Jani Luoma

LANGATTOMILLE ANTURIVERKOILLE SOVELTUVAT MAC-PROTOKOLLAT

Tietotekniikan Pro Gradu -tutkielma

Ohjelmistotekniikan linja

5.6.2008

Jyväskylän yliopisto

Tietotekniikan laitos

Tekijä: Jani Luoma

Yhteystiedot: email: jani.luoma@nsn.com

Työn nimi: LANGATTOMILLE ANTURIVERKOILLE SOVELTUVAT MAC-PROTOKOLLAT

Title in English: MAC PROTOCOLS FOR WIRELESS SENSOR NETWORKS

Työ: Pro Gradu

Sivumäärä: 100

Linja: Ohjelmistotekniikka

Teettäjä: Kokkolan Yliopistokeskus Chydenius, Jyväskylän yliopisto, tietotekniikan laitos

Avainsanat: Langattomat anturiverkot, virransäästö, MAC-protokolla

Keywords: Wireless Sensor Networks, energy efficiency, MAC protocol

Tiivistelmä: Yksi suurimmista haasteista anturiverkkojen kehityksessä on noodien eliniän pidentäminen. MAC-kerroksen toiminnalla on suuri merkitys myös virrankulutuksen hallinnassa. Ainoa langattomille anturiverkoille suunniteltu standardoitu protokollapino on IEEE 802.15.4, joka sisältää sekä fyysisen että MAC-kerroksen määrittelyt. Kyseisen standardin MAC-ratkaisun lisäksi tässä työssä käsitellään useita erilaisia, tutkimusasteella olevia, anturiverkoille suunniteltuja MAC-ratkaisuja ja vertaillaan niiden suorituskykyä erityisesti virransäästön suhteen.

Abstract: One of the most challenging tasks in sensor network development is to prolong the lifetime of the nodes. Functionality of the MAC layer has a big impact on power consumption of the network nodes. So far, the only protocol stack which is originally designed for Wireless Sensor Networks, and has been approved as a standard, is the IEEE 802.15.4. This standard defines the physical and the MAC layer. In addition to this

standard MAC solution, also other MAC layer solutions are compared in this paper. The other MAC protocols are proposals for a good Wireless Sensor Network MAC solution, and in research state. Special attention is paid to energy efficiency.

Termiluettelo

Ad hoc [Latin.]	tätä tarkoitusta varten, tiettyä tarkoitusta varten
ADC	Analogue to Digital Converter
AI-LMAC	An Adaptive, Information-centric and Lightweight Medium Access Protocol
BER	Bit Error Rate
B-MAC	Berkeley Media Access Control
Broadcast Message	Yleislähetyssanoma jota ei ole osoitettu tietylle vastaanottajalle tai vastaanottajaryhmälle
CAP	Contention Access Period
CCA	Clear Channel Assesment
CFP	Contention Free Period
CSMA	Carrier Sense Multiple Access
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear To Send
DDT	Data Distribution Table
DOS	Denial Of Service
ECN	Explicit Content Notification
FFD	Full Functionality Device

GTS	Guaranteed Time Slot
HCL	High Contention Level
IP	Internet Protocol
IPv6	Internet Protocol version 6
LAN	Local Area Network
LCL	Low Contention Level
LLC	Link Layer Control
L-MAC	Lightweight Medium Access Protocol
LPL	Low Power Listening
LR-WPAN	Low Rate Wireless Personal Area Networks
MAC	Medium Access Control
MAN	Metropolitan Area Network
MIC	Message Integrity Code
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
MSN	Maximum Slot Number
NAV	Network Allocation Vector
NFS	Network File System
OSI	Open Systems Interconnection
PAN	Personal Area Network

PDU	Protocol Data Unit
PHR	Physical Header
PPDU	Physical Protocol Data Unit
RAM	Random Access Memory
RFD	Reduced Functionality Device
ROM	Read Only Memory
RPC	Remote Procedure Call
RSSI	Receive Signal Strength Indicator
RTS	Request To Send
SHR	Synchronization Header
S-MAC	Sensor-Medium Access Control
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TF	Time Frame
T-MAC	Timeout Medium Access Control
UDP	User Datagram Protocol
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Networks
Z-MAC	Zebra Medium Access Control

Sisältö

1 Johdanto

Langattomat anturiverkot on muutamien lähivuosien aikana voimakkaasti kasvanut tutkimuksen ala tietoliikenteessä. Viimeaikainen kehitys langattomassa viestinnässä sekä digitaalielektroniikassa on mahdollistanut entistä pienempien ja edullisempien laitteiden valmistuksen. Verkoissa käytettävien laitteiden, anturinoodien, toivottavia ominaisuuksia ovat muun muassa pienikokoisuus, pitkä elinikä ja edullinen hinta. Nämä vaatimukset ovat osittain keskenään ristiriitaisia, joten on ymmärrettävää, että ne asettavat suuria haasteita ja rajoitteita niin piirisuunnittelulle, kuin myös käytettävälle sovellukselle ja protokollaratkaisuille.

Vaikka jonkintyyppisiä anturiverkkoja on jo kaupallisessakin käytössä, ovat lupaavimmat sovellukset vasta tekemässä tuloaan markkinoille. Nämä sovellukset yleensä vaativat, että anturit eivät ole riippuvaisia ulkoisesta energianlähteestä. Näin ollen yksi merkittävä tekijä langattomien anturiverkkojen tutkimuskohteena ja kaupallisten sovellusten markkinoille tulon nopeuttajana on virransäästö. Yleisimpiä käyttökohteita, joista langattomien anturiverkkojen yhteydessä puhutaan, ovat muun muassa ympäristöntarkkailu, sotilasteknologia, lääketieteelliset sovellukset ja älykkäät toimistotilat. Anturiverkoista kerrotaan yleisellä tasolla enemmän luvussa 2, jossa käydään läpi myös tärkeimmät käsitteet.

MAC-protokollan toteutuksella on hyvin suuri vaikutus tiedonsiirtoverkon toimintaan, oli kyse sitten minkä tyyppisestä verkosta tahansa. Ominaisuuksia, joiden perusteella anturiverkkojen MAC-protokollaa voidaan arvioida, ovat muun muassa törmäysten välttäminen, energiansäästö, skaalautuvuus, kanavan käyttöaste, viive, välityskyky ja tasapuolisuus.

Tässä työssä käsitellään erilaisia anturiverkoille suunniteltuja MAC-protokollakerroksen ratkaisuja ja vertaillaan niiden ominaisuuksia keskittyen erityisesti virransäästöominaisuuksiin. Anturiverkkojen MAC-protokollat poikkeavat merkittävästi perinteisistä ratkaisuista. Käytettävällä MAC-ratkaisulla voidaan hyvin merkittävästi vaikuttaa muun muassa virransäästöön ja täten koko anturiverkon elinikään.

Käsiteltäväksi valitut MAC-ratkaisut on pyritty valitsemaan siten, että ne olisivat ominaisuuksiltaan erilaisia. Mukana on tällä hetkellä ainoana standardina olemassa oleva IEEE 802.15.4, jota pyritään vertaamaan ominaisuuksiltaan muihin, tutkimusasteella oleviin, MAC-ehdotelmiin.

2 Langattomat anturiverkot

Tässä luvussa käydään läpi tärkeimmät anturiverkkoihin liittyvät käsitteet sekä verrataan langattomien anturiverkkojen ominaisuuksia perinteisiin verkkoihin nähden.

Puhuttaessa yleisesti anturiverkoista voidaan ajatella, että esimerkiksi kansalliset sähkönjakeluverkot monine antureineen muodostavat anturiverkon. Toinen esimerkki varhaisista anturiverkoista on tutka-asemien muodostama lennonjohtojärjestelmä. Nämä järjestelmät kehitettiin kauan ennen kuin itse termi anturiverkot tuli yleiseen käyttöön. [6]

Tässä työssä tarkastellaan kuitenkin hyvin toisentyyppisiä anturiverkkoja. Nämä anturiverkot ovat langattomia, radiotien välityksellä kommunikoivia ja usein vailla etukäteen määriteltyä infrastruktuuria. Toisin sanoen verkkojen levittäminen ja käyttöönotto tapahtuu *ad hoc* -tyyppisesti. Myös anturiverkon laitteiden koko on hyvin pieni, ja laitteet saattavat sovelluksesta riippuen liikkua verkossa huomattavasti. Englanninkielisessä kirjallisuudessa näistä langattomista anturiverkoista on käyttöön vakiintunut lyhenne WSN (*Wireless Sensor Networks)*.

Monet työssä esitetyistä käyttökohteista eivät vielä ole ajankohtaisia laitteiden koosta, kustannuksista sekä sovelluksen käyttöön liian lyhyestä käyttöiästä johtuen. Työn yksi tarkoitus on perehtyä nimenomaan käyttöiän pidentämiseen MAC-kerroksen protokollaratkaisuilla.

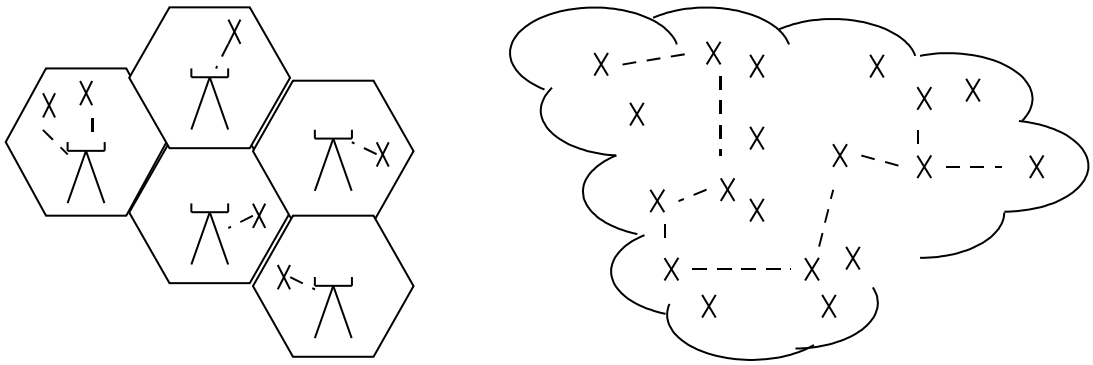
2.1 Ad hoc -verkot

Useissa anturiverkkoja käsittelevissä artikkeleissa mainitaan termi *Ad Hoc*, joka on latinaa ja on käännettynä: tätä tarkoitusta varten, tiettyä tarkoitusta varten. Tietoverkkojen yhteydessä käytettynä tällä tarkoitetaan rakenteettomia verkkoja, toisin sanoen verkkoja, joilla ei ole tarkkaa ennalta määriteltyä hierarkkiaa.

Ad hoc -verkot ovat langattomia, usein *multihop*-tyyppisiä verkkoja, joissa verkon toimilaitteet eli noodit ylläpitävät verkon yhteyskelpoisuutta. Esimerkkinä *ad hoc* -verkosta on tilanne, jossa urheilutapahtuman ajaksi tai luonnonkatastrofin jälkeen joudutaan pystyttämään langaton tiedonsiirtoverkko. Verkko asennettaisiin ainoastaan tilapäistä

käyttöä varten. Toinen esimerkki tilapäisen verkon käytöstä on palaveri, jossa osallistujat voivat jakaa tietoa toistensa kanssa päätelaitteiden avulla [27]. *Ad hoc* -verkko koostuu useista noodeista (*engl. node*), jotka konfiguroituvat itsenäisesti, ilman mitään ulkopuolista infrastruktuuria, muodostaen täten verkkorakenteen.

Erona perinteisiin verkkoihin on myös se, että *ad hoc* -verkon noodit joutuvat suorittamaan moninaisia tehtäviä, jotka perinteisissä verkoissa on keskitetty tarkoitusta varten kehitetyille laitteille. Ominaista ad hoc -verkoille on muun muassa tukiaseman puuttuminen, muuttuva rakenne sekä verkon noodeille asetettava reititysvaatimus [7]. Kuvassa 1 on pyritty havainnollistamaan eroa perinteisten, rakenteellisten, ja *ad hoc* -tyyppisten verkkojen välillä.



Kuva 1. Rakenteellinen ja *ad hoc* -verkko

Yksi hyvin tärkeä *ad hoc* -verkon ominaisuus on verkon noodien liikkuvuus *(engl. mobility)*. Liikkuvuus käsittää sekä sijainnin muutoksen, noodin poistumisen verkosta, että uuden noodin mukaan liittymisen. Liikkuvuudella on suuria vaikutuksia jokaiseen verkon protokollakerrokseen. Protokollakerroksia käsitellään myöhemmin luvussa 3. [10]

2.2 Langattomien anturiverkkojen määritelmä

Langaton anturiverkko saattaa koostua kymmenistä tai jopa tuhansista noodeista, jotka sijaitsevat hyvin lähellä toisiaan. Myöhemmin tässä työssä käytetään termiä anturiverkko, jolla siis tarkoitetaan langatonta anturiverkkoa.

Sovellusalueita, joilla anturiverkkoja tullaan käyttämään, ovat muun muassa terveydenhoito, sotilasteknologia, ympäristöntarkkailu sekä turvallisuus. Anturiverkkojen sovellusalueita käydään läpi tarkemmin kappaleessa 2.5.

Jokainen anturiverkon noodi sisältää virtalähteen, prosessorin, pienen määrän muistia, yhden tai useamman anturin sekä radio-osan. Noodien rakennetta käsitellään tarkemmin luvussa 2.3. Nämä anturinoodit sijoitetaan joko tarkkailtavan ilmiön alueelle tai hyvin lähelle sitä. [3]

Noodien tarkka sijainti ei ole ennalta määritelty, vaan ne voidaan sovellusalueesta riippuen sijoittaa kohdealueelle jopa pudottamalla lentokoneesta. Tämä asettaa anturiverkolle itseorganisoituvuuden vaatimuksen.

Anturiverkon ominaisuuksiin kuuluu myös, että noodit eivät lähetä välttämättä kaikkea keräämäänsä dataa käsittelemättömänä eteenpäin, vaan tekevät yksinkertaisia toimenpiteitä keräämälleen tiedolle ja lähettävät ainoastaan tarpeellisen ja osittain käsitellyn datan eteenpäin. [3]

Suurimmat erot anturiverkkojen ja ad hoc -verkkojen välillä ovat [3]

- noodien lukumäärä voi olla suuruusluokassa moninkertainen ad hoc verkkoon verrattuna
- anturinoodit ovat erittäin tiheästi sijoitettuna
- anturiverkon noodeilla on merkittäviä rajoituksia virransaannin, prosessointitehon ja muistikapasiteetin suhteen
- anturiverkon noodeilla ei välttämättä ole yksilöllistä identiteettitunnistetta, johtuen anturinoodien mahdollisesti hyvin suuresta lukumäärästä

Anturiverkon eliniän pituuden jatkaminen on merkittävä tutkimuksen kohde anturiverkkojen alueella. Verkon noodeille ja siten koko verkolle asetettavia vaatimuksia ovat halpa hinta, pienikokoisuus, automaattinen verkkokonfigurointi, topologiamuutokset

sekä pitkä elinikä. Nämä vaatimukset ovat osittain keskenään ristiriitaisia. Esimerkiksi tiedon esiprosessoinnilla tähdätään verkon eliniän pidentämiseen virransäästön avulla. Prosessointi vaatii kuitenkin myös muistia, jota laitteissa hintavaatimuksen vuoksi on yleensä hyvin vähän. Mitä vähemmän radio-osan täytyy olla aktiivinen tai lähetystilassa, sitä vähemmän myös virtalähteen energiaa tarvitaan.

Anturiverkon eliniän pidentämiseen voidaan vaikuttaa monilla erilaisilla keinoilla. Tässä työssä keskitytään vertailemaan erilaisia MAC (*engl. Medium Access Control)* -protokollakerroksen ratkaisuja ja erityisesti menetelmiä, joilla verkon elinikää voidaan pidentää.

