



TAMPEREEN TEKNILLINEN YLIOPISTO  
*Sähkötekniikan koulutusohjelma*

VESA JÄRVINEN  
LANGATON SENSORIVERKKOSOVELLUS  
KOTIYMPÄRISTÖÖN  
Diplomityö

Tarkastaja: Karri Palovuori  
Tarkastaja ja aihe hyväksytty  
Tiedekuntaneuvoston kokouksessa  
9.11.2011

# Tiivistelmä

TAMPEREEN TEKNILLINEN YLIOPISTO

Sähkötekniikan koulutusohjelma

JÄRVINEN, VESA: Langaton sensoriverkkosovellus kotiympäristöön

Diplomityö, 78 sivua, 8 liitesivua

Toukokuu 2012

Pääaine: Elektroniikan laitesuunnittelu

Tarkastaja: Karri Palovuori

Avainsanat: Langaton sensoriverkko, sensoriteknologia, langaton viestintä, MAC-protokolla

Langattomat sensoriverkot voivat tuoda suuria etuja verrattuna tavallisiin langallisiin sensoreihin. Tässä työssä esitellään kotiympäristöön rakentamani ja suunnittelemani sensoriverkkosovelluksen. Samalla tehdään katsaus minkälaisia muita sovelluksia langattomille sensoriverkoille on olemassa, minkälaisia langattoman viestinnän protokollia niissä käytetään ja tutustaan lyhyesti rakentamassani työssä käytettyihin sensoriteknologioihin.

## Abstract

TAMPERE UNIVERSITY OF TECHNOLOGY

Masters's Degree Programme in Electrical Engineering

JÄRVINEN, VESA: Wireless sensor network application to a home environment

Master of Science Thesis, 78 pages, 8 Appendix pages

May 2012

Major: Electronic Devices Design

Examiner: Karri Palovuori

Keywords: Wireless sensor networks, sensor technology, wireless communication, MAC-protocol

Wireless sensor networks can bring great advantages over traditional wired sensors. In this work I present a wireless sensor network application to a home environment, which I built and designed. Also I will be taking a look at what kind of other applications have been developed for wireless sensor networks and what kind of wireless communication protocols have been used on them. Finally I will be taking a brief look at the different kinds of sensor technologies that have been used on the sensor network that I have built.

## **Alkusanat**

Tämä diplomityö tehtiin Tampereen teknillisen yliopiston Elektroniikan laitokselle syksyn 2011 ja kevään 2012 välisenä aikana. Työn aihe oli mielenkiintoinen, sillä langattomien sensoriverkkojen rooli tulevaisuudessa tulee kasvamaan. Työn tarkastaja toimi Karri Palovuori, jota haluan kiittää ohjauksesta ja neuvoista.

# Sisällys

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Sensoriverkkojen ja sensorisolmujen rakenne . . . . .	2
1.2	Langattomien sensoriverkkojen haasteet ja rajoitteet . . . . .	4
1.3	Oman sensoriverkkosovelluksen esittely . . . . .	5
<b>2</b>	<b>Sovelluksia langattomille sensoriverkoille</b>	<b>6</b>
2.1	Rakennuksien kunnan tarkkailu . . . . .	6
2.1.1	Prototyyppejä . . . . .	7
2.2	Tarkkuusmaatalous . . . . .	8
2.2.1	Prototyyppejä . . . . .	8
2.3	Liikenteen valvonta . . . . .	9
2.3.1	Prototyyppejä . . . . .	9
2.4	Terveystenhoito . . . . .	10
2.4.1	Prototyyppejä . . . . .	10
2.5	Elin- ja luonnonympäristöjen monitorointi . . . . .	12
2.5.1	Prototyyppejä . . . . .	13
<b>3</b>	<b>Langaton tiedonsiirto sensoriverkoissa</b>	<b>14</b>
3.1	Medium access control . . . . .	14
3.1.1	Kilpailuvapaa MAC . . . . .	14
3.1.2	Kilpailupohjainen MAC . . . . .	17
3.1.3	MAC-protokollat sensoriverkoissa . . . . .	20
3.2	Reititysprotokollat . . . . .	23
3.2.1	Datakeskiset protokollat . . . . .	25
3.2.2	Hierarkiset protokollat . . . . .	28
3.2.3	Sijaintipohjaiset protokollat . . . . .	32
<b>4</b>	<b>Sensortechnologiasta</b>	<b>35</b>
4.1	Kosteussensorit . . . . .	35
4.1.1	Kapasitiiviset kosteussensorit . . . . .	36
4.1.2	Resistiiviset kosteussensorit . . . . .	39
4.1.3	Lämmönjohtavuus kosteussensorit . . . . .	39
4.2	Valosensorit . . . . .	40
4.3	Lämpösensorit . . . . .	41
<b>5</b>	<b>Oman sensoriverkon suunnittelu</b>	<b>44</b>
5.1	Piirikytkentöjen suunnittelu . . . . .	44
5.1.1	Päämoduuli . . . . .	44
5.1.2	Kannettavat solmut . . . . .	48
5.2	Piirilevyjen suunnittelu . . . . .	51
5.2.1	Päämoduuli . . . . .	51
5.2.2	Kannettavat solmut . . . . .	53
5.3	Ohjelmisto . . . . .	56
5.3.1	Yleiskatsaus reititysprotokollaan . . . . .	56
5.3.2	Radion ominaisuuksista . . . . .	58
5.3.3	Protokollassa käytetyt viestit . . . . .	60
5.3.4	Kannettavien solmujen ohjelmisto . . . . .	61

5.3.5 Päämoduulin ohjelmisto . . . . .	69
<b>6 Johtopäätökset</b>	<b>74</b>
<b>Lähteet</b>	<b>76</b>
<b>A Liite 1: Päämoduulin kytkentäkaavio</b>	<b>79</b>
<b>B Liite 2: Päämoduulin alapuolen valotusmaski</b>	<b>80</b>
<b>C Liite 3: Päämoduulin yläpuolen valotusmaski</b>	<b>81</b>
<b>D Liite 4: Päämoduulin alapuolen osasijoittelu</b>	<b>82</b>
<b>E Liite 5: Päämoduulin yläpuolen osasijoittelu</b>	<b>83</b>
<b>F Liite 6: Sensorisolmun 2 kytkentäkaavio</b>	<b>84</b>
<b>G Liite 7: Sensorisolmun 3 alapuolen asemapiirustus</b>	<b>85</b>
<b>H Liite 8: Sensorisolmun 3 yläpuolen asemapiirustus</b>	<b>86</b>

## Termit ja niiden määritelmät

Ad Hoc	Tiettyyn tarkoitukseen suunniteltu.
AD-muunnin	Analogia-digitaalimuunnin. Muuttaa analogisen signaalin digitaalseksi.
CSMA	Eräs MAC-protokolla. (Carrier Sense Multiple Access)
CSMA/CA	CSMA-protokollan muunnelma. (Carrier sense multiple access with collision avoidance)
CTS	Kättelysignaali protokollassa. (clear to send)
EKG	Sydänsähkökäyrä.
FIFO	Jono, jossa ensimmäiseksi saapunut data käsitellään ensin. (First In, First Out)
GSM	Maailmanlaajuisesti käytetty matkapuhelinjärjestelmä. ((Global System for Mobile Communications)
I2C	Kahdensuuntainen tiedonsiirtoväylä. (Inter-integrated circuit)
ISM-kaista	Maailmanlaajuinen radiotaajuuskaista. (Industrial, Scientific and Medical)
LCD	Nestekidenäyttö (Liquid Crystal Display) .
LPU	Kannettava laite UbiCom-järjestelmässä. (Local Processing Unit)
MAC-protokolla	Kommunikointiprotokollan osa, joka vastaa kanavalle pääsystä. (medium access control)
MOSFET	Transistorityyppi. (metal-oxide-semiconductor field-effect transistor)
NTC	Fyysinen suure laskee lämpötilan nousetessa (negative temperature coefficient)
RTC	Reaaliaikakello. (Real Time Clock)
RTD	Vastuslämpötila-anturi. (resistance temperature detector)
RTS	Kättelysignaali protokollassa. (ready to send)
SIM	Integroitu piiri, jonka avulla tunnistetaan käyttäjiä matkapuhelinverkoissa. (subscriber identity module)
SPI	Kahdensuuntainen tiedonsiirtoväylä. (Serial Peripheral Interface Bus)
TDMA	Aikajakokanavointi, eräs kanavanvaraustekniikka. (Time division multiple access)
USART	Sarjaliikenteen lähetys- ja vastaanotto piiri, joka muuntaa rinnakkaisdatan sarjadataksi. (Universal Synchronous/Asynchronous Receiver-

$A$	Transmitter)
$C$	Pinta-ala.
$d$	Kapasitanssi.
$p$	Välimatka.
$P_R$	Veden osapaine.
$P_T$	Vastaanotettu teho.
$r$	Lähetetty teho.
$RH$	Etäisyys
$\Delta\omega$	Suhteellinen kosteus.
$\epsilon_0$	Odotetun ja mitatun värähtelytaajuuden erotus.
$\epsilon_r$	Tyhjiön permittiivisyys.
$\omega_d$	Eristeen suhteellinen permittiivisyys.
$\omega_h$	Mitattu värähtelytaajuus.
	Odotettu värähtelytaajuus.

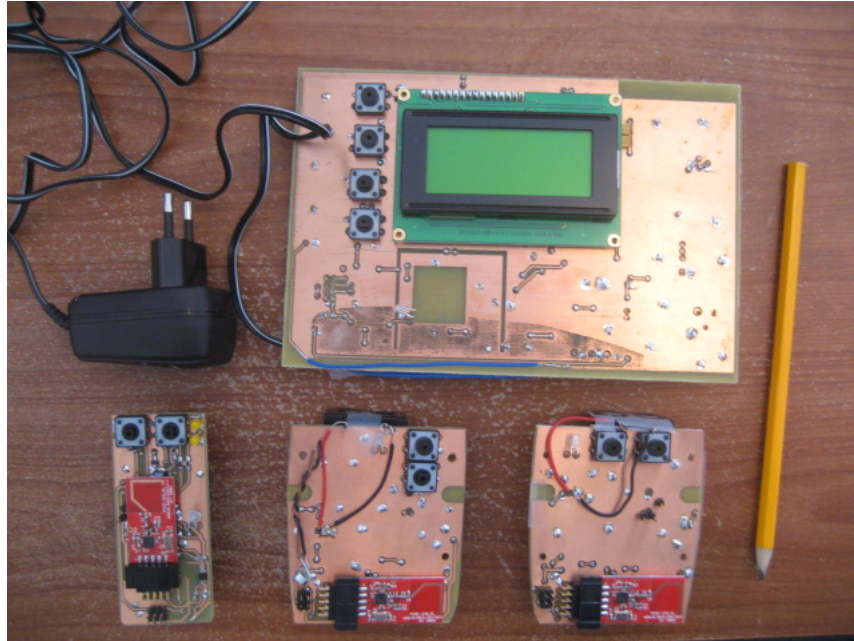


# 1 Johdanto

Sensoreiden käyttö rakennuksissa, laitteistoissa ja ympäristössä sekä sensoridatojen tehokas tiedonsiirto voi tuoda suuria hyötyjä yhteiskunnalle. Mahdollisia hyötyjä ovat muun muassa luonnonvarojen tehokas käyttö, parantunut teollisuuden tuotantokyky, liikenteen parempi sujuvuus ja rakennuksien lisääntynyt turvallisuus. Kuitenkin sensoreiden käyttöön liittyy useita ongelmia. Pitkät johdot voivat helposti katketa ja liittimet voivat mennä rikki. Pitkiin johtoihin liittyy myös suuret asennus- ja huoltokustannukset. Tämä voi puolestaan vähentää käytettävien sensoreiden määrää ja näin huonontaa saatavan datan laatua. Langattomat sensoriverkot voivat vähentää näitä kustannuksia helpolla asennuksella ja johtojen sekä liittimien puuttumisella.

Langattomat sensoriverkot ovat kokoelma pieniä *sensorisolmuja*(engl. sensornode), joilla on kyky langattomaan viestintään yleensä lyhyillä välimatkoilla. Langattomassa sensoriverkossa monet yksittäiset sensorisolmut monitoroivat yhteistoinnassa suurikokoista ympäristöä. Sensorisolmut reitittävät dataa usein *niulusolmulle*(engl. sinknode), joka kerää sensoriverkon datat ja analysoi sekä mahdollisesti reitittää sensoridatat eteenpäin. Ideaalisesti sensoriverkko on helposti skaalautuva, kuluttaa vähän energiaa, pystyy keräämään dataa nopeasti ja luotettavasti pitkällä aikavälillä sekä on halpa ostaa ja asentaa eikä vaadi paljoa huoltoa.

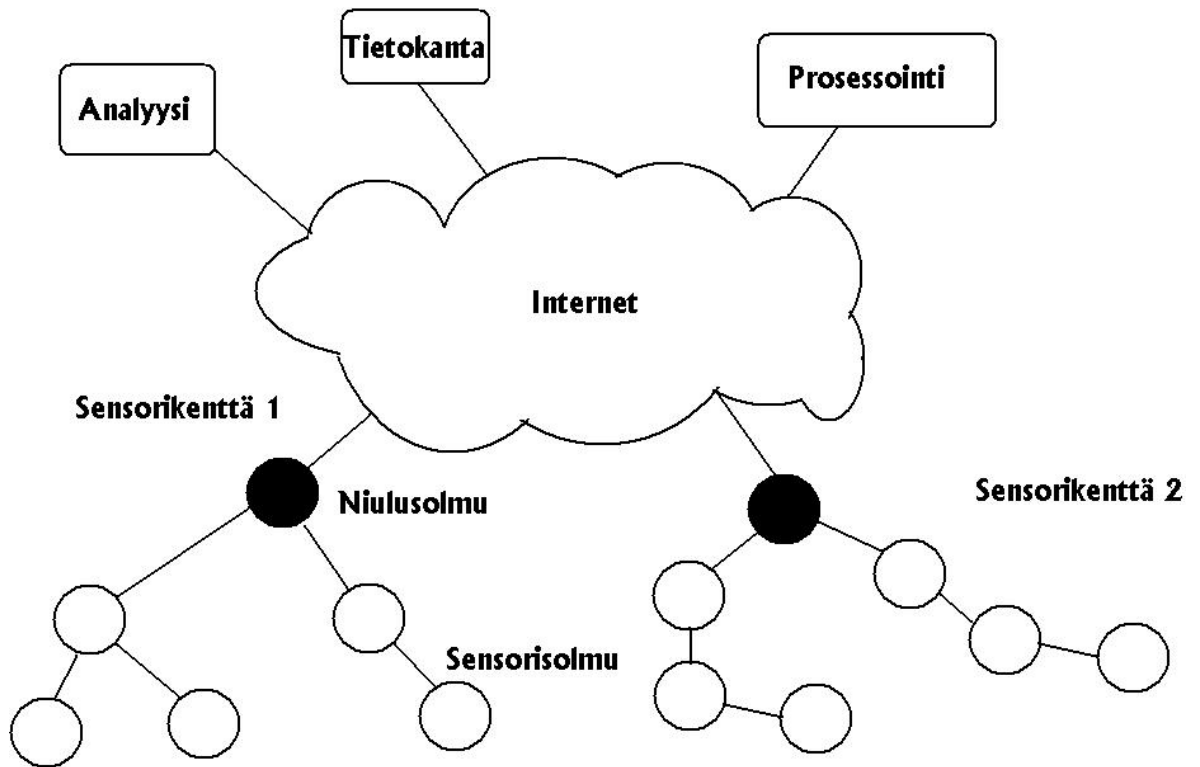
Tämän diplomityön tarkoituksena on suunnitella, rakentaa ja ohjelmoida koti-ympäristöön soveltuva langaton sensoriverkko. Sensoriverkkoon kuuluu päämoduuli eli niulusolmu, johon kaikki muut solmut reitittävät, sekä kolme kannettavaa solmua. Nämä on esitettyinä kuvassa 1.1. Samalla tehdään katsaus sensoriverkoissa käytetyistä teknologioista ja sensoriverkoille keksityistä sovelluksista. Kappaleessa 2 tutustaan moniin eri sovelluksiin, joita on sensoriverkoille keksitty. Kappaleessa 3 käsitellään erilaisia tapoja reitittää dataa langattomasti sensoriverkoissa. Itse sensorit, niiden käyttö ja oikean sensortyyppin valinta on tärkeä osa sensoriverkon suunnittelua. Kappaleessa 4 tehdään lyhyt katsaus eri sensorteknologioihin. Kappaleessa 5 käsitellään rakentamani sensoriverkon piirikytken, piirilevyn ja ohjelmiston suunnittelua.



***Kuva 1.1:** Rakennettu langaton sensoriverkko. Ylhäällä päämoduuli ja alhaalla kolme kannettavaa solmua.*

## 1.1 Sensoriverkkojen ja sensorisolmujen rakenne

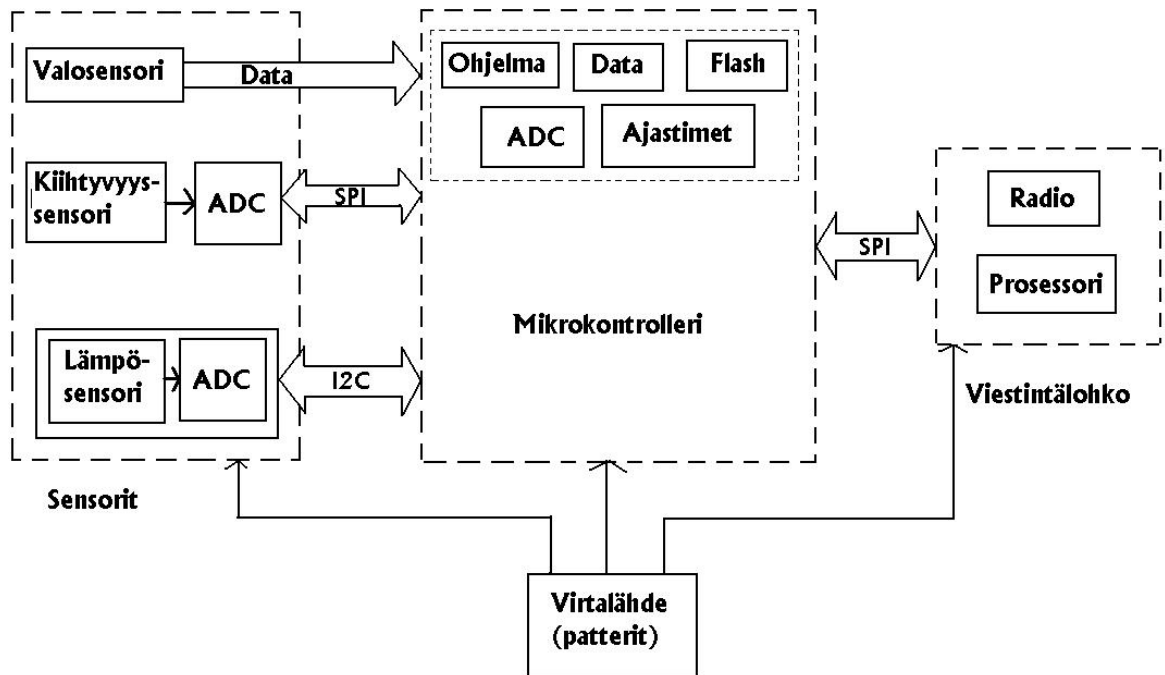
Kuvassa 1.2 on esitetty tyypillisen langattoman sensoriverkon rakenne. Sensoriverkko koostuu useista sensorisolmuista, jotka muodostavat yhdessä *sensorikentän*, joka kommunikoi yhden niulusolmun kanssa. Sensoriverkko voi koostua useasta eri sensorikentästä, jotka ovat yhdistettynä Internetin kautta toisiinsa. Internetin välityksellä sensoridatat voi tallentaa tietokantaan, prosessoida ja analysoida. Yksittäisen solmun tehtävänä on sensoroinnin ja data reitittämisen lisäksi myös verkon itsensä analysointia ja muiden nodien sensoridatojen yhdistäminen omiin sensoridatoihin.



*Kuva 1.2: Tyypillisen langattoman sensoriverkon rakenne.*

Yksittäisen sensorisolmun tyypillinen arkkitehtuuri on esitty kuvassa 1.3. Arkkitehtuuri voidaan jakaa neljään eri osaan: sensorit, prosessori, viestintälohko ja virtalähde. Sensoreiden ja prosessorin välinen rajapinta voi koostua esimerkiksi SPI- tai I2C-väylästä, jonka kautta digitaaliseen muotoon muutettu sensoridata siirretään prosessorille. Muunnos analogisesta signaaleista digitaaliseen muotoon voidaan tehdä sensorikomponentissa itsessään, erillisessä AD-muutajassa tai prosessorin sisäisellä AD-muuntajalla.

Myös viestintälohko voidaan liittää prosessoriin käyttäen esimerkiksi SPI-väylää. Viestintälohko on useimmiten sensorisolmun eniten energiaa kuluttava lohko, joten sen kuluttaman energian minimoimiseen kannattaa panostaa.



*Kuva 1.3: Tyypillisen sensorisolmun arkkitehtuuri.*

## 1.2 Langattomien sensoriverkkojen haasteet ja rajoitteet

Yksi tärkeimmistä rajoitteista sensoriverkoilla on niiden rajallinen energia. Sensorisolmut toimivat usein pattereilla, jotka täytyy ladata tai vaihtaa niiden kuluttua loppuun. Mutta joillekin solmuille kumpikaan vaihtoehto ei ole mahdollista, jonka seurauksena solmu jää pois käytöstä pattereiden loputtua. Joidenkin solmujen on pystyttävä toimimaan useita vuosia ilman pattereiden vaihtoa. Tämän takia on tärkeää suunnitella sensorisolmut energiatehokkaiksi. Iso osa sensorisolmun energiankulutuksesta kuuluu radiolle. Sensoriverkkojen MAC-protokollien (medium access control) yhtenä tärkeänä tarkoituksena onkin pitää radiot pois päältä, kun kommunikointia ei ole odotettavissa. MAC-protokollista langattomissa sensoriverkoissa käsitellään enemmän kappaleessa 3.1.

Tärkeä haaste on myöskin se, että sensoriverkkojen on kyettävä toimimaan itsenäisesti pitkiä aikoja. Sensorisolmujen on pystyttävä itsenäisesti muodostamaan verkon, jonka kautta ne reitittävät dataa niulusolmulle. Solmujen on pystyttävä itsenäisesti reagoimaan ja mukautumaan muuttuviin olosuhteisiin ilman ihmisten väliintuloa. Esimerkiksi mikäli jokin reititykseen käytetty solmu verkossa menee rikki, muiden solmujen on osattava löytää omatoimisesti uusi reitti niulusolmulle.

Langaton viestintä aiheuttaa myös haasteita sensoriverkon suunnittelijoille. Langattomat signaalit vaimentuvat nopeasti kulkiessaan ovien tai seinien läpi. Korkeataajuiset signaalit myös heijastuvat helposti eri pinnoista eivätkä näin pääse ete-

nemään. Vastaanotetun signaalin teho  $P_R$  riippuu kääntäen etäisyyden  $r$  neliöstä:

$$P_R = \propto \frac{P_T}{r^2} \quad (1.1)$$

, jossa  $P_T$  on lähetetty teho. Joten jos signaalin kulkema etäisyys kaksinkertaistuu, vastaanotettu teho pienenee neljännesosaan. Eli vastaanotettu teho pienenee nopeasti etäisyyden funktiona. Täten on järkevää jakaa yksi pidemmälle menevä lähetys useampaan eri lähetykseen, joiden yksittäiset kulkemat etäisyydet ovat lyhyempiä. Tämän hoitaa sensoriverkon MAC-protokolla. Langattoman viestinnän haasteellisuutta lisää se, että kaikki solmut eivät ole yhtä aikaa hereillä, vaan tehon säästämiseksi vain osat solmuista on hereillä tiettyyn aikaan.

Sensoriverkkojen suunnittelussa on tarkoituksena luoda halpoja, pieniä ja energiatehokkaita sensorisolmuja. Tästä syystä solmuilla on rajoitetun energian lisäksi myös rajoittunut laskentakyky ja datan tallennuskyky. Tämä on otettava huomioon esimerkiksi ohjelmistoa tehdessä. Solmujen ohjelmisto on usein oltava suhteellisen pieni ja mahdollisimman vapaa paljon laskentakykyä vaativista toimenpiteistä. Reititysprotokollaa suunniteltaessa on otettava huomioon, että solmun muistiin ei välttämättä mahdu kaikkien mahdollisien määränpäiden kattava reititystaulukko, vaan on ehkä tyydyttävä käyttämään taulukkoa vain lähimmistä naapureista.

### 1.3 Oman sensoriverkkosovelluksen esittely

Tämän diplomityön tarkoituksena on rakentaa kotiympäristöön tarkoitettu langaton sensoriverkko. Sensoriverkko koostuisi yhdestä päämoduulista ja kolmesta kannettava solmusta. Kannettavat solmut reitittävät sensoridataa päämoduuliin käyttäen 2,4 GHz:n radiomoduleita. Päämoduulissa sensoridatoja voi tarkastella LCD-näytön kautta. Päämoduuli toimisi verkkovirralla, mutta kannettavat solmut toimisivat paterilla. Myös päämoduuli sisältäisi sensoreita kuten valosensorin, lämpötilasensorin ja kosteussensorin. Jokainen kannettava solmu sisältäisi kaksi sensoria, mutta ne olisivat eri sensoreita eri solmuissa.

Päämoduuli sisältää myös GSM-moduulin, jonka kautta käyttäjä voi myös kommunikoida sensoriverkon kanssa. Käyttäjä voisi lähettää halutessaan tekstiviestin päämoduulille sisältäen yhden taulukon 1.1 komennoista. Päämoduuli vastaa komentoon lähettämällä käyttäjälle takaisin tekstiviestillä pyydytetyt datat. ”HAE\_NODEX”-käsky hakee solmun, jonka tunniste on X, kaikki sensoridatat ja lähettää ne tekstiviestinä käyttäjälle. Komento HAE\_LAMPO sen sijaan hakee kaikkien solmujen, joilla on lämpösensori, lämpötiladat ja lähettää ne käyttäjälle.

**Taulukko 1.1:** *Sensoriverkolle lähetettävät tekstiviestikäskyt.*

Komento	Toiminto	Esimerkkivastaus
HAE_NODEX	Hakee solmun X sensoridatat	“Node2: 23.1C 300 Lx”
HAE_LAMPO	Hakee kaikkien solmujen lämpötiladatat	“Node2: 23.1 C Node3: 22.0 C Node3: 22.0 C Node4: 21.0 C”

Tämän lisäksi yksi sovellus sensoriverkolle on esimerkiksi jääkaapin tilanteen valvonta. Mittaamalla valoisuutta voidaan päätellä, onko jääkaapin ovi jäänyt vahingossa auki. Jos jääkaapin ovi on auki, sen sisällä palaa lamppu valaisten jääkaappia ja valosensoria. Mikäli valosensori mittaa kohonneita valoisuusarvoja 2 minuuttia putkeen, solmu lähettää tästä varoituksen radiolla päämoduulille. Päämoduuli puolestaan varoittaa käyttäjää tästä lähettämällä tekstiviestin “Jaakaappi auki!” käyttäjän matkapuhelimeen. Käyttäjä voi laittaa tämän sovelluksen päälle painamalla solmun nappia.

Yksi sensorisolmuista sisältää myös mikrofoniin. Tätä voidaan käyttää esimerkiksi murtohälyttimenä. Solmun nappia painamalla solmu alkaa kahden minuutin viiveen jälkeen mittaamaan mikrofoniin avulla asunnossa kuuluvia ääniä. Mikäli asunnossa kuuluu tarpeeksi kovia ääniä, solmu lähettää tästä ilmoituksen päämoduulille, joka puolestaan lähettää tekstiviestin “Murtohälytys!” käyttäjän matkapuhelimeen.

## 2 Sovelluksia langattomille sensoriverkoille

Langattomille sensoriverkoille on kehitetty useita erilaisia sovelluksia. Jotkut niistä on vielä melko futuristisia, mutta useimmille löytyy jo nyt käytännön hyötyä. Sovelluksia on kehitetty muun muassa rakennuksien kunnan tarkkailuun, terveydenhuoltoon, maatalouteen, liikenteen valvontaan ja luonnonympäristöjen tutkimiseen. Tässä kappaleessa käydään läpi näitä sovelluksia ja samalla tarkastellaan millaisia teknisiä ratkaisuja niissä on käytetty. Lisäksi arvioidaan, kuinka hyvin kyseiset prototyypit ovat onnistuneet ja kerrotaan minkälaisia ongelmia niissä on kohdattu.

### 2.1 Rakennuksien kunnan tarkkailu

Erilaisten rakennuksien kunnan tarkastaminen on aikaisemmin tehty joko visuaalisella tarkastuksella, akustisilla menetelmillä, ultraäänellä tai tutkilla. Nämä ovat kalliita, paljon työtä vaativia menetelmiä, jotka vaativat isoja ja paljon tehoa vieviä hienostuneita laitteita. Langattomat sensoriverkot tarjoavat vaihtoehtoisen, kevyemmän ja halvemmän tavan monitoroida rakennuksien kuntoa[9, s.17].

On useita syitä, miksi langattomat sensoriverkot soveltuvat hyvin rakenteiden kunnan monitorointiin. Ensinnäkin langattomat sensorit voidaan sijoittaa paikkoi-

hin, joihin isompia langallisia sensoreita ei voi laittaa. Langattoman viestinnän takia on helpompaa sijoittaa useampia sensoreita samaan kohteeseen, jolloin voidaan yhdistää eri sensorien mittaustuloksia. Tämä helpottaa esimerkiksi paikallistamaan rakenteellisten vaurioiden sijaintia. Lisäksi ideaalisesti sensorien asentaminen ja huoltaminen ei vaadi rakennuksen normaalin toiminnan keskeytymistä [9, s.18].

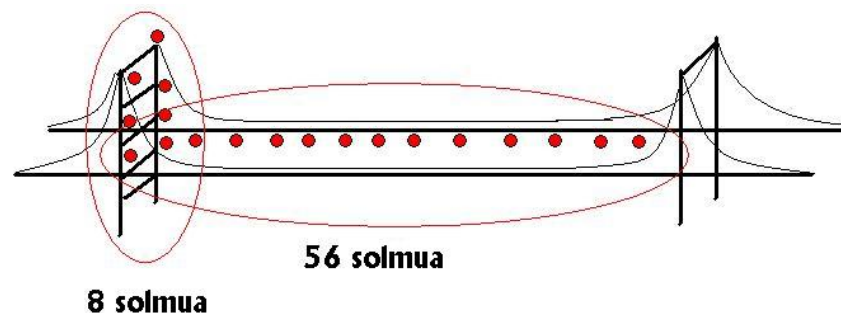
Maanjäristykset ovat yksi rasite rakennuksille, joiden vaikutuksia rakenteille usein halutaan tietää. Maanjäristykset aiheuttavat rakennuksissa värähtelyä yleensä alle kymmenien hertsien taajuudella. Tätä värähtelyä voidaan mitata esimerkiksi kiihtyvyyssensoreilla, kallistusmittareilla tai pietsosähköisillä sensoreilla. Vahinkojen havaitseminen voidaan tehdä vertaamalla rakennuksen odotettua eli tervettä värähtelytaajuutta  $\omega_h$  varsinaiseen mitattuun värähtelytaajuuteen  $\omega_d$ :

$$\delta\omega = \omega_h - \omega_d. \quad (2.1)$$

Mittaamalla värähtelytaajuuden ja sen poikkeaman odotetusta taajuudesta useammasta paikasta rakennuksessa voidaan arvioida vahingon sijainti [9, s.18-19].

### 2.1.1 Prototyyppejä

Esimerkkinä rakennuksien kunnan monitoroinnista langattomalla sensoriverkolla voidaan mainita Golden Gate -siltaan [20] asennettu prototyyppi sensoriverkosta. Golden Gate sijaitsee San Fransiscossa, jossa tapahtuu usein maanjäristyksiä siirroslinjan läheisyydestä johtuen. Täten myös Golden Gate -silta kokee mekaanisia rasituksia maanjäristyksien takia. 64 langatonta sensoria asennettiin Golden Gate -siltaan kuvan 2.1 mukaisesti. Tavoitteena oli mitata sekä maanjäristyksien että normaaliin rasituksen kuten tuulen aiheuttamia sillan värähtelyjä ja täten arvioida sillan kuntoa.



**Kuva 2.1:** Golden Gate -siltaan asennettiin 64 langatonta sensoria.

Kaikki 64 sensoria mittaavat synkronisesti sillan värähtelyjä 1 kHz:n taajuudella. Synkroninen mittaus oli haastavaa johtuen siitä, että jokaisella solmulla on oma itsenäinen kello. Kuitenkin prototyypissä onnistuttiin mittaamaan synkronises-

ti alle  $10\mu s$  aikaerolla. Prototyypissä päästiin noin  $30\mu G$  tarkkuuteen värähtelyjen kiihtyvyyksien mittauksessa.

Langaton sensoriverkko pystyttiin asentamaan ilman, että liikennettä sillan yli täytyi pysäyttää. Loppujen lopuksi sensoriverkon mittaamat datat vastasivat hyvin sillan teoreettisia malleja ja aikaisempia tutkimustuloksia sillasta.

Prototyyppiä testatessa kohdatiin mielenkiintoinen ongelma. Sensoriverkon datan siirtämistä varten oleva reititysprotokolla oli toiminut hyvin laboratorioolosuhteissa, mutta varsinaisessa kohteessa reititysprotokolla meni rikki. Tekijät arvelivat, että vilkkaan liikenteen aiheuttamat häiriöt olivat vian takana.

## 2.2 Tarkkuusmaatalous

Perinteisesti maataloudessa peltoja pidetään homogeenisina alueina, joilla ei ole paljoa eroa esimerkiksi kasvukyvyn, rikkaruohojen tai tuholaisien lukumäärän suhteen. Tällöin pelloille levitetään ravinteita, tuholaismyrkkyjä ja vettä tasaisesti. Kuitenkin oikeasti pellon eri osilla voi olla suuretkin erot ravinteiden tarpeille tai tuholaisien määrällä. Jos peltoa kuitenkin pidetään homogeenisena alueena, tämä voi johtaa ravinteiden tehottomaan käyttöön ja pellon tuottavuuden menetykseen.

Langattomilla sensoriverkoilla voidaan tunnistaa pellon ominaisuuksien eroavaisuudet eri osissa peltoa ja näin tarkemmin kohdentaa ravinteiden ja muiden resurssien käyttöä. Tämä paitsi vähentää resurssien tuhlausta ja täten rahan tuhlausta, mutta myös parantaa pellon tuottavuutta. Tämän tapaisessa *tarkkuusmaataloudessa* teknologisesti suurin haaste on kerätä tarpeeksi dataa kuvaamaan koko peltoa useiden päivän ajan. Tähän haasteen langattomat sensoriverkot soveltuvat hyvin.

### 2.2.1 Prototyyppejä

Yksi esimerkki langattomien sensoriverkkojen käytöstä maataloudessa on sienitautien leviämisen monitorointi pellossa [7]. Hollannissa rakennetussa prototyypissä haluttiin havaita perunataudin leviämistä pellolla, jotta maanviljelijä voisi käyttää torjunta-aineita vain tarvittaessa niihin osiin pellossa, jotka ovat vaarassa saada perunaruttoa. Perunaruton leviäminen riippuu suuresti pellon paikallisista sääoloista. Kosteus on tärkeä tekijä perunaruton kehityksessä. Lisäksi lämpötilalla on myös merkitystä. Tämän takia pellolle sijoitettiin yhteensä 150 langatonta sensorisolmua mittaamaan näitä suureita.

Solmujen asennuksissa otettiin huomioon radion kuuluvuuden heikkeneminen, kun perunasato oli kukassa. Noin 30 sensoritonta solmua toimi tietoliikenteen välittäjinä. Lisäksi solmut asennettiin 75 cm korkeuteen, kun taas itse sensorit asennettiin 20-60 cm korkeuteen.

Tehon säästämiseksi solmut mittasivat kosteutta ja lämpötilaa minuutin välein, mutta lähettivät datan eteenpäin vain kerran kymmenessä minuutissa. Radion



välityksellä lähetettävä data minimoitiin pakkaamalla data “delta encoding” -tekniikalla. Lisäksi tehoa säästettiin pitämällä radio päällä vain 7 prosenttia ajasta.

Sensoriverkon kokoama data kerättiin yhteen pellon reunalla olevaan niulusolmuun, josta edelleen data siirrettiin WiFi:n kautta tietokoneelle. Tietokoneelta data voitiin siirtää Internetin kautta eri servereille.

