

**Jukka Nissi**

MELUA MITTAAVAN LANGATTOMAN ANTURIVERKON  
KLUSTEROINTI

Tietotekniikan pro gradu -tutkielma

Ohjelmistotekniikan linja

22.11.2015

Jyväskylän yliopisto

Tietotekniikan laitos

**Tekijä:** Jukka Nissi

**Yhteystiedot:** jukka.nissi@anvianet.fi

**Työn nimi:** Melua mittaavan langattoman anturiverkon klusterointi

**Title in English:** Clustering in noise measuring wireless sensor network

**Ohjaaja:** Ismo Hakala

**Työ:** Pro gradu -tutkielma

**Sivumäärä:** 92

**Linja:** Ohjelmistotekniikka.

**Avainsanat:** langaton anturiverkko, klusterointi, melu, MH-LEACH, DSBCA

**Keywords:** wireless sensor network, clustering, noise, MH-LEACH, DSBCA

**Tiivistelmä:** Langattomat anturiverkot soveltuvat moniin eri käyttötarkoituksiin, mutta anturisolmujen niukat resurssit asettavat kuitenkin omat haasteensa. Erityisesti anturisolmujen energiaa tulisi säästää, jotta verkon elinikä olisi mahdollisimman pitkä. Energiankulutusta voidaan vähentää jakamalla anturisolmut ryhmiin, klustereihin. Klusterointiprotokollan suoritus tapahtuu kierroksittain sisältäen yleensä seuraavat neljä vaihetta: pääsolmun valinta, klusterin muodostus, datan kokoaminen ja tiedonsiirto. Klusterointiprotokollan valinta voi olla haastavaa, sillä valintaan vaikuttaa monet tekijät. Klusterointiprotokollalle asetetut tavoitteet vaihtelevat ja ne määräytyvät usein sovelluksen vaatimusten mukaan. Myös pääsolmun ominaisuudet voivat vaihdella ja pääsolmun valinta voidaan tehdä todennäköisyyteen perustuen, deterministisesti tai ennalta määräten. Klusterin muodostuksessa täytyy huomioida mm. klustereiden lukumäärä, koko ja tiheys sekä klustereiden sisäinen topologia. Lisäksi datan kokoamispiste täytyy sijoittaa oikein ja arvioida aiheuttaako datan kokoamisesta johtuva viive ongelmia. Myös klusterin sisäinen ja klustereiden välinen tiedonsiirto täytyy huomioida klusterointiprotokollan valinnassa. Reititystä käsitellään erikseen, koska tiedonsiirto on keskeinen osa klusterointiprotokollan

toimintaa. Työssä kerrotaan myös melusta ja melun mittauksesta langattomassa anturiverkossa, esitellään kivimurskaimen melua mittaava langaton anturiverkko sekä vertaillaan MH-LEACH ja DSBCA -klusterointiprotokollan toimintaa kyseisessä verkossa.

**Abstract:** Wireless sensor networks are suitable for many different applications, however, the sensor nodes scarce resources set their own challenges. The energy of sensor nodes should be saved to prolong lifetime of the network. Energy consumption can be reduced by dividing the sensor nodes into groups, clusters. Clustering protocols are executed in rounds, one round includes usually four steps: cluster head selection, cluster formation, data aggregation and data transfer. The selection of the clustering protocol can be challenging, due to many factors that influence the choice of the right protocol. Objectives of the clustering protocols can vary and often depend on the application requirements. Cluster head properties can also vary a lot and the cluster head can be selected probability based, deterministic or preset. In the cluster formation phase for instance, you should consider the number of clusters, the size and density as well as the internal topology of the clusters. In addition, the data aggregation point must be positioned correctly and evaluate the effects that aggregation delay may cause. Also intra-cluster and inter-cluster communication must be taken into account when choosing the clustering protocol. Routing is discussed in own chapter in this study, because communication is a key part of the clustering protocol activities. This study also discusses about the noise and noise monitoring in the wireless sensor networks. It also presents the wireless sensor network that monitors noise of the stone crusher, as well as compares operation of MH-LEACH and DSBCA clustering protocols in that network.

## Termiluettelo

ACE	An Emergent Algorithm for Highly Uniform Cluster Formation
AD-muunnin	Analogia-digitaalimuunnin
BCDCP	Base-Station Controlled Dynamic Clustering Protocol
BMP	Bit Map Picture
CDMA	Code Division Multiple Access
CHEP	Cluster Head Election Protocol
CNS	Center at Nearest Source
CSMA	Carrier Sense Multiple Access
DCA	Distributed Clustering Algorithm
DSBCA	A Balanced Clustering Algorithm with Distributed Self-Organization for Wireless Sensor Networks
EEHC	Energy Efficient Hierarchical Clustering
FFD	Full Function Device
GIT	Greedy Incremental Tree
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
HEED	Hybrid Energy-Efficient Distributed Clustering
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standardization Organisation
JPEG	Joint Photographic Experts Group
LEACH	Low Energy Adaptive Clustering Hierarchy
LEACH-C	LEACH-centralized

LEACH-ERE	LEACH Expected Residual Energy
LLC	Link Layer Control
LR-WPAN	Low Rate Wireless Personal Area Network
MAC	Medium Access Control
MBC	Mobility-based clustering protocol
MH-LEACH	Multi-Hop Low Energy Adaptive Clustering Hierarchy
OSI	Open Systems Interconnection
Overhead	Verkon ylimääräinen viestiliikenne, joka ei ole varsinaista dataa
PANEL	Position-based aggregator node election protocol
PAN-koordinaattori	LR-WPAN verkon hallintaa hoitava laite
PSO-BC	A centralized PSO-based clustering algorithm
RFD	Reduced Function Device
SAR	Sequential Assignment Routing
SPIN	Sensor Protocols for Information via Negotiation
SPT	Shortest Paths Tree
SSCS	Service Specific Convergence Sublayer
TDMA	Time Division Multiple Access
TinyPEDS	Tiny persistent encrypted data storage in asynchronous wireless sensor networks

# Sisältö

<b>1</b>	<b>JOHDANTO</b> .....	<b>1</b>
<b>2</b>	<b>LANGATTOMAT ANTURIVERKOT</b> .....	<b>3</b>
2.1	SOVELLUSKOHTEET .....	3
2.2	PROTOKOLLAPINO .....	5
2.2.1	OSI -malli .....	5
2.2.2	IEEE 802.15.4 LR-WPAN.....	7
2.3	VERKKOKOMPONENTIT .....	8
2.4	VERKKOTOPOLOGIAT.....	9
2.4.1	Tähtitopologia.....	9
2.4.2	Vertaisverkkotopologia.....	11
<b>3</b>	<b>REITITYS JA VIESTINTÄ LANGATTOMISSA ANTURIVERKOISSA</b> .....	<b>14</b>
3.1	REITITYKSEN HAASTEET LANGATTOMISSA ANTURIVERKOISSA .....	15
3.2	VIESTINTÄ .....	17
3.3	REITITYSPROTOKOLLAT.....	18
3.3.1	Viestinnän aloittaja .....	19
3.3.2	Reitin muodostus .....	19
3.3.3	Verkon rakenne.....	20
3.3.4	Luokittelu protokollan toiminnan mukaan .....	22
3.3.5	Luokittelu seuraavan hypyn valinnan mukaan .....	25
<b>4</b>	<b>KLUSTEROINTI</b> .....	<b>27</b>
4.1	TAVOITTEET .....	28
4.2	KLUSTEROINTIMENETELMÄT .....	30
4.3	VAIHEET .....	33
4.4	PÄÄSOLMU.....	33
4.4.1	Pääsolmun ominaisuuksia.....	33
4.4.2	Pääsolmun valinta .....	35
4.4.3	Pääsolmun kierrätys .....	41
4.5	KLUSTERIN MUODOSTUS .....	42
4.6	DATAN KOKOAMINEN .....	44
4.6.1	Datan kokoamispisteiden sijoittaminen .....	45
4.6.2	Datan kokoamiskeemoja .....	45
4.6.3	Datan kokoamisesta johtuva viive .....	48
4.7	KLUSTERIN SISÄINEN JA KLUSTEREIDEN VÄLINEN TIEDONSIIRTO.....	48
<b>5</b>	<b>KLUSTEROINTIPROTOKOLLAT</b> .....	<b>51</b>

5.1	LEACH .....	51
5.1.1	Pääsolmun valinta ja kierrätys .....	52
5.1.2	Klusterin muodostus ja siirtoaikataulun luonti .....	52
5.1.3	Tiedonsiirtovaihe .....	53
5.1.4	Vahvuudet ja heikkoudet .....	54
5.1.5	MH-LEACH .....	55
5.2	DSBCA .....	58
5.2.1	Pääsolmun valinta .....	60
5.2.2	Klustereiden muodostus.....	62
5.2.3	Pääsolmun kierrätys .....	62
5.2.4	Vahvuudet .....	63
<b>6</b>	<b>KLUSTEROINTI MELUVERKOSSA.....</b>	<b>64</b>
6.1	MELU.....	64
6.2	MELUNMITTAUS LANGATTOMISSA ANTURIVERKOISSA .....	65
6.3	KIVIMURSKAIMEN MELUA MITTAAVAN LANGATTOMAN ANTURIVERKON KLUSTEROINTI.....	68
6.3.1	MH-LEACH .....	70
6.3.2	DSBCA .....	74
6.3.3	Klusterointiprotokollien vertailu.....	79
<b>7</b>	<b>YHTEENVETO .....</b>	<b>84</b>
	<b>LÄHTEET .....</b>	<b>87</b>

# 1 Johdanto

Langattomat anturiverkot koostuvat pienistä, edullisista, vähän virtaa kuluttavista ja monitoimisista anturisolmuista. Ne voivat mitata monia eri suureita kuten lämpötilaa, valoisuutta, kosteutta, säteilyä ja melua. Anturisolmut koostuvat neljästä peruskomponentista: anturista, prosessorista, lähetinvastaanotinyksiköstä ja akusta. Lisäksi ne voivat sisältää muita sovelluskohtaisia lisäkomponentteja. Johtuen anturisolmujen rajallisista resursseista, täytyy langattomissa anturiverkoissa kiinnittää erityistä huomiota verkon ylimääräiseen viestiliikenteeseen ja anturisolmujen lepoaikoihin. Muuten anturisolmujen verkossa oloaika jää lyhyeksi, sillä niiden pienet akut eivät kestä jatkuvaa energiankulutusta. Verkon viestiliikennettä ja samalla anturisolmujen energiankulutusta voidaan vähentää tehokkaalla verkon rakenteella eli jakamalla anturisolmut ryhmiin, klustereihin. Klusterit koostuvat joukosta anturisolmuja, joille jaetaan erilaisia rooleja kuten pääsolmu, gateway-solmu tai jäsensolmu. Pääsolmu on klusterin koordinaattori, jonka vastuulla on yleensä klusterin sisäisen tiedonsiirron järjestäminen, klusterin ylläpito sekä jäsensolmujen välittämän datan kokoaminen ja välitys.

Klusterointiprotokollien kirjo on laaja ja sopivan klusterointiprotokollan valinta voi olla haastavaa. Valintaan vaikuttaa niin sovelluksen asettamat vaatimukset, anturisolmujen resurssit kuin myös verkon toimintaympäristö. Lisäksi monet klusterointiprotokollat on suunniteltu hyvin spesifisiin verkkoihin. Tämän pro gradu -tutkielman aiheena on melua mittaavan langattoman anturiverkon klusterointi. Työssä tarkastellaan yleisesti klusterointia langattomissa anturiverkoissa, esitellään erilaisia klusterointiprotokollia sekä käsitellään klusterointiprotokollan valinnassa huomioitavia asioita. Lopuksi esitetään kivimurskaimen melua mittaava langaton anturiverkko, simuloidaan MH-LEACH ja DSBCA -protokollien toimintaa kyseisessä verkossa sekä vertaillaan näitä protokollia ja pohditaan sopivatko ne kivimurskaimen meluverkon klusterointiin.

Jotta lukija saa käsityksen langattomista anturiverkoista sekä niiden erityispiirteistä, käsitellään tutkielman alussa langattomia anturiverkkoja. Luvussa käydään läpi sovelluskohteita sekä esitetään protokollapino, verkkokomponentit ja eri verkkotopologiat. Koska reititys liittyy tiiviisti klusterointiprotokollan toimintaan, käsitellään luvussa kolme



erikseen langattomien anturiverkkojen reititystä ja viestintää. Neljännessä luvussa käsitellään tarkemmin klusterointia langattomissa anturiverkoissa sekä esitetään klusteroinnin tavoitteita ja klusterointimenetelmiä. Luvussa käsitellään myös tarkemmin pääsolmujen ominaisuuksia ja tehtäviä sekä esitellään pääsolmun valintamenetelmiä ja käsitellään pääsolmun kierrätystä. Lisäksi käsitellään klusterin muodostusta, datan kokoamista sekä klustereiden tiedonsiirtoa. Luvussa viisi esitellään klusterointiprotokollat LEACH, MH-LEACH ja DSBCA. Lopuksi käsitellään yleisesti melua ja melun mittausta langattomissa anturiverkoissa sekä esitellään kivimurskaimen melua mittaava langaton anturiverkko. Luvun lopussa simuloidaan klusterointiprotokollien MH-LEACH ja DSBCA toimintaa kivimurskaimen meluverkossa ja vertaillaan niiden sopivuutta kivimurskaimen meluverkkoon.

## 2 Langattomat anturiverkot

Viimeaikaisen langattoman tietoliikenteen ja elektroniikan huima kehitys on mahdollistanut halvat, vähän virtaa kuluttavat ja monitoimiset anturisolmut, jotka ovat pieniä ja pystyvät kommunikoimaan toisilleen langattomasti lyhyillä etäisyyksillä. Nämä pienet anturisolmut kykenevät mittaamaan eri suureita, prosessoimaan dataa sekä pitävät sisällään tiedonvälitykseen tarvittavat komponentit. Langattomat anturiverkot edustavat merkittävää kehitystä perinteisiin antureihin.

Langaton anturiverkko koostuu yleensä suuresta määrästä anturisolmuja, jotka on levitetty tiheästi tietyn alueen sisälle tai hyvin lähelle sitä. Koska anturisolmujen sijaintia verkossa ei tarvitse verkon rakennusvaiheessa määritellä, voidaan anturisolmut ripotella satunnaisesti vaikeapääsyiseen maastoon tai luonnonkatastrofialueelle. Toisaalta tämä tarkoittaa myös sitä, että langattomien anturiverkkojen protokollien ja algoritmien täytyy kyetä organisoimaan verkko itsenäisesti. Toinen langattomien anturiverkkojen ainutlaatuinen ominaisuus on anturisolmujen tehokas yhteistoiminnallisuus. Jokaisella anturisolmulla on prosessori ja sen sijaan, että se lähettäisi raakadataa tiedonkeruusolmulle, käyttävät anturisolmut niiden dataprosessointi mahdollisuutta. Tällöin anturisolmut toteuttavat paikallisesti yksinkertaisimmat laskutoimitukset ja toimittavat vain vaadittua tai osittain prosessoitua dataa tiedonkeruusolmulle.

### 2.1 Sovelluskohteet

Langattomat anturiverkot soveltuvat moniin sovelluksiin, sillä ne voivat koostua monentyyppisistä antureista. Akyildizin ym. [4] mukaan anturit voivat mitata mm. lämpötilaa, kosteutta, ajoneuvojen liikettä, valoa, painetta, maaperän koostumusta, melua, kohteiden läsnäoloa tai poissaoloa, kohteen mekaanista rasitusta tai kohteen muita ominaisuuksia kuten nopeutta, suuntaa ja kokoa. Anturisolmuja voidaan käyttää jatkuvaan mittaamiseen, tapahtumien havainnoimiseen ja tunnistamiseen, paikantamiseen sekä toimilaitteiden paikalliseen valvontaan. Akyildiz ym. [4] jakavat sovellukset seuraavasti viiteen ryhmään: sotilas-, ympäristö-, terveys- ja kotisovelluksiin sekä muihin kaupallisiin sovelluksiin.

Langaton anturiverkko voi olla kiinteä osa sotilasjärjestelmää ja sitä voidaan hyödyntää niin komento-, hallinta-, viestintä-, laskenta-, tiedustelu-, valvonta- ja paikantamisjärjestelmissä. Langattomat anturiverkot sopivat hyvin sotilaskäyttöön, sillä niiden käyttöönotto on nopeaa ja ne ovat itseorganisoituvia sekä vikasietoisia. Sotilaskäytössä langattomia anturiverkkoja voidaan käyttää niin omien joukkojen, laitteistojen ja ampumatarvikkeiden seurantaan kuin myös vihollisjoukkojen havaitsemiseen ja liikkeiden seurantaan. Myös taistelukentällä käytettävien ydin-, biologisten ja kemiallisten aseiden havaitseminen välittömästi mahdollistaa joukkojen suojautumisen ja mahdollisen evakuoinnin aloittamisen.

Ympäristösovelluksissa langattomia anturiverkkoja käytetään moniin tarkoituksiin. Seurataan esimerkiksi lintujen, pienten eläinten ja hyönteisten liikkeitä tai kasveihin ja eläimiin vaikuttavia ympäristöolosuhteita, kuten kastelua tai maaperän koostumusta. Varsinkin täsmäviljelyssä on tärkeää tietää maaperän kosteus ja koostumus, jotta kasville voidaan taata sopiva kasvuympäristö. Langattomia anturiverkkoja voidaan käyttää myös erilaisten luonnonkatastrofien havainnointiin ja seurantaan, kuten esimerkiksi metsäpalojen ja tulvien havaitsemiseen.

Langattomia anturiverkkoja voidaan hyödyntää myös monenlaisissa terveyssovelluksissa. Esimerkiksi ihmiseen asennettavalla anturilla voidaan mitata tietyin väliajoin erilaisia fysiologisia tietoja ja näitä talletettuja tietoja voidaan käyttää myöhemmin lääketieteelliseen tutkimiseen. Asennetun anturiverkon avulla voidaan myös seurata vanhusten käyttäytymistä ja havaita esimerkiksi vanhuksen kaatuminen. Nämä pienet anturisolmut sallivat myös vapaamman liikkumisen, jolloin lääkärit voivat tunnistaa ennalta määriteltäviä oireita aikaisemmin. Langattomia anturiverkkoja voidaan käyttää myös seuraamaan ja valvomaan lääkäreitä ja potilaita sairaalassa. Näin voidaan valvoa esimerkiksi potilaan sykettä tai verenpainetta sekä tarvittaessa paikantamaan potilas tai lääkäri.

Myös kotiautomaatiossa voidaan hyödyntää langattomia anturiverkkoja. Kodin laitteisiin voidaan integroida anturisolmu, jolloin laitteet voivat kommunikoida keskenään ja

kytkettyä ulkoiseen verkkoon esimerkiksi Internetin kautta. Näin loppukäyttäjät voivat hallita kodin laitteita helpommin niin paikallisesti kuin etänäkin.

Langattomien anturiverkkojen tarjoamat kaupalliset mahdollisuudet ovat lähes rajattomat. Esimerkkejä mahdollisia sovelluksista ovat kiinteistön olosuhteiden seuranta ja ohjaus, interaktiiviset esineet esimerkiksi museoissa, ajoneuvojen seuranta, varkauksien havainnointi ja raportointi omistajalle ja älykkäät varastot. Myös teollisuudessa voidaan hyödyntää langattomia anturiverkkoja. Langattomia antureita voidaan asentaa sellaisiin paikkoihin joihin perinteisiä antureita ei voitaisi asentaa ja näin haluttua toimintoa voidaan tarkemmin tarkkailla. Myös asennuskustannukset ja asennukseen kuuluva aika saattavat olla huomattavasti pienemmät langattomia antureita käytettäessä.

## 2.2 Protokollapino

Jokainen verkkotekniikka voidaan tavalla tai toisella mallintaa OSI (*Open Systems Interconnection*) -mallin avulla. Myös standardin IEEE 802.15.4 LR-WPAN (*Low Rate Wireless Personal Area Networks*) [23] tiedonsiirtoprotokolla koostuu OSI-mallin mukaisesti kerroksista. Seuraavaksi käydään läpi sekä OSI-malli että standardin IEEE 802.15.4 LR-WPAN tiedonsiirtoprotokollan rakenne.

### 2.2.1 OSI -malli

OSI-malli on kansainvälisen standardointiorganisaation, ISO:n (*International Standardization Organisation*) kehittämä malli, joka koostuu seitsemästä kerroksesta. Siinä jokainen kerros muodostaa oman itsenäisen kokonaisuuden ja jokaisella kerroksella on omat tehtävänsä ja kukin kerros käyttää yhtä alemman kerroksen palveluja ja tarjoaa palveluja yhtä kerrosta ylemmäs. Anttilan kirjassa [6] on esitetty Kuvan 1 mukaiset OSI-mallin seitsemän kerrosta.



Kuva 1. OSI-malli [6].

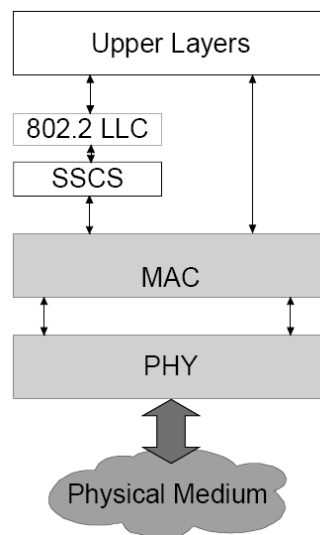
- 7) Sovelluskerroksen tehtävänä on tarjota verkkopalveluja sovelluksille, kuten tiedostojen avaaminen, sulkeminen, kirjoittaminen ja lukeminen.
- 6) Esitystapakerroksella määritetään millaisessa muodossa välitettävä data esitetään. Esimerkiksi onko kuva JPEG vai BMP muodossa.
- 5) Yhteysjaksokerroksen tehtävänä on sovellusten toimintojen koordinointi laitteiden välillä, kuten lähetyksen käynnistys ja lopetus. Yhteysjaksokerros huolehtii myös datan välittämisestä oikeassa järjestyksessä.
- 4) Kuljetuskerros huolehtii ylemmiltä kerroksilta vastaanotetun datan pilkkomisesta sopivan kokoiseen palasiin eli segmentteihin. Kuljetuskerros voi toimia joko yhteydellisenä tai yhteydettömänä. Jos se toimii yhteydellisenä, tällöin kommunikoivat laitteet muodostavat yhteyden ennen varsinaista lähetystä. Yhteydettömänä toimiessaan, data vain lähetetään verkkoon ja toivotaan, että se menee perille.
- 3) Verkkokerros pakkaa kuljetuskerrokselta saamansa datan verkkoon mahtuviin paketteihin ja välittää ne vastaanottajan verkkokerroksen osoitteen perusteella vastaanottajalle. Toisin sanoen tällä kerroksella tapahtuu reititys, joka on tyypillisesti riippumatonta alempien kerrosten tekniikoista.

2) Siirtoyhteyskerroksen tehtävänä on rakentaa kehys, jonka sisälle pakataan verkkokerrokselta saatu data. Siirtoyhteyskerros lisää kehykseen otsikot, joihin kuuluu mm. vastaanottajan ja lähettäjän siirtoyhteyskerroksen osoite. Siirtoyhteyskerros koostuu IEEE:n lähiverkkoihin määrittelemistä kahdesta osasta: LLC (*Link Layer Control*) ja MAC (*Medium Access Control*) -alikerroksesta. Siirtoyhteyskerros on hyvin voimakkaasti riippuvainen fyysisestä kerroksesta.

1) Fyysisellä kerroksella määritetään kullekin verkkotekniikalle ominaisia bittien välitykseen liittyviä asioita, kuten liittimien tyypit, käytetyt koodausmenetelmät ja jännitetasot.

### 2.2.2 IEEE 802.15.4 LR-WPAN

Standardi IEEE 802.15.4 LR-WPAN [23] määrittää langattomiin anturiverkkoihin soveltuvan tiedonsiirtoprotokollan fyysisen kerroksen ja MAC-alikerroksen. Kuvassa 2 on esitetty LR-WPAN laitteen protokollapino, joka on yksinkertaistettu versio OSI-mallin protokollapinosta.



Kuva 2. LR-WPAN laitteen protokollapino [23].

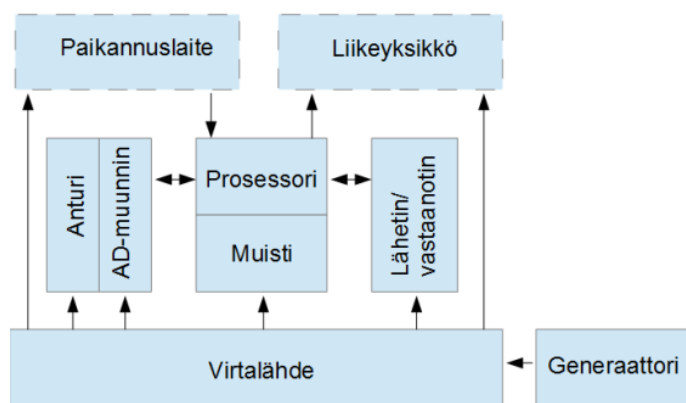
LR-WPAN laite sisältää fyysisen kerroksen, joka pitää sisällään lähetin vastaanotinyksikön radiotaajuuden ja sen matalatasoisen hallintamekanismin. MAC-alikerros toteuttaa jokaiselle tiedonsiirtotyypille pääsyn fyysiselle kanavalle. Ylemmät kerrokset pitävät

sisällään verkkokerroksen ja sovelluskerroksen. Verkkokerros mahdollistaa verkkokokoonpanon ja sen muokkaamisen sekä viestien reitityksen. Sovelluskerros taas mahdollistaa laitteen tarkoitetun toiminnan. Standardi IEEE 802.15.4 ei kuitenkaan sisällä ylempien kerrosten määrittelyitä. MAC-alikerrokselle pääsee ylemmiltä kerroksilta joko suoraan tai standardin IEEE 802.2 määrittelemän tyypin 1 LLC:n kautta ja läpi SSCS:n (*Service Specific Convergence Sublayer*).

### 2.3 Verkkokomponentit

Langaton anturiverkko koostuu vähintään yhdestä verkkokoordinaattorilaitteesta ja vähintään yhdestä verkkolaitteesta. Lähteen [20] mukaan jokainen verkko sisältää tarkalleen yhden verkkokoordinaattorin, jota kutsutaan PAN-koordinaattoriksi. Standardi IEEE 802.15.4 määrittää kahdenlaisia laitteita, FFD (*Full Function Device*) ja RFD (*Reduced Function Device*) -laitteen. FFD-laite tukee kaikkia MAC -alikerroksen tarjoamia palveluita ja se voi toimia verkossa PAN-koordinaattorina. RFD laite taas on yksinkertaisempi laite, joka tukee vain tiettyjä MAC-alikerroksen tarjoamia palveluja, eikä se voi toimia PAN-koordinaattorina. Käytännössä verkon kaikki laitteet ovat tyypillisesti samanlaisia ja anturisolmut toimivat joko FFD tai RFD moodissa riippuen siitä, mikä niiden tehtävä on verkossa.

Akyildiz ym. [4] ovat esittäneet artikkelissaan anturisolmun rakenteen. Sen mukaan anturisolmu koostuu neljästä peruskomponentista: anturista, prosessorista, lähetinvastaanotinyksiköstä ja akusta. Lisäksi anturisolmu voi sisältää sovelluskohtaisia lisäkomponentteja kuten paikannuslaitteen, energiaa tuottavan generaattorin ja liikeyksikön. Anturiyksikkö koostuu yleensä kahdesta aliyksiköstä: anturista ja analogia-digitaalimuuntimesta. AD-muunninta tarvitaan, koska anturiyksikkö tuottaa analogista signaalia, joka täytyy muuntaa prosessorille digitaaliseen muotoon. Anturisolmun rakenne on esitetty Kuvassa 3.



Kuva 3. Anturisolmun rakenne [4].

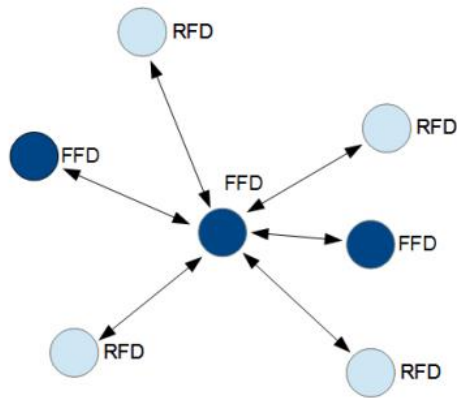
## 2.4 Verkkotopologiat

IEEE standardin 802.15.4 MAC-alikerros antaa sovellustuen kahden tyyppisen verkkotopologian rakentamiseen: tähtitopologian ja vertaisverkkotopologian. Nämä verkkotopologiat on esitetty lähteessä [20] ja tämän alaluvun teksti pohjautuu kyseiseen lähteeseen, ellei lähde ole erikseen mainittu. Tähtitopologiassa kaikki verkon laitteet kommunikoivat suoraan PAN-koordinaattorina toimivan FFD-laitteen kanssa. Vertaisverkkotopologiassa kaikki FFD-laitteet voivat kommunikoida toisen kuuluvuusalueellaan sijaitsevan FFD-laitteen kanssa. Vertaisverkkotopologia mahdollistaa monimutkaisempien, suurempien, itseorganisoituvien ja itsestään toipuvien ad hoc -tyyppisten verkkojen muodostamisen. Tämän tyyppisten verkkojen hallinta tapahtuu OSI-mallin verkkokerroksella eikä täten kuulu standardiin IEEE 802.15.4. MAC-alikerros vain toimittaa toiminnot, jotka vastaavat verkkokerrokset vaativat.

### 2.4.1 Tähtitopologia

Kuvassa 4 esitetty tähtitopologia on hyvä valinta sovelluksille, joilla on tarve peittää jokin tietty rajoitettu alue siten, että PAN-koordinaattori on muiden verkon laitteiden kuuluvuusalueella. Standardin 802.15.4 mukaan tähtitopologiassa PAN-koordinaattorina voi toimia vain FFD-laite ja muut verkon laitteet voivat olla joko FFD tai RFD -laitteita.





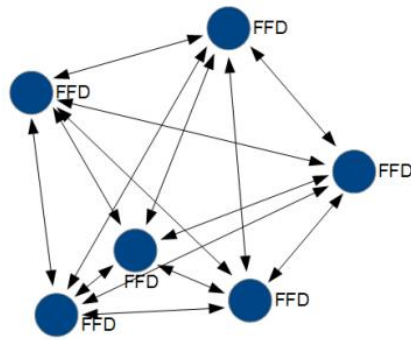
Kuva 4. Tähtitopologia [20].

Verkon muodostus tähtitopologiassa, kuten muissakin verkkotopologiatyypeissä, alkaa sillä että FFD aktivoituu ensimmäisen kerran ja siitä tulee PAN-koordinaattori. Jokainen tähtitopologiaverkko toimii itsenäisesti eikä häiritse muita verkkoja. Tämä on toteutettu siten, että PAN-koordinaattorin tunnuksiksi on valittu sellainen tunnus, jota ei sillä hetkellä käytetä yhdessäkään muussa verkossa, joka toimii kyseisen verkon radioetäisyydellä. Tämän jälkeen PAN-koordinaattori voi aloittaa beacon-sanomien lähettämisen sekä antamaan verkkoon pyrkiville laitteille lupia liittyä verkkoon. Tähtiverkkoon haluavien laitteiden pitää kuunnella PAN-koordinaattorin lähettämiä beacon-sanomia. Kun laite on vastaanottanut beacon-sanoman, lähettää se pyynnön liittyä verkkoon PAN-koordinaattorille, joka tämän jälkeen joko hyväksyy tai hylkää pyynnön.