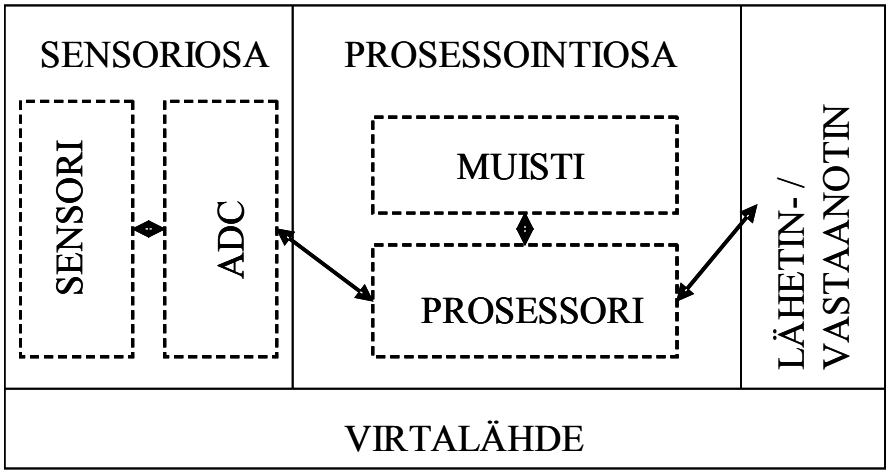
2.3 Anturinoodin rakenne

Anturiverkon noodi koostuu periaatteessa neljästä eri fyysisestä osasta: lähetin-vastaanotinyksiköstä, virtalähteestä, yhdestä tai useammasta anturista sekä prosessointiosasta. Joissakin lähteissä prosessorikontrolleri ja muistiosa esitetään omina pääkomponentteinaan. [2],[18]

Anturiosa voidaan jakaa kahteen alijärjestelmään: varsinaiseen anturiin sekä ADC *(engl. analog to digital converter)* muuntajaan. ADC-osa muuttaa siis anturilta tulevan analogisen signaalin prosessorille soveltuvaan digitaaliseen muotoon.

Prosessointiosa koostuu itse mikroprosessorista, sekä tarvittavista käyttömuistista *(Random Access Memory)* ja haihtumattomasta *(Read Only Memory)* -muistista.

Kuvassa 2 on esitetty lohkokaavio, joka pyrkii selventämään anturinoodin rakennetta.



Kuva 2. Anturinoodin lohkorakenne

Antureita on montaa eri tyyppiä, ja käytettävä anturi vaihtelee verkon sovellusalueesta riippuen. Yhdessä noodissa voi olla myös enemmän kuin yksi anturi.

Antureita on muun muassa mekaanisia, magneettisia, elektromagneettisia, akustisia, lämpötilaan reagoivia, optisia, sekä kemiallisiin ja biologisiin aineisiin reagoivia. Nämä anturityypit voidaan edelleen jakaa tarkempiin alaluokkiin, esimerkiksi mekaanisista antureista mainittakoon paineen, väännön, venymän ja värinän havainnointiin tarkoitetut anturit. Liikehavainnointiin voidaan käyttää esimerkiksi asennon, nopeuden, kulmanopeuden ja kiihtyvyyden ilmaisevia antureita. [8]

Tämän päivän teknologialla anturinoodin koko voi olla tyypillisesti verrattavissa korttipakkaan, mutta tulevaisuudessa tavoitteena ovat *smart dust* [34] -tyyppiset anturit, jotka ovat kooltaan nimensä mukaisesti hyvin pieniä. Mitä pienemmiksi ja pitkäikäisemmiksi anturinoodit voidaan tehdä, sitä laajempi on myös mahdollisten sovellusten kenttä. Tämän päivän anturinoodiratkaisuilla voidaan päästä kuukausien, jopa vuosien mittaiseen kestoon tavallisia paristoja käyttämällä. Tulevaisuuden anturinoodit voinevat käyttää aurinkoenergiaa lisäenergiana ja saavuttaa useiden vuosien eliniän.

Eräänä esimerkkinä yleisesti saatavilla olevasta kaupallisesta ratkaisusta on Crossbow Berkely Mote MICA2, jossa on 4kB EEPROM-muistia, 4kB SRAM-käyttömuistia ja 128kB ohjelmamuistia. 8-bittisen prosessorin kellotaajuus 8MHz ja laitteen koko ilman

paristoja 58x32x7 millimetriä. Painoa kyseisellä laitteella on 18 grammaa ilman paristoja. [9] Uudempia tuotteita edustavat muun muassa Jennicin ja Texas Instrumentsin ratkaisut. Jennic JN513x -sarjan laitteet jotka perustuvat 32-bittiselle RISC (*Reduced Instruction Set Computer*) -mikrokontrolleriarkkitehtuurille. Kyseisen sarjan laite sisältää 192kB lukumuistia, ja 8kB - 96kB käyttömuistia.

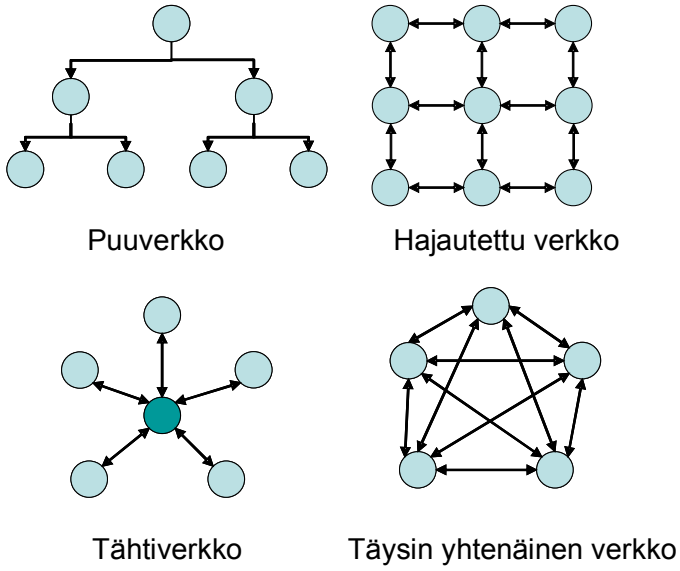
Mikrokontrollereita, joita usein käytetään anturinoodisovelluksissa, ovat muun muassa Intel StrongARM, TI MSP 430 sekä Atmel ATMega [18].

2.4 Anturiverkon rakenne

Anturiverkon voidaan yleensä ajatella koostuvan kahdentyyppisistä laitteista: tavallisesta anturinoodista sekä kerääjänoodista (*engl. sink node)*. Tavallinen noodi havainnoi ympäristöään, lähettää omia havaintojaan sekä edelleen lähettää muiden noodien dataa kerääjänoodille. Kerääjänoodi voi olla yhteydessä loppukäyttäjään esimerkiksi Internetin kautta [2].

Kerääjänoodi on kiinnostunut ainoastaan kollektiivisesta informaatiosta, jota saman havainnon tehneet noodit sille lähettävät [25]. Yksittäisen anturinoodin informaatiolla ei siis ole välttämättä suurta merkitystä. Sama filosofia pätee yleisestikin anturiverkkoihin: kaikki verkon noodit suorittavat yhteistä tehtävää.

Langattomien anturiverkkojen tärkeimpiä perustopologioita voidaan sanoa olevan tähtiverkon (*engl. star network)*, täysin yhtenäisen verkon (*engl. fully connected network)*, hajautetun verkon (*engl. mesh)* sekä puuverkon (*engl. tree network)*.



Kuva 3. Yleisimmät perusverkkotopologiat

Kuvassa 3 on esitetty tärkeimmät anturiverkkojen yhteydessä esiintyvät perustopologiat. Tähtiverkon etuihin kuuluu kommunikoinnin yksinkertaisuus. Jokainen noodi lähettää tietonsa keskusnoodille. Verkon laitteet eivät siis kommunikoi lainkaan keskenään, vaan ainoastaan keskusnoodin kanssa.

Tähtitopologian suuri puute on sen joustamattomuudessa. Verkon noodeja ei voida viedä kovin kauaksi keskusnoodista. Anturiverkkosovelluksen täytyy olla siis sen mukainen, että kyseinen topologia on järkevä vaihtoehto. Tähtitopologia on myös hyvin haavoittuva vikatilanteissa: mikäli keskusnoodi vikaantuu, on koko verkko poissa toiminnasta. Toisaalta tähtiverkon etuja ovat hyvin pieni viive viestinvälityksessä sekä keskitetty verkonhallinta. Sovellusalueita joissa tähtitopologia on käyttökelpoinen, ovat muun muassa kotiautomaatio, lelut sekä henkilökohtainen terveydenhuolto. [14]

Puuverkko on tähtiverkosta astetta pidemmälle jalostettu verkkotopologia. Kyseisen verkkotyypin juurinoodi toimii myös kerääjänoodina, ja tämän alapuolella sijaitsevat noodit ovat yhteydessä toisiinsa suorien kommunikaatiolinkkien välityksellä. [26]

Täysin yhtenäisessä verkossa kaikki noodit voivat kommunikoida toistensa kanssa. Kaikki noodit ovat toistensa kuuluvuusalueella, ja näin ollen noodit muodostavat useita kommunikointiyhteyksiä. Kyseinen verkkotyyppi on kuitenkin erikoistapaus, eikä tällaista

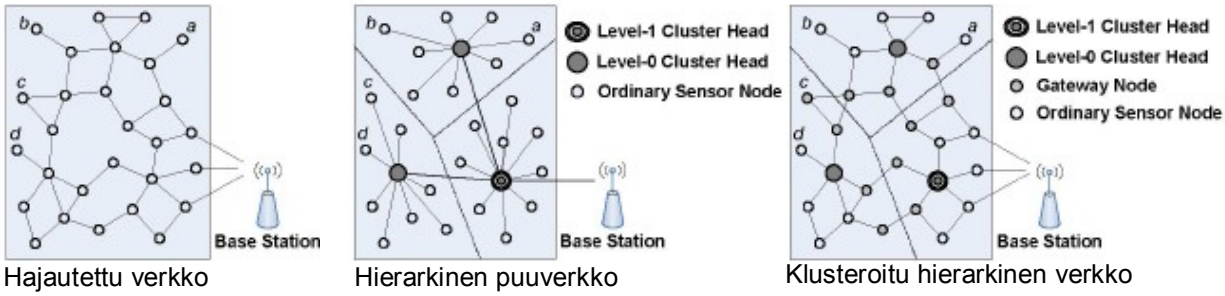
välttämättä esiinny liikkuvuutta vaativassa anturiverkkosovelluksessa. Mikäli kaikki verkon noodit kuitenkin ovat toistensa kuuluvuusalueella, on tällainen verkko paras mahdollinen vikasietoisuuden suhteen. Muut noodit eivät ole riippuvaisia yhden noodin viestinvälitysyhteyksistä, vaan voivat kommunikoida muiden noodien mahdollisesta vikaantumisesta huolimatta.

Hajautetussa verkossa noodit ovat kommunikointietäisyydellä yleensä lähimpiin naapureihinsa. Kyseinen verkkotyyppi on hyvin todennäköinen laajalle maantieteelliselle alueelle levitetyn anturiverkkosovelluksen tapauksessa. Koska noodien välillä on todennäköisesti useampia eri vaihtoehtoisia reititysmahdollisuuksia, on myös hajautettu verkko hyvin vikasietoinen. Vaikka hajautetussa verkossa kaikki noodit ovat yleensä samankaltaisia ominaisuuksiensa puolesta, voidaan joidenkin noodien kohdalla ottaa käyttöön lisäominaisuuksia, asettamalla niitä toimimaan ns. keskusnoodeina. [1]

Koska kaikki verkon noodit eivät ole toistensa kuuloetäisyydellä, täytyy noodien kyetä välittämään liikennettä myös eteenpäin. Hajautetut verkot ovat siis *multihop*-tyyppisiä, reititystoiminnallisuuden omaavia verkkoja. Yksi reititysvaatimuksen haittapuoli on, että tietyt noodit kuormittuvat muita enemmän, koska ne joutuvat reitittämään muiden noodien liikennettä. Virrankulutus verkon noodien välillä ei siis jakaudu tasaisesti.

Käyttämällä reititystä voidaan kuitenkin myös toteuttaa tiettyjä virransäästöominaisuuksia. Joissain tapauksissa on edullista lähettää sanoma toisen noodin kautta. Energian kulutuksen kannalta on edullisempaa käyttää reititystä kahden noodin välisessä sanomaliikenteessä, koska etäisyyden puolittaminen tarkoittaa tehon kulutuksen pudotusta neljännesosaan alkuperäisestä [18]. Haittapuolena reitityksessä on monimutkaisempi toteutus verkkokerroksella sekä tiedonsiirtonopeuden hidastuminen. Lisää tietoa reitityksestä, verkkotopologioihin liittyvistä ongelmista sekä näihin sovellettavista algoritmeista on saatavilla lähteestä [22].

Huomattavaa on, että käytännön anturiverkko saattaa rakenteeltaan koostua useista erityyppisistä verkkotopologioista, jotka ovat järjestyneet eri aliverkkoihin. Varsinkin laajalle alueelle levitettävät anturiverkkosovellukset ovat hyvin todennäköisesti topologian suhteen hybridiverkkoja. [1]



Kuva 4. Langattomien anturiverkkojen hybriditopologioita [26]

Kuvassa 4 on esitetty aiemmin perustopologioiden yhteydessä mainittu puuverkko havainnollisemmalla tavalla. Kerääjänoodi on siis juurinoodi ja eri hierarkkian mukaisesti noodit ovat yhteydessä toisiinsa suorien linkkien välityksellä.

Myös aiemmin mainittu hajautettu verkko on esitetty muodossa, joka paremmin havainnollistaa käytössä olevan anturiverkkosovelluksen noodien sijoittumista. Noodit eivät tule olemaan vakioetäisyydellä toisistaan tietyn rakenteen mukaisesti. Niillä ei myöskään ole kommunikointiyhteyttä välttämättä enempään kuin yhteen toiseen noodiin.

Verkon koon kasvaessa hyvin suureksi, on edullista pilkkoa verkon rakennetta pienempiin osiin. Koska hajautetussa verkossa joudutaan liikennettä reitittämään eri noodien välillä, täytyy noodien ylläpitää reititystaulukkoa muistissaan. Anturinoodien rajallisten ominaisuuksien vuoksi ei kovin suurten verkkojen reititystietoja voida ylläpitää noodien muistissa. Jotta muistirajoituksen ja suurten reititystaulukoiden aiheuttamasta ongelmasta päästään eroon, voidaan suurten hajautettujen verkkojen rakenne jakaa edelleen pienempiin klustereihin muodostaen näin niin sanottu klusteroitu hierarkkinen verkko.

Klusteroidussa hierarkkisessa verkossa noodit muodostavat paikallisen naapuri-informaation perusteella klustereita ja valitsevat näille klustereille *cluster head* -noodin, joka on vastuussa ko. klusterin liikenteestä toisen *cluster head* -noodin kanssa. Näiden paikallisten *cluster head* -noodien joukosta valitaan seuraavan tason *cluster head* -noodi. *Cluster head* -noodien valinta, sekä hierarkiatasojen lukumäärä voivat riippua useista eri tekijöistä, muun muassa käytettävissä olevasta energiansaannista, naapurien lukumäärästä, etäisyydestä, lähetysalueen kattavuudesta ja nooditiheydestä alueella. Merkittävin ero

hierarkkisen puurakenteen ja klusteroidun hierarkkisen verkon välillä on se, että jälkimmäisessä liikenne on *multihop*-tyyppistä, kun edellisessä kommunikoidaan yhden hypyn etäisyydellä olevan naapurin kanssa. [26]

Klusteroidun hierarkkisen topologian etuihin kuuluu jo aiemmin mainittu mahdollisuus erittäin suuriin verkkoihin. Haittapuolena tämä kuitenkin tuottaa sen, että verkon ylläpitoon liittyvää sanomanvaihtoa on enemmän kuin muissa topologioissa. [11]

2.5 Anturiverkkojen sovellusalueita

Tässä luvussa käydään läpi eräitä anturiverkkojen sovellusalueita. Tämä auttanee lukijaa ymmärtämään paremmin anturiverkoille asetettavia vaatimuksia ja anturiverkkojen erityispiirteitä. Johtuen siitä, että kyseinen teknologian ala on vielä suhteellisen uusi, eivät kaikki kuvatut sovellusalueet ole vielä tämän päivän toteutuksella mahdollisia. Tutkimuksen ja kehityksen myötä laitteiden koko, hinta ja elinikä voidaan kuitenkin saada sellaiselle tasolle, että kaikki seuraavat sovellukset ovat anturiverkoille toteutettavissa.