## 2.3 Liikenteen valvonta

Jatkuvasti lisääntyvä autoilu pahentaa ruuhkia ja aiheuttaa suuret rahalliset menetykset. Ruuhkissa seisovat autot kuluttavat kallista bensiiniä ja ihmiset käyttävät turhaan aikaa ruuhkissa. Uusien autoteiden rakentaminen vähentäisi ruuhkia, mutta niiden rakentaminen on kallista ja usein lisäteille ei ole yksinkertaisesti tilaa tiheästi asutetuilla kaupunkialueilla. Langattomat sensoriverkot voivat auttaa liikenteen ohjaamisessa ja valvonnassa ja näin vähentää ruuhkia.

Langattomilla sensoriverkoilla voidaan paremmin valvoa ja säädellä liikennettä. Sensoriverkot voivat kerätä tietoa autojen määrästä ja nopeuksista eri teillä ja näin päätellä liikenteen ja ruuhkien määrää. Sensoriverkko voi sitten ohjata autoilijan pahempien ruuhkien ohitse käyttäen vaihtoehtoisia reittejä.

Useimmat kaupalliset liikenne raportit on tehty käyttäen videokameroita. Palkatut työntekijät katsovat liikennekameroita ja päättelevät niistä liikenteen ja ruuhkien määrän eri teillä. Tämä menetelmä on kallis muualla kuin kaikkein vilkkaimmilla teillä. Hiljaisemmilla teillä työntekijöiden käyttö on liian kallista. Langattomilla sensoriverkoilla liikenteen valvonta voidaan automatisoida [21, s.6].

### 2.3.1 Prototyyppejä

Yleisimmin käytetty sensorityyppi liikenteen havaitsemiseen on tiehen rakennettava induktiivinen virtajohto. Tämä useita metrejä pitkä käämi haudataan tien alle ja kytketään sähkötolppaan, joka syöttää virtaa käämille. Tämä virta synnyttää käämissä magneettikentän. Kun auto ajaa tiellä käämin ylitse, se muuttaa käämin synnyttämää magneettikenttää ja samoin käämin induktanssia. Tämä induktanssin muutos voidaan havaita käämiin menevän virran muutoksesta. Jos kaksi käämiä sijoitetaan tietyn etäisyyden päästä toisistaan, voidaan havaita auton nopeus [21, s.6].

Huonona puolena käämien käytössä liikenteen valvonnassa on niiden suuri koko ja tarve autotien hajoittamiselle käämien asennusta varten. Sen sijaan magneettiset sensorit ovat pienikokoisempia ja helpompia asentaa kuin induktiiviset sensorit. Magneettiset sensorit havaitsevat muutoksia Maan omassa magneettikentässä, kun auto tulee sen lähelle [21, s.6]. Magneettisia sensoreita käytettiin langattoman sensoriverkon prototyypissä Massachusettsissa [21]. Havaitakseen auton, sensorisolmu odottaa, kunnes se huomaa poikkeaman normaalista Maan magneetti-

kentästä. Tämän jälkeen solmu alkaa mittaamaan magneettikenttää 2 kHz:n taajuudella käyttäen kahta eri sensoria, jotka on sijoitettu solmun etu- ja takapäähän. Vertaamalla magneettikentän muuttumisen aikaeroa solmun etu- ja takapäässä voidaan laskea auton nopeus [21, s.12].

Kyseisen prototyypin hyvänä puolena on sen helppo asennettavuus. Sensorisolmut asennettiin teiden kuoppiin, joten asennusta varten ei tarvinnut hajottaa teitä. Solmut joutuivat kovalle koetukselle fyysisesti, kun autot ja rekat ajoivat sen ylitse, mutta solmut kestivät hyvin tämän rasitteen [21, s.32-33].

Kuten Golden Gate -sillan sensoriverkon prototyypissä, myös tässä prototyypissä koettiin ongelmia radion kuuluvuudessa. Laboratorio-olosuhteissa pystyttiin vastaanottamaan paketteja jopa satojen metrien päästä melkein ilman pakettien menetyksiä, mutta varsinaisissa käyttöolosuhteissa kuuluvuus oli paljon huonompi. Vain joka kolmas paketti pystyttiin vastaanottamaan ilman virheitä, vaikka etäisyys olikin paljon pienempi. Tekijät arvelivat, että sisätiloissa seinät vaimensivat ulkopuolelta tulevia häiriösignaaleita, näin helpottaen sensoriverkon langatonta viestintää. Ulkotiloissa sensoriverkon täytyi jakaa 900 MHz:n taajuuskaista muiden käyttäjien kanssa, joka vaikeutti langatonta viestintää. Tästäkin huolimatta sensoriverkosta saatiin täsmällistä tietoa, koska paketteja lähetettiin enemmän, kuin mitä tarvittiin ja samalla käytettiin tarkistussummia [21, s.33].

## 2.4 Terveydenhuolto

Useita eri sovelluksia langattomille sensoriverkoille on suunniteltu terveydenhuollon alalle. Näihin sovelluksiin kuuluu muun muassa potilaiden monitorointia, joilla on Parkinsonin tauti, epilepsia tai esimerkiksi sydänongelmia. Terveydenhuoltoa varten suunnitellut langattomat sensoriverkot eivät juurikaan toimi itsenäisinä kokonaisuuksina, vaan toimivat yhtenä osana kattavaa terveydenhuoltoa [9, s.30].

Varsinkin Yhdysvalloissa terveydenhuollon kustannukset ovat nousseet vuosi vuodelta ja nykyään peräti 16 prosenttia [34] maan bruttokansantuotteesta koostuu terveydenhuollon kustannuksista. Myös monissa muissa länsimaissa terveydenhuollon kustannukset kasvavat uhkaavasti vanhenevan väestön takia. Langattomat sensoriverkot voi olla osa ratkaisua laskea terveydenhuollon hintaa.

### 2.4.1 Prototyyppejä

Joshua Weaver kehitti MIT:n oppilaitoksessa prototyypin Parkinsonin taudista kärsivien potilaiden monitorointiin käyttäen langatonta sensoriverkkoa [33]. Parkinsonin tauti on neurologinen sairaus, jonka yhtenä oireista on kehon tahaton vapina. Tärkeä osa Parkinsonin taudin hoitamista on oikean lääkeannostuksen selvittäminen. Weaverin kehittämällä sensoriverkolla voidaan havaita, kun potilaaseen iskee tahattoman vapinan kohta. Pitämällä kirjaa kohtausten määristä, voidaan

arvioida tarvittavan lääkeannostuksen määrä.

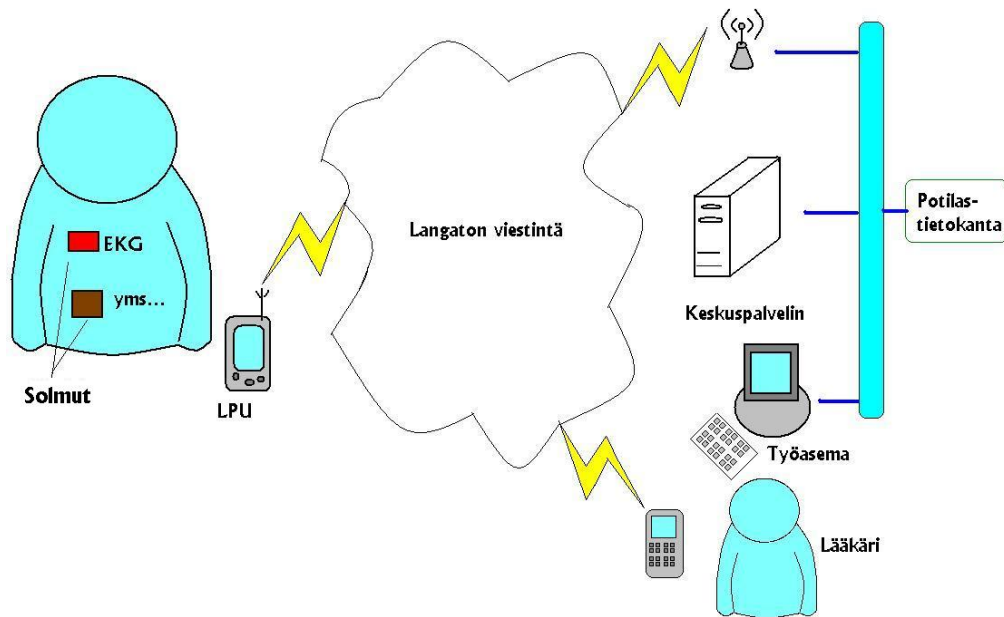
Solmut sisälsivät radion ja prosessorin lisäksi kiihtyvyyssensoreita havaitsemaan tärinäkohtauksia sekä flash-muistia yhteensä 17 tunnin mittauksien tallentamiseen. Solmut kiinnitettiin potilaiden nilkkoihin ja ranteisiin. Mittaukset tehtiin samalla kun potilaat jatkoivat normaalia arkielämää.

Prototyypillä oli viisi tärkeää vaatimusta:

- *Kestävyys*. Koska solmuja tulee käyttämään tärinäkohtauksista kärsiviä ihmisiä, solmut täytyy rakentaa näitä rasituksia kestäväksi.
- *Helppokäyttöisyys*. Jotta sensoriverkko voidaan antaa lääkäreiden ja mahdollisesti itse potilaiden käytettäväksi, järjestelmän täytyy olla helppo asentaa ja käyttää.
- *Keveys*. Sensorisolmujen täytyy olla keveitä ja pienikokoisia, jotta vanhahkot potilaat voivat käyttää niitä tunteja putkeen.
- *Tarkkuus*. Kiihtyvyyssensoreiden täytyy olla tarpeeksi tarkkoja havaitsemaan kehon hienovaraisia vapinoita. Lisäksi sensoreiden pitää pystyä ottamaan näytteitä vähintään 30 kertaa sekunnissa, jotta värinät varmasti havaitaan.
- *Verkoitettavuus*. Systemin pitää pysyä kommunikoimaan ainakin viiden sensorin kanssa yhteisellä väylällä.

Prototyypillä pystyttiin havaitsemaan tärinäkohtaukset 91 prosentissa tapauksista. Tekijät arvelivat, että näin hyvillä tuloksilla voidaan korvata ihmistyöntekijät Parkinsonin potilaiden tärinäkohtauksien mittaamisessa. Tällä tavoin tarkkoja mittauksia pystytään tekemään enemmän ja näin potilaiden elämää voidaan helpottaa tarkemmilla lääkeannostuksilla.

UbiCom-järjestelmässä [17] hyödynnetään langatonta sensoriverkkoa esimerkiksi leikkauksen jälkeiseen potilaan tilan monitorointiin. Kuvassa 2.2 on esitettyinä järjestelmän rakenne. Sensoriverkon solmut ovat päällä pidettäviä laitteita, joissa on fysiologisia sensoreita kuten EKG- (sydänsähkökäyrä) ja lämpötilasensoreita.



*Kuva 2.2: UbiCom-järjestelmän rakenne.*

LPU(local processing unit) voi olla melkein mikä tahansa kannettava laite kuten kämmentietokone tai matkapuhelin. LPU:n tarkoituksena on kerätä langattomalla yhteydellä sensorisolmujen datat ja välittää ne eteenpäin palvelimelle. LPU myös analysoi dataa ja havaitessaan poikkeaman potilaan elintoiminnoissa, LPU antaa varoituksen potilaalle. Näin vältetään suuret viivet vakavimmissa tapauksissa, koska LPU voi itse varoittaa potilaita. Keskuspalvelin tallettaa LPU:lta tulevat datat ja suorittaa pidemmän aikavälin ennustuksia potilaan terveydestä. Lopuksi lääkärit ja hoitajat voivat työkoneelta tarkastella potilaiden tämänhetkistä ja menneisyyden terveydellistä tilannetta.

## 2.5 Elin- ja luonnonympäristöjen monitorointi

Erilaisten elin- ja luonnonympäristöjen monitorointi ja tutkiminen langattomien sensoriverkkojen avulla voi tarjota suuria hyötyjä tieteelle ja yhteiskunnalle. Nämä ympäristöt ovat yleensä niin laajoja, että niiden tutkiminen langallisilla sensoreilla on miltei mahdotonta. Langattomilla sensoriverkoilla pystytään mittaamaan isompia alueita suuremmalla resoluutiolla ja vieläpä pidemmällä aikavälillä kuin millään muulla menetelmällä. Langattoman sensoriverkon toimintaa voidaan muuttaa etäältä niiden asennuksien jälkeenkin. Täten sensoriverkko voi sopeutua hyvin ympäristön muuttuviin olosuhteisiin tai tutkimuksien muuttuviin päämääriin [13, s.349-350].

Suurien alueiden monitorointi vaatii paljon halpoja solmuja, jotka voidaan helposti asentaa kaikkialle halutussa ympäristössä. Langattomien sensoriverkkojen käyttöä on tutkittu muun muassa seuraavissa [13, s.15] sovelluksissa:

- Metsäpalojen varoitusjärjestelmänä.
- Tulva-alueiden varoitusjärjestelmänä.
- Maaperän kosteuden monitorointiin.
- Mikroilmaston ja auringon säteilymäärän kartoittamiseen.
- Jokien ympäristöjen tarkkailuun ja ennustukseen.

Haaste erilaisten eläimien ja kasvien tutkimisessa niiden elinympäristöissä on tarkkailijoiden aiheuttamat tahattomat muutokset niiden käyttäytymiseen. Tutkijan läsnäolo voi vaikuttaa merkittävästi tutkittaviin eläinpopulaatioihin aiheuttamalla ylimääräistä stressiä, vähentämällä lisääntymismahdollisuuksia, lisäämällä saalistusta tai aiheuttamalla eläinten massamuuttoa uusille alueille. Myös kasvis-tolle voi tapahtua vahinkoa, kun tutkijat astuvat kasvien päälle tai kun he tuovat elinympäristölle vieraita elementtejä mukanaan. Luonnon häiriintymistä voidaan vähentää käyttämällä langattomia sensoriverkkoja luonnonympäristöjen tutkimiseen. Sensoriverkko voidaan asentaa paikalle ennen eläinten lisääntymiskautta tai muuta herkkää aikaa, jolloin eläimet eivät häiriinny yhtä paljon. Tämän jälkeen sensoriverkko voi toimia itsenäisesti lisääntymiskauden aikana ilman, että tutkijoiden tarvitsee olla paikan päällä. Myös kasvien tutkiminen on helpompaa, kun sensorit voidaan asentaa kasvien ollessa horroksessa, tai kun maa on vielä jäässä. Tämän lisäksi vaikeakulkuisten alueiden tutkiminen on helpompaa ja turvallisempaa, kun tutkijoiden ei tarvitse yhtä usein olla paikan päällä [13, s.350].

### 2.5.1 Prototyyppejä

Kuuluista prototyypistä [25] langattomien sensoriverkkojen käytöstä luonnonympäristöjen tarkkailuun otettiin käyttöön Yhdysvalloissa 'Great Duck Island' -nimisellä saarella. Prototyypin tavoitteena oli selvittää Myrskykeiju (Leach's Storm Petrel)-nimisen lintulajin lisääntymistapoja. Tutkijat olivat kiinnostuneita miten lintujen pesät ovat käytössä lisääntymiskauden aikana ja miten alueen mikroilmasto vaikuttaa lisääntymiskauteen.

Kyseiset linnut häiriintyvät helposti ihmisen läsnäolosta, joten langattoman sensoriverkon avulla lintuja voidaan tarkkailla ilman, että ne häiriintyvät. Yhteensä 32 solmua asennettiin pesiin ja pesien lähelle. Solmut pystyivät mittaamaan kosteutta, ilmanpainetta, lämpötilaa ja valoisuutta. Lisäksi solmut sisälsivät infrapunasensorin, jota käytetään havaitsemaan linnun läsnäoloa pesässä. Infrapunasensori havaitsee linnun vertaamalla sen näkemän alueen lämpötilaa muun ympäristön lämpötilaan. Jos infrapunasensorin mittaama pesän lämpötila on muutaman asteen ympäristöä lämpimämpi, voidaan päätellä linnun olevan pesässä.

Pesät esiintyvät saarella ryhmissä, jolloin myös solmut muodostavat ryhmiä langattomassa verkossa. Jokaisella solmuryhmällä on yksi solmu, jolla on pitkän kantaman antenni tiedonsiirtoa keskustietokoneelle varten. Keskustietokone puolestaan siirtää tiedot tietokantaan satelliittilinkin avulla. Solmut mittaavat sensoreillaan ja lähettävät datan eteenpäin noin kerran minuutissa.

## 3 Langaton tiedonsiirto sensoriverkoissa

Tässä kappaleessa käsitellään langattomaan tiedonsiirtoon liittyviä seikkoja sensoriverkoissa. Tutustutaan muutamaa MAC- ja reititysprotokollaan ja arvioidaan niiden hyviä ja huonoja puolia sensoriverkoissa.

### 3.1 Medium access control

Langattomissa sensoriverkoissa eri solmut joutuvat useimmiten jakamaan tiedonsiirrossa yhden yhteisen mediumin. MAC (medium access control) on protokolla, joka on vastuussa yhteisen mediumin jakamisesta kaikille solmuille. Oikean MAC-protokollan valinta vaikuttaa merkittävästi sensoriverkon toimintakykyyn, sillä sensoriverkot toimivat usein ISM-kaistoilla (industrial, scientific and medical). Nämä kaistat ovat usein ahkerassa käytössä, joten erilaisia törmäyksiä tai häiriöitä muiden lähetyksien kanssa voi tapahtua usein. Oikealla MAC-protokollan valinnalla voidaan vähentää törmäyksiä ja näin parantaa sensoriverkon tehokkuutta. Lisäksi MAC-protokolla määrää, mitä tehdään törmäyksen jälkeen.

Sensoriverkolle suunnitellun MAC-protokollan ominaispiirre on energiankulutuksen huomioiminen. Koska solmut toimivat usein pattereilla, solmujen radioita ei kannata jättää päälle jatkuvasti kuuntelemaan, josko jokin toinen solmu haluaa lähettää dataa. Muita asioita, joita MAC-protokollan täytyy ottaa huomioon energiankulutuksen kannalta, ovat muun muassa pakettien uudelleenlähetykset, verkon kontrollipaketit ja tarvittavan lähetystehon säätö. Usein sensoriverkoissa tyydytään pienempään tiedonsiirtonopeuteen ja pidempiin latensseihin jos näin saadaan tehonkulutusta pienennettyä.

#### 3.1.1 Kilpailuvapaa MAC

Kilpailuvapaat MAC-protokollat jakavat yhteisen langattoman mediumin siten, että vain yksi solmu kerrallaan käyttää sitä. Tämä tuo mukanaan monia etuja. Esimerkiksi törmäyksiä tapahtuu vähemmän kuin kilpailupohjaisissa protokollissa. Jos mediumin jako on tehty kiinteästi, solmujen tarvitsee pitää radiota päällä vain silloin, kun niillä on jotain lähetettävää tai vastaanotettavaa. Tämän takia kilpailuvapaat protokollat ovat erittäin energiatehokkaita. Lisäksi kiinteät ajoitukset rajoittavat suurinta mahdollista viivettä, mitä datapaketti voi kokea matkalla niulusolmulle.

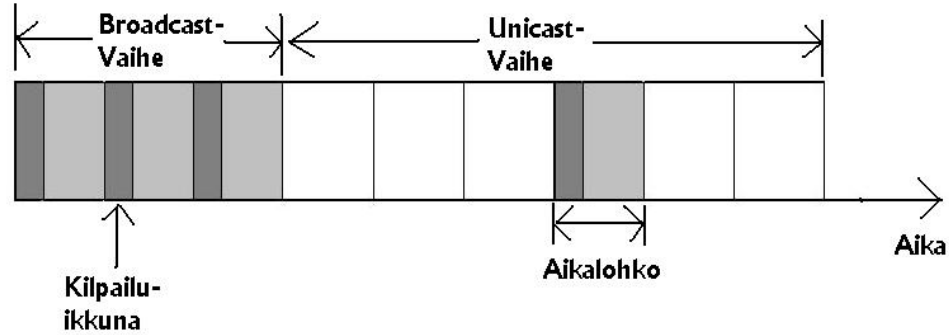
Vastaavasti kilpailupohjaisissa protokollissa ei ole mitään vastaavaa taetta maksimiiveelle [9, s.137].

Kilpailuvapaiden protokollien käytössä on myös huonot puolensa. Jos ajoitukset on asetettu kiinteästi, tämä voi tuoda huomattavia haasteita suunnitella protokolla, joka tehokkaasti käyttää hyväkseen olemassa olevaa kaistaa. Lisäksi, jos sensoriverkon koossa, tiheydessä tai topologiassa tapahtuu muutoksia, tämä voi vaatia muutoksia ajoituksiin. Mikäli muutoksia tapahtuu usein, esimerkiksi jos monet solmut ovat jatkuvasti liikkeessä, kiinteiden ajoitusten käyttö protokollassa voi olla huono ratkaisu [9, s.137].

Seuraavassa kuvataan muutamien kilpailuvapaiden protokollien toimintaa.

- *Y-MAC*: Tämä protokolla käyttää hyväkseen useita eri kanavia sekä jakaa mediumin TDMA-pohjaisesti (Time Division Multiple Access) useisiin aikalohkoihin. Kanavan käyttö jakautuu myös kahteen eri vaiheeseen kuvan 3.1 mukaisesti. Kaikki solmut heräävät ensimmäisen vaiheen (broadcast period) alussa kuuntelemaan, onko tulossa lähetyksiä. Tässä vaiheessa lähetetyt paketit ovat jokaiselle solmulle kuunneltavaksi tarkoitettuja kontrolliviestejä. Ennen kuin lähetys alkaa, solmut kilpailevat mediumista kilpailuikkunassa. Ne solmut, jotka haluavat lähettää, valitsevat satunnaisesti, mihin aikaan tässä ikkunassa alkavat lähettää *aluketta* (engl. preamble). Ennen alukkeen lähettämistä solmu tarkistaa lähettääkö joku muu aluketta. Jos joku muu on jo ehtinyt lähettää alukkeen, solmu menee takaisin nukkumaan eikä yritä varata mediumia. Jos kukaan muu ei lähetä, solmu lähettää aluketta kilpailuikkunan loppuun asti näin varaten mediumin. Kilpailuikkunan jälkeen solmu, joka onnistui varamaan mediumin, lähettää paketin kaikille solmuille. Muut solmut heräävät tätä varten kuuntelemaan [8].

Protokollan toinen vaihe (unicast period) on muuten samanlainen, mutta lähetykset menevät vain yhdelle solmulle. Jokaisella solmulla on oma aikalohko tässä vaiheessa. Koska vastaanottajat heräävät kuuntelemaan mahdollisia lähetyksiä varten vain omalla aikalohkollaan, tämä protokolla on erittäin energiatehokas varsinkin, jos käytetyssä radiossa vastaanottaminen kuluttaa enemmän energiaa kuin lähettäminen [8].



*Kuva 3.1: Y-MAC protokollan arkkitehtuuri.*

Protokollan toisessa vaiheessa solmun aikalohkon on vain niin pitkä, että vain yksi solmu voi lähettää datapaketin yhdessä aikalohkossa. Jos useampi solmu haluaisi lähettää, normaalisti muiden solmujen olisi odotettava seuraavaan kertaan. Tämä voi ruuhkauttaa sensoriverkon helposti. Tämän takia Y-MAC-protokollassa käytetään monta eri kanavaa. Jos solmu ei pystynyt ruuhkan takia lähettämään dataa kohdesolmulle yhdellä kanavalla, se voi yrittää uudestaan toisella kanavalla. Tämä vähentää pakettien kokemaa latenssia, kun dataliikennettä on paljon [8].

- *Lightweight Medium Access Protocol (LMAC)*: Myös LMAC-protokolla perustuu TDMA-tekniikkaan. LMAC jakaa ajan lohkoihin ja yhden lohkon omistaa yksi solmu. Mikään kontrolliyksikkö ei ohjaa lohkojen varaamista, vaan solmut itse huolehtivat lohkojen jaosta. Solmut käyttävät omaa aikalohkoansa lähettämään paketin, joka sisältää kaksi osaa: kontrolliosan ja dataosan. Kaikki solmut pyrkivät kuuntelemaan kontrolliosan. Tämä osa kertoo muun muassa, mille solmulle dataosa on osoitettu, etäisyyden niulusolmulle ja nykyisen aikalohkon tunnuksen. Vain kohdesolmu jää kuuntelemaan tulevaa datapakettia - muut solmut menevät heti nukkumaan. Kontrolliosa sisältää myös bittimaskin, joka kertoo mitkä aikalohkot ovat varattu ja mitkä vapaina. Varattu lohko merkitään loogisella ykkösellä ja vapaa lohko merkitään loogisella nollalla. Lohkot varataan näistä vapaista lohkoista satunnaisesti, joka voi johtaa siihen, että kaksi solmua yrittävät varata saman lohkon. Tällöin kontrolliviestit törmäävät ja lohkojen valinta täytyy suorittaa uudestaan[14]. Radiolähetyksien alussa lähetetään aina ensimmäiseksi aluke, jonka avulla vastaanottaja säätelee vastaanottoherkkyytensä ja synkronisoi itsensä lähettäjän kanssa. Ilman tätä prosessia tapahtuisi liian monta vastaanottovirhettä. Vaikka alukkeen käyttö on tarpeellista, sen vastaanotto ja lähettäminen kuluttaa energiaa. LMAC-protokolla vähentää näiden alukkeiden käyttöä lähettämällä kontrollipaketin ja datapaketin peräkkäin, jolloin ei tarvitse toista erillistä alu-



ketta [14].

Myös radion päälle ja pois laittaminen kuluttaa turhaan energiaa. Kun radio laitetaan pois päältä, tyypillisesti myös radion oskillaattori menee pois päältä. Kun radio taas laitetaan päälle, oskillaattorin käynnistämisessä kuluu aikaa. Tämä kuluttaa turhaan energiaa. LMAC pyrkii minimoimaan radion päälle- ja poiskytkemiset siten, että solmujen ei tarvitse suorittaa mitään kättelymekanismeja ennen datasiirtoa [14].

- *Mobile LMAC (MLMAC)*: Paikallaan pysyvälle sensoriverkolle LMAC on hyvä protokolla, mutta koska aikaloikat lasketaan vain kerran, ei se sovellu protokollaksi liikkuville sensoriverkoille. Tätä varten on kehitetty LMAC:sta protokolla liikkuville sensoriverkoille, joka pystyy sopeutumaan muuttuviin topologioihin. MLMAC-protokolla perustuu myös aikaloikojen jakoon, mutta toisin kuin LMAC-protokollassa synkronointi ei välttämättä aina ala niulusolmusta. Sen sijaan solmu, joka haluaa ensimmäiseksi lähettää, aloittaa synkronisaation [18].

Koska solmut voivat liikkua verkossa, jokin solmu  $X$  voi poistua toisen solmun  $Y$  kuuluvuusalueelta. Tällöin solmut eivät enää kuule toistensa kontrolliviestejä, jolloin ne merkitsevät kyseisen aikaloikon vapaaksi kontrolliviesteisiinsä. Kun taas solmu  $X$  siirtyy toisen solmun  $Z$  kuuluvuusalueelle ja  $Z$  tietää toisen solmun  $W$ , joka käyttää samaa aikaloikoa kuin solmu  $X$ , solmujen  $X$  ja  $W$  kontrolliviestit törmäävät. Tällöin solmu  $Z$  ei kuule yhtään kontrolliviestiä ja merkitsee aikaloikon vapaaksi. Solmut  $X$  ja  $W$  kuulevat solmun  $Z$  kontrolliviestin ja huomaavat, että aikaloiko on vapautettu ja päättelevät, että on täytynyt tapahtua kontrolliviestien törmäys. Tämän takia solmut  $X$  ja  $W$  yrittävät löytää toisen aikaloikon itselleen [18].

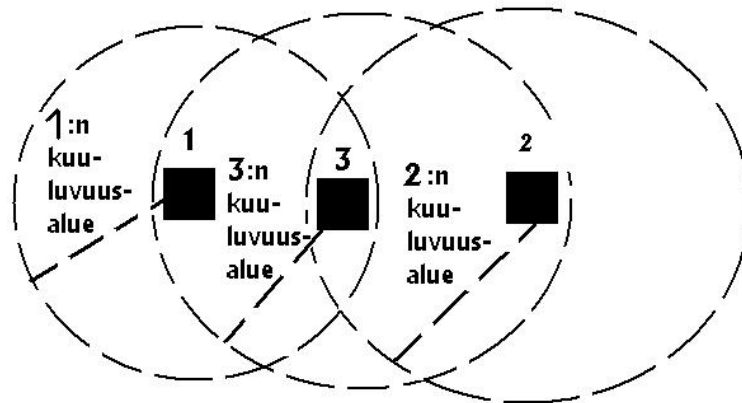
### 3.1.2 Kilpailupohjainen MAC

Kilpailupohjaiset MAC-protokollat eivät ajoita eri solmujen lähetystä siten, ettei törmäyksiä tapahdu, vaan luottavat muihin mekanismeihin kilpailutilanteen ratkaisemiseksi. Seuraavassa käsitellään muutamien kilpailupohjaisien protokollien toimintaa.

*Aloha* on ensimmäinen protokolla, jota on käytetty langattomissa tiedonsiirto verkoissa. Aloha-protokollassa solmu lähettää paketin heti, kun sillä on jotain dataa lähetettävänä. Mitään tarkistusta etukäteen, onko kanava vapaana, ei tehdä. Sen sijaan käytetään kuittauspaketteja varmistamaan, että lähetys onnistui. Jos tulee törmäys toisen lähetyksen kanssa, solmu yrittää myöhemmin uudestaan. *Slotted Aloha* on Alohan parannettu versio. Siinä törmäyksiä yritetään välttää määräämällä tietyt ajanjaksot lähetyksille. Solmu saa lähettää vain joidenkin ajanjaksojen alus-

sa. Vaikka Slotted-Aloha vähentää törmäyksiä, se vaatii solmujen synkronointia keskenään[23].

Kuvassa 3.2 on esitettyä niin sanottu piiloasemaongelma(engl. hidden node problem) kolmen solmun(1-3) kesken. Kuvassa ympyrät esittävät solmujen radioi- den kuuluvuusalueita. Molempien solmujen 1 ja 2 kuuluvuusalueet ylittyvät solmulle 3, joten ne voivat lähettää paketteja sille. Mutta solmut 1 ja 2 eivät tiedä toisistaan, koska niiden kuuluvuusalueet eivät ole niin suuria. Tämän takia ne voivat yhtä aikaa lähettää pakettin solmulle 3, jolloin tapahtuu törmäys ilman, että solmut 1 tai 2 edes huomaavat törmäystä. Pahimmillaan törmäyksessä molemmat paketit menetetään.

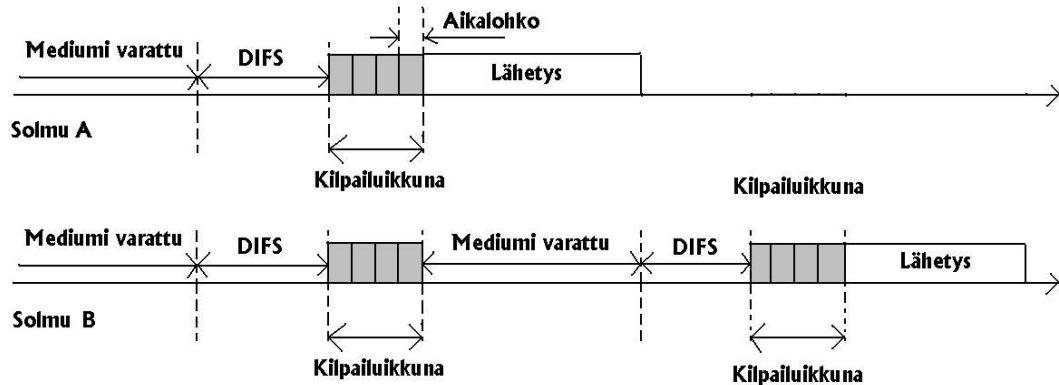


**Kuva 3.2:** Piiloasemaongelma kolmen solmun kesken.

Monet MAC-protokollat perustuvat CSMA(carrier sense multiple access)-tekniikkaan, joka on yksi yksinkertaisimmista tavoista säädellä solmujen pääsyä yhteiselle kanavalle. CSMA jakautuu kahteen eri versioon: non-persistent ja p-persistent CSMA. Non-persistent CSMA:ssa solmu, joka haluaa lähettää dataa, ensin kuuntelee kanavaa selvittääkseen, onko kanava vapaa. Jos solmu havaitsee kanavalla toisen lähetyksen, solmu myöhästyttää omaa lähetystään ja kokeilee myöhemmin uudelleen. Jos kanava on vapaa, solmu aloittaa lähetyksen välittömästi. P-persistent CSMA:ssa solmu ei ala heti lähettämään dataa, kun kanava on vapaa, vaan lähettää dataa todennäköisyydellä  $p$ . Eli todennäköisyydellä  $1-p$  solmu ei lähetä dataa, vaikka kanava olisikin vapaa. Tällä tavoin vähennetään törmäyksiä tilanteissa, joissa solmut päättävätkin aloittaa lähettämisen samaan aikaan [22, s.5].

CSMA/CA (carrier sense multiple access with collision avoidance) on CSMA:n laajennettu versio, joka pyrkii vähentämään törmäyksiä. Kuvassa 3.3 on esitetty menetelmän perusperiaate. CSMA/CA:ssa solmu ei lähetä dataa heti, kun kanava on vapaa, vaan odottaa tietyn aikavälin nimeltä DIFS, ennen kuin mahdollisesti yrittää lähettää. DIFS'in jälkeen on useisiin ikkunoihin jaettu kilpailuaikaväli, jolloin solmut saavat yrittää varata kanavan. Solmut valitsevat satunnaisesti kilpailuaikavälistä yhden ikkunan, jolloin ne haluavat lähettää. Mikäli solmu oli ensimmäinen kanavalle pyrkijä, se saa kanavan haltuunsa ja aloittaa lähettämisen. Jos joku toinen solmu

oli valinnut aikaisemman ikkunan, joutuu solmu odottamaan, kunnes tämän toisen solmun lähetykset loppuu. Tämän jälkeen on odotettava vielä uudestaan DIFS aikaväli [9, s.129].



**Kuva 3.3:** CSMA/CA:n ajoituskaavio.

Lisäksi CSMA/CA-menetelmässä käytetään kontrolliviestejä törmäyksien välttämiseksi. Kun solmu on huomannut kanavan olevan vapaa, se lähettää kohdesolmulle RTS-viestin (request to send), joka kysyy kohteelta, onko se valmis vastaanottamaan. Tähän kohde vastaa lähdesolmulle CTS-viestillä (clear to send), joka kertoo lähteelle, että kohde on valmis vastaanottamaan. Jos lähdesolmu ei saa CTS-viestiä tietyn ajan kuluessa, se yrittää lähetykstä myöhemmin uudestaan. Kohdesolmu ei vastaa RTS-viestiin CTS-viestillä, jos se ei pysty lähettämään tai vastaanottamaan dataa kunnolla. Kaikki muut solmut jotka kuulevat RTS- tai CTS-viestejä kanavalla, tietävät kanavan olevan varattu ja jättävät oman lähetyksensä myöhempään ajankohtaan. Vaikka CSMA/CA-menetelmä pienentää piiloasemaongelman vaikutuksia, se vaatii useiden kontrolliviestien lähetyksiä. Usein sensoriverkoissa kontrolliviestit ovat melkein samanpituisia kuin itse dataviestit, jolloin saadut hyödyt voivat jäädä pieniksi. Hyödyn suuruus riippuu verkon topologiasta, dataliikenteen määrästä ja kanavan ominaisuuksista, joten joissain tapauksissa CSMA/CA-menetelmästä ei ole hyötyä sensoriverkoissa [22, s.6].

Pienillä sensoriverkon tiheyksillä Aloha-protokollan on todettu toimivan jopa tehokkaammin (pienempi todennäköisyys sensoriverkon katkoksille) kuin CSMA:n. Mutta kun sensoriverkon tiheys kasvaa, CSMA:lla on parempi tehokkuus. Tämä johtuu siitä, että todennäköisyys solmun perääntymiselle lähetyksestä kasvaa johtaen vähempiin törmäyksiin. CSMA:n tehokkuus paranee entisestään, jos lähettäjän sijasta vastaanottaja havaitsee onko kanava vapaa vai ei. Slotted-Alohan tehokkuus on todettu olevan kaikkein paras kaikilla tiheyksillä [19].