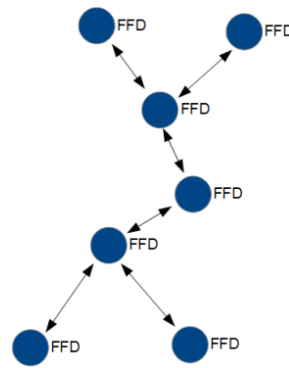
Koska tähtitopologiaverkot ovat yhden hypyn verkkoja, ne sopivat hyvin tilanteisiin, joissa lyhyt saantiviive on kriittinen vaatimus sovellukselle. Monen hypyn vertaisverkoissa saantiviive on usein pidempi kuin tähtitopologiaverkoissa. Toisin kuin joissain vertaisverkoissa, tähtitopologiaverkossa PAN-koordinaattori kuulee kaikkia verkkolaitteita ja ohjaa suoraan pääsyä jaetuille kanaville. Tämän takia reititys tähtitopologiaverkoissa katsotaan tapahtuvan ennemminkin MAC-kerroksella kuin verkkokerroksella osana vertaisverkon reititysalgoritmeja. Tämän idean pohjalta tähtitopologiaverkkojen reititys on määritelty IEEE:n standardissa 802.15.4. Tämän tasoisella reitityksellä on myös joitain toteutusetuja, koska yksinkertainen viestinvälitys ei tarvitse ylempää protokollakerroksia.

## 2.4.2 Vertaisverkkotopologia

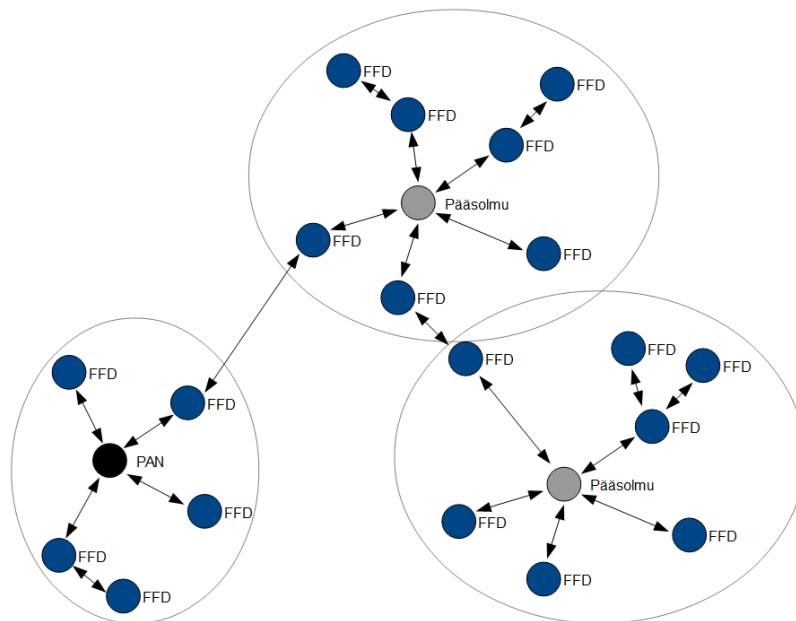
Vertaisverkkotopologia mahdollistaa jokaisen FFD-laitteen kommunikoinnin kuuluvalle alueelle olevien muiden FFD-laitteiden kanssa. Kuvassa 5 on esitetty tällainen yhden hypyn verkko. Jos FFD-laite onkin monen hypyn päässä ja kuuluvalle alueen ulkopuolella kuten Kuvassa 6, voidaan sille kuitenkin lähettää viestejä muiden FFD-laitteiden kautta. Tämä topologia mahdollistaa monimutkaisten verkkojen muodostamisen, kuten ad hoc -tyyppisten, itseorganisoituvien sekä itsestään toipuvien verkkojen muodostamisen. Tässä aluvuossa käsitellään tarkemmin Kuvan 7 klusteriverkkotopologia, koska tutkielma käsittelee klusterointia.



Kuva 5. Yhden hypyn verkko.



Kuva 6. Monen hypyn verkko.



Kuva 7. Klusteriverkkotopologia.

## **Klusteriverkkotopologia**

Johtuen anturisolmujen niukoista resursseista langattomissa anturiverkoissa on monesti tarpeen muodostaa anturisolmuista klustereita. Klusterointi mahdollistaa anturisolmujen hierarkisen rakenteen ja näin ollen mahdollistaa myös niukkojen resurssien tehokkaamman käytön. Klusteroinnilla pienennetään yksittäisten anturisolmujen energiankulutusta vähentämällä anturisolmujen tiedonvälitystä tai pidentämällä anturisolmujen lepoaikaa, jolloin anturisolmu kuluttaa hyvin vähän virtaa. Vähentämällä yksittäisten anturisolmujen energiankulutusta, pidennetään samalla myös verkon elinikää. Klusteripohjaisissa langattomissa anturiverkoissa anturisolmut on ryhmitelty joukoksi itsenäisiä klustereita. Jokaisella klusterilla on pääsolmu (*engl. Cluster-head*), jonka kautta klusterin jäsensolmut välittävät keräämänsä datan tiedonkeruusolmulle. Pääsolmun tehtäviä ovat mm. klusterisolmujen välinen koordinointi, kerätä klusterin dataa ja välittää kerätty data tiedonkeruusolmulle. [16]

Lähteessä [20] on käsitelty klusteriverkon verkonmuodostusta. Klusteriverkossa verkon ensimmäinen laite toimii PAN-koordinaattorina, kuten muissakin standardin IEEE 802.15.4 verkoissa. Klusteriverkoissa verkkolaitteiden välillä on isä-lapsi suhteita. Kun uusi laite pyytää lupaa liittyä verkkoon ja PAN-koordinaattori hyväksyy sen pyynnön, tulee siitä PAN-koordinaattorin lapsisolmu ja PAN-koordinaattorista uuden laitteen isäsolmu. Seuraavasta verkkolaitteesta voi tulla taas joko PAN-koordinaattorin tai toisen verkkolaitteen lapsisolmu. Verkkolaitteilla voi olla monta lapsisolmua, mutta vain yksi isäsolmu.

Verkon uusien laitteiden löytäminen on mahdollistettu käyttämällä beacon-sanomia. Uusi verkkolaite voi kuulla useita beacon-sanomia, sillä beacon-sanomia lähettävät kaikki verkon laitteet. PAN-koordinaattori kontrolloi klusteriverkon rakennetta sallimalla ja estämällä uusien laitteiden liittymisen verkkoon. Esimerkiksi PAN-koordinaattori voi estää usean hypyn päässä sijaitsevien laitteiden liittymisen verkkoon ja sallia verkkoon pelkästään yhden hypyn päässä sijaitsevia laitteita. Tällöin verkkoon muodostuu kooltaan pienempiä klustereita ja verkkoon lähetettyjen viestien saantiajat pysyvät lyhyinä.

Klusteriverkon yhtenä etuna on se, että siinä voidaan päivittää verkon tila tietyin väliajoin. Tämän ansiosta PAN-koordinaattori saa tiedon verkon hajonneista tiedonvälityslinkeistä ja kadonneista anturisolmuista. Yksi tapa, jolla verkon tilanpäivitys voidaan tehdä, on lähettää tilanpäivityspyyntöviesti kaikille laitteille, jotka eivät ole isälaitteita. Viestin saatuaan jokainen isälaite välittää viestin sen lapsilaitteille, jolloin verkon jokainen laite joko välittää tai vastaanottaa kyseisen viestin. Kun verkon laite, jolla ei ole lapsilaitteita, vastaanottaa tilanpäivityspyyntöviestin, vastaa se tilanpäivitysvastausviestillä PAN-koordinaattorille. Tilanpäivitysvastausviesti on hyvin lyhyt ja se pitää sisällään laitteen verkko-osoitteen. Tilanpäivitysvastausviestin PAN-koordinaattorille välittävät laitteet lukevat viestin ja saavat näin viestistä tietoa klusterin muista laitteista. Tämän jälkeen viestin välittävä laite lisää omat tietonsa viestiin ja välittää sen edelleen omalle isälaitteelle. Lopulta tilanpäivitysvastausviesti pitää sisällään jokaisen verkon laitteen reititystaulun. Lisäksi PAN-koordinaattori ja jokainen viestin välittävä laite on tietoinen myös mahdollisista vaihtoehtoisista reiteistä. Tämän ansiosta verkon luotettavuus paranee, sillä viesti voidaan välittää vaihtoehtoista reittiä, kun verkkolaite tai tiedonvälityslinkki on hajonnut.

### 3 Reititys ja viestintä langattomissa anturiverkoissa

Langattomien anturiverkkosovellusten tehtävänä on havainnoida ja kerätä erilaista tietoa ympäristöstään riippuen sovellustyypistä. Protokollapinon verkkokerroksen päätehtävänä on löytää reitti ja välittää data anturisolmulta tiedonkeruusolmulle tai tukiasemalle. Pienissä verkoissa anturisolmut voivat viestiä suoraan tiedonkeruusolmun kanssa. Yleensä langattomat anturiverkot kattavat ison maantieteellisen alueen, joten suora viestintä anturisolmun ja tiedonkeruusolmun välillä ei ole mahdollista. Juuri tämän takia anturisolmut eivät pelkästään lähetä ja vastaanota dataa, vaan myös välittävät sitä eteenpäin verkossa. Reittitietoja muihin anturisolmuihin ylläpidetään reititystauluissa ja reititysprotokollan tehtävänä on näiden taulujen rakentaminen ja ylläpitäminen. Viestintä langattomissa anturiverkoissa voidaan jakaa kahteen luokkaan: sovellusviestintään ja verkonhallintaan liittyvään viestintään. Havainnoidun datan välittäminen anturisolmulta tiedonkeruusolmulle on sovellusviestintää ja viestintä liittyen verkon muodostukseen, ylläpitoon ja optimointiin on verkonhallintaan liittyvää viestintää.

Langattomien anturiverkkojen reititys eroaa kiinteiden verkkojen reitityksestä ja on haastavaa johtuen verkon rajoitteista ja anturisolmujen hyvin rajallisista resursseista. Varsinkin energiankulutuksen pitäisi pysyä pienenä, jotta anturisolmujen elinikä olisi mahdollisimman pitkä. Langattomien anturiverkkojen toimintaympäristö aiheuttaa myös omat haasteensa reititykseen. Toimintaympäristö voi olla vaikeapääsyinen ja tällöin anturisolmujen sijoittaminen täytyy tehdä satunnaisesti eli esimerkiksi pudottamalla anturisolmut kohdealueelle. Tällöin osa alueesta voi tulla huonosti katettua tai jotkin anturisolmut eivät ole radiokantaman sisällä muista anturisolmuista tai tiedonkeruusolmusta. Usein langattomat anturiverkot suunnitellaankin siten, että ne ovat tiheään sijoiteltuja, jotta kyseisiltä ongelmilta vältyttäisiin. Langattomissa anturiverkoissa verkon topologia muuttuu usein anturisolmujen vikaantuessa tai liikkeessa ja uusien anturisolmujen liittyessä verkkoon. Topologiamuutokset aiheuttavat muutoksia reititystauluihin ja isoissa verkoissa näitä muutoksia voi tulla usein. Myös sovellus asettaa reititykselle omat vaatimuksensa. Esimerkiksi tieto mitatusta suureesta tai havainnoidusta tapahtumasta pitää olla tiedonkeruusolmulla ajoissa ja olla mahdollisimman tarkka.

### 3.1 Reitityksen haasteet langattomissa anturiverkoissa

Reitityksen haasteita langattomissa anturiverkoissa on käsitelty kattavasti Zhengin ja Jamalipourin kirjassa [50], jonka mukaan reititys langattomissa anturiverkoissa on haastavaa johtuen verkon rajoitteista ja eroaa monella tapaa kiinteiden verkkojen reitityksestä. Langattomien anturiverkkojen resurssit kuten energia, radiokantama, kaistanleveys, prosessointiteho ja tiedon varastointikyky ovat hyvin rajalliset. Ratkaisevin näistä on kuitenkin energia, sillä se määrittää anturisolmun eliniän. Varsinkin vaikeapääsyisessä tai vihamielisessä ympäristössä, kuten taistelukentällä, on mahdotonta päästä anturisolmujen luo ja ladata tai vaihtaa akkuja. Anturisolmu vikaantuu eikä toimi oikein, jos sen energiataso laskee tietyn raja-arvon alapuolelle. Vikaantunut anturisolmu vaikuttaa suuresti myös verkon suorituskykyyn. Tämän takia anturisolmuille suunniteltujen algoritmien tulisi olla mahdollisimman energiatehokkaita, jotta anturisolmujen ja samalla koko verkon elinikä pitenisi. Toinen reitityksen haaste on anturisolmujen paikkatietojen hallinta. Suurin osa reititysprotokollista olettaa, että anturisolmut ovat varustettu joko GPS (*Global Positioning System*) -vastaanottimella tai käyttävät jotain muuta paikannustekniikkaa. Käyttämällä GPS-paikannusta saadaan anturisolmuille tarkat paikkatiedot, mutta toisaalta ei ole kustannustehokasta varustaa jokaista anturisolmua GPS-vastaanottimella, sillä anturisolmuja voi verkossa olla todella paljon. Käyttämällä jotain muuta paikannusmenetelmää saattaa anturisolmujen paikkatiedoissa esiintyä epätarkkuutta.

Kuten aiemmin mainittiin, langattoman anturiverkon toimintaympäristö voi olla vaikeapääsyinen, joten uusien anturisolmujen käyttöönotto ja vanhojen anturisolmujen korvaaminen tai niiden akkujen lataaminen voi olla vaikeaa. Näin ollen deterministinen anturisolmujen käyttöönotto ei ole aina mahdollista, vaikka se auttaisi kattamaan kentän asianmukaisesti ja minimoisi tarvittavien anturisolmujen määrän saavuttaakseen sovelluksen määrittämät erityiset kattavuusvaatimukset. Todellisuudessa sovellus voi vaatia osittaisen, täyden tai ylimääräisen kattavuuden. Osittaisessa kattavuudessa vain tietty prosenttiosuus kentästä on katettu. Täydessä kattavuudessa koko kenttä on katettu ja ylimääräisessä kattavuudessa, jokainen kentän paikka on katettu usealla anturisolmulla samanaikaisesti. Tapauksissa, joissa deterministinen anturisolmujen sijoittaminen ei ole mahdollista, jää ainoaksi vaihtoehdoksi satunnainen anturisolmujen sijoittaminen.

Satunnainen sijoittaminen ei kuitenkaan takaa, että sovelluksen asettamat kattavuusvaatimukset täyttyvät. Tällöin jotkin kentän alueet voivat tulla katettua huonosti tai ei lainkaan ja johtaa ongelmaan nimeltä kattavuusaukko (*engl. coverage hole*). Satunnaisessa sijoittamisessa ei voida myöskään taata, että kaikki anturisolmut olisivat radiokantaman sisällä muista anturisolmuista tai tiedonkeruusolmusta. Tämä tilanne johtaa toiseen ongelmaan, nimeltään liitettävyyssaukko (*engl. connectivity hole*). Juuri näiden kahden syyn takia useimmat langattomat anturiverkot suunnitellaan siten, että anturisolmut ovat tiheään sijoiteltuja. Oletamus, että anturisolmut ovat tiheään sijoiteltuja, on yksi yleisimpiä oletamuksia langattomien anturiverkkojen reitityksen suunnittelussa.

Langattomissa anturiverkoissa verkon topologia muuttuu usein uusien anturisolmujen liittyessä verkkoon ja olemassa olevien anturisolmujen poistuessa verkosta. Kun uusi anturisolmu päättää liittyä verkkoon, täytyy joidenkin anturisolmujen päivittää myös reittitietonsa. Myös anturisolmun poistuessa verkosta, vikaantuneen anturisolmun naapureiden täytyy päivittää reittitietonsa. Anturisolmut voivat olla myös liikkuvia ja liikkeudessaan kentällä muuttuu samalla myös verkon topologia. Kaikki verkon topologiamuutokset vaikuttavat siis myös anturisolmujen reititystauluihin. Monissa tapauksissa on välttämätöntä lisätä verkon anturisolmujen määrää, jotta energiansa loppuun kuluttaneiden tai vikaantuneiden anturisolmujen poistuessa verkosta, voidaan taata anturikentän kattavuus- ja liitettävyyssvaatimukset. Yhteys anturisolmusta tiedonkeruusolmuun on hyvin tärkeää. Jos havaittu data ei voi saavuttaa tiedonkeruusolmua tai ei ole olemassa reittiä anturisolmujen ja tiedonkeruusolmun välillä, kattavuuden ylläpitämisestä tulee merkityksetöntä. Verkon moitteettoman toiminnan kannalta on tärkeää, että yhteys anturisolmuilta tiedonkeruusolmulle on kunnossa. Myös verkon skaalautuvuus on haaste langattomissa anturiverkoissa, sillä anturisolmuja voi olla verkossa satoja, tuhansia tai jopa enemmän. Reititysprotokollan täytyy toimia, vaikka anturisolmujen määrä olisi suuri. Anturisolmujen ominaisuudet voivat myös vaihdella ja anturisolmujen väliset linkit eivät aina ole symmetrisiä. Reititysprotokollan pitää huomioida myös tällaiset tilanteet, joissa anturisolmujen välinen kommunikointi voi tapahtua vain yhteen suuntaan.

Reitityksessä täytyy huomioida myös sovelluksen asettamat vaatimukset. Useissa sovelluksissa tieto mitatusta suureesta tai havainnoidusta tapahtumasta pitää olla tiedonkeruusolmulla ajoissa ja olla mahdollisimman tarkka, jotta tiedonkeruusolmu voi tehdä parempia päätöksiä. Esimerkiksi tunkeilijan havainnointi- ja seurantasovelluksessa useiden anturisolmujen täytyy samanaikaisesti kerätä tietoa tunkeutujasta sekä seurata tarkasti sen liikkeitä. Reititysprotokollan täytyy varmistaa datan perille meno sekä tiedon tarkkuus huolimatta verkossa tapahtuvista muutoksista, jotta tiedonkeruusolmu voi kerätä tarpeellisen ja oikean tiedon ilmiöstä ajoissa. Verkon vikasietoisuuden aste riippuu käytettävästä sovelluksesta ja siksi reititysprotokollan täytyy olla vikasietoinen.

### 3.2 Viestintä

Tilak ym. [46] jakavat langattomien anturiverkkojen viestinnän kahteen luokkaan: sovellusviestintään ja verkonhallintaan liittyvään viestintään. Verkkoprotokollan on tuettava näitä molempia viestintätyyppiejä. Sovellusviestinnällä tarkoitetaan verkkoliikennettä, joka liittyy anturisolmun havaitseman ilmiön tai mittaaman suureen tiedon välittämiseen tiedonkeruusolmulle. Sovellusviestintä pitää sisällään kaksi eri toimintamallia. Yhteistoimintamallissa anturisolmut kommunikoivat toisten anturisolmujen kanssa selvittääkseen havainnoijan kiinnostuksen kohteen sekä tiedon mitä dataa pitää välittää. Verkon sisäinen datan prosessointi on esimerkki tällaisesta toimintamallista. Esimerkiksi SPIN (*Sensor Protocols for Information via Negotiation*) -protokolla vähentää päällekkäisen ja tarpeettoman tiedon välittämistä käyttämällä neuvotteluprosessia ennen varsinaista datan lähetystä. Ei-yhteistoimintamallissa anturisolmut eivät tee yhteistyötä muiden anturisolmujen kanssa liittyen tiedon välitykseen.

Verkonhallintaan liittyvällä viestinnällä tarkoitetaan viestintää, jota tarvitaan verkon muodostukseen, ylläpitoon ja optimointiin. Langattomat anturiverkot ovat ad hoc tyyppisiä eli anturisolmut voivat liikkua, niitä voi liittyä verkkoon lisää tai ne voivat vikaantua. Anturisolmujen täytyy olla kykeneviä löytämään reitit muihin kiinnostaviin anturisolmuihin sekä tiedonkeruusolmuun huolimatta verkossa tapahtuvista muutoksista. Verkonhallintaan liittyvää viestintää tarvitaankin pitämään verkko toimintakuntoisena, varmistamaan toiminta dynaamisessa ympäristössä ja optimoimaan suorituskykyä.

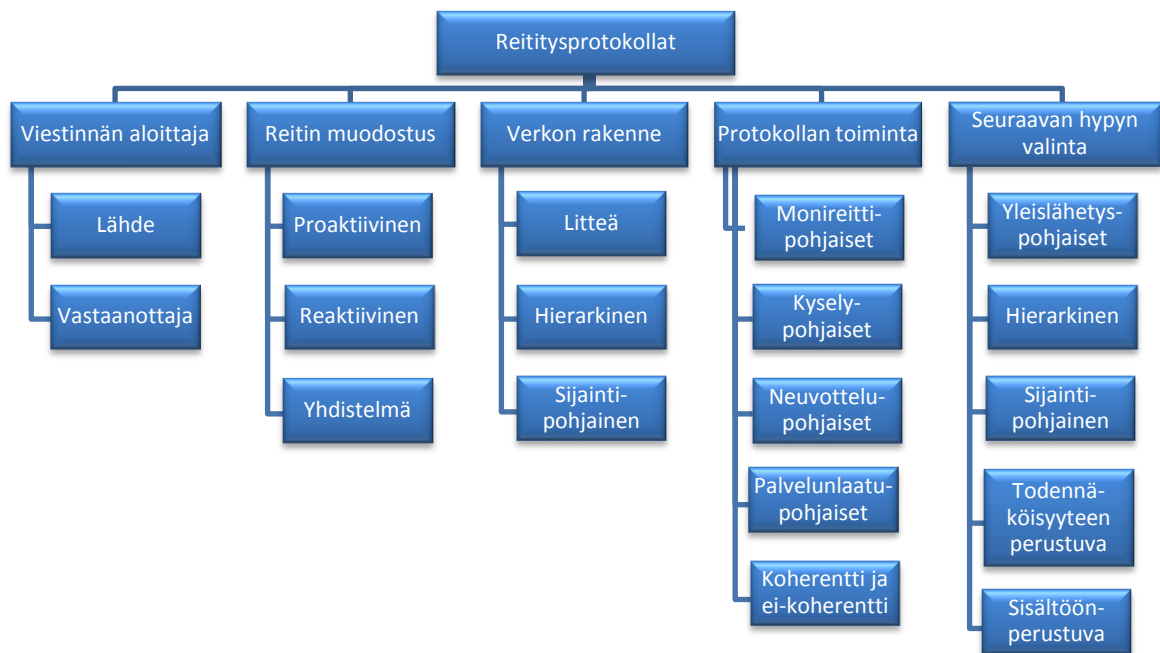


Sovelluksen toiminta ja vaatimukset vaikuttavat suuresti verkonhallintaan liittyvään viestintään, sillä verkon täytyy osata muodostua itsenäisesti siten, että se parhaiten tyydyttää sovelluksen vaatimuksia. Verkonhallintaan liittyvä viestintä edustaa verkkoprotokollan overhead:a eli ylimääräistä verkkoliikennettä, joka ei ole varsinaista dataa. On tärkeää minimoida verkon overhead, mutta samalla varmistaa että verkko tukee tehokasta sovellusviestintää.

Myös Ács ja Buttyán [2] käsittelevät artikkelissaan langattomien anturiverkkojen viestintää. Sen mukaan reititysprotokollat tukevat viestintää anturisolmulta anturisolmulle, tiedonkeruusolmulta anturisolmulle sekä anturisolmulta tiedonkeruusolmulle. Ensisijaisesti ad hoc -verkoille suunnitellut reititysprotokollat tukevat anturisolmujen välistä viestintää ja niitä voidaan käyttää myös langattomissa anturiverkoissa. Viestintä tiedonkeruusolmulta anturisolmulle on tarpeen, jotta tiedonkeruusolmun datapyyntöjä voidaan välittää anturisolmuille. Tyypillisesti viestintä tapahtuu siten, että tiedonkeruusolmu lähettää pyynnön verkkoon ja mikä tahansa anturisolmu, jolla on pyydetty data, voi vastata pyyntöön. Jos jotkin anturisolmut ovat yksilöityjä eli niillä on esimerkiksi tietty id tai paikkatieto, voi tiedonkeruusolmu lähettää pyynnön yksittäislähetystenä (*engl. unicast*) tietyille anturisolmille tai monilähetystenä (*engl. multicast*) tietyille anturisolmuille. Viestintä anturisolmulta tiedonkeruusolmulle on tarpeen, jotta anturisolmut voivat vastata tiedonkeruusolmun lähettämiin pyyntöihin. Tyypillisesti viestintä tapahtuu käänteisenä monilähetystenä eli jokainen anturisolmu voi lähettää viestin tiedonkeruusolmulle. Viestintä anturisolmulta tiedonkeruusolmulle voi tapahtua myös yksittäislähetystenä, jos verkossa on esimerkiksi monta tiedonkeruusolmua tai vain yksi anturisolmu on vastuussa datan mittauksesta ja välittämisestä.

### 3.3 Reititysprotokollat

Reititysprotokollat voidaan luokitella usealla eri tavalla. Hussein [22] luokittelee reititysprotokollat viestinnän aloittajan, reitin muodostuksen, verkon rakenteen, protokollan toiminnan sekä seuraavan hypyn valinnan mukaan. Tämä jaottelu on esitetty Kuvassa 8.



Kuva 8. Reititysprotokollien luokittelu [22].

### 3.3.1 Viestinnän aloittaja

Reititysprotokollat voidaan luokitella sen mukaan, kumpi aloittaa viestinnän lähde vai vastaanottaja. Jos viestinnän aloittajana toimii lähde, tällöin anturisolmu lähettää datan tukiasemalle tai tiedonkeruusolmulle heti uuden datan mittauksen/havainnoinnin jälkeen. Tällaiset protokollat käyttävät joko aika- tai tapahtumaohjattua tiedon raportointia. Jos vastaanottaja toimii viestinnän aloittajana, tällöin anturisolmut lähettävät dataa vastaan saatuun pyyntöviestin vastaanottajalta eli protokolla käyttää kyselyohjattua tiedon raportointia. [5]

### 3.3.2 Reitin muodostus

Reititysprotokollat voidaan jakaa Pantazisin ym. [40] tavoin proaktiivisiin, reaktiivisiin ja yhdistelmiin riippuen siitä milloin reittitieto lasketaan. Proaktiivisessa reititysprotokollassa jokainen anturisolmu ylläpitää yhtä tai useampaa reititystaulua, jotka pitävät sisällään reittitiedon muihin anturisolmuihin. Reititystaulut päivitetään aina tietyin väliajoin tai kun verkon topologia muuttuu. Proaktiiviset reititysprotokollat muodostavat näin ollen tarvittavat reititystaulut jo ennen kuin niitä todella tarvitaan. Suurin hyöty tästä on se, että

reitit ovat heti saatavilla reititystaulusta eikä niiden muodostus aiheuta viivettä kuten reaktiivisissa reititysprotokollissa. Tämän johdosta proaktiivinen reititys sopii paremmin reaaliaikaiseen viestintään. Toisaalta tietyn väliajoin suoritettu reititystaulujen päivitys aiheuttaa ylimääräistä viestintää ja kuormittaa näin verkkoa. Anturiverkot koostuvat yleensä suuresta määrästä anturisolmuja, joten proaktiivinen reititys vaatii suuria reititystauluja ja näin ollen myös paljon muistia.

Toisin kuin proaktiivisissa reititysprotokollissa, reaktiivisissa protokollissa reitin etsintä suoritetaan vasta kun sitä tarvitaan. Reaktiivinen reititys ei vaadi jatkuvaa päivitysviestien lähetystä kuten proaktiivinen reititys, joten se säästää verkon kaistaa ja anturisolmujen energiaa. [40] Lähdenoodi, joka tietää joko vastaanottajan id:n tai osoitteen, käynnistää reitin etsintäprosessin. Prosessi päättyy kun vähintään yksi reitti on löytynyt tai kun kaikki mahdolliset reitit on tutkittu. Reittiä ylläpidetään kunnes se joko särkyi tai lähde ei sitä enää tarvitse. [13]

Myös näiden yhdistelmät ovat mahdollisia. Esimerkiksi klusterin sisäinen reititys tehdään proaktiivisesti ja klusteiden välinen reititys reaktiivisesti. Näin ollen reititystaulut eivät kasva kovin suuriksi ja linkkivika tai anturisolmun katoaminen vaikuttaa vain kyseiseen klusteriin.

### **3.3.3 Verkon rakenne**

Verkon rakenteella on merkittävä vaikutus reititysprotokollan toimintaan langattomissa anturiverkoissa. Al-Karaki ja Kamal [5] luokittelevat reititysprotokollat verkon rakenteen mukaan litteisiin, hierarkisiin ja sijaintipohjaisiin.

Litteissä reititysprotokollissa verkon rakenne on homogeeninen eli jokaisella anturisolmulla on yhtäläiset datan prosessointi- ja kommunikointikyvyt sekä samanlainen rooli datapakettien välittäjänä. Anturisolmuille ei ole mahdollista antaa globaalia tunnistetietoa, johtuen verkon anturisolmujen suuresta määrästä. Tämä on johtanut datakeskeiseen reititykseen, missä tukiasema lähettää kyselyitä tietyille alueille ja odottaa tietoa valitulla alueella sijaitsevilta anturisolmuilta. Koska haluttua dataa pyydetään

kyselyillä, on välttämätöntä pystyä erottelemaan havainnoitavat tapahtumat attribuuttiarvopareilla.

Dargie ja Poellabauer ovat käsitelleet kirjassa [13] sekä hierarkisia että sijaintipohjaisia reititysprotokollia. Hierarkisia tai klusteripohjaisia reititysprotokollia käytetään verkoissa, joissa vaaditaan skaalautuvuutta sekä tehokasta viestintää. Hierarkisissa reititysprotokollissa anturisolmut ryhmitellään klustereihin. Jokaisella klusterilla on pääsolmu, jonka tehtävänä on välittää klusterin jäsensolmujen havainnoimaa dataa tiedonkeruusolmulle. Pääsolmut voivat olla tehokkaampia ja vähemmän energiarajoitteisia kuin klusterin jäsensolmut, mutta ne voivat myös olla samanlaisia kuin muut anturisolmut. Hierarkinen reititys on tehokas tapa vähentää energiankulutusta klusterin sisällä sekä vähentää tukiasemalle lähetettävien viestien lukumäärää käyttämällä datan kokoamista ja fuusiota [5]. Klusteroinnilla saadaan vähennettyä radiotien ruuhkaa ja näin ollen myös pakettien törmäyksiä. Se saattaa helpottaa myös reititysprosessia, mutta jos pääsolmut eivät voi kommunikoida suoraan tiedonkeruusolmun kanssa, täytyy reititysprotokollan muodostaa myös reitit pääsolmuista tiedonkeruusolmuille. Tällöin reittien pituudet voivat kasvaa verrattuna litteään reititysprotokollaan. Klusterointia käsitellään tarkemmin luvussa 4.