Anturiverkkojen vikasietoisuus, itseorganisoituminen sekä nopea levitys ja käyttöönotto tekevät tekniikasta erittäin lupaavan sotilaallisten sovellusten kannalta. Kyseisellä sovellusalueella anturiverkoille asetetaan myös vaatimuksia, joita yleensä muilla alueilla ei esiinny. Esimerkkinä voidaan mainita häirintä: vihollinen pyrkinee tekemään anturiverkon toimintakyvyttömäksi, ei pelkästään tuhoamalla anturinoodeja fyysisesti, vaan myös tarkoituksellisella häirinnällä. Vihollinen voi häiritä verkon toimintaa lähettämällä häirintäsignaalia samalla radiotaajuudella, jota anturiverkon noodit käyttävät kommunikointiin. Yhtä ainutta radiotaajuutta käyttävien anturiverkkojen häirintään tämä menetelmä on hyvin yksinkertainen ja tehokas. [37]

Sotilaallisiin sovelluksiin kuuluu muun muassa omien joukkojen, laitteiston ja ammusten monitorointi. Komentoporras voi reaaliajassa saada tietoa omien joukkojen tarkasta sijainnista, ammusten ja muun tarvittavan laitteiston saatavuudesta taistelukentällä. Jokainen ajoneuvo, sotilas tai kriittinen tarvike voidaan varustaa anturinoodilla, ja tiedot lähettää sink-noodille johtajien käyttöön. Tällaisesta tiedosta voidaan edelleen koostaa yhteenvetoja ylemmille tahoille, esimerkiksi strategisia päätöksiä varten. [2]

Vihollisen liikkeisiin kohdistuvia toimenpiteitä ovat muun muassa taistelukentän olosuhteiden ja vihollisen liikkeiden seuranta. Maastoon tai oletetuille kulkureiteille voidaan levittää anturiverkko ja näin voidaan saada tietoa vihollisen liikkeistä hyvin lyhyellä varoitusajalla. Kemiallisen ja biologisen sodankäynnin varalle anturiverkkoja voidaan käyttää antamaan lisäaikaa joukkojen suojautumiseen ja evakuointiin. Anturiverkkoja voidaan käyttää myös opastamaan älykkäitä ammuksia kohteeseen [2].

Anturiverkkoja voidaan käyttää esimerkiksi erilaisten luonnonkatastrofien ennakoimiseen ja seurantaan, kuten tulivuorien purkausten ennustamiseen, metsäpalojen tarkkailuun, tulvavaroitusten antamiseen sekä myös tavanomaisempiin tarkoituksiin, kuten eläinten seuraamiseen ja havainnointiin [2]. Aktiivisia tulivuoria voidaan tarkkailla anturiverkkojen avulla ja saada lisäaikaa evakuointia varten mahdollisen purkauksen varalta. [35] Maanviljelyksen yhteydessä anturiverkoille löytyy mahdollisia sovelluksia esimerkiksi tuholaisten havainnoinnissa, maaperän ja ilmaston ominaisuuksien muutosten seurannassa [2].

Mahdollisen onnettomuuden, esimerkiksi saaste-, myrkky- tai kaasupäästön seuraamisessa anturiverkko voidaan levittää lentokoneesta käsin tarkkailtavan ilmiön sisään tai välittömään läheisyyteen. Esimerkiksi myrkyllisen kaasupilven liikkumista ja hajoamista voitaisiin seurata lentokoneesta käsin levitettävällä anturiverkolla.

Sairaalassa jokaiselle potilaalle voidaan asentaa yksi tai useampia antureita, esimerkiksi yksi verenpaineen seurantaan ja toinen sydämen toimintaa varten. Myös potilaan lääkeallergiat voitaisiin tallentaa anturinoodin tietoihin. Jos lääkepakkauksiin voitaisiin integroida anturi, olisi tällöin väärän lääkityksen aiheuttama terveysriski minimaalinen. Lääkkeisiin integroitua anturisovellusta voitaisiin myös käyttää varaston seurantaan ja luvattoman käytön ehkäisemiseen [2]. Lääkäreillä anturi voisi olla mukana paikannusta varten, jolloin voitaisiin paikalle saada lähin vaadittavan erikoisalan asiantuntija.

Vanhusten terveydenhoidossa anturiverkoilla voitaisiin mahdollistaa vanhusten pidempään jatkuva kotona asuminen. Anturisovellusta voitaisiin käyttää vanhuksen asennon ja liikkumisen tarkkailuun. Havaittaessa jotakin hälyttävää hoitohenkilökunta saisi tiedon tästä välittömästi. [20],[32]

Kodin tavanomaisiin laitteisiin, kuten jääkaappiin, mikroaaltouuniin jne. voidaan integroida anturinoodi. Laitteet pystyvät täten kommunikoimaan keskenään sekä kytkeytymään ulkoisiin verkkoihin, kuten Internetiin [2].

Tällaisen melko perinteisen kotiautomaation lisäksi anturiverkot mahdollistavat ns. älykkäiden tilojen toteuttamisen. Esimerkkinä tällaisesta mainittakoon ns. älykäs lastentarha, jossa anturinoodeja on itse opetustilassa sekä leluissa ja muissa esineissä. Anturiverkkosovellus ei pelkästään keräisi tietoa lasten käyttäytymisestä, vaan myös reagoisi tilanteen mukaan tukeakseen lasten ongelmaratkaisukyvyn kehittymistä. Tällainen mahdollistaisi lasten yksilöllisten ominaisuuksien huomioimisen oppimistarkoituksessa. Myös opettaja pystyisi jatkuvasti seuraamaan lasten kehittymistä yksilötasolla [28].

Anturiverkkojen sovellusalueita on tutkimuksen alla hyvin paljon, ja eräitä mahdollisia kaupallisia alueita ovat muun muassa kulkuneuvojen seuranta ja varkauksien ehkäisy, älykkäät toimistot, interaktiiviset huoneet esimerkiksi museoissa, lelut, älykkäät ilmastointiratkaisut ja varastotietojen hallinta [2]. Myös teollisuudessa anturiverkkoja voidaan käyttää esimerkiksi koneiden kunnon tarkkailuun ja huoltovälin tarkempaan ennakointiin.

Alueita, joilla anturiverkkoja voidaan käyttää, löydetään jatkuvasti lisää. Mitä pienemmiksi ja pitkäikäisemmiksi laitteet saadaan, sitä laajempi sovelluskenttä voi olla.

3 Protokollat ja kerroksellisuus

Tässä luvussa käydään läpi tietoliikenneprotokollien kerroksellisen rakenteen käsitteitä sekä kerroksellisesta rakenteesta saavutettavia etuja. Kappaleessa 3.1 on lyhyt kuvaus yleisesti kirjallisuudessa viitatusta OSI (Open Systems Interface)-mallista. Kyseisen kappaleen teksti perustuu pääasiassa lähteiden [41] ja [4] käyttöön. Kappaleessa 3.2 kuvataan IEEE:n 802 standardin protokollapinon rakennetta, sekä eroja OSI-viitemalliin. Kappaleessa 3.3 esitellään IEEE 802.15.4 -standardin anturiverkkoihin tarkoitettu protokollaratkaisu.

3.1 OSI-viitemalli

Tietoliikenneprotokollat muodostuvat yleensä useista eri kerroksista. Hyvin usein protokollien kerrosmallista puhuttaessa viitataan OSI (*Open Systems Interconnection*)-kerrosmalliin. Kyseisen viitemallin tarkoitus on tarjota yleispätevä rakenne tietoliikennejärjestelmille ja toimia viitekehyksenä standardeille. [36]

Kerrosrakenteen periaatteena on se, että jokainen kerros käyttää alemman kerroksen sille tarjoamia palveluja. Kommunikaatio tietoliikenneverkoissa on hyvin monimutkaista, joten protokollan pilkkominen useisiin kerroksiin tuo tiettyjä etuja. Näitä ovat muun muassa seuraavat seikat: [36]

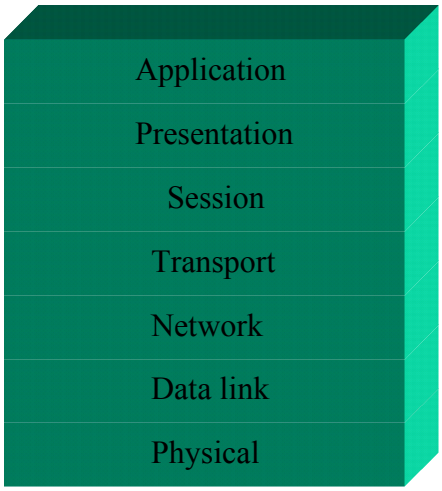
- Eri organisaatiot voivat olla vastuussa eri kerroksista.
- Testaus ja ylläpidettävyys helpottuvat, mikä käytännössä tarkoittaa ohjelmistovikojen vähentymistä.
- Protokollakerrosten vaihdettavuus. Pinon kerrosratkaisu voidaan vaihtaa toiseen tarvittaessa. Muut kerrokset eivät vaadi minkäänlaista muutosta, sillä rajapintamääritys pysyy yhä edelleen samana.

Kuvassa 5 on pyritty havainnollistamaan protokollien kerroksisuutta. Lähetyksessä N -kerroksen tietue pakataan otsikkotietoineen alempaan, N-1 kerroksen, tietueeseen, joka puolestaan sisältää omat otsikkotietonsa.

Kuva 5. Protokollakerroksien toimintaa kuvaava esimerkki

OSI-kerrosmallissa protokollapino muodostuu seitsemästä eri kerroksesta [41]. Kyseinen ratkaisu on kuitenkin sellaisenaan hyvin raskas, ja OSI-malli ei saavuttanutkaan kovin suurta levinneisyyttä. Hyvin useissa protokollaratkaisuissa kerrosten määrä on pienempi, esimerkiksi TCP/IP (*Transmission Control Protocol / Internet Protocol*), joka koostuu viidestä eri kerroksesta. OSI-mallia käytetään kuitenkin laajalti referenssimallina, joten sen läpikäyminen on perusteltua.

Laitteet kommunikoivat keskenään OSI-mallin kerrosten kautta. Kukin kerros kommunikoi vastapään vastaavan kerroksen kanssa. Laitteen sisällä kukin protokollakerros käyttää hyväkseen alemman kerroksen palveluita. Seuraavissa kohdissa käydään lyhyesti läpi kuvassa 6 esitettyjen OSI-mallin kerrosten tehtävät.[4]



Kuva 6. OSI-mallin protokollakerrokset

1. Fyysinen kerros (Physical layer)

   Fyysinen kerros tarjoaa mekaaniset, elektroniset, funktionaaliset ja proseduraaliset palvelut fyysisten yhteyksien muodostamiseen, ylläpitämiseen ja käytöstä poistamiseen bittitason siirtokerrosyhteyksille. Käsiteltäessä langattomia anturiverkkoja voidaan esimerkiksi radiotaajuuden käyttö lukea fyysiselle kerrokselle kuuluviin asioihin.

2. Siirtoyhteyskerros (Data Link layer)

   Siirtoyhteyskerros tarjoaa funktionaaliset ja proseduraaliset palvelut yhteydettömälle toimintatavalle verkkotietueiden välillä sekä siirtoyhteyksien muodostamisen, ylläpitämisen ja vapauttamisen yhteydelliselle toimintatavalle. Siirtoyhteyskerros on rakennettu yhden tai useamman fyysisen yhteyden päälle. Kerroksen tehtävä edellä mainitun kerrostoiminnan mukaan on muodostaa siirtoyhteyskehys, jonka sisälle verkkokerroksen tietue pakataan. Siirtoyhteyskerros tarjoaa luotettavan tiedonsiirron fyysisen yhteyden yli.

3. Verkkokerros (Network layer)

   Verkkokerroksen tehtävänä on pakata kuljetuskerrokselta saatu data verkkokerroksen pakettiin ja reitittää se vastaanottajan verkko-osoitteen perusteella seuraavalle linkille. Tätä prosessia kutsutaan reititykseksi.

4. Kuljetuskerros (Transport layer)

   Kuljetuskerroksen tehtävänä on pilkkoa ylemmiltä kerroksilta saatu data sopivankokoisiin segmentteihin ja välittää ne edelleen vastaanottajalle. Toimintatapoja on kaksi: yhteydellinen ja yhteydetön. Esimerkkeinä kuljetuskerroksen protokollista ovat TCP/IP -maailmasta tutut TCP (*Transmission Control Protocol*) ja UDP (*User Datagram Protocol*).

5. Yhteysjaksokerros (Session layer)

   Kerroksen tehtävänä on sovelluksen toimintojen koordinointi laitteiden välillä, josta esimerkkinä voidaan mainita lähetyksen käynnistäminen ja pysäyttäminen. Esimerkkeinä yhteysjaksokerroksen protokollista ovat NFS (*Network File System*) ja RPC (*Remote Procedure Call*).

6. Esityskerros (Presentation layer)

   Esitystapakerroksella on määritelty, millaisessa muodossa välitettävä informaatio esitetään, esimerkiksi JPEG-kuvaformaatti ja ASCII-merkistö. Esitystapakerros toimii myös tulkkina kääntäen esimerkiksi ASCII-merkistöä käyttävän sanoman johonkin toiseen merkistöön.

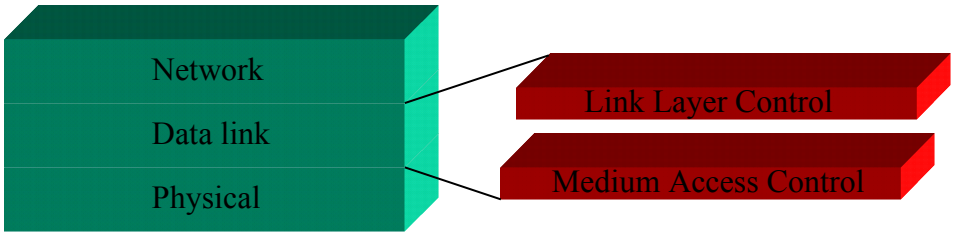
7. Sovelluskerros (Application layer)

   Sovelluskerroksen tehtävänä on tarjota verkkopalvelut varsinaisille sovellusohjelmille. Tällaisia palveluja ovat esimerkiksi tiedoston avaaminen verkkolevyltä, sähköpostin käyttäminen sekä verkkotulostus.

3.2 IEEE 802 standardiperhe

IEEE-standardointikomitea kehittää määrittelyjä LAN ja MAN (*engl. Local Area Network, Metropolitan Area Network)* paikallis- ja kaupunkiverkoille. Komitean alaisuudessa on useita eri työryhmiä, joista tunnetuimpia ovat muun muassa Token Ring (802.5), Ethernet (802.3), WLAN (802.11), langaton laajakaista (802.16) ja WPAN (802.15). [16]

IEEE tietoliikennestandardi 802 määrittelee ainoastaan kaksi alimmaista OSI-viitemallin mukaista protokollakerrosta: fyysisen (*Physical Layer*) ja siirtoyhteyskerroksen (*Data Link Layer*). OSI-mallin mukainen siirtoyhteyskerros on edelleen jaettu kahteen alikerrokseen *(engl. sublayer)*: LLC (*Link Layer Control)* - ja MAC-kerroksiin (*Medium Access Control)* kuvan 7 mukaisesti. [16]



Kuva 7. Siirtokerroksen jakautuminen alikerroksiin

Lyhyesti kuvattuna LLC-alikerroksen tehtävänä on toteuttaa luotettavuusvaatimus siirtoyhteyskerroksen osalta, kun taas MAC-alikerroksen tehtävä on vastata pakettien siirrosta laitteelta toiselle.