### 3.1.3 MAC-protokollat sensoriverkoissa

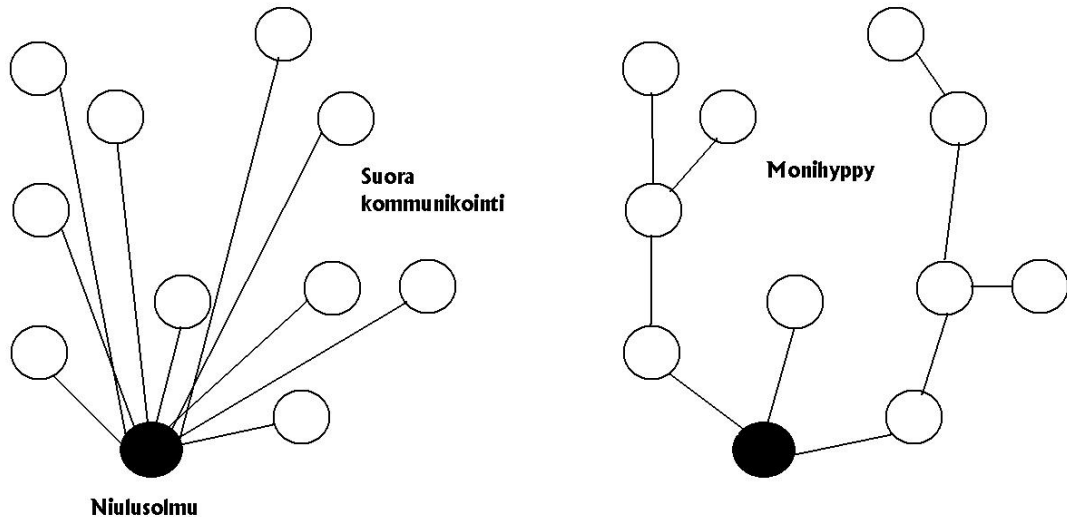
Edelliset kappaleet kuvasivat yleisesti MAC-protokollia tietoliikenneverkoissa hieman sivuaen niiden käyttöä sensoriverkoissa. Tämä kappale tarkastelee tarkemmin MAC-protokollien soveltuvuutta sensoriverkoissa.

Perinteiset MAC-protokollat eivät sovellu suoraan käytettäväksi sensoriverkoissa muun muassa solmujen rajallisten resurssien, monihyppy reitityksen ja erilaisten sovelluskohtaisten vaatimuksien takia. Monet MAC-protokollat osaavat hyvin optimoida latenssin, suoritustehon ja tasapuolisuuden välillä. Sen sijaan sensoriverkoissa täytyy ottaa erityisesti energiankulutus huomioon, jota perinteiset MAC-protokollat eivät tee. Esimerkiksi sensoriverkoissa radio vie huomattavan paljon energiaa päällä ollessaan, joten usein sensoriverkoissa radiot sammutetaan solmuista kun niitä ei tarvita. Tietysti, kun radiot ovat pois päältä, sensoriverkossa langaton tiedonsiirto ei toimi näiden solmujen kautta, joten MAC-protokollan suunnittelu on monimutkaisempaa. Kuitenkin tämän tekniikan avulla radiot ovat päällä usein vain yhden prosentin ajasta, jolloin saavutetaan merkittävät säästöt energiankulutuksessa [22, s.12].

Keskitettyssä MAC-protokollassa kanavalle pääsyn hoitaa sensoriverkossa yksi kontrollisolmu, joka päättää mikä solmu saa lähettää ja milloin. Toisaalta hajautetussa protokolla ei ole yksittäistä kontrollisolmuja, vaan jokainen solmu hoitaa itse pääsyn kanavalle. Eli toisin sanoen jokainen solmu on kontrollisolmu.

Useiden seikkojen takia keskitetty MAC-protokolla sensoriverkoissa ei toimi hyvin, vaan yleensä käytetään hajautettua kontrollia. Jotta kontrollisolmu pystyisi reagoimaan muutoksiin sensoriverkossa, muiden solmujen täytyy lähettää sille dataa verkon tilasta. Koska sensoriverkoilla on yleensä alhainen tiedonsiirtonopeus ja käytössä on monihyppy-tekniikka, kontrollisolmu on hidas reagoimaan verkon muutoksiin. Kun kontrollisolmu on saanut selville verkon tilan, voi se olla jo vanhentunutta tietoa. Lisäksi kaiken tämän tiedon jakaminen langattomasti kuluttaa huomattavasti energiaa. Kuitenkin tiedon jakaminen paikallisesti lähekkäisten solmujen kesken verkon tilasta voi auttaa paremman reitityksen muodostamisessa, joten MAC-protokollaa suunniteltaessa on tehtävä kompromissi tiedon jakamisen ja energiankulutuksen kanssa [22, s.17].

Monihyppy-tekniikassa (kuva 3.4) paketti reititetään useamman solmun kautta lähdesolmusta kohdesolmulle, vaikka lähdesolmu mahdollisesti pystyisikin lähettämään paketin suoraan kohdesolmulle. Tällä tavalla säästetään energiaa, koska vaikka paketin reitittämiseen meneekin monta eri lähetystä, voidaan käyttää paljon pienempää lähetystehoä. Lähetystehon pienentäminen tällä tavalla on hyvä taktiikka, koska tarvittavan lähetystehon määrä kasvaa etäisyyteen verrattuna tyypillisesti toiseen tai jopa neljäänteen potenssiin. Jos lähetysteho kasvaisi etäisyyden ensimmäiseen potenssiin verrannollisesti, monihyppy-tekniikalla ei saavutettaisi energiansäästöjä [22, s.12-13].



**Kuva 3.4:** Monihyppy-tekniikassa paketti reititetään useamman solmun kautta, vaikka suora reitti olisikin mahdollista.

Sensoriverkkojen laaja sovellusalue tekee MAC-protokollien suunnittelusta haasteen kehittäjille, sillä jokainen eri sovellus voi tuottaa täysin erityyppistä liikennettä ja näin ollen protokollan vaatimukset ovat erilaiset. Tämän takia sensoriverkkojen protokollat ovatkin yleensä Ad Hoc -tyyppisiä, eli vain juuri kyseiseen sovellukseen sopivia. Sensoriverkkoihin voidaan tietyn rajoituksen soveltaa yleisille langattomille verkoille tehtyjä Ad Hoc -verkkojen tutkimustuloksia. Kuitenkin Ad Hoc -verkoissa on keskitetty enemmän laitteiden liikkuvuuteen, kun taas sensoriverkkojen solmut ovat yleensä paikallaan tai liikkuvat vain vähän. Kuitenkin esimerkiksi piiloasemaongelma on sensoriverkoissa samantapainen kuin langattomissa verkoissa [22, s.13].

Verrattuna tavallisiin langattomiin tietoliikennejärjestelmiin sensoriverkkojen tuottamat viestit ovat yleensä lyhyempiä ja näin ollen kontrolliviestien muodostama ylimääräinen rasite (engl. overhead) on suhteessa isompi. Tämän takia saattaa olla optimaalisinta käyttää pienempää ja yksinkertaisempaa MAC-protokollaa. Joskin erittäin yksinkertainen protokolla ei välttämättä pysty vähentämään tehonkulutusta mukautumalla kanavan olosuhteisiin esimerkiksi vähentämällä lähetystehoja. Toisaalta liian monimutkainen protokolla voi kuluttaa solmun prosessoria niin paljon, että prosessori ei vietä paljoakaan aikaa tehonsäästötiloissa. Lisäksi monimutkaisempi protokolla voi hoitaa tiettyjä toimintoja vähemmällä energialla, kuin jos se jätettäisiin sovellustason hoidettavaksi [22, s.13,16].

Suurin ero sensoriverkkojen MAC-protokollissa verrattuna muihin langattoman viestinnän MAC-protokollisiin on keskittyminen käytettävän radion energiankulutuksen minimoiseen. Radio on yleensä kaikkein suurin tehonkuluttaja sensorisolmussa, joten sen tehonkulutuksen pienentäminen on ensisijaisen tärkeää patterikäyttöisissä solmuissa. Radion tehonkulutusta voidaan pienentää välttämällä *törmäyksiä, ylikuumumista, turhaa kuuntelemista* ja *ylimääräistä rasitetta*. Törmäykset eivät juurikaan

haittaa sensoriverkkojen tiedonsiirtonopeutta, koska niillä on usein alhaiset vaatimukset datasiirron nopeudelle ja latenssille. Vaikka törmäykset eivät sensoriverkoissa ole yhtä haitallisia kuin muissa langattomissa verkoissa, törmäyksissä tuhlataan energiaa. Törmäyksien jälkeen tarvittavat uudelleenlähetykset voivat moninkertaistaa radion käyttämän energian, sillä radio on tällöin korkeimmassa energiankulutustilassaan sen sijaan, että se olisi tehonsäästötilassa. Ylikuulumista tapahtuu, kun solmu vastaanottaa viestin, jonka kohde ei ollutkaan kyseinen solmu. Turhien viestien vastaanottamisessa tuhlataan energiaa, joten joissakin sensoriverkkojen MAC-protokollissa voidaan viestin vastaanottaminen lopettaa aikaisin. Tämä lopettaminen voidaan hoitaa lukemalla ensin viestissä oleva kohdesolmun osoite, ja jos osoite ei vastaa kyseisen solmun osoitetta, vastaanottaminen lopetetaan ja solmu menee takaisin tehonsäästötilaan [22, s.14].

Turhaa kuuntelemista tapahtuu, kun mikään solmu ei ole lähettämässä mitään dataa, mutta kuitenkin useat solmut yrittävät vastaanottaa viestejä. Vaikka monet radiot kuluttavat kuuntelutilassa vähemmän tehoa kuin lähetystilassa, tämäkin tehonkulutus on yleensä huomattavan paljon suurempi, kuin jos radio olisi tehonsäästötilassa. Esimerkiksi tässä diplomityössä käytetty radio nRF23L01P kuluttaa kahdessa eri tehonsäästötilassa 900 nA ja 26  $\mu$  A virtaa, lähetystilassa 7-11 mA ja kuuntelutilassa 12,6 - 13,5 mA virtaa [28, s.14]. Eli kyseisessä radiossa kuuntelu vie jopa enemmän virtaa kuin lähetys. Turhaa kuuntelemista voidaan vähentää asettamalla radiolle ajastin, jolloin tietyn ajan päästä lopetetaan kuunteleminen, jos viestejä solmulle ei ole tullut. Vaikka monet radiot kuluttavat kuuntelutilassa yhtä paljon energiaa riippumatta vastaanottavatko ne viestejä tai ei, on olemassa joitakin radioita, jotka voivat kuunnella kanavaa erittäin alhaisella tehonkulutuksella. Tämän tyyppiset radiot voivat huomattavasti vähentää turhan kuuntelemisen aiheuttamaa energian tuhlausta [22, s.14-15].

Monissa radiomoduuleissa on olemassa vain yksi tehonsäästötila, jossa lähes kaikki piirit radiossa ovat pois päältä. Vaikka tämänkaltainen tehonsäästötila vie vain vähän energiaa, siitä herääminen on yleensä hidasta. Heräämisen aikana radiolla ei voi tehdä mitään hyödyllistä työtä ja tämä heräämisaika täytyy ottaa huomioon MAC-protokollaa suunniteltaessa. Radioilla, joissa on useita erilaisia tehonsäästötiloja, voidaan toteuttaa joustavampi MAC-protokolla. Nämä ylimääräiset tehonsäästötilat vievät enemmän energiaa, mutta niiden ansiosta MAC-protokolla voi nopeammin reagoida muuttuviin tilanteisiin [22, s.15]. Diplomityössä käytetyssä radiossa on olemassa kaksi erilaista tehonsäästötilaa: *power down* ja *standby-I*. Power-down tila vie vain 900 nA virtaa, mutta siitä herääminen kestää jopa 4.5 ms riippuen käytetyn kideoskillaattorin sarjainduktanssin arvosta. Sen sijaan standby-I tila vie 26  $\mu$  A virtaa, mutta heräämisaika on vain 130  $\mu$  s [28, s.14,s.22]. Nämä heräämisajat täytyy ottaa huomioon varsinkin MAC-protokollan ajoituksessa. Esimerkiksi jos solmun radio yrittää nukkua lyhyemmän ajan, kuin mitä kestää herätä,

voi protokollan ajoitukset mennä pieleen.

Lisäksi sensoriverkkojen MAC-protokollissa täytyy ottaa huomioon solmujen rajallinen muistikapasiteetti. Tehokkaat MAC-protokollat voivat vaatia ison reititystaulukon, jossa pidetään kirjaa tunnetuista naapurisolmuista ja niiden ominaisuuksista. Tämä voi vähentää tilaa, mitä jää sensoreiden datan keräykseen. Lisäksi isokokoinen MAC-protokolla aiheuttaa enemmän muistin käyttöä, jolloin myös muistin tehonkulutus kasvaa. Täten turhan monimutkaista MAC-protokollaa ei kannata välttämättä käyttää sensoriverkossa[22, s.16].

## 3.2 Reititysprotokollat

MAC-protokollat määrittävät, miten solmut pääsevät käsiksi yhteiseen kommunikaatiomediumiin. Tämän jälkeen on vielä päätettävä miten datapaketit reititetään lähdesolmulta niulusolmulle. Tätä varten on olemassa erilaisia reititysprotokollia, joita käsitellään tässä kappaleessa.

Kaikkein yksinkertaisin tapa reitittää dataa on käyttää yhden hypyn reititystä. Siinä reititettävä data siirretään suoraan lähdesolmulta niulusolmulle. Tämän tapainen suora kommunikaatio lähde- ja niulusolmun välillä on kuitenkin yleensä epäkäytännöllistä. Suorassa kommunikaatiossa lähetystehot täytyy olla suuria, jotta kaikki solmut ovat niulusolmun kuuluvuusalueella. Tämä kuluttaa huomattavan paljon energiaa, joka on huono asia patterikäyttöisissä sensorisolmuissa. Toisaalta jos käytetään pieniä lähetystehoja, sensoriverkon kattama alue jää pieneksi. Tämän takia melkein aina langattomissa sensoriverkoissa käytetään monihyppyreititystä, eli datapaketit siirretään lähdesolmulta niulusolmulle käyttäen muita solmuja välietappeina.

Langattomien sensoriverkkojen reititysprotokollien tekeminen on haastava tehtävä monista syistä. Usein solmut ovat sijoitettu ympäristöön ilman mitään tiettyä rakennetta, jolloin syntyvät reititystopologiat ovat myös satunnaisia. Solmujen pitää osata tällöin järjestäytyä itsenäisesti ilman ihmisten apua. Tähän järjestäytymiseen kuuluu solmun sijainnin määrittäminen, solmun naapurien tunnistaminen ja reitin muodostaminen niulusolmulle. Tämän lisäksi reititysprotokollien on otettava huomioon solmujen rajalliset resurssit ja langattoman mediumin epäluotettavuuden. Esimerkiksi sensorisolmuilla on rajallinen kyky tallentaa ja prosessoida dataa sekä rajallinen määrä energiaa käytettävissä. Reititysprotokollien on pärjättävä näiden rajoitteiden kanssa, mutta samalla pystyttävä mukautumaan jatkuviin muutoksiin topologiassa, kun solmut menevät rikki tai liikkuvat paikasta toiseen.

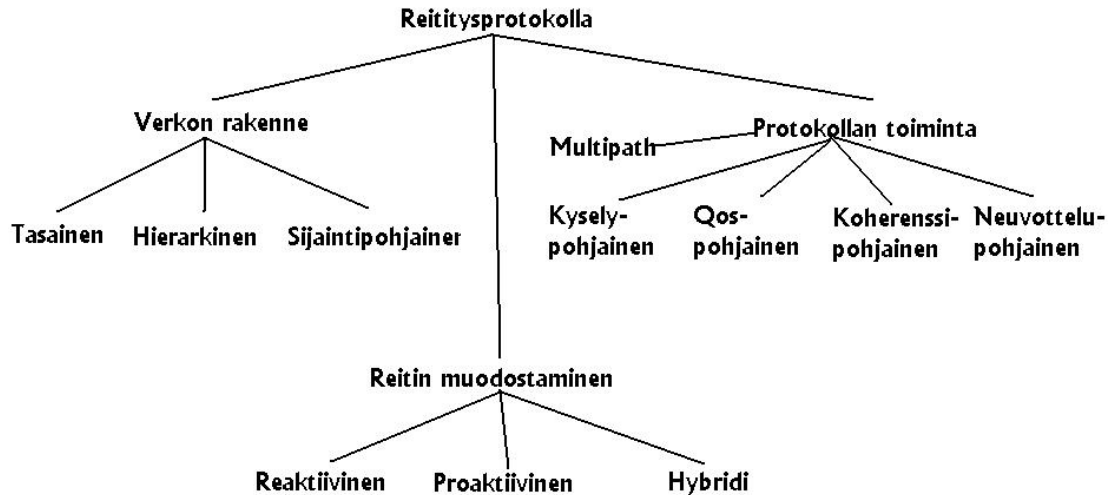
Reititysprotokollat voidaan luokitella monella eri tavalla. Kuvassa 3.5 on esitetty kolme eri tapaa miten reititysprotollia voidaan luokitella reitin muodostamisen, sensoriverkon rakenteen ja protokollan toiminnan mukaan. Useimmat langattomat sensoriverkot voidaan jakaa rakenteensa mukaan kolmeen eri kategoriaan. Tasaisissa rakenteissa(engl. flat-based) kaikilla solmuilla on sama toiminnallisuus ja rooli ver-

kossa. Hierarkisissa rakenteissa(engl. hierarchical-based) eri solmuilla voi olla erilaisia tehtäviä reitityksessä. Esimerkiksi jotkut solmut voivat reitittää dataa muiden solmujen puolesta, kun taas muut solmut vain synnyttävät dataa ja välittävät sen eteenpäin. Sijaintipohjaisissa rakenteissa(engl. location-based) käytetään hyväksi solmujen sijaintitietoa reitityksen ohjaamiseen [9, s.164].

Reititysprotokollat voidaan jakaa myös kolmeen eri kategoriaan riippuen siitä, miten ne muodostavat reitin niulusolmulle. *Reaktiivisissa* protokollissa reitti niulusolmulle muodostetaan vain siinä vaiheessa, kun solmulla on jotain dataa lähetettävänä. Koska reitti muodostetaan vasta, kun dataa pitäisi lähettää, reaktiivisissa protokollissa esiintyy enemmän viivettä datan lähetyksen kanssa. Sen sijaan *ennakoivissa* protokollissa reitti niulusolmulle muodostetaan ennen, kuin solmulla on mitään dataa lähetettävänä. Tällä tavalla voidaan välttää reaktiivisen protokollan viiveet. Ennakoivat protokollat joutuvat ylläpitämään taulukkoa, jossa on listattu kaikki päämäärät, reitit niille ja kyseiseen reittiin liittyvä kustannus. Näiden taulukoiden ylläpito voi olla kallista, sillä joissakin protokollissa tätä varten on lähetettävä sensoriverkon jokaiselle solmulle tietoa omasta solmusta ja sen naapureista. Ennakoivien protokollien huonona puolena on myös se, että ne voivat löytää reittejä, joita ei koskaan tule käytetyksi. Tämä tuhlaa solmujen rajallista energiaa. Lisäksi ennakoivien protokollien reittitiedot voivat olla jo vanhentuneita siihen mennessä, kun niitä tullaan käyttämään. Lopuksi joissakin *hybridi* protokollissa esiintyy sekä reaktiivisten että ennakoivien protokollien ominaisuuksia [9, s.165].

Kolmas tapa luokitella reititysprotokollia on niiden toiminnan mukaan. Esimerkiksi *neuvottelupohjaisissa* protokollissa solmut lähettävät keskenään neuvotteluviestejä ennen varsinaisen datasiirron alkua. Näin pyritään välttämään ylimääräisten datapakettien siirtoja. *Multipath*-tyyppisissä protokollissa käytetään yhtä aikaa montaa eri reittiä. Näin pyritään saavuttamaan suurempi datasiirron nopeus tai luotettavampi tiedonsiirto. Tämän tekniikan huonona puolena on tietenkin suurempi energiankulutus, kun paketteja lähetetään enemmän, kuin mitä olisi tarpeellista. *Tiedustelupohjaisissa*(engl. query-based) protokollissa datapakettien vastaanottajat lähettävät muille solmuille tiedustelupaketteja. Näissä paketeissa kysytään, onko solmuilla mitään lähetettävää vastaanottavalle solmulle. Eli toisin sanoen vastaanottavat solmut aloittavat tiedonsiirron. *QoS-pohjaisissa* (Quality-of-service) protokollissa pyritään valitsemaan reitti, joka toteuttaa jonkin QoS-vaatimuksen. Tämä vaatimus voi olla esimerkiksi pieni latenssi, pieni energiankulutus tai vähäinen pakettien menetys. Lopuksi *koherenssi*-tyyppisissä protokollissa solmut tekevät vain vähän esiprosessointia datapaketille ennen, kuin lähettävät paketin eteenpäin. Sen sijaan *ei-koherenssi*-tyyppisissä protokollissa solmut tekevät huomattavan paljon prosessointia datalle ennen, kuin ne lähettävät paketit eteenpäin [9, s.165].





*Kuva 3.5: Muutama tapa miten reititysprotokollia voidaan luokitella.*

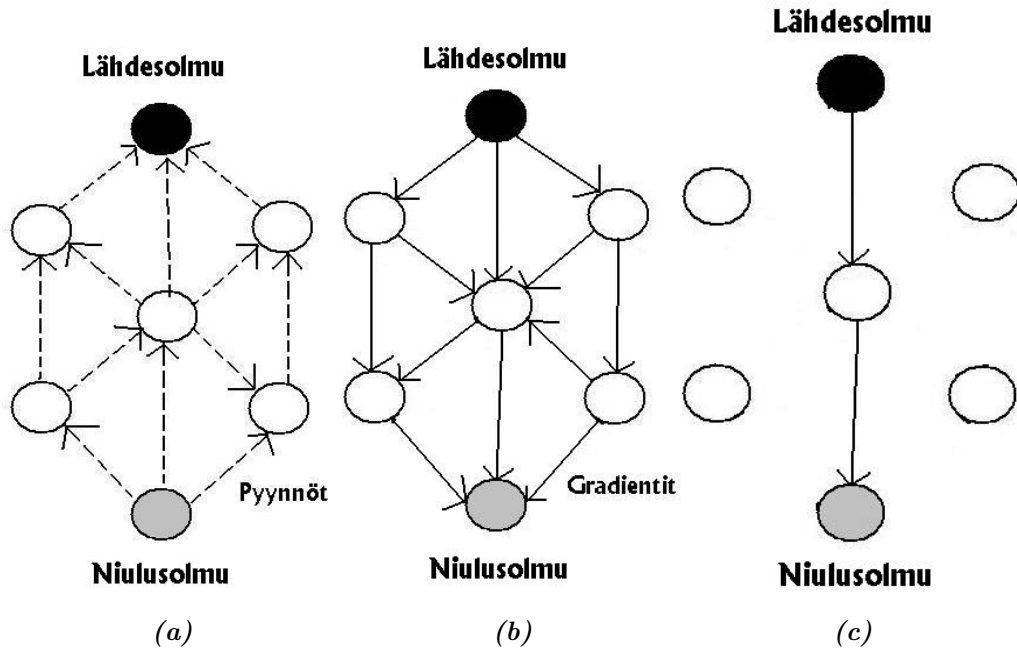
Kuvan 3.5 lisäksi protokollia voidaan jakaa kahteen eri luokkaan riippuen siitä, miten ne viittaavat sensoriverkkojen solmuihin. *Solmukeskisissä* protokollissa sensoridata lähetetään tietylle solmulle tai solmuryhmille. *Datakeskisissä* protokollissa itse solmuihin ei suoraan viitata, vaan solmut kuvataan niiden ominaisuuksien avulla. Esimerkiksi niulusolmu voi pyytää kaikilta solmuilta lämpötilatietoa, mutta pyyntöön vastaa vain ne solmut, joilla on lämpötilasensori [9, s.165].

### 3.2.1 Datakeskiset protokollat

Tämä kappale käsittelee erilaisia datakeskisiä protokollia, niiden ominaisuuksia ja hyviä ja huonoja puolia.

- *Tulvitus*: Tulvitus(engl. flooding) on helpoin ja yksinkertaisin tapa reitittää paketteja verkossa. Tässä tekniikassa solmu lähettää datapaketin kaikille naapureilleen, jotka puolestaan lähettävät saman paketin eteenpäin kaikille omille naapureilleen. Näin jokainen solmu verkossa vastaanottaa kyseisen paketin. Tulvituksen etuna on sen yksinkertaisuus, kun taas sen huonona puolena on suuri datasiirron määrä. Koska datapaketti lähetetään kaikille solmuille verkossa, vaikka kohteena olisikin vain yksi tietty solmu, tämä aiheuttaa paljon turhaa datasiirtoa verkossa. Turhaa datasiirtoa voidaan vähentää lisäämällä pakettiin laskuri, joka kertoo kuinka monta kertaa kyseisen paketin voi vielä lähettää eteenpäin. Joka kerta, kun solmu lähettää paketin eteenpäin, se laskee laskurin arvoa. Tämän lisäksi jokaiseen pakettiin voidaan lisätä pakettikohtainen tunniste, jonka avulla solmu ei lähetä eteenpäin samaa pakettia kahteen kertaan [9, s.169].

- *Juoruaminen* (engl. gossiping): Tämä on samantapainen reititystapa kuin tulvitus sillä erolla, että solmu ei välttämättä lähetä jokaista pakettia eteenpäin. Solmu lähettää paketin eteenpäin vain tietyllä todennäköisyydellä  $p$ , jolloin se jättää lähettämättä paketin todennäköisyydellä  $1 - p$ . Tällä välttyään ainakin osittain tulvituksen aiheuttamasta suuresta määrästä tiedonsiirtoa. Jos todennäköisyys  $p$  on liian iso, saavutetut hyödyt tulvitukseen verrattuna jää pieniksi. Toisaalta jos  $p$  on liian pieni, on mahdollista, että paketti ei pääse päämääräänsä. Jos esimerkiksi solmun ainoa naapuri päättääkin olla lähettämättä pakettia eteenpäin, paketti menetetään [9, s.170].
- *Ohjattu diffuusio* (engl. directed diffusion): Ohjatussa diffuusiassa vastaanottaja lähettää verkolle pyyntöjä tietystä datasta. Nämä pyynnöt muodostavat verkkoon *gradientteja*, joita käytetään sensoridatan reitittämiseen takaisin pyynnön lähettäjälle. Kuva 3.6 esittää protokollan toimintaperiaatteen. Aluksi niulusolmu lähettää pyyntöjä verkkoon kuvan 3.6a mukaisesti. Solmut lähettävät pyyntöjä eteenpäin kaikille muille solmuille, josta se itse ei ole saanut samaista pyyntöä. Nuolet esittävät pyyntöjen etenemistä. Tämän jälkeen solmut muodostavat gradientin niulusolmulle päin. Tämä gradientti osoittaa niihin solmuihin, joista solmu on saanut kyseisen pyynnön. Kuva 3.6b esittää tämän vaiheen. Lähdesolmulla on usein useampia gradientteja niulusolmulle päin. Näistä gradientteista solmu valitsee yhden suunnan jonkin säännön perusteella, johon se lähettää paketin sensoridataa. Jokainen solmu lähdesolmun ja niulusolmun välillä tekee samoin. Tämä viimeinen vaihe on esitetty kuvassa 3.6c. Koska niulusolmun täytyy jokaista sensoridatan siirtoa varten lähettää pyyntö verkolle, tämä ei ole energiatehokas protokolla, mikäli sovelluksessa vaaditaan jatkuvaa sensoridatan hakemista [10].



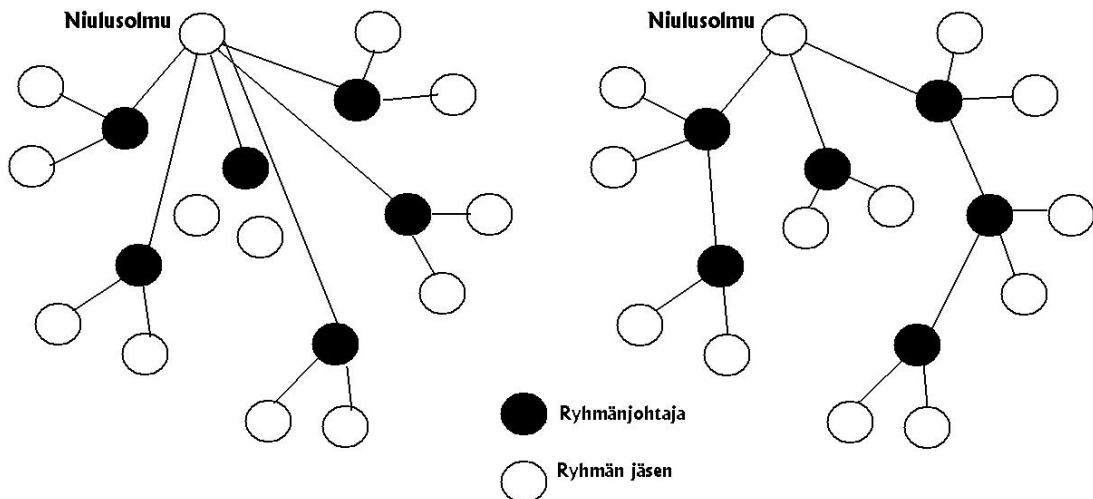
**Kuva 3.6:** Ohjatun diffuusion toimintaperiaate: (a)pyyntöjen lähettäminen, (b)gradienttien asettaminen, (c) datan reititys.

- *Gradienttipohjainen reititys:* Yksi variaatio ohjatulle diffuusion toimintaperiaatelle on niin sanottu gradienttipohjainen reititys. Samalla tavalla kuin ohjatussa diffuusion toimintaperiaatessa niulusolmu lähettää pyyntöpaketteja verkkoon. Gradienttipohjaisessa reitityksessä gradientit niulusolmulle päin muodostetaan sen perusteella, kuinka monen hyppyn päässä niulusolmu on lähdesolmusta. Tämän avulla solmu voi päätellä oman etäisyytensä niulusolmusta. Gradientit muodostetaan naapurisolmujen etäisyyksien erona. Tämän jälkeen solmu lähettää paketit sille naapurisolmulle, jolla on suurin gradientti. Gradienttipohjaisessa reitityksessä myös yritetään levittää datasiirtoa eri solmuille, jotta sensoriverkon resursseja tulisi käytetyksi tasaisesti. *Energiapohjaisessa* taktiikassa solmu asettaa oman etäisyytensä suuremmaksi, kun sen energia alkaa loppua. Tällä tavalla muut solmut lähettävät harvemmin dataa tälle solmulle. *Stokastisessa* taktiikassa solmu valitsee satunnaisesti solmun, jolle reititetään, jos näillä solmuilla on yhtä suuri gradientti. *Datavirta*-taktiikassa (engl. stream-based), solmu nostaa omaa etäisyyttään, jos sen kautta kulkee jo jokin datavirta. Kaikki muut solmut, jotka eivät jo lähetä tämän solmun kautta paketteja, näkevät tämän etäisyyden noston. Mutta solmut, jotka jo lähettävät kyseisen solmun kautta, eivät näe tätä nostoa. Tällä tavalla uudet datavirrat etsivät toisen reitin niulusolmulle ja reitityksen kuormitus eri solmuille tasaantuu [9, s.175].

### 3.2.2 Hierarkiset protokollat

Hierarkisissa reititysprotokollissa kaikilla solmuilla ei ole samaa roolia reitityksessä vaan roolit vaihtelevat riippuen solmujen ominaisuuksista. Hierarkisissa protokollissa solmut jaetaan *ryhmiin* (engl. cluster), jossa on yksi *ryhmänjohtaja* (engl. cluster head). Kaikki muut solmut kommunikoivat vain ryhmänjohtajan kanssa, eivätkä keskenään. Tämän lisäksi vain ryhmänjohtajat reitittävät paketteja eteenpäin kohti niulusolmua. Solmut, jotka toimivat ryhmänjohtajina, ovat yleensä tehokkaampia ja niillä on enemmän energiaa käytettävissään. Ryhmänjohtajat voivat käyttää esimerkiksi verkkovirtaa, kun taas muut solmut toimivat pattereilla. Koska dataliikenne suunnataan näille ryhmänjohtajille, muut pattereilla toimivat solmut säästävät energiaa, koska niiden ei tarvitse reitittää muiden solmujen paketteja.

Kuva 3.7 esittää hierarkisten protokollien periaatteen. Vasemmanpuoleisessa kuvassa kaikki ryhmänjohtajat pystyvät suoraan kommunikoimaan niulusolmun kanssa. Oikeanpuoleisessa kuvassa ryhmänjohtajat eivät välttämättä kaikki pysty kommunikoimaan suoraan niulusolmun kanssa, vaan joutuvat reitittämään paketteja muiden ryhmänjohtajien kautta [9, s.180].

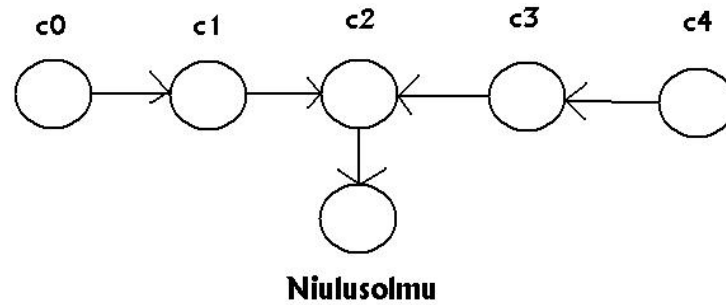


**Kuva 3.7:** Ryhmien muodostaminen hierarkisissa protokollissa. Ryhmänjohtajat kommunikoivat joko suoraan niulusolmun kanssa (vasen) tai muiden ryhmänjohtajien kautta (oikea).

Hierarkisissa protokollissa on kuitenkin omat haasteensa. Tärkein näistä on ryhmien muodostaminen, joka voi olla haastavaa varsinkin, kun solmut ovat liikkuvia. Lisäksi jos ryhmänjohtajana toimiva solmu menee rikki, siitä aiheutuu enemmän ongelmia, kuin tavallisen solmun rikkoutumisesta. Verkon jakaminen ryhmiin voi myös pidentää reititysmatkoja, koska nyt solmut eivät välttämättä aina lähetä paketteja lyhintä reittiä pitkin, vaan reititys rajoittuu vain ryhmänjohtajille [9, s.180].

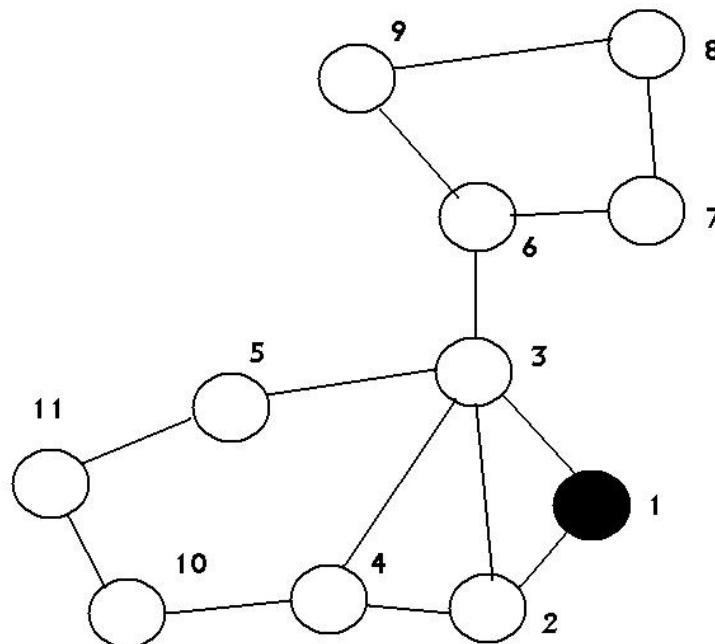
Seuraavassa käsitellään muutamien hierarkisten protokollien toimintaa.