Sijaintipohjaisissa reititysprotokollissa anturisolmut ovat kykeneviä määrittämään sijaintinsa käyttäen erilaisia paikannusmenetelmiä. Sen sijaan, että reititys perustuisi verkon topologiaan, perustuvat reitityspäätökset paikkatietoon välitettäessä dataa kohti määränpäättä. Etäisyys naapurianturisolmuihin voidaan arvioida signaalinvahvuuden perusteella ja naapurianturisolmujen suhteelliset koordinaatit saadaan selville vaihtamalla paikkatietoja naapurianturisolmujen kesken [5]. Sijaintipohjaiset reititysprotokollat voidaan jakaa vielä lähetystavan mukaan. Yksittäislähetyksessä data välitetään paikkatietoon perustuen yhteen ainoaan määränpäähän. Tällöin lähettäjän täytyy tietää sekä oma että määränpään paikkatieto. Paikkatieto voidaan saada joko lähettämällä kysely määränpäälle käyttäen esimerkiksi tulvivaa reititystä tai käyttämällä paikkatiedon välittäjää. Tulvivassa reitityksessä anturisolmu välittää pyynnön omille naapurisolmuilleen, jotka välittävät pyynnön taas omille naapurisolmuilleen. Paikkatiedon välittäjä sen sijaan on palvelu, joka palauttaa anturisolmun tunnusta vastaavan

paikkatiedon. Monilähetyksessä sama data välitetään useaan määränpään. Tyypillisesti anturisolmun paikkatieto on tärkeämpi tieto kuin anturisolmun identiteetti. Käyttämällä paikkatietoon perustuvaa lähetystä (*engl. geocasting*), voidaan vähentää merkittävästi sekä käytettävää kaistanleveyttä että energian kulutusta. Tällöin kyselyt välitetään pelkästään kiinnostavalle maantieteelliselle alueelle eikä tulvimalla koko verkkoon.

### **3.3.4 Luokittelu protokollan toiminnan mukaan**

Protokollan toiminnan mukaan reititysprotokollat voidaan luokitella artikkelin [5] mukaisesti monireitti-, kysely-, neuvottelu- ja palvelunlaaturajaisiin sekä koherentteihin ja ei-koherentteihin. Jotkin reititysprotokollat voivat toimintansa perusteella kuulua useampaan edellä mainittuun luokkaan.

Langattoman anturiverkon reitityksen suunnittelussa tulisi verkon skaalautuvuuden ja energiatehokkuuden lisäksi miettiä myös tiedonsiirron luotettavuutta sekä vikasietoisuutta, jotta verkko olisi toimintakykyinen myös anturin tai linkin vikaantumisen jälkeenkin. Käyttämällä monireittipohjaista reititystä, jossa lähdesolmun ja tiedonkeruusolmun välillä on vähintään yksi vaihtoehtoinen reitti, voidaan parantaa verkon vikasietoisuutta. Tällöin esimerkiksi linkin vioittuessa, voidaan kohdealueelta välittää dataa tiedonkeruusolmulle käyttämällä vaihtoehtoisia reittejä. [50] Käyttämällä vaihtoehtoisia reittejä samanaikaisesti, voidaan parantaa myös tiedonsiirron luotettavuutta. Tähän on Radin ym. [42] mukaan kaksi eri lähestymistapaa. Ensimmäisessä lähetetään alkuperäisen datapakettin kopio useaa reittiä pitkin. Tiedon siirto on onnistunut, jos yksikin datapaketti tulee perille. Toisessa lähestymistavassa jokainen lähdenoodi lisää datapakettiin virheen havaitsemiseen sekä korjaukseen tarvittavaa lisätietoa ja lähettää sen useaa eri reittiä määränpään. Näin ollen, jos joidenkin datapakettien välitys epäonnistuu, voidaan luotettava tiedonsiirto kuitenkin taata rekonstruoimalla puuttuvat datapaketit tiedonkeruusolmulle onnistuneesti välitetyistä datapaketeista.

Kyselypohjaiset reititysprotokollat toimivat reaktiivisesti. Niissä tiedonkeruusolmu levittää verkkoon kyselyitä tapahtumasta tai haluamastaan tiedosta. Esimerkiksi mikä on tietyn alueen lämpötila. Kyseisen alueen anturisolmut, joilta haluttu tieto löytyy, lähettävät lämpötilatiedon kyselyn lähettäjälle. Yleensä nämä kyselyt kuvataan joko luonnollisella

kielellä tai korkean tason kyselykielellä. Kyselyt pitävät sisällään kuvauksen mitä tietoa halutaan vastaanottaa. Datan ominaisuuksille täytyy käyttää attribuuttipohjaista nimeämistä, jotta tapahtuma tai haluttu tieto pystytään tunnistamaan. Kyselypohjaisissa reititysprotokollissa jokaisella anturisolmulla on muistissaan taulu, joka pitää sisällään vastaanotettavat kyselyt sekä tiedon mitä dataa pitää missäkin kyselyssä palauttaa. [5] Ohjattu leviäminen (*engl. directed diffusion*) [24] on esimerkki tämän tyyppisestä reititysprotokollasta. Siinä tiedonkeruusolmu lähettää intressiviestejä anturisolmuille. Intressiviesti sisältää havainnoitavan tapahtuman kuvauksen eli tiedon mistä tapahtumasta tai tiedosta viestin lähettäjä on kiinnostunut. Intressiviestin kulkiessa verkon läpi, määritetään jokaiselle lähdesolmun ja tiedonkeruusolmun väliselle polulle gradientti. Kun lähdesolmulla on intressiviestiin liittyvää dataa, lähetetään se gradienttipolkuja pitkin kyselyn lähettäjälle.

Neuvottelupohjaisten reititysprotokollien ideana on vähentää päällekkäistä tietoa ja estää tarpeettoman tiedon välittäminen seuraavalle anturisolmulle tai tiedonkeruusolmulle käyttämällä neuvotteluprosessia ennen varsinaista datan lähetystä [5]. Esimerkki tällaisesta reititysprotokollasta on SPIN [30].

SPIN sisältää kaksi ominaisuutta: neuvottelu ja resurssien mukauttaminen, joiden avulla voidaan välttää tulvivan reitityksen klassiset ongelmat. Tulvivassa reitityksessä anturisolmu lähettää datan aina jokaiselle naapurisolmulleen riippumatta siitä onko se jo saanut kyseisen datan, joten anturisolmu voi saada useamman kopion samasta datasta. Anturisolmujen toimintasäteet voivat sijoittua osittain päällekkäin, jolloin nämä anturisolmut voivat havaita saman tapahtuman ja näin ollen sama data lähetetään useammasta anturisolmusta. Tulviva reititys tuhlaa myös energiaa eikä huomioi toiminnassaan saatavilla olevia resursseja. SPIN-protokolla määrittää 3 viestityyppiä, joiden avulla viestintä anturisolmujen kesken tapahtuu. ADV-viestillä anturisolmu voi mainostaa naapurisolmuilleen keräämäänsä dataa. REQ-viestillä pyydetään taas lähettämään haluttua dataa. ADV- ja REQ-viestit sisältävät pelkästään meta-dataa ja ovat näin ollen pienempiä kuin DATA-viestit. Meta-datan esitysmuoto on sovelluskohtainen ja se kuvaa yksiselitteisesti anturisolmun keräämän/pyytämän datan. DATA-viestillä välitetään varsinainen data. SPIN sovellukset ovat tietoisia resursseistaan ja mukautuvat

kulloisenkin resurssitilanteen mukaan. Ne voivat pollata järjestelmän resursseja selvittääkseen kuinka paljon energiaa on käytettävissä. Sovellus voi myös laskea, paljonko energiaa kuluu laskutoimituksiin sekä datan lähettämiseen ja vastaanottamiseen. Näiden tietojen avulla anturisolmut voivat tehdä päätöksiä käyttääkseen resurssejaan tehokkaasti. SPIN ei kuitenkaan määritä mitään tiettyä energianhallintamenettelytapaa.

Palvelunlaatupohjaisissa reititysprotokollissa verkon täytyy tasapainoilla energian kulutuksen ja datan laadun välillä. Erityisesti verkon on täytettävä tietyt palvelunlaadulle asetetut rajat (esim. viive, energiankulutus tai kaistanleveys) välittäessään dataa tukiasemalle. Esimerkiksi aikakriittisissä sovelluksissa datan täytyy saavuttaa tiedonkeruusolmu tietyssä ajassa. [5]

SAR (*Sequential Assignment Routing*) [43] on esimerkki sekä palvelunlaatupohjaisesta että monireittipohjaisesta reititysprotokollasta. Protokolla luo useita puita, joiden juurisolmu sijaitsee yhden hypyn päässä tiedonkeruusolmusta. Näin saadaan muodostettua useita reittejä jokaiselta anturisolmulta tiedonkeruusolmulle. Puut kasvavat tiedonkeruusolmulta pois päin välttämällä anturisolmuja, joiden energiavarannot ovat pienet tai palvelunlaatu on matala (esimerkiksi viive on suuri). Anturisolmut määrittävät jokaiselle polulle kaksi parametria: energiatason sekä additiivisen palvelunlaatuarvon. Energiataso ilmaistaa pakettien lukumäärällä, joka voidaan lähettää ilman energian ehtymistä olettaen että polku on yksinomaan kyseisen anturisolmun käytössä. Polun palvelunlaatu ilmaistaa additiivisella arvolla, jossa suurempi arvo merkitsee matalampaa palvelunlaatua. Jokainen polun anturisolmu lisää energiankulutusta sekä viivettä ja näin ollen kasvattaa myös additiivista palvelunlaatuarvoa. Anturisolmu laskee polun palvelunlaatuarvon ja datapaketin prioriteettitason perusteella painotetun palvelunlaatuarvon jokaiselle lähetettävälle datapaketille. Korkean prioriteettitason datapaketit lähetetään pitkin polkuja, joiden palvelunlaatu on korkea. SAR-protokolla on vikasietoinen ja se toipuu nopeasti anturisolmujen vikaantuessa, kuten muutkin monireittipohjaiset protokollat, mutta isoissa verkoissa puiden muodostus ja ylläpito vaatii paljon verkon resursseja [13].

Datan prosessointi on tärkeä osa langattomien anturiverkkojen toimintaa ja tämän takia myös reititysprotokollat käyttävät erilaisia datan prosessointitekniikoita. Al-karaki ja

Kamal [5] esittelevät kaksi langattomissa anturiverkoissa käytössä olevaa datan prosessointitekniikkaa: koherentti ja ei-koherentti datan prosessointi. Ei-koherentissa prosessoinnissa anturisolmut prosessoivat raakadatan paikallisesti ennen kuin välittävät sen muille anturisolmuille jatkoprosessoitavaksi. Anturisolmuja, jotka suorittavat datan jatkoprosessoinnin, kutsutaan datan kokoajiksi. Koherentissa prosessoinnissa data välitetään datan kokoajille minimaalisen prosessoinnin jälkeen. Minimaalinen prosessointi pitää tyypillisesti sisällään aikaleiman lisäämisen dataan ja duplikaattidatan poistamisen. Energiatehokkaaseen reititykseen valitaan normaalisti koherentti datan prosessointi. Ei-koherentissa reitityksessä dataliikenne on suhteellisen pieni. Koherentissa reitityksessä dataa sen sijaan liikkuu enemmän, joten energiätehokkuus täytyy saavuttaa polkujen optimoinnilla. Ei-koherentissa datanprosessoinnissa on kolme vaihetta. Vaiheessa 1, kohde havaitaan, sen tiedot kerätään ja esikäsitellään. Vaiheessa 2, ennen kuin anturisolmu voi lähettää datan eteenpäin jatkoprosessointiin, täytyy sen ilmoittaa aikeistaan kaikille naapurisolmuilleen. Tämä tulisi tehdä mahdollisimman aikaisessa vaiheessa, jotta jokaisella anturisolmulla olisi paikallinen ymmärrys verkon rakenteesta. Vaiheessa 3 valitaan keskussolmu. Keskussolmulla tulee olla riittävästi energiaa ja laskentakykyä, koska sen tehtävänä on suorittaa vaativampaa datan prosessointia.

### **3.3.5 Luokittelu seuraavan hypyn valinnan mukaan**

Seuraavan hypyn valinnalla on tärkeä rooli monen hypyn verkkojen reitityksessä. Hussein [22] jakaa reititysprotokollat seuraavan hypyn valinnan mukaan yleislähetyspohjaisiin, hierarkisiin, sijaintipohjaisiin, todennäköisyyteen perustuviin ja sisältöön perustuviin.

Yleislähetyspohjaiset reititysprotokollat toimivat hyvin suoraviivaisesti. Jokainen anturisolmu päättää itsenäisesti välittääkö se viestiä eteenpäin vai ei. Jos anturisolmu päättää välittää viestin, se yksinkertaisesti lähettää sen yleislähetysviestinä jokaiselle naapurisolmulleen.

Hierarkisissa reititysprotokollissa klusterin anturisolmut välittävät viestejä ylemmän hierarkiatason anturisolmulle eli klusterin pääsolmulle. Pääsolmu suorittaa datan kokoamisen ja välittää tiedonkeruusolmulle pelkästään kootun datan. Klusterilla tarkoitetaan joukkoa anturisolmuja, jotka välittävät viestejä samalle pääsolmulle. Datan



kokoaminen vähentää verkon liikennettä ja näin ollen myös energian kulutusta. Hierarkiset reititysprotokollat pidentävät verkon elinikää ja ovat myös hyvin skaalautuvia. Hierarkinen reititys on pääosin kaksikerroksinen, jossa ensimmäisellä kerroksella valitaan klusterin pääsolmu ja toisella kerroksella suoritetaan reititys. Klusterointia ja datan kokoamista käsitellään tarkemmin luvussa 4.

Sijaintipohjaiset reititysprotokollat valitsevat seuraavan hypyn kohti määränpäättä perustuen naapurisolmujen ja määränpään sijaintitietoon. Määränpään sijainti voi olla joko alueen keskipiste tai tietyn anturisolmun tarkka sijainti. Sijaintipohjaiset reititysprotokollat voivat vähentää tulvivan reitityksen aiheuttamaa ylimääräistä verkkoliikennettä, mutta toisaalta naapurisolmujen sijaintitietojen laskeminen lisää verkon liikennettä ja energiankulutusta. Paikallisen minimin ongelma on yleinen kaikille hajautetuille sijaintipohjaisille reititysprotokollille. On mahdollista, että kaikki naapurisolmut ovat kauempana määränpäästä kuin anturisolmu itse. Jokainen protokolla käyttää eri reititystekniikkaa tämän ongelman kiertämiseksi.

Osa reititysprotokollista valitsee seuraavan hypyn perustuen todennäköisyyteen. Nämä protokollat olettavat, että kaikki anturisolmut ovat homogeenisiä ja satunnaisesti sijoiteltuja. Tällaisissa protokollissa anturisolmut valitsevat satunnaisesti seuraavan hypyn eli naapurisolmun, jolle välittävät kulloisenkin viestin. Tietyn naapurisolmun todennäköisyys tulla valituksi seuraavaksi hypyksi on kääntäen verrannollinen sen kustannuksiin.

Jotkin reititysprotokollat taas valitsevat seuraavan hypyn perustuen viestin sisältöön. Tämän tyyppiset reititysprotokollat soveltuvat hyvin useimpiin langattomiin anturiverkkoihin, koska tiedonkeruusolmu ei ole kiinnostunut tietyn anturisolmun datasta vaan tietyyntyyppisestä datasta sen alkuperästä riippumatta.

## 4 Klusterointi

Langattomien anturiverkkojen käyttö on kasvanut viime vuosina useissa sovelluksissa, kuten ympäristön tai taistelukentän valvonnassa. Yleensä pienet anturisolmut ripotellaan maastoon, johon ne jätetään valvomattomina mittaamaan ja raportoimaan ympäristön eri tunnuslukuja kuten lämpötilaa, painetta, kosteutta, valoa tai kemiallista aktiivisuutta. Anturisolmujen lähettämät raportit kerätään esimerkiksi tukiaseman tai tiedonkeruusolmun toimesta. Anturisolmut ovat yleensä erittäin energiarajoitteisia, joten energiatehokkuuden parantaminen on erittäin tärkeää langattomissa anturiverkoissa. [49]

Energian kulutusta voidaan vähentää tehokkaalla verkon rakenteella jakamalla anturisolmut ryhmiin eli klustereihin. Jokaisessa klusterissa anturisolmuille jaetaan erilaisia rooleja, kuten pääsolmu, gateway-solmu tai jäsensolmu. Pääsolmu toimii klusterin koordinaattorina ja sen vastuulla on yleensä klusterin sisäisen tiedonsiirron järjestäminen, klusterin ylläpito sekä datan kokoaminen ja välitys. Pääsolmut kuluttavat enemmän energiaa kuin jäsensolmut, sillä niiden kautta kulkee paljon viestiliikennettä ja ne saattavat lähettää dataa myös pitempiä matkoja [49]. Jotta pääsolmut eivät kuluttaisi energiaansa nopeasti loppuun, täytyy pääsolmun roolia kierrättää. Gateway-solmujen kautta voidaan välittää dataa klustereiden välillä, mutta ne eivät kuitenkaan ole välttämättömiä langattomissa anturiverkoissa. Jäsensolmut ovat anturisolmuja, jotka pystyvät kommunikoimaan pelkästään pääsolmun kanssa joko suoraan tai muiden anturisolmujen kautta. [47]

Klusterointiprotokollille asetetut tavoitteet vaihtelevat riippuen mm. käytetystä sovelluksesta, anturisolmujen resursseista ja toimintaympäristön luonteesta. Tavoitteena voi olla esimerkiksi anturisolmujen jakaantuminen tasaisesti klustereiden kesken tai klustereiden lukumäärän minimoiminen. Suurin huolenaihe langattomissa anturiverkoissa on kuitenkin verkon elinikä, koska anturisolmut ovat energiarajoitteisia.

Klusterointiprotokollan suoritus voi tapahtua joko keskitetysti esimerkiksi tukiasemassa tai hajautetusti paikallisissa anturisolmuissa. Hajautetut klusterointimenetelmät ovat näistä yleisempiä, koska ne soveltuvat paremmin suuriin verkkoihin. Myös keskitetyn ja hajautetun klusterointimenetelmän yhdistelmä on mahdollinen. Klusterointiprotokollaa

suoritetaan kierroksittain ja jokainen kierros alkaa järjestelyvaiheella, jossa valitaan pääsolmut ja muodostetaan klusterit. Pääsolmut voidaan joko ennalta määrätä tai valita anturisolmujen joukosta. Valinta anturisolmujen joukosta tehdään joko todennäköisyyteen perustuen tai deterministisesti tiettyjen paikallisten mittareiden perusteella. Klusterin muodostuksessa anturisolmut valitsevat pääsolmunsu joidenkin kriteereiden perusteella ja pyytävät lupaa liittyä ko. pääsolmun klusteriin. Anturisolmu liittyy klusteriin, kun pääsolmu hyväksyy anturisolmun pyynnön. Järjestelyvaiheen jälkeen tulee vakaantilanvaihe, missä klusterin jäsensolmut välittävät havainnoimaansa dataa pääsolmulle, joka kokoaa datan ja välittää sen eteenpäin muiden pääsolmujen tai gateway-solmujen kautta tukiasemalle.

Langattomien anturiverkkojen sovellukset eivät välttämättä ole kiinnostuneita datalähteen identiteetistä vaan ympäristöstä mitattavasta tiedosta. Tällöin on kyse datakeskeisestä verkkotoiminnasta, jonka vahvuus on kyky operoida dataa samalla kun sitä kuljetetaan verkossa. Esimerkiksi datan kokoamisessa kootaan sisällöltään yhtenevät sanomat yhteen ja lähetetään pelkästään koottu tieto verkon yli. Näin saadaan vähennettyä lähetettävien viestien määrää ja säästettyä merkittävästi energiaa.

#### 4.1 Tavoitteet

Kirjallisuudessa klusterointiprotokollille asetetut tavoitteet vaihtelevat ja ne määräytyvät usein sovelluksen vaatimusten mukaan. Esimerkiksi jos sovellus on herkkä viiveelle, tällöin klusterin sisäinen ja ulkoinen liitettävyyss sekä reitityksen polun pituus ovat kriteereinä pääsolmun valinnassa ja anturisolmujen ryhmittelyssä. Artikkelissa [1] on esitetty yleisiä verkon klusterointitavoitteita: kuormantasaus, vikasietoisuus, liitettävyyden parantaminen ja viiveen vähentäminen, klustereiden lukumäärän minimointi sekä verkon eliniän maksimointi.

Anturisolmujen jakaantuminen klustereiden kesken tasaisesti on yleensä tavoitteena silloin, kun pääsolmu suorittaa datan prosessointia tai klusterin sisäisiä hallintatehtäviä. Ottamalla huomioon pääsolmujen tehtävät, on intuitiivista jakaa kuorma pääsolmujen kesken siten, että ne voivat saavuttaa odotetut suorituskykytavoitteet. Kuormantasaus on tärkeämpi kysymys langattomissa sensoriverkoissa silloin, kun pääsolmu valitaan vapaiden

anturisolmujen joukosta. Tällaisessa tapauksessa samankokoisten klustereiden muodostus on ratkaisevaa pidentämään verkon elinikää, sillä se estää pääsolmujen nopeaa energiankulutusta ja ennen aikaista vikaantumista. Anturisolmujen tasainen jakaantuminen voi vaikuttaa myös viiveeseen.

Langattomat sensoriverkot toimivat usein karuissa ympäristöissä ja siksi anturisolmuilla on suuri riski altistua toimintahäiriöille ja fyysisille vioille. Pääsolmun vikaantumisen sietäminen on usein välttämätöntä sovelluksissa, joissa pyritään välttämään sensorisolmun mittaaman tärkeän datan hävittämistä. Intuitiivisin tapa toipua pääsolmun viasta on verkon uudelleenklusterointi. Uudelleenklusterointi ei pelkästään kuormita anturisolmujen resursseja. Se on usein myös hyvin häiritsevä meneillään oleville operaatioille. Tämän vuoksi samanaikaiset viansietotekniikat olisivat tässä tapauksessa soveltuvampia. Varapääsolmujen sijoittaminen on merkittävin malli pääsolmun viasta toipumiseen. Varapääsolmun valinta ja rooli vaihtelevat normaaleissa verkko-operaatioissa. Jos pääsolmuilla on pitkän kantaman radiot, voivat läheiset pääsolmut ottaa vikaantuneen klusterin anturisolmut omaan klusteriinsa anturisolmujen radiokantaman rajoissa. Kuormantasauksen lisäksi, pääsolmun kierrättämisellä parannetaan vikasietoisuutta.

Klustereiden välinen liitettävyyden vaatimus monissa sovelluksissa, ellei pääsolmuilla ole todella pitkänmatkan tiedonvälityskykyä, esimerkiksi satelliittilinkkiä. Varsinkin kun pääsolmut valitaan anturisolmujen joukosta. Liitettävyyden tavoite voidaan rajoittaa pelkästään takaamaan reitin saatavuutta pääsolmusta tukiasemalle tai olemaan rajoittavampi asettamalla raja-arvo reitin pituudelle. Kun useampi anturisolmu omaksuu pääsolmun roolin, liitettävyyden tavoite tekee verkon klusteroinnista yhden yhdistetyn dominoivan joukon ongelman varianteista. Toisaalta, kun datan viive on huolenaiheena, klusterin sisäisestä liitettävyydestä tulee suunnittelun tavoite tai rajoite. Usein viive huomioidaan asettamalla datan kulkemalle reitille sallittu maksimi hyppyjen määrä,  $K$ . Esimerkiksi  $K$ -hypyn klusterointi on  $K$ -dominoivan joukon ongelma.

Klustereiden lukumäärän minimoiminen on varsin yleinen tavoite silloin, kun pääsolmuina toimii erityisen resurssirikkaita anturisolmuja. Verkon suunnittelija haluaa usein käyttää mahdollisimman vähän näitä resurssirikkaita anturisolmuja, koska ne ovat kalliimpia ja

haavoittuvampia kuin tavalliset anturisolmut. Esimerkiksi jos pääsolmut ovat kannettavia tietokoneita, robotteja tai liikkuvia ajoneuvoja, asettaa se luonnostaan joitain rajoituksia anturisolmujen lukumäärälle. Tämän tyyppisten anturisolmujen sijoittaminen voi olla vaikeaa esimerkiksi silloin, kun langaton anturiverkko toimii sota-alueella tai metsässä. Lisäksi tällaisten anturisolmujen koko on huomattavasti suurempi kuin tavallisten anturisolmujen, joten ne ovat helpommin havaittavia. Anturisolmun näkyvyys ei ole toivottua monissa langattomien anturiverkkojen sovelluksissa, kuten rajan vartioinnissa, sotilastiedustelussa ja infrastruktuurin turvallisuudessa.

Verkon elinikä on suurin huolenaihe langattomissa anturiverkoissa, koska anturisolmut ovat energiarajoitteisia. Kun pääsolmut ovat resursseiltaan rikkaampia kuin tavalliset anturisolmut, on välttämätöntä minimoida klusterin sisäisen tiedonvälityksen energiankulutus. Jos vain mahdollista, pääsolmut tulisi sijoittaa mahdollisimman monen samaan klusteriin kuuluvan anturisolmun lähelle. Toisaalta, jos pääsolmut ovat tavallisia anturisolmuja, voidaan verkon ikää pidentää rajaamalla anturisolmujen kuormaa esimerkiksi kierrättämällä pääsolmua tai muodostamalla samankokoisia klustereita. Yhdistetyn klusteroinnin ja reitinvalinnan katsotaan maksimoivan verkon eliniän. Mukautuva klusterointi on myös toimiva valinta pidentämään verkon elinikää.

## 4.2 Klusterointimenetelmät

Klusterointiprotokollan suoritus tapahtuu joko keskitetysti esimerkiksi tukiaseman toimesta tai hajautetusti paikallisissa anturisolmuissa. Myös näiden yhdistelmä eli hybrid menetelmä on mahdollinen.

Keskitetyissä protokollissa kaikki sovelluksen vaatima laskenta, kuten klusterointiprotokollan suoritus, tapahtuu keskitetysti esimerkiksi tukiaseman toimesta. Monet keskitetyt klusterointiprotokollat vaativat täyden tiedon verkon topologiasta ja/tai jäljellä olevasta energiasta, jotta voivat keskitetysti laskea optimaaliset klusterit [17]. Esimerkiksi klassista K-means klusterointiprotokollaa voidaan käyttää, jos haluttujen klustereiden lukumäärä voidaan päättää etukäteen ja anturisolmujen sijainnit ovat tiedossa. [49]

Keskitettyjen protokollien tehokkuus on rajoitettua suurissa verkoissa, missä kaiken tarvittavan tiedon keräys keskitetysti on sekä aikaa vievää että energiaa kuluttavaa. [49] Keskitetty rakenne on altis myös yksittäisille vioille [15].

Esimerkki keskitetystä klusterointiprotokollasta on BCDCP (*Base-Station Controlled Dynamic Clustering Protocol*) [36]. Siinä klusterit muodostetaan siten, että jokainen pääsolmu palvelee suurin piirtein yhtä montaa liikkuvaa anturisolmua. Näin saadaan tasattua pääsolmujen kuormitusta. Klusterin muodostuksen alussa tukiasema vastaanottaa verkon jokaiselta anturisolmulta tiedon jäljellä olevasta energiasta. Tämän tiedon perusteella tukiasema laskee verkon kaikkien anturisolmujen keskimääräisen energiatason ja valitsee joukon anturisolmuja, joiden energiataso on keskiarvon yläpuolella. Tästä anturisolmujen joukosta valitaan pääsolmut kyseiselle kierrokselle. Tämän jälkeen tukiasema jakaa verkon valitun anturisolmujoukon perusteella klustereihin käyttämällä klusterinhalkaisualgoritmia. Klusterinhalkaisualgoritmi jakaa verkon ensin kahteen klusteriin ja etenee jakamalla klusterit aina pienempiin klustereihin. Tukiasema suorittaa klusterinhalkaisualgoritmin niin monta kertaa, että haluttu klusterimäärä on saavutettu. Iteratiivinen klusterinhalkaisualgoritmi varmistaa pääsolmujen tasaisen jakaantumisen maksimoimalla jokaisessa halkaisuvaiheessa pääsolmujen välisen etäisyyden. BCDCP käyttää monen hypyn reititystä siirtäessään havaittua dataa klusterista muiden klusterien pääsolmujen kautta tukiasemalle.

Younis ym. käsittelevät hajautettuja klusterointimenetelmiä artikkelissa [49]. Hajautetuissa klusterointimenetelmissä anturisolmu tekee päätöksensä klusteriin liittymisestä tai pääsolmuksi ryhtymisestä yhden hypyn päässä olevilta naapureilta saamansa tiedon perusteella. Hajautetut klusterointiprotokollat ovat keskitettyjä klusterointiprotokollia soveltuvampia suurissa langattomissa anturiverkoissa. Nämä menetelmät ovat luonteeltaan joko iteratiivisia tai satunnaisia.

Iteratiivisissa klusterointimenetelmissä anturisolmut odottavat tiettyä tapahtumaa tai tietyn anturisolmun päättävän roolistaan ennen kuin tekevät oman päätöksensä. Esimerkiksi DCA:ssa (*Distributed Clustering Algorithm*) [8] anturisolmu odottaa painoltaan suurempien naapurianturisolmujen tekevän ensin päätöksensä, ovatko ne pääsolmuja

vai liittyvätkö ne olemassa olevaan klusteriin, ennen kuin solmu tekee oman päätöksensä. Solmun painolla tarkoitetaan jonkin parametrin arvoa esimerkiksi energiatasoa. Jos anturisolmun paino on suurempi kuin yhdenkään muun yhden hypyn päässä sijaitsevan anturisolmun, valitaan se pääsolmuksi. Jos anturisolmu vastaanottaa useita pääsolmuilmoituksia, se tekee valinnan näiden väliltä jonkin ehdon mukaan, esimerkiksi korkeampi paino voittaa. Jos mikään anturisolmu painavampi naapuri ei päästä tulla pääsolmuksi, päättää anturisolmu olevansa itse pääsolmu.

Iteratiivisten klusterointitekniikoiden ongelmana on algoritmin nopeuden riippuminen verkon halkaisijasta eli pisimmästä polusta. Kaksiulotteisessa kentässä, jossa on  $n$  kappaletta anturisolmuja, DCA-protokollan aikavaativuus on  $O(\sqrt{n})$ . Iteratiivisten tekniikoiden suorituskyky on myös hyvin herkkä pakettien häviöille. Jos esimerkiksi DCA-protokollassa anturisolmu  $u$  havaitsee, että sillä on naapurina painavampi anturisolmu  $v$ , anturisolmun  $u$  pitää odottaa anturisolmun  $v$  päätöstä ennen kuin voi itse tehdä päätöksensä. Jos anturisolmu  $v$  vikaantuu ennen kuin se on ennättänyt tehdä päätöksensä, jää anturisolmu  $u$  odottamaan anturisolmun  $v$  päätöstä määräämättömäksi ajaksi.

Jotkin protokollat rajaavat kunkin anturisolmun iterointikertojen määrää, jotta välttyttäisiin edellä mainituilta ongelmilta. Esimerkiksi ACE (*An Emergent Algorithm for Highly Uniform Cluster Formation*) -protokollassa [12] anturisolmu tekee päätöksensä käytettävissä olevan tiedon perusteella, kun maksimi iterointikertojen määrä on saavutettu. Näillä iteraatiokerroilla saavutetaan stabiili keskimääräinen klusterikoko.

Satunnaisissa klusterointimenetelmissä jokainen anturisolmu päättää itsenäisesti roolistaan klusteroidussa verkossa. Satunnainen klusterointimenetelmä takaa nopean klusteroinnin ja saavuttaa joitain hyviä ominaisuuksia, kuten tasapainoiset klusterikoot. Myös ylimääräisen viestiliikenteen määrä pysyy vähäisenä. Esimerkki satunnaisesta klusterointimenetelmästä on alaluvussa 5.1 esitetty LEACH.

### 4.3 Vaiheet

Klusterointiprotokollien suoritus tapahtuu kierroksittain ja yksi kierros sisältää yleensä Kuvassa 9 esitetyt neljä vaihetta: pääsolmun valinta, klusterin muodostus, datan kokoaminen ja tiedonsiirto.



Kuva 9. Klusterointiprotokollan vaiheet [37].