Tarkennettuna MAC-alikerroksen tehtäviä ovat muun muassa kehysten tunnistaminen ja rajaaminen, laiteosoitteiden hallinta, LLC-alikerroksen pakettien siirto linkin yli, suojautuminen tiedonsiirtovirheitä vastaan tarkistussummien avulla sekä fyysisen median käytön kontrollointi. [16]

Koska MAC-kerroksen yksi tehtävä on tarjota pääsy fyysiselle siirtotielle, voidaan tällä vaikuttaa hyvin voimakkaasti virransäästöön. Myöhemmin osoitetaan tarkemmin, että merkittävin osa virransäästöstä MAC-kerroksella saadaan aikaan toteuttamalla mahdollisimman pieni fyysisen median käyttöaste.

IEEE on määritellyt myös anturiverkoille soveltuvan protokollastandardin IEEE 802.15.4, joka määrittää protokollat sekä fyysiselle että MAC-kerrokselle. Kyseistä standardia käsitellään tarkemmin seuraavassa kappaleessa.

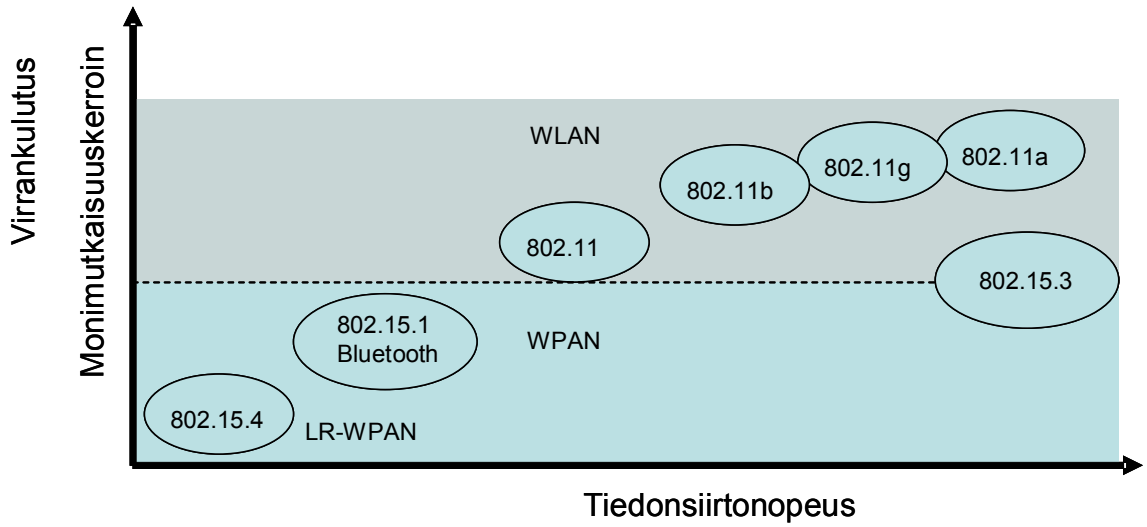
3.3 Langattomat anturiverkot ja 802 -standardiperhe

Tässä kappaleessa kerrotaan yleistä taustaa IEEE 802.15.4 standardin syntymiselle, selitetään lyhyesti protokollapinon erityispiirteet ja ominaisuudet verrattuna muihin langattomiin protokollaratkaisuihin. Myös fyysisen kerroksen ratkaisuja käydään läpi siinä määrin kuin työn ymmärtämisen helpottumiselle on nähty tarpeelliseksi. Koska työn varsinainen aihe liittyy erilaisiin MAC-kerroksen ratkaisuihin, on fyysisen kerroksen tarkastelu jätetty tarkoituksella hyvin pinnalliseksi. Tarkemmin tämän standardin MAC-kerroksen toiminnallinen määrittely käydään läpi luvussa 5.

Langattomien WPAN-verkkojen käsite on ollut olemassa jo useiden vuosien ajan, mutta varsinaiseen vauhtiin kyseisen käsitteen määrittely pääsi IEEE 802.15 työryhmän perustamisen myötä. Kyseisen työryhmän kehittämät standardit ovat suunnattuja lähinnä lyhyen kantaman tiedonsiirtoon. Työryhmä on organisoitu kolmeen eri aliryhmään. Alun perin ryhmän ensimmäisenä päätavoitteena oli 802.15.1 standardin kehittäminen *Bluetooth*-teknologian pohjalta. Seuraavana vuorossa oli korkeiden tiedonsiirtonopeuksien WPAN-protokolla 802.15.3, joka kehitettiin multimedia laitteiden väliseen tiedonsiirtoon. [11]

Kolmantena vuorossa ollut IEEE:n 802.15 standardin Task Group 4, perustettiin vuonna 2000 aloittamaan langattoman tiedonsiirtoratkaisun kehittäminen, joka mahdollistaa matalan tiedonsiirtonopeuden, alhaisen kompleksisuuden ja pitkän käyttöiän omaavan tiedonsiirron toteuttamisen. Tämä standardi on siis nimeltään IEEE 802.15.4. Työn sisältönä oli kehittää sekä fyysisen, että MAC-protokollakerroksen spesifikaatiot. Työssä on käytetty standardin versiota 802.15.4-2006. [12],[14]

IEEE 802.15.4 protokolla sallii korkean viiveen ja alhaisen tiedonsiirtonopeuden laitteiden välillä, korostaen näiden myönnytysten avulla virransäästön ja alhaisen kompleksisuuden merkitystä. Näillä ominaisuuksilla protokolla sopiikin mainiosti langattomien anturiverkkojen konseptiin.



Kuva 8. WPAN ja WLAN standardien tiedonsiirtonopeus / monimutkaisuusaste [11]

Kuvassa 8 on pyritty havainnollistamaan eri standardien sijoittumista toisiinsa nähden, kun kyseessä on tiedonsiirtonopeus ja virrankulutus [11]. Kuvasta puuttuu tuore WiMAX-ratkaisu, joka IEEE-standardina tunnetaan nimellä 802.16. Kyseistä tekniikkaa hyväksikäyttämällä päästään jopa 75MB/s tiedonsiirtonopeuksiin. WiMAX-teknologiaa käytetään lähinnä haja-asutusalueilla korvaamaan langallisia laajakaistayhteyksiä. [15]

Millä keinoilla alhainen virrankulutus sitten saavutetaan? Yksi tärkeimmistä keinoista on minimoida laitteen aktiivisen tilan osuus. Käsitteellä *duty cycle* tarkoitetaan laitteen aktiivisen tilan suhdetta tarkkailuajanjaksoon. IEEE 802.15.4 -standardi on pyritty suunnittelemaan siten, että *duty cycle* olisi mahdollisimman alhainen; laitteen lähetin- ja vastaanotinyksikkö saattavat olla inaktiivisessa tilassa jopa yli 99 prosenttia ajasta. On huomioitavaa, että vaikka laitteen lähetin- ja vastaanotinyksikkö onkin lepotilassa, kuluttaa laite silti virtaa. Myös niin sanotussa *standby* -tilassa laite kuluttaa jonkin verran virtaa ajastimien yms. vuoksi. Edellä mainittu virrankulutus on kuitenkin hyvin nimellinen verrattuna aktiivisen tilan virrankulutukseen. [11]

Myös radiotielle asetettavat vaatimukset on suunniteltu tehonkulutusta ja kustannuksia silmälläpitäen. Lähetysteho ja radion vastaanoton herkkyysvaatimukset ovat helposti saavutettavissa edullisilla komponenteilla. Myös modulaatiotekniikka on valittu siten, että se olisi mahdollisimman yksinkertainen toteuttaa, tehokas ja näin ollen virrankulutuksen sekä kustannusten suhteen perusteltu ratkaisu. Standardissa käytettävät modulaatiotekniikat

ovat Binary Phase Shift Keying (BPSK) 868/915 MHz taajuusalueelle, ja Offset Quadrature Phase Shift Keying (O-QPSK) 2.4GHz taajuusalueelle. [11]

Standardi määrittelee siis kaksi fyysisen kerroksen versiota: ensimmäinen taajuusalueille 868/915MHz ja jälkimmäinen 2,4GHz taajuusalueelle. Molemmat taajuusalueet ovat ns. vapaita taajuuksia, joita voidaan vapaasti käyttää ilman hallinnointi- tms. maksuja. 868MHz alueen taajuudet ovat vapaasti käytettävissä Euroopan alueella, 915MHz taajuuksista osa Pohjois-Amerikassa, Australiassa, Uudessa-Seelannissa sekä osissa Etelä-Amerikkaa. 2,4GHz taajuusalue on käytännöllisesti katsoen käytettävissä ympäri maailman.

Taulukossa 1 ovat lueteltuina standardin mahdollistamat tiedonsiirtonopeudet eri taajuusalueilla, sekä myös käytössä olevien kanavien lukumäärä.

Taajuusalue	Siirtonopeus	Kanavien lukumäärä
868.0 - 868.6 MHz	20kb/s	1
902 - 928 MHz	40kb/s	10
2.40 - 2.48GHz	250kb/s	16

Taulukko 1. IEEE 802.15.4 standardin tiedonsiirtonopeudet sekä kanavalukumäärät

Standardin käytössä olevat 27 eri kanavaa on numeroitu 0-26, alkaen alimmasta taajuusalueesta; 868MHz alueen ainut kanava on järjestysnumeroltaan 0, kun taas 915MHz alueen kanavat ovat 1-10, ja 2.4GHz alueen kanavat numeroiltaan 11-26. [11]

4 Anturiverkkojen MAC-protokollat

Tässä luvussa käydään läpi erilaisia anturiverkoille suunniteltuja Medium Access Control -ratkaisuja. Protokollat on esitetty kronologisessa järjestyksessä siten, että järjestys tukee eri ominaisuuksien ymmärtämistä. Järjestyksessä jälkimmäisenä esitetty protokollaratkaisu sisältää piirteitä aiemmista ratkaisuista. Vaikka esimerkiksi S-MAC ei energiankulutuksen suhteen pystykään kilpailemaan uudempien protokollaratkaisujen kanssa, on se kuitenkin yleisesti käytetty vertailukohde MAC-protokollien esityksissä. Kirjoittamishetkellä on jo saatavilla hyvin monia erilaisia ehdotuksia anturiverkon MAC-protokollaksi. Kaikkia näitä ei voitu työn liiallisen laajenemisen vuoksi ottaa mukaan.

Joukkoon on pyritty valitsemaan sellaisia ratkaisuja, joissa käytetään hyvin erilaisia keinoja energiansäästön saavuttamiseen. AI-LMAC poikkeaa muista sikäli, että se on suunniteltu tietyntyyppisen anturiverkkosovelluksen käyttöön. Ratkaisu on sisällytetty työhön juuri siitä syystä, että se on sovelluskohtainen. Sovelluskohtaisen optimoinnin mahdollistama virransäästö ei ole kovin laajasti tutkittu alue, mutta kuitenkin hyvin mielenkiintoinen aihe.

IEEE 802.15.4 -standardin MAC-protokollaa käsitellään muista poiketen omassa luvussaan 5. Syynä tähän on se, että kyseessä on joukon protokollaratkaisuista ainut, joka on standardoitu. Muut ovat tutkimustason MAC-ratkaisuja, toisin sanoen tutkijaryhmien ehdotuksia siitä minkälainen tulisi hyvän anturiverkoille soveltuvan MAC-ratkaisun olla.

Minkä tahansa jaetun siirtotien tiedonsiirtoverkossa MAC-protokollan toiminta on erittäin tärkeä seikka. Anturiverkoille soveltuvaa MAC-protokollaa suunniteltaessa joudutaan tekemään kompromisseja useiden eri ominaisuuksien suhteen. Hyvin harvoin jotakin osa-aluetta voidaan parantaa ilman, että sillä olisi mitään vaikutuksia muihin ominaisuuksiin. Anturiverkkojen MAC-protokollaratkaisuissa on muihin MAC-ratkaisuihin verrattuna omat erityisrajoitteensa, johtuen rajoitetusta energiansaannista ja muista seikoista, joita on jo aiemmissa kappaleissa käsitelty. Seuraavaksi käydään läpi yleisimmät käsitteet anturiverkkojen MAC-protokollakerroksen ominaisuuksiin liittyen. [38]

1. Törmäysten välttäminen (*engl. Collision Avoidance)* on jokaisen MAC-ratkaisun perustehtäviä. Kilpailuperustaisissa MAC-ratkaisuissa hyväksytään tietty määrä törmäyksiä, vaikka kaikki MAC-protokollat pyrkivät välttämään tiheään tapahtuvat törmäykset

2. Energiansäästö (*engl. Energy Efficiency)* on hyvin tärkeä tutkimuskohde anturiverkkojen alueella. Tässä työssä keskitytään tutkimaan erityisesti MAC-protokollaratkaisujen energiansäästöominaisuuksia.

3. Skaalautuvuus ja adaptiivisus (*engl. Scalability and Adaptivity)* liittyy läheisesti verkon kokoon, nooditiheyteen sekä topologiaan. Anturiverkoissa noodit voivat poistua verkosta virtalähteen ehtymisen vuoksi, ja myös uusia noodeja voi liittyä verkkoon kesken sen elinkaaren. Noodien paikka voi myös vaihdella verkossa sen eliniän aikana. Hyvä MAC-ratkaisu kykenee sopeutumaan kaikkiin näihin muutostilanteisiin.

4. Kanavan käyttöaste (*engl. Channel Utilization)* kuvaa sitä, kuinka suuri osa käytettävästä tiedonsiirtokapasiteetista on käytössä. Muita usein käytettyjä termejä ovat myös kaistanleveyden käyttöaste (*engl. Bandwidth Utilization)* ja kanavakapasiteetti (*engl. Channel Capacity)*. Esimerkiksi matkapuhelinverkoissa kanavan käyttöaste pyritään maksimoimaan, sillä se on yleensä kallisarvoisin resurssi. Anturiverkoissa kanavan käyttöasteen maksimointi on yleensä hyvin toissijainen tavoite.

5. Viive tai latenssi (*engl. Latency)* kertoo kuluneen ajan siitä hetkestä, kun lähettäjällä on valmis paketti siirrettävänä siihen saakka, kun vastaanottava noodi on sen onnistuneesti vastaanottanut. Yleensä anturiverkkosovelluksissa latenssin minimointi ei ole ensisijainen tavoite. Esimerkiksi valvonta- tai tarkkailusovelluksissa latenssilla ei ole suurta vaikutusta, koska latenssiaika on murto-osa fyysisen objektin liikenopeudesta.

6. Välityskyvyllä (*engl. Throughput)* tarkoitetaan siirretyn datan määrää aikayksikköä kohti. Kyseiseen suureeseen vaikuttavat monet muut tekijät, esimerkiksi törmäysten

välttämisen tehokkuus, kanavan käyttöaste, latenssi ja hyötykuorman osuus siirrettävästä datasta. Yleensä myös MAC-protokollat, jotka pyrkivät maksimoimaan verkon eliniän energiankulutuksen minimoinnin avulla, hyväksyvät pienemmän siirtonopeuden ja kasvaneen latenssiajan.

7. Tasapuolisuus (*engl. Fairness)* kuvastaa sitä, kuinka hyvin eri noodit pääsevät käyttämään siirtotietä silloin, kun niillä on dataa lähetettävänään. Tämä on hyvin tärkeä ominaisuus perinteisissä verkoissa, mutta anturinoodisovelluksissa yleensä myös hyvin toissijainen seikka. Tämän johtuu siitä, että noodit suorittavat yhteistä tehtävää, ja tästä johtuen yksittäisen noodin tuottamalla informaatiolla ei yleensä ole suurta merkitystä koko sovelluksen kannalta. Noodit ovat tyypillisesti suurimman osan ajasta ei-aktiivisessa tilassa, ja toisaalta tiettynä ajanhetkenä yksittäisellä noodilla saattaa olla hyvin paljon dataa lähetettävänä. Tällaisessa tapauksessa on tärkeämpää antaa kyseisen noodin lähettää tietonsa kuin antaa kyseisellä ajanhetkellä kaikille noodeille tasapuolinen pääsy siirtotielle.