- *LEACH, (Low Energy Adaptive Clustering Hierarchy)*: LEACH-protokollassa ryhmänjohtajaa ei valita etukäteen, vaan ryhmänjohtajaa vaihdellaan solmusta toiseen, jotta energiankulutus jakautuisi tasaisesti kaikille solmuille. Muutoin, jos ryhmänjohtajan roolia ei vaihdettaisi, tämä kuluttaisi valitun solmun energian loppuun nopeasti. Koska LEACH pyrkii tasaisesti kuormittamaan solmuja, tämä protokolla sopii hyvin sensoriverkkoihin, jossa ei ole erityisiä voimakkaampia solmuja, kuten verkkovirralla toimivia solmuja. Ryhmänjohtajat kommunikoivat suoraan niulusolmun kanssa, eivätkä käytä muita ryhmänjohtajia reititykseen. Kaikki muut solmut valitsevat mihin ryhmään ne haluavat liittyä sen perusteella, mille ryhmänjohtajalle ne pystyvät lähettämään dataa pienimmällä mahdollisella energiamäärällä. Ryhmänjohtajat pitävät radionsa jatkuvasti päällä, mutta säästääkseen energiaa muut solmut pitävät radionsa päällä vain silloin, kun niillä on dataa lähetettävänä. Vain noin 5 prosenttia sensoriverkon solmuista toimii kerralla ryhmänjohtajina. [15].
- *PEGASIS, (Power-Efficient Gathering in Sensor Information Systems)*: PEGASIS on parannettu versio LEACH-protokollasta. Sen sijaan, että protokolla muodostaisi sensoriverkossa ryhmiä, PEGASIS ketjuttaa solmut reititystä varten. Jokainen solmu reitittää dataa vain lähimpien naapureiden kautta. Näin saavutetaan energiansäästöä, kun voidaan käyttää mahdollisimman pientä lähetystehoa seuraavan naapurin saavuttamiseen ketjussa. Vain yksi solmu reitittää suoraan dataa niulusolmulle. Koska PEGASIS-protokollassa ei tarvitse käyttää resursseja ryhmien muodostamiseen ja käytössä on monihyppyteknikka, protokollan suorituskyky on parempi kuin LEACH-protokollassa. Sen sijaan pakettien reitittämisessä voi olla huomattavia viiveitä, varsinkin jos paketti lähetetään ketjun loppupäästä. Lisäksi, koska vain yksi solmu toimii linkkinä niulusolmulle, tämä voi toimia pullonkaulana dataliikenteelle. Kuva 3.8 esittää protokollan toiminnan. Solmut c0 ja c4 reitittävät paketteja vastaavasti solmuille c1 ja c3. Solmut c1 ja c3 lisäävät pakettiin oman datansa ja lähettävät paketin eteenpäin solmulle c2. Solmu c2 toimii linkkinä niulusolmulle [24].



**Kuva 3.8:** Solmujen ketjuttaminen PEGASIS-protokollassa.

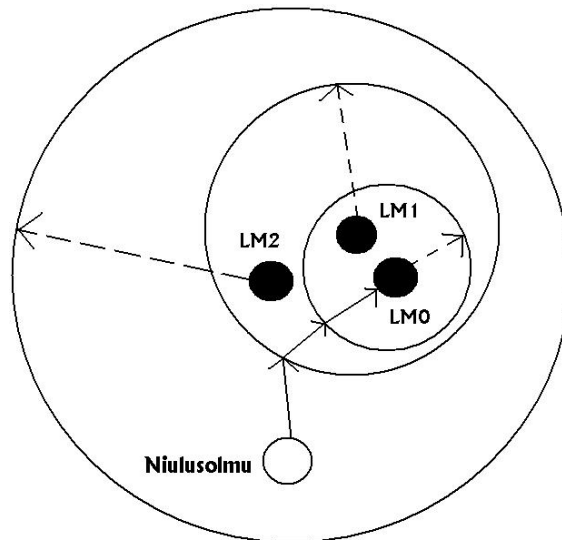
- *Landmark routing:* Tässä reititystekniikassa solmut käyttävät hierarkisia osoitteita toisilleen. Landmarktai maamerkki on solmu, johon tietyn etäisyyden päässä olevat solmut tietävät osoitteen. Esimerkiksi kuvassa 3.9 solmut 2-6 tietävät reitin solmulle 1. Mutta samalla solmut 7-11 eivät tiedä reittiä solmulle 1. Tällöin solmu 1 on maamerkki, joka on näkyvissä kaikille solmuille, jotka ovat korkeintaan kahden hypyn päässä. Tällöin siis solmu 1 on maamerkki, jolla on säde 2.



**Kuva 3.9:** Maamerkit landmark routing -tekniikassa.

Reitittäminen tapahtuu käyttäen näitä maamerkkejä. Solmun osoite sensoriverkossa on sarja maamerkkejä, jotka ovat lähinnä kyseistä solmua. Esimerkik-

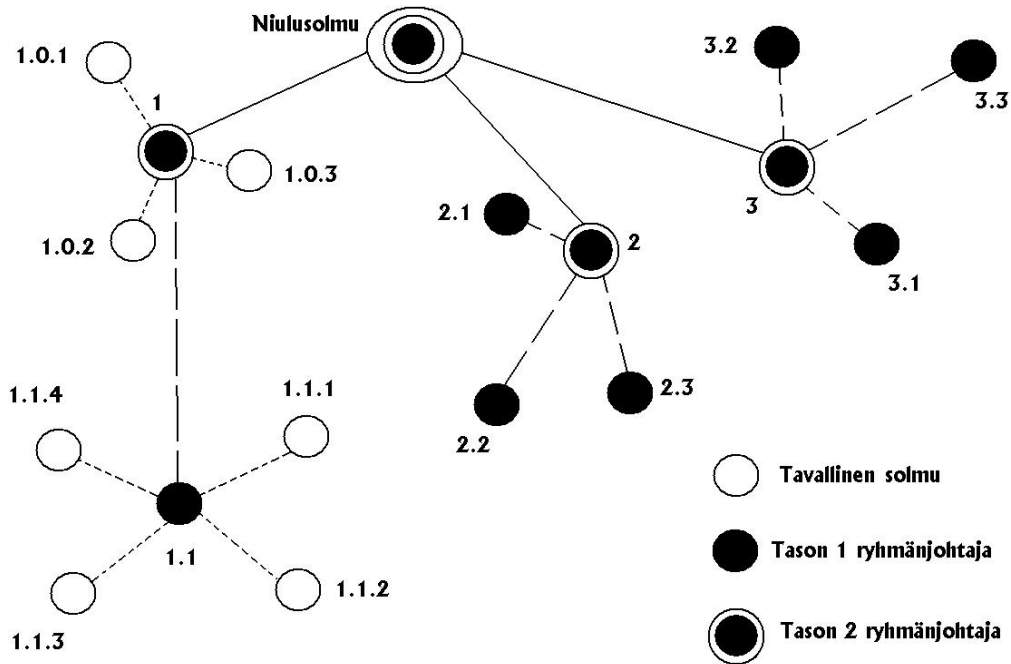
si kuvassa 3.10 solmun LM0 osoite on LM2.LM1.LM0. Jos lähdesolmu haluaa reitittää solmulle LM0, mutta sillä ei ole reititystaulukossa reittiä maamerkeille LM1 tai LM0, se lähettää datapaketin solmulle LM2, jolle lähdesolmu löytää reitin. Jokainen tällä reitillä tekee saman reitityspäätöksen, kunnes paketti saapuu solmulle, joka tietää reitin solmulle LM1. Tällöin solmu lähettääkin paketin kohti solmua LM1. Vastaavasti, kun paketti saapuu solmun LM0 lähistölle, valitaan reitti kohti solmua LM0 [32].



**Kuva 3.10:** Maamerkkien käyttö reitityksessä landmark routing -tekniikassa.

- *TEEN (Threshold sensitive Energy Efficient sensor Network)*: Myös TEEN-protokollassa solmut muodostavat ryhmiä, joissa tavalliset solmut kommunikoiivat vain ryhmänjohtajien kanssa. Ryhmänjohtajat lähettävät kaikille ryhmän jäsenille kaksi eri kynnyisarvoa. Ensimmäinen kynnyisarvo, niin sanottu kova kynnyisarvo, kertoo kuinka suuri tietyn sensoriarvon täytyy olla, ennen kuin solmu lähettää sensoriarvot ryhmänjohtajalle. Toinen kynnyisarvo, jota nimitetään pehmeäksi kynnyisarvoksi, kertoo kuinka paljon tietty sensoriarvo täytyy muuttua, ennen kuin solmu lähettää sensoridataa. Molempien ehtojen täytyy toteutua, jotta sensoridataa voi lähettää. Kova kynnyisarvo varmistaa sen, että dataliikennettä syntyy vain silloin, kun sensoriverkko havaitsee jotain mielenkiintoista. Pehmeä kynnyisarvo sen sijaan vähentää ylimääräisiä lähetyksiä. Nämä kynnyisarvot siis vähentävät lähetyksiä ja näin säästävät energiaa. Kuitenkin, koska protokolla reagoi nopeasti muuttuviin sensoriarvoihin, se sopii hyvin käytettäväksi aikakriittisissä sovelluksissa. TEEN-protokollassa on kaksi erilaista ryhmänjohtajaa. Tason 1 ryhmänjohtajat keräävät datat ryhmänjäseniltä ja lähettävät ne eteenpäin tason 2 ryhmänjohtajille. Nämä tason 2 ryhmänjohtajat pystyvät kommunikoimaan suoraan niulusolmulle. Joten toisin kuin LEACH-protokollassa, kaikki

ryhmänjohtajat eivät kommunikoi suoraan niulusolmun kanssa. Myös tason 2 ryhmänjohtajat voivat muodostaa ryhmiä tavallisten solmujen kanssa. Kuva 3.11 esittää TEEN-protokollan hierarkiset ryhmänmuodostukset [26].



**Kuva 3.11:** Hierarkisten ryhmien muodostus TEEN-protokollassa.

### 3.2.3 Sijaintipohjaiset protokollat

Sijaintipohjaiset protokollat käyttävät hyväkseen solmujen maantieteellistä sijaintitietoa reititykseen. Tätä varten solmujen täytyy tietää oman sijaintinsa lisäksi myös kohteen sijainnin ja mahdollisesti myös naapureidensa sijainnit. Sen sijaan solmujen ei tarvitse tietää verkon topologiasta mitään. Kohteet tunnustetaan niiden maantieteellisten sijainnin perusteella. Kohteen sijainti voidaan saada selville esimerkiksi tulvimalla verkko paketeilla, joissa kysytään kohteen sijaintia. Kun paketti pääsee kohteelle, kohde vastaa lähettämällä sijaintitietonsa takaisin.

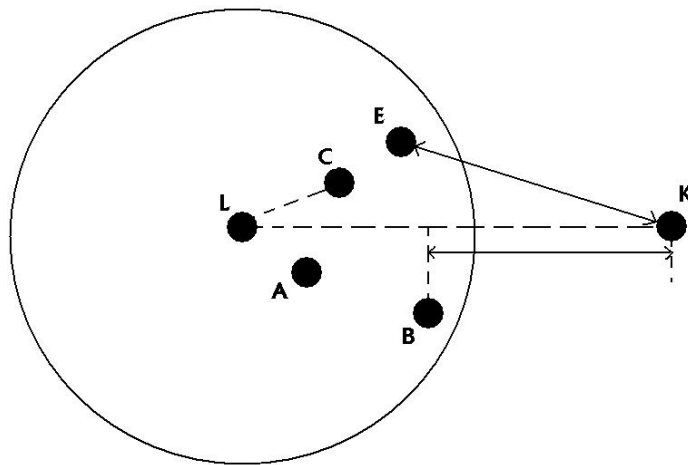
Usein solmujen identiteetin tunteminen ei ole yhtä tärkeää kuin niiden sijainnin tunteminen. *Geocasting*-tekniikalla paketteja lähetetään solmuille, jotka sijaitsevat tietyllä alueella. Näin voidaan kyselypakettien lähettämistä vähentää, kun ei tarvitse koko verkolle lähettää kyselyitä. Kun paketti saapuu halutulle alueelle, se voidaan reitittää kaikille alueen solmuille (*multicast*) tai ainakin yhdelle alueen solmuista (*anycast*).

Sijaintipohjaisien protokollien hyvänä puolena on se, että vain maantieteellistä sijaintia tarvitaan reititykseen eikä tarvitse ylläpitää reititystaulukoita. Lisäksi tarve kontrollipakettien käyttöön on pienempi kuin muissa protokollissa.

Sijaintipohjaisissa protokollissa voidaan käyttää monta eri taktiikkaa seuraavan hypyn valitsemiseen. Seuraavassa listataan yleisimpiä tekniikoita [9, s.183-184, 186].



- *Greedy*: Tässä taktiikassa datapaketti lähetetään naapurille, joka on lähinnä haluttua kohdetta. Näin pyritään minimoimaan lähetyksien määrä. Kuvassa 3.12 tämä olisi solmu A. Tämän taktiikan huonona puolena on se, että paketti voi saapua solmulle, jolla ei ole yhtään naapuria, joka veisi paketin lähemmäs kohdetta. Tällaisien aukkojen ympäri reitittäminen on yksi tärkeimmistä haasteista sijaintipohjaisissa protokollissa.
- *Nearest with Forwarding Progress (NFP)*: NFP-taktiikassa solmu lähettää datapaketin sille naapurille, joka on kaikkein lähinnä lähettävää solmua ja joka on kuitenkin lähempänä kohdesolmua. Kuvassa 3.12 tämä olisi solmu B. Taktiikan tarkoituksena on käyttää mahdollisimman pientä lähetystehoä, jotta törmäyksiä muiden lähetyksien kanssa tapahtuisi vähemmän. Tämä edellyttää, että solmuilla on radiot, joiden lähetystehoä voidaan säätää.
- *Most Forwarding Progress within Radius (MFR)*: MFR-taktiikassa lähdesolmu lähettää datapaketin sille naapurille, jolle lähettäminen saavuttaa eniten edistystä kohti kohdesolmua. Tämä edistys määritellään etäisyydeksi, joka saadaan projisoimalla lähdesolmun ja naapurisolmun välinen etäisyys suoralle viivalle lähdesolmusta kohdesolmuun. Kuvassa 3.12 tämä naapuri on solmu C. Taktiikan tarkoituksena on minimoida tarvittavat lähetykset paketin saamiseksi perille.
- *Compass Routing*: Tässä taktiikassa ensin piirretään viiva lähdesolmun ja kohdesolmun välille. Lisäksi piirretään toinen viiva lähdesolmun ja naapurisolmun välille. Näiden kahden viivojen välille muodostuu kulma  $\alpha$ . Lähdesolmu lähettää datapaketin sille naapurille, jonka kulma  $\alpha$  on pienin. Kuvassa 3.12 tämä olisi solmu D.

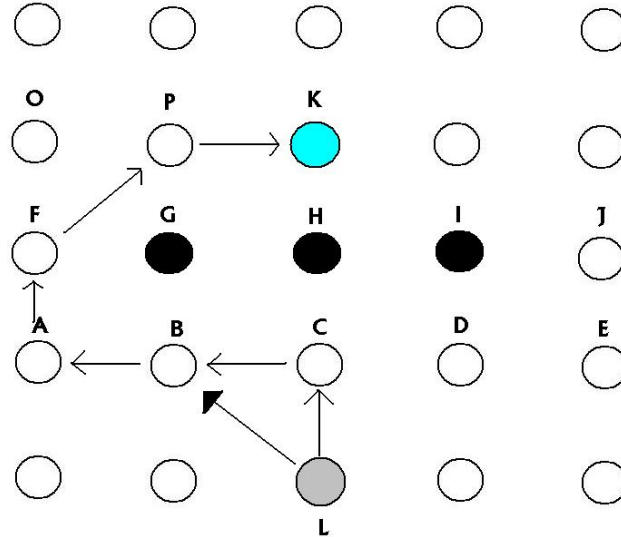


**Kuva 3.12:** Eri taktiikoita seuraavan hypyn valitsemiseen sijaintipohjaisissa protokollissa. Solmu L pyrkii reitittämään dataa solmulle K.

- *Minimum Energy Mobile Wireless Networks (MEMWN)*: Nimensä mukaisesti tämä protokolla pyrkii minimoimaan käytetyn energian verkoissa, joissa solmut ovat liikkuvia. Jokaisen solmun on tiedettävä oma sijaintinsa esimerkiksi GPS:n avulla. Protokolla laskee jokaiselle solmulle reititysalueen. Näille reititysalueessa oleville solmuille reitittäminen kohti niulusolmua on kaikkein energiatehokkainta, joten solmu valitsee alueella sijaitsevista solmuista yhden, jolle lähettää datan. Jokainen solmu laskee paikallisesti oman reititysalueensa ja käyttäen jokaisen solmun reititysalueita muodostuu reitti niulusolmulle. Koska protokolla perustuu solmujen paikalliseen tietoon, se osaa mukautua hyvin verkossa tapahtuviin muutoksiin kuten solmujen liikkumiseen tai rikkoutumiseen. Tästä syystä se sopii hyvin sensoriverkoille, jossa solmut ovat liikkeessä [30].
- *Geographic Adaptive Fidelity (GAF)*: Tässä algoritmossa verkko jaetaan maantieteellisiä sijainteja käyttäen ruudukkoihin. Yhdessä ruudussa olevilla solmuilla lasketaan olevan yhtä suuri energiavaatimus niulusolmulle reitittäessä. Samassa ruudussa olevat solmut menevät vuorotellen nukkumaan ilman, että reititettävyyden ruudun kautta huononee. Tällä tavalla vain tarvittava määrä solmuja on päällä yhtä aikaa ja tehonkulutus jakautuu tasaisesti ruudun solmujen kanssa. Solmut voivat olla kolmessa eri tilassa: *aktiivinen*, *tehonsäästö* tai *etsintä*. Etsintä-tilassa solmu selvittää, mitä muita solmuja on samassa ruudussa. Aktiivinen-tilassa solmu reitittää paketteja kohti niulusolmua. Tehonsäästö-tilassa solmu sulkee radion ja menee tehonsäästötilaan. Algoritmillä saavutetaan varsinkin tiheissä verkoissa suuret energiansäästöt [12].
- *Geographic and Energy Aware Routing (GEAR)*: GEAR on geocast-tyyppinen protokolla, jossa datapaketti ensin reititetään kohdealueelle ja tämän jälkeen datapaketti reititetään kaikille solmuille tässä alueessa. Jokainen solmu verkossa ylläpitää reititystä varten kahta eri kustannuslukua. *Arvioitu kustannusluku* ottaa huomioon reitillä olevien solmujen jäljellä olevan energian sekä maantieteellisen etäisyyden kohteeseen. *Opittu kustannus* on sen sijaan tarkennus arvioituun kustannukseen, jossa otetaan huomioon verkossa mahdollisesti olevat aukot. Mikäli verkossa ei ole yhtään aukkoja, arvioitu kustannus ja opittu kustannus ovat samoja [11].

Kun paketti saapuu solmulle eikä solmulla ole yhtään naapuria, joka olisi lähempänä kohdetta, solmu voi tästä päätellä olevansa aukossa. Tässä tapauksessa solmu valitsee naapureistaan sen, jolla on pienin opittu kustannus. Tämän jälkeen solmu asettaa omaksi opituksi kustannukseksi valitun naapurin opitun kustannuksen lisättyinä kustannus paketin lähettämisestä naapurille. Tällöin solmun opittu kustannus kasvaa naapuriansa suuremmaksi, joten jatkossa paketit menevät suoraan tälle naapurille. Kuvassa 3.13 on esitettyinä

tämä tilanne. Aluksi lähdesolmu L lähettää paketin solmulle A. Solmu A huomaa olevansa aukossa ja lähettää paketin eteenpäin solmulle B. Samalla solmu nostaa opittua kustannustaan, joka tulee suuremmaksi kuin solmun B opittu kustannus. Seuraavan kerran, kun solmu S haluaa lähettää kohdesolmulle K, solmu S lähettääkin paketin solmulle B. Näin verkon aukko tullaan kierrettyksi pienemmällä matkalla [11].



*Kuva 3.13: Reitittäminen aukkojen ympäri GEAR-protokollassa.*

## 4 Sensoriteknologiasta

Tärkeä osa langattoman sensoriverkon suunnittelemista on oikeanlaisien sensorien valinta ja niiden käyttö. Sensorit täytyy osata valita muun muassa hinnan, tarkkuuden, kestävyuden ja mittausalueen perusteella. Tässä kappaleessa esitellään niiden sensoryyppien toimintaperiaatteita, joita käytetään myös tässä diplomityössä. Lisäksi pohditaan sensoreiden valintaan liittyviä seikkoja.

### 4.1 Kosteussensorit

*Suhteellinen kosteus* (engl. relative humidity, RH) tarkoittaa sitä, kuinka paljon kosteutta ilmassa on verrattuna siihen, kuinka paljon sitä voisi olla. Lämmin ilma pysyy sitomaan enemmän kosteutta kuin kylmä ilma. Suhteellinen kosteus  $RH$  voidaan myös määrittää seuraavan kaavan [29, e.12-7] avulla:

$$RH = \frac{p(H_2O)}{p'(H_2O)} \times 100\% \quad (4.1)$$

, jossa  $p(H_2O)$  on veden osapaine ilmassa ja  $p'(H_2O)$  on vesihöyryllä kyllästetyn ilman osapaine eli kyllästysosapaine.

Yleisimmät kosteussensorytyypit perustuvat kapasitiiviseen, resistiiviseen tai lämmönjohtavuuden mittaukseen.

#### 4.1.1 Kapasitiiviset kosteussensorit

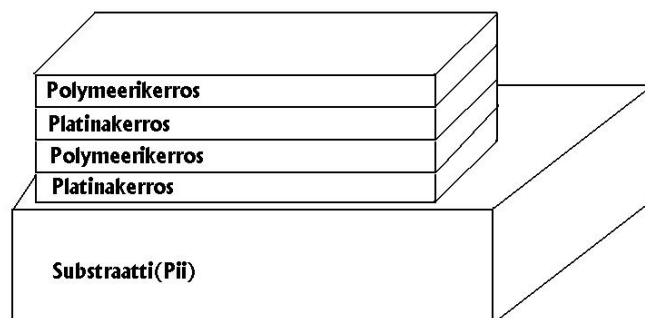
Kapasitiiviset kosteussensorit mittaavat kosteuden aiheuttamaa kapasitanssin muutosta erilaisissa kondensaattorirakenteissa. Mittauksessa kahden metallisen elektrodin välillä oleva eriste altistetaan ympäristön kosteudelle. Kondensaattorin kapasitanssi  $C$  voidaan laskea kaavasta

$$C = \epsilon_0 \epsilon_r \frac{A}{d} \quad (4.2)$$

, jossa  $\epsilon_0$  on tyhjiön permittiivisyys,  $\epsilon_r$  on eristeen suhteellinen permittiivisyys,  $A$  on kondensaattorien levyjen pinta-ala ja  $d$  niiden välinen etäisyys. Mitä suurempi kosteus on, sitä enemmän vesihiukkasia kondensaattorin levyjen välissä on ja sitä enemmän eristeen suhteellinen permittiivisyys lähenee veden permittiivisyyttä, joka on 81 [31, s.110]. Permittiivisyyden muutos voidaan havaita kapasitanssin muutoksena. Tyypillisesti kapasitanssi muuttuu 0.2-0.5 pF yhtä RH:n prosenttiyksikköä kohden [38, s.272].

Kapasitiiviset kosteussensorit ovat yleisesti käytettyjä teollisissa ja kaupallisissa sovelluksissa. Ne pystyvät ainoana kosteussensorytyyppeinä mittaamaan suhteellista kosteutta tarkasti jopa 0% asti. Niiden etuna on myös erittäin alhainen lämpötilakerroin, jolloin niitä voidaan käyttää myös korkeammissa lämpötiloissa [38, s.272].

Lämpöpysyvillä (engl. thermoset) polymeerityyppisillä kosteussensoreilla on ominaista nopea vaste, hyvä lineaarisuus, alhainen hystereesi ja hyvä pitkän ajan stabiilius. Kuvassa 4.1 on esitettyä niiden rakenne.

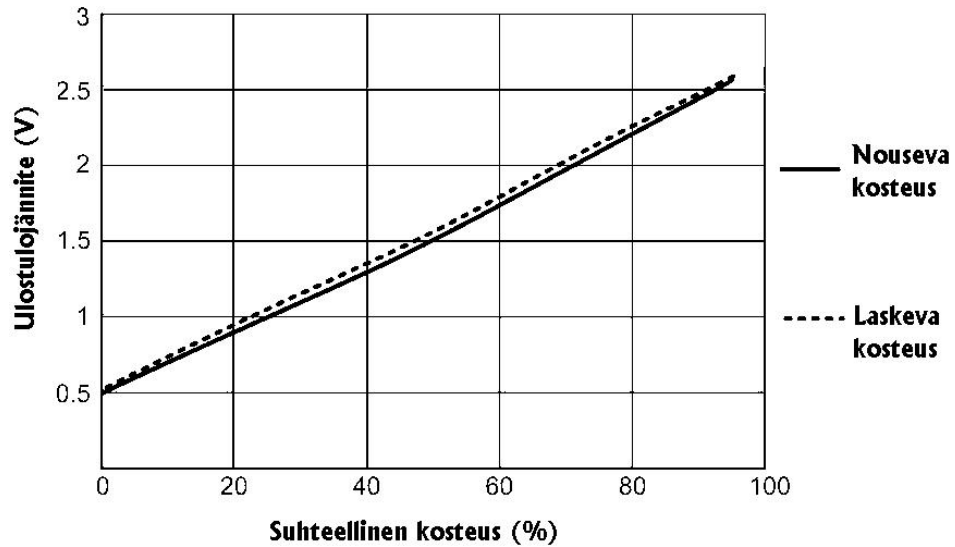


*Kuva 4.1: Lämpöpysyvien kosteussensoreiden rakenne.*

Ylempi platinakerros suojaa eristerakennetta ulkopuolisilta vaikutteilta, kun taas polymeerikerros tarjoaa mekaanista suojaa pölyltä ja liialta. Kaikki lika, joka pääsee eristerakenneseen, hidastaa sensorin vastetta kosteuden muutokseen. Lämpöpysyvät

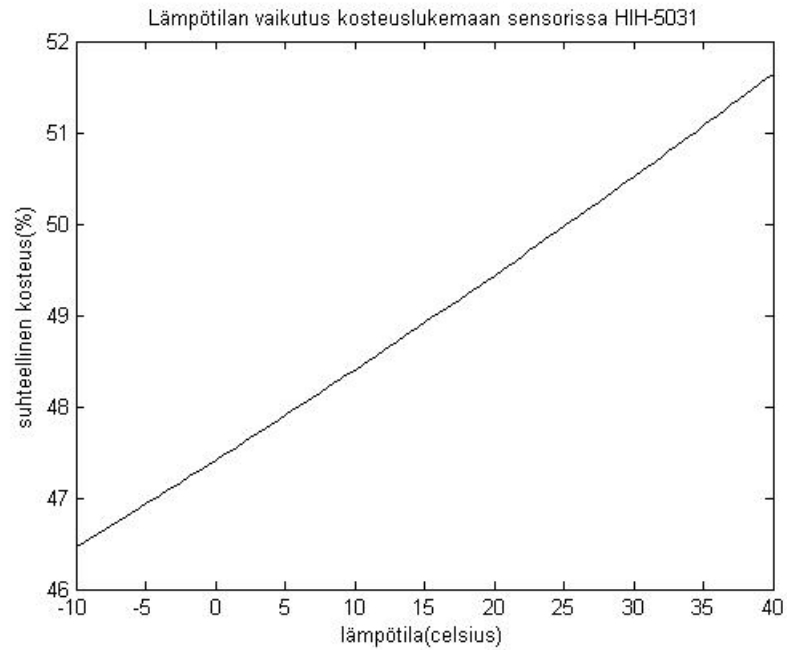
kosteussensorit toimivat suuremmissa lämpötiloissa ja ovat kestävämpiä ulkopuolisia rasitteita vastaan kuin lämpöplastiset (engl. thermoplastic) kosteussensorit [38, s.273].

Diplomityössä käytetty kosteussensori HHH-5031 on lämpöpysyvä sensori, jolla on erittäin hyvä lineaarisuus ja pieni hystereesi kuten kuvassa 4.2 näkyy.



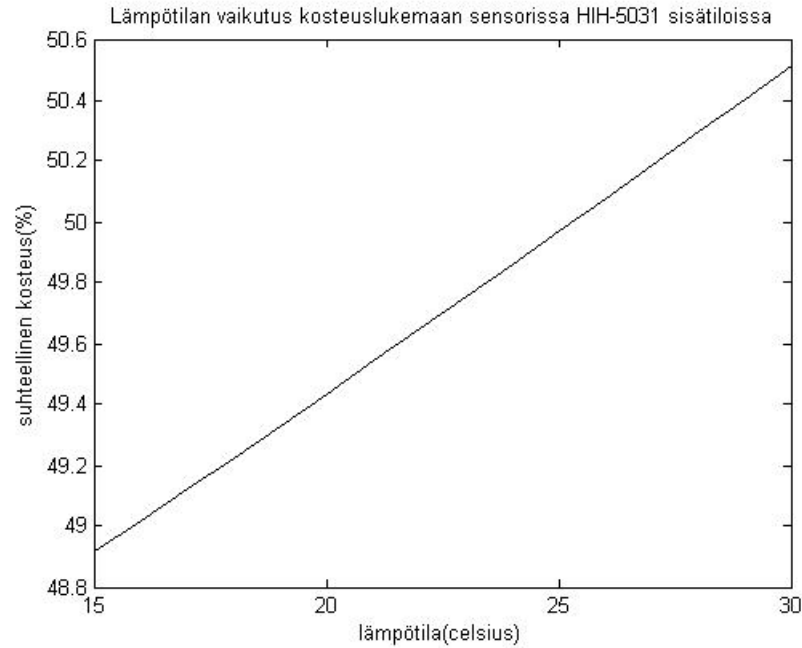
**Kuva 4.2:** Kosteussensorin antama jännite suhteellisen kosteuden funktiona [16].

Lisäksi kyseisellä sensorilla on suhteellisen pieni lämpötilakerroin kosteuden lukemalle kuten kuvassa 4.3 näkyy. Kuvassa on piirrettyä sensorin antama suhteellisen kosteuden lukema, kun todellinen arvo on 50% ja lämpötila vaihtelee -10 ja +40 asteen välillä.



**Kuva 4.3:** Lämpötilan vaikutus kosteussensorin lukemaan.

Oikeasti tässä diplomityön sovelluksessa kosteussensori tulee olemaan koko ajan sisätiloissa, jolloin lämpötilan vaihtelut eivät tule olemaan niin suuria kuin mitä kuvassa 4.3 on esitetty. Sen sijaan kuvassa 4.4 on näytetty sama kuvaaja, mutta vain 15 ja 30 celsiusasteen välillä. Tämä on realistisempi vaihteluväli lämpötilalle sisätiloissa. Huomataan, että kosteuskokema vaihtelee vain 50.5% ja 49% välillä. Tämä on riittävä tarkkuus moniin sovelluksiin ilman, että täytyy tehdä mikroprosessorilla lämpötilakompensointia.



**Kuva 4.4:** Lämpötilan vaikutus kosteussensorin lukemaan sisälämpötiloissa.

#### 4.1.2 Resisttiiviset kosteussensorit

Resistiiviset kosteussensorit mittaavat kosteuden aiheuttamia impedanssin muutoksia esimerkiksi johtavissa polymeereissä tai suoloissa. Tyypillisesti näillä sensoreilla on eksponentiaalinen vaste kosteuden muutokseen, mutta ne voidaan linearisoida analogisin tai digitaalisin keinoin. Resisttiiviset kosteussensorit ovat halpoja ja pienikokoisia, mutta ne rikkoutuvat helpommin joutuessaan kosketuksiin ulkomaailman lian kanssa [38, s.273,278].

#### 4.1.3 Lämmönjohtavuus kosteussensorit

Lämmönjohtavuuteen perustuvat kosteussensorit käyttävät hyväkseen sitä seikkaa, että vedellä on suhteellisen iso lämpökapasiteetti, jolloin kosteassa ympäristössä oleva NTC-termistori (Negative Temperature Coefficient) lämpenee hitaammin kuin kuivassa oleva termistori. Koska NTC-termistorin resistanssi muuttuu lämpötilan muutoksen mukaan, myös kosteuden muuttuminen nähdään resistanssin muuttumisena. Tämä resistanssin muutos voidaan mitata Wheatstonen sillalla käyttäen kahta NTC-termistoria, jossa toinen termistori on suljettu kuivaan tyypeen ja toinen on alistettu ympäristölle. Tällä tavalla saadaan selville ilman absoluuttinen kosteus [38, s.274].

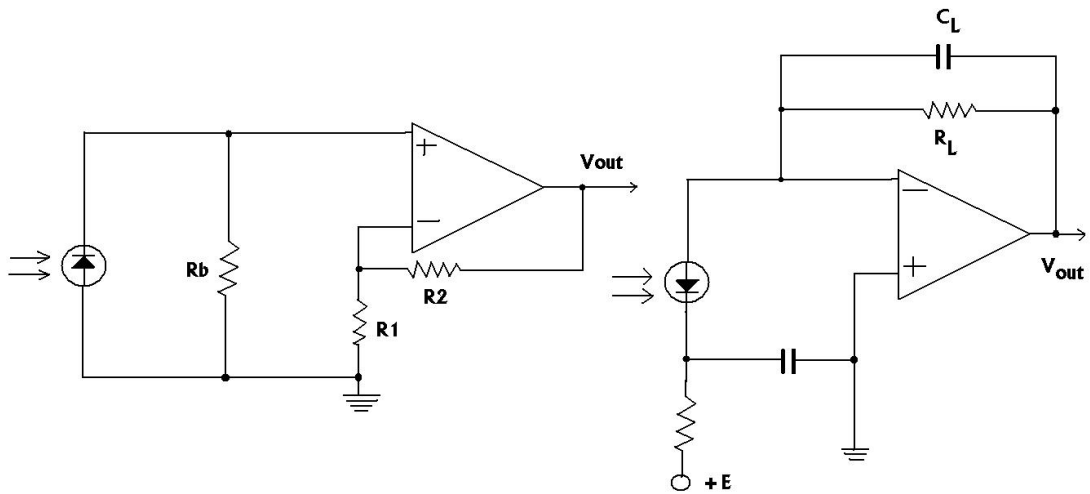
Lämmönjohtavuuteen perustuvat sensorit ovat kestäviä ja niillä voi saavuttaa paremman resoluution kuin kapasitiivilla tai resistiivisillä kosteussensoreilla. Toisaalta tämä sensori reagoi kaikkien kaasujen kanssa, joiden lämpöominaisuudet ovat erilaisia tyyppien kanssa. Tämä voi aiheuttaa mittausvirheitä [38, s.278].

## 4.2 Valosensorit

Valon havaitseminen ja sen kirkkauden mittaaminen on yksi tärkeimmistä ja tavallisimmista sensorioperaatioista. Tämän vuoksi valosensoreita on vuosien varrella tutkittu ja kehitetty runsaasti ja nykyään on saatavilla laajoja valikoimia halpoja valosensoreita.

Valosensorit mittaavat valoisuutta kahdella eri tavalla. Joko niiden mittaustapa perustuu valosähköiseen ilmiöön tai valon absorptioon. Valosähköisessä ilmiössä valon energia vapauttaa materiaalin rakenteesta elektronin. Tämä voidaan havaita virran muutoksena. Sen sijaan absorptiossa valon energia muutetaan materiaalissa lämmöksi, jonka jälkeen materiaalin lämpötila mitataan [38, s.307].

Valodiodeja voidaan biasoida ja käyttää kahdella eri tavalla. Kuvassa 4.5 on esitetty nämä kaksi tapaa nimeltä *valosähköinen* (engl. photovoltaic) ja *valojohtavuus* (engl. photoconductive). Valosähköisessä tavassa valodiodia ei biasoida, jolloin vuotovirta diodin läpi pienenee. Täten diodin läpi kulkeva virta johtuu vain valon aiheuttavasta virrasta, jolloin sensori antaa tarkempia lukemia. Haittapuolena tässä on hitaampi vasteaika johtuen diodin kapasitanssista, joka aiheuttaa viivettä [38, s.308-309].



**Kuva 4.5:** Valosähköinen(vasen) ja valojohtavuus(oikea) mittaustavat .

Valojohtavuus-mittaustavassa valodiodi on biasoitu, jolloin PN-liitoksen tyhjennysalue kasvaa ja samalla kapasitanssi pienenee. Täten saadaan nopeampi vaste, mutta vastaavasti vuotovirta suurenee, jolloin heikkojen valoiskuuksien havainnoitseminen huononee [38, s.309].