Protokollan suoritus alkaa järjestelyvaiheella, jossa ensimmäisenä valitaan pääsolmut ja tämän jälkeen muodostetaan klusterit. Järjestelyvaihetta seuraa vakaantilanvaihe, joka voidaan jakaa datan kokoamiseen sekä tiedonsiirtoon. Vakaantilanvaiheen aikana anturisolmut välittävät siis havainnoimaansa dataa pääsolmulle, joka suorittaa datan kokoamisen ja välittää kootun datan tukiasemalle. Vakaantilanvaiheen jälkeen alkaa uusi kierros ja se alkaa taas järjestelyvaiheella. [37]

### 4.4 Pääsolmu

Jokaisessa klusterissa on aina pääsolmu, joka toimii klusterin koordinaattorina. Pääsolmujen ominaisuudet ja tehtävät vaihtelevat sovelluksesta ja klusterointiprotokollasta riippuen. Myös pääsolmujen valinta ja kierrätys vaihtelevat eri klusterointiprotokollissa. Seuraavaksi käydään läpi pääsolmujen ominaisuuksia ja tehtäviä sekä esitellään pääsolmun valintamenetelmiä ja käsitellään pääsolmun kierrätystä.

#### 4.4.1 Pääsolmun ominaisuuksia

Klusterointiprotokollan valintaan vaikuttaa verkkomalli, varsinkin anturisolmujen kyvyt ja verkon sisäinen tietojenkäsittely. Abbasi ja Younis käsittelevät pääsolmun ominaisuuksia ja tehtäviä artikkelissa [1]. Sen mukaan klusterointiprotokollat voidaan jaotella seuraavien



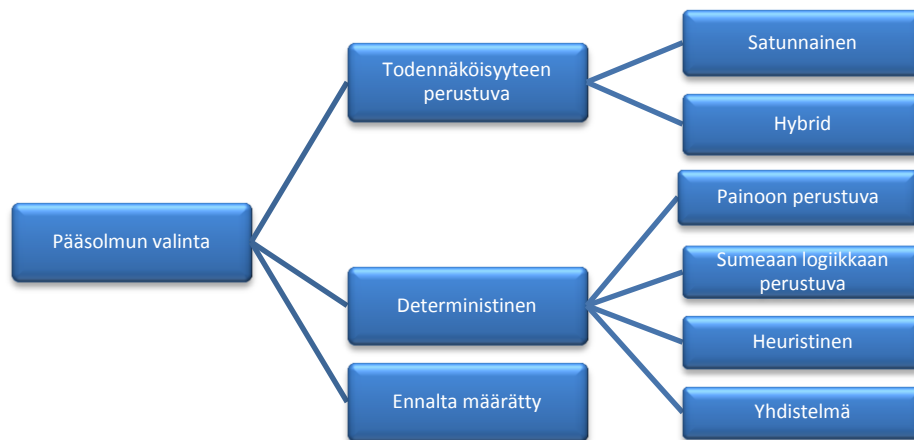
pääsolmun attribuuttien mukaan: anturisolmun tyyppi, rooli ja liikkuvuus. Joissain klusterointiprotokollissa kaikki anturisolmut ovat resursseiltaan samanlaisia ja pääsolmut valitaan näiden anturisolmujen joukosta. Kun taas toisissa klusterointiprotokollissa pääsolmut ovat laskenta- ja tiedonsiirtokyvyiltään muita anturisolmuja huomattavasti parempia.

Pääsolmun tehtävät vaihtelevat sovelluksesta riippuen. Pääsolmun tehtävänä voi yksinkertaisesti olla vain välittää klusterin sisäisten anturisolmujen lähettämää dataa tai suorittaa anturisolmuilta kerätyn datan kokoamista tai fuusiota. Joskus pääsolmu voi toimia myös tiedonkeruusolmuna tai tukiasemana, joka suorittaa erilaisia toimenpiteitä riippuen havaitsemastaan ilmiöstä tai kohteesta. Pääsolmun ollessa liikkuva, on klusterin jatkuva ylläpito tarpeen, sillä klusterin rakenne muuttuu dynaamisesti. Toisaalta liikkumaton pääsolmu pyrkii muodostamaan kiinteitä klustereita ja helpottaa näin ollen klusterin sisäistä ja ulkoista verkonhallintaa. Joskus pääsolmut voivat kulkea rajoitettuja etäisyyksiä parantaakseen verkon suorituskykyä.

Vuonna 2011 Deng ym. esittelivät MBC (*Mobility-based clustering protocol*) -protokollan [14], missä pääsolmut valitaan jäljellä olevan energian sekä liikkuvuuden perusteella. Mitä enemmän on jäljellä olevaa energiaa ja mitä pienempi on anturisolmun nopeus, sitä todennäköisemmin se valitaan pääsolmuksi. Klusterin muodostuksen ajan pääsolmujen ja klusterin jäsenolmujen väliset linkit ovat vakaita. Klusterin muodostuksen jälkeen vakaantilan vaiheessa jäsenolmut lähettävät dataa pääsolmuille käyttäen aikajakokanavointia eli jokainen jäsenolmu lähettää dataa vuorollaan TDMA (*Time Division Multiple Access*) -aikataulujen mukaan. Jos anturisolmu poistuu sen hetkisestä klusterista, liittyy se uuteen klusteriin lähettämällä uudelle pääsolmulle liittymisviestin. Näin voidaan välttää pakettien häviöitä, kun anturisolmu on menettänyt tai on menettämässä yhteyden pääsolmuun. Dengin ym. artikkelissa esittämät laajat simuloinnit osoittavat, että MBC on energiankulutukseltaan parempi kuin LEACH ja se kykenee huomioimaan myös anturisolmujen liikkuvuuden.

#### 4.4.2 Pääsolmun valinta

Klusterointiprotokollat käyttävät eri menetelmiä pääsolmun valintaan. Pääsolmut voidaan ennalta määrätä tai valita anturisolmujen joukosta erilaisten valintaperusteiden mukaan. Artikkelissa [3] klusterointiprotokollat on jaoteltu pääsolmun valinnan mukaan todennäköisyyteen perustuviin, deterministisiin sekä ennalta määrättyihin, kuten Kuvassa 10. Seuraavaksi käydään tarkemmin läpi nämä kolme pääsolmun valintamenetelmää. Tässä alaluvussa viitataan artikkeliin [3], ellei toisin ole ilmoitettu.



Kuva 10. Luokittelu pääsolmun valinnan mukaan [3].

#### Todennäköisyyteen perustuva valinta

Klusterointiprotokollien, joissa pääsolmun valinta tehdään todennäköisyyteen perustuen, päätavoitteena on pidentää verkon elinikää mahdollisimman paljon. Jotkin tällaiset protokollat, kuten LEACH, valitsevat pääsolmun sattumanvaraisesti. Todennäköisyyteen perustuvat protokollat ovat yksinkertaisia ja lähes optimaalisia ylimääräisen viestiliikenteen suhteen. Ylimääräistä viestintää aiheutuu siitä, että anturisolmut tarvitsevat paikallista tietoa pystyäkseen organisoitumaan klustereihin. Ylimääräisen viestinnän tulisi pysyä mahdollisimman vähäisenä, jotta klusterointiprotokolla olisi tehokas. Todennäköisyyteen perustuvat protokollat voidaan jakaa satunnaisiin ja hybrideihin.

Esimerkki satunnaisesta protokollasta on EEHC (*Energy Efficient Hierarchical Clustering*) [7], joka jakaa verkon hierarkiatasoihin. Algoritmi toimii alhaalta ylös periaatteella ja se voidaan jakaa alku- ja laajennusvaiheeseen. Alkuvaiheessa jokainen anturisolmu ilmoittaa

itsensä alimman tason eli tason 1 pääsolmuksi todennäköisyydellä  $p$ . Näitä pääsolmuja kutsutaan vapaaehtoisiksi pääsolmuiksi. Kaikki  $k$ -hypyn päässä olevat anturisolmut saavat tämän ilmoituksen joko suoraan tai välitettynä. Jokaisesta tämän ilmoituksen vastaanottavasta anturisolmusta, joka ei itse ole pääsolmu, tulee lähimmän klusterin jäsen. Jos ilmoitus ei saavuta anturisolmua asetetussa ajassa  $t$ , tulee siitä pakotettu pääsolmu. Laajennusvaiheessa muodostetaan hierarkian ylemmät kerrokset toistamalla klusterointiprosessia rekursiivisesti. Alimman tason eli tason 1 pääsolmut ilmoittavat itsensä tason 2 pääsolmuiksi todennäköisyydellä  $p_2$ . Tämä päätös välitetään kaikille  $k_2$  hypyn päässä oleville anturisolmuille. Tason 1 pääsolmut, jotka vastaanottavat ilmoituksen tason 2 pääsolmuilta, liittyvät lähimmän tason 2 pääsolmun klusteriin. Kaikista muista tason 1 pääsolmuista tulee tason 2 pakotettuja pääsolmuja. Pääsolmujen valinta tasoilla  $3,4,\dots,h$  tehdään samalla tavalla. Algoritmi takaa  $h$ -hypyn liitettävyyden pääsolmujen ja tukiaseman välillä. Oletetaan, että klusterointihierarkiassa on  $h$  tasoa ja taso 1 matalin ja taso  $h$  korkein. Anturisolmut lähettävät keräämänsä datan tason 1 pääsolmuille. Tason 1 pääsolmut kokoavat datan ja lähettävät sen tason 2 pääsolmuille jne. Korkeimman tason pääsolmut lähettävät lopulta kootun datan tukiasemalle.

Hybridi menetelmissä pääsolmun valinta ei perustu pelkästään todennäköisyyteen. Esimerkiksi Younisin ja Fahmin esittelemässä HEED (*Hybrid Energy-Efficient Distributed Clustering*) -protokollassa [48] pääsolmun valinta perustuu anturisolmun jäljellä olevaan energiaan sekä toissijaiseen parametriin, kuten naapurisolmun etäisyyteen tai anturisolmun asteeseen. Siinä pääsolmuksi valitulla anturisolmulla on keskimääräisesti suhteellisen korkea jäljellä oleva energia verrattuna klusterin jäsensolmuihin. Ennen kuin anturisolmu alkaa suorittaa HEED-protokollaa, lasketaan anturisolmun todennäköisyys tulla pääsolmuksi seuraavasti

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}},$$

missä  $C_{prob}$  on pääsolmujen prosentuaalinen osuus kaikista anturisolmuista.  $C_{prob}$  ei vaikuta lopullisiin klustereihin. Sillä rajoitetaan pelkästään klusteroinnin alkuvaiheessa pääsolmuiksi ilmoittautuvien anturisolmujen määrää.  $E_{residual}$  on arvioitu anturisolmun

jäljellä oleva energiamäärä ja  $E_{max}$  on anturisolmun maksimi energiamäärä ja se on tyypillisesti sama kaikilla anturisolmuilla. Tämän jälkeen jokainen anturisolmu käy läpi useita iteraatiokertoja kunnes se löytää pääsolmun. Jos se ei kuule yhtään pääsolmun ilmoitusviestiä, valitsee se itsensä pääsolmuksi ja lähettää ilmoitusviestin naapurisolmuilleen. Anturisolmu voi ilmoittaa naapurisolmuilleen kaksi eri tilaa: alustava tila ja lopullinen tila. Jos anturisolmun  $CH_{prob} < 1$ , tällöin anturisolmu on alustava pääsolmu ja se voi myöhemmin vaihtaa tilansa tavalliseksi jäsensolmuksi. Jos taas anturisolmun  $CH_{prob} = 1$ , tulee siitä pysyvä pääsolmu. Jokainen anturisolmu valitsee pääsolmukseen anturisolmun, jonka viestintäkustannukset ovat pienimmät.

### **Deterministinen valinta**

Toisin kuin todennäköisyyteen perustuvissa klusterointiprotokollissa, deterministisissä klusterointiprotokollissa pääsolmun valinta tehdään tiettyjen mittareiden perusteella. Yleensä nämä mittarit saadaan paikallisesti ja ne perustuvat anturisolmun tilaan. Tavanomaisimmat mittarit pääsolmun valinnassa ovat jäljellä oleva energia, anturisolmun aste ja etäisyys naapurisolmuihin tai tukiasemaan. Anturisolmut saavat nämä tiedot naapurianturisolmuiltaan viestien vaihdon avulla. On olemassa erilaisia deterministisiä protokollia ja ne voidaan jakaa neljään kategoriaan: painoon perustuviin, sumeaan logiikkaan perustuviin, heuristisiin sekä yhdistelmiin.

Painoon perustuvissa protokollissa, pääsolmun valinta tehdään anturisolmujen painojen perusteella. Anturisolmun paino lasketaan erilaisten mittareiden, kuten esimerkiksi jäljellä olevan energian, etäisyys tukiasemaan tai naapurianturisolmujen määrän perusteella. Näiden lähestymistapojen perusteena on kilpailu, jossa pääsolmuiksi valitaan painoltaan suurimmat anturisolmut ja näin ollen saavutetaan parempi suorituskyky. Vaikka monet painoon perustuvista protokollista ovat hajautettuja, on julkaistu myös keskitettyjäkin lähestymistapoja. Esimerkki hajautetusta painoon perustuvasta protokollasta on alaluvussa 4.2 esitetty DCA.

Joidenkin klusterointiprotokollien pääsolmujen valinta perustuu sumeaan logiikkaan. Sen avulla voidaan tehdä reaaliaikaisia päätöksiä ilman täydellistä tietoa ympäristöstä. Sumea

logiikkaa käytetään yleensä mallintamaan ihmisen päätöksenteon käyttäytymistä. Siinä syötteen/ulostulon suhde ilmaistaan joukkona kielellisiä sääntöjä tai relaatiomääritelmänä. Sumean logiikan järjestelmä sisältää seuraavat neljä osaa: sumeuttajan (*engl. fuzzifier*), päätöksentekologiikan, selkeyttäjän (*engl. defuzzifier*) ja sääntökannan. Yleensä syöte on täsmällistä, joten syöte pitää sumeuttaa ja päätöksentekologiikka muuntaa täsmällisen syötteen joukoksi kielellisiä arvoja. Sääntökanta määrittää taas järjestelmän käytöksen. Päätöksentekologiikalta ulostuleva data täytyy vielä selkeyttää eli muuttaa täsmälliseksi arvoksi. Monimutkaisuus on sumeaan logiikkaan perustuvien klusterointiprotokollien suurin ongelma, sillä algoritmin suoritukseen vaaditaan paljon viestienvaihtoa. Monimutkaisten algoritmien suoritus resursseiltaan heikoissa anturisolmuissa kuluttavat paljon energiaa.

Leen ja Chengin esittelemä LEACH-ERE (*LEACH Expected Residual Energy*) [31] on esimerkki sumeaa logiikkaa käyttävästä klusterointiprotokollasta. Siinä jokainen pääsolmun valinta perustuu ennustettuun jäljellä olevaan energiaan. Sumealle säätimelle tulee syötteenä jäljellä oleva energia  $E_{residual}$  ja ennustettu jäljellä oleva energia  $E_{expResidual}$  ja se palauttaa parametrin *chance* eli anturisolmun todennäköisyyden tulla valituksi uudeksi pääsolmuksi. Mitä suurempi on parametrin *chance* arvo, sitä todennäköisemmin anturisolmu tulee valituksi uudeksi pääsolmuksi.

Viimeisen vuosikymmenen aikana heuristisia klusterointiprotokollia on tutkittu merkittävästi. Heuristinen algoritmi pyrkii ratkaisemaan annetun tehtävän lähelle tarkkaa optimiratkaisua. Heuristiset lähestymistavat mahdollistavat pääsolmujen hyvän jakaantumisen verkossa ja tarjoavat energiatehokkaan verkon löytämällä parhaan joukon pääsolmuja ja parhaimmat klustereiden koot. Heuristisissa klusterointiprotokollissa käytetään monen tyyppisiä optimointitekniikoita kuten geneettistä algoritmia, differentiaalievoluutiota, muurahais-, mehiläis-, bakteeri- tai parviontimointia. [3] Geneettisissä algoritmeissa ratkaisuehdotusten joukosta eli populaatiosta kehitetään uusia sukupolvia käyttämällä geneettisiä operaatioita, joita ovat uudelleenvalinta, risteytys ja mutaatio. Uusia sukupolvia luodaan kunnes on löydetty tarpeeksi hyvä ratkaisu. Differentiaalievoluutio on samankaltainen geneettisen algoritmin kanssa, mutta ne eroavat siinä että geneettisissä algoritmissa populaatiot ovat binäärijonoja ja

differentiaalievoluutiassa taas vektoreita. Muurahais-, mehiläis- ja bakteerioptimoinnissa mallinnetaan muurahaisten, mehiläisten tai bakteerien ravinnonetsintää. Esimerkiksi muurahaiset jättävät kulkiessaan jälkeensä feromoneja ja näin ollen muurahaisten kulkiessa lyhimmän reitin feromonipitoisuus kasvaa. Muurahaisen tullessa risteykseen, valitsee se reitikseen lyhyemmän eli suuremman feromonipitoisuuden omaavan reitin. Parviontimointi on myös populaatiopohjainen menetelmä, joka mallintaa eläinten käytöstä parvissa ja laumoissa. Populaatiot koostuvat partikkeleista, joilla on sekä sijainti- että nopeusvektori. Toisin kuin muissa populaatiopohjaisissa menetelmissä, siinä ei synny uusia ratkaisuja eikä siinä tehdä minkäänlaisia valintoja. Näin ollen populaatio pysyy samana, mutta sen jäsenet eli partikkelit lähestyvät ohjautuvalla mutaatiolla (*engl. directed mutation*) kohti parasta ratkaisua. [35]

Esimerkki keskitetystä parviontimointipohjaisesta klusterointiprotokollasta on Bennanin ja El Ghanamin vuonna 2012 esittelemä PSO-BC (*a centralized PSO-based clustering algorithm*) [9]. Siinä tukiasema valitsee pääsolmuiksi anturisolmut, joiden energia on suurempi kuin verkon anturisolmujen keskimääräinen energia. Algoritmin kustannusfunktio (*engl. fitness function*) on muotoa

$$f(P_j) = f_1(P_j) + \lambda f_2(P_j),$$

missä  $f_1$  on anturisolmujen ja lähimmän pääsolmun välinen euklidisen etäisyyden summa ja  $f_2$  on pääsolmujen ja tukiaseman välisen euklidisen etäisyyden summa.  $\lambda$  on kerroin joka säätelee pääsolmujen ja tukiaseman etäisyyttä ja näin ollen myös klusterin energiankulutusta. Mitä suurempi on kerroin  $\lambda$ , sitä lähempänä tukiasemaa pääsolmut sijaitsevat. Artikkelissa esitetyt simuloinnit osoittavat, että verkon eliniän suhteen PSO-BC protokolla suoriutuu paremmin kuin LEACH-C (*LEACH-centralized*).

Jokainen näistä tekniikoista käyttää eri kustannusfunktioita saavuttaakseen omat tavoitteensa. Klusterointi kuuluu NP-täydellisten optimointiongelmien luokkaan ja eri algoritmien suorituskyvyt voivat vaihdella suuresti. Heurististen menetelmien suorituskyvyn arvioimisessa tärkeä mittari energiatehokkuuden lisäksi on myös suppenemisaika. Kyseiset menetelmät ovat usein keskitettyjä ja suoritetaan tukiaseman

johdolla, koska ne tarvitsevat globaalin tiedon verkon rakenteesta. Yleensä heuristiset menetelmät eivät ole skaalautuvuudeltaan hyviä ja näin ollen eivät sovellu suuriin verkkoihin. Parvioptimointi on osoittanut olevansa suorituskyvyiltään muita heuristisia menetelmiä parempi.

Jotkin menetelmät käyttävät useita erilaisia mittareita kuten anturisolmun aste, läheisyys, liikkuvuus tai linkin tila saavuttaakseen tavoitteensa. Tällaisia protokollia kutsutaan yhdistelmiksi. Esimerkki tällaisesta yhdistelmästä on alaluvussa 5.2 esitetty DSBCA, missä klustereiden muodostus huomioi sekä tukiaseman etäisyyden että anturisolmujen tiheyden.

### **Ennalta määrätty valinta**

Joissain klusterointiprotokollissa pääsolmut tai klusterit voidaan valita tai määrätä ennalta ennen kuin anturisolmuja on edes sijoitettu. Tällaisia protokollia kutsutaan ennalta määrättyiksi. Periaatteessa tällaisilla protokollilla on suuri ongelma, sillä klustereiden muodostus tapahtuu anturisolmuihin ladatun tiedon perusteella jo ennen anturisolmujen sijoittamista ja näin ollen pääsolmujen valinta ja klustereiden muodostus ei ole dynaamista eikä huomioi lainkaan verkon tai anturisolmujen tiloja. Jos verkon tila eroaa suuresti tehtyjen oletusten kanssa, johtaa se tällöin huonoon suorituskykyyn. Tästä johtuen useimmat protokollat muodostavat klusterit dynaamisesti ja ennalta määrättyjä protokollia käytetäänkin vain tietyissä sovelluksissa.

Esimerkki ennalta määrätystä klusterointiprotokollasta on Buttyanin ja Schafferin vuonna 2007 esittelemä PANEL (*Position-based aggregator node election protocol*) [11]. Sen päätavoitteena on tukea luotettavia ja pysyviä tallennussovelluksia, kuten TinyPEDS (*Tiny persistent encrypted data storage in asynchronous wireless sensor networks*) -sovellusta. PANEL olettaa, että anturisolmut ovat sijoitettuna rajatulle alueelle ja jaettu maantieteellisesti klustereihin. Siinä klusterointi määritetään jo ennen verkon käyttöönottoa ja jokaisella anturisolmulla on esisyötetty tieto mihin klusteriin se kuuluu. PANEL-protokollan toiminnot ovat jaettu ajanjaksoihin. Jokaisen ajanjakson alussa lasketaan vertailupiste täysin hajautetusti ja klusterin anturisolmut valitsevat vertailupistettä lähinnä sijaitsevan anturisolmun ajanjakson pääsolmuksi. PANEL käyttää sijaintipohjaista reititystä klustereiden väliseen tiedonsiirtoon eli reitittämään viestit

kaukaisilta pääsolmuilta kohti tietyn klusterin vertailupistettä. Viestin saapuessa klusteriin reititetään se eteenpäin kohti pääsolmua pääsolmun valinnassa muodostetun reititystaulun perusteella. PANEL käyttää reititykseen GPSR (*Greedy Perimeter Stateless Routing*) -protokollaa, mutta sen tilalla voidaan käyttää mitä tahansa sijaintipohjaista reititysprotokollaa. Buttyanin ja Schafferin artikkelissa esitellyt simuloinnit osoittavat, että PANEL on suorituskyvyltään merkittävästi parempi kuin LEACH tai HEED. Toisaalta protokollan klusterin muodostus on monimutkainen, joten se voi johtaa suurempaan energian kulutukseen.

#### **4.4.3 Pääsolmun kierrätys**

Klusteroiduissa langattomissa anturiverkoissa pääsolmulla on suurempi kuorma verrattuna klusterin muihin anturisolmuihin. Tämän takia pääsolmut kuluttavat energiansa huomattavasti nopeammin kuin tavalliset anturisolmut. Kierrättämällä pääsolmun roolia klusterin sisällä, voidaan kuormaa jakaa tasaisesti klusterin anturisolmujen kesken ja estää pääsolmun enneaikainen kuoleminen. [47]

Pääsolmun kierrättämiseen on monia mahdollisuuksia. Yksi tapa on käyttää ajastinta, jonka päätyminen käynnistää klusterointialgoritmin. Toinen tapa on käyttää dynaamisia parametreja, kuten jäljellä oleva pariston elinikä, käynnistämään klusterointialgoritmi. Tällöin pääsolmu käynnistää uuden pääsolmun valintaprosessin, jos sen pariston jäljellä oleva elinikä tippuu ennaltamäärätyn raja-arvon alapuolelle. [49]

Esimerkiksi Liun ja Lin esittelemä CHEP (*Cluster Head Election Protocol*) [33] valitsee uuden pääsolmun jäljellä olevan energiamäärän perusteella. Protokolla sisältää kaksi vaihetta: alkuvaihe ja uudelleenklusterointivaihe. Alkuvaiheessa CHEP jakaa verkon klustereihin, jollakin hajautetulla klusterointiprotokollalla. Tämän jälkeen kaikki klusterin sisäiset anturisolmut lähettävät tiedon jäljellä olevasta energiamäärästään pääsolmulle, joka valitsee uudeksi pääsolmuksi suurimman energiamäärän omaavan anturisolmun. Päätös uudesta pääsolmusta lähetetään klusterin anturisolmuille yleislähetystenä. Tämän jälkeen uudet pääsolmut muodostavat ylemmän tason klusterin. Uudelleenklusterointivaiheessa valitaan uusi pääsolmu. Pääsolmut toimivat aina tietyn klusterointijakson ajan, jonka



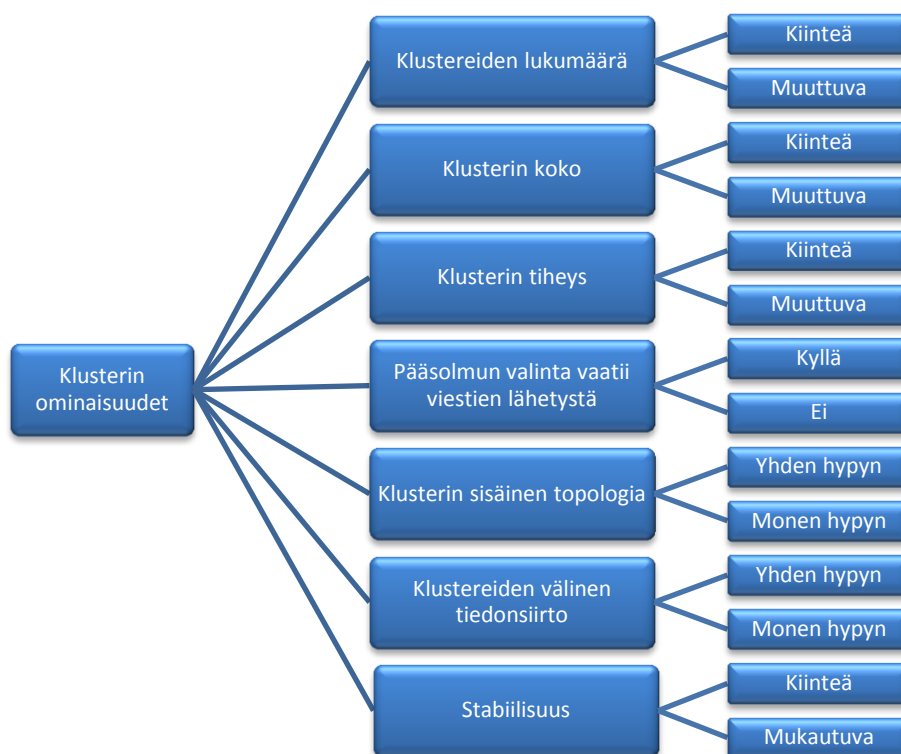
loputtua pääsolmu luovuttaa roolinsa anturisolmulle, jonka jäljellä oleva energiamäärä on suurin.

Pääsolmun kierrätysmenetelmä on tyypillisesti riippumaton klusterointiprotokollasta. On selvää, että tiheä pääsolmun kierrätys lisää ylimääräistä viestiliikennettä ja verkkokatkoja. Harvempi pääsolmun kierrätys saattaa taas aiheuttaa joidenkin anturisolmujen kuolemisen enneaikaisesti. Saavuttaakseen optimaalisen verkon eliniän, on olennaista löytää kompromissi näiden vaihtoehtojen väliltä. Yleensä sovellukset päättävät pääsolmun kierrätystiheyden heuristisesti perustuen joihinkin intuitiivisiin tekijöihin, kuten oletettuun pariston elinikään ja anturisolmun asteeseen. [49]

#### 4.5 Klusterin muodostus

Pääsolmun valintavaiheen jälkeen on vuorossa klusterin muodostus vaihe. Klusterin muodostus alkaa sillä, että pääsolmuiksi valitut anturisolmut lähettävät yleislähetysviestinä mainosviestin, jolla ne ilmoittavat muille anturisolmuille olevansa pääsolmuja. Anturisolmut valitsevat optimaalisen pääsolmun tiettyjen kriteereiden perusteella ja lähettävät tälle liittymisviestin. Kriteereinä voi olla esimerkiksi anturisolmun ja pääsolmun välinen etäisyys, joka ilmaistaa signaalin vahvuutena tai monenko hypyn päässä anturisolmu on pääsolmusta. Anturisolmu liittyy klusteriin, kun pääsolmu on hyväksynyt sen liittymisviestin. Pääsolmu voi myös hylätä liittymisen esimerkiksi silloin, jos klusterin tiheydelle asetettu raja-arvo ylittyisi liittymisen johdosta. [37]

Klusterin muodostukseen liittyy joitain ominaisuuksia ja klusterointiprotokollat voidaan jaotella niiden mukaan, kuten Gupta ym. jaottelevat artikkelissa [19]. Kuvassa 11 esitetyt ominaisuudet liittyvät klusterin sisäiseen rakenteeseen.



Kuva 11. Klusterin ominaisuudet [19].

Klustereiden lukumäärä kertoo, montako klusteria muodostetaan yhden kierroksen aikana. Mitä suurempi on klustereiden lukumäärä, sitä pienempiä klustereita muodostetaan. Joissakin klusterointiprotokollissa pääsolmujen valinta on ennalta määrätty, jolloin klustereiden määrä pysyy vakiona. Jos taas pääsolmut valitaan satunnaisesti anturisolmujen kesken, voi klustereiden lukumäärä vaihdella.

Klusterin koko määrittää polun maksimipituuden jäsenolmusta pääsolmuun. Pienemmät klusterit ovat energiankulutuksen kannalta parempia, sillä tiedonsiirtoetäisyys pysyy tällöin pienenä ja vähentää pääsolmun kuormaa. Klusterin koko voi olla joko kiinteä tai muuttuva. Joissain klusterointiprotokollissa klusterin koko on kiinteä, jos klusterit ovat pysyvät muuttumattomina koko eliniän. Muulloin klusterin koko voi vaihdella klustereittain.

Klusterin tiheydellä tarkoitetaan klusterin jäsenolmujen määrää suhteessa klusterialueeseen. Tiheiden klustereiden suurena haasteena on saada minimoitua pääsolmujen energian kulutus. Klusterin tiheys voi olla joko kiinteä tai muuttuva. Kiinteää

klusterointia käytävissä klusterointiprotokollissa on klusterin tiheys aina harva, mutta dynaamisissa klusterointiprotokollissa voivat klustereiden tiheydet vaihdella.

Pääsolmun valintaan vaadittava viestimäärä vaihtelee eri klusterointiprotokollissa. Deterministiset klusterointiprotokollat vaativat pääsolmun valintaan aina viestin lähetyksiä. Mitä enemmän viestin lähetyksiä vaaditaan pääsolmun valintaan, sitä enemmän kuluu myös energiaa.

Stabiilisuuden perusteella klusterointiprotokollat jaetaan kiinteisiin ja mukautuviin. Jos klusterin jäsenolmut eivät ole kiinteitä, on tällöin kyseessä mukautuva klusterointiprotokolla. Kiinteissä klusterointiprotokollissa klustereiden määrä ei muutu koko klusterointiprosessin aikana ja lisää näin anturiverkon stabiilisuutta.

Klusterin sisäisen tai klustereiden välinen tiedonsiirto tapahtuu yhden hypyn tai monen hypyn menetelmää käyttäen. Alaluvussa 4.7. käsitellään asiaa tarkemmin.