Tässä työssä läpikäytävät MAC-protokollat voidaan jakaa kahteen eri luokkaan: aikajakoperusteisiin TDMA (*engl. Time Division and Multiple Access)* ja niin sanottuihin kilpailuperusteisiin (*engl. Contention Based*) protokolliin.

TDMA-protokollissa jokaiselle noodille taataan oma aikaväli, jolla se voi lähettää tai vastaanottaa viestejä. TDMA-protokollien haittapuolena on niiden vaatima tarkka synkronointi. Noodien täytyy kyetä ylläpitämään hyvin tarkasti samaa aikataulua, jotta aikajakomenetelmää voidaan käyttää.

Kilpailuperusteiset protokollat puolestaan eivät aseta kovin merkittäviä tarkkuusvaatimuksia synkronoinnin suhteen. Tässä protokollatyypissä noodit joutuvat kilpailemaan pääsystä siirtotielle. Tähän käytetään CSMA-algoritmia *(Carrier Sense with Multiple Access)*, joka on selostettu tarkemmin kappaleessa 5.3.1.

Kuten aiemmin mainittiin, anturiverkkojen protokollaratkaisuissa pyritään minimoimaan energiankulutus muiden parametrien kustannuksella. Tiedonsiirtonopeus, tasavertaisuus

noodien välillä sekä sanomavälityksen viive ovat yleensä toissijaisia asioita anturiverkoille soveltuvissa protokollaratkaisuissa.

Aiemmin mainittujen törmäysten lisäksi alueita, joissa energiaa kuluu hukkaan ja joissa energiankäyttö pyritään erilaisissa MAC-ratkaisuissa minimoimaan, ovat ylikuuleminen (*engl. overhearing)*, kontrollipaketeista aiheutuva kuormitus (*engl. control packet overhead)* ja turha kuuntelu (*engl. idle listening)*. [38]

Törmäyksellä tarkoitetaan tilannetta, jossa kaksi laitetta lähettää yhtäaikaisesti jaetulla medialla. Tällöin paketit korruptoituvat ja lähetys joudutaan uusimaan. Jokainen uudelleenlähetetty sanoma voidaan nähdä turhana energiankulutuksena, ja näiden määrä pyritään eri tekniikoita käyttämällä saamaan mahdollisimman pieneksi. [38]

Ylikuuleminen puolestaan tarkoittaa tilannetta, jossa noodi vastaanottaa viestin, jota ei ole tarkoitettu sille. Noodi on tällöin aktiivisessa tilassa, ja aivan kuten lähetyskin, myös sanoman vastaanotto kuluttaa energiaa. [38]

Kontrollipakettien aiheuttamasta kuormituksesta energian kulutuksen suhteen tarkoitetaan sitä, että mitä enemmän kontrollisanomia protokollassa käytetään, sitä vähemmän suhteessa verkossa kulkee varsinaisia tietoa sisältäviä hyötysanomia. Kontrollipaketeista ei kokonaan voi protokollissa päästä eroon, mutta näiden määrää voidaan pienentää laskemalla vaatimustasoa muiden vaatimusten suhteen. [38]

Turha kuuntelu tarkoittaa aikajaksoa, jolloin noodi aktiivisesti kuuntelee siirtotietä, mutta ei vastaanota mitään sanomia. Turha kuuntelu on kaikista merkittävin näistä neljästä edellä mainitusta energiaa tuhlaavasta alueesta. [38]

Seuraavissa alaluvuissa käydään läpi kirjoitushetkellä eräitä uusimpia anturiverkoille soveltuvia MAC-protokollaratkaisuja. Monet näistä ratkaisuista pohjautuvat aiemmalle tutkimustyölle, joista voidaan ainakin mainita MAC-ratkaisut kuten PAMAS, Aloha sekä LEACH. Nämä on päätetty jättää tämän työn ulkopuolelle, koska ne eivät kykene kilpailemaan energiankulutuksen osalta nykyisten ratkaisujen kanssa.

4.1 S-MAC

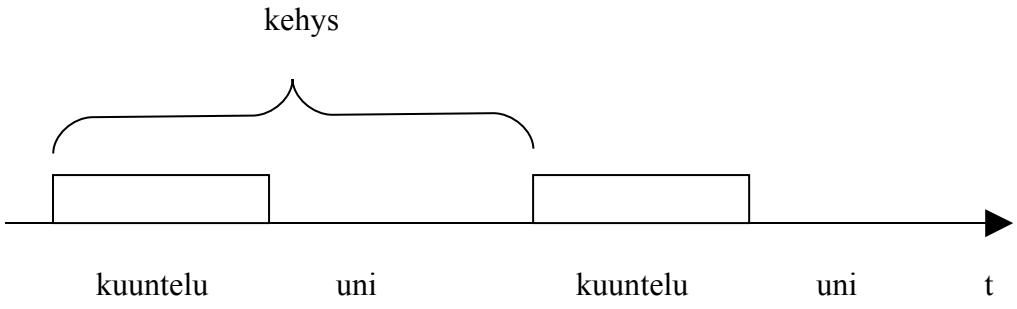
Sensor-MAC, eli S-MAC protokolla pyrkii vähentämään energiankäyttöä jokaisella neljällä, kappaleessa 4, mainitulla osa-alueella. Turhaa kuuntelua S-MAC vähentää asettamalla noodit toistuvin väliajoin unitilaan (*periodic sleep)*. Tämä pienentää niin sanottua *duty cycle* -arvoa, joka kertoo kuinka suuren osan ajasta noodi on aktiivisessa tilassa. Unitilassa noodin radio on siis kokonaan kytkettynä pois toiminnasta. [39]

Periodinen uni kuitenkin aiheuttaa viestinvälitykseen viivettä, jota pyritään lieventämään käyttämällä adaptiivista kuuntelua (*engl. adaptive listening)*. Toinen tekniikka, joka liittyy tehokkaaseen pitkien viestien välittämiseen verkossa, on viestinvälitys (*engl. message passing)*. [39]

Seuraavissa kappaleissa käydään läpi S-MAC-protokollaratkaisun erityispiirteet, joilla pyritään vähentämään energiankulutusta sekä kompensoimaan energiankulutuksen vähentämiseen tähtäävien tekniikoiden aiheuttamia haittavaikutuksia.

4.1.1 Periodinen kuuntelu- ja unitila

Monissa anturiverkkosovelluksissa noodit saattavat olla *idle*-tilassa suuren osan ajasta. S-MAC-ratkaisussa pyritään säästämään energiaa asettamalla noodit periodiseen unitilaan. Noodin tila vaihtelee siis valveillaolon, eli kuuntelun tai lähettämisen, sekä nukkumisen välillä. Unitilassa noodin radio-osa kytketään pois toiminnasta. Noodi asettaa ajastimen, jonka lauettua noodi siirtyy unitilasta kuunteluun. Mikäli kuuntelujakson aikana noodi ei vastaanota sille osoitettuja paketteja, siirtyy se jälleen takaisin unitilaan. Jaksoa, johon kuuluu kokonainen uni- ja kuuntelutila kutsutaan kehykseksi *(engl. frame)*. [39]



Kuva 9. Periodinen unikehys

Kuvassa 9. on havainnollistettu periodisen unikehyksen rakennetta. Kuten aiemmin mainittiin, termi *duty cycle* tarkoittaa kuuntelujakson ja koko kehyksen suhdetta. Tätä suhdetta voidaan muuttaa käytettävän sovelluksen vaatimusten mukaiseksi. Jokainen noodi voi myös tehdä päätöksen oman kuuntelu- / unijaksonsa kestosta. Toivottavaa tosin on, että vierekkäiset noodit käyttäisivät samaa jaksoa. Koska anturiverkot saattavat koostua jopa tuhansista noodeista, ts. muodostavat *multihop*-verkon, eivät kaikki noodit voi kuitenkaan käyttää samaa jaksotusta. [39]

Noodit vaihtavat aikataulutietonsa lähettämällä periodisesti SYNC-paketin välittömille naapureilleen. Jaksoa, jonka aikana noodi lähettää SYNC-pakettinsa kutsutaan synkronointijaksoksi *(engl. synchronization period)*.

Tästä seuraa eräs S-MACin ominaisuus, jota kutsutaan virtuaaliseksi klusteroinniksi. Noodit muodostavat verkossa matalan, *peer-to-peer*-topologian. Verkossa ei ole erikseen vaadittavaa kontrollointia ja koordinointia, vaan virtuaaliset klusterit muodostuvat noodien käyttämien aikataulujen perusteella. Kommunikointi pysyy myös *peer-to-peer*-tyyppisenä. Tällainen ratkaisu mahdollistaa hyvin joustavan verkkotopologian ja uusien noodien mukaantulon verkkoon. [39]

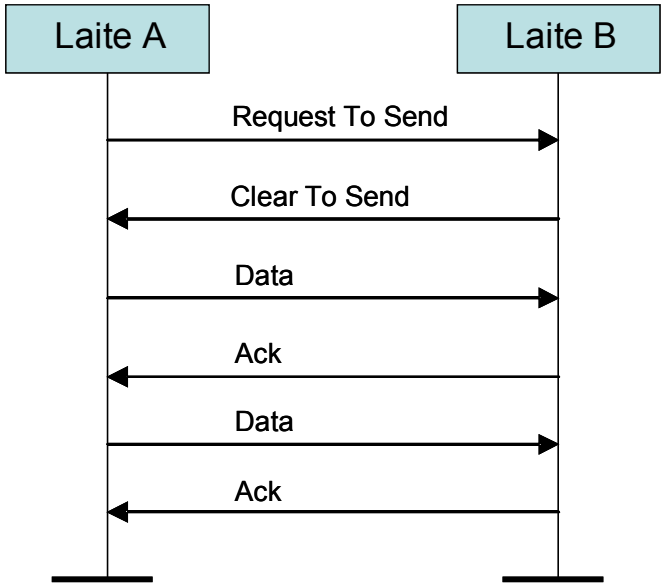
Periodisen unitilan käyttö on energiatehokasta, mutta lisää kuitenkin viivettä verkon kommunikoinnissa. Kappaleessa 4.1.3 käydään läpi koordinoitu unitilan käyttö - tekniikka, jolla periodisen unitilan käytöstä johtuvaa viivettä voidaan pienentää.

4.1.2 Törmäysten välttäminen

Mikäli useat noodit haluavat samanaikaisesti lähettää viestin tietylle noodille, ne yrittävät suorittaa lähetyksensä, kun vastaanottava noodi aloittaa kuuntelujaksonsa. Edellä mainittu tilanne johtaa törmäykseen, joiden välttämisessä S-MAC käyttää kahta eri tekniikkaa: fyysistä ja virtuaalista kantoaallon tunnistusta *(engl. carrier sense)*, sekä RTS/CTS-sanomien vaihtoa. [39]

Virtuaalisessa kantoaallon tunnistuksessa käytetään jokaisessa sanomassa mukana olevaa kestokenttää *(engl. duration field)*. Mikäli noodi vastaanottaa sanoman, joka on osoitettu jollekin toiselle noodille, se lukee sanomasta kestokentän arvon. Noodi tallettaa kestotiedon NAV:iin kokonaislukuna *(Network Allocation Vector)* ja asettaa erillisen NAV-ajastimen. Joka kerta kun NAV-ajastin laukeaa, noodi vähentää NAV-laskurin arvoa yhdellä. Mikäli noodi aikoo aloittaa lähetyksen, se tutkii NAV:in arvon ja mikäli se on nolla, voi lähetys alkaa. Muussa tapauksessa noodi voi päätellä, että media ei ole vapaa lähetystä varten. Fyysinen kantoaallon tunnistus tapahtuu nimensä mukaisesti fyysisellä kerroksella. Mikäli molemmat kantoaallon tunnistustyypit indikoivat kantoaallon olevan vapaa, voi lähetys alkaa. [39]

Mikäli noodi huomaa, että media on vapaa lähetystä varten, suoritetaan sanomien vaihto käyttämällä RTS/CTS/DATA/ACK sekvenssiä lähettäjän ja vastaanottajan välillä. Kuvassa 10 on havainnollistettu sekvenssin kulku.



Kuva 10 RTS/CTS/DATA/ACK-kättelysekvenssi

Kuvassa 10 noodi A haluaa lähettää paketin noodille B. Noodi A lähettää ensin RTS-paketin *(Request To Send)*, jonka kaikki noodin A välittömät naapurit kuulevat (olettaen että törmäystä ei tapahtunut). Noodin A naapurit tietävät nyt, että ne eivät voi lähettää sanomia, koska noodi A on varannut median käyttöönsä. Vastaanotettuaan RTS-paketin (olettaen jälleen että törmäystä ei tapahtunut), noodi B vastaa lähettämällä CTS-paketin *(Clear To Send)* noodille A. Noodin B naapurit kuulevat CTS-paketin lähetyksen, joten myös ne tietävät että ne eivät voi lähettää sanomia. Koska noodin A naapurit on hiljennetty aiemmin, ei törmäystä tapahdu ja noodi A vastaanottaa CTS-paketin. Tämän jälkeen noodi A lähettää DATA-paketin, jonka noodi B vastaanottaa. Törmäyksiä ei tapahdu tässäkään vaiheessa, koska noodin B naapurit on aiemmin hiljennetty. Vastaanotettuaan onnistuneesti DATA-paketin, lähettää noodi B ACK-kuittauspaketin noodille A. Vastaanotettuaan kyseisen paketin noodi A tietää, että DATA-paketti on vastaanotettu onnistuneesti ja noodi A voi lähettää seuraavan DATA-paketin. [18]

Onnistuneiden RTS/CTS -pakettien vaihdon jälkeen noodit käyttävät unijaksonsa datan lähetykseen normaalin energiansäästön sijaan. Noodit eivät noudata normaaleja unijaksojaan ennen kuin lähetys on suoritettu. Poikkeuksena RTS/CTS -sanomavaihdolle on *broadcast*-pakettien lähetys. Tämän tyyppiset paketit on tarkoitettu kaikkien verkon

noodien kuultavaksi, ja *broadcast*-paketit voidaan lähettää ilman RTS/CTS -sanomavaihtoa. [18][39]

4.1.3 Koordinoitu uni

Aiemmin läpikäyty periodinen unitila vähentää tuntuvasti turhaan kuuntelua. Näiden unijaksojen täytyy kuitenkin olla koordinoituja, eivätkä yksittäiset verkon noodit voi valita omia unijaksojaan riippumatta muista verkon noodeista. Tässä kappaleessa käydään läpi S-MACin toteuttama koordinoitu uni. Tämä käsittää sekä unijakson valinnan että ylläpidon.

Ennen kuin noodi voi ruveta käyttämään uni- ja kuuntelujaksoja, sen täytyy määritellä itselleen aikataulu kyseisiä jaksoja varten sekä vaihtaa aikataulutietoja naapureidensa kanssa. Jokainen noodi ylläpitää tunnettujen naapurien aikataulutiedoista omaa taulukkoaan *(engl. schedule table)*. Oman aikataulun määritys ja käyttöönotto tapahtuu seuraavasti: [39]

1. Noodi kuuntelee ennalta määritellyn keston ajan muilta noodeilta tulevia aikataulupaketteja. Tämä kesto on vähintään synkronointijakson mittainen. Mikäli noodi ei määritellyn ajan puitteissa vastaanota aikataulutietoja muilta noodeilta, se määrittelee oman aikataulunsa ja alkaa noudattaa sitä. Samaan aikaan noodi lähettää *broadcast*-tyyppistä SYNC-pakettia, jotta se voisi kertoa naapurinoodeille oman aikataulunsa.