Valon kirkkaus mitataan lukseissa, joka kertoo kuinka suuri valoteho osuu tiettyä pinta-alaa kohti. Luksin yksikkönä on täten  $lm/m^2$  eli lumen per neliömetri. Tyypillisiä luksiarvoja eri valaistustilanteissa on esitetty taulukossa 4.2. Taulukosta nähdään, että sisätiloissa valaistus ei juurikaan ole suurempi kuin 1000 luksia, ellei



aurinko paista suoraan ikkunasta sisään. Täten diplomityössä käytetyn valosensorin mittausalueeksi onkin asetettu 0-1000 luksia. Kirkkaimmissa valaistuksissa sensorin luksimäärä satureoituu noin 1000 luksiin, mutta tämän mittausalueen pitäisi olla riittävä sisätiloissa.

**Taulukko 4.2:** Valoisuuden luksimäärä eri tilanteissa [35].

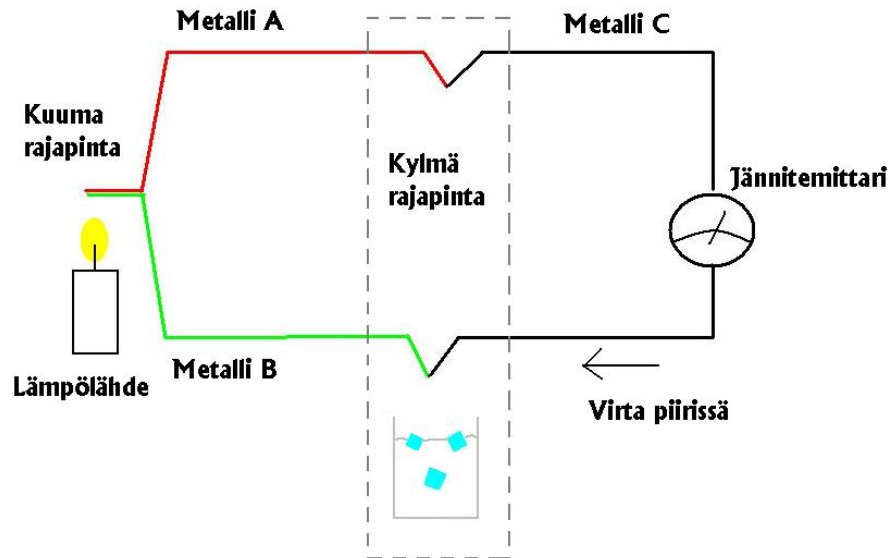
Valoisuus(luksi)	Esimerkki
1	Täysikuu yöllä päiväntasajalla
50	Tyypillinen olohuonevalaistus
80	Käytävävalaistus
320-500	Toimistovalaistus
1000	Pilvinen päivä

Yksi tärkeä seikka sopivaa valosensoria valittaessa on ympäristön lämpötilan huomioiminen. Sen lisäksi, että valosensorissa vapautuu varauksenkuljettajia valon takia, myös lämpö voi vapauttaa varauksenkuljettajia. Mitä pienempi valodiodin kynnysjännite on, sitä enemmän lämpöenergia pystyy vapauttamaan varauksenkuljettajia. Vaikka piimateriaalista tehty valodiodi toimii hyvin huoneenlämpötilassa, 100 asteen lämpötilassa diodin läpi kulkee huomattavasti virtaa ilman valoakin. Muista materiaaleista tehdyistä diodeilla voi olla suurempi kynnysjännite, jolloin ne toimivat korkeammilla lämpötiloilla paremmin [38, s.311].

### 4.3 Lämpösensorit

Lämpötilalla on suuri vaikutus materiaalien ominaisuuksiin ja niiden prosesseihin molekyyalitasolla. Tämän takia lämpötila on yksi mitatuimmista suureista myös sensoriverkoissa. Lämpösensorit mittaavat lämpötilan aiheuttamia muutoksia johonkin toiseen fyysiseen suureeseen kuten resistanssiin tai jännitteeseen. Lämpösensorit mittaavat lämpötilaa joko suoralla fyysisellä kontaktilla tai ne mittaavat kohteen lähettämää infrapunasäteilyä. Infrapunamittaus toimii hyvin kiinteiden aineiden ja nesteiden mittaamiseen, mutta ei juurikaan toimi läpinäkyvien aineiden, kuten kaasujen lämpötilan mittaamiseen [38, s.531].

*Termoparien* mittaustapa perustuu *lämpösähköiseen ilmiöön*, jossa lämpötilaerot aiheuttavat materiaalissa sähkövirran ja näin myös jännitteen. Materiaalin kuumassa puolessa olevat varauksenkuljettajat diffusoituvat kylmälle puolelle näin aiheuttaen sähkövirran. Sähkövirran aiheuttama jännite voidaan mitata käyttämällä kuvan 4.6 mukaisesti kahta eri metallia, joiden päät ovat yhdessä mittauspisteessä. Lämpötilaero aiheuttaa hieman eri jännitteen molemmissa metalleissa, johtuen metallien erilaisuuksista. Tämä jännite-ero voidaan mitata, ja se on verrannollinen lämpötilaan [37].



*Kuva 4.6: Termoparien toimintaperiaate.*

Termoparit ovat kaikkein yleisin sensorityyppi lämpötilan mittaamiseen. Niiden etuna on laaja mittausalue verrattuna muihin sensoreihin ja ne ovat erittäin kestäviä. Lisäksi termopareja voidaan helposti valmistaa paikan päällä hitsaamalla tai juotamalla. Termopari on myös kaikkein halvin sensorityyppi, eikä se tarvitse ulkoista tehonlähdettä toimiakseen [2].

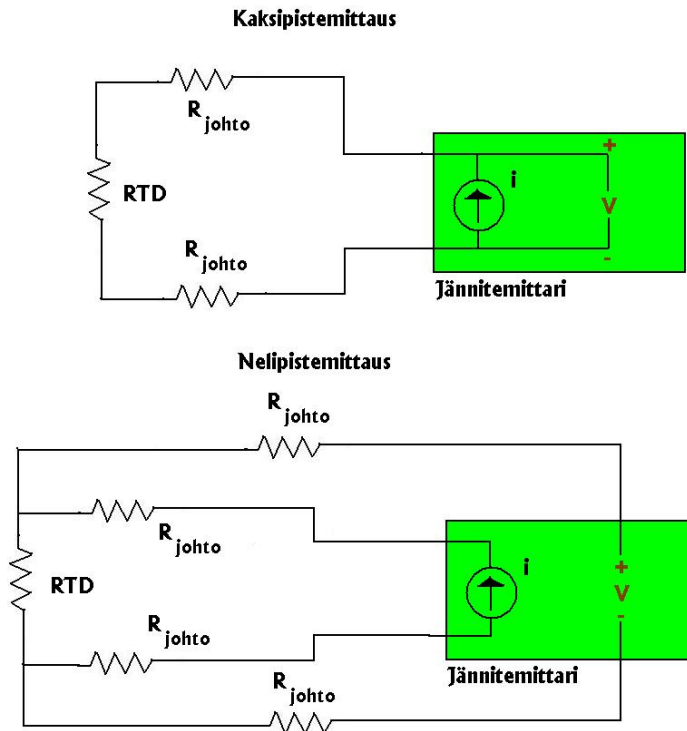
Termoparien huonona puolena on niiden epälineaarinen vaste annetun ulostulojännitteen ja lämpötilan välillä. Lisäksi lämpösähköisen ilmiön aiheuttamat jännite-erot ovat pieniä - vain noin  $5\mu V$  yhtä celsiusastetta kohti. Termoparit ovat vähiten herkkiä ja vähiten stabiileja kaikista lämpöensensoreista [2].

*Termistori* on vastus, jonka resistanssi muuttuu lämpötilan vaikutuksesta. Termistori on kaikkein herkin lämpötilan vaihtelulle kaikista lämpöensensoreista aiheuttaen suurimman parametrin muutoksen lämpötilan takia. Suuren herkkyyden hintana on lineaarisuuden menettäminen, sillä termistorit ovat jopa epälinearisempia kuin termoparit. Termistorin vaste riippuu suuresti niiden valmistusprosessin parametreista, minkä takia ei ole mitään standardoituja termistorikäyriä lämpötilalle samalla tavalla kuin termopareille [2]. Lineaarisuutta voidaan kuitenkin parantaa käyttämällä Wheatstonen siltaa [38, s.534]. Termistorit ovat erittäin pieniä ja näin ne reagoivat nopeasti lämpötilan muutoksiin. Toisaalta pienen kokonsa vuoksi termistorit voivat lämmittää itseään, sillä ne vaativat virtalähteen toimiakseen. Tämä virta voi helposti muuttaa termistorin lämpötilaa [2].

Verrattuna termopareihin termistorit ovat tarkempia mutta kalliimpia. Lisäksi termistorit eivät pysty mittaamaan yhtä laajalla lämpötila-alueella [2].

Vastuslämpötila-anturit (engl. resistance temperature detector, RTD) toimivat samalla periaatteella kuin termistorit eli niiden resistanssi muuttuu lämpötilan seurauksena. RTD:ien tyypillinen resistanssiarvo huoneenlämpötilassa on sadan ohmin

luokkaa siinä, missä termistoreilla vastaava resistanssiarvo on kilo-ohmeissa. Koska RTD:issa resistanssi muuttuu tyypillisesti vain 0.385 ohmia astetta kohden, kaikki ylimääräinen johdotuksen aiheuttama resistanssi aiheuttaa suuret virheet mittaukseen. Tämän takia RTD:ien kanssa on käytettävä nelipistemittausta tavallisen kaksipistemittauksen sijasta kuvan 4.7 mukaisesti. RTD on kaikkein tarkin ja stabiilein lämpötilasensori ja on lineaarisempi kuin termopari tai termistori. RTD on kuitenkin kallis ja hidas mittaamaan lämpötilan muutoksia, joten se soveltuu parhaiten sovelluksiin, jossa tarkkuus on tärkeintä mutta hinta ja nopeus toissijaisia [2].



**Kuva 4.7:** Kaksipistemittauksessa johdotuksen muodostama resistanssi aiheuttaa suuret mittausrvirheet. Nelipistemittauksessa on erilliset johtimet itse mittausta varten, joissa ei kulje virtaa. Täten johdotuksien yli ei jää mittausta vääristävää jännitettä.

Lopuksi digitaalinen lämpösensori integroidussa piirissä tarjoaa kaikkein lineaarisimman vasteen lämpötilan ja ulostulojännitteen kanssa. IC-sensori on helpokäyttöinen tapa mitata lämpötilaa ja myös suhteellisen halpa [2]. Lämpötilan mittausta perustuu PN-liitoksen eli diodin kynnsjännitteen muuttumiseen lämpötilan vaikutuksesta. Diodin kynnsjännitettä käytetään IC-piireissä mittaamaan lämpötilaa, koska sen integroiminen piiriin on helppoa ja halpaa [36].

Sen sijaan IC-sensorit ovat fyysisesti isoja, jonka takia ne ovat hitaita reagoimaan lämpötilan muutoksiin. Lisäksi koska nämä IC-sensorit ovat puolijohdepohjaisia, niillä on rajoitettu lämpötila-alue, jossa niitä voi käyttää. IC-sensorit ovatkin parhaimmillaan huoneenlämpötilassa [2].

Useat IC-sensorit tekevät itse signaalinkäsittelyä jättäen mikrokontrollerin vapaaksi tekemään muita töitä. Lisäksi liittäminen mikrokontrolleriin on usein tehty helpoksi käyttäen standardoitua kommunikointiprotokollaa [38, s.541]. Esimerkiksi tähän diplomityöhön valittu digitaalinen lämpösensori ADT7301 sisältää helpokäyttöisen SPI-rajapinnan, johon valitulla mikrokontrollerilla on integroitu SPI-lohko. Tällä tavalla lämpötilan mittaus tapahtuu vaivattomasti parilla rivillä koodia.

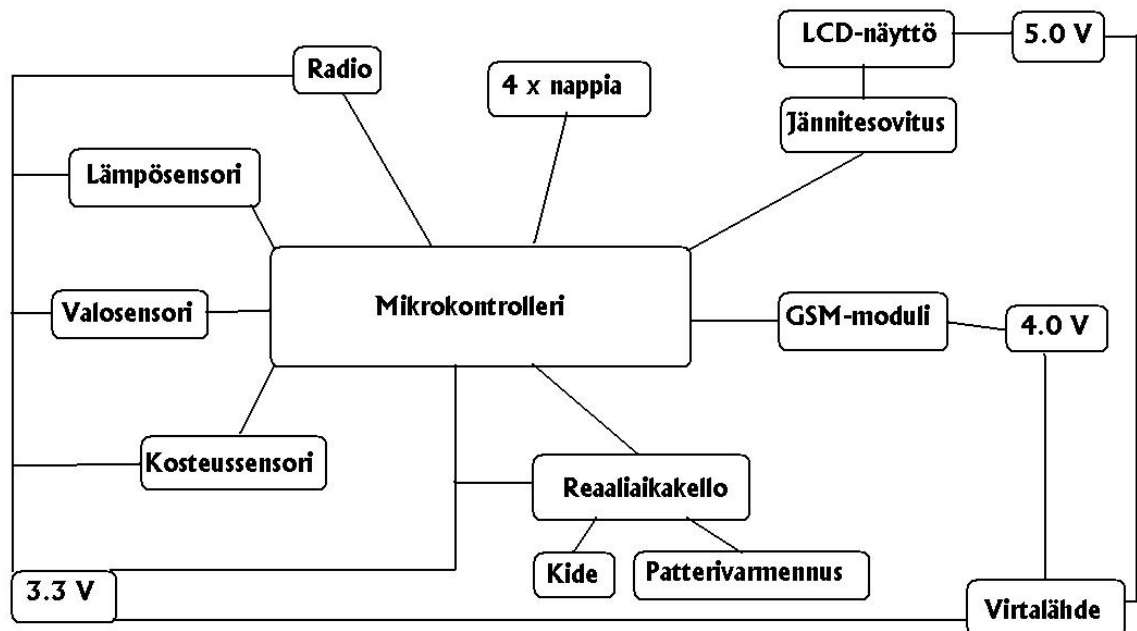
## 5 Oman sensoriverkon suunnittelu

Tässä kappaleessa käsitellään oman sensoriverkon suunnittelua. Piirikytkennän, piirilevyn ja ohjelmiston suunnittelu käydään yksityiskohtaisesti läpi.

### 5.1 Piirikytkentöjen suunnittelu

#### 5.1.1 Päämoduuli

Kuvassa 5.1 on esitettynä sensoriverkon päämoduulin lohkokkaavio. Useiden eri syiden takia mikroprosessoriksi päämoduuliin valittiin Atmelin Atmega324p. Atmega342p sisältää tarvittavan määrän pinnejä(44, joista 32 on ohjelmoitavia I/O-pinnejä) kaikkien oheiskomponenttien ohjaamiseen. Joskin osat LCD:n ohjaamiseen käytetyt pinnit ovat muussakin käytössä(SPI-rajapinnassa), koska muuten pinnit olisivat loppuneet kesken. Prosessorin muistiin tullaan tallentamaan sensorisolmujen dataa, joten oli valittava prosessori, jossa on paljon muistia. Atmega324p sisältää 32 kilotavua ohjelmoitavaa muistia, jonka pitäisi olla riittävä määrä. Alunperin oli suunnitteilla käyttää ulkoista muistia sensoridatojen tallentamiseen, mutta lopulta päätettiin prosessorin sisäisen muistin riittävän. Atmega324p sisältää myös oheislaitteiden kanssa kommunikoidusta helpottavia rautalohkoja kuten SPI- , I2C- ja USART-rautalohkot. USART-lohkoa käytetään myös laitteiston ja ohjelmiston testaamiseen. Lisäksi Atmega324p sisältää sisäisen monikanavaisen AD-muuntajan, jota käytetään valosensorin ja kosteussensorin mittauksien lukemiseen.



*Kuva 5.1: Lohkokaavio päämoduulista.*

Radioyhteyttä varten käytetään Olimexin MOD-NRF24LR-moduulia, joka on rakennettu Nordic Semiconductorsin nRF24L01P-piirin ympärille. Tämä radiopiiri toimii lupavapaalla 2,4 GHz taajuusalueella. Käytän tätä valmista moduulia, koska tarvittavan analogisen antennin ja muun piirin tarvitsema RF-suunnittelu olisi mahdollisesti liian vaikeaa ja työlästä tähän projektiin. Moduuli sisältää radiopiirin, jonka käyttö on helppoa ja ennestään tuttua tekijälle. Radiopiirin ominaisuuksiin kuuluu muun muassa mahdollisuus vaihtelevan pituisiin paketteihin, automaattinen vastaanoton kuittaus ja automaattiset pakettien uudelleenlähettykset. Lisäksi radiopiirillä voi vaihdella lähetystehoja 0 dBm ja -18 dBm välillä. Radiopiiri on kytketty prosessoriin SPI-rajapinnalla.

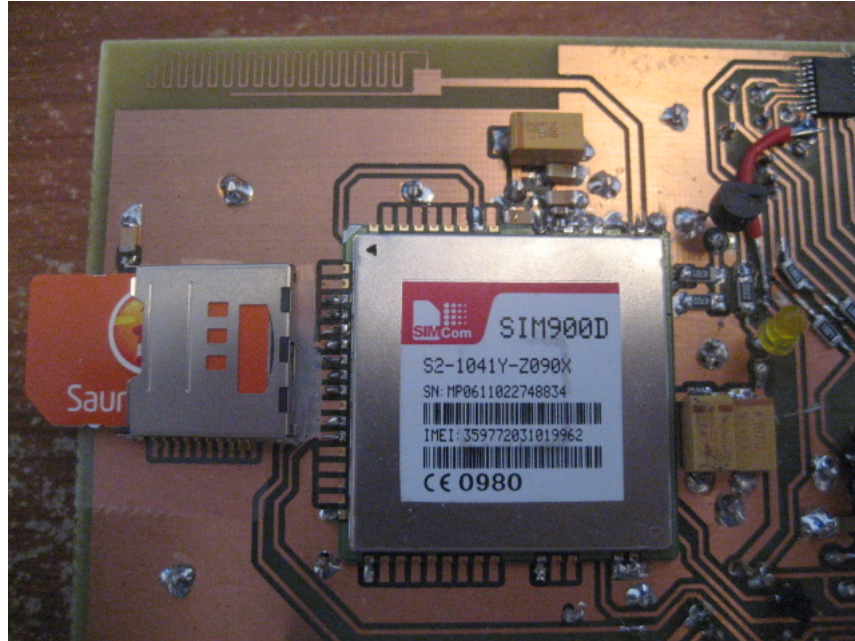
Valosensoriksi valittiin Sharpin GA1A2S100LY-sensorin. Sensori oli edullinen ja sisältää lineaarisen virta-luksi vasteen näkyvällä valolla. Lisäksi valosensorin sallitun käyttöjännitealueen (2.7 - 3.6 V) takia voidaan käyttää samaa jänniteregulaattoria prosessorin kanssa. Valosensorin ulostulojännitettä suodatetaan yksinkertaisella alipäästösuodattimella mahdollisten häiriöiden poistamiseksi.

Lämpösensoriksi valittiin Analog Devices'in ADT7301 digitaalinen 13-bittinen sensorin. Sensorilla on helppokäyttöinen SPI-rajapinta ja se on sovellukseen nähden riittävän tarkka. Tarkkuudeksi on datalehdellä ilmoitettu  $\pm 0,5\text{ }^{\circ}\text{C}$ [4]. Lisäksi komponentilla on virtaa säästävää tehonsäästötilaa, joka on hyödyllinen patterikäyttöisissä kannettavissa solmuissa. Tehonsäästötila vähentää komponentin  $1\mu\text{A}$ :in virrankulutusta  $300\mu\text{A}$ :n kulutukseen tyhjäkäynnissä. Kosteuden mittaamiseen käytetään HIH-3031 analogista kosteussensoria. Aluksi pyrin etsimään digitaalista kosteussensoria, mutta niiden hinnat olivat niin korkeita, että tyydyin analogiseen senso-

riin. Kyseisestä sensoriperheestä on olemassa myös H1H-3030-malli, jota päätin olla käyttämättä, koska siinä ei ole kosteuden tiivistymistä estävää suodatinta. Myös kosteusensorin ulostulojännitettä suodatetaan yksinkertaisella alipäästösuodattimella mahdollisten häiriöiden poistamiseksi.

Reaaliaikapiirillä (engl. real time clock, RTC) pidetään kirjaa, koska solmut ovat lähettäneet viimeksi sensoridatansa päämoduulille. RTC-piiri voi toimia patterivarmennuksella, jolloin kello toimii myös, kun virtalähdettä ei ole kytketty. Reaaliaikakellona käytetään Maximin piiriä DS1339, joka sisältää kaikki tarvittavat toiminnot I2C-rajapinnalla. RTC-piiri toimii 3.3 voltin käyttöjännitteellä.

Tekstiviestien lähettämiseen ja vastaanottamiseen käytetään Simcomin SIM900D GSM-moduulia. Kyseistä piiriä oli saatavilla yliopistolla, joten päätin käyttää sitä. Piiri toimii USART-rajapinnalla, joten sitä voi käyttää prosessorin USART-rautalohkon kautta. Antennina toimii kuvassa 5.2 näkyvä piirilevyantenni. Samassa kuvassa näkyy SIM-kortin pidike. Alunperin GSM-moduulin käyttöjännitteenä piti olla 3.5 voltia, jolloin moduuli olisi voinut käyttää samaa regulaattoria prosessorin, radiomoduulin ja parin muun oheiskomponentin kanssa. GSM-moduuli vaatii vähintään 3.4 voltia toimiakseen ja toisaalta radiomoduulin takia käyttöjännitettä ei voitu nostaa juurikaan ylemmäs. Jos radiomoduulille olisi annettu 3.6 voltin tai suurempi käyttöjännite, jota jännitetasoa myös prosessori käyttäisi kommunikoidaan radiomoduulin kanssa, olisi se rikkonut radiomoduulin. Valitettavasti säädettävän regulaattorin LM317 jännite tippuu helposti 0.1 voltia tasoon 3.4 voltia, jolloin GSM-moduuli lakkaa toimimasta. Tämän takia päätettiinkin laittaa GSM-moduulille erillinen regulaattori, joka antaa moduulille 4.0 voltin käyttöjännitteen. Samalla prosessori, radiomoduuli ja pari muuta komponenttia toimivat 3.3 voltin jännitteellä. GSM-moduuli vaati lähettäessään jopa 2.0 ampeerin virtapiikkejä, mutta LM317-regulaattori pystyy tarjoamaan vain 1.5 ampeeria enimmillään. Tämän vuoksi on laitettu kolme isoa  $470\mu F$  tantaalikondensaattoria tyydyttämään GSM-moduulin virrantarvetta.



*Kuva 5.2: Käytetty GSM-moduuli, sen antenni ja SIM-kortin pidike.*

Regulaattorina prosessorille, radiomoduulille, valosensorille ja kosteussensorille toimii Microchipin 3.3 voltin MCP1703-regulaattori, joka pystyy tarjoamaan riittävästi virtaa kaikille näille komponenteille ja jonka ulostulojännite on myös sopiva. 3.3 voltin jännite on myös riittävä Atmega324p:n 8 MHz:n sisäiselle oskillaattorille, jota aion käyttää.

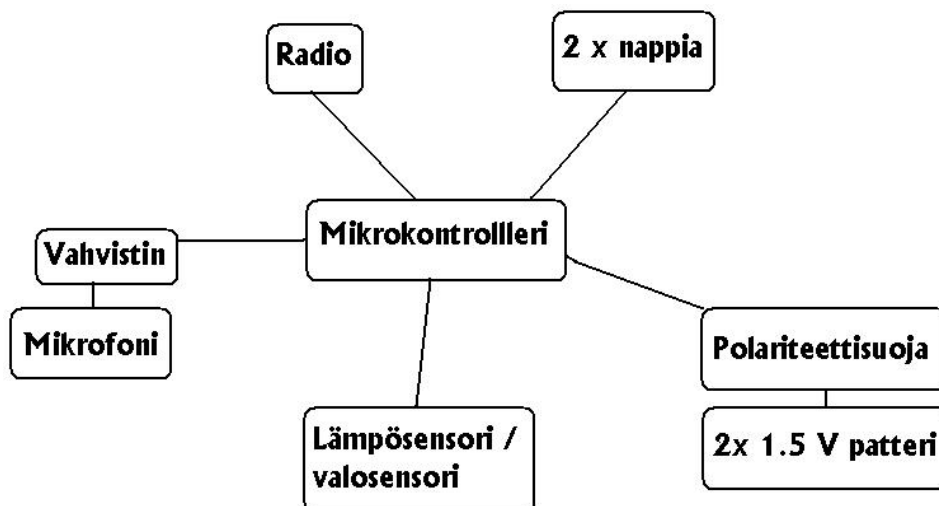
Näyttönä toimii Powertipin PC1604 neljärivinen LCD-näyttö. Jokainen rivi koostuu 16 merkistä. LCD-näyttö toimii 5 voltin käyttöjännitteellä, joten sitä varten käytetään Microchipin MCP1701-regulaattoria, joka pystyy tarjoamaan 250 mA virtaa viiden voltin jännitteellä. Mutta koska prosessori toimii 3.3 voltin jännitteellä, tarvitaan prosessorin ja LCD:n väliin jännitesovitus. Tästä huolehtii Texas Instruments TXS0108-jännitesovitinpiiri, joka pystyy muuttamaan 5 voltin jännitteet 3.3 voltin jännitteiksi ja päinvastoin. Jännitesovitinpiirissä on kahdeksan sisääntuloa ja ulostuloa, joten sitä voidaan käyttää LCD:n datalinjojen ohjaamiseen. Datalinjojen sovituksen lisäksi tarvitaan vielä jännitesovitus LCD:n kolmen kontrollisignaalin kanssa. Tämä hoidetaan kolmella erillisellä transistorilla. LCD vaatii myös negatiivisen jännitteen kontrastisäätöä varten. Tämä ei ollut tiedossa piirikytkentää tehtäessä, joten negatiivinen jännite täytyi lisätä jälkikäteen. Negatiivinen jännite saadaan tehtyä varauspumpun avulla.

Käyttöliittymää varten on neljä nappia, jotka on suodattu kytkinvärähtelyjen poistamiseksi. Virtalähteenä käytetään 230 voltin, 50 hertsin verkkoon liitettävää Ansmannin verkkovirtalähdettä. Tämä pystyy antamaan enintään 800 mA virtaa 6 voltin jännitteellä.

### 5.1.2 Kannettavat solmut

Toisin kuin päämoduuli, kannettavat solmut toimivat pattereilla, joten piirikytken suunnittelussa ja komponenttien valinnassa on otettava virrankulutus enemmän huomioon.

Kuvassa 5.3 on esitettyä lohkokaaviokuva sensoriverkon kannettavista solmuista. Alunperin kannettavien solmujen prosessorina piti olla jokin Atmelin Attiny-perheen prosessoreista. Mutta näissä prosessoreissa ei ole sisäänrakennettua USART-lohkoa, joka on tärkeä ominaisuus testaamisessa, joten päätin käyttää sitenkin Atmega-perhettä. Täten valitsin prosessoriksi Atmega88:n, joka on muutoin samanlainen prosessori, kuin alunperin valittu Attiny86, mutta siinä on myös USART-lohko. Lisäksi tehonkulutus Atmega88-prosessorissa on suurin piirtein sama kuin Attiny86:ssa, sekä prosessoreiden hinta olivat samoja. Attiny88 oli myös yksi harkittu prosessori, koska se olisi kuluttanut vähemmän tehoa kuin Attiny86 ja Atmega88, mutta tämäkään prosessori ei sisällä USART-lohkoa ja se olisi ollut hieman kalliimpi. Atmega88 sisältää myös tarvittavan SPI-rautalohkon lämpösensoria varten ja sisäisen AD-muuntajan valosensoria ja patterin mittausta varten.



*Kuva 5.3: Lohkokaavio kannettavasta solmusta.*

Eri kannettavilla solmuilla on eri sensorit. Kolmesta valmistetusta solmusta kahdella on valosensori ja lämpösensori, yhdellä on mikrofoni ja valosensori. Valosensori on sama kuin päämoduulissa eli Sharpin GA1A2S100LY -sensori. Myös lämpösensori on sama eli Analog Devices'in ADT7301 digitaalinen 13-bittinen sensoria.

Kuvassa 5.4 on esitettyä kytkentäkaavio mikrofonin vahvistuskytkentä. Kytkentä on rakennettu MCP6291-operaatiovahvistimen ympärille. Kyseinen operaatiovahvistin on "rail-to-rail" -tyyppiä, eli se pystyy laittamaan ulostulonsa erittäin lähelle käyttöjännitetasoa. Datalehden [27] mukaan se pääsee jopa 15 mV päähän käyttöjännitteitä. Tämä ominaisuus on hyvä olla patterikäyttöisessä laitteessa, jos-



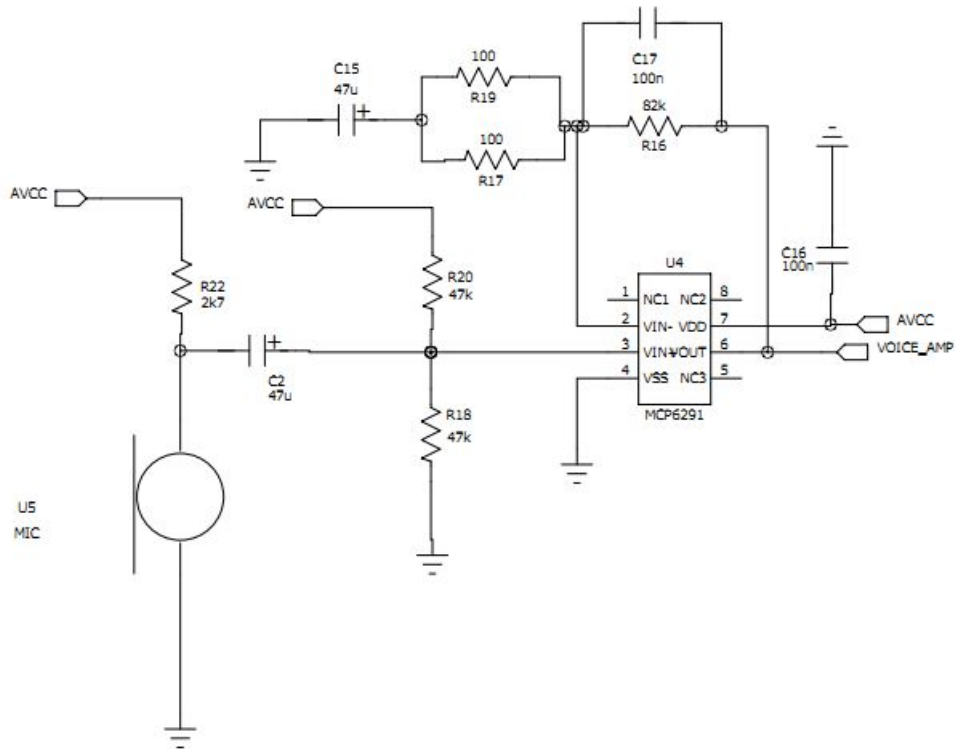
sa käyttöjännite on alhainen. MCP6291 on sisäisesti rakennettu kytkemällä kaksi operaatiovahvistinta peräkkäin. Näin tällä kaskadikytkennällä saadaan aikaiseksi suuri vahvistus - noin 80 dB 1 kilohertsin taajuudella. Koska vahvistus on niin suuri, pärjätään vain yhdellä operaatiovahvistinpiirillä. Näin säästetään komponenttimäärissä ja niiden vaatimista tilasta, joka on tärkeää varsinkin kannettavissa solmuissa. Kuvassa 5.4 vastus R22 määrää biasointivirran mikrofonille. Vastukset R18 ja R20 asettavat biasointijännitteen operaatiovahvistimen sisääntulolle käyttöjännitteen puoleenväliin. Vastukset R16, ja R19 sekä kondensaattorit C17 ja C15 muodostavat negatiivisen takaisinkytkennän. Audiotaajuuksilla vahvistus muodostuu vastuksien R16 ja R19 suhteesta. Tällöin valituilla arvoilla vahvistukseksi tulee

$$\frac{R16}{R19} = \frac{82k\Omega}{100\Omega} = 820. \quad (5.1)$$

Tällöin vahvistin vahvistaa amplitudiltaan noin 1.5 mV signaalin, joka on suurinpiirtein normaalin puhevoimakkuuden aiheuttama jännite mikrofonissa, jännitteeksi 1.2 voltia. Tämä on tarpeeksi suuri jännite prosessorin AD-muuntajalle. Kondensaattori C17 pienentää vahvistusta audiotaajuisia suuremmilla taajuuksilla, ja kondensaattori C15 pienentää vahvistusta audiotaajuisia pienemmillä taajuuksilla.

Koska operaatiovahvistinkytkentä on analoginen, se kuluttaa virtaa myös silloin, kun sitä ei käytetä. Kannettavissa solmuissa, joissa on rajallinen määrä energiaa tarjolla, tämä on ongelma. Tämän takia käyttöjännite vahvistinpiirille kulkee MOSFET-transistorin kautta, jota ohjaa prosessori. Kun halutaan mitata vahvistimen antamaa jännitettä, prosessori ohjaa MOSFET:in auki, jolloin vahvistinkytkentä saa käyttöjännitteen. Kun ei enää haluta mitata, prosessori ohjaa MOSFET:in kiinni. Tämä säästää virtaa, kun mikrofoni ei ole käytössä.

Toinen tapa, miten säästetään virtaa vahvistimen ulostulon mittauksessa, on prosessorin sisäisen komparaattorin käyttäminen. Mikäli audiovahvistimen mittaus tehtäisiin mittamaalla sen ulostuloa jatkuvasti sisäisellä AD-muuntajalla, tämä kuluttaisi huomattavasti virtaa. Tämän vuoksi käytetäänkin prosessorin sisäistä komparaattoria. Kun vahvistimen ulostulojännite nousee asetetun vertailujännitteen yläpuolelle, synnyttää komparaattori keskeytyksen sen merkiksi, että mikrofoni on havainnut jotain. Vasta tämän jälkeen aletaan mittaamaan AD-muuntimella, onko varmasti mikrofoni havainnut jotain.



**Kuva 5.4:** Vahvistinkytkentä mikrofonille.

Tehonlähteenä kannettavilla solmuilla on kaksi AA-kokoista 1.5 voltin patteria kytkettynä sarjaan, joka antaa yhteensä 3.0 voltin käyttöjännitteen solmulle. Tämä on riittävän suuri jännite kaikille solmun komponenteille. Koska käytössä on niin pieni käyttöjännite, sitä ei kannata hukata käyttämällä esimerkiksi regulaattoria. Siksi patterijännitettä käytetään suoraan muutamaa kondensaattoria lukuunottamatta ilman regulointia. Heti patterikytkennän jälkeen sarjassa on MOSFET-transistori, joka toimii suojana väärälle napaisuudelle. Mikäli käyttäjä laittaa patterit väärin päin kiinni, MOSFET ei johda virtaa läpi. Mikäli patterit ovat oikein päin, MOSFET johdattaa. MOSFET:in yli jäävä jännite oli niin pieni, ettei sitä yleismittarilla edes pystynyt mittaamaan. Täten jo alhaista jännitettä ei hukata turhaan. Oheiskomponenteista lämpösensori ja valosensori vaativat kaikkein suurimman käyttöjännitteen eli 2.7 voltia vähintään. Tämän vuoksi käytössä valosensori ja lämpösensori lakkaavat toimimasta ensimmäiseksi. Kuitenkin prosessori ja radio vaativat 1.8 ja 1.9 voltia toimiakseen, joten solmu voi jatkaa muiden solmujen datojen reitittämistä, vaikka omia sensoreita ei voi enää käyttääkään. Käyttöjännitettä mitataan prosessorin sisäisellä AD-muuntajalla. Ensin käyttöjännitteen taso tuodaan alas jännitteenjaolla ja tätä jännitettä suodatetaan kondensaattorilla. Tämän jälkeen jännite viedään prosessorin AD-muuntajalle, jossa referenssijännitteenä käytetään prosessorin itse muodostamaa 1.1 voltin jännitettä. Normaalisti referenssijännitteenä AD-muunnoksissa on käyttöjännite, mutta koska tarkoituksena on mitata itse käyttöjännitettä, se ei

voi toimia samalla referenssijännitteellä AD-muunnokselle. Käyttöjännitettä mittaamalla voidaan arvioida onko jännite riittävän suuri tekemään lämpötilamittauksia. Lisäksi reititysprotokollassa käytetään tietoa, kuinka suuri on jäljellä oleva solmun käyttöjännite. Tätä käsitellään lisää kappaleessa 5.3.4.

Kannettavilla solmuilla on kaksi nappia, joita käytetään sensoriverkon sovelluksissa. Napeissa käytetään suodatuksia kytkinvärähtelyjen poistamiseksi.