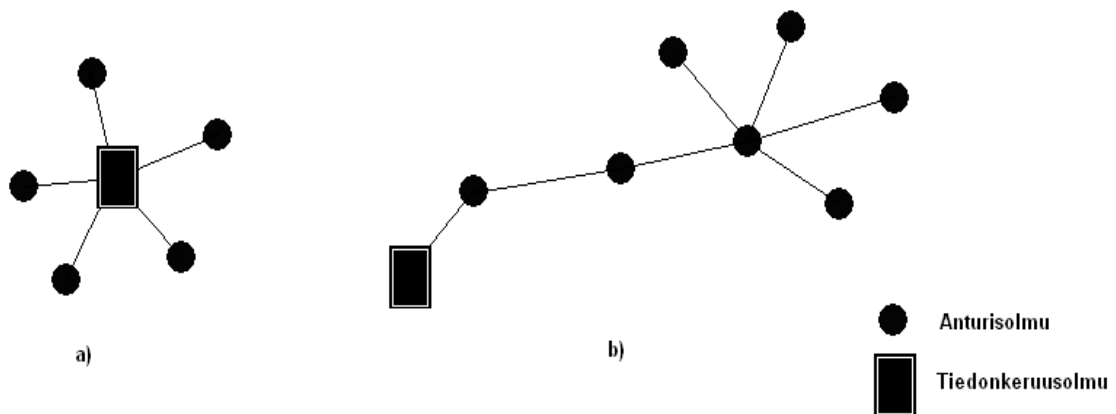
#### 4.6 Datan kokoaminen

Perinteisissä tietoliikenneverkoissa viestintä keskittyy yleensä viestintäpareihin eli datan lähettäjään ja vastaanottajaan. Langattomissa anturiverkoissa sovellukset eivät ole niinkään kiinnostuneita tietyn anturisolmun identiteetistä vaan ympäristöstä mitattavasta informaatiosta. Varsinkin tiheästi sijoitelluissa langattomissa anturiverkoissa useat anturisolmut raportoivat samasta tapahtumasta. Tällöin sovellus ei kuitenkaan ole kiinnostunut siitä miltä anturisolmulta tarkalleen data on peräisin vaan datan sisällöstä. Kun datan sisällöstä ollaan enemmän kiinnostuneita kuin sen lähettäjistä, on kyse datakeskeisestä verkkotoiminnasta. [27]

Todellinen vahvuus datakeskeisessä lähestymistavassa on sen kyky operoida dataa samalla kun sitä kuljetetaan verkossa. Yksinkertaisin esimerkki tällaisesta verkon sisäisestä prosessoinnista on datan kokoaminen. [27] Sen ideana on koota yhteen sisällöltään yhtenevät sanomat ja lähettää pelkästään koottu tieto verkon yli. Näin saadaan eliminoitua ylimääräisiä viestejä ja minimoitua verkossa tapahtuvaa tiedonsiirtoa ja näin ollen myös säästettyä energiaa. [29] Tyypillisiä tapauksia ovat duplikaattiviestien poistaminen sekä minimi-, maksimi- ja keskiarvon laskeminen anturisolmujen mittaamasta datasta [5].

#### 4.6.1 Datan kokoamispisteiden sijoittaminen

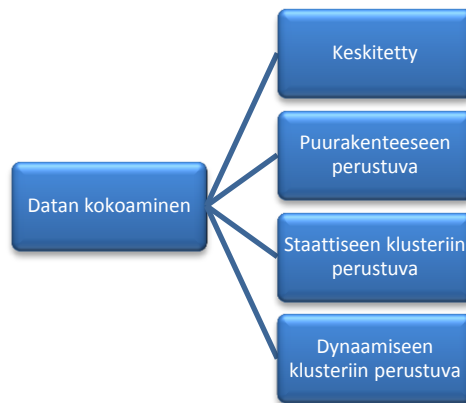
Jotta datan kokoamisesta saadaan maksimi hyöty, täytyy datan kokoamispisteet sijoitella oikein, sillä todellinen datan kokoamisesta saatava hyöty riippuu datalähteen sijainnista suhteessa tiedonkeruusolmuun. Esimerkiksi Kuvassa 12a kaikki datalähteet ovat levittäytyneet tasaisesti, mutta polut datalähteestä tiedonkeruusolmulle eivät kulje ristiin, joten datan kokoaminen ei ole mahdollista muuta kuin tiedonkeruusolmussa. Datan kokoamisesta saadaan suurin hyöty silloin, kun datalähteet ovat lähekkäin, sijaitsevat kaukana tiedonkeruusolmusta ja niiden polut yhdistyvät lähellä datalähteitä, kuten Kuvassa 12b. Datan kokoamisen tulisikin tapahtua lähellä lähteitä ja mahdollisimman varhain. [27]



Kuva 12. Tapaukset joissa datan kokoaminen on turhaa ja hyödyllistä [27].

#### 4.6.2 Datan kokoamiskeemoja

Artikkelissa [41] Park jaottelee datan kokoamiskeemat Kuvan 13 mukaisesti 4 kategoriaan: keskitettyyn, puurakenteeseen perustuvaan, staattiseen klusteriin perustuvaan ja dynaamiseen klusteriin perustuvaan datan kokoamiseen.



Kuva 13. Datan kokoamisskeemojen jaottelu [41].

Keskitetyssä datan kokoamisessa kaikki data paketit kulkevat tiedonkeruusolmulle, joka suorittaa datan kokoamisen. Datan kokoamista ei suoriteta data paketteja välittävillä anturisolmuilla.

Puurakenteeseen perustuvassa datan kokoamisessa datapaketteja välittävät anturisolmut yhdistävät eri lähteiltä tulevia datapaketteja ja lähettävät kootun datapaketin tiedonkeruusolmulle [41]. Artikkelissa [29] on esitelty kolme puurakenteeseen perustuvaa datan kokoamisskeemaa: CNS (*Center at Nearest Source*), SPT (*Shortest Paths Tree*) ja GIT (*Greedy Incremental Tree*). CNS:ssä kaikki lähdesolmut lähettävät datan suoraan tiedonkeruusolmua lähimpänä olevalle lähdesolmulle, joka kokoaa datan ja lähettää sen tiedonkeruusolmulle. SPT:ssä jokainen lähdesolmu lähettää tietonsa tiedonkeruusolmulle lyhintä reittiä pitkin. Päällekkäiset polut yhdistetään muodostamaan kokoava puu. GIT on sekventiaalinen skeema. Siinä ensimmäisessä vaiheessa kokoava puu pitää sisällään pelkästään lyhimmän polun tiedonkeruusolmun ja lähimmän lähdesolmun välillä. Tämän jälkeen jokaisessa vaiheessa liitetään kokoavaan puuhun seuraava lähin lähdesolmu.

Staattiseen klusteriin perustuvassa datan kokoamisessa klusterointi tapahtuu proaktiivisesti eli anturisolmut ryhmitellään klustereihin klusterointiprotokollan alustusvaiheessa. Jokainen klusteri pitää sisällään yhden tai useamman jäsensolmun sekä yhden pääsolmun, joka voi aikatauluttaa tai koota klusterinsa jäsensolmujen lähettämää dataa. [26] Havaitessaan tietyn tapahtuman jäsensolmu lähettää tiedon tapahtumasta pääsolmulle, joka kokoaa saamansa tiedot ja lähettää kootun tiedon tiedonkeruusolmulle [41].

Jungin ym. [26] mukaan staattista klusterointia käyttävässä datan kokoamisessa data saadaan nopeasti ja helposti koottua ja välitettyä tiedonkeruusolmulle ylimääräisen viestiliikenteen pysyessä suhteellisen pienenä. Toisaalta jokaisen pääsolmun täytyy lähettää yleislähetysviestejä aina määräajoin, kun klustereita ollaan muodostamassa uudestaan, ja tämä aiheuttaa paljon ylimääräistä viestintää. On myös mahdollista, että useampi kuin yksi klusteri havaitsee saman tapahtuman. Tällöin datan kokoamisen tehokkuus kärsii. Staattinen klusterointi saavuttaa siis parhaat tulokset datan kokoamisessa, jos tapahtuma havaitaan pelkästään yhdessä klusterissa. Tätä staattisten klusterointitekniikoiden ongelmaa voidaan lieventää käyttämällä usean hypyn klustereita. Tällä tavalla suurennetaan klustereiden kokoja ja vähennetään klustereiden kokonaismäärää. Täten todennäköisyys, että saman tapahtuman havaitsee useampi klusteri samanaikaisesti, pienenee. Toisaalta käyttämällä usean hypyn klustereita syntyy myös ylimääräistä viestintää, sillä data voi joutua kulkemaan usean anturisolmun kautta pääsolmulle.

Dynaamiseen klusteriin perustuvassa datan kokoamisessa ei tarvita klusterointiprotokollissa alustusvaihetta luomaan klustereita. Klusterit luodaan reaktiivisesti eli pelkästään silloin, kun anturisolmut havaitsevat halutun tapahtuman. [41] Tällöin klusterin jäsenet äänestävät pääsolmun jäsenten joukosta esimerkiksi sen perusteella, että kenellä on eniten energiaa tai on lähimpänä havaittua tapahtumaa. Valittu pääsolmu suorittaa datan kokoamisen paikallisesti ja lähettää kootun datapaketin tiedonkeruusolmulle. Tämän tekniikan etuna on se, että pelkästään välttämättömät anturisolmut osallistuvat datan kokoamiseen ja säästävät näin huomattavasti muiden anturisolmujen energiaa. Myös ohjausviesteistä aiheutuva ylimääräinen viestintä pysyy pienenä, sillä pelkästään valittu pääsolmu lähettää yleislähetysviestejä. Lisäksi dynaamista klusterointia käyttävässä datan kokoamisessa datan kokoamisnopeus on erittäin korkea, sillä datan kokoaminen suoritetaan lähellä havaittua tapahtumaa eikä datan tarvitse kulkea useaa hyppyä pääsolmulle. [26]

Dynaamiseen klusteriin perustuva datan kokoaminen aiheuttaa Jungin ym. [26] mukaan myös joitain vakavia ongelmia. Ensinnäkin klusterit muodostetaan vasta kun havaitaan jokin tapahtuma, joten pääsolmun valinta aiheuttaa ylimääräistä viivettä ennen kuin dataa

voidaan lähettää pääsolmulle koottavaksi. Datan kokoaminen on mahdollista vasta klusterin valintavaiheen päätyttyä. Myös nopeasti liikkuva havainnoitava kohde aiheuttaa ongelmia dynaamisessa klusteroinnissa. Jos kohde liikkuu nopeasti, täytyy dynaaminen klusteri luoda usein. Aina kun kohde liikkuu pääsolmun havainnointialueelta pois, täytyy uusien kohteen havainnoivien anturisolmujen käynnistää klusterin valintavaihe uudestaan. Tämä aiheuttaa viivettä ja kasvattaa verkon ylimääristä liikennettä.

#### **4.6.3 Datan kokoamisesta johtuva viive**

Datan kokoaminen vähentää verkossa tapahtuvaa tiedonsiirtoa ja näin ollen myös anturisolmujen energiankulutusta, mutta saattaa aiheuttaa verkkoon enemmän viivettä [29]. Jos esimerkiksi kokoamispisteessä joudutaan odottamaan dataa kauempana olevilta lähteiltä, jotta se saadaan yhdistettyä lähemmiltä lähteiltä saatuun dataan. Datan kokoamista suorittavan anturisolmun kuin myös tiedonkeruusolmun täytyy päättää, kauanko dataa odotetaan sen jokaiselta lapsisolmulta. [27]

Yksinkertaisimmassa tapauksessa datan kokoamista suorittava anturisolmu tietää ketkä sen naapurisolmuista on sen lapsisolmuja ja odottaa niiden jokaisen vastausta. Tämä saattaa kuitenkin viedä kauan aikaa johtuen radiokanavan huojunnasta, väliaikaisesta anturiviasta tai yksinkertaisesti erittäin tasapainottomasta puurakenteesta. Odottamalla pitkään saadaan enemmän dataa mukaan datan kokoamiseen ja näin ollen lisättyä kootun datan tarkkuutta. Odottaminen lisää viivettä ja mahdollisesti myös anturisolmujen energiankulutusta johtuen radiovastaanottimen joutokäynnistä. [27]

Haastavammassa tapauksessa aika, joka anturisolmulta kuluu mittaamansa tai kokoamansa datan välitykseen isäsolmulle, on satunnainen muuttuja. Jos odottamiseen liittyy myös jokin kustannus, on kyseessä yleisempi optimaalisen pysähtymissäännön ongelma. Artikkelissa [10] Broder ja Mitzenmacher ovat tutkineet tätä ongelmaa tarkemmin. [27]

### **4.7 Klusterin sisäinen ja klustereiden välinen tiedonsiirto**

Jokaisen klusterointitekniikan tärkeä tavoite on verkon liitettävyyys. Klusterin sisäinen tiedonvälitys tapahtuu anturisolmun ja pääsolmun välillä joko suoraan yhden hypyn menetelmällä tai käyttäen monen hypyn menetelmää. [49] Suora viestintä ei aina ole

mahdollista, varsinkin jos anturisolmun toimintasäde on rajallinen ja pääsolmujen määrä on rajattu. Tällöin anturisolmun ja pääsolmun välinen data joudutaan välittämään usean anturisolmun kautta eli käyttämään monen hypyn menetelmää. [1]

Myös klustereiden välinen tiedonsiirto voidaan hoitaa joko yhden hypyn tai monen hypyn menetelmää käyttäen. Jos pääsolmu kykenee pitkänmatkan viestintään, voidaan yhteys pääsolmusta tukiasemaan muodostaa suoraan. Muutoin reitti pääsolmusta tukiasemaan kulkee muiden pääsolmujen ja/tai anturisolmujen kautta. [44] Younis ym. [49] esittelevät kaksi klustereiden välisessä tiedonsiirrossa omaksuttua lähestymistapaa.

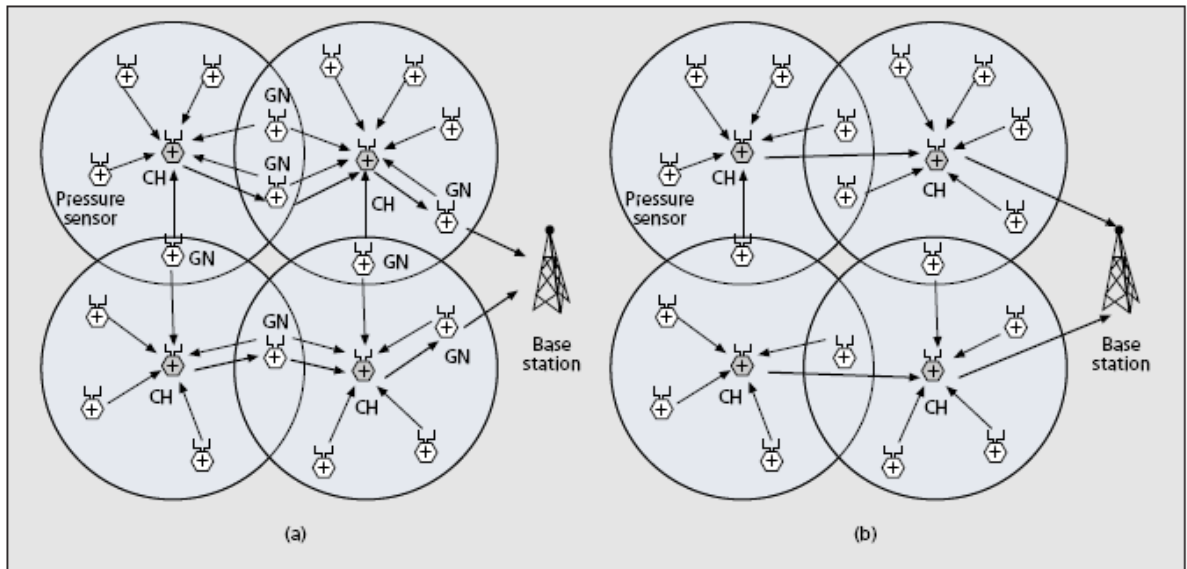
Ensimmäisessä lähestymistavassa pääsolmujen tai pääsolmun ja tukiaseman välinen reititys tapahtuu klusterin reunalla olevien anturisolmujen kautta. Tällöin klusterin reunalla olevat anturisolmut toimivat yhdyskäytävinä joko klustereiden tai klusterin ja tukiaseman välillä. Tällaisia anturisolmuja kutsutaan gateway-solmuiksi. Tämä lähestymistapa sopii verkoille, jotka käyttävät kiinteää lähetystehoä. Verkon täytyy kuitenkin olla tarpeeksi tiheä, jotta klustereiden yhteisellä peittoalueella on tarpeeksi gateway-solmuja välittämään klustereiden välistä dataa. Kuvassa 14a on esitetty reititys käyttäen gateway-solmuja.

Toisessa lähestymistavassa reitityksestä vastaa pelkästään klustereiden pääsolmut. Eli pääsolmun ja tukiaseman välinen tiedonsiirto tapahtuu joko suoraan tai muiden pääsolmujen kautta. Tämä lähestymistapa on sopiva jos:

- anturisolmu voi säätää lähetystehoaan,
- pääsolmujen tiheys ja klustereiden välinen tiedonsiirtoetäisyys täyttää Guptan ja Kumarin [18] määrittämät liitettävyysehdot.

Tämän lähestymistavan etuna on se, että kaikki klusterin tavalliset anturisolmut voidaan asettaa nukkumaan, kun ne eivät mittaa tai lähetä dataa. Kuvassa 14b on esitetty reititys, jossa käytetään pelkästään klustereiden pääsolmuja.





Kuva 14. Klustereiden välinen tiedonsiirto [49].

## 5 Klusterointiprotokollat

Tässä luvussa esitellään ehkä langattomien anturiverkkojen tunnetuin ja eniten tutkittu klusterointiprotokolla LEACH ja sen laajennus MH-LEACH sekä uudempi protokolla DSBCA. Näiden protokollien toimintaa ja sopivuutta kivimurskaimen meluverkkoon tarkastellaan myöhemmin alaluvussa 6.3. Nämä protokollat valittiin siksi, että haluttiin verrata yksinkertaisempaa todennäköisyyteen perustuvaa MH-LEACH -protokollaa DSBCA-protokollaan, jossa pääsolmun valintaan vaikuttaa anturisolmujen jäljellä oleva energia, yhteystiheys sekä monestiko anturisolmu on ollut pääsolmuna. MH-LEACH protokollan valintaan vaikutti myös se, että siinä klustereiden välinen tiedonsiirto tapahtuu monen hypyn menetelmällä. Tässä luvussa viitataan aina alaluvun alussa protokollan nimen yhteydessä mainittuun lähteeseen, ellei toisin ole mainittu.

### 5.1 LEACH

LEACH (*Low Energy Adaptive Clustering Hierarchy*) [21] on ehkä tunnetuin langattomille anturiverkoille suunniteltu itseorganisoituva ja adaptiivinen klusterointiprotokolla. Siinä anturisolmut muodostavat paikallisia klustereita, joissa yksi anturisolmu kerrallaan toimii klusterin pääsolmuna. Klusterin pääsolmua kierrätetään satunnaisesti, jotta anturisolmujen energiankulutus jakautuisi tasaisesti kaikkien anturisolmujen kesken ja verkon elinikä pitenisi. LEACH-protokollassa tehdään seuraavat olettamukset: verkon tukiasema on kiinteä ja sijaitsee kaukana muista anturisolmuista, anturisolmut ovat toiminnoiltaan ja ominaisuuksiltaan yhteneviä ja energiarajoitteisia sekä kykeneviä kommunikoimaan suoraan tukiaseman kanssa.

LEACH-protokollan toiminta on jaettu kierroksiin ja jokainen kierros alkaa järjestelyvaiheella ja päättyy vakaantilan vaiheeseen. Järjestelyvaiheessa valitaan klustereiden pääsolmut ja muodostetaan klusterit sekä luodaan siirtoaikataulut. Järjestelyvaiheen jälkeen on vuorossa vakaantilan vaihe, jolloin varsinainen tiedonsiirto anturisolmuilta tukiasemalle tapahtuu. Vakaantilan vaihe on kestoaltaan pidempi kuin järjestelyvaihe, jotta verkon overheadin määrä saadaan minimoitua.

### 5.1.1 Pääsolmun valinta ja kierrätys

Järjestelyvaiheessa kun klustereita ollaan muodostamassa, kukin anturisolmu päättää tuleeko siitä kyseisellä kierroksella klusterin pääsolmu vai ei. Tämä päätös perustuu ennalta määritettyyn pääsolmujen ja verkon anturisolmujen kokonaismäärän suhteeseen sekä kuinka monesti anturisolmu on aiemmin ollut jo pääsolmuna. Anturisolmu  $n$  valitsee satunnaisluvun väliltä  $(0,1)$ , jota verrataan tämän jälkeen kynnyksiarvoon  $T(n)$ . Anturisolmusta tulee pääsolmu kierrokselle  $r$ , jos sen valitsema satunnaisluku on pienempi kuin kynnyksiarvo. Kynnyksiarvo lasketaan seuraavasti:

$$T(n) = \begin{cases} \frac{P}{1 - P * \left(r \bmod \frac{1}{P}\right)}, & \text{jos } n \in G \\ 0, & \text{muuten} \end{cases}, \quad (1)$$

missä  $P$  on pääsolmujen osuus anturisolmujen kokonaismäärästä (esim. 0.05),  $r$  on nykyinen kierros ja  $G$  on anturisolmujen joukko, joita ei ole valittu pääsolmuiksi edellisten kierrosten aikana. Käyttämällä tätä kynnyksiarvoa jokainen anturisolmu tulee valituksi pääsolmuksi  $\frac{1}{P}$  kierroksen sisällä. Jokaisella anturisolmulla on siis mahdollisuus tulla valituksi pääsolmuksi todennäköisyydellä  $P$  kierroksen  $0$  aikana. Anturisolmut, jotka ovat pääsolmuja kierroksella  $0$ , eivät voi olla pääsolmuja seuraavien  $\frac{1}{P}$  kierroksen aikana. Näin ollen todennäköisyys, että jäljellä olevista anturisolmuista tulee pääsolmuja kasvaa, koska mahdollisia pääsolmuehdokkaita on vähemmän.  $\frac{1}{P} - 1$  kierroksen jälkeen  $T$  on  $1$  jokaiselle anturisolmulle, joka ei ole ollut vielä pääsolmu, joten ne tulevat valituiksi pääsolmuiksi.  $\frac{1}{P}$  kierroksen jälkeen kierrokset aloitetaan alusta, jolloin jokaisen anturisolmun on mahdollista tulla jälleen valituksi pääsolmuksi.

### 5.1.2 Klusterin muodostus ja siirtoaikataulun luonti

Ilmoitusvaiheen jälkeen jokainen pääsolmuksi valittu anturisolmu lähettää kullakin kierroksella muille verkon anturisolmuille ilmoitusviestin, jotta muut anturisolmut tietävät mikä anturisolmu toimii kyseisellä kierroksella pääsolmuna. Ilmoitusviestin lähetyksessä pääsolmut käyttävät CSMA (*Carrier Sense with Multiple Access*) pohjaista MAC-protokollaa ja kaikki pääsolmut lähettävät ilmoitusviestinsä käyttäen samaa lähetystehoä.

Ilmoitusvaiheen aikana jäsensolmujen täytyy pitää radiovastaanottimensa päällä kuullakseen kaikkien pääsolmujen ilmoitusviestit. Jäsensolmu päättää ilmoitusviestien signaalinvahvuuksien perusteella sellaisen klusterin, jonka pääsolmun kanssa kommunikointi kuluttaa vähiten energiaa. Jäsensolmujen täytyy ilmoittaa pääsolmulle liittymisistään klusteriin, joten jokainen jäsensolmu lähettää kyseisen tiedon pääsolmulle käyttäen taas CSMA MAC -protokollaa. Tämän vaiheen aikana kaikkien pääsolmujen täytyy pitää radiovastaanottimensa päällä. Jäsensolmujen määrän perusteella pääsolmut luovat TDMA-aikataulut, milloin kukin klusterin jäsensolmu voi lähettää dataa pääsolmulle. Aikataulutieto lähetetään klusterin jäsensolmuille yleislähetysviestinä.

### **5.1.3 Tiedonsiirtovaihe**

Tiedonsiirto voi alkaa, kun klusterit on muodostettu ja TDMA-aikataulut luotu. Jäsensolmut lähettävät aina datan pääsolmulle niille varatun siirtoajan aikana. Jokainen jäsensolmu voi sammuttaa radionsa siirtoaikansa loputtua ja käynnistää sen seuraavan siirtoaikansa alussa. Näin jäsensolmujen energiakulutusta saadaan vähennettyä ja elinikää pidennettyä. Pääsolmujen täytyy pitää vastaanottimensa aina päällä, jotta se voi vastaanottaa dataa klusterin jokaiselta jäsensolmulta. Pääsolmu kokoaa klusterinsa jäsensolmuilta vastaanottamansa datan ja pakkaa sen yhteen ainoaan viestiin ja lähettää sen tukiasemalle. Tiedonsiirto pääsolmulta tukiasemalle kuluttaa paljon energiaa, sillä LEACH-protokollassa oletetaan että tukiasema sijaitsee kaukana muista anturisolmuista. Datan kokoamisella saavutetaankin huomattava energiansäästö, sillä tiedonsiirto pääsolmujen ja tukiaseman välillä vähenee huomattavasti. Tiedonsiirtovaiheen jälkeen alkaa seuraava kierros ennalta määritetyn ajan kuluttua. Seuraava kierros alkaa taas pääsolmujen valinnalla ja klustereiden muodostuksella.

LEACH-protokollassa käytetään CDMA (*Code Division Multiple Access*) koodijakokanavointia, jolla pyritään vähentämään viereisten klustereiden radioliikenteen aiheuttamaa häiriötä. CDMA on radiotien kanavanvaraustekniikka, missä klusterit erotetaan toisistaan yksilöllisellä hajautuskoodilla. Ryhtyessään pääsolmuksi, anturisolmu valitsee hajautuskoodin ja informoi klusterin anturisolmuja käyttämään tätä koodia lähettäessään dataa pääsolmulle. Pääsolmu vastaanottaa pelkästään sellaisia signaaleja,

jotka sisältävät pääsolmun valitseman koodin. Koodijakokanavoinnin ansiosta klusterin sisäinen tiedonsiirto ei häiriinny viereisten klustereiden tiedonsiirrosta.

#### **5.1.4 Vahvuudet ja heikkoudet**

LEACH-protokolla sisältää monia hyviä ominaisuuksia pienentämään energiankulutusta ja samalla pidentämään verkon elinikää. Se on täysin hajautettu lähestymistapa eikä vaadi globaalia tietoa verkosta. Klusterin sisäinen tiedonsiirto tapahtuu suoraan jäsensolmun ja pääsolmun välillä, joten jäsensolmut eivät kuluta energiaa datan välitykseen. Pääsolmun suorittama datan kokoaminen pienentää verkon liikennettä, joten liikennemäärä pysyy kohtuullisena laajoissakin verkoissa. Tämän lisäksi myös pääsolmujen kierrätys ja jäsensolmun radion sammuttaminen oman siirtoaikansa ulkopuolella pidentää verkon elinikää. Protokolla ei myöskään tarvitse anturisolmujen paikkatietoja klustereiden muodostukseen, joten se on tehokas ja yksinkertainen. [39]

Artikkelissa [34] Liu esittelee LEACH-protokollan heikkouksia. LEACH-protokollassa klusterin ja tukiaseman välinen tiedonsiirto tapahtuu suoraan yhdellä hypyllä, joten se ei sovellu maantieteellisesti laaja-alaisiin verkkoihin. Ei myöskään ole realistista olettaa, että jokainen anturisolmu voi kommunikoida suoraan kaukana sijaitsevan tukiaseman kanssa. Lisäksi pitkänkantaman viestintä suoraan pääsolmulta tukiasemalle voi kasvattaa energiankulutuksen liian suureksi. LEACH jakaa klusterin anturisolmujen kuormaa kierrättämällä pääsolmua, mutta se ei kuitenkaan huomioi pääsolmun valinnassa anturisolmujen energiatasoja. Pienemmän energian omaavat anturisolmut toimivat pääsolmuna yhtä monta kierrosta kuin enemmän energiaa omaavat anturisolmut ja kuolevat näin ollen ennenaikaisesti. Pääsolmujen valinta tehdään todennäköisyyteen perustuen, joten ne eivät välttämättä jakaannu tasaisesti koko verkon alueella. Näin ollen on mahdollista, että kaikki pääsolmut keskittyvät tietylle verkon alueelle ja joidenkin anturisolmujen läheisyydessä ei ole yhtään pääsolmua. Lisäksi dynaaminen klusterointi lisää ylimääräistä verkkoliikennettä ja esimerkiksi pääsolmujen kierrätys ja pääsolmujen ilmoitusviestien lähettäminen kuluttavat energiaa.

LEACH-protokolla on hyvin tunnettu ja sitä on myös tutkittu paljon. Näin ollen siitä löytyy paljon erilaisia paranneltuja versioita eri käyttötarkoituksiin. Esimerkiksi E-LEACH

-protokollassa pääsolmun valinnassa huomioidaan anturisolmun jäljellä oleva energia ja seuraavassa alaluvussa esiteltävässä MH-LEACH -protokollassa viestintä pääsolmulta tukiasemalle tapahtuu monen hypyn menetelmällä.

### 5.1.5 MH-LEACH

MH-LEACH (*Multi-Hop Low Energy Adaptive Clustering Hierarchy*) [38] on paranneltu versio LEACH-protokollasta, jonka päätavoitteena on muodostaa yhteys klustereiden pääsolmujen ja tukiaseman välille käyttäen monen hypyn menetelmää. Sen päätarkoituksena on lähettää data klusterin pääsolmulta tukiasemalle lähimmän pääsolmun kautta, joka välittää sen edelleen kohti tukiasemaa. Näin saadaan vähennettyä pääsolmun virrankulutusta ja pidennettyä verkon käyttöikä, sillä viestin lähettäminen kauas tukiasemalle vie enemmän virtaa kuin sen lähettäminen lähimmälle pääsolmulle. Myös verkon maantieteellinen skaalautuvuus paranee huomattavasti verrattuna LEACH-protokollaan.

### Energiankulutusmalli

Anturisolmun energiankulutus lasketaan seuraavasti.  $k$  bitin kokoisen viestin lähetyksen etäisyyden  $d$  päähän kuluttaa energiaa

$$E_{Tx}(k, d) = E_{Tx-elec}(k) + E_{Tx-amp}(k, d) \quad (2)$$

$$E_{Tx}(k, d) = \begin{cases} E_{elec} * k + \epsilon_{fs} * k * d^2, & d < d_0 \\ E_{elec} * k + \epsilon_{mp} * k * d^4, & d \geq d_0 \end{cases} \quad (3)$$

ja viestin vastaanottaminen kuluttaa energiaa

$$E_{Rx}(k) = E_{Rx-elec}(k) = E_{elec} * k, \quad (4)$$

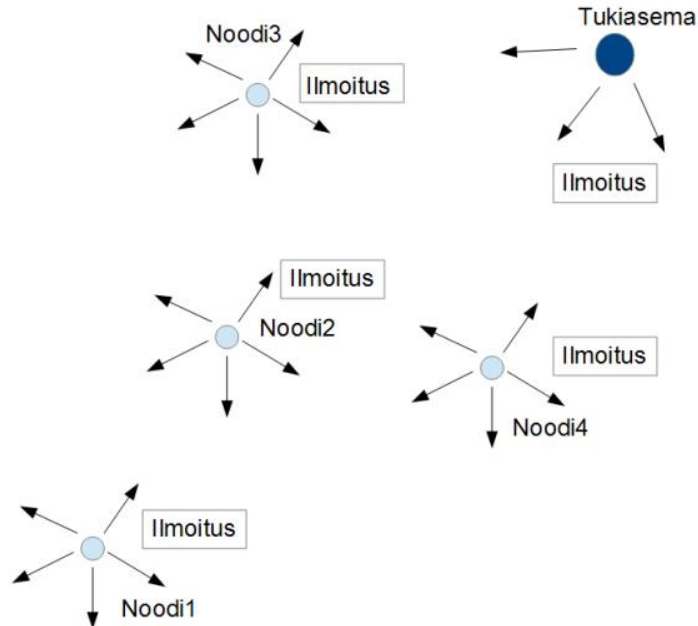
missä  $E_{Tx-elec}$  on lähetykseen kulutettu energia,  $E_{Rx-elec}$  on vastaanottamiseen kulutettu energia,  $E_{Tx-amp}$  on kulutettu energia lähetyksen vahvistamiseen,  $d_0$  on kynnysetäisyys arvojen  $E_{elec}, \epsilon_{fs}, \epsilon_{mp}$  mukaan,  $\epsilon_{fs}$  on vapaan tilan mallin mukainen parametri ( $fs$ ), jota käytetään jos etäisyys lähteestä kohteeseen on vähemmän kuin  $d_0$ ,  $\epsilon_{mp}$  on monitiemallin

mukainen parametri ( $mp$ ), jota käytetään jos etäisyys lähteestä kohteeseen on enemmän tai yhtä paljon kuin  $d_0$  ja  $E_{elec}$  on energiankulutus per bitti lähetyksessä tai vastaanotossa.