2. Mikäli noodi vastaanottaa aikataulun toiselta noodilta ennen kuin on määritellyt itselleen aikataulun, se ottaa saman aikataulun käyttöön. Noodi pyrkii ilmoittamaan naapureilleen käyttöönottamansa aikataulun seuraavan kuuntelujakson aikana.

3. Jos noodi vastaanottaa erilaisen aikataulun sen jälkeen, kun se on valinnut omansa ja alkanut noudattamaan sitä, on kaksi eri toimintavaihtoehtoa. Jos noodilla ei ole muita naapureita, se hylkää aiemmin käyttöönottamansa aikataulun ja alkaa noudattaa uutta vastaanottamaansa aikataulua. Mikäli noodi noudattaa jo samaa aikataulua yhden tai useamman naapurin kanssa, se ottaa molemmat aikataulut

käyttöön. Tämä tarkoittaa unikehyksen kannalta sitä, että noodi siirtyy unitilasta kuuntelutilaan molempien aikataulujen kuunteluintervallien mukaan.

Hajautetussa verkossa, jossa jokainen noodi kuulee toisensa, noudattavat kaikki noodit samaa aikataulua. Mikä tahansa noodi ehtiikin ensimmäisenä lähettämään SYNC-pakettinsa, tulevat kaikki muut noodit ottamaan kyseisen aikataulun käyttöönsä.

Multihop-verkossa kaksi noodia voi asettaa muille eri aikatauluja toisistaan riippumatta. Tämä tapahtuu, mikäli nämä kaksi kyseistä noodia ovat toistensa kantoalueen ulkopuolella. Haittapuolena tässä on se, että mikäli jotkin noodit joutuvat seuraamaan useampaa kuin yhtä aikataulua, tulee myös näiden noodien virrankulutus olemaan huomattavasti suurempi kuin yhtä aikataulua noudattavien.

Toinen vaihtoehto on antaa rajanoodien ottaa käyttöön vain toinen aikatauluista. Jos noodi noudattaa ensin kuulemaansa aikataulua ja vastaanottaa myöhemmin toisen, se voi kuitenkin halutessaan kommunikoida myös toista aikataulua noudattavien noodien kanssa. *Broadcast*-viestejä käsiteltäessä täytyy raja noodin kuitenkin lähettää kahden aikataulun mukaan.

Mikäli verkkoon liittyy uusi noodi, voi se epäonnistua synkronoinnissa muuhun verkkoon nähden. Syynä tähän on se, että naapuri noodin lähettämä SYNC-paketti saattaa korruptoitua joko törmäysten tai lähetyshäiriöiden (*engl. interference)* vuoksi. Naapuri on myös saattanut siirtää SYNC-pakettinsa lähettämistä sen vuoksi, että se ei ole voittanut kilpailua siirtotielle pääsystä, ts. media on kyseisellä ajanhetkellä varattu muiden noodien toimesta. Mikäli kyseinen uusi noodi on niin sanottu rajanoodi kahta eri aikataulua seuraavien noodiryhmien välissä, saattaa se kuulla ainoastaan yhden aikataulun kahden sijasta. Tämä tapahtuu silloin, mikäli aikataulut eivät ole lainkaan päällekkäisiä (*engl. overlapping)*. Jotta kyseinen tilanne voitaisiin välttää, on S-MAC-protokollassa käytössä periodinen naapurien etsintä (*engl. periodic neighbour discovery)*. Periodisen naapurien etsinnän tapahtuessa noodi kuuntelee kokonaisen synkronointijakson ajan muita lähetyksiä. Se kuinka usein tämä suoritetaan, riippuu noodin naapurien lukumäärästä. Mitä enemmän sillä on naapureita, sitä harvemmin periodinen naapurien etsintä suoritetaan. Mikäli noodilla taas ei ole yhtään tunnettua

naapuria, on todennäköisempää että se ei ole löytänyt niitä, ja näin ollen kyseinen toimenpide suoritetaan useammin. [39]

4.1.4 Synkronoinnin ylläpito

Koska noodit ylläpitävät aikataulutietojaan naapureiden kesken, voi sisäisen kellon ajastusvirhe *(engl. clock drift)* aiheuttaa synkronointiin virhettä. S-MAC käyttää kahta eri menetelmää synkronointivirheen minimointiin. Ensimmäinen synkronointivirheitä minimoiva ominaisuus S-MAC-ratkaisussa on kuunteluperiodin ja kellovirheen suhde. Kuunteluperiodi 0,5s on yli 10^4 kertaa pidempi kuin tyypillisesti havaittavat kellovirheet.

Vaikka kuuntelujakson ja tyypillisen kellovirheen suhde on hyvin pieni, on olemassa menetelmä, jolla minimoidaan pitkäaikaisen kellovirheen vaikutus (*engl. long-term clock drift)*. S-MACissa ongelma on ratkaistu sen avulla, että sanomissa käytettävät aikaleimat ovat suhteellisia. Kuten aiemmin on mainittu, aikataulutietoja ylläpidetään lähettämällä SYNC-paketteja. Kyseinen paketti sisältää lähettäjän osoitteen, sekä lähettäjän seuraavan unijakson ajankohdan. Aikaleiman suhteellisuudella tarkoitetaan siis sitä, että noodin seuraava unijakso riippuu siitä ajanhetkestä jolloin lähettävä noodi on aloittanut SYNC-paketin lähettämisen. [39]

Jotta sekä SYNC- että DATA-pakettien vastaanottaminen on mahdollista, on noodin kuunteluperiodi jaettu kahteen osaan em. paketteja varten. Molemmilla osilla on oma kilpailuikkunansa *(engl. contention window)*, jonka aikana lähettäjä suorittaa kantoaallon tunnistuksen. Lähettäjä valitsee satunnaisesti aikavälin kilpailuikkunasta, suorittaa kantoaallon tunnistuksen, ja mikäli media on vapaa lähettää se SYNC-paketin. Sama pätee myös DATA-paketeille. [39]

4.1.5 Adaptiivinen kuuntelu

Aiemmin mainittiin, että turha kuuntelu on merkittävä energianhukkaaja anturiverkoissa. Tätä ongelmaa minimoidaan periodisella unella, mutta tämän menetelmän haittapuolena todettiin olevan viiveen aiheuttaminen sovellukselle. Viive voi myös kertautua jokaisen noodin kohdalla, mikäli ne tiukasti noudattavat omia aikataulujaan.

Adaptiivinen kuuntelu on tekniikka, jolla noodi voidaan asettaa low-duty-cycle -toiminnasta aktiivisempaan tilaan pyrittäessä vähentämään periodisen unen aiheuttamaa viivettä. Adaptiivisen kuuntelun idea on lyhyesti siinä, että herätetään noodi silloin, kun se ylikuulee naapurinsa vastaanottaman paketin. Tämä tulisi ajoittaa lähetyksen loppuun, ja optimitapauksessa joko RTS- tai CTS-pakettiin. Mikäli herätetty noodi sattuu olemaan seuraava kohde reitityksessä, lähettäjän ei tarvitse odottaa, että vastaanottava noodi päätyy aikataulunsa mukaiseen kuuntelujaksoon. Mikäli herännyt noodi taas ei ole reitityksen seuraava kohde, ts. ei vastaanota pakettia, se siirtyy jälleen noudattamaan omaa aikatauluaan.

Myöskään kaikki seuraavan reitityshypyn noodit eivät välttämättä ylikuule naapurinsa vastaanottamia paketteja. Adaptiivisen kuuntelun mukaan lähetettyyn RTS-pakettiin ei siis välttämättä saada CTS-vastausta. Tässä tapauksessa lähettävä noodi siirtyy noudattamaan jälleen normaalia aikatauluaan, ts. siirtyy unitilaan. [39]

4.1.6 Ylikuulemisen välttäminen

Kuten aiemmin todettiin, ylikuuleminen on yksi lähde turhassa energiankulutuksessa. S-MAC pyrkii minimoimaan ylikuulemista sallimalla noodien siirtymisen unitilaan välittömästi, kun ne ovat ylikuulleet RTS- tai CTS-paketin. RTS- ja CTS-paketit ovat lyhyitä verrattuna varsinaisiin datapaketteihin, joten näiden ylikuuleminen ei aiheuta niin suurta energianhukkaamista kuin jos ylikuunneltaisiin DATA- ja sitä seuraavia ACK-paketteja.

Huomattavaa on, että joskus ylikuuleminen on haluttava ominaisuus. Jotkin algoritmit voivat esimerkiksi kerätä tietoa verkosta ylikuulemisen perusteella. S-MAC voidaan tällaisessa tapauksessa konfiguroida siten, että sovelluskohtainen ylikuuleminen on mahdollista.

4.1.7 Sanoman välitys

Sanomalla tarkoitetaan kokonaisuutta, joka voi koostua useista datapaketeista. Yleensä vastaanottajan täytyy saada kaikki datapaketit, jotta se voi suorittaa prosessointia tähän

dataan liittyen. *Message passing* -tekniikalla pyritään tehokkaaseen viestinvälitykseen sekä viiveen että energiankulutuksen suhteen. Yksi vaihtoehto on lähettää koko sanoma yhdellä kertaa, jolloin viive olisi hyvin pieni. Tämä vaihtoehto ei kuitenkaan ole energiankulutuksen kannalta paras mahdollinen. Mikäli sanoma täytyy lähettää uudelleen, nousee energiankulutus korkeaksi. Sanoman pilkkominen useaan osaan parantaa energiatehokkuutta uudelleenlähetyksen suhteen, mutta lisää *control overhead* -ilmiötä useiden RTS/CTS -sanomien vuoksi. Myös viive lisääntyy verrattuna yhdellä kertaa lähetettävään sanomaan.

S-MAC-ratkaisussa on valittu menettelytapa, jossa sanoma pilkotaan useaksi pienemmäksi osaksi, fragmentiksi. Nämä kehykset lähetetään purskeena *(engl. burst)* käyttäen vain yhtä RTS/CTS -pakettiparia. Vastaanottaja lähettää jokaista vastaanotettua datafragmenttia vasten ACK-paketin lähettäjälle. Mikäli lähettäjä ei kuule ACK-pakettia välittömästi lähetetyn fragmentin jälkeen, suoritetaan kyseisen fragmentin uudelleenlähetys. Kuten aiemmin työssä on mainittu, jokaisessa paketissa on kestokenttä, joka kertoo vaaditun ajan jäljellä olevien datafragmenttien ja ACK-pakettien lähetykseen. Mikäli uusi noodi liittyy verkkoon kesken sanoman lähetyksen, se pystyy siirtymään unitilaan loppulähetyksen ajaksi. Mikäli lähetystä joudutaan pitkittämään esimerkiksi hävinneiden pakettien tai törmäysten vuoksi, ei unitilassa oleva noodi saa tästä tietoa välittömästi. Kuuntelutilaan siirtynyt noodi pystyy kuitenkin sopeuttamaan aikataulunsa välittömästi kuultuaan uuden paketin.

ACK-paketin lähetyksellä jokaisen datafragmentin jälkeen pyritään pääsemään eroon piilotetun päätelaitteen ongelmasta *(engl. hidden terminal problem)*. Kyseinen tilanne esiintyy siinä tapauksessa, että naapurinoodi herää kesken lähetyssekvenssin tai uusi noodi liittyy verkkoon. Jos noodi on välitön naapuri ainoastaan vastaanottajalle, mutta ei lähettäjälle, se ei luonnollisesti pysty kuulemaan lähettävän noodin paketteja. Mikäli vastaanottava noodi ei lähettäisi ACK-paketteja jokaisen datafragmentin jälkeen, naapurinoodi päättelisi median olevan vapaa ja voisi omalla lähetyksellään korruptoida vastaanotettavat datapaketit.

Mikäli vastaanottavan noodin virtalähde tyhjenee lähetyksen jo alettua, ei se luonnollisestikaan voi enää vastata sanomiin. Tällöin ovat pakettien lähetysyritykset kyseiselle noodille lopetettava. Kyseistä tilannetta varten S-MAC-protokollassa on määritelty enimmäismäärä uudelleenlähetyksille. [39]

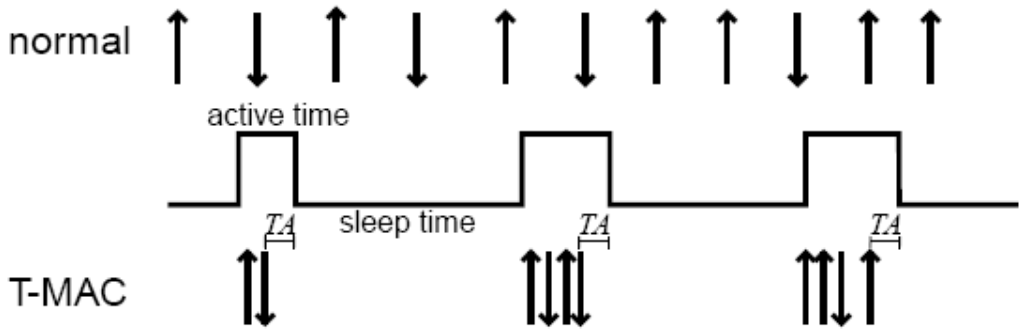
4.2 T-MAC

Time-out MAC eli T-MAC on kilpailuperusteinen MAC-protokolla, joka pyrkii minimoimaan turhaa kuuntelua käyttämällä adaptiivista *duty cycle* -menetelmää. Van Dam ja Langendoen [30] ovat päätelleet, että S-MAC:in käyttämämä kiinteä *duty cycle* ei tuota parasta mahdollista energiansäästöä. Tämä johtuu siitä, että sanomaliikenteen määrä verkossa vaihtelee suuresti, ja käyttämällä kiinteää *duty cycle* -arvoa turhan kuuntelun minimointia ei ole optimoitu parhaalla mahdollisella tavalla.

T-MAC:in perusidea on lähettää kaikki vaadittavat sanomat vaihtelevan mittaisissa purskeissa. Lähetysten välillä noodit asetetaan unitilaan - aivan kuten S-MAC-ratkaisussakin. Mekanismi, jolla optimaalisen mittainen aktiivinen aika päätellään vaihtelevan kuorman alla, on hyvin yksinkertainen. T-MAC käyttää *timeout*-menetelmää, mikäli noodi ei enää vastaanota sanomia. T-MAC käyttää myös S-MAC-ratkaisun mukaista synkronointi- ja klusterointimenetelmää. [30],[39]

4.2.1 Peruskuvaus protokollasta

Kuvassa 11 on esitetty tyypillinen tilanne T-MAC-protokollaa käytettäessä. Kuten S-MAC-ratkaisussakin, noodit nukkuvat periodisesti lähetysten välillä. Jokainen noodi herää vuorollaan kommunikoidakseen naapuriensa kanssa ja tämän jälkeen siirtyy unitilaan, joka kestää seuraavan kehyksen alkuun saakka.



Kuva 11. Tyypillinen tilanne adaptiivista T-MAC:iä käytettäessä [30]

Noodit käyttävät samaa RTS/CTS/DATA/ACK -mekanismia kommunikoinnissaan kuin esimerkiksi S-MAC.

Noodi vastaanottaa ja mahdollisesti lähettää sanomia niin kauan kuin se pysyy aktiivisessa tilassa. Aktiivinen periodi päättyy mikäli yhtään aktivaatiotapahtumaa (*engl. activation event)* ei ole tapahtunut ajan *TA*-kuluessa. Aktivaatiotapahtuma on joko

- periodisen kehysajastimen laukeaminen,
- datan vastaanottamistapahtuma radiotieltä,
- sanomavaihdon havainnointi radiotiellä, esimerkiksi törmäys,
- noodin oman datapaketin tai kuittaussanoman lähetyksen lopettaminen tai
- RTS/CTS-sanomaparin sisällöstä saatu tieto siitä, että naapurin datalähetys on loppunut.