## 5.2 Piirilevyjen suunnittelu

Tässä kappaleessa käsitellään sekä päämoduulin että kannettavien solmujen piirilevyjen suunnittelua.

### 5.2.1 Päämoduuli

Kuvissa B.1, C.1, D.1 ja E.1 on esitettyä päämoduulin piirilevyn molemmat puolet. Piirilevy on reititetty enimmäkseen alapuolella ja yläpuoli on jätetty mahdollisimman ehjäksi maatasoksi. Piirilevyn koko on 172 x 115 mm.

Vasempaan yläreunaan on sijoitettu kaikki GSM-moduuliin liittyvät komponentit. GSM-antenni on sijoitettu aivan vasempaan yläreunaan, jotta sen ympärillä olisi mahdollisimman vähän kuparia heikentämässä signaalia. Lisäksi antennin alta on poistettu maataso samasta syystä. GSM-moduuli(U14) vaatii isoja virtapiikkejä lähettäessään tekstiviestejä, joten sitä varten ovat tantaalikondensaattorit(C24,C26,C28) on sijoitettu mahdollisimman lähelle moduulia. Kaikkein lähinnä GSM-moduulin käyttöjännitepinniä on pieni 0603-koteloineen keraamikondensaattori(C30), jonka tarkoituksena on tarjota virtaa nopeasti moduulille. GSM-moduulin vieressä vasemmalla on SIM-kortin pidike(U16).

LCD-näyttö on sijoitettu piirilevyn keskelle ylhäälle ja käyttöliittymän neljä nappia sen vierelle. Sekä napit ja LCD-näyttö on sijoitettu piirilevyn alapuolelle, eli eri puolelle kuin useimmat muut komponentit. Piirilevyn kuvissa ei ole esitettyä varauspumppua LCD:n kontrastisäätöä varten, koska se lisäitiin jälkikäteen piirilevyn valmistuksen jälkeen.

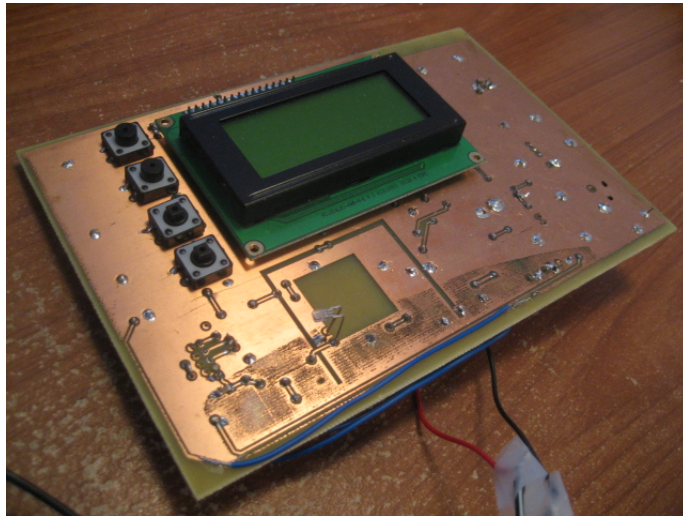
Koska piirilevy oli niin täynnä tavaraa, oli vaikeaa sijoitella analogiset komponentit (kosteussensori U3 ja valosensori U11) ja digitaaliset komponentit niin, että maavirrat kulkisivat luonnostaan erillään. Tämän vuoksi päätin erottaa maalueet fyysisesti toisistaan. Analoginen maa näkyy alhaalla oikealla neliskulmaisena alueena. Prosessorin(U1) oikeassa alanurkassa sijaitsevat pinnit sisäiselle AD-muuntajalle. Tämä reuna on juuri tarkoitettu käytettäväksi analogisen maan alueella datalehden[5, s.250] mukaan. Digitaalinen ja analoginen maa yhdistetään toisiinsa 3.3 voltin regulaattorin(U6) kohdalla. Tällä tavalla digitaalisten komponenttien maavirrat eivät kulje analogisen maan alueen kautta häiriten analogisia signaaleita. Samasta syystä 3.3 voltin käyttöjännitevedot on haaroitettu tantaalikondensaatto-

rin U6 kohdalta, jotta analogiset ja digitaaliset virrat eivät kulkisi samoja vetoja pitkin.

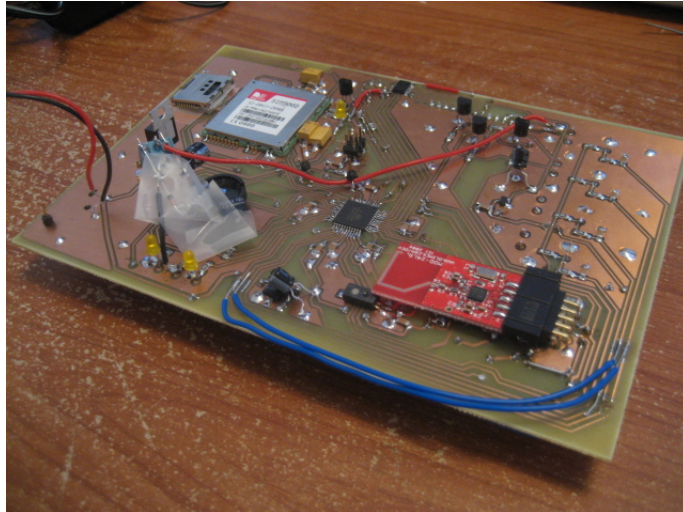
Radion sijainnin takia analogiseen maahan täytyi jättää neliskulmainen aukko, sillä radion antenni on juuri tämän alueen yläpuolelle. Käytetty 2,4GHz:n radiosignaali heijastuu ja vaimentuu helposti ohueistakin johtavista metalleista. Aukolla pyritään siihen, että radiosignaali ei vaimentuisi tai heijastuisi liikaa kuparialueen takia.

Lämpösensori on sijoitettu radiomoduulin alle, joka oli vetojen reitityksen kannalta hyvä paikka, koska sekä radiomoduuli että lämpösensori käyttävät samoja SPI-signaaleita. Myöhemmin kytkentää testatessa kuitenkin ilmeni, että tämä ei ollutkaan hyvä ratkaisu. Radiomoduulin takia lämpösensorin lukema heittää kahdella asteella todellisesta arvosta. Tämä huomattiin vertaamalla päämoduulin lämpösensorin antamaa lukemaa kannettavien solmujen ja kaupasta ostetun lämpömittarin lukemia. Lisäksi, kun radiomoduulin ottaa pois, lämpösensorin lukema näyttää hetken kuluttua oikeaa arvoa. Joten ilmeisesti radiomoduuli häiritsee lämpösensorin mittausta.

Kuvissa 5.5 ja 5.6 on esitettyä rakennettu päämoduuli.



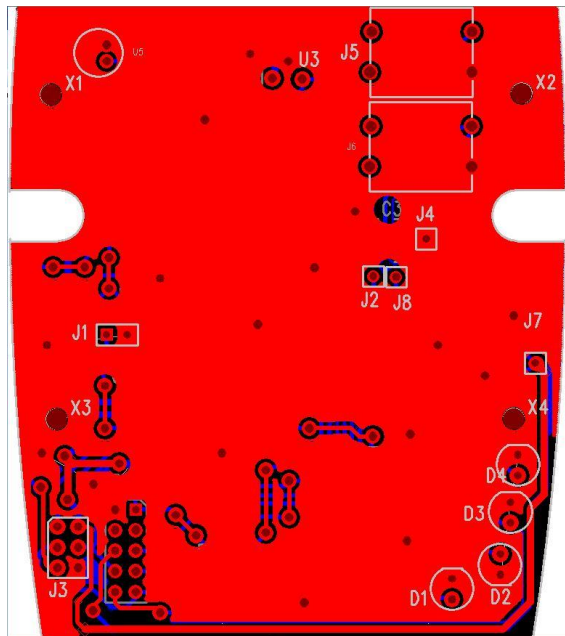
*Kuva 5.5: Päämoduuli kuvattuna maapuolelta.*



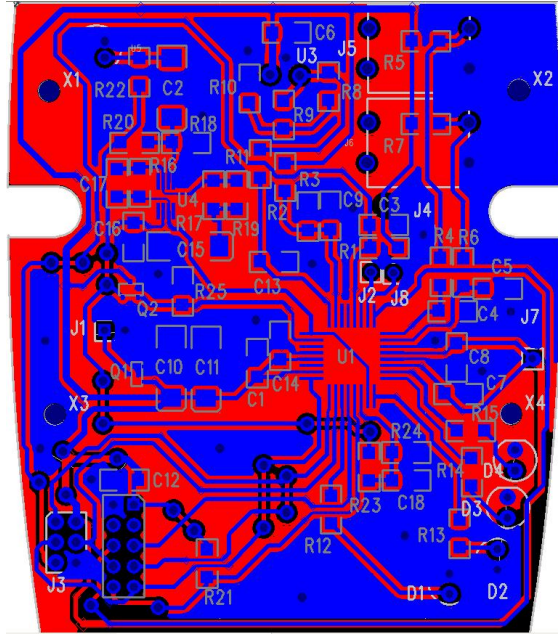
*Kuva 5.6: Päämoduuli kuvattuna reitityspuolelta.*

### 5.2.2 Kannettavat solmut

Kuvissa 5.7 ja 5.8 on esitettyä ensimmäisen valmistetun solmun piirilevyn molemmat puolet. Piirilevyn koko ja muoto määräytyi valitun Hammondin kotelon perusteella. Joskin piirilevyjä ei kaikkia koteloitu ajan ja rahan puutteen takia. Piirilevyn yläreunan leveys on 67 mm ja levyn korkeus 78,2 mm.



*Kuva 5.7: Kannettavan solmun 2 piirilevyn yläpuoli.*



**Kuva 5.8:** Kannettavan solmun 2 piirilevyn alapuoli.

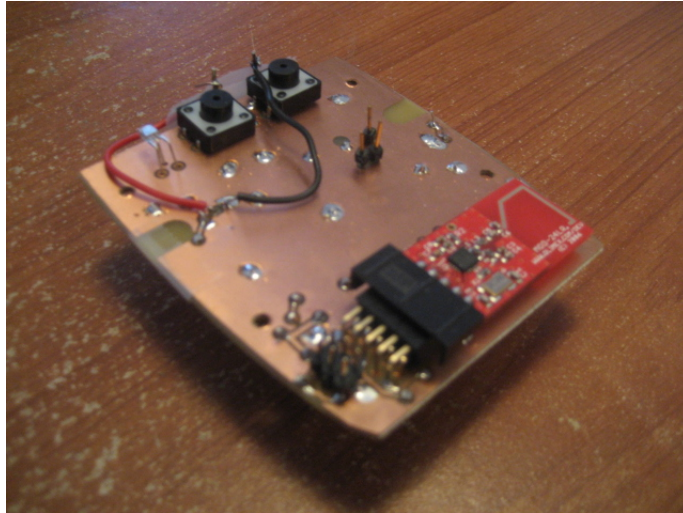
Piirilevy on reititetty enimmäkseen alapuolella ja yläpuoli on jätetty mahdollisimman ehjäksi maatasoksi. Kaikki analogiset komponentit on sijoitettu vasempaan ylänurkkaan. Siellä sijaitsevat mikrofoni, siihen liityvä operaatiovahvistin, valosensorin komponentit sekä patterinmittaukseen liittyvät komponentit. Tällä sijoituksella on pyritty siihen, että digitaalisen osuuden signaalit eivät kuljisi analogisen osuuden alueella häiriten analogisia signaaleita. Prosessorin(U1) vasemmassa ylänurkassa sijaitsevat pinnit sisäiselle AD-muuntajalle. Tämä reuna on juuri tarkoitettu käytettäväksi analogisen maan alueella datalehden[6, s.260] mukaan. Valitettavasti jostain syystä analogisen komparaattorin pinni sijaitsee prosessorin päinvastaisessa nurkassa, joten siihen on reititettävä itse prosessorin alta. Napit ovat analogisen alueen lähellä, mutta napit(J5,J6) eivät yleensä ole niin suuressa käytössä, että siitä aiheutuisi suuria häiriöitä analogiselle puolelle. Myöskin USART-liitäntä(J2,J8) on lähellä analogiselle aluetta, mutta sitä käytetään vain testaukseen, joten varsinaisessa solmun käytössä USART ei ole päällä muodostamassa häiriöitä.

Radio ja ohjelmointiliitin on sijoitettu vasempaan alanurkkaan mahdollisimman kauas analogisesta alueesta. Patteriliitin(J1) ja siihen liittyvät MOSFET-polaarisuoja(Q1) ja tantaalikondensaattorit(C10,C11) käyttöjännitesuodatukseseen on sijoitettu analogisen ja digitaalisen alueen väliin. Tällä tavalla digitaalisen puolen vetämät virrat käyttöjännitteestä eivät kulje analogisen maan-alueen kautta ja päinvastoin. Itse käyttöjännitevedot on saman periaatteen mukaisesti haaroitettu kondensaattoreiden C10 ja C11 juurista. Näin digitaalisten komponenttien käyttämät virrat eivät kulje samaa käyttöjännitevetoa pitkin analogisten komponenttien virtojen kanssa.

Testaukseen käytetyt ledit(D1-D4) on sijoitettu oikeaan alanurkkaan.

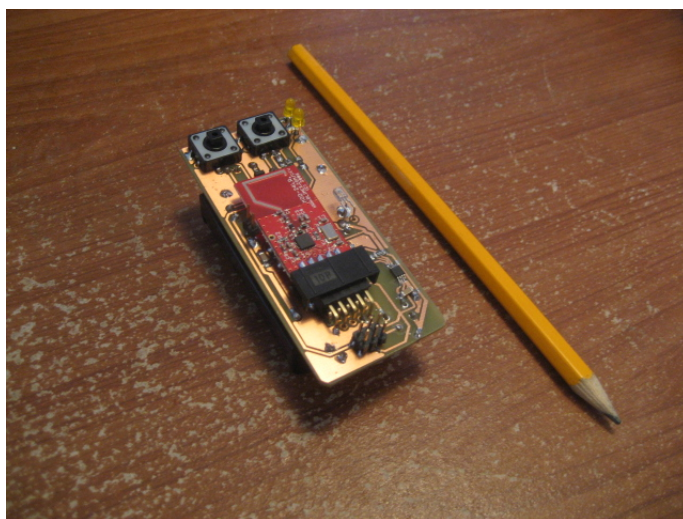


Seuraavaksi valmistetun solmu 3:sen, joka on esitettynä kuvassa 5.9, layout on suunniteltu samalla samalla periaatteella kuin solmu 2:sen. Ainoana erona solmu 2:seen on se, että mikrofoni-kytkennän sijasta solmulla on lämpösensori. Koska lämpösensori sisältää sekä digitaalista että analogista osuutta, se on sijoitettu piirilevyn digitaalisen ja analogisen alueen rajalle. Piirilevyn on esitettynä liitteenä sivulla 85.



*Kuva 5.9: Kannettava solmu 3.*

Viimeiseksi valmistettu kannettava solmu 4 sisältää samat sensorit kuin solmu 3 eli lämpösensorin ja valosensorin. Solmun piirilevyn suunnittelussa oli tavoitteena valmistaa fyysiseltä kooltaan mahdollisimman pieni piirilevy. Kuvassa 5.10 on esitettynä kyseinen kannettava solmu.



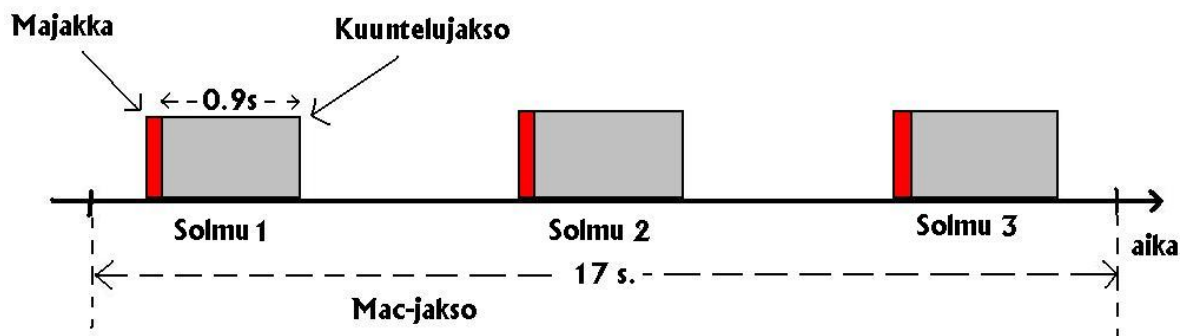
*Kuva 5.10: Kannettava solmu 4.*

## 5.3 Ohjelmisto

### 5.3.1 Yleiskatsaus reititysprotokollaan

Tärkeä osa reititysprotokollan suunnittelussa sensoriverkossa on tehonkulutuksen minimointi. Solmut toimivat usein pattereilla, joten energiaa on käytössä rajallisesti. Radio kuluttaa helposti kaikkein eniten tehoa. Tämän takia radiota ei kannata pitää jatkuvasti päällä, vaan radio laitetaan pois päältä, kun sitä ei tarvitse. Mutta koska solmun radio ei ole jatkuvasti päällä, muiden solmujen on jotenkin tiedettävä, koska naapurisolmun radio on päällä. Kehittämässäni protokollassa tämä hoidetaan *majakoiden* avulla.

Kuvassa 5.11 on esitettyä sensoriverkon MAC-protokollan ajoituskaavio. Yksi MAC-jakso on noin 17 sekunnin pituinen. Jokainen verkossa oleva solmu lähettää majakan kerran jaksossa. Kuten aiemmin todettu, tämän majakan tarkoituksena on kertoa muille solmuille solmun olemassaolosta. Majakan lähettämisen jälkeen solmu siirtyy kuuntelutilaan, jolloin muut solmut voivat lähettää sille dataa. Muutoin solmun radio on pois päältä energian säästämiseksi. Solmu lähettää majakan tasaisin väliajoin aina 17 sekunnin välein. Kun solmu haluaa tietää, mitä muita solmuja on lähistöllä ja onko niillä reittiä niulusolmulle, solmu kuuntelee koko 17 sekunnin jakson ajan etsien majakoita. Tätä nimitetään jatkossa *kokojaksokuunteluksi*.



**Kuva 5.11:** Ajoituskaavio sensoriverkon MAC-protokollasta.

Majakan lähettämisen jälkeen solmu pysyy kuuntelutilassa noin 0,9 sekuntia. Tällöin solmun radion pulssisuhteeksi (engl. duty cycle) tulee

$$\text{Pulssisuhte} = \frac{0,9s}{17s} \times 100\% = 5.3\%. \quad (5.2)$$

. Joskin tämä luku ei ota huomioon solmun itse suorittamia data reitityksiä muille solmuille eikä kokojaksokuunteluita. Normaalisti kokojaksokuuntelu suoritetaan vain kerran solmun käynnistettyä. Mutta jos solmu menettää reitin niulusolmulle, se suorittaa uudestaan kokojaksokuuntelun. Riippuen, kuinka hyvät tai huonot radiolinkit solmujen välillä on, kokojaksokuunteluita saatetaan suorittaa huomattavasti



enemmänkin. Tällöin radion pulssisuhde voi nousta jopa 10 prosenttiin. Tarkemmin protokollan toiminnasta esimerkiksi linkin menettämisessä käsitellään kappaleessa 5.3.4.

Alunperin oli tarkoituksena, että kun solmut olivat valinneet oman majakan ja kuuntelujakson ajoituksen, ne eivät tunkeutuisivat muiden solmujen kuuntelujaksolle. Tiedossa kuitenkin oli, että prosessoreiden RC-oskillaattoreiden ajanpitokyky ei ole paras mahdollinen, jolloin ajoitukset muuttuvat vähitellen. Tämä johtaa loppujen lopuksi siihen, että solmujen kuuntelujaksot tapahtuvat samaan aikaan. Tällöin tarkoituksena oli, että se solmu, joka kuulee naapurin lähettämän majakan omalla kuuntelujaksollaan, päättelee kuuntelujaksojen olevan päällekkäin, ja etsii uuden ajankohdan MAC-jaksossa lähettää oma majakkansa. Huonona puolena tässä on se, että solmu joutuu suorittamaan kokojaksokuuntelun, joka vie energiaa. Lisäksi kaikki solmun muodostamat linkit naapureidensa kanssa katkeaa, kun solmu asettaa uuden ajoituksen majakallensa. Seurauksena näiltä solmuilta ei saada sensoridataa, kunnes uusi reitti niulusolmulle on muodostettu. Alunperin käsitys oli, että RC-oskillaattorit olisivat kuitenkin sen verran tarkkoja, että tätä ei tapahtuisi liian usein. Kuitenkin testatessa tätä taktiikkaa huomattiin, että prosessoreiden kellot käyvät huomattavan eri tahtia, jolloin solmujen kuuntelujaksot ajautuivat päällekkäin melko nopeasti. Tästä seurauksena oli solmujen miltei jatkuva kokojaksokuunteluiden tekeminen, joka puolestaan johti siihen, että solmuilta ei saanut sensoridataa tarpeeksi usein. Tästä syystä päätin sallia solmujen kuuntelujaksojen menevän päällekkäin. Nyt, kun solmu vastaanottaa naapurin majakan omalla kuuntelujaksollaan, solmu jatkaa toimintaansa normaalisti eikä reagoi majakkaan mitenkään. Tämä muutos auttoi merkittävästi sensoridatan reitittämistä, eikä kuuntelujaksojen hetkittäiset päällekkäisyydet tuoneet mitään huomattavia ongelmia.

Toisin kuin kannettavat solmut, niulusolmu eli päämoduuli toimii verkkovirralla. Joten virrankulutus ei ole niulusolmussa niin suuri ongelma kuin kannettavissa solmuissa. Tästä syystä niulusolmun radio on jatkuvasti päällä valmiina vastaanottamaan sensoridataa kannettavilta solmuilta. Tästä huolimatta myös niulusolmu lähettää majakoita kerran MAC-jaksossa ilmoittamaan muille lähistöllä oleville solmuille olemassaolostaan. Niulusolmun ohjelmistoa käsitellään tarkemmin kappaleessa 5.3.5.

Kaikki viestintä tapahtuu samalla taajuudella (2,4 GHz), mutta viestintään käytetään eri loogisia kanavia eli *dataputkia*. Tästä puhutaan enemmän kappaleessa 5.3.2.

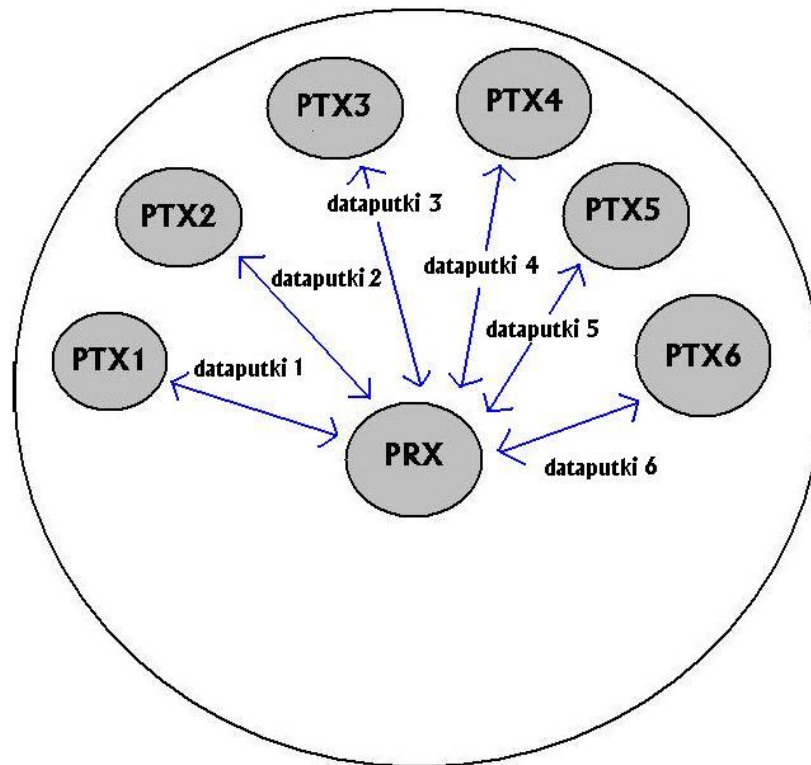
Solmut tallentavat *naapurilistaan* tietoja naapureistaan, jotka solmu etsii kokojaksokuuntelussa. Naapurilistassa jokaiselle naapurille tallennetaan naapurin tunnistetiedot, etäisyys niulusolmulle, majakan ajoitus MAC-jaksossa, käytetty dataputki (jos linkki on muodostettu) ja naapurin linkin tilanne. Naapurin linkki voi olla neljässä eri tilassa: “LINK\_NOT\_TRIED”, “LINK\_ESTABLISHED”, “LINK\_LOST” tai

“LINK\_INVALID”. “LINK\_NOT\_TRIED” tarkoittaa että naapurin majakka on havaittu, sillä on vapaita dataputkeja tarjolla, mutta yhteyttä ei ole kokeiltu muodostaa. “LINK\_ESTABLISHED” tarkoittaa, että linkki naapurin kanssa on muodostettu. “LINK\_LOST” tarkoittaa, että linkki naapuriin on menetetty tai että naapurilla ei ollut alkujaankaan vapaita dataputkeja tarjottavana. “LINK\_INVALID” tarkoittaa, että kyseinen naapuritieto listassa ei ole laillinen. Naapurilistassa muuttujalla “nextHopNode” viitataan siihen naapuriin, jolle solmu reitittää tai pyrkii muodostamaan linkin. Enemmän näitä tiloja ja dataputkien käyttöä käsitellään kappaleissa 5.3.2 ja 5.3.4.

### 5.3.2 Radion ominaisuuksista

Käytetty radiomoduuli nRF24L01P sisältää mahdollisuuden käyttää automaattisesti lähetettyjä kuittauspaketteja. Näiden avulla lähettyvä radio voi varmistaa, että paketti on päässyt perille. Radiomoduulissa voi myös käyttää automaattisia uudelleenlähetyksiä, mikäli kuittauspakettia ei vastaanotettu. Näitä ominaisuuksia käytetään sensoriverkon protokollassa sensoridatojen lähetyksessä. Mutta toisaalta majakoiden lähettämiseen ei käytetä kuittauspaketteja, koska majakka ei ole suunnattu kenellekkään tietylle solmulle.

Jotta kuittauspaketteja voidaan käyttää, on kahden solmun välille ensin muodostettava oma dataputki. Yksi dataputki on oma looginen kanavansa yhdellä radiotaajuudella. Jotta vastaanottava solmu osaa lähettää kuittauspaketin vain yhdelle solmulle, eli paketin lähettäneelle solmulle, on solmuilla oltava oma dataputki. Kuva 5.12 esittää tämän tilanteen. Yhdessä radiossa voi olla enintään kuusi eri dataputkea. Eli yksi radio voi vastaanottaa paketteja korkeintaan kuudelta eri radiolta käyttäen kuittauksia. Tässä järjestelmässä radiolla ei ole uniikkia osoitetta, vain korkeintaan jopa kuusi eri osoitetta.



**Kuva 5.12:** Dataputket nRF24L01P-radiomoduulissa. PTX tarkoittaa lähettävää radiota, PRX tarkoittaa vastaanottavaa radiota.

Sensoriverkon protokollassa yksi näistä solmun radion dataputkista on aina varattu julkiselle kanavalle. Julkiselle kanavalle lähetetyt viestit tulee vastaanotetuksi kaikissa solmuissa, kunhan solmun radio on päällä. Julkisella kanavalla lähetetään muun muassa majakat.

Kolmea radion dataputkista käytetään sensoridatojen reitittämiseen solmujen kautta. Näissä lähetyksissä käytetään kuittauspaketteja ja automaattisia uudelleenlähetyksiä. Mutta ennen kun voidaan lähettää sensoridataa kahden solmun välillä kuittauksien kanssa, on solmujen sovittava käyttää samaa dataputkea. Kaikki solmut ilmoittavat majakoissaan vapaana olevien dataputkien numerot eli tunnisteet. Kun solmu haluaa muodostaa linkin naapurinsa kanssa, se valitsee naapurin majakasta yhden dataputken numeron ja lähettää varauspyynnön (DATAPIPE\_REQUEST) dataputkea varten julkisella kanavalla. Naapurisolmu vastaa tähän pyyntöön lähettämällä viestin takaisin (DATAPIPE\_GRANT), jossa sallitaan dataputken käyttö. Molemmat näistä viesteistä lähetetään ilman kuittauksia tai uudelleenlähetyksiä julkisella kanavalla. Tämän jälkeen solmu alkaa reitittämään naapurillensa sensoridataa tällä dataputkella käyttäen kuittauksia ja uudelleenlähetyksiä. Enemmän sensoriverkon protokollan käyttämistä viesteistä käsitellään kappaleessa 5.3.3 ja itse protokollaa käsitellään enemmän kappaleessa 5.3.4.

### 5.3.3 Protokollassa käytetyt viestit

Taulukossa 5.3 on esitettynä kaikki protokollassa käytetyt viestit. Jokaisen viestityypin ensimmäisenä tavuna on viestin tyyppin ilmoittava tunniste. Tunnisteen avulla solmut voivat päätellä miten tulkita sen jälkeen tulevat tavut.

Ensimmäisen viestityypin, eli majakan, tunniste on “BEACON\_ID” ja viesti koostuu yhteensä kolmesta kuuteen tavua dataa. Majakkaviesti sisältää tunnisteen lisäksi majakan lähettäneen solmun tunnisteen, etäisyyden niulusolmulle ja vapaiden dataputkien numerot. Etäisyys niulusolmulle lasketaan ottamalla huomioon hyppyjen määrä niulusolmulle sekä jäljellä oleva patterin jännite.

“DATAPIPE\_REQUEST”-viestillä solmu pyytää naapurilta omaa dataputkea solmujen väliseen kommunikointiin. Viesti sisältää sekä pyynnön lähettävän solmun tunnisteen ja pyynnön kohteena olevan tunnisteen. Tämän lisäksi viesti sisältää dataputken numeron, jota solmu pyytää itselle.

“DATAPIPE\_GRANT”-viestillä myönnetään linkkiä pyytävälle solmulle oma dataputki. Viesti sisältää viestin tunnuksen lisäksi viestin lähettäjän ja vastaanottajan tunnisteet sekä myönnetyn dataputken numeron.

Varsinaisia sensoridatoja varten on omat tunnuksensa. Lämpötiladata lähetetään viestitunnuksella “TEMP\_CODE” ja valoisuusdata tunnuksella “LUX\_CODE”. Toinen tavu viestissä kertoo viestin lähettäjän tunnisteen. Kolmas ja neljäs tavu sisältävät varsinaisen sensoridatan. Viimeinen eli viides tavu kertoo sen solmun tunnisteen, josta sensoridata on alunperin lähtenyt. Eli kun sensoridata etenee sensoriverkossa, toinen tavu viestissä muuttuu jokaisen lähetyksen jälkeen, mutta viides tavu pysyy aina samana niulusolmulle asti.

“BAT\_CODE”-viestillä solmu kertoo nykyisen käyttöjännitteensä tilanteen päämoduulille. BAT\_CODE-viestiä ei lähetetä jokaisella MAC-jaksolla toisin kuin sensoridatat. Patterin jännite ei muutu nopeasti, joten harvempi mittausväli riittää. Lisäksi energiankulutuksen kannalta kannattaa lähetettyjen viestien määrä pitää mahdollisimman pienenä. Tämän takia BAT\_CODE-viesti lähetetään vain joka kolmannella MAC-jaksolla.

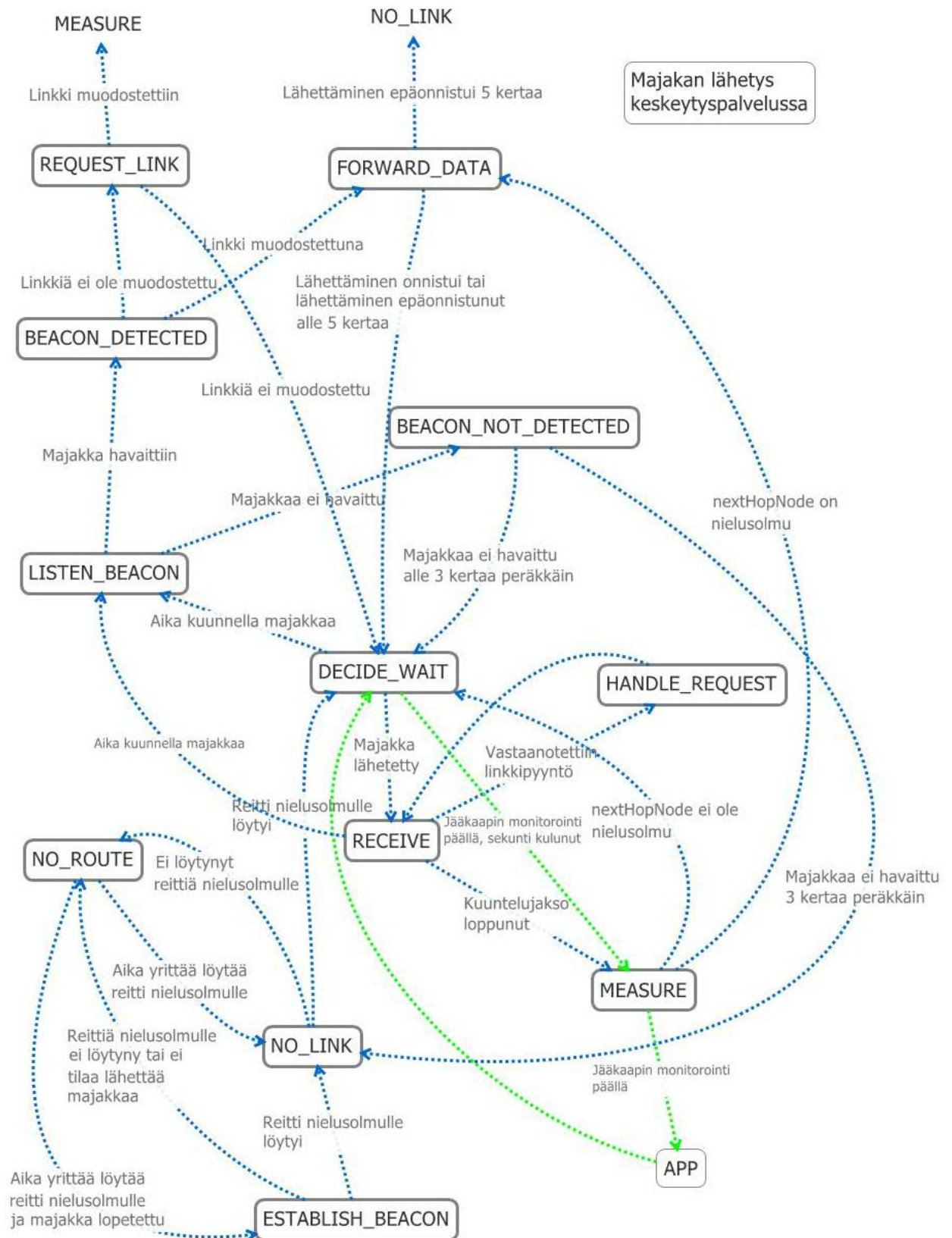
Lopuksi solmut voivat lähettää erilaisia varoitus- ja hälytysviestejä. Tunnuksella “FRIDGE\_ALARM\_OPEN” kulkevalla viestillä solmu lähettää päämoduulille varoituksen jääkaapin aukiolosta. Viestin loput tavut ovat aivan samat kuin “LUX\_DATA”-viestin, eli viesti sisältää valoisuusarvoja valosensorista. Jos solmu lähettää “FRIDGE\_ALARM\_OPEN”-paketin, se ei enää lähetä samalla MAC-jaksolla “LUX\_DATA”-pakettia turhaan. Tällä tavalla yhdistämällä kahden eri viestityypin sisällöt vähennetään lähetettävien pakettien määrää ja säästetään radion kuluttamaa energiaa. Samalla tavalla yhdistetään “SECURITY\_ALARM” ja “LUX\_DATA”-paketit. “SECURITY\_ALARM”-paketti kertoo, että solmu on mikrofonin kautta kuullut ääniä asunnossa.