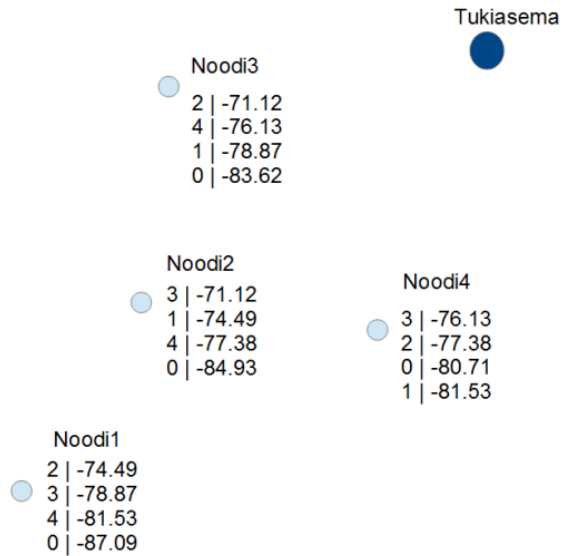
### Algoritmin toiminta

Yksi algoritmin päätavoitteista on löytää pääsolmulle mahdolliset reitit kohti tukiasemaa ja säästää energiaa käyttämällä muita pääsolmuja välittäjäsolmuina lähettäessään data-paketteja kohti tukiasemaa. Seuraavan pääsolmun valinnassa täytyy huomioida onko sillä tarpeeksi energiaa. Jos pääsolmu ei voi lähettää data-pakettia kyseisen pääsolmun kautta, yrittää se löytää toisen mahdollisen pääsolmun.

Vaiheessa 1 pääsolmut valitaan kuten LEACH-protokollassa. Tämän jälkeen pääsolmut lähettävät yleislähetysviestinä ilmoitusviestin (Kuva 15), jolloin kaikki kuuluvuusalueella olevat pääsolmut muodostavat reititystaulut vastaanotetun signaalinvoimakkuuden ( $RSSI$ ) mukaan. Myös tukiasema lähettää ilmoitusviestin ja suorittaa samat toiminnot kuin pääsolmut.



Kuva 15. MH-LEACH, Pääsolmujen ja tukiaseman tunnistus [38].



Kuva 16. MH-LEACH, alustavien reititystaulujen luonti, vastaanotetun signaalinvahvuus  $RSSI$  (dBm) [38].

Esimerkiksi Kuvassa 16 pääsolmu 3 vastaanottaa ilmoitusviestin kolmelta pääsolmulta (2, 4 ja 1) ja tukiasemalta (0). Tukiasemalta vastaanotetun signaalin  $RSSI$  arvo on esimerkiksi -83,62 dBm.

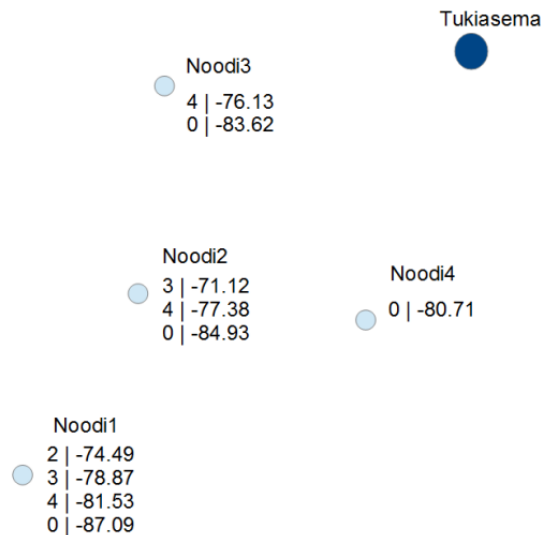
Vaiheessa 2 jokainen pääsolmu lähettää alustavan reititystaulunsa tukiasemalle, joka tarkistaa voiko pääsolmu olla toisen pääsolmun reitillä. Tämän tarkistuksen jälkeen tukiasema lähettää korjatut reitit takaisin anturisolmuille. Tämä toiminto on tarpeen, sillä alustavat reititystaulut eivät ole välttämättä optimaalisia. Esimerkiksi anturisolmulla 3 lähin pääsolmu on anturisolmu 2 ja anturisolmulla 2 taas pääsolmu 3. Tämä aiheuttaisi verkkoon silmukan, eikä data-paketti saavuttaisi koskaan tukiasemaa. Anturisolmun 2 reititystaulusta löytyy myös toinen ongelma. Siinä toiseksi lähin pääsolmu on anturisolmu 1, joka ei ole olisi hyvä vaihtoehto sillä tällöin data-paketti lähetettäisiin vastakkaiseen suuntaan tukiasemasta. Vaiheen 2 algoritmi on seuraava

$IF\{RSSI_X < RSSI_Y\} THEN$   
 “Anturisolmu Y on mahdollinen reitti anturisolmulle X”,

missä  $RSSI_X$  on anturisolmun X tukiasemalta vastaanottaman paketin  $RSSI$  ja  $RSSI_Y$  on anturisolmun Y tukiasemalta vastaanottaman paketin  $RSSI$ .



Jos anturisolmu Y vastaanottaa tukiaseman ilmoitusviestin suuremmalla *RSSI* arvolla kuin anturisolmu X, tulkitaan anturisolmun Y olevan lähempänä tukiasemaa kuin anturisolmu X ja näin ollen anturisolmu Y voi olla anturisolmun X reitillä. Esimerkiksi anturisolmun 2 tukiaseman *RSSI* arvo on -84.93dBm ja anturisolmun 3 *RSSI* arvo on -83.62dBm, joten anturisolmu 3 on mahdollinen reitti anturisolmulle 2 sillä  $-84.93\text{dBm} < -83.62\text{dBm}$ .



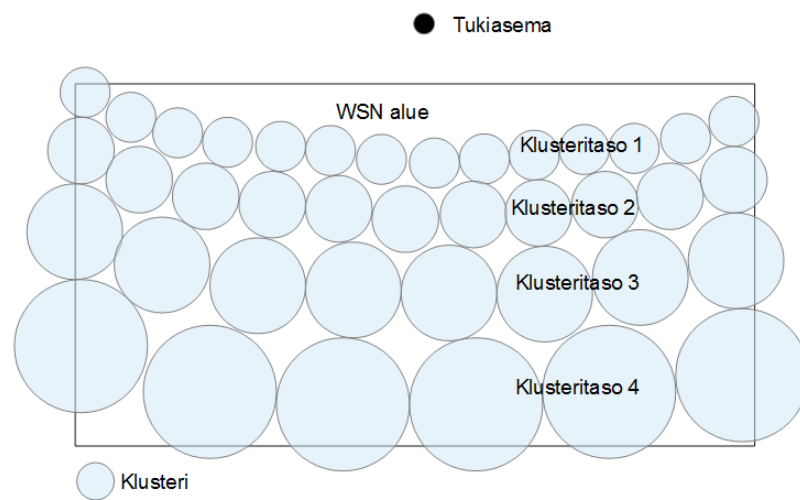
Kuva 17. MH-LEACH, korjatut reititystaulut [38].

## 5.2 DSBCA

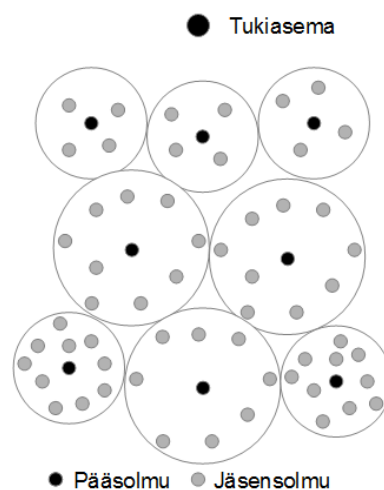
DSBCA (*a Balanced Clustering Algorithm with Distributed Self-Organization for Wireless Sensor Networks*) [32] -protokolla on esitelty vuonna 2013. Siinä klustereiden muodostus huomioi sekä tukiaseman etäisyyden että anturisolmujen tiheyden, koska sen tarkoituksena on luoda energian suhteen tasapainoisempia klustereita ja välttää myös klustereita, joissa on liikaa anturisolmuja. Kaikkien klustereiden täytyy myös kommunikoida tukiaseman kanssa, mutta pitkänkantaman langaton viestintä kuluttaa enemmän energiaa, joten DSBCA-protokollassa klusterit välittävät dataa kauempana sijaitsevilta klustereilta kohti tukiasemaa. Reititystaulujen muodostamiseen ei artikkelissa [32] kuitenkaan oteta kantaa.

Tasaisesti jakautuneessa verkossa DSBCA muodostaa klusterikerroksia, kuten Kuvassa 18, missä kauempana tukiasemasta sijaitsevat klusterikerrokset ovat säteeltään suurempia ja

Samalla tasolla sijaitsevat klusterit ovat säteeltään identtisiä. Kuvassa 19 esitetyssä epätasaisesti jakautuneessa verkossa klustereiden kokoon vaikuttaa kaksi tekijää: etäisyys tukiasemasta sekä yhteystiheys. Mitä kauempana klusteri sijaitsee tukiasemasta ja mitä pienempi on klusterin yhteystiheys, sitä suurempi on klusterin säde. Ja päinvastoin, mitä lähempänä klusteri on tukiasemaa ja mitä suurempi yhteystiheys, sitä pienempi on klusterin säde. Toisaalta jos anturisolmujen yhteystiheys on kauempana tukiasemasta suurempi, voi lähempänä tukiasemaa olevat klusterit olla säteeltään suurempia kuin kaukana tukiasemasta sijaitsevat klusterit.



Kuva 18. DSBCA tasaisesti jakautuneessa verkossa [32].



Kuva 19. DSBCA epätasaisesti jakautuneessa verkossa [32].

DSBCA-protokollassa tehdään seuraavat olettamukset. Kaikki anturisolmut muodostavat verkon topologian itseorganisoituvasti sekä lähettävät dataa samalla signaalin voimakkuudella ja taajuudella, jolloin signaalin maksimikantama  $L$  on sama kaikilla anturisolmuilla. Anturisolmujen tilanvaihdot, kuten anturisolmun kuoleminen, hillitsevät algoritmin päivitysaikaa. Jokaisen anturisolmun sijainti on kiinteä tietyllä ajanjaksolla ja kaikki anturisolmut ovat jakaantuneet Poisson prosessin mukaan intensiteetillä  $\lambda$ .

DSBCA-protokolla voidaan jakaa kolmeen vaiheeseen: pääsolmun valintaan, klustereiden muodostukseen sekä pääsolmun kierrätysvaiheeseen.

### 5.2.1 Pääsolmun valinta

Aivan aluksi DSBCA valitsee satunnaiset anturisolmut laukaisemaan klusterointiprosessin. Tämän jälkeen valittu anturisolmu  $u_t$  laskee yhteystiheyden sekä etäisyyden tukiasemasta päättääkseen klusterin säteen  $k$  ja siitä tulee väliaikainen pääsolmu. Yhteystiheydellä tarkoitetaan anturisolmun  $u_t$   $k$ -hypyn naapuruston linkkien lukumäärän suhdetta anturisolmujen lukumäärään. Säde  $k$  lasketaan seuraavasti

$$k = \text{floor} \left[ \frac{\beta D(u_t)}{D_k(u_t)} \right], \quad (5)$$

missä  $D(u)$  on anturisolmun  $u$  etäisyys tukiasemasta,  $D_k(u)$  on anturisolmun  $u$  yhteystiheys,  $\beta$  on sovelluksen määrittämä anturiparametri ja  $\text{floor}$  pyöristyksen laskeminen. Etäisyys  $D(u)$  voidaan arvioida seuraavasti

$$D(u) = 10 \frac{|RSSI - A|}{10 \cdot n}, \quad (6)$$

missä  $RSSI$  ilmaisee vastaanotetun signaalin voimakkuuden,  $A$  on signaalin voimakkuus 1 metrin etäisyydellä tukiasemasta ja  $n$  on signaalin etenemisvakio. Anturisolmun  $u$   $k$ -hypyn naapurit  $N_k(u)$  määritellään seuraavasti

$$N_k(u) = \{v \in V | v \neq u \wedge d(u, v) \leq k\}, \quad (7)$$

missä  $d(u, v)$  on hyppyjen määrä anturisolmun  $u$  ja anturisolmun  $v$  välillä. Anturisolmun yhteystiheys lasketaan seuraavasti

$$D_k(u) = \frac{|\{(t, v) \in E \mid t, v \in N_k(u) \cup \{u\}\}|}{|N_k(u)|}, \quad (8)$$

missä  $|N_k(u)|$  on anturisolmun  $u$   $k$ -hypyn päässä olevien naapurisolmujen lukumäärä. Tässä algoritmin vaiheessa pääsolmuksi valitaan anturisolmu, jonka paino on suurin  $k$ -hypyn naapurustossa. Anturisolmun laskettavaan painoon vaikuttaa anturisolmun jäljellä oleva energia, yhteystiheys sekä monestiko anturisolmu on ollut pääsolmuna. Näin saadaan muodostettua energialtaan sekä sijainniltaan tasapainoisempia klustereita. Anturisolmun paino lasketaan seuraavasti

$$W(u) = \phi P[D_k(u)] + \varphi P\left[\frac{R_e(u)}{E(u)}\right] - \gamma P[H(u)], \quad (9)$$

$$0 \leq \phi, \varphi, \gamma, \leq 1, \gamma < \phi + \varphi < 1$$

missä  $\phi, \varphi, \gamma$  ovat sovelluksen määrittämät vaikutuskertoimet,  $R_e(u)$  on anturisolmun  $u$  jäljellä oleva energia,  $E(u)$  on anturisolmun  $u$  energia alussa ja lukumäärä  $H(u)$  kertoo kuinka monta kertaa anturisolmu  $u$  on tullut valituksi pääsolmuksi.

Aloitusvaiheessa anturisolmu  $u_t$  liipaisee klusterointiprosessin ja lähettää Hello-viestejä sen  $k$ -hypyn päässä oleville naapurisolmuille. Naapurisolmut laskevat suhteellisen painonsa  $W(u)$ , jonka jälkeen painoltaan suurin anturisolmu valitaan pääsolmuksi. Tämän jälkeen pääsolmut lähettävät  $k$ -hypyn päässä oleville naapurisolmuilleen yleislähetysviestinä Head\_message:n, jossa ne ilmoittavat olevansa pääsolmu ja pyytävät liittymään klusteriinsa. Head\_message sisältää pääsolmun tunnisteen HID:n, lähettävän anturisolmun tunnisteen SID:n sekä tiedon monenko hypyn päässä pääsolmu sijaitsee (HD). Algoritmi hylkää kauempaa kuin  $k$ -hypyn päästä tulevat Head\_message:t. Näin voidaan varmistaa, että klustereiden koko ei kasva  $k$ -hyppyä suuremmaksi. Anturisolmut liittyvät klusteriin lähettämällä Join\_message-viestin. Jos anturisolmu kuuluu jo johonkin klusteriin ja vastaanottaa tällöin Head\_message:n, liittyy se uuteen klusteriin vain jos sen paino on pienempi kuin nykyisessä klusterissa. Koska Head\_message lähetetään vain sen  $k$ -hypyn päässä sijaitseville naapurianturisolmuille, on mahdollista että jotkin anturisolmut

eivät vastaanota koskaan Head\_message:a. Anturisolmu julistaa itsensä pääsolmuksi jos se ei vastaanota Head\_message:a ajassa  $T(w)$  ( $T(w) < T(k)$ ), missä  $T(w)$  on odotusaika ja  $T(k)$  on päivitysaika eli aika jonka jälkeen klusterointi suoritetaan uudestaan.  $T(w)$  ja  $T(k)$  ajat tulisi määrittää siten että verkon jokainen anturisolmu löytää oman pääsolmunsaa ja että klusterointiprosessi suoritetaan uudestaan aina ajan  $T(k)$  välein.

### 5.2.2 Klustereiden muodostus

DSBCA asettaa kynnyksarvon klusterin koolle. Klusterin anturisolmujen määrä ei voi ylittää tätä kynnyksarvoa. Näin vältetään muodostamasta suuria klustereita, jotka aiheuttaisivat ylimääräistä overheadia. Pääsolmu vertaa klusterinkokoa kynnyksarvoon, kun se vastaanottaa Join\_message-viestin anturisolmulta. Uusi jäsenolmu hyväksytään klusteriin, jos klusterin koko on pienempi kuin kynnyksarvo, muuten pyyntö hylätään. Klusterointiprosessi lopetetaan, jos hylätyllä anturisolmulla on jo pääsolmu. Muuten se etsii toisen sopivan klusterin johon liittyä.

Jokainen klusterin jäsenolmu ylläpitää klusterin tiedoista taulua, johon se tallettaa esimerkiksi HID:n, SID:n, HD:n sekä muuta tarpeellista tietoa. Anturisolmut päivittävät taulua vastaanottaessaan datapaketteja. Esimerkiksi jos anturisolmu vastaanottaa paketin, jonka HD on pienempi kuin sen ylläpitämässä taulussa, on se löytänyt lyhyemmän reitin pääsolmulle. Tällöin klusteritietotauluun talletetaan uusi HD sekä SID, jonka kautta reitti kulkee. Tavallisella anturisolmulla on tallessa vain yksi HID tieto, sillä se kuuluu vain yhteen klusteriin. Vierekkäisten klustereiden peittoalueet voivat olla myös osittain päällekkäin, tällöin anturisolmuilla jotka kuuluvat useaan klusteriin on klusteritietotaulussa useampi HID tallessa.

### 5.2.3 Pääsolmun kierrätys

DSBCA-protokollassa kierrätetään pääsolmua tietyin väliajoin, jotta anturisolmujen energiankulutus jakautuisi tasaisesti. Klusteri on vakaa kunnes pääsolmun uudelleenvalintaprosessi liipaistaan ajan  $T(k)$  kuluttua. Pääsolmu kerää kaikkien jäsenolmujensa painot ja valitsee seuraavaksi pääsolmuksi anturisolmun, jonka paino on suurin. Tällä tavoin viestinnän kustannukset pienenevät. Pääsolmun uudelleenvalinta

tapahtuu vanhassa klusterissa, joten yleislähetysviesti väliaikaisesta pääsolmusta ja siihen liittyvät vastaukset  $k$ -hypyn naapurisolmuilta ovat tarpeettomia.

Keskimääräinen klusterin overhead voidaan approksimoida seuraavasti

$$N_p = \frac{\ell k(4k - 1)(k + 1)}{6} \sim O(k^3),$$

missä  $\ell$  on solmun keskimääräinen aste. Joten viestinnän väheneminen, mikä saavutetaan valitsemalla uusi pääsolmu vanhasta klusterista, voidaan ilmaista suunnilleen seuraavasti:

$$N_b + N_r \sim 2N_b \sim O(k^3)$$

missä  $N_b$  yleislähetysviestin overhead ja  $N_r$  vastausviestien overhead.

#### 5.2.4 Vahvuudet

Monet klusterointiprotokollat olettavat, että anturisolmut ovat jakautuneet tasaisesti verkossa eivätkä huomioi klustereiden muodostuksessa anturisolmujen etäisyyttä tukiasemaan. Käytännössä anturisolmut jakautuvat yleensä epätasaisesti, joten verkon rakenteesta voisi tulla epätasapainoinen ja johtaa joidenkin anturisolmujen liialliseen kuormitukseen ja näin ollen nopeaan kuolemiseen. DSBCA-protokolla luo tasapainoisia ja kohtuullisen kokoisia klustereita, sillä klusterin säteeseen vaikuttaa sekä yhteystiheys että pääsolmun etäisyys tukiasemaan ja klusterin koolle määritetään kynnyksarvo. Klusterin säde kasvaa mitä kauempana tukiasemasta ollaan ja mitä pienempi on yhteystiheys. Pääsolmua kierrätetään klusterin sisällä ja pääsolmuksi valitaan aina painavin anturisolmu. Painoon vaikuttaa anturisolmun jäljellä oleva energia, yhteystiheys ja kuinka monesti anturisolmu on toiminut pääsolmuna. Näin saadaan anturisolmujen kuormaa jaettua tasaisemmin ja säästetään vähän energiaa omaavia anturisolmuja. Valitsemalla uusi pääsolmu vanhan klusterin jäsensolmujen joukosta saadaan vähennettyä viestien lähetystä ja näin ollen myös energian kulutusta. Artikkelissa [32] esitetyt simuloinnit osoittavat, että verrattuna perinteisiin klusterointiprotokolliin, DSBCA muodostaa vakaamman ja järkevämmän klusterirakenteen sekä parantaa verkon elinikää merkittävästi. Lisäksi se on skaalautuva ja toimii erikokoisissa verkoissa.

## 6 Klusterointi meluverkossa

Tässä luvussa käsitellään nimensä mukaisesti meluverkon klusterointia. Alaluvussa 6.1 kerrotaan yleisesti melusta, sen ominaisuuksista ja melun aiheuttamista haitoista sekä käsitellään melun mittauksessa huomioitavia tekijöitä. Alaluvussa 6.2 kerrotaan tarkemmin miten melua voidaan mitata langattomalla anturiverkolla. Alaluvussa 6.3 esitellään kivimurskaimen meluverkko ja sen asettamat vaatimukset sekä simuloidaan MH-LEACH ja DSBCA -protokollien toimintaa kyseisessä verkossa. Lisäksi samassa alaluvussa vertaillaan näiden kahden klusterointiprotokollan toimintaa ja soveltuvuutta kivimurskaimen meluverkkoon.

### 6.1 Melu

Ympäristöministeriön julkaisussa [25] on melua käsitelty monipuolisesti ja siinä melua on kuvattu seuraavasti. Melu on yleisin ja yksi tärkeimmistä ympäristön laatua heikentävistä ympäristötekijöistä, joka aiheuttaa myös terveyshaittoja. Melulla tarkoitetaan ääntä, joka on epämiellyttävää, häiritsevää, terveydelle vahingollista tai haitallista hyvinvoinnille. Sitä esiintyy kaikissa toimintaympäristöissä kaikkina kellonaikoina, eikä ääniympäristö ole koskaan täysin hiljainen. Melun aiheuttamat haitat ovat lisääntyneet varsinkin kaupungistumisen ja liikenteen kasvun myötä. Lyhytkestoinen, tilapäinen tai satunnainen melu on pääosin viihtyvyyskysymys, mutta pitkään jatkuessaan se aiheuttaa myös terveysvaikutuksia. Melu määräytyy sekä fysiologisten vaikutusten että aistittujen häiritsevyyspiirteiden perusteella. Melun häiritsevyys ei riipu pelkästään äänitasosta, vaan myös ajallisesta esiintymisestä, sen koetusta elämyspiirteistä ja merkityssisällöstä sekä yksilöllisistä ja yhteisöön sidoksissa olevista tekijöistä. Meluhaittojen yksilölliset vaihtelut johtuvat ihmisten fysiologisista eroista ja meluhaittojen vaikutukset ovat erilaisia eri-ikäisillä ja ilmenevät erilaisina naisilla ja miehillä.

Melua mitattaessa täytyy huomioida useita tekijöitä, kuten melulähteen laatu, mittauspisteen etäisyys äänilähteestä ja sijainti heijastaviin pintoihin nähden, mittausajankohta ja mittauksen kesto. Myös mittauksessa käytetyt taajuus- ja aikapainotukset, mittausten määrät sekä niiden edustavuus melulähteen ja tutkittavan vaikutuksen kannalta täytyy ottaa huomioon. Melun äänekkyyys riippuu äänen

taajuussisällöstä, joten mittauksessa otetaan huomioon eri taajuuksien osaäänien suhteellinen painoarvo. Jos kaikki taajuuskaistat huomioidaan samanarvoisina, on tällöin kyseessä lineaarinen äänenpainetason mittaus. Kuulojärjestelmän herkkyys vähenee taajuuden pienentyessä, joten painottamaton äänenpainetason mittaus ei anna oikeanlaista käsitystä äänen voimakkuudesta. Tästä johtuen eri taajuuksia painotetaan eri tavalla ja käytetään standardisoituja painotuskäyriä. Painotuskäyrä A mallintaa normaalikuuloisen taajuudesta riippuvia herkkyyseroja noin 60 fonin äänekkyystasolla. Painotuskäyrä C on suurin piirtein taajuudesta riippumaton keskitaajuuksilla noin 100 fonin tasolla, joten sitä voidaan käyttää esimerkiksi impulssiäänien mittauksissa. Taajuuspainotuksen lisäksi voidaan valita aikapainotus. Standardin mukainen aikapainotus S (slow) antaa äänitason keskiarvona 2 sekuntia kestävältä aikajaksolta, kun taas aikapainotus F (fast) kuvaa äänitasoa vastaavasti 250 ms ajalta ja aikapainotus I (impulse) 35 ms ajalta.

Keskiäänitaso on yleisin vaihtelevan ympäristömelun voimakkuutta kuvaava suure, joka ilmaisee keskimääräisen äänitason tarvittavan aikaikkunan osalta. Sitä voidaan käyttää esimerkiksi verrattaessa samanlaisten melulähteiden häiritsevyyttä keskenään. Samanlaiset melulähteet aiheuttavat melua, jotka ovat taajuussisällöltään samanlaisia, mutta niiden ajalliset vaihtelut ovat erilaisia. Keskiäänitaso ei sovellu pienitaajuisen melun ja hyvin kapeakaistaisen melun vertailuun muiden melutyyppeiden kanssa. Äänialtistusaso puolestaan kuvaa meluannoksen tasoa, jonka aikarajoitettu äänitapahtuma tuottaa. Se yhdistää lyhyiden tapahtumien enimmäistason, keston ja lukumäärän yhdeksi mittaluvuksi. Melun haittoja analysoitaessa on tärkeää huomioida myös altistusajan kesto ja vuorokauden aika. Myös yksittäisten melutapahtumien lukumäärä ja ajallinen jakauma voivat olla olennaisia asioita melun vaikutuksien kannalta.

## 6.2 Melunmittaus langattomissa anturiverkoissa

Tan ja Jarvis kertovat artikkelissa [45] melunmittauksesta langattomissa anturiverkoissa seuraavasti. Ääni on väliaineessa etenevää pitkittäistä aaltoliikettä, jonka väliaineena voi toimia kaasu, neste tai kiinteä aine. Ihmisen korva pystyy aistimaan näiden akustisten aaltojen paineenvaihteluita ja muuntamaan ne neurologisiksi ärsykeiksi, jotka aiheuttavat kuuloaistimuksen aivoissa. Vastaavasti mikrofoni muuntaa paineenvaihtelut sähköiseksi



signaaliksi, josta voidaan laskea äänilähteen äänenvoimakkuus. Ihmisen korva on erittäin herkkä ja koska pienimmän ja suurimman aistittavan äänenpaineen välinen ero on suuri, esitetään äänenpainetaso yleensä logaritmisella asteikolla, desibeleissä. Hetkellinen äänenpainetaso (SPL) lasketaan seuraavasti

$$L_p(t) = 10 \log_{10} \left( \frac{p_{RMS}^2(t)}{p_{ref}^2} \right) = 20 \log_{10} \left( \frac{p_{RMS}(t)}{p_{ref}} \right) \text{ (dB)},$$

missä  $p_{RMS}(t)$  on äänenpaineen neliöllinen keskiarvo (RMS) ja  $p_{ref}$  on viite äänenpaine, joka on tavallisesti 20  $\mu\text{Pa}$ .  $p_{RMS}(t)$  ei ole todellisesti hetkellinen, koska neliöllinen keskiarvo lasketaan tietyllä ajanjaksolla. Tämä ajanjakso ei ole vakio, joten se täytyy olla pituudeltaan vähintään mitattavan taajuuden jakson pituinen. Olettaen, että diskreetissä aikasarjassa  $p(i)$  on paine mitattuna tietyllä näytteenotto hetkellä ja  $N$  on näytteenottojen lukumäärä, saadaan  $p_{RMS}$  laskettua seuraavasti

$$p_{RMS} = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} p^2(i)}.$$

Melutason mittaamiseen pitkällä aikavälillä käytetään yleensä aikakeskiarvotettua äänentaso eli keskiäänitaso  $L_{eqT}$ , joka on keskiarvo peräkkäisistä  $p_{RMS}^2$ -arvoista. Olettaen että kyseessä on diskreetti aikasarja, jossa  $N$  on  $p_{RMS}$ -arvojen lukumäärä mitatun ajan kuluessa, saadaan  $L_{eqT}$  laskettua seuraavasti

$$L_{eqT} = 10 \times \log_{10} \left( \frac{1}{N} \sum_{i=0}^{N-1} \frac{p_{RMS}^2(i)}{p_{ref}^2} \right) \text{ (dB)}.$$

Keskiäänitaso ei ole melun äänitason tavallinen keskiarvo. Määritelmään sisältyvä neliöön korotus tarkoittaa, että keskimääräistä suuremmat äänenpaineet saavat korostetun painoarvon lopputuloksessa. Jotta anturisolmun mikrokontrolleri voisi laskea  $p_{RMS}$ :n, tulee  $p$ :n olla digitaalisessa muodossa ja tämän takia paineen aaltomuoto täytyy muuntaa. Anturisolmun mikrofonissa on kalvo, jonka ilman paineenvaihtelut saavat värähtelemään. Kalvoon kiinnitetty kela liikkuu magneettikentässä ja siihen indusoituu vaihtojännite.

Mikrofonin herkkyys eli mikrofonin antama jännitetaso tietyllä äänenpainetasolla voidaan laskea seuraavasti

$$S = 20 \times \log_{10} \left( \frac{E \times p_0}{E_{ref} \times p} \right) (dB).$$

Tavanomaisesti  $E_{ref}$ :n arvoksi asetetaan 1 V ja  $p_0$ :n arvoksi 1 Pa. Jos  $S$  on tiedossa, saadaan yhtälöstä helposti laskettua jännitteen muutos  $E$ , jonka paineenmuutos  $p$  aiheuttaa. Mikrofonit tuottaa jatkuvaa aaltomuotoista jännitettä, joka useissa melunmittausjärjestelmissä on A-painotettua. Tällä tarkoitetaan sitä, että signaalin taajuuskomponentteja on vaimennettu tai vahvistettu vastaamaan ihmisen korvan havaitsemia eri taajuuksia. Yleensä mikrofonilta tulevaa signaalia vahvistetaan aktiivivahvistimella ennen kuin se muutetaan analogia-digitaalimuuntimella digitaaliseen muotoon. AD-muunnin muuttaa jatkuvan signaalin ajan suhteen diskreeteiksi jännitetasoiksi ja muuttaa jännitetasot binääriluvuiksi, jotka sitten talletetaan mikrokontrollerin muistiin. AD-muuntimen toimintaan vaikuttaa kaksi parametria: näytteenottotaajuus ja erottelukyky. Korkeampi näytteenottotaajuus palauttaa todenmukaisemman signaalin alkuperäiseen signaaliin verrattuna. Resoluutio kertoo sen, kuinka moneen digitaaliseen arvoon analoginen signaali voidaan jakaa. Jos resoluutio on esimerkiksi 2 bittiä ja jatkuva jännite on väliltä 0 ja 1V, tällöin analoginen signaali voidaan jakaa neljään tasoon Taulukon 1 mukaisesti.