Noodit siis siirtyvät unitilaan, mikäli eivät täytä aktiivisen tilan kriteerejä. Täten *TA* määrittää turhan kuuntelun keston minimin. Edellä kuvattu menetelmä siirtää kaiken viestiliikenteen kehyksen alkuun. Koska aktiivisen tilan ulkopuolella tapahtuva viestiliikenne täytyy puskuroida, määräytyy kehyksen maksimikesto viestipuskurin koon perusteella. [30]

4.2.2 RTS-operaatio ja TA-arvon valitseminen

T-MAC poikkeaa monista muista kilpailuperusteisistä protokollista kilpailuikkunansa suhteen. T-MAC käyttää kiinteää aikajaksoa kilpailuikkunan keston määrityksessä. Koska T-MACin mukaisesti noodit lähettävät pakettinsa kehyksen alussa, on siirtotie äärimmilleen varattuna. Noodit joutuvat ankaraan kilpailuun siirtotielle pääsystä. T-MAC ei kasvata kilpailuikkunan kokoa tästä huolimatta. On odotettavissa, että media on silti mitä todennäköisimmin varattu, vaikka ikkunan kestoa kasvatettaisiinkin. RTS-sanoman lähetys alkaa aina odottamalla ja kuuntelemalla satunnaisesti valitun ajanjakson, jonka kesto on sidottu kilpailuikkunan kokoon. Tämä puolestaan on pyritty optimoimaan maksimaalista viestinvälityskuormaa varten. Kilpailuaika on käytössä aina huolimatta siitä, onko vielä tapahtunut törmäystä vai ei.

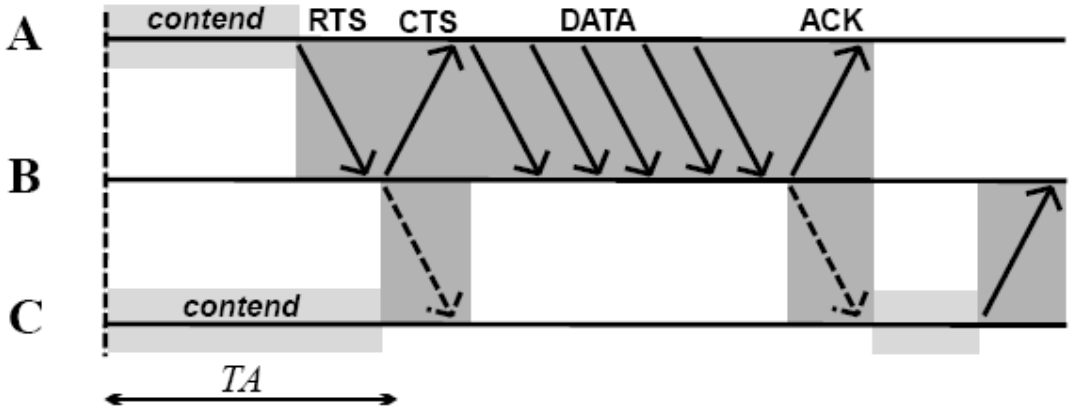
Mikäli noodi lähettää RTS-paketin, mutta ei saa vastauksena CTS-pakettia, yksi seuraavista asioista on tapahtunut:

1. Vastaanottava noodi ei ole kuullut RTS-pakettia törmäyksen vuoksi.
2. Vastaanottava noodi ei saa vastata ylikuullun RTS- tai CTS-paketin vuoksi.
3. Vastaanottava noodi on unitilassa.

Mikäli lähettävä noodi ei saa vastausta *TA*-arvon aikana, se voi siirtyä unitilaan. Tämä olisi kuitenkin virheellinen ratkaisu tapauksissa 1 ja 2. Tällöin lähettävä noodi siirtyisi unitilaan, vaikka vastaanottava noodi olisi hereillä. T-MAC yrittää uudelleenlähetystä kaksi kertaa. Mikäli kumpikaan näistä uudelleenlähetyksistä ei tuota vieläkään CTS-pakettia vastauksena, siirtyy lähettävä noodi unitilaan.

Noodin ei pitäisi myöskään siirtyä unitilaan, mikäli sen naapurit viestivät keskenään; kyseinen noodi saattaa olla tässä tapauksessa seuraava vastaanottaja. RTS- tai CTS-paketin vastaanottamisen aloitus on yksi aiemmin läpikäydyistä aktivaatiotapahtumista, joka näin ollen aloittaa uuden *TA*-jakson.

Mikäli noodi ei ole riittävän lähellä vastaanottaakseen naapurinoodin lähettämän RTS-paketin, täytyy *TA*-jakson kuitenkin olla riittävän pitkä, jotta noodi voi kuulla toisen naapurin lähettämän CTS-paketin.

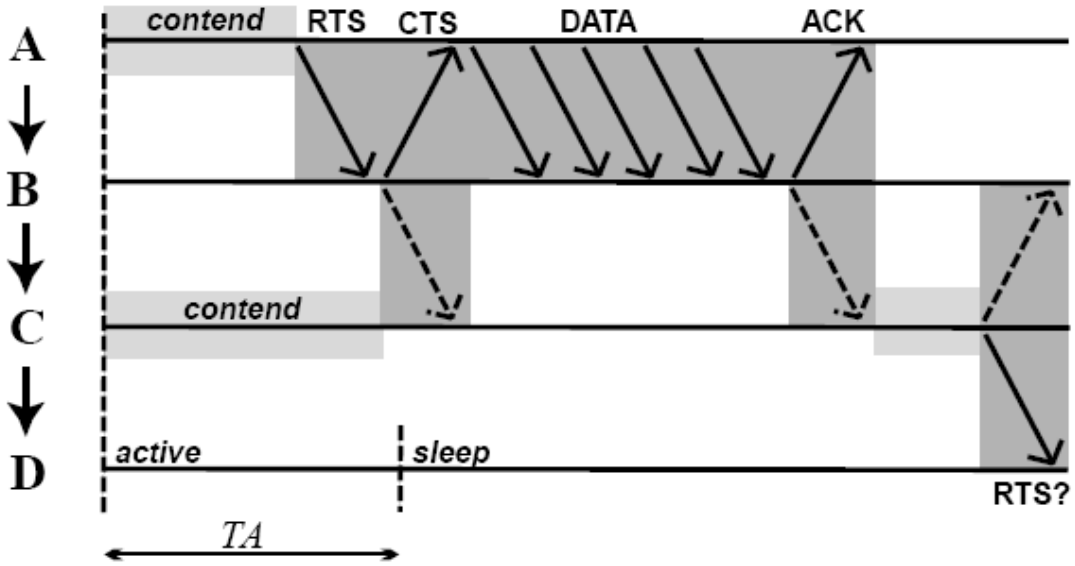


Kuva 12. *TA*-aikajakson pituuden määrittäminen CTS-aloituksen mukaan [30]

Kuvassa 12 kyseinen tilanne on esitetty siten, että noodi C ylikuulee noodin B lähettämän CTS-paketin. Tämän lähetyksen kohteena on siis noodi A, joka on noodin C kuuntelualueen ulkopuolella. Koska *TA* on määritelty niin pitkäksi, että C kuulee CTS-paketin aloituksen, ei se myöskään häiritse noodien A ja B kommunikointia aloittamalla omaa lähetystään. Tällöin tapahtuisi törmäys noodien C ja B lähettämien sanomien välillä. [30]

4.2.3 Varhaisen nukkumisen ongelma

Varhaiset simulaatiot T-MACia käytettäessä paljastivat ns. liian varhaisen nukkumisen ongelman (*engl. early sleeping problem*). Ilmiö tuli selkeästi esille, mikäli verkon liikenne oli enimmäkseen yksisuuntaista. Samankaltainen liikenne on hyvin tyypillistä *node-to-sink*-sanomavaihdossa. Ongelma on havainnollistettu kuvassa 13.



Kuva 13. *Early sleeping problem* [30]

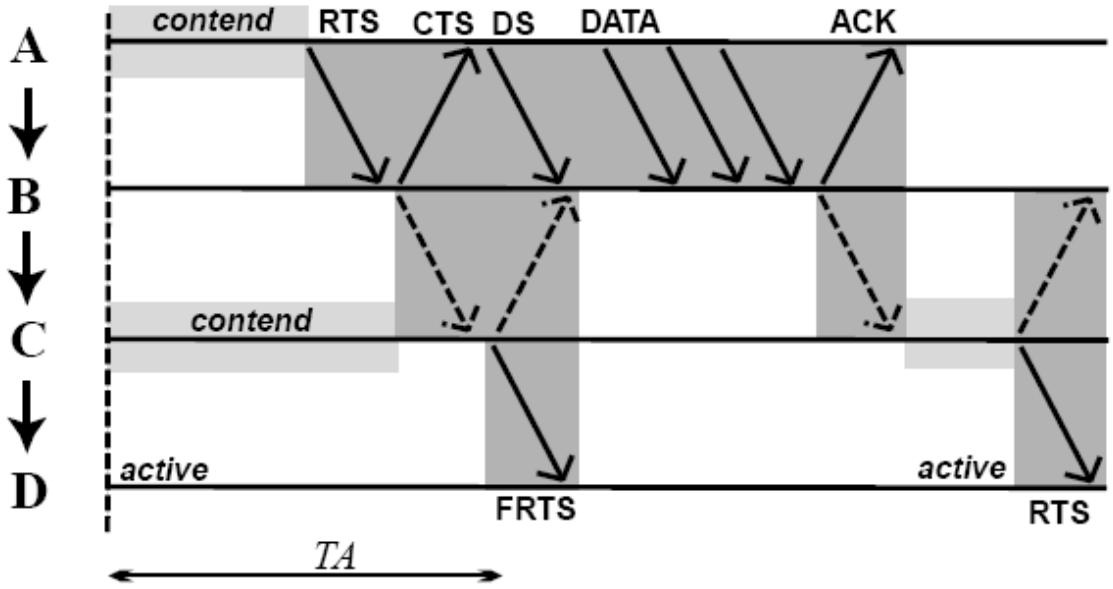
Kaikkien noodien A, B, C ja D voidaan ajatella muodostavan verkon solun. Sanomaliikenne kulkee ylhäältä alas; ts. noodi A kommunikoi pelkästään B:n kanssa, B puolestaan pelkästään C:n kanssa jne. Tarkastellaan tilanteessa noodia C. Joka kerta kun se pyrkii lähettämään D:lle, sen täytyy kilpailla siirtotielle pääsystä. Tällöin se saattaa hävitä joko noodille B, kuulemalla tämän lähettämän RTS-paketin, tai epäsuorasti noodille A ylikuulemalla noodin B lähettämän CTS-paketin.

Mikäli C häviää kilpailun noodille B sen lähettämän RTS-paketin vuoksi, se vastaa CTS-paketilla, jonka noodi D voi ylikuulla. Tässä tapauksessa noodi D ei siirry unitilaan, vaan kuten aiemmin selitettiin, pysyy hereillä niin kauan kunnes noodien B ja C viestinvaihto on päättynyt. Mikäli C puolestaan häviää kilpailun ylikuulemalla noodin B lähettämän CTS-paketin vuoksi, täytyy C:n olla lähettämättä mitään. Koska noodi D ei kuule A ja B noodeja ei se myöskään ole tietoinen näiden aktiivisuudesta. Tällöin noodin D aktiivinen aikaikkuna umpeutuu, ja se siirtyy unitilaan. Seuraava tilaisuus, jolloin noodi C voi lähettää paketin D:lle, on seuraavan kehyksen alussa.

Early sleeping -ongelma tarkoittaa siis yksinkertaistettuna sellaista tilannetta, jossa noodi siirtyy unitilaan vaikka naapurinoodilla on sille lähetettäviä viestejä odottamassa. T-MAC-

protokollaan on kehitetty kaksi erilaista ratkaisua, jotta tämä ongelma voitaisiin välttää: FRTS-menetelmä (*engl. future request-to-send)* ja lähetysjärjestyksen priorisointi puskurin täyttöasteen mukaan (*engl. taking priority on full buffers)*. Nämä ratkaisut käydään läpi seuraavissa kappaleissa. [30]

FRTS-menetelmä pyrkii ratkaisemaan *early sleeping* -ongelman ilmoittamalla vastaanottavalle noodille, että lähettäjällä on vielä viesti odottamassa. Tätä viestiä ei voida lähettää siitä syystä, että lähettäjä ei pääse käyttämään siirtotietä. FRTS-menetelmä toimii seuraavasti: Mikäli noodi ylikuulee CTS-paketin, joka on osoitettu toiselle noodille, se lähettää välittömästi FRTS-paketin. Kyseinen tilanne on havainnollistettu kuvassa 14.



Kuva 14. FRTS-paketti pitää noodin D aktiivisessa tilassa [30]

FRTS-paketti sisältää tiedon siitä, kuinka kauan estävää kommunikaatiota vielä tulee esiintymään ennen kuin noodi itse pystyy lähettämään. Tämä informaatio, joka siis lähetetään noodilta C noodille D, on saatu B:ltä ylikuullusta CTS-paketista. Noodi ei lähetä FRTS-pakettia mikäli se havainnoi liikennettä välittömästi CTS-paketin jälkeen. Sama pätee luonnollisesti tilanteeseen, jossa se on jo aiemmin estynyt lähettämästä muiden CTS- tai RTS-pakettien vuoksi.

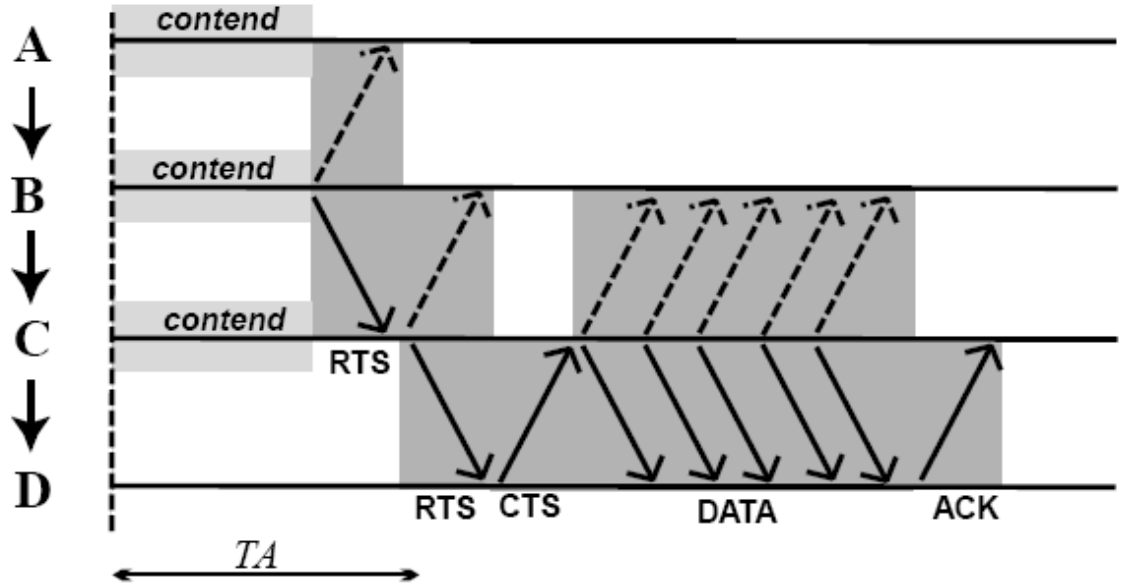
Noodi, joka vastaanottaa FRTS-paketin, eli kuvan 14 tapauksessa noodi D, tietää, että se on seuraavana vastaanottovuorossa, kun siirtotie on vapaa. Tästä syystä se tietää pysyä hereillä eikä siirry unitilaan kuten tapahtuisi ilman FRTS-pakettia.