**Taulukko 5.3:** Sensoriverkon protokollassa käytetyt viestit.

Viestityyppi(1.tavu)	2.tavu	3.tavu	4.tavu	5.tavu	6.tavu
BEACON_ID(Majakka)	solmun tunniste	etäisyys	1.dataputken numero	2.dataputken numero	3.dataputken numero
DATAPIPE_REQUEST	naapurin tunniste	solmun tunniste	dataputken numero	-	-
DATAPIPE_GRANT	solmun tunniste	naapurin tunniste	dataputken numero	-	-
TEMP_CODE (lämpötiladata)	solmun tunniste	1.datatavu	2.datatavu	lähteen tunniste	-
LUX_CODE (valoisuusdata)	solmun tunniste	1.datatavu	2.datatavu	lähteen tunniste	-
BAT_CODE (jännitedata)	solmun tunniste	1.datatavu	2.datatavu	lähteen tunniste	-
SECURITY_ALARM	solmun tunniste	1.datatavu	2.datatavu	lähteen tunniste	-
FRIDGE_ALARM_OPEN	solmun tunniste	1.datatavu	2.datatavu	lähteen tunniste	-

### 5.3.4 Kannettavien solmujen ohjelmisto

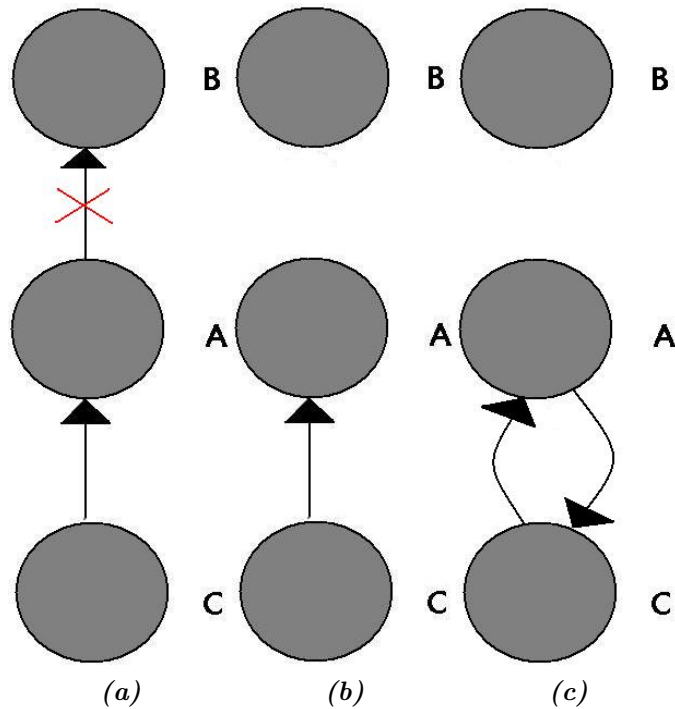
Kannettavien solmujen pääohjelma on päätetty toteuttaa tilakoneen muodossa. Kuvassa 5.13 on esitettyä tämä tilakone. Solmun käynnistettyä se alkaa toimintansa tilasta ESTABLISH\_BEACON. Tässä tilassa solmu ensimmäiseksi kuuntelee koko MAC-jakson ajan etsien mahdollisia naapurisolmuja. Aina kun solmu vastaanottaa naapurin majakan, se ottaa majakasta talteen naapurisolmun tunnisteen, etäisyyden niulusolmulle sekä majakan ajoituksen MAC-jaksossa. Mikäli naapurilla on vapaita dataputkeja majakan mukaan, solmu asettaa naapurin tilaksi LINK\_NOT\_TRIED naapurilistaan. Jos naapurilla ei ollut yhtään vapaata dataputkea, se asettaa naapurin tilaksi LINK\_LOST. Tällä tavalla solmu ei turhaan yritä muodostaa linkkiä naapurille. Tämän jälkeen, kun kaikkien naapureiden ajoitukset on saatu selville, solmu laskee oman majakkansa lähetysajan. Solmu asettaa majakan lähetyksen mahdollisimman etääksi ajallisesti muista majakoista. Lopuksi solmu siirtyy tilaan NO\_LINK.



**Kuva 5.13:** Kannettavien solmujen ohjelmiston tilakone. Vihreällä viivalla on esitetty solmukohtainen koodi.

NO\_LINK-tilassa solmu asettaa ensin omaksi etäisyydekseen niulusolmulle mak-

simietäisyyden, joka tarkoittaa, että solmulla ei ole reittiä niulusolmulle. Jos tilaan tultiin muualta kuin ESTABLISH\_BEACON-tilasta, asetetaan naapurilistassa next-HopNoden tilaksi LINK\_LOST. Alunperin tässä tilassa haettiin naapurilistasta seuraava paras reitti niulusolmulle, jos aikaisempi linkki oli katkennut. Mutta, koska prosessoreiden kellot käyvät niin eri tahtiin, naapurilistassa olevat ajoitukset muille solmuille olivat jo melkein aina vanhentuneet. Tällöin solmu yritti aina turhaan kuunnella majakkaa naapurilistan ajoituksen mukaan. Tästä syystä NO\_LINK-tilaa muutettiin siten, että solmu suorittaa heti uuden kokojaksokuuntelun, jos linkki on katkennut. Toisaalta, jos NO\_LINK-tilaan tullaan suoraan ESTABLISH\_BEACON-tilasta, voidaan naapurilistan ajoituksia käyttää. Tällöin naapurilistasta etsitään se solmu, jonka tila on LINK\_NOT\_TRIED ja jonka etäisyys niulusolmulle on kaikkein pienin. Naapurua valittaessa otetaan huomioon myös se, onko kyseinen naapuri lähettänyt viime aikoina paketteja solmulle. Alunperin oli nimittäin koodissa ongelma, jos esimerkiksi solmu A menetti yhteyden valitsemalleen naapurisolmulle B. Kuva 5.14 näyttää tämän ongelman. Solmulla A oli ennen yhteyden menettämistä myös toinen naapuri C, joka reititti datansa solmulle A. Jos nyt yhteyden menetettyään solmu A alkaa etsimään uudestaan naapureita, joilla olisi reitti niulusolmulle, solmu yhä löytää naapurin C majakan. Tämä majakka kertoo, että solmulla C olisi reitti niulusolmulle, koska se ei ole vielä ehtinyt päivittää tilaansa heijastamaan katkennutta A-B linkkiä. Näin A alkaakin reitittää solmulle C luullen, että sillä on reitti niulusolmulle, ja samalla solmu C reitittää yhä solmulle A luullen, että sillä on reitti niulusolmulle. Näin syntyy silmukka, jossa solmut A ja C reitittävät dataa turhaan toisilleen kuluttaen energiaa ilman, että data pääsee koskaan niulusolmulle. Tämän takia solmu A aina tarkistaa reitittääkö kyseinen naapurisolmu itselle dataa, ennen kuin se alkaa reitittämään tälle naapurille.



**Kuva 5.14:** Silmukan muodostuminen sensoriverkossa. Solmu A menettää linkin solmulle B (a), solmu A etsii uutta reittiä ja löytää solmun C (b), solmu A alkaa reitittämään solmulle C ja silmukka muodostunut (c).

Jos naapurilistasta löytyi mahdollinen reitti niulusolmulle, solmu siirtyy tilaan `DECIDE_WAIT`. Mikäli yhdelläkään naapurilla ei ole reittiä niulusolmulle tai yhtään naapuria ei löytynyt, solmu siirtyy tilaan `NO_ROUTE`. Käsitellään ensimmäiseksi tilanne, jossa reitti niulusolmulle on löytynyt.

Solmu viettää suurimman osan ajastaan tilassa `DECIDE_WAIT`. Tässä tilassa radio on pois päältä energian säästämiseksi ja prosessori on idle-tehonsäästötilassa, jossa kaikki ajastimet ovat käynnissä, mutta prosessori itse on pysähtyneenä. Prosessorilla olisi vieläkin vähemmän tehoa vieviä tiloja, mutta niissä ajastimet ovat pysähtyneet, joten niitä ei voida käyttää. Tilasta `DECIDE_WAIT` siirrytään joko tilaan `RECEIVE` tai tilaan `LISTEN_BEACON` riippuen, onko oma majakka lähetetty, vai onko aika kuunnella naapurin majakkaa. Jos ajastinkeskeytys tapahtuu majakan lähettämisen merkiksi, solmu lähettää majakan itse keskeytyspalvelufunktiossa. Yleensä ei ole suositeltavaa suorittaa niin monimutkaisia toimintoja kuin majakan lähetys keskeytyspalvelufunktiossa, mutta majakan lähettäminen on aikakriittistä, joten se tehdään heti keskeytyspalvelufunktiossa. Majakan lähettämisen jälkeen solmu siirtyy `RECEIVE`-tilaan. Sen sijaan, jos on aika kuunnella naapurisolmun majakkaa, solmu herää hieman ennen, kuin naapurisolmun on aika lähettää majakkansa naapurilistan mukaan. Herättyään tähän ajastinkeskeytykseen solmu siirtyy tilaan `LISTEN_BEACON`.

Tilassa `LISTEN_BEACON` solmu laittaa radion päälle kuuntelutilaan. Solmu



kuuntelee majakkaa hetken aikaa. Jos solmu havaitsee radiolähetysten tänä aikana, lähetysten pakettitunnus tarkistetaan. Jos paketin tunnus on muu kuin "BEACONID", pakettia ei käsitellä sen enempää. Jos paketin tunnus oli "BEACONID", tarkistetaan vielä lähettäjän tunniste majakasta. Jos tunniste oli odotetun solmun tunniste, päivitetään majakasta naapurilistaan tieto naapurin etäisyydestä niulusolmulle. Samalla päivitetään naapurilistaan majakan ajoitus. Alunperin majakan ajoitusta ei päivitetty aina majakan havaitsemisen jälkeen, vaan majakan ajoitus otettiin vain talteen, kun suoritettiin kokojaksokuuntelu. Tämä osoittautui kuitenkin huonoksi ratkaisuksi, koska solmujen kellot käyvät niin eri tahtia, että solmut joutuvat miltei heti epätahtiin. Joten ottamalla joka kerta uusi ajoitus talteen majakalle, solmut pysyvät tahdissa paremmin. Majakan havaittua solmu siirtyy tilaan BEACON\_DETECTED. Ellei majakkaa havaittukaan, solmu siirtyy tilaan BEACON\_NOT\_DETECTED.

Riippuen naapurisolmun ja sen linkin tilasta, solmu etenee BEACON\_DETECTED-tilasta eri tavoin. Mikäli linkki solmun ja naapurisolmun välillä on jo muodostettu, eli naapurilistassa naapurin tila on "LINK\_ESTABLISHED", solmu siirtyy tilaan FORWARD\_DATA lähettämään dataa kahta poikkeusta lukuunottamatta. Ensimmäinen poikkeus on se, jos naapurisolmu on menettänyt reitin niulusolmulle. Tämä huomataan majakan antamasta naapurin etäisyydestä niulusolmulle. Tällöin mennään takaisin tilaan DECIDE\_WAIT odottamaan seuraavaa tapahtumaa. Mikäli solmu on kuunnellut majakan ja todennut naapurin olevan ilman reittiä kolme kertaa peräkkäin, solmu muuttaa naapurin tilaksi "LINK\_LOST" naapurilistassa ja siirtyy tilaan NO\_LINK etsimään uutta reittiä. Toinen poikkeus on se, jos majakan mukaan naapuri on jo vapauttanut solmun käyttämän dataputken. Tällöin solmu asettaa naapurin tilaksi LINK\_NOT\_TRIED naapurilistaan ja siirtyy itse tilaan REQUEST\_LINK. Jos alunperin BEACON\_DETECTED-tilaan tultaessa naapurin tila oli LINK\_NOT\_TRIED ja majakan mukaan naapurilla on vapaita dataputkia, solmu siirtyy myöskin REQUEST\_LINK-tilaan. Jos taas naapurilla ei ollut vapaita dataputkeja, solmu asettaa naapurin tilaksi LINK\_LOST ja menee takaisin tilaan NO\_LINK. Muutoin, jos linkki on muodostettu naapurille, mennään lähettämään dataa tilaan FORWARD\_DATA. Poikkeuksen tässä tekee se, jos naapuri onkin päämoduuli. Koska päämoduulin radio on aina päällä, datan lähetys ei ole rajoittunut vain heti majakan vastaanottamisen jälkeen. Tämän takia päämoduulille datan lähettäminen jätetäänkin myöhemmäs(tilan MEASURE jälkeen) ja vain linkin muodostamiset tehdään heti majakan jälkeen.

Tilaan BEACON\_NOT\_DETECTED tullaan, kun solmu on yrittänyt kuunnella naapurin majakkaa, mutta ei onnistunut siitä. Tämä voi johtua monesta syystä. Naapuri on voinut liikkua kuuluvuusalueen ulkopuolelle, naapurin patterit ovat saattaneet loppua, ilmassa voi olla häiriöitä, jotka haittaavat radiolähetystä tai majakan lähettämisen ajoitus on voinut muuttua. Jos tähän tilaan tullaan tarpeeksi monta

kertaa peräkkäin, solmu päättää, että naapuriin tuskin saadaan enää yhteyttä. Naapurin tilaksi muutetaan tällöin LINK\_LOST ja solmu siirtyy tilaan NO\_LINK. Muutoin BEACON\_NOT\_DETECTED-tilasta mennään tilaan DECIDE\_WAIT.

REQUEST\_LINK-tilassa valitulta naapurisolmulta pyydetään linkkiä eli omaa dataputkea radiossa. Solmu valitsee naapurin majakassa ilmoitetuista vapaista dataputkien numeroista satunnaisesti yhden ja lähettää naapurille pyynnön (DATAPIPE\_REQUEST-paketti), johon sisällytetään tämä dataputken numero. Tämä pyyntö lähetetään julkisella dataputkella ilman kuittauksia. Pyyntö lähetämisen jälkeen solmu laittaa radion kuuntelutilaan ja odottaa vastausta naapurisolmulta. Jos naapurisolmu vastaa pyyntöön DATAPIPE\_GRANT-paketilla, solmu asettaa naapurilistaan naapurin tilaksi LINK\_ESTABLISHED ja tallettaa naapurilistaan myös käytetyn dataputken numeron. Solmu laskee myös päivitetyn etäisyytensä niulusolmulle perustuen naapurisolmun etäisyyteen niulusolmulle (joka oli annettu majakassa) ja jäljellä olevaan patterijännitteeseen. Lopuksi solmu siirtyy tilaan MEASURE. Jos naapurisolmu ei vastannut pyyntöpakettiin tietyn ajan sisällä, solmu siirtyy takaisin tilaan DECIDE\_WAIT.

FORWARD\_DATA-tilaan tullaan joko naapurisolmun majakan havaitsemisen jälkeen BEACON\_DETECTED -tilasta, tai jos naapurisolmu on niulusolmu, tilaan tullaan MEASURE-tilasta. Tilan tarkoituksena on reitittää eteenpäin sensoridatoja. Tilan FORWARD\_DATA alussa radio laitetaan lähetystilaan. Lähetystä ei kuitenkaan aloiteta heti, vaan solmu odottaa satunnaisen ajan ennen lähetystä. Tämä tehdään sen takia, että monet muutkin solmut voivat haluta lähettää samalle solmulle ja lähetykset voivat törmätä, jos solmut alkavat lähettämään heti majakan havaitsemisen jälkeen. Odotusaika on kuitenkin maksimissaan noin 0.8 sekuntia. Odotusajan jälkeen solmu laittaa omat sensoridatat lähetyspakettiin ja lisää samaan pakettiin FIFO:sta myös muiden solmujen datoja, jos niitä on. Alunperin koodissa jokainen eri sensoridata lähetettiin omassa paketissaan, mutta tämä ei ollut hyvä ratkaisu paristakin syystä. Ensinnäkin radion vastaanottopuskurissa on tilaa vain kolmelle eri paketille, joten vastaanottavan solmun radion puskuri täytyy helposti, jos jokainen eri sensoridata lähetetään omassa paketissa. Radion vastaanottopuskurin yhteen pakettiin mahtuu enimmäkseen 32 tavua, joten yhteen pakettiin mahtuu useampikin sensoridata. Toiseksi useamman pienemmän paketin lähettäminen kuluttaa enemmän energiaa kuin yhden isomman paketin. Tämä sen takia, että jokaisen lähetyksen alussa radio lähettää alukkeen radioiden synkronisaatiota varten. Alukkeen lähettäminen kuluttaa energiaa, joten sensoridatojen yhdistäminen yhteen pakettiin säästää energiaa. Toisin kuin majakkojen ja linkkipyyntöjen lähettämisessä, datan lähettämisessä käytetään kuittauspaketteja ja automaattisia uudelleenlähetyksiä. Maksimissaan näitä automaattisia uudelleenlähetyksiä on kaksi ennen kuin radio lopettaa yrittämisen. Tämän lisäksi solmu itse pitää myös kirjaa epäonnistuneista lähetyksistä. Mikäli

viisi kertaa peräkkäin lähetys on epäonnistunut, solmu asettaa naapurilistaan naapurin tilaksi LINK\_LOST ja siirtyy itse NO\_LINK -tilaan. Muussa tapauksessa FORWARD\_DATA-tilasta siirrytään DECIDE\_WAIT tilaan.

HANDLE\_REQUEST-tilassa käsitellään naapurisolmujen lähettämiä linkin muodostamiseen tarkoitettuja pyyntöpaketteja eli DATAPIPE\_REQUEST-paketteja. Tähän tilaan tullaan RECEIVE-tilasta, solmun vastaanotettua DATAPIPE\_REQUEST-paketin. Tilan alussa radio laitetaan lähetystilaan. Tämän jälkeen solmu tarkistaa mikä solmu on pyytämässä dataputkea ja mikä pyydetyn dataputken numero on. Mikäli kysytty dataputki oli vapaana, merkitsee solmu sen nyt varatuksi ja lähettää naapurisolmulle DATAPIPE\_GRANT-paketin, joka sisältää kysytyn dataputken numeron, naapurisolmun tunnisteen ja solmun oman tunnisteen. Solmu samalla merkitsee myös, kuka naapurisolmuista on varannut kyseisen dataputken. Tämä paketti lähetetään julkisella kanavalla ilman kuittauksia. Mikäli naapurisolmulla oli jo aikaisemmin varattuna dataputki, solmu vapauttaa tämän dataputken, jottei samalle naapurisolmulle ole varattuna kahta dataputkea samaan aikaan. Koska dataputkeja on rajallinen määrä, on tärkeää vapauttaa dataputket, joille ei ole tullut lähetyksiä vähään aikaan. Tämän takia jokaiseen dataputkeen liittyy laskuri, joka laskee kuinka monta MAC-jaksoa on kulunut ilman, että on tullut yhtäkään pakettia dataputkelle. Mikäli on kulunut viisi MAC-jaksoa, ilman yhtäkään saapunutta pakettia dataputkelle, solmu merkitsee dataputken vapaaksi. Tämä tehdään RECEIVE-tilassa. HANDLE\_REQUEST-tilan lopussa radio laitetaan takaisin kuuntelutilaan ja solmu menee takaisin RECEIVE-tilaan.

RECEIVE-tilaan tullaan DECIDE\_WAIT-tilasta majakan lähetyksen jälkeen tai HANDLE\_REQUEST-tilan jälkeen. Tilan tarkoituksena on vastaanottaa mahdollisten naapurien paketteja. Tilan alussa radio laitetaan kuuntelutilaan. Kuuntelutilassa pysytään noin 0,9 sekuntia. Tänä aikana, jos solmu vastaanottaa paketteja, solmu ensin tarkistaa dataan liittynä paketin tunnuksen. Jos kyseessä on sensoridata, eli paketin tunnus on HUM\_CODE, TEMP\_CODE tai LUX\_CODE, solmu laittaa paketin FIFO:on odottamaan lähetystä. Paketissa oleva lähettäjän tunniste korvataan solmun omalla tunnisteella. Samalla solmu nolaa käytetyn dataputken laskurin. Mikäli paketin tunnus oli DATAPIPE\_REQUEST, solmu siirtyy HANDLE\_REQUEST-tilaan. Kuunteluajan umpeuduttua solmu lisää kaikkien käytössä olevien dataputkien laskuria yhdellä. Mikäli laskuri saavuttaa maksimiarvon, eli dataputki on ollut käyttämättömänä liian kauan, solmu merkitsee kyseisen dataputken vapaaksi. Jos solmulla on reitti niulusolmulle, solmu siirtyy tilaan MEASURE. Muutoin solmu siirtyy tilaan DECIDE\_WAIT.

Alunperin reititysprotokollassa kahden eri solmun kuuntelujaksot eivät voineet olla samaan aikaan. Mutta ajoitusongelmien takia tästä jouduttiin luopumaan. Nyt kahden solmun kuuntelujaksot voivat olla samaan aikaan. Tämä voi aiheuttaa ongelmia jos esimerkiksi solmu A yrittää reitittää solmulle B, mutta molemmilla on

samat ajoitukset kuuntelujaksolle. Tällöin solmu A ei voinut reitittää solmulle B, koska solmu A oli myös kuuntelemassa. Itse asissa tilanne oli vieläkin pahempi kuin tämä. Testatessa ensimmäisen kerran korjattua reititysprotokollaa, solmut jäivät jumiin ohjelmassa päällekkäisien ajoitusten takia. Tämä korjattiin sillä, että nyt solmu A siirtyy RECEIVE-tilasta heti pois tilaan LISTEN\_BEACON, jos solmun B kuuntelujakso on osittain päällekkäin solmun A kuuntelujakson kanssa. Tämän korjauksen avulla solmujen ohjelmat eivät enää jääneet jumiin ja solmut pystyivät reitittämään toisillensa, vaikka kuuntelujaksot menevätkin päällekkäin.

MEASURE-tilassa luetaan solmulla olevien sensorien antamat lukemat. Ennen mittausta solmu tarkistaa, että käyttöjännite on tarpeeksi suuri sensoreiden käyttöön. Se tehdään mittaamalla käyttöjännite jännitejaon kautta ja käyttämällä vertailujännitteenä prosessorin muodostamaa sisäistä 1.1 voltin jännitettä. Mikäli käyttöjännite on alle 2.7 voltia, sensoreita ei käytetä. Analogisten signaalien mittausta varten prosessorin "IDLE"-tehonsäästötila muutetaan täksi ajaksi tilaan "ADC Noise Reduction", joka sammuttaa prosessorin AD-muunnoksen ajaksi, jotta prosessori ei aiheuttaisi häiriöitä mittaukseen. Jos solmun naapurisolmu on niulusolmu, solmu siirtyy tämän jälkeen tilaan FORWARD\_DATA lähettämään mittaustulokset heti niulusolmulle. Tämä voidaan tehdä sen takia, että niulusolmulla on aina radio päällä, joten ei tarvitse odottaa niulusolmun majakan vastaanottamista. Muutoin solmu siirtyy tilaan DECIDE\_WAIT odottamaan seuraavaa tapahtumaa.

NO\_ROUTE-tilaan tullaan joko NO\_LINK-tilasta tai ESTABLISH\_BEACON-tilasta. Tähän tilaan tullaan, kun solmu on kuunnellut koko MAC-jakson naapureiden majakoita eikä löytänyt yhtään reittiä niulusolmulle. Tilassa radio laitetaan pois päältä energian säästämiseksi. NO\_ROUTE-tilan tarkoituksena on pysyä tehonsäästötilassa ja vain aika ajoin herätä tarkistamaan, onko tullut reittiä niulusolmulle. Aluksi reittiä niulusolmulle etsitään noin 50 sekunnin välein menemällä NO\_LINK-tilaan. Jos reittiä ei vielä löydy, tätä väliä pidennetään vähitellen, kunnes lopulta reittiä niulusolmulle haetaan noin 15 minuutin välein. Kun solmu on kolme kertaa yrittänyt löytää reittiä niulusolmulle ja epäonnistunut, solmu lopettaa oman majakkansa lähettämisen. Näin säästetään energiaa, koska majakkaa ei kannata lähettää ellei reittiä niulusolmulle löydy. Jos majakan lopettamisen jälkeen solmu alkaa etsimään reittiä niulusolmulle, se tehdään menemällä tilaan ESTABLISH\_BEACON eikä menemällä tilaan NO\_LINK.

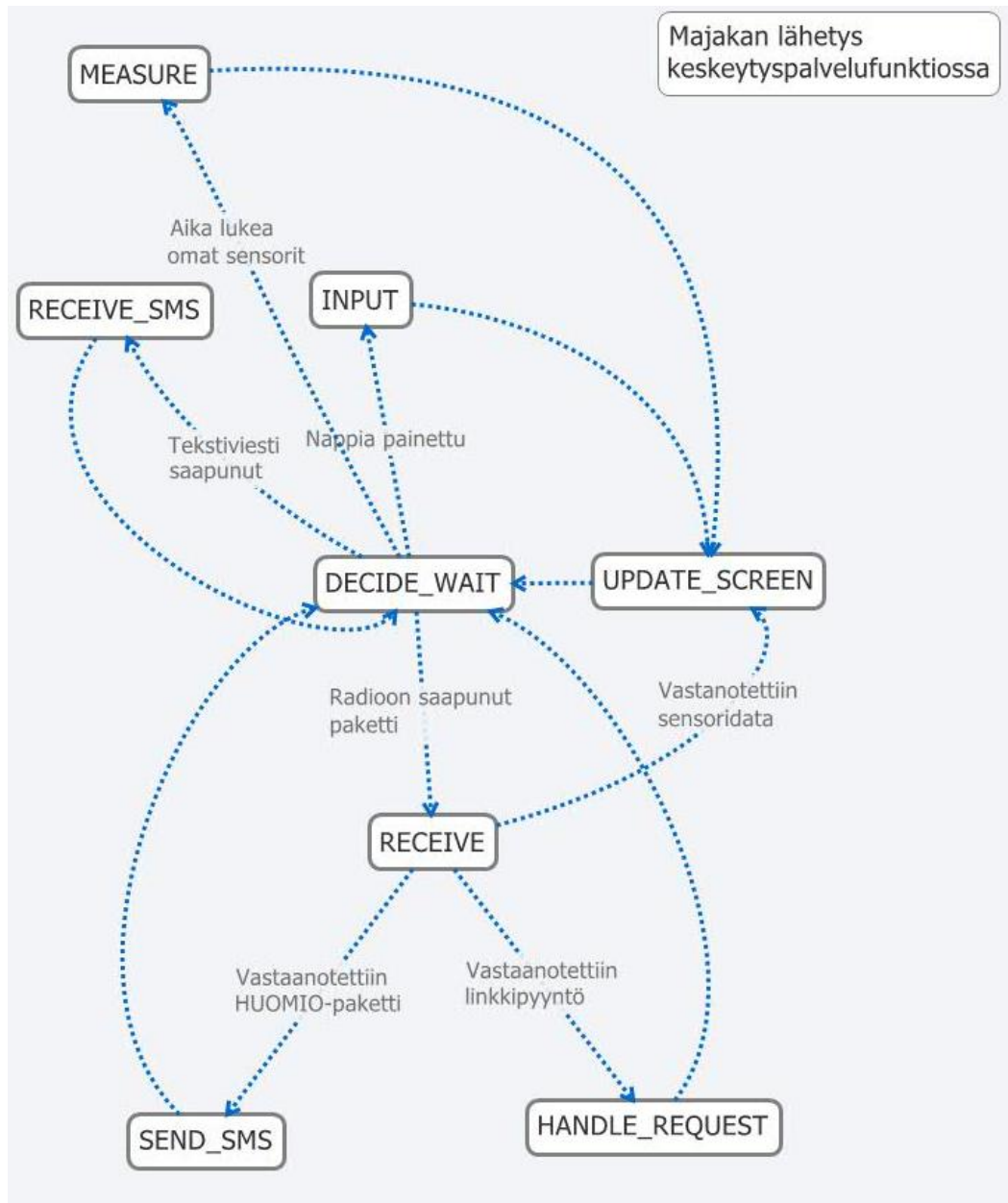
Yllä oleva ohjelmiston kuvaus koskee kaikkia kannettavia solmuja. Tämän lisäksi solmut sisältävät myös jonkin verran solmukohtaisia koodeja. Tavoitteena ohjelmistoa kirjoitettaessa oli, että solmujen ohjelmisto olisi mahdollisimman paljon samanlainen ja solmukohtaisia koodeja olisi mahdollisimman vähän. Kuitenkin jotkut suunnitelluista sovelluksista vaatii solmukohtaisia koodeja. Lisäksi solmut sisältävät eri sensoreita, joten niitä varten on tehtävä omat koodit. Esimerkiksi Solmu 3 sisältää koodia jääkaapin valvontaan. Nappia painamalla solmu alkaa mittaamaan

yhden sekunnin välein valosensorin arvoja. Eli yhden sekunnin välein solmu siirtyy DECIDE\_WAIT-tilasta MEASURE-tilaan käyttämään sensoreita. Tämän jälkeen solmu siirtyy tilaan APP. Tässä tilassa solmu laskee, kuinka kauan valosensori on mitannut valon olevan päällä. Mikäli 2 minuuttia peräkkäin kaikki valosensorin arvot kertovat valon olevan päällä, solmu lähettää seuraavan kerran päästessään FORWARD\_DATA-tilaan FRIDGE\_ALARM\_OPEN-paketin. Jos tämä koodi olisi laitettu päämoduuliin, valosensorin lukemia olisi voitu tarkastella vain 17 sekunnin välein. Tämä olisi ollut liian harva väli tarkastelemaan onko jääkaapin ovi auki.

Solmu 2 sisältää puolestaan solmukohtaista koodia murtohälytyssovellukseen. Solmun nappia painamalla käyttäjä voi ottaa sovelluksen käyttöön. Napin painamisen jälkeen solmu odottaa noin 2 minuuttia ennen, kuin se alkaa käyttämään mikrofonia havaitsemaan kuuluuko asunnossa mitään ääniä. Tämä viive on sitä varten, että käyttäjä ehtii itse poistua asunnosta ennen, kuin murtohälytys menee päälle. Viiveen laskenta tehdään DECIDE\_WAIT-tilassa käyttäen apuna MAC-jakson 17 sekunnin aikaväliä. Tämän kahden minuutin jälkeen solmun prosessori laittaa analogisen komparaattorin keskeytyksen päälle. Komparaattoria kannattaa käyttää AD-muuntajan sijasta monitoroimaan mikrofonikytkennän ulostuloa, koska jatkuva AD-muuntajan käyttö kuluttaisi enemmän energiaa. Jos mikrofoni kuulee ääniä, sen synnyttämä signaalin vaihtelu aiheuttaa komparaattorissa keskeytyksen. Tämän jälkeen vasta prosessori alkaa käyttämään AD-muuntajaa mittamaan mikrofonin vahvistimen ulostuloa. Jos AD-muuntaja havaitsee useampia mittauservoja tietyn arvon yläpuolella, solmu pääättelee, että asunnossa kuuluu ääniä. Seuraavan kerran, kun solmu menee FORWARD\_DATA-tilaan, se lähettää SECURITY\_ALARM -paketin kohti päämoduulia.

### 5.3.5 Päämoduulin ohjelmisto

Kuvassa 5.15 on esitettyinä päämoduulin eli sensoriverkossa niulusolmuna toimivan tilakone. Koska päämoduulin ei tarvitse huolehtia sensoridatan reitittämisestä eteenpäin, sen ohjelmisto hieman yksinkertaisempi kuin kannettavien solmujen. Päämoduuli vastaa data keräämisestä kannettavilta solmuilta ja esittää ne ja omat sensoridatat LCD-näytöllä.



**Kuva 5.15:** Päämoduulin ohjelmiston tilakone.

Majakan lähetys tapahtuu samoin kuin kannettavissa solmuissa, eli se tapahtuu ajastimen keskeytyspalvelufunktiossa. Mutta toisin kuin kannettavien solmujen ohjelmistossa, päämoduulin tilakoneessa ei ole ESTABLISH\_BEACON-tilaa, jossa otetaan ensin selville muiden solmujen majakoiden ajoitukset ennen oman majakan ajoituksen asettamista. Päämoduuli vain yksinkertaisesti alkaa lähettämään majakkaa heti saatuaan virrat.

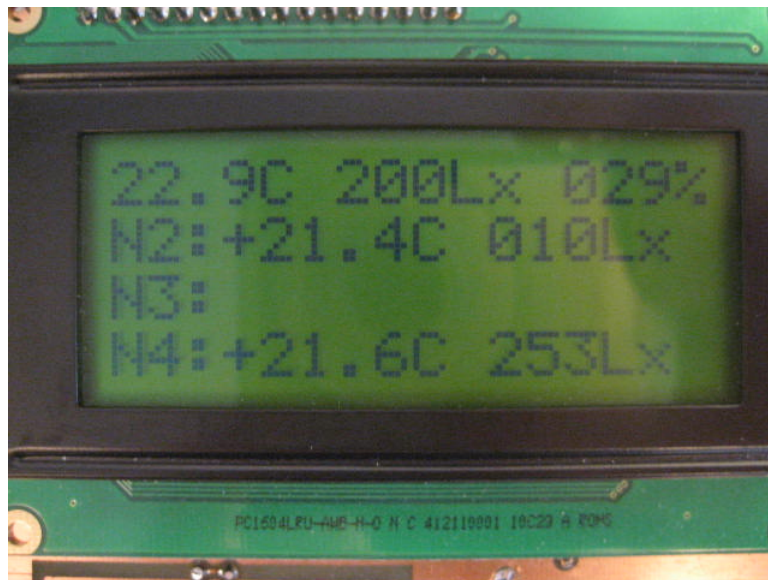
Päämoduuli alkaa suorittamaan tilakonetta tilasta DECIDE\_WAIT. Tämä tila on samankaltainen kuin kannettavissa solmuissa. Tässä tilassa yksinkertaisesti vain odotetaan tapahtumia ja siirrytään muihin tiloihin näiden perusteella. Jos GSM-moduuli aiheuttaa prosessorissa keskeytyksen laittamalla RI-pinnan alas, tämä tar-

koittaa tekstiviestin saapuneen ja päämoduuli siirtyy tilaan RECEIVE\_SMS. Jos taas radiomoduuli aiheuttaa keskeytyksen, päämoduuli siirtyy tilaan RECEIVE. Jos käyttäjä on painanut yhtä neljästä napista, päämoduuli siirtyy tilaan INPUT. Toisin kuin kannettavissa solmuissa, päämoduulilla ei ole erillistä kuuntelujaksoa, jonka päätteksi päämoduuli voisi mennä MEASURE-tilaan käyttämään sensoreita. Sen sijaan päämoduuli käyttää ajastinkeskeytystä mennäkseen MEASURE-tilaan kerran MAC-jaksossa DECIDE\_WAIT -tilasta.

RECEIVE-tila on miltei identtinen kannettavien solmujen RECEIVE-tilan kanssa. Poikkeuksena on se, että vastaanotettuja sensoridatoja ei laiteta FIFO:oon talteen, vaan jokaisella solmulla on paikkansa päämoduulin prosessorin muistissa, johon sensoridatat tallennetaan. RECEIVE-tilasta siirrytään HANDLE\_REQUEST-tilaan, jos radio vastaanotti DATAPIPE\_REQUEST-paketin. Muutoin siirrytään UPDATE\_SCREEN-tilaan.

HANDLE\_REQUEST-tila toimii aivan samalla tavalla kuin kannettavissa solmuissa, joten sen toimintaa ei kerrata enää tässä.

UPDATE\_SCREEN-tilassa päivitetään LCD-näytöllä näkyvät merkit. Näytöllä on viisi eri tilaa, ja näytön tilaa voi käyttäjä vaihtaa painamalla nappeja. Perustilassa näytöllä näkyy päämoduulin sensoridatat ylhäällä ja sen alapuolella kolmen eri kannettavan solmun sensoridatat. Tämä on esitettyinä kuvassa 5.16. Käyttäjä voi nappeja painamalla vaihdella, mitkä solmut näkyvät ruudulla.



**Kuva 5.16:** Päämoduulin LCD-näytön perustila.

Näytön toisessa tilassa ruudulla näytetään, koska viimeksi solmuilta on saatu sensoridataa. Näin käyttäjä voi arvioida, ovatko ruudulla näytetyt sensoridatat ajan tasalla ja ovatko solmut vielä verkossa kiinni. Kuva 5.17 näyttää tämän tilan.





**Kuva 5.17:** LCD-näytöllä näkyy, koska solmut ovat viimeksi lähettäneet sensori-dataa.

Kolmannessa näytön tilassa näytössä näkyy nykyinen kellonaika kuvan 5.18 mukaisesti.