Taulukko 1. Analogisen signaalin jako digitaalisiin tasoihin [45].

Jännite (V)	digit. arvo
0 - 0,25	00
0,25 - 0,5	01
0,5 - 0,75	10
0,75 - 1	11

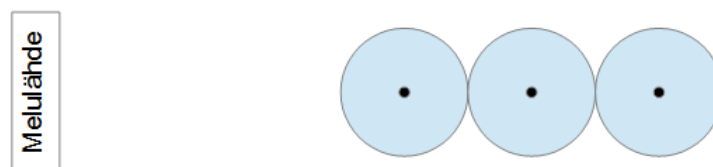
Mikrokontrollerin suoritin voisi käsitellä suoraan AD-muuntimen tuotosta ja käyttää sitä syötteenä  $p_{RMS}$  laskennassa. AD-muuntimen tuotos on riittävä syötteenä, vaikka yhtälö vaatiikin syötteenä äänenpainetasoa. Ilman AD-muuntimen tarkkuushäviöitä

kokonaisluvun ja äänen paineentason suhde on lineaarinen, joten se voidaan myöhemmin skaalata.

Monet melumittaukset perustuvat keskiäänitasoon  $L_{eqT}$ , mutta joka tilanteeseen se ei kuitenkaan sovi. Jos esimerkiksi melua mitattavan ajanjakson aikana on hyvin lyhyitä mutta äänekkäitä jaksoja, jäävät nämä keskiäänitasossa huomaamatta. Esimerkiksi jos normaalisti hyvin hiljainen alue kärsii säännöllisesti lentokoneiden melusta. Tällöin keskiäänentaso olisi hyvin matala eikä siitä kävisi ilmi lentokoneiden aiheuttama meluhaitta. Tällaisissa tilanteissa mitattavan ajanjakson maksimiäänepaine olisi informatiivisempi mittari.

### 6.3 Kivimurskaimen melua mittaavan langattoman anturiverkon klusterointi

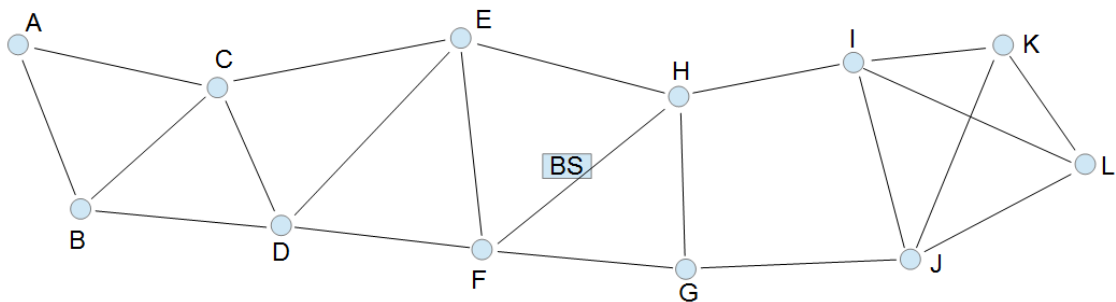
Kaukana sijaitseva kivimurskain on pistemäinen melulähde ja sen synnyttämän melun vaimenemiseen vaikuttaa ensisijaisesti etäisyys melulähteeseen. Myös erilaiset esteet sekä äänen absorboituminen väliaineeseen vaimentavat melua. Absorboitumiseen vaikuttaa osaltaan myös ilman kosteus sekä lämpötila. Kivimurskaimen meluverkossa tulisi pyrkiä siihen, että anturisolmujen muodostettavat klusterit sijaitsisivat jonossa melulähteestä pois päin, kuten Kuvassa 20. Tällöin melua voidaan mitata samanaikaisesti usealta alueelta ja melun vaimenemista arvioida haluttuun suuntaan. Optimaalisessa verkossa klusterit olisivat mahdollisimman stabiileja eikä päällekkäisiä klustereita muodostuisi.



Kuva 20. Klustereiden sijainti.

Kivelä ja Hakala [28] ovat esittäneet aluepohjaisen mittausmenetelmän. Siinä mittausalue on katettu usealla melua mittaavalla anturisolmulla. Anturisolmut ovat jaettu ryhmiin ja yksi ryhmä kerrallaan mittaa melua koko mittausalueelta. Gateway-solmu laskee ryhmän

mittausten perusteella alueen keskiäänitason  $L_{Aeq}$ , maksimikeskiäänitason  $L_{Aeq\_max}$  ja minimikeskiäänitason  $L_{Aeq\_min}$ .  $L_{Aeq}$  lasketaan geometrisesti painotettuna keskiarvona ja se kertoo mittausalueen keskimääräisen ekvivalenttitason. Tätä mittausmenetelmää käytettäessä täytyy melulähteen etäisyyden olla tarpeeksi suuri verrattuna klusterin halkaisijaan. Näin estetään ettei kahden anturisolmun antamat tulokset eroa mainittavasti toisistaan melulähteen etäisyydestä johtuen. Käytetään tätä aluepohjaista mittausmenetelmää myös kivimurskaimen meluverkossa, mutta muutetaan verkon rakennetta siten, että mittausalue jaetaan kolmeen klusteriin ja jokaisen klusterin pääsolmu laskee jäsensolmujensa mittausten perusteella keskiäänitason  $L_{Aeq}$  klusterin alueella. Näin saadaan samanaikainen keskiäänitaso  $L_{Aeq}$  kolmelta eri alueelta ja näin ollen voidaan melun vaimenemista arvioida klustereiden suuntaisesti. Kuvassa 21 on esitetty esimerkki miten anturisolmut voivat sijaita sekä anturisolmujen väliset linkit. Melulähde sijaitsee verkon vasemmalla puolella.



Kuva 21. Kivimurskaimen meluverkko.

Anturisolmulle voidaan määrittää vastaanotettavalle signaalille jokin minimi arvo mitä heikkomat signaalit hylätään. Näin voidaan taata, että anturisolmujen väliset linkit ovat luotettavia ja mahdollisimman virheettömiä.  $RSSI$  arvon perusteella voidaan arvioida signaalin lähteen etäisyys  $D$  käyttäen kaavaa 6. Tässä työssä käytetään  $RSSI$  arvojen sijaan yksinkertaisesti etäisyyksiä ja tavallisen anturisolmun vastaanottaman signaalin maksimietäisyydeksi on määritetty 40m. Toisin sanoen rajoitetaan klusterin maksimi säteeksi 40m. Jotta datapakettien reititys pääsolmulta tukiasemalle voisi toimia, täytyy pääsolmujen pystyä vastaanottaa myös kauempaa muilta pääsolmulta tulevia datapaketteja ja välittää niitä kohti tukiasemaa. Kauimpana tukiasemasta sijaitsee 86 metrin etäisyydellä

anturisolmu A, joten määritetään maksimietäisyydeksi tukiasemien ilmoitusviesteille ja pääsolmun tukiasemalle lähetettäville viesteille 90 metriä. Tällöin toimisi myös pahin tilanne, missä tukiaseman vasemmalla puolella pelkästään anturisolmu A olisi pääsolmu eli sen pitäisi lähettää data suoraan tukiasemalle.

Oletetaan myös, että lähempänä oleva anturisolmu on yhteydenotossa aina nopeampi. Taulukossa 2 on esitetty verkon anturisolmujen väliset etäisyydet ja anturisolmujen peittäämä alue on n. 35m\*160m.

Taulukko 2. Kivimurskaimen meluverkon etäisyydet (m).

	A	B	C	D	E	F	G	H	I	J	K	L	BS
A		27	32										86
B	27		28	31									75
C	32	28		23	38								55
D		31	23		40	31							44
E			38	40		33		35					26
F				31	33		32	38					18
G						32		26		35			24
H					35	38	26		27				21
I								27		32	23	39	47
J							35		32		36	30	54
K									23	36		22	70
L									39	30	22		80
BS	86	75	55	44	26	18	24	21	47	54	70	80	

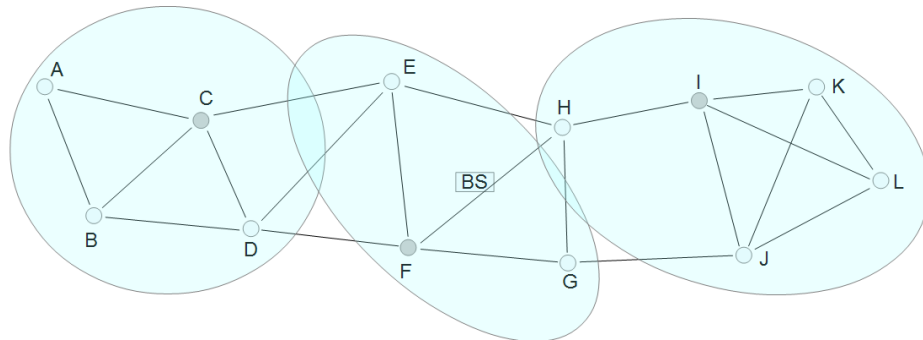
Seuraavaksi käydään läpi miten MH-LEACH ja DSBCA muodostavat klusterit ja toimivat kivimurskaimen meluverkossa sekä vertaillaan näiden kahden protokollan eroja sekä sopivuutta kyseiseen verkkoon.

### 6.3.1 MH-LEACH

MH-LEACH -protokollan toiminta tapahtuu kierroksittain ja jokainen kierros voidaan jakaa kahteen vaiheeseen. Ensimmäisessä vaiheessa protokolla valitsee klustereiden pääsolmut, kuten LEACH-protokollassa. Siinä anturisolmut valitsevat satunnaisluvun väliltä (0,1) ja sitä verrataan kynnsarvoon  $T(n)$ , joka voidaan laskea kaavasta 1. Satunnaisluvun ollessa kynnsarvoa  $T(n)$  pienempi, tulee anturisolmusta pääsolmu

kyseiselle kierrokselle. Jos anturisolmu on jo valittu aiemmalla kierroksella pääsolmuksi, on kynnyksiarvo tällöin 0 ja näin ollen anturisolmu ei tule enää valituksi pääsolmuksi. Pääsolmujen osuudeksi anturisolmujen kokonaismäärästä valitaan meluverkon tapauksessa 25 %:a ja näin ollen asetetaan  $P:n$  arvoksi 0,25. Tämä tarkoittaa sitä, että jokainen anturisolmu tulee valituksi pääsolmuksi ainakin kerran 4 kierroksen aikana. Valitut pääsolmut lähettävät ilmoitusviestin, jolloin kaikki kuuluvalualueella olevat pääsolmut muodostavat signaalinvoimakkuuden (*RSSI*) perusteella alustavat reititystaulut. Toisessa vaiheessa pääsolmut lähettävät alustavat reititystaulunsa tukiasemalle, joka korjaa reititystaulut.

Ensimmäisellä kierroksella ( $r=0$ ) kynnyksiarvo  $T(n)$  on 0,25 jolloin pääsolmuiksi valitaan esimerkiksi anturisolmut C, F ja I. Olettamuksen mukaan anturisolmu liittyy aina lähimpään pääsolmuun, jolloin klusterit muodostuisivat Kuvan 22 mukaisesti ja alustavien reititystaulujen sisällöt olisivat kuten Taulukossa 3.



Kuva 22. MH-LEACH, ensimmäinen kierros ( $r=0$ ).

Taulukko 3. MH-LEACH, ensimmäisen kierroksen ( $r=0$ ) alustavat reititystaulut.

Anturisolmu C	
F	48
BS	55

Anturisolmu F	
BS	18
C	48
I	64

Anturisolmu I	
BS	47
F	64

Anturisolmujen C ja I reititystaulut ovat optimaalisia, sillä tukiasema on lähempänä anturisolmua F kuin anturisolmua C, joten anturisolmu F voi olla anturisolmun C reitillä. Tukiasema on myös lähempänä anturisolmua F kuin anturisolmua I, joten anturisolmu F

voi olla anturisolmun I reitillä. Anturisolmun F reititystaulusta poistetaan anturisolmut C ja I, koska tukiasema on lähempänä anturisolmua F kuin anturisolmua C tai I. Ensimmäisen kierroksen ( $r=0$ ) lopulliset reititystaulut on esitetty Taulukossa 4.

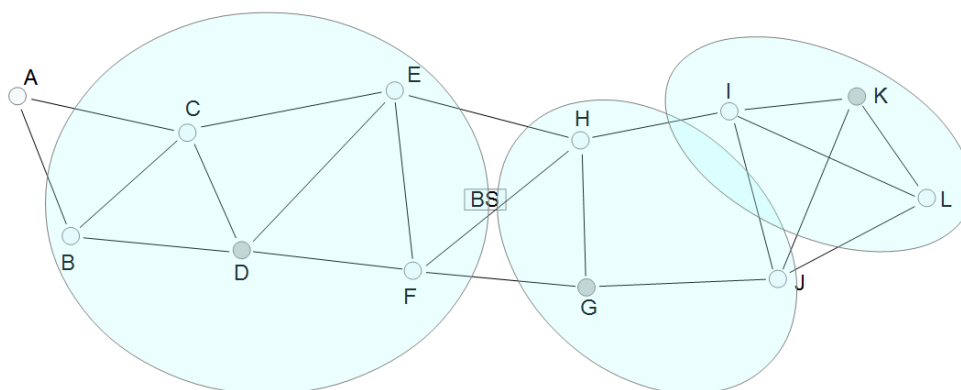
Taulukko 4. MH-LEACH, ensimmäisen kierroksen ( $r=0$ ) lopulliset reititystaulut.

Anturisolmu C	
F	48
BS	55

Anturisolmu F	
BS	18

Anturisolmu I	
BS	47
F	64

Toisella kierroksella ( $r=1$ ) kynnysarvo  $T(n)$  on 0,333 ja kierroksen 0 pääsolmut eivät voi tulla enää valituiksi pääsolmuiksi. Jos pääsolmuiksi valitaan anturisolmut D, G ja K tällöin klusterit muodostuisivat kuten Kuvassa 23 ja alustavat reititystaulut kuten Taulukossa 5.



Kuva 23. MH-LEACH, toinen kierros ( $r=1$ ).

Taulukko 5. MH-LEACH, toisen kierroksen ( $r=1$ ) alustavat reititystaulut.

Anturisolmu D	
BS	44
G	62

Anturisolmu G	
BS	24
K	59
D	62

Anturisolmu K	
G	59
BS	70

Pelkästään anturisolmun G reititystaulu ei ole optimaalinen. Anturisolmut K ja D eivät voi olla anturisolmun G reitillä, joten ne poistetaan Taulukon 6 lopullisista reititystauluista.

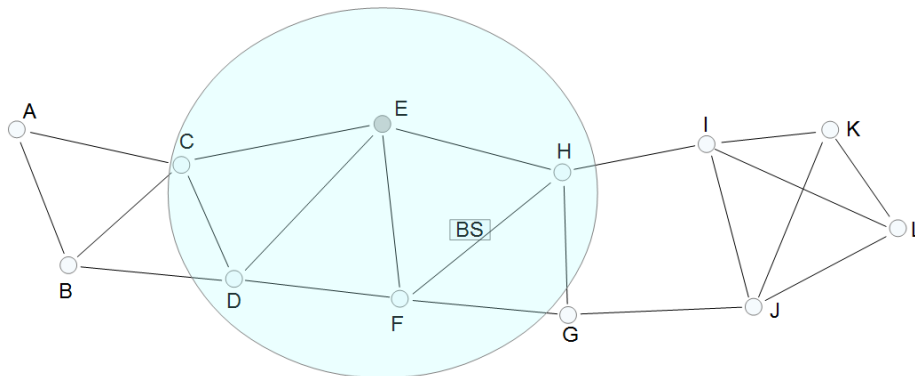
Taulukko 6. MH-LEACH, toisen kierroksen ( $r=1$ ) lopulliset reititystaulut.

Anturisolmu D	
BS	44
G	62

Anturisolmu G	
BS	24

Anturisolmu K	
G	59
BS	70

Edellisillä kierroksilla on simuloitu sitä, että jos sattumoisin pääsolmuiksi on valittu juuri 25 %:a anturisolmuista. LEACH-protokollassa pääsolmun valinta tehdään anturisolmujen arpoman satunnaisluvun perusteella, joten pääsolmujen määrä voi vaihdella kierrosten välillä. Kolmannella kierroksella ( $r=2$ ) kynnyksarvo  $T(n)$  on 0,5 ja pääsolmuiksi voi tulla valituiksi vain ne anturisolmut, jotka eivät ole vielä tulleet valituiksi. Eli mahdollisia pääsolmuja on 6, mutta jos pelkästään anturisolmun E arpoma satunnaisluku on pienempi kuin 0,5. Tällöin kierroksella on vain yksi pääsolmu Kuvan 24 mukaisesti.

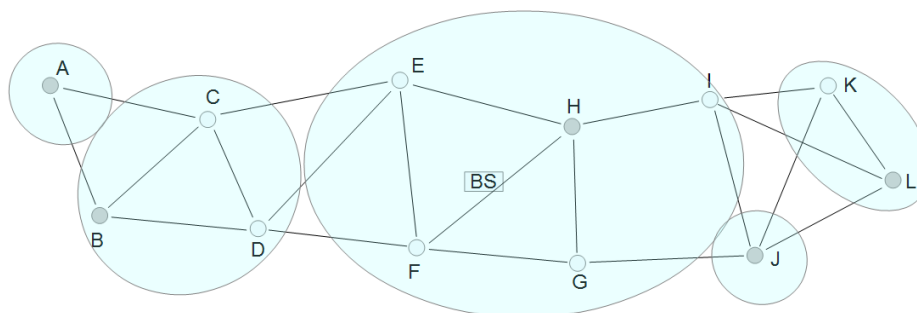


Kuva 24. MH-LEACH, kolmas kierros ( $r=2$ ).

Koska pääsolmuja on vain yksi, ei pääsolmu kuule vaiheen 1 ilmoitusviestiä kuin tukiasemalta, joten alustava reititystaulu on optimaalinen ja siinä on vain reitti tukiasemaan.

Kierroksella neljä ( $r=3$ ) mahdollisia pääsolmuja on jäljellä 5 kappaletta ja koska kynnyksarvo  $T(n)$  on 1, tulevat kaikki 5 anturisolmuja valituiksi pääsolmuiksi Kuvan 25 mukaisesti ja alustavat reititystaulut muodostuvat Taulukon 7 mukaisesti.





Kuva 25. MH-LEACH, neljäs kierros ( $r=3$ ).

Taulukko 7. MH-LEACH, neljännen kierroksen ( $r=3$ ) alustavat reititystaulut.

Anturisolmu A		Anturisolmu B		Anturisolmu H		Anturisolmu J		Anturisolmu L	
B	27	A	27	BS	21	L	31	J	31
BS	86	BS	75	J	43	H	43	H	63
				L	63	BS	54	BS	80

Lopulliset reititystaulut on esitetty Taulukossa 8. Anturisolmujen A, B ja L reititystaulut ovat tarkistusalgoritmin perusteella optimaalisia. Anturisolmun H reititystaulusta poistetaan anturisolmut J ja L. Ja anturisolmun J reititystaulusta poistetaan pelkästään anturisolmu L.

Taulukko 8. MH-LEACH, neljännen kierroksen ( $r=3$ ) lopulliset reititystaulut.

Anturisolmu A		Anturisolmu B		Anturisolmu H		Anturisolmu J		Anturisolmu L	
B	27	A	27	BS	21	H	43	J	31
BS	86	BS	75			BS	54	H	63
								BS	80

Neljännen kierroksen ( $r=3$ ) jälkeen kaikki anturisolmut ovat tulleet valituiksi pääsolmuiksi, jolloin kierrokset aloitetaan alusta ja seuraavalla kierroksella jokainen anturisolmu voi tulla taas valituksi pääsolmuksi.

### 6.3.2 DSBCA

DSBCA toimii siten, että se valitsee aloitusvaiheessa satunnaisesti tietyn määrän anturisolmuja laukaisemaan klusterointiprosessin. Nämä anturisolmut laskevat niiden

etäisyyden tukiasemaan sekä yhteystiheyden, joiden mukaan lasketaan klusterin säde  $k$  kaavan 5 mukaan. Jotta klustereiden koko pysyisi sopivana ja energiankulutus pienenä, rajoitetaan klusterin koko maksimissaan 2 hypyn kokoiseksi asettamalla  $\beta = 0,06$ . Minimi hyppyjen määrä on luonnollisesti 1. Taulukossa 9 on esitetty anturisolmujen etäisyydet, yhteystiheydet sekä laskettu säteet  $k$ .

Taulukko 9. Anturisolmujen etäisyydet tukiasemasta, yhteystiheys ja säde  $k$ .

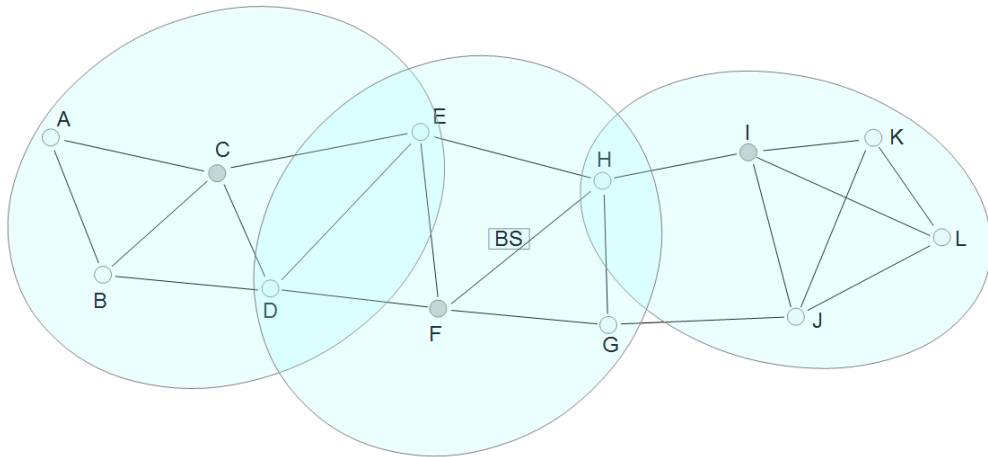
Anturisolmu	Etäisyys tukiasemaan	yhteystiheys	$k$
A	86	21/12	2
B	75	21/12	2
C	55	21/12	1
D	44	21/12	1
E	26	21/12	1
F	18	21/12	1
G	24	21/12	1
H	21	21/12	1
I	47	21/12	1
J	54	21/12	1
K	70	21/12	2
L	80	21/12	2

Aloitusvaiheessa valitaan väliaikaisiksi pääsolmuiksi esimerkiksi anturisolmut A, D ja K laukaisemaan klusterointiprosessi. Anturisolmujen A ja K säteet ovat 2, joten maksimissaan 2-hypyn päässä olevat anturisolmut vastaanottavat pääsolmujen lähettämät Hello-viestit. Anturisolmun D lähettämän Hello-viestin vastaanottaa vain 1-hypyn päässä olevat naapurisolmut, koska klusterin D säde on 1. Hello-viestit lähetetään yleislähetysviestinä, mutta algoritmi hylkää paketit, joiden HD on suurempi kuin säde  $k$ . Näin varmistetaan ettei klusteri voi olla  $k$ -hyppyä suurempi. Taulukossa 10 on esitetty pääsolmujen  $k$ -hypyn päässä olevat naapurisolmut sekä monenko hypyn päässä ne ovat. Esimerkiksi anturisolmun A 2-hypyn sisällä olevat naapurisolmut ovat B, C, D ja E.

Taulukko 10. Anturisolmujen  $k$ -hypyn naapurisolmut.

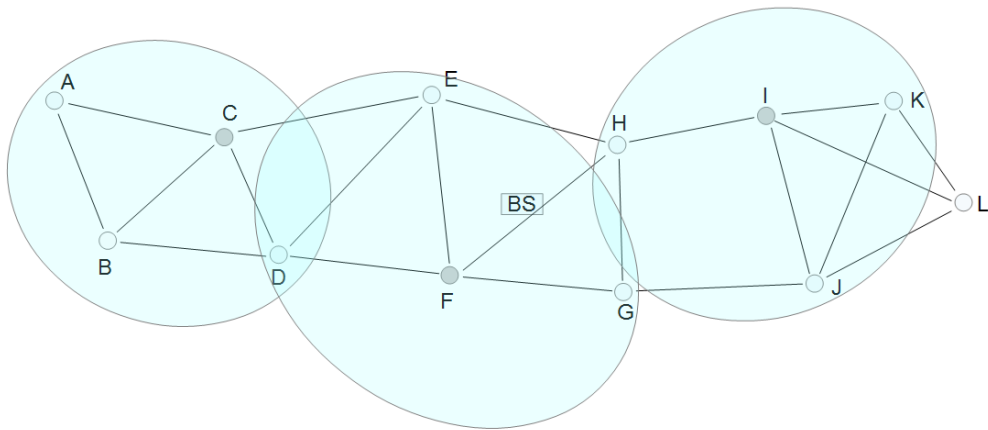
Pääsolmut												k-hypyn naapurisolmut	
A	B	C	D	E	F	G	H	I	J	K	L		
	1	1	2	2									A
1		1	1	2	2								B
1	1		1	1									C
	1	1		1	1								D
		1	1		1		1						E
			1	1		1	1						F
					1		1		1				G
				1	1	1		1					H
							1		1	1	1		I
						1		1		1	1		J
						2	2	1	1		1	K	
						2	2	1	1	1		L	

Naapurisolmut laskevat painonsa kaavan 9 mukaan ja painavin anturisolmu valitaan aina klusterin pääsolmuksi. Anturisolmun painoon vaikuttaa pelkästään jäljellä oleva energia sekä monestikko anturisolmu on valittu pääsolmuksi, koska yhteystiheys on tässä meluverkon tapauksessa kaikilla anturisolmuilla vakio. Aloituvaiheessa yhtään anturisolmua ei ole vielä valittu pääsolmuksi, joten pääsolmun valinta perustuu pelkästään anturisolmujen jäljellä olevaan energiamäärään. Jos painavimmat anturisolmut ovat esimerkiksi C (klusteri A), F (klusteri D) ja I (klusteri K), valitaan ne pääsolmuiksi ja ne lähettävät 1-hypyn naapureilleen Head-viestin, jolla ne ilmoittavat olevansa pääsolmu. Eli pääsolmun C Head-viestin vastaanottavat anturisolmut A, B, D ja E. Pääsolmun F Head-viestin vastaanottavat anturisolmut D, E, G ja H ja pääsolmun I Head-viestin vastaanottavat anturisolmut H, J, K ja L. Näin ollen anturisolmut D, E ja H vastaanottavat Head-viestin kahdelta pääsolmulta, joten ne voivat liittyä kahteen klusteriin kunhan anturisolmun paino on pienempi kuin kummankin pääsolmun ja jos kyseiset klusterit eivät ole vielä täynnä. DSBCA asettaa klustereiden koolle kynnyksarvon, jota enempää anturisolmuja ei voi klusterissa olla. Jos valitaan kynnyksarvoksi 5, tällöin muodostuisi Kuvan 26 mukainen verkko eli klustereita muodostuisi kolme ja jokaisessa klusterissa olisi 5 anturisolmua.



Kuva 26. DSBCA klusterointikierron 1, klusterin maksimi koko 5.

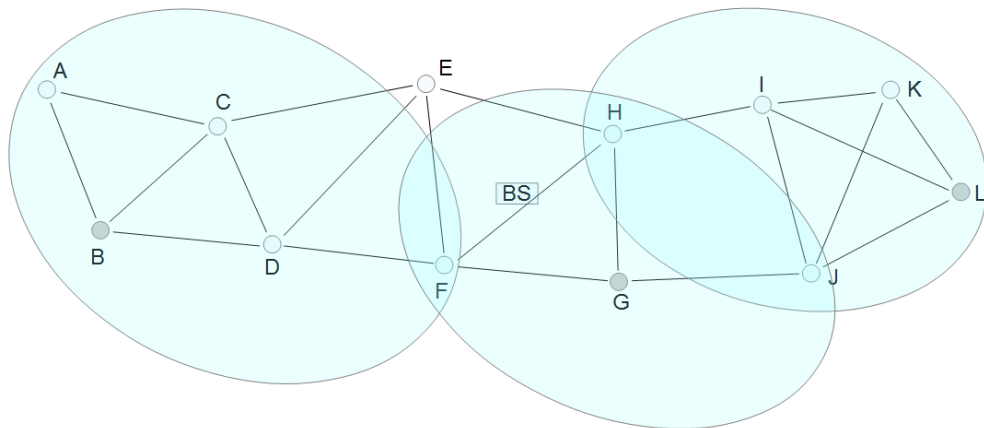
Jos taas klusterin koon kynnyksarvo olisi 4, tällöin klustereihin liittyisivät aina 3 pääsolmua lähinnä olevaa anturisolmua, joten verkosta muodostuisi Kuvan 27 mukainen. Eli muodostuisi 3 klusteria, joissa anturisolmujen määrä olisi 4. Anturisolmu L jää ilman klusteria, sillä klusteri I on täynnä ja näin ollen sen Join-viesti hylätään eikä toista mahdollista klusteria ole.



Kuva 27. DSBCA klusterointikierron 1, klusterin maksimi koko 4.

Klustereiden väliseen reititykseen ei artikkelissa [32] oteta kantaa, joten valitaan reitityksen tapahtuvan samalla tavalla kuin MH-LEACH -protokollassa. Tällöin klusterointikierroksella 1 reititystaulut olisivat samanlaiset kuin MH-LEACH -protokollan ensimmäisellä kierroksella eli alustavat reititystaulut olisivat Taulukon 3 ja lopulliset reititystaulut Taulukon 4 mukaiset.

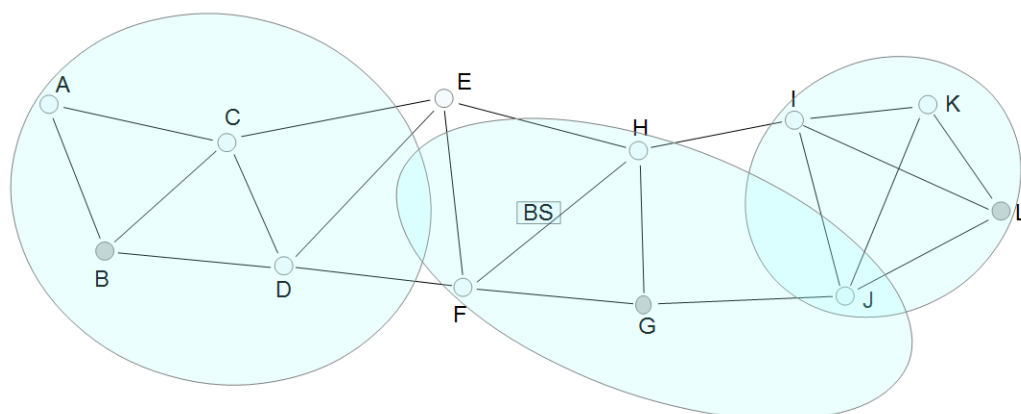
Ajan  $T(k)$  jälkeen klusterointi suoritetaan aina uudestaan. Uudelleen klusteroinnissa pääsolmut keräävät jäsenosojensa painot ja valitsevat uudet pääsolmut. Näin saadaan uudelleen klusteroinnissa vaadittavaa viestintää vähennettyä ja näin ollen säästettyä energiaa. Kuvan 26 tapauksessa, missä klusterin maksimikoko on 5, seuraavien pääsolmujen valinta menisi seuraavasti. Klusterissa C todennäköisintä on, että anturisolmun B paino on suurin ja tulee valituksi uudeksi pääsolmuksi, sillä anturisolmu A on ollut väliaikaisena pääsolmuna ja anturisolmut D ja E ovat kuuluneet kahteen klusteriin ja näin ollen kuluttaneet todennäköisesti enemmän energiaa. Samoin perustein klusterista F valitaan uudeksi pääsolmuksi anturisolmu G ja klusterista I joko anturisolmu J tai L. Oletetaan, että se on anturisolmu L, jolloin verkko olisi Kuvan 28 kaltainen.