Koska FRTS-paketti häiritsisi normaalin aikataulun mukaan lähetettävää CTS-paketin jälkeistä datapakettia, tulee CTS-paketin vastaanottamisen jälkeen noodin odottaa FRTS-paketin lähetykseen kuluva aika. Muutoin tapahtuisi törmäys FRTS- ja DATA- pakettien välillä. Estääkseen muita noodeja varaamasta mediaa tänä aikana noodin, joka lähetti alkuperäisen RTS-paketin, eli noodin A, täytyy lähettää pienikokoinen DS-paketti (*engl. Data-Send Packet)*. Välittömästi DS-paketin jälkeen sen täytyy lähettää normaali datapaketti. Tällä toiminnalla vältytään siis korruptoimasta varsinaisia datapaketteja.

Jotta FRTS-menetelmä on mahdollinen, täytyy *TA*-arvoa pidentää yhden kontrollipaketin verran. Tämä lisää hieman *Packet Control Overhead* -ilmiöstä johtuvaa energiankulutusta, mutta parantaa vastaavasti huomattavan paljon siirtonopeutta, mikäli verkossa esiintyy paljon yksisuuntaista liikennettä. Toisaalta, mikäli verkossa on hyvin vähän liikennettä, on myös FRTS-menetelmästä aiheutuva ylimääräinen energiankulutus vastaavasti pieni. [30]

4.2.4 Lähetysjärjestyksen priorisointi puskurin täyttöasteen mukaan

Mikäli noodin reititystaulut tai lähetyspuskuri ovat lähes täynnä, on noodin edullista pyrkiä lähettämään viestejä vastaanottamisen sijaan. Kun noodi vastaanottaa RTS-paketin, joka on osoitettu sille itselleen, se lähettää välittömästi RTS-paketin jollekin toiselle noodille. Normaalisti noodi vastaisi CTS-paketilla RTS-paketin lähettäjälle. Tämä tilanne on nähtävissä kuvassa 15. Noodi C vastaanottaa RTS-paketin noodilta B, mutta sen sijaan, että se vastaisi B:lle CTS-paketilla, se lähettääkin oman RTS-paketin noodille D. Kuvatulla menettelyllä on kaksi vaikutusta. Ensinnäkin noodilla on huomattavan paljon suurempi mahdollisuus lähettää viesti, koska se käytännössä voittaa siirtotien käyttöönsä kuultuaan kilpailevan RTS-viestin. Esimerkiksi kuvan 15 tilanteessa noodi C voi jo lähettää pakettinsa D:lle, vaikka se on hävinnyt kilpailun siirtotiestä noodille B. Täten myös liian varhaisen nukkumisen ongelman todennäköisyys pienenee.



Kuva 15. Puskurin täyttöasteen mukaan suoritettava lähetyspriorisointi [30]

Toinen vaikutus edellä mainitulla menetelmällä on se, että se toteuttaa rajoitetun toiminnallisuuden vuonvalvonnan verkkoon. Tämä on hyödyllistä, mikäli kyseessä on *sink-to-node* -kommunikointi.

Puskurin täyttöasteen mukaan suoritettava priorisointi on kuitenkin tarkoitettu nimenomaan *sink-to-node* -tyyppiseen verkkoon. Mikäli kommunikointi verkossa onkin enimmäkseen kaksisuuntaista, pyrkivät kaikki noodit priorisoimaan omaa lähetystään, mikä puolestaan johtaa huomattavaan kasvuun törmäysten määrässä. Tästä syystä T-MAC käyttää raja-arvoa ongelman välttämiseksi. Noodi voi käyttää lähetyspriorisointia ainoastaan, mikäli se on hävinnyt kilpailun siirtotielle pääsystä kahdesti. [30]

4.3 B-MAC

Berkeley Medium Access Control eli B-MAC on saanut nimensä Kalifornian yliopiston Berkeleyn mukaan, jonka tutkijat Joseph Polastre ja David Culler ovat yhteistyössä Jason Hillin kanssa protokollan kehittäneet.

B-MAC ratkaisun tavoitteena on ollut kehittää MAC-kerroksen protokollaratkaisu, joka on vähän virtaa kuluttava, sisältää tehokkaan törmäysten välttämisen, on toteutukseltaan

yksinkertainen, tehokkaasti siirtotien hyödyntävä vaihtelevilla liikennemäärillä, uudelleenkonfiguroitava ylempien kerrosten toimesta, sopeutuva muuttuviin radiorajapinnan ja verkon olosuhteisiin sekä skaalautuva verkon noodien lukumäärän suhteen. [21]

Nämä tavoitteet ovat hyvin yleisiä kaikille anturiverkkojen MAC-ratkaisuille. Kuitenkin uudelleenkonfiguroitavuus on ominaisuus, jota ei aiemmissa MAC-ratkaisuissa ole toteutettu. Myös erityinen huomio radiotien olosuhteiden muutoksiin poikkeaa aiemmista MAC-ratkaisuista. Aiempiin tutkimustuloksiin tukeutuen on havaittu, että muuttuvilla olosuhteilla radiotiellä on hyvin suuri vaikutus sanomien perillepääsyyn. Esimerkiksi noodi, joka vastaanottaa onnistuneesti 90 % prosenttia sille osoitetuista sanomista, saattaa omistaa naapurin joka saavuttaa ainoastaan 50 % onnistumisprosentin vastaanotettujen sanomien suhteen. Koska anturiverkkosovellukset eivät ole irrallisia ympäristöstään vaan pikemminkin osa sitä, on esimerkiksi lyhytaikaisilla rajuilla sadekuuroilla tai ovien avautumisella ja sulkeutumisella huomattava vaikutus radiorajapinnan olosuhteisiin. [21]

Liikenteen määrä anturiverkossa saattaa vaihdella suuresti ajanhetken mukaan. MAC-ratkaisun konfiguroitavuus antaa täten mahdollisuuden sopeuttaa kerroksen toiminta kulloisenkin liikenteen määrän mukaan ja saavuttaa mahdollisesti tätä kautta suurempia virransäästöhyötyjä kuin kiinteästi toimivassa MAC - protokollassa. [21]

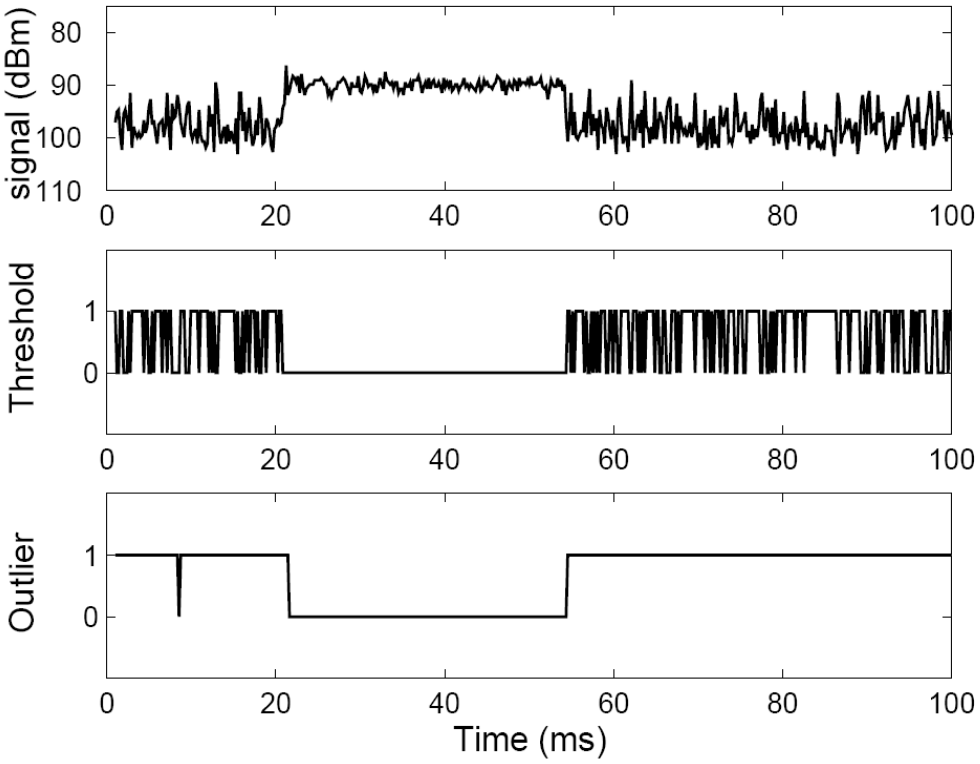
Verrattuna esimerkiksi S-MAC:iin B-MAC sisältää ainoastaan ydintoiminnot jotka kuuluvat MAC-kerrokselle. Ratkaisu käyttää CCA- *(engl. Clear Channel Assessment)* ja *Packet Backoff* -menetelmiä kanavan tilan arviointiin, link layer -kuittaussanomia luotettavan siirron saavuttamiseen sekä LPL (*engl. Low Power Listening)* -tekniikkaa alhaisen virrankulutuksen mahdollistavan kommunikoinnin toteuttamiseen.

B-MAC toteuttaa konfiguroitavuuden toteuttamiseksi ylimääräisiä rajapintoja ylemmille kerroksille, joiden kautta voidaan konfiguroida edellä mainittuja CCA-, LPL- ja *backoff*-toimintoja. Näin ollen B-MAC tarjoaa palvelut, joiden avulla kulloisenkin tilanteen mukaan voidaan optimoida virrankulutusta, viivettä, siirtonopeutta, noodien tasavertaisuutta median saannissa sekä luotettavuutta.

4.3.1 Kanavan varaustilan arviointi

Jotta voitaisiin tehokkaasti toteuttaa törmäysten välttäminen, täytyy MAC-protokollan pystyä mahdollisimman hyvin tunnistamaan onko siirtotie varattu vai vapaa. Kuten aiemmin mainittiin, saattavat radiotien olosuhteet muuttua hyvinkin merkittävästi. B-MAC pyrkii toteuttamaan tehokkaan CCA-menetelmän toteuttamalla ohjelmistopohjaisen metodin arvioimaan taustakohinan tasoa. Signaalin voimakkuutta mitataan silloin, kun oletetaan siirtotien olevan vapaa; esimerkiksi välittömästi paketin lähetyksen jälkeen ja silloin, kun radiokerrokselta ei tule varsinaista dataa. Näytteet tallennetaan FIFO-tyyppiseen (*engl. First-in-First-Out)* puskuriin. Jonon mediaani lisätään eksponentiaalisesti painotettuun, liukuvaan keskiarvoon, laskuarvolla alfa. Mediaania käytetään yksinkertaisena suodattimena lisäämään kohinatason arvioinnin luotettavuutta.

Monet muut protokollat käyttävät ainoastaan yhtä näytettä ja vertaavat sitä taustakohinan perustasoon. Tällainen menetelmä saattaa kuitenkin antaa usein virheellisen tiedon siitä, että siirtotie olisi vapaa. Jokainen virheellinen tulkinta siirtotien tilasta laskee tehokasta kaistanleveyden käyttöä ja lisää myös osaltaan turhaa kuuntelua. Kuten aiemmin on mainittu, saattaa taustakohinan taso vaihdella huomattavan paljon. Paketin vastaanottamisen aikana signaalin taso pysyy puolestaan kohtuullisen tarkasti vakiona.



Kuva 16. CCA-menetelmän käyttäytyminen kanavan tilaa arvioitaessa [21]

Kuvassa 16 tämä voidaan nähdä graafisesti esitettynä. Paketti vastaanotetaan välillä 22ms - 54ms. Ylimmäinen kuvaaja on lähetin-vastaanotinyksiköltä saatu RSSI- (*engl. Receive Signal Strength Indicator)* arvo. Keskimmäisestä kuvaajasta puolestaan voidaan nähdä CCA-algoritmilta tuleva arvo, jossa 1 tarkoittaa kanavan olevan vapaa ja nolla puolestaan kanavan olevan varattu. Alimmainen kuvaaja näyttää *outlier*-algoritmin antaman tuloksen kanavan tilasta. B-MAC etsii näytteiden joukosta arvoja, joissa kanavalta mitattu energia on huomattavasti pienempi kuin vertailukohteena käytetty peruskohinataso. Jos tällainen *outlier*-arvo havaitaan näytejakson aikana, päätellään kanavan olevan vapaa. Tämä perustuu siihen, että kanavalta mitattu signaalin voimakkuus ei mitenkään voi olla peruskohinatason alapuolella, jos aktiivinen lähetys on käynnissä. Mikäli taas viiden näytejakson aikana ei löydetä yhtään vaadittua *outlier*-arvoa päätellään kanavan olevan varattu. Kuvan 16 esimerkissä *outlier*-algoritmi näyttää toimivan varsin tehokkaasti. Esimerkissä voidaan nähdä ainoastaan yksi virhetulkinta siirtotien tilasta. [21]

CCA-menetelmää käytettäessä, B-MAC hyödyntää *backoff*-ajastusmenetelmää paketteja lähetettäessä. B-MAC ei itse määrittele käytettävää *backoff*-aikaa, vaan tämä tieto kysytään ylemmältä kerrokselta. Ylempi kerros puolestaan palauttaa joko oletuksena käytössä olevan *initial backoff time* -arvon tai hylkää tapahtuman kokonaan. Jos tapahtuma hylätään, valitaan pieni satunnainen *backoff*-arvo. *Initial backoff time* -ajan jälkeen suoritetaan CCA *outlier* -algoritmi. Mikäli kanava ei ole vapaa, palautetaan tieto *congestion backoff time* -tapahtumasta. Jos *backoff*-aikaa ei ole lainkaan annettu, valitaan jälleen pieni arvo satunnaiselle *backoff*-ajalle. Ottamalla CCA-algoritmi käyttöön tai poistamalla se käytöstä, ylempien kerrosten palvelut voivat pyrkiä säätämään noodikohtaisen tasavertaisuuden toteutumista sekä tarvittavaa siirtonopeutta verkossa. CCA-algoritmi voidaan myös konfiguroida kokonaan pois käytöstä. Tällöin aikataulutus (*scheduling)* voidaan toteuttaa MAC-kerroksen yläpuolella olevilla kerroksilla. [21]

4.3.2 Alhaisen virrankulutuksen kuuntelutila

Duty cycle -suhteen määrittelemiseen käytetään B-MAC ratkaisussa periodista kanavan näytteistystä, josta käytetään nimitystä LPL (*engl. Low Power Listening)*. Joka kerta noodin herätessä unitilasta se kytkee radionsa päälle ja tarkistaa onko kanavalla liikennettä. Mikäli noodi havaitsee lähetyksen, pitää se itsensä aktiivisessa tilassa niin kauan, kunnes se vastaanottaa paketin. Vastaanoton jälkeen noodi siirtyy jälleen unitilaan. Mikäli kyseessä on virheellinen kanavan tilan tunnistus, toisin sanoen mitään pakettia ei vastaanoteta, *timeout*-tapahtuma siirtää noodin takaisin unitilaan. CCA-metodin luotettava toiminta on erittäin tärkeää pyrittäessä saavuttamaan LPL-menetelmän mahdolliset hyödyt. Taustakohinan perustason määritystä ei käytetä pelkästään löytämään oikea ajoitus lähetykselle vaan myös tarkistamaan kanavan tila LPL-menetelmää käytettäessä. Virheelliset kanavan tunnistukset CCA-algoritmin toimesta lisäävät turhaan kuunteluun hukkaantuvaa energiaa huonontamalla *duty cycle* -suhdetta.

Jotta voitaisiin varmistua luotettavasta tiedonsiirrosta, on ns. *preamble*-arvo mitoitettu saman kestoiseksi kuin kanavan aktiivisuuden tarkistuksen intervalli. Mikäli kanavan tila tarkistetaan 100ms välein, asetetaan myös *preamble*-arvo siten, että sen lähetys kestää vähintään saman verran. Kyseessä on varsinaista sanomaa edeltävä lähetys, jonka

perusteella noodi vastaanottaa dataa. Tämän *preamble*-lähetyksen aikana noodi herää unitilasta, tarkistaa onko kanavalla liikennettä ja vastaanottaa viestin. Turhaa kuunteluun kuluu energiaa siinä tapauksessa, että noodi herää tarkistamaan kanavan tilan ja toteaa