**Kuva 5.18:** LCD-näytöllä näkyy nykyinen kellonaika.

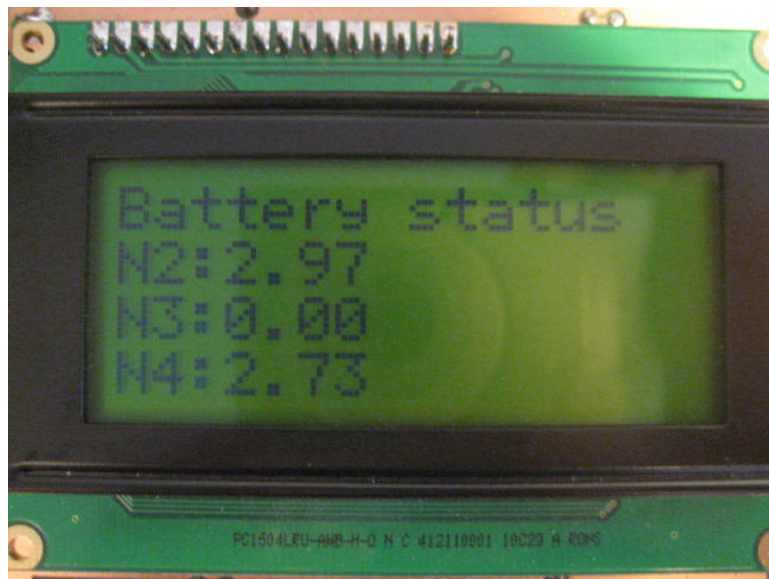
Neljännessä näytön tilassa käyttäjä voi asettaa nykyisen kelloajan käyttäen nappeja. Kuvassa 5.19 on esitetty tämä tilanne.





*Kuva 5.19: LCD-näytöllä näkyy kellon asettaminen.*

Viidennessä tilassa käyttäjä voi tarkkailla kannettavien solmujen käyttöjännitteiden tiloja. Kuvassa 5.20 on esitettyä tämä tilanne. Käyttäjä voi nappeja painamalla vaihdella, mitkä solmut näkyvät ruudulla.



*Kuva 5.20: LCD-näytöllä näkyy solmujen käyttöjännitteet.*

Kun LCD-näyttö on saatu päivitettyä, siirrytään tilaan DECIDE\_WAIT.

INPUT-tilaan tullaan DECIDE\_WAIT-tilasta käyttäjän painettua jotain nappia. Tilassa käsitellään napin painallus eri tavalla riippuen missä tilassa näyttö kyseisellä hetkellä on. Lopuksi siirrytään tilaan UPDATE\_SCREEN päivittämään napin painalluksen aiheuttamat muutokset näytöllä.

RECEIVE\_SMS-tilaan tullaan DECIDE\_WAIT-tilasta, kun GSM-moduuli on aiheuttanut keskeytyksen prosessorissa tekstiviestin saapumisen takia. Tässä tilassa prosessori lukee GSM-moduulilta USART-liitynnän kautta saapuneen tekstiviestin ja tarkistaa sisältääkö se jonkin taulukossa 1.1 olevan komennnon. Jos tekstiviestistä löytyy jokin käskyistä, prosessori vastaa siihen lähettämällä käyttäjälle tekstiviestinä takaisin pyydetty sensoritiedot. Tämän jälkeen siirrytään takaisin tilaan DECIDE\_WAIT.

HANDLE\_ALARM-tilaan tullaan, jos päämoduuli on vastaanottanut hälytyspaketin eli joko FRIDGE\_ALARM\_OPEN- tai SECURITY\_ALARM-paketin. Päämoduuli vastaa pakettiin lähettämällä käyttäjälle tekstiviestin “Jaakaappi au-ki!” tai “Murtohälytys!”.

## 6 Johtopäätökset

Sensoriverkkojen suunnittelu varsinkin langattoman reitityksen suhteen on yhä erittäin sovelluskohtainen asia. Sensoriverkkojen vaatimukset vaihtelevat sovelluksesta toiseen niin paljon, että on usein suunniteltava sensoriverkko jokaista sovellusta varten erikseen.

Tavallisia MAC- ja reititysprotokollia ei voida suoraan soveltaa sensoriverkkoihin, sillä ne eivät ota huomioon sensorisolmujen rajallisia prosessointikykyjä ja resursseja. Toisin kuin muissa tietoliikenneverkoissa sensoriverkoissa reititysprotokollat ovat datakeskeisiä, jossa varsinaisen solmun identiteetillä ei ole väliä, vaan solmuihin viitataan niiden tuottaman datan avulla. Lisäksi reitityksessä voidaan ottaa huomioon solmujen maantieteellistä sijaintia. Kaikki nämä seikat tekevät reitityksen sensoriverkoissa erilaiseksi haasteeksi kuin muissa tietoliikenneverkoissa. Joskin joitakin tavallisissa tietoliikenneverkoissa käytetyistä ideoista voidaan rajallisesti hyödyntää myös sensoriverkoissa.

Diplomityötä varten rakennetussa sensoriverkossa oli aluksi ongelmia reitityksen kanssa. Sensorisolmut eivät pysyneet tahdissa reititysprotokollan vaatimalla tavalla. Ajoitusongelmat johtuivat solmujen oskillaattoreista, joiden taajuudet poikkesivat toisistaan niin paljon, että solmut ajautuivat epätahtiin nopeammin, kuin oli alunperin luultu. Ongelmaa lievitettiin ottamalla kohdenaapurin majakan ajoitus ylös jokaisella MAC-jaksolla sen sijaan, että se tehtäisiin vain kerran. Vaikka tämä toimenpide auttoi reititystä, se ei poistanut ajoitusongelmia kokonaan. Ajoitusongelmat ratkaistiin lopulta antamalla solmujen kuuntelujaksojen mennä päällekkäin, jolloin linkit solmujen välillä säilyivät pidempään kokojaksokuunteluiden tekemättä jättämisen takia.

Sensoriverkon energiankulutuksen mittauksien mukaan yksittäisen solmun patterin kestävät noin kolme kuukautta mittaustoimintaa. Tämän jälkeen solmu pystyy vielä reitittämään noin vuoden verran, koska radiomoduulin vaatima käyttöjännite

on pienempi kuin sensorien.

Kun sensoriverkon reititys oli saatu kuntoon, varsinaisten sovelluksien lisääminen meni helposti ja nopeasti. Sensoriverkon käyttäminen matkapuhelimen avulla, jääkaapin tilan seuranta ja murtohälytyssovellus saatiin toimimaan melko vaivattomasti sen jälkeen, kun reititys oli saatu kuntoon.

## Lähteet

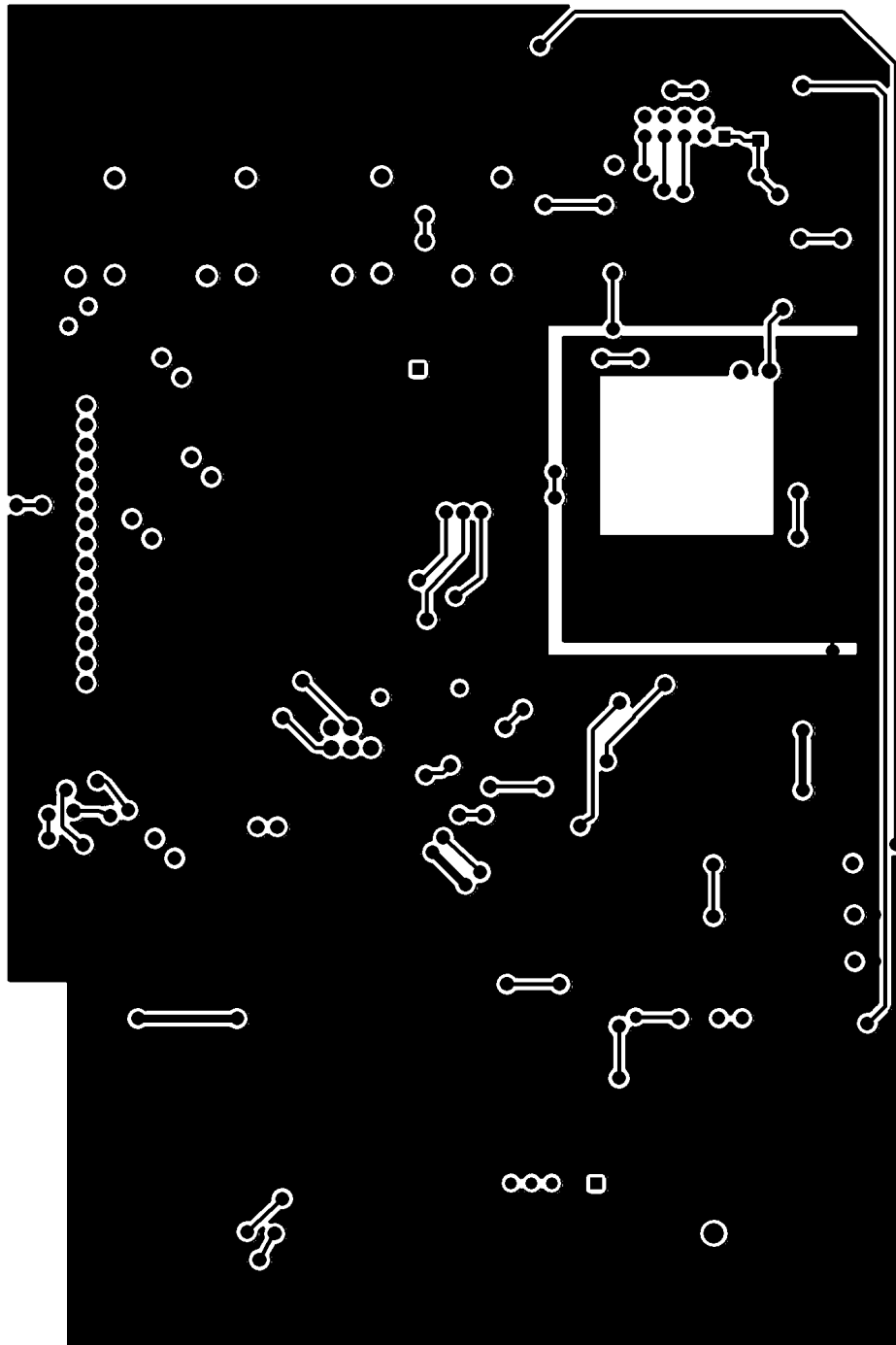
- [1] Knaian, A. A Wireless Sensor Network for Smart Roadbeds and Intelligent Transportation Systems. Massachusetts Institute of Technology, 2000. Saatavissa: <http://resenv.media.mit.edu/pubs/theses/AraKnaian-Thesis.pdf>
- [2] Agilent Technologies. Hints for Selecting the Correct Temperature Sensor for Your Application USA, April 16, 2008. Saatavissa: <http://cp.literature.agilent.com/litweb/pdf/5989-7800EN.pdf>
- [3] Allaboutcircuits.com Insulator breakdown voltage. [WWW]. [viitattu: 17.2.2010] Saatavissa: [http://www.allaboutcircuits.com/vol\\_1/chpt\\_12/8.html](http://www.allaboutcircuits.com/vol_1/chpt_12/8.html)
- [4] Analog Devices. ADT7301 Datasheet. USA, 2005.
- [5] Atmel Corporation. Atmega324p datasheet. 2010.
- [6] Atmel Corporation. Atmega88 datasheet. 2011.
- [7] Baggio, A. Wireless sensor networks in precision agriculture. Delft University of Technology, Alankomaat Saatavissa: [www.sics.se/realwsn05/papers/baggio05wireless.pdf](http://www.sics.se/realwsn05/papers/baggio05wireless.pdf)
- [8] Cha, H., Kim, Y., Sihm, H. Y-MAC: An Energy-efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks. 2008 International Conference on Information Processing in Sensor Networks. Saatavissa: <http://www.cecs.uci.edu/papers/cpsweek08/papers/ipsn08/3157a053.pdf>
- [9] Dargie, W., PoellaBauer, C. Fundamentals Of Wireless Sensor Networks 1. painos. 2010, John Wiley & Sons. 311 p.
- [10] Estrin, D., Govindan, R., Intanagonwiwat, C. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. Saatavissa: [http://www.isi.edu/div7/publication\\_files/directed\\_diffusion\\_scalable.pdf](http://www.isi.edu/div7/publication_files/directed_diffusion_scalable.pdf)
- [11] Estrin D., Govindan R., Yu, Y. Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks. 2001. Saatavissa: [http://www2.parc.com/spl/members/zhao/stanford-cs428/readings/Networking/Estrin\\_geo-routing01.pdf](http://www2.parc.com/spl/members/zhao/stanford-cs428/readings/Networking/Estrin_geo-routing01.pdf)
- [12] Estrin D., Heidemann J., Xu, Y. Geography-informed Energy Conservation for Ad Hoc Routing. Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (ACM Mobicom), July 16-21, 2001, Rome, Italy Saatavissa: <http://research.cens.ucla.edu/people/estrin/resources/conferences/2001july-Xu-Estrin-Geography.pdf>
- [13] Hac, A. Wireless Sensor Network Designs. 1. painos. 2003, John Wiley & Sons. 391 p.
- [14] Havinga, P.J.M., van Hoesel, L.F.W. A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches Saatavissa: [http://eprints.eemcs.utwente.nl/12718/01/VanHoesel\\_INSS04\\_048.pdf](http://eprints.eemcs.utwente.nl/12718/01/VanHoesel_INSS04_048.pdf)

- [15] Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Proceedings of the 33rd Hawaii International Conference on System Sciences ,2000 . Saatavissa:<http://pdos.csail.mit.edu/decouto/papers/heinzelman00.pdf>
- [16] Honeywell. HIH-5030/5031 Series Low Voltage Humidity Sensors USA, 2010.
- [17] Jason W.P. Ng, Benny P.L.Lo Ubiquitous Monitoring Environment for Wearable and Implantable Sensors, Lontoo, UK Saatavissa: <http://www.ubicomp.org/ubicomp2004/adjunct/posters/ng.pdf>
- [18] Karnapke, R., Mank, S., Nolte, J. MLMAC – An Adaptive TDMA MAC Protocol for Mobile Wireless Sensor Networks. 2009 Old City Publishing, Inc. Saatavissa: <https://www-docs.tu-cottbus.de/betriebssysteme/public/publication/2009/Aswin120.pdf>
- [19] Kaynia, M., Jindal, N. Performance of ALOHA and CSMA in Spatially Distributed Wireless Networks [Viitattu 7.11.2011] Saatavissa: <http://www.ece.umn.edu/nihar/papers/ICC-paper-Mariam.pdf>
- [20] Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., ja Turon, M. Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks, 2007, Saatavissa: [http://www.eecs.berkeley.edu/binetu-de/work/ipsn07\\_ggb.pdf](http://www.eecs.berkeley.edu/binetu-de/work/ipsn07_ggb.pdf)
- [21] Knaian, A. A Wireless Sensor Network for Smart Roadbeds and Intelligent Transportation Systems. Massachusetts Institute of Technology, 2000. Saatavissa: <http://resenv.media.mit.edu/pubs/theses/AraKnaian-Thesis.pdf>
- [22] Kredo II,K., Mohapatra P. Medium Access Control in Wireless Sensor Networks. 2006, [viitattu 7.11.2011] Saatavissa: [spirit.cs.ucdavis.edu/pubs/journal/SensorMac.pdf](http://spirit.cs.ucdavis.edu/pubs/journal/SensorMac.pdf)
- [23] Laynetworks.com. Aloha Protocols. [WWW]. [viitattu 7.11.2011]. <http://www.laynetworks.com/ALOHA%20PROTOCOL.htm>
- [24] Lindsey, S., Raghavendra, C.S. Power-Efficient Gathering in Sensor Information Systems. Saatavissa: <http://ceng.usc.edu/raghu/pegasisrev.pdf>
- [25] Mainwaring A., Polastre J. Wireless Sensor Networks for Habitat Monitoring Atlanta,Georgia, USA, 2002, Saatavissa: <http://www.polastre.com/papers/wsna02.pdf>
- [26] Manjeshwar, A., Agrawal, D.P. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. University of Cincinnati, Ohio 2001, Saatavissa:[http://www.sis.pitt.edu/is3966/group3\\_paper3.pdf](http://www.sis.pitt.edu/is3966/group3_paper3.pdf)
- [27] Microchip Technology Inc. MCP6291/1R/2/3/4/5 Datasheet 2007
- [28] Nordic Semiconductors. nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification v1.0, Syyskuu 2008,

- [29] Perry, R.H., Green, D.W. Perry's Chemical Engineers' Handbook 7. painos, McGraw-Hill
- [30] Rodoplu, V., Meng, T.H. Minimum Energy Mobile Wireless Networks. IEE Journal On Selected Areas In Communications, vol. 17, NO. 8, August 1999, Saatavissa: [http://www.ece.ucsb.edu/stnlabs/pubs/rom99\\_jsac.pdf](http://www.ece.ucsb.edu/stnlabs/pubs/rom99_jsac.pdf)
- [31] Tekniikan käsikirja, osa3: Sähkötekniikka 1969 8. painos Gummerus
- [32] Tsuchiya, P.F. The Landmark Hierarchy: A New Hierarchy for Routing In Very Large Networks. 1988, Saatavissa:<http://www.cs.cmu.edu/~srini/15-744/S02/readings/Tsu88.pdf>
- [33] Weaver, J. A Wearable Health Monitor To Aid Parkinson Disease Treatment. Massachusetts Institute of Technology, 2003. Saatavissa:<http://dspace.mit.edu/bitstream/handle/1721.1/17598/53364297.pdf>
- [34] Wikipedia. Health Care in The United States. [WWW]. [Viitattu 29.12.2011] [http://en.wikipedia.org/wiki/Health\\_care\\_in\\_the\\_United\\_States](http://en.wikipedia.org/wiki/Health_care_in_the_United_States)
- [35] Wikipedia. Lux. [WWW]. [Viitattu 18.11.2011] <http://en.wikipedia.org/wiki/Lux>
- [36] Wikipedia. Silicon bandgap temperature sensor. [WWW]. [Viitattu 20.11.2011] [http://en.wikipedia.org/wiki/Silicon\\_bandgap\\_temperature\\_sensor](http://en.wikipedia.org/wiki/Silicon_bandgap_temperature_sensor)
- [37] Wikipedia. Thermocouple. [WWW]. [Viitattu 19.11.2011]. <http://en.wikipedia.org/wiki/Thermocouple>
- [38] Wilson, J.S. Sensor Technology Handbook 1.painos. USA, 2005, Newnes. 691 p.



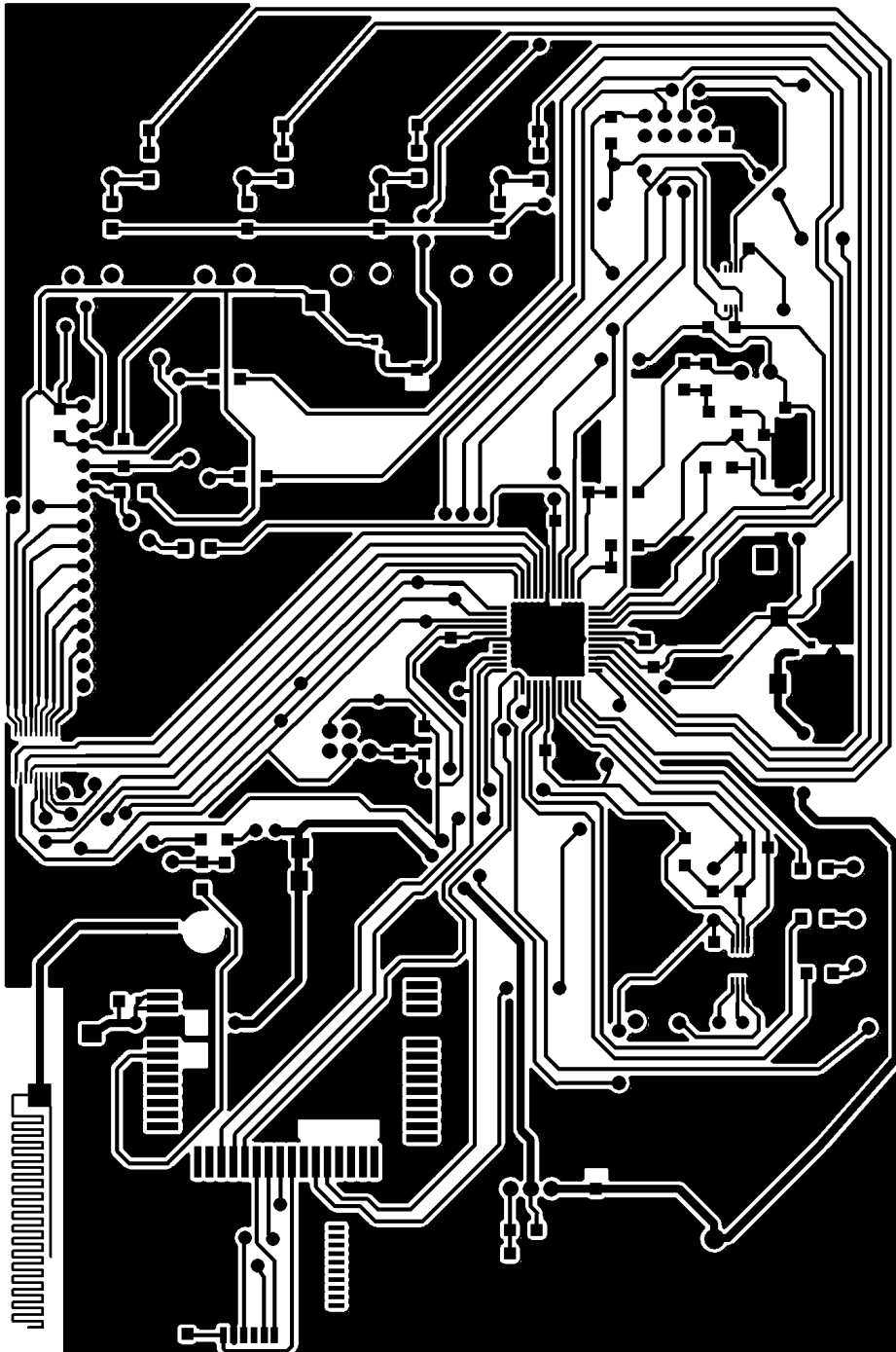
## B Liite 2: Päämoduulin alapuolen valotusmaski



*Kuva B.1:* Päämoduulin alapuolen valotusmaski

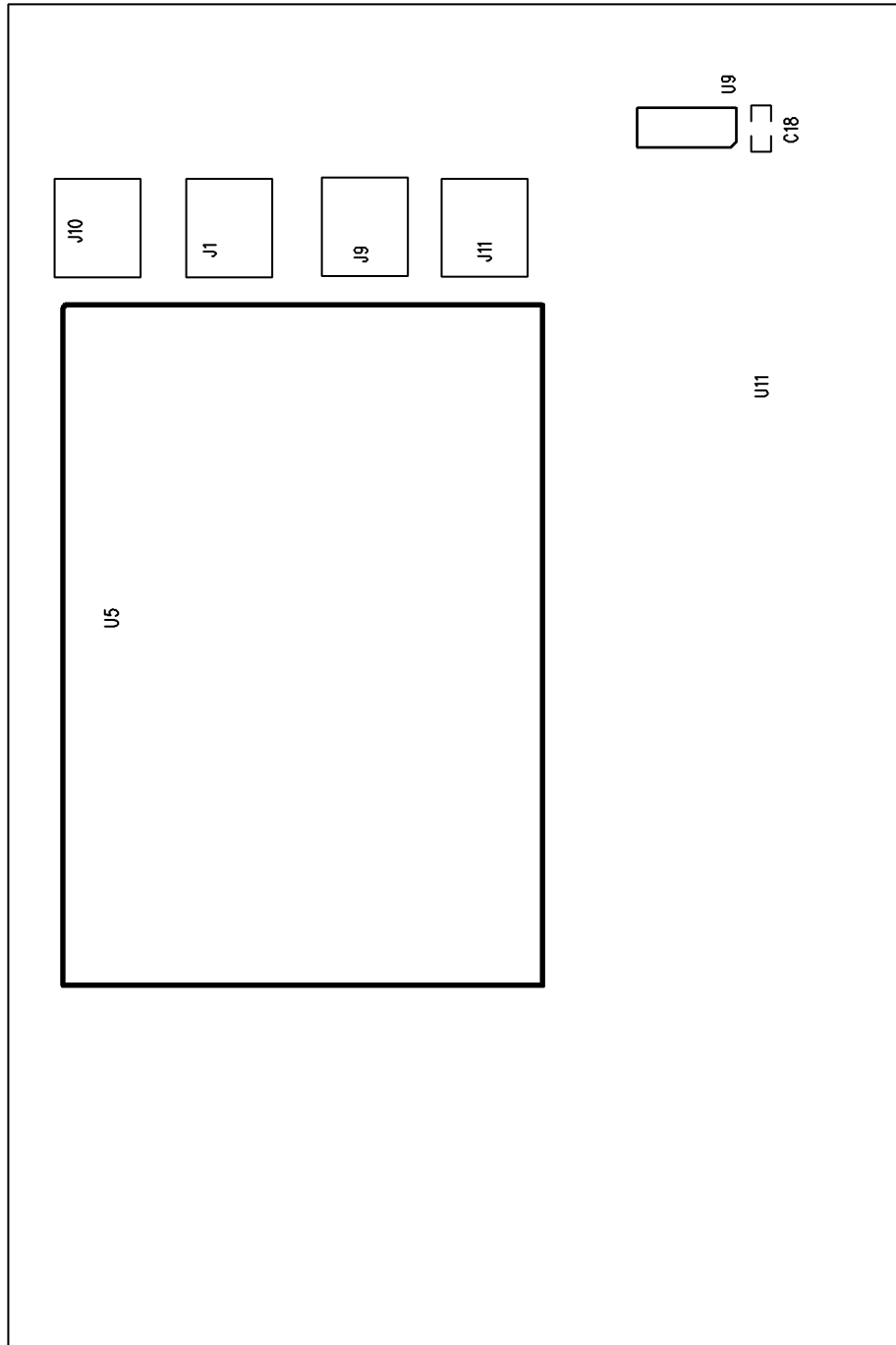


## C Liite 3: Päämoduulin yläpuolen valotusmaski



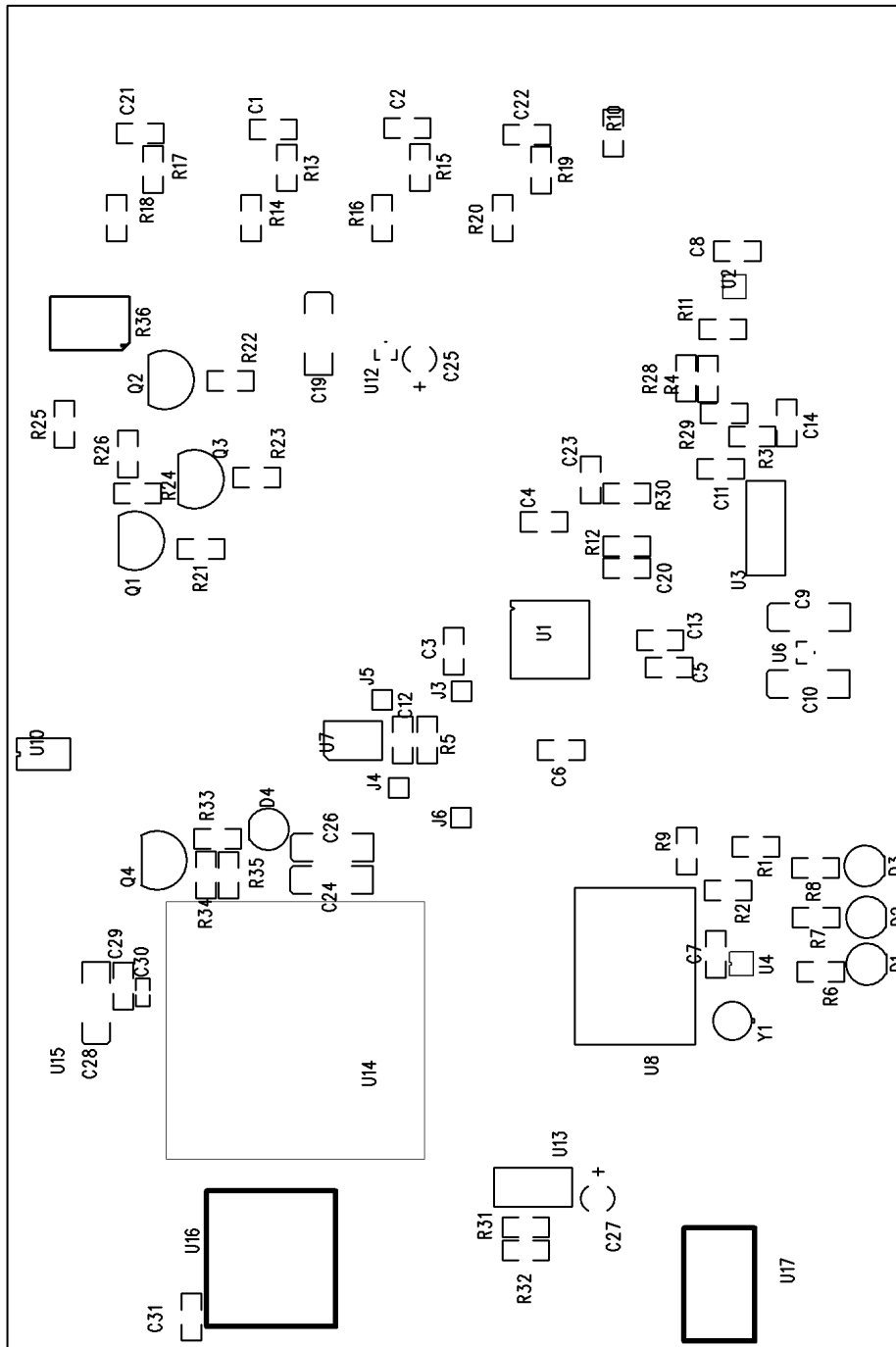
*Kuva C.1:* Päämoduulin yläpuolen valotusmaski

## D Liite 4: Päämoduulin alapuolen osasijoittelu



*Kuva D.1:* Päämoduulin alapuolen osasijoittelu

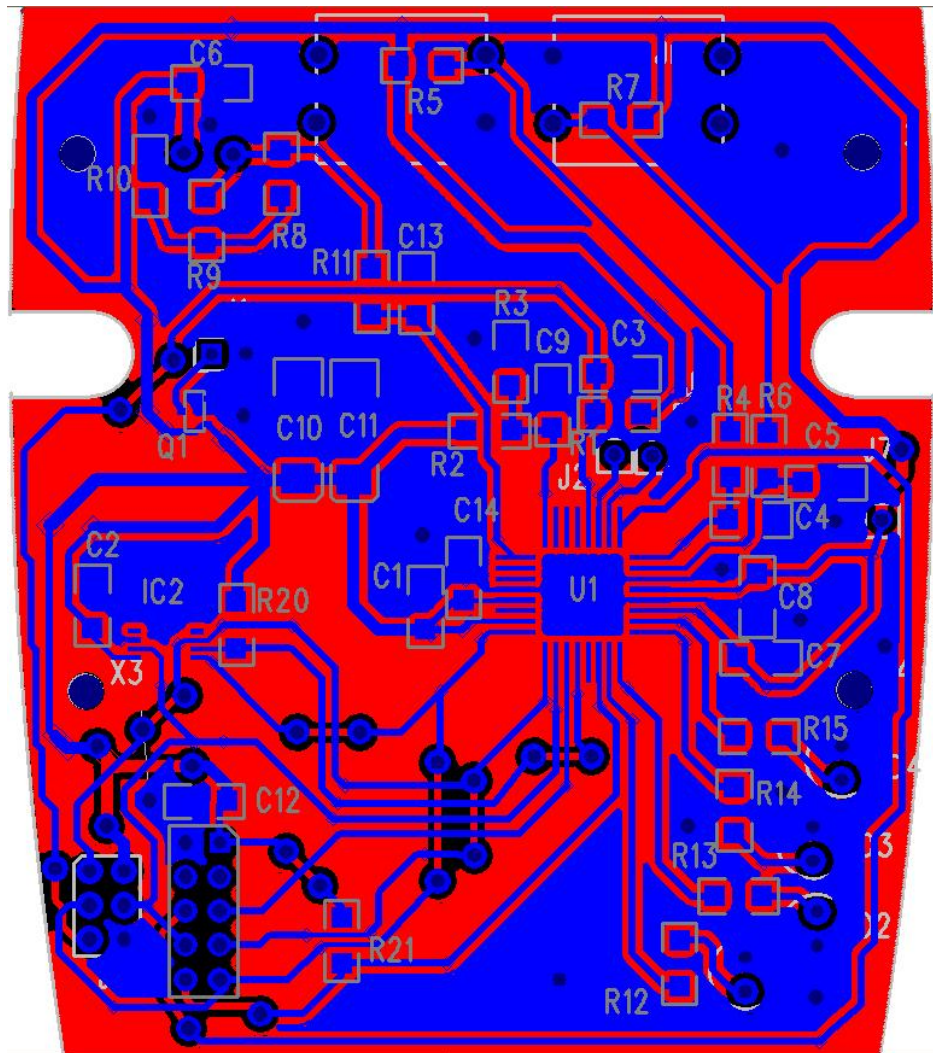
## E Liite 5: Päämoduulin yläpuolen osasijoittelu



**Kuva E.1:** Päämoduulin yläpuolen osasijoittelu

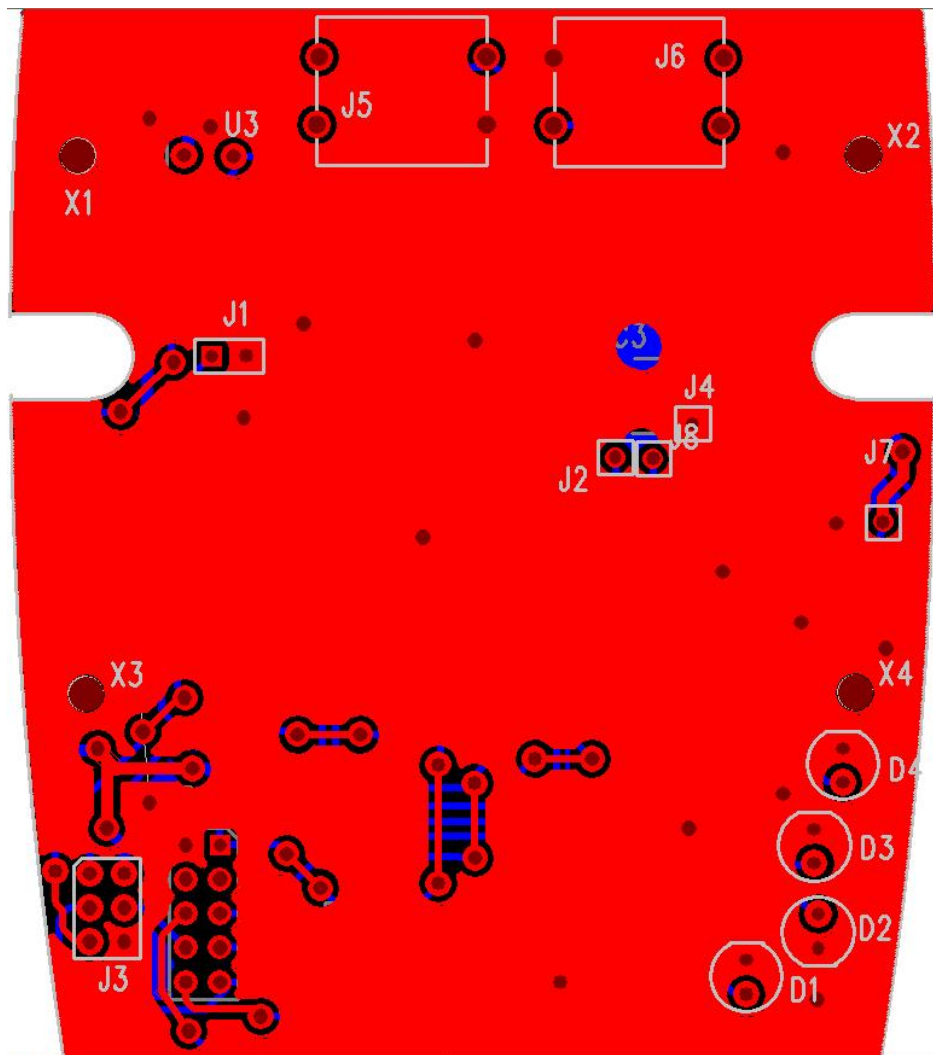


## G Liite 7: Sensorisolmun 3 alapuolen asemapiirustus



*Kuva G.1:* Solmun 3 alapuolen asemapiirustus

## H Liite 8: Sensorisolmun 3 yläpuolen asemapiirustus



*Kuva H.1:* Solmun 3 yläpuolen asemapiirustus