Kuva 28. DSBCA klusterointikierron 2, klusterin maksimi koko 5.

Pääsolmujen B ja L klustereiden säteet ovat 2, joten maksimissaan 2-hypyn päässä olevat anturisolmut vastaanottavat niiden lähettämät Head-viestit ja yrittävät liittyä niiden klusteriin. Anturisolmun B Head-viestin vastaanottaa 5 anturisolmuja ja anturisolmu E yrittää näistä liittyä klusteriin viimeisenä, joten sen Join-viesti hylätään koska klusteri on jo täynnä. Pääsolmun L Head-viestin vastaanottaa 4 anturisolmuja, joten ne kaikki voivat liittyä klusteriin L. Klusterin G säde on 1, joten anturisolmut F, H ja J vastaanottavat pääsolmun G Head-viestin ja liittyvät sen klusteriin.

Jos klusterin koon kynnyksarvo on 4, tällöin anturisolmut F ja H liittyisivät pelkästään klusteriin G Kuvan 29 mukaisesti, koska klusterit B ja L ovat jo täynnä.



Kuva 29. DSBCA klusterointikierron 2, klusterin maksimi koko 4.

Klusterointikierron 2 alustavat reititystaulut olisivat alla olevan Taulukon 11 mukaiset ja lopulliset reititystaulut olisivat kuten Taulukossa 12. Ajan  $T(k)$  jälkeen klusterointi suoritetaan jälleen uudestaan.

Taulukko 11. DSBCA klusterointikierron 2, alustavat reititystaulut.

Anturisolmu B	
BS	75

Anturisolmu G	
BS	24
L	69

Anturisolmu L	
G	69
BS	80

Taulukko 12. DSBCA klusterointikierron 2, lopulliset reititystaulut.

Anturisolmu B	
BS	75

Anturisolmu G	
BS	24

Anturisolmu L	
G	69
BS	80

### 6.3.3 Klusterointiprotokollien vertailu

Molemmat protokollat MH-LEACH ja DSBCA mahdollistavat aluepohjaisen melunmittauksen. Ne myös pienentävät kivimurskaimen meluverkon energian kulutusta ja pidentävät verkon elinikää verrattuna klusterioimattomaan verkkoon. MH-LEACH -protokollassa klusterin sisäinen tiedonsiirto tapahtuu suoraan jäsensolmun ja pääsolmun välillä. Sen sijaan DSBCA-protokollassa klusterin sisäinen tiedonsiirto riippuu klusterin säteestä  $k$ , joka on taas riippuvainen pääsolmun etäisyydestä tukiasemaan sekä yhteystiheydestä. Näin ollen klusterin jäsensolmut saattavat joutua välittämään dataa

klusterin uloimmilta jäsensoiluilta. Klusterin sisäisten reititystaulujen muodostus sekä tiedon välitys lisäävät verkon liikennettä ja näin ollen myös lisäävät energian kulutusta ja lyhentävät verkon elinikää. Molemmissa protokollissa pääsolmut suorittavat datan kokoamista ja lähettävät tiedonkeruusolmulle pelkästään koottua dataa, joten liikennemäärä pysyy kohtuullisena laajoissakin verkoissa. MH-LEACH lähettää pääsolmujen kokoaman datan lähempänä tukiasemaa sijaitsevien pääsolmujen kautta tukiasemalle. Klustereiden välisen reitityksen reititystaulut muodostetaan kuuluvuusalueella olevien pääsolmujen ja tukiaseman lähettämän ilmoitusviestin *RSSI* arvon perusteella. Pääsolmut lähettävät alustavat reititystaulut tukiasemalle, joka keskitetysti laskee lopulliset reititystaulut ja lähettää ne takaisin pääsolmuille. Artikkelissa [32] ei oteta kantaa klustereiden väliseen reititykseen, joten kivimurskaimen meluverkossa valittiin DSBCA-protokollan käyttävän samaa reititystapaa kuin MH-LEACH.

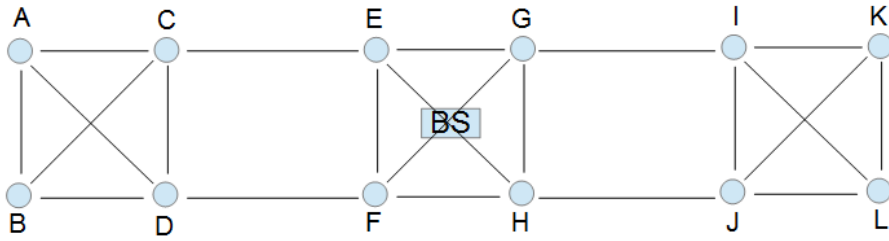
MH-LEACH -protokolla jakaa klusterin anturisolmujen kuormaa kierrättämällä pääsolmua samalla tavoin kuin LEACH-protokollassa. Se ei kuitenkaan huomioi pääsolmun valinnassa anturisolmujen jäljellä olevaa energiaa, kuten DSBCA. Näin ollen vähän energiaa omaavat anturisolmut toimivat pääsolmuina yhtä monesti kuin korkeamman energiatason anturisolmut ja saattavat kuluttaa energiansa loppuun aikaisessa vaiheessa. DSBCA-protokollassa klusterin uusi pääsolmu valitaan anturisolmujen painon perusteella eli klusterin painavin anturisolmu valitaan uudeksi pääsolmuksi. Anturin painoon vaikuttavat yhteystiheys, jäljellä olevan energian suhde alussa olevaan energiaan ja monestiko anturisolmu on valittu pääsolmuksi. Painon laskennassa voidaan sovelluksesta riippuen määrittää vaikutuskertoimet eli voidaan halutessa painottaa esimerkiksi jäljellä olevaa energiaa. Varsinkin kivimurskaimen meluverkon tapauksessa on järkevää painottaa jäljellä olevaa energiaa, koska yhteystiheys on kaikilla anturisolmuilla sama, eikä sillä ole suurta merkitystä kuinka monesti anturisolmu on jo toiminut pääsolmuna. Valitsemalla pääsolmuksi aina energiarikkaimman anturisolmun, pysyvät anturisolmujen energiatasot suunnilleen samansuuruisina.

MH-LEACH -protokollassa pääsolmun valinta perustuu täysin todennäköisyyteen, joten pääsolmujen määrä ja niiden sijainti voi vaihdella paljon eri kierrosten välillä. Esimerkiksi kivimurskaimen meluverkossa on asetettu  $P$ :n arvoksi 0.25 eli pääsolmujen osuus

anturisolmujen kokonaismäärästä on 25 %. On kuitenkin mahdollista, että pääsolmuksi ei valita kierroksella yhtään anturisolmua tai valitaan kaikki 12 anturisolmua. Tämä on toki epätodennäköistä, mutta kuitenkin mahdollista. On myös mahdollista, että kaikki pääsolmut keskittyvät samalle alueelle ja osassa verkkoa ei ole lainkaan pääsolmuja. DSBCA-protokollassa tämä ei ole mahdollista, koska uudeksi pääsolmuksi valitaan klusterin anturisolmuista aina vähintään yksi. Artikkelissa [32] ei oteta kantaa miten DSBCA toimii uuden pääsolmun valinnassa, jos painavimpia anturisolmuja on klusterissa useita ja ne ovat saman painoisia. Jos nämä kaikki anturisolmut valitaan pääsolmuiksi, lisääntyy klustereiden määrä verkossa. Artikkelissa ei ole myöskään otettu kantaa tilanteeseen, jossa anturisolmu kuuluu kahteen tai useampaan klusteriin, kuten anturisolmu D Kuvassa 26 ja se olisi näiden kaikkien klustereiden painavin anturisolmu. Tällöin se valittaisiin uudeksi pääsolmuksi kaikkiin klustereihin joihin se kuuluu ja näin ollen klustereiden määrä vähenisi.

Kivimurskaimen melun mittausta ajatellen optimaalinen verkko olisi sellainen, että päällekkäisiä klustereita ei muodostuisi. Myös klustereiden kokojen ja sijaintien tulisi pysyä mahdollisimman stabiileina, jotta mittaustulokset olisivat vertailukelpoisia. Näin ollen MH-LEACH ei sovellu kivimurskaimen meluverkon klusterointiin, johtuen juuri todennäköisyyteen perustavasta pääsolmujen valinnasta. Energian kulutuksen kannalta olisi myös hyvä, jos klusterin sisäinen viestintä tapahtuisi 1-hypyn menetelmällä ja pääsolmujen valinnassa huomioitaisiin anturisolmujen jäljellä oleva energia. DSBCA-protokollassa nämä toteutuisivat Kuvan 30 kaltaisessa verkossa, jos asetetaan klusterin koon kynnyksarvoksi 4, asetaan säteen  $k$  laskennassa  $\beta = 0,04$  ja näin ollen verkon jokaisen klusterin säteeksi tulisi 1. Lisäksi algoritmin suorituksen alussa jokaisesta joukosta {A, B, C, D}, {E, F, G, H} ja {I, J, K, L} valitaan yksi anturisolmu väliaikaiseksi pääsolmuksi. Taulukossa 13 on esitetty anturisolmujen etäisyydet tukiasemasta, anturisolmujen yhteystiheydet sekä laskettu säteet  $k$ .



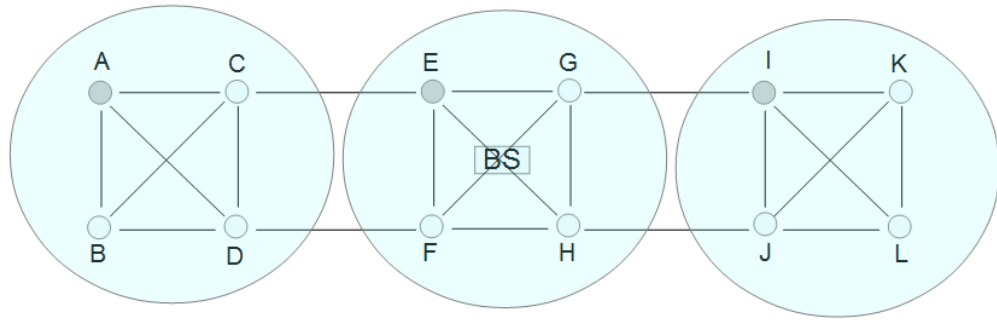


Kuva 30. DSBCA, optimaalinen verkon rakenne.

Taulukko 13. DSBCA, optimaalisen verkon anturisolmujen etäisyydet tukiasemasta, yhteystiheys ja säde  $k$ .

Anturisolmu	Etäisyys tukiasemaan	yhteystiheys	$k$
A	83,2	22/12	1
B	83,2	22/12	1
C	55,8	22/12	1
D	55,8	22/12	1
E	19,8	22/12	1
F	19,8	22/12	1
G	19,8	22/12	1
H	19,8	22/12	1
I	55,8	22/12	1
J	55,8	22/12	1
K	83,2	22/12	1
L	83,2	22/12	1

Kaikkien klustereiden säteet ovat siis 1, joten maksimissaan yhden hypyn päässä pääsolmusta olevat anturisolmut voivat vastaanottaa pääsolmun lähettämän Head-viestin. Jos väliaikaisiksi pääsolmuiksi valitaan esimerkiksi anturisolmut A, E ja I. Tällöin verkko olisi Kuvan 31 kaltainen ja klusteriin A liittyisivät anturisolmut B, C ja D. Klusteriin E liittyisivät anturisolmut F, G ja H ja klusteriin I liittyisivät anturisolmut J, K ja L.



Kuva 31. Optimaalisen verkon klusterit.

Tässä verkossa klusterit pysyvät stabiileina, sillä klustereiden maksimi koko on 4 ja verkon rakenne on sellainen, että klustereiden anturisolmut ovat lähempänä toisiaan kuin muiden klustereiden anturisolmuja. Esimerkiksi jos klusterin A uudeksi pääsolmuksi valitaan C, vastaanottavat anturisolmut A, B, D ja E sen lähettämän Head-viestin. Anturisolmut A, B ja D ovat lähempänä pääsolmua C kuin anturisolmu E, joten klusteriin C ennättää liittyä todennäköisimmin anturisolmut A, B ja D. Tällöin anturisolmun E Join-viesti hylätään, koska klusteri C on jo täynnä. Eli anturisolmut pysyisivät aina samassa klusterissa ja jokaisessa klusterissa pääsolmua kierrätettäisiin anturisolmujen energiatasojen mukaan. Pääsolmuksi valittaisiin uudelleenklusteroinnissa aina klusterin energiarikkain anturisolmu. Näin ollen klusterit pysyisivät stabiileina ja jokaisen klusterin mitaamat suuret eli keskiäänitaso  $L_{Aeq}$ , maksimikeskiäänitaso  $L_{Aeq\_max}$  ja minimikeskiäänitaso  $L_{Aeq\_min}$  ovat aina mitattu samalta alueelta ja ovat näin ollen vertailukelpoisia. Myös melun vaimenemista on helppo arvioida sillä, klustereiden lukumäärä, koot ja sijainnit suhteessa toisiinsa pysyvät muuttumattomina.

## 7 Yhteenveto

Langattomia anturiverkkoja voidaan hyödyntää hyvin monentyyppisissä sovelluksissa. Niitä käytetään mm sotilas-, ympäristö-, terveys- ja kotisovelluksissa sekä muissa kaupallisissa sovelluksissa. Ne voivat mitata lukemattomia suureita ja ne pystyvät kommunikoimaan keskenään langattomasti lyhyillä etäisyyksillä kuluttaen kuitenkin vähän energiaa. Langattomat anturiverkot koostuvat pienistä, paristokäyttöisistä anturisolmuista, jotka kykenevät prosessoimaan dataa ja sisältävät tiedonvälitykseen tarvittavat komponentit. Anturisolmut voivat paikallisesti suorittaa yksinkertaisia laskutoimituksia ja toimittaa vain vaadittua tai osittain prosessoitua dataa tiedonkeruusolmulle. Usein langattomat anturisolmut ripotellaan satunnaisesti vaikeasti saavutettavalle alueelle esimerkiksi lentokoneesta käsin, joten anturisolmujen täytyy kyetä muodostamaan verkkonsa itsenäisesti. Juuri datan prosessointikyky sekä itsenäinen verkonmuodostus mahdollistavat langattomien anturiverkkojen käytön sovelluksissa, joihin perinteiset anturit eivät sovellu.

Yleensä anturisolmut on levitetty maantieteellisesti isolle alueelle. Näin ollen suora viestintä anturisolmulta tiedonkeruusolmulle on mahdotonta. Jotta anturisolmujen mittaama data saadaan siirrettyä tiedonkeruusolmulle, täytyy anturisolmujen myös välittää dataa eteenpäin verkossa. Reititysprotokollan tehtävänä on tarvittavien reititystaulujen muodostaminen ja ylläpitäminen. Tutkielmassa käsitellään omassa luvussaan reititystä, koska se on keskeinen osa klusterointiprotokollan toimintaa. Klusterinmuodostuksen lisäksi klusterointiprotokollan täytyy osata myös reitittää dataa niin klusterin sisällä kuin klustereiden välillä. Klusterointiprotokollasta voidaan käyttää myös nimitystä klusteripohjainen reititysprotokolla.

Anturisolmujen resurssit ovat hyvin rajalliset, joten energian säästäminen on langattomissa anturiverkoissa ensisijaisen tärkeää. Verkon klusteroinnilla voidaan vähentää verkon viestiliikennettä ja samalla pienentää anturisolmujen energiankulutusta. Sopivan klusterointiprotokollan valinta voi olla vaikeaa, sillä sovellus asettaa protokollalle omat vaatimuksensa. Vaatimuksia voivat olla esimerkiksi datan lyhyt saantiviive, datan tarkkuus, datalähteen tarkka paikantaminen tai vikasietoisuus. Varsinkin turvallisuuteen

liittyvissä sovelluksissa on tärkeää, että viestit kulkevat varmasti perille. Esimerkiksi jos potilas lähettää hätäviestin, on elintärkeää että viesti saapuu perille. Myös käytettävien anturisolmujen resurssit sekä verkon toimintaympäristö täytyy huomioida protokollan valinnassa. Klusterointiprotokollan suoritus tapahtuu joko keskitetysti esimerkiksi tukiaseman toimesta tai hajautetusti paikallisissa anturisolmuissa. Monet keskitetyt klusterointiprotokollat vaativat tarkan tiedon verkosta, joten ne eivät sovellu kovin suuriin verkkoihin. Hajautetut menetelmät ovat yleisempiä, koska langattomat anturiverkot koostuvat usein suuresta määrästä anturisolmuja ja ne sopivat paremmin suuriin verkkoihin.

Klusterointiprotokollien suoritus tapahtuu kierroksittain, jossa yksi kierros sisältää pääsolmun valinnan, klusterin muodostuksen, datan kokoamisen ja tiedonsiirron. Nämä vaiheet poikkeavat eri klusterointiprotokollissa ja vaikuttavat protokollan valintaan. Klusterointiprotokollan valinnassa on tärkeää huomioida miten pääsolmu valitaan sekä huomioida pääsolmujen ominaisuudet ja tehtävät. Pääsolmujen valinta voidaan tehdä ennalta määräten, todennäköisyyteen perustuen tai deterministisesti. Ennalta määrätty ja todennäköisyyteen perustuva pääsolmun valinta ovat varsin yksinkertaisia eivätkä vaadi niin paljon viestintää kuin deterministiset protokollat. Deterministisissä protokollissa pääsolmujen valinta tehdään paikallisten mittareiden mukaan, jolloin pääsolmun valinnassa voidaan huomioida esimerkiksi anturisolmujen jäljellä oleva energia ja jakaa näin anturisolmujen energiankulutusta tasaisemmin. Myös muodostettavien klustereiden lukumäärä, koko ja tiheys kannattaa huomioida protokollan valinnassa. Esimerkiksi tiheiden ja suurien klustereiden haasteena on energian kulutus.

Klusterin sisäinen ja klustereiden välinen tiedonsiirto täytyy myös huomioida protokollan valinnassa, sillä jokaisen klusterointiprotokollan yksi tärkeimmistä tavoitteista on verkon liitettävyys. Klusterin sisäinen tiedonsiirto voi tapahtua joko yhden tai monen hypyn menetelmällä. Energiankulutuksen kannalta olisi aina parasta, jos klusterin sisäinen tiedonsiirto tapahtuisi suoraan yhden hypyn menetelmällä. Tämä ei kuitenkaan ole aina mahdollista, sillä anturisolmun toimintasäde voi olla rajallinen tai pääsolmujen määrä on rajattu. Myös klustereiden välinen tiedonsiirto voi tapahtua joko yhden hypyn tai monen hypyn menetelmällä. Pitkänkantaman viestintä vaatii kuitenkin enemmän energiaa, joten

usein on parempi vaihtoehto käyttää monen hypyn viestintää. Klusterointiprotokollissa suoritetaan yleensä datan kokoamista. Jotta datan kokoamisesta olisi hyötyä, täytyy datan kokoamispaikat sijoittaa oikein. Datan kokoamisen tulisikin tapahtua lähellä lähteitä ja mahdollisimman varhain. Datan kokoaminen aiheuttaa aina viivettä verkkoon, joten sen vaikutusta täytyy arvioida varsinkin viiveherkissä sovelluksissa.

Tutkielmassa vertaillaan klusterointiprotokollien MH-LEACH ja DSBCA toimintaa kivimurskaimen melua mittaavassa langattomassa anturiverkossa. Kivimurskaimen meluverkko on hyvä esimerkki sovelluksesta, joka ohjaa vahvasti klusterointiprotokollan valintaa. Siinä klustereiden tulisi sijaita jonossa melulähteestä poispäin, klustereiden tulisi pysyä mahdollisimman stabiileina ja päällekkäisiä klustereita ei saisi muodostua. Tällöin melun mittaus olisi tarkkaa ja melun vaimenemista voitaisiin arvioida haluttuun suuntaan. Simuloidussa verkossa oli aseteltu 12 kappaletta anturisolmuja siten, että ne sijaitsivat suurin piirtein kahdessa jonossa kivimurskaimesta poispäin. Simuloinnissa huomattiin, että kumpikaan protokolla ei suoriutunut klusteroinnissa optimaalisesti. Todennäköisyyteen perustuvassa MH-LEACH -protokollassa pääsolmujen lukumäärä vaihtelee suuresti ja muodostetut klusterit saattavat kattaa vain tietyn osan alueesta. DSBCA-protokolla suoriutui protokollista paremmin ja muodosti tasapainoisempia klustereita kuin MH-LEACH. Myös klustereiden lukumäärä pysyi samana, mutta klustereiden jäsensolmujen määrä vaihteli ja myös päällekkäisiä klustereita muodostui. Tutkielmassa havaittiin, että DSBCA-protokollalla saadaan aikaiseksi optimaalinen verkko, kunhan anturisolmut ovat sijoitettu oikein ja klusterointiparametrit ovat oikein asetettu.

## Lähteet

- [1] Abbasi A. A. ja Younis M., *A survey on clustering algorithms for wireless sensor networks*, Computer Communications, Vol. 30, No. 14-15 (2007), s.2826-2841.
- [2] Ács G. ja Buttyán L., “A Taxonomy of Routing Protocols for Wireless Sensor Networks”, Budapest University of Technology and Economics, Department of Telecommunications, Laboratory of Cryptography and Systems Security (2007).
- [3] Afsar M. M. ja Tayarani-N M.-H., *Clustering in sensor networks: A literature survey*, Journal of Network and Computer Applications, Vol. 46 (2014), s.198-226.
- [4] Akyildiz I., Su W., Sankarasubramaniam Y. ja Cayirci E., *A Survey on Sensor Networks*, IEEE Communications Magazine, Vol. 40, No. 8 (2002), s. 102–114.
- [5] Al-Karaki J. N. ja Kamal A. E., *Routing Techniques in Wireless Sensor Networks: A Survey*, IEEE Wireless Communications, Vol. 11, No. 6 (2004), s. 6-28.
- [6] Anttila A., ”TCP/IP tekniikka”, Helsinki Media, Juva, 2001.
- [7] Bandyopadhyay S. ja Coyle E. J., *An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks*, kirjassa “INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications”, s. 1713-1723, 2003.
- [8] Basagni S., *Distributed Clustering Algorithm for Ad Hoc Networks*, kirjassa “Proceedings Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)”, IEEE Computer Society, s. 310-315, 1999.
- [9] Bennani K. ja El Ghanami D., *Particle swarm optimization based clustering in wireless sensor networks: the effectiveness of distance altering*, kirjassa “2012 International Conference on Complex Systems (ICCS)”, IEEE, s. 1-4, 2012.

- [10] Broder A. ja Mitzenmacher M., *Optimal Plans for Aggregation*, kirjassa “Proceedings of the 21st Annual Symposium on Principles of Distributed Computing”, ACM Press, s. 144-152, 2002.
- [11] Buttyan L. ja Schaffer P., *Position-Based Aggregator Node Election in Wireless Sensor Networks*, kirjassa “IEEE International Conference on Mobile Adhoc and Sensor Systems”, IEEE, s.1-9, 2007.
- [12] Chan H. ja Perrig A., *ACE: An Emergent Algorithm for Highly Uniform Cluster Formation*, kirjassa “Proceedings of the First European Workshop on Sensor Networks”, Springer Berlin Heidelberg, s.154-171, 2004.
- [13] Dargie W. ja Poellabauer C., “Fundamentals of Wireless Sensor Networks : Theory and Practice”, John Wiley & Sons, 2010.
- [14] Deng S., Li J. ja Shen L., *Mobility-based clustering protocol for wireless sensor networks with mobile nodes*, IET Wireless Sensor Systems, Vol. 1, No. 1 (2011), s.39-47.
- [15] Estrin D., Govindan R., Heidemann J. ja Kumar S., *Next Century Challenges: Scalable Coordination in Sensor Networks*, kirjassa “In MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking” , ACM, s. 263-270, 1999.
- [16] Feng Z. ja Leonidas G., “Wireless Sensor Networks : an information processing approach”, Morgan Kaufmann, San Fransisco, 2004.
- [17] Förster Al., Förster An. ja Murphy A. L., *Optimal cluster sizes for wireless sensor networks: An experimental analysis*, kirjassa “Ad Hoc Networks: First International Conference, ADHOCNETS 2009”, Springer Berlin Heidelberg , s. 49-63, 2010.
- [18] Gupta P. ja Kumar P. R., *Critical Power for Asymptotic Connectivity*, kirjassa “Proceedings of the 37th IEEE Conference on Decision and Control”, IEEE, s. 1106-1110, 1998.

- [19] Gupta, S. K., Jain, N. ja Sinha P., *Clustering Protocols in Wireless Sensor Networks: A Survey*, International Journal of Applied Information Systems (IJ AIS), vol. 5, No. 2 (2013), s.41-50.
- [20] Gutierrez J., Callaway E. ja Barrett R., “Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensors with IEEE 802.15.4”, Wiley-IEEE Standards Association, 2007.
- [21] Heinzelman W. R., Chandrakasan A. ja Balakrishnan H., *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*, kirjassa “Proceedings of the 33rd Annual Hawaii International Conference on System Sciences”, IEEE, s. 1-10, 2000.
- [22] Hussein M. S., *Survey of Routing Protocols in Wireless Sensor Networks*, International Journal of Sensors and Sensor Networks, Vol. 2, No.1. (2014), s. 1-6.
- [23] 802.15.4-2006: IEEE Standard for Information Technology, Local and Metropolitan Area Networks, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs), IEEE Std., Sept. 8, 2006.
- [24] Intanagonwiwat C., Govindan R., Estrin D., Heidemann J. ja Silva F., *Directed Diffusion for Wireless Sensor Networking*, IEEE/ACM Transactions on Networking, Volume 11, No. 1 (2003), s. 2-16.
- [25] Jauhiainen T., Vuorinen H.S. ja Heinonen-Guzejev M., ”Ympäristömelun vaikutukset”, Suomen ympäristö 3/2007, Ympäristöministeriö (2007).
- [26] Jung W.-S., Lim K.-W., Ko Y.-B. ja Park S.-J., *A Hybrid Approach for Clustering-based Data Aggregation in Wireless Sensor Networks*, kirjassa “Third International Conference on Digital Society”, IEEE Computer Society, s.112-117, 2009.
- [27] Karl H. ja Willig A., “Protocols and Architectures for Wireless Sensor Networks”, John Wiley & Sons Ltd, England, 2005.



- [28] Kivelä I. ja Hakala I., *Area-based environmental noise measurements with a wireless sensor network*, kirjassa “Proceedings of Euronoise 2015”, C. Glorieux, Ed., s. 2351-2356, 2015.
- [29] Krishnamachari B., Estrin D. ja Wicker S., *The Impact of Data Aggregation in Wireless Sensor Networks*, kirjassa “Proceedings of 22nd International Conference on Distributed Computing Systems Workshops, 2002”, IEEE, s. 575–578, 2002.
- [30] Kulik J., Heinzelman W. R. ja Balakrishnan H., *Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks*, *Wireless Networks*, Vol. 8, No. 2/3 (2002), s. 169-185.
- [31] Lee J.-S. ja Cheng W.-L., *Fuzzy-Logic-Based Clustering Approach for Wireless Sensor Networks Using Energy Predication*, *IEEE Sensors Journal*, Vol. 12, No. 9 (2012), s.2891-2897.
- [32] Liao Y., Qi H. ja Li W., *Load-Balanced Clustering Algorithm With Distributed Self-Organization for Wireless Sensor Networks*, *IEEE sensors journal*, Vol. 13, No. 5 (2013), s.1498-1506.
- [33] Liu J.-S. ja Lin C.-H. R., *Energy-efficiency clustering protocol in wireless sensor networks*, *Ad Hoc Networks*, Vol. 3, No. 3 (2005), s. 371-388.
- [34] Liu X., *A Survey on Clustering Routing Protocols in Wireless Sensor Networks*, *Sensors* 12, No.8 (2012), s.11113-11153.
- [35] Luke S., “Essentials of Metaheuristics”, saatavilla WWW-muodossa <URL: <http://cs.gmu.edu/~sean/book/metaheuristics/Essentials.pdf> >, viitattu 12.4.2015.
- [36] Muruganathan S. D., Ma D. C., Bhasin R. I. ja Fapojuwo A. O., *A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks*, *IEEE Radio Communications*, Vol. 43, No. 3 (2005), s.8-13.

- [37] Naeimi S., Ghafghazi H., Chow C.-O. ja Ishii H., *A Survey on the Taxonomy of Cluster-Based Routing Protocols for Homogeneous Wireless Sensor Networks*, *Sensors* 12, No. 6 (2012), s.7350-7409.
- [38] Neto J. H. B., Rego A., Cardoso A. R. ja Celestino Jr. J. *MH-LEACH: A Distributed Algorithm for Multi-Hop Communication in Wireless Sensor Networks*, kirjassa “The Thirteenth International Conference on Networks”, IARIA, s. 55–61, 2014.
- [39] Norouzi A. ja Zaim A. H., *An Integrative Comparison of Energy Efficient Routing Protocols in Wireless Sensor Network*, *Wireless Sensor Network*, Vol. 4, No. 3. (2012), s. 65-75.
- [40] Pantazis N. A., Nikolidakis S. A. ja Vergados D. D., *Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey*, *IEEE Communications Surveys & Tutorials*, Vol. 15, No. 2 (2013), s.551-591.
- [41] Park S., *Performance Evaluation of Data Aggregation Schemes in Wireless Sensor Networks*, kirjassa “2006 IEEE Wireless Communications and Networking Conference”, IEEE Communications Society, s. 413-418, 2006.
- [42] Radi M., Dezfouli B., Bakar A. K. ja Lee M., *Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges*, *Sensors* 12, No. 1 (2012), s.650-685.
- [43] Sohrabi K., Gao J., Ailawadhi V. ja Pottie G. J., *Protocols for self-organization of a wireless sensor network*, *IEEE Personal Communications*, Vol. 7, No. 5 (2000), s.16-27.
- [44] Soro S. ja Heinzelman W. B., *Prolonging the Lifetime of Wireless Sensor Networks via Unequal Clustering*, kirjassa “Proceedings. 19th IEEE International Parallel and Distributed Processing Symposium”, IEEE, s. 1-8, 2005.
- [45] Tan W. M. ja Jarvis S. A., *On the design of an energy-harvesting noise-sensing WSN mote*, *EURASIP Journal on Wireless Communications and Networking* (2014), s. 1–18.

- [46] Tilak S., Abu-Ghazaleh N. B. ja Heinzelman W., *A Taxonomy of Wireless Micro-Sensor Network Models*, ACM Mobile Computing and Communications Review, Vol. 1, No. 2 (2002), s. 1-8.
- [47] Wei D., Kaplan S. ja Chan H. A., *Energy Efficient Clustering Algorithms for Wireless Sensor Networks*, kirjassa "IEEE International Conference on Communications Workshops", IEEE Communications Society, s.236-240, 2008.
- [48] Younis O. ja Fahmy S., *HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks*, IEEE Transactions on Mobile Computing, Vol. 3, No. 4 (2004), s. 366–379.
- [49] Younis O., Krunz M. ja Ramasubramaniam S., *Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges*, IEEE Network, vol. 20, No. 3 (2006), s. 20–25.
- [50] Zheng J. ja Jamalipour A., "Wireless Sensor Networks: A Networking Perspective", John Wiley & Sons, Hoboken, New Jersey, 2009.