkoituksia on kuitenkin odotettavissa, että konekommunikaatio tulee vähitellen yleistymään.

Yksi merkittävistä puutteista koskee tällä hetkellä laitteiden dokumentaatiota ja niiden käyttöönottoa. Pahimmillaan dokumentaatiota ei ole missään helposti saatavilla tai se sisältää dokumentoituna ominaisuuksia, jotka itse laitteesta kuitenkin puuttuvat. Tätä ongelmaa ei yhtään helpota se, ettei konekommunikaatioon varsinaisesti ole mitään standardia, mikä on johtanut laitevalmistajat luomaan omia keskenään epäyhteensopivia ratkaisuja.

Hallintasovelluksen puute on ajanut laitteiden käyttäjät tilanteeseen, jossa pienen laitemäärän hallinta on vielä mahdollista manuaalisesti toteutettuna. Markkinoilta puuttuvat toistaiseksi suuren laitemäärän hallintaan soveltuvat työkalut.

Työssä kehitetty hallintasovellus onnistui täyttämään sille asetetut vaatimukset ja saatiin toimiva ja helppokäyttöinen kokonaisuus. Työn aikana selvitettiin useita eri vaihtoehtoja tiedonsiirron toteuttamiseksi moduulille ja todettiin CORBA-yhteyden käyttäminen toimivaksi ratkaisuksi.

Hallintasovellus osoittautui niin toimivaksi, että se päätettiin tuottaa ja se on toimitettu koekäyttöön. Työtä on tarkoitus jatkaa laajentaen sovelluksen tukea useampien laitevalmistajien laitteille. Sovelluksen ominaisuuksia tullaan lisäämään mm. koekäyttöpalauteen perusteella. Lisäksi tutkitaan mahdollisuuksia muidenkin kaupallisten tietoliikennestandardien, kuten WLAN:n ja Bluetooth:n, tukemiseen sekä tulevien uusien tietoliikennetarkaisujen käyttämiseen tiedonsiirron alustana.

LÄHTEET

- [1] Tekes. M2M Evoluutio, versio 2.0, 14.2.2005.
- [2] Schumacher, Jenni. Mobiili vaatii uskallusta. Tietoviikko, 2004. No. 34, s. 12-13. ISSN 0359-8543
- [3] Rinnemaa, Tommi. Langaton viestintä valtaa automaattit. Tekniikka & Talous, 2004. No. 37, s. 2. ISSN 0785-997X
- [4] Sun Microsystems. Over The Air User Initiated Provisioning, 2001
- [5] Siemens. TC45 Datasheet, 2003, A31001-W25-A202-X-7600
- [6] Siemens. TC45 Java User's Guide 01, 2003
- [7] Sony Ericsson. M2mpower Press Release, 12.3.2003
- [8] Sony Ericsson. GR47/GR48 Technical Description, 2003
- [9] Sony Ericsson. M2mpower Application Guide, LZT 123 7588 R1A
- [10] Nokia. Nokia 12 Press Release, 16.6.2003
- [11] Nokia. Nokia 12 GSM Module Product Specification 2.0, 2004, 9231512
- [12] Nokia. Nokia 12 GSM Module Software Developer's Guide 1.0, 2004, 9231710
- [13] Object Management Group. Common Object Request Broker Architecture: Core Specification 3.0.3, 2004
- [14] SyncML Initiative. SyncML Sync Protocol 1.1, 2002
- [15] Levkowitz H.; Vaarala S. RFC 3519 - Mobile IP Traversal of Network Address Translation (NAT) Devices, 2003
- [16] Kitsos, P.; Sklavos, N.; Koufopavlou, O. An End-to-End Hardware Approach Security for the GPRS, IEEE MELECON, 2004
- [17] Mitnick, Kevin D. The Art of Deception: Controlling the Human Element of Security, 2002, Wiley, ISBN 0471237124
- [18] Michael, Meyer. TCP Performance over GPRS, IEEE WCNC, 1999
- [19] Nokia. Nokia M2M Platform Software Developer's Guide 1.0, 2003, 9311036
- [20] Toivola, Teemu. M3 Software Requirements Specification, 2004, Sofnetix

- [21] Debian – The Universal Operating System [www-dokumentti] Saatavissa <http://www.debian.org/>. Viitattu 30.12.2004.
- [22] The Apache HTTP Server Project [www-dokumentti] Saatavissa <http://httpd.apache.org/>. Viitattu 3.1.2005.
- [23] MySQL: The world's most popular open source database [www-dokumentti] Saatavissa <http://www.mysql.com/>. Viitattu 30.12.2004.
- [24] Kannel: Open Source WAP and SMS gateway [www-dokumentti] Saatavissa <http://kannel.org/>. Viitattu 6.1.2005.
- [25] Tactus Telemetry Solutions [www-dokumentti] Saatavissa <http://www.tactustelemetry.com/>. Viitattu 5.1.2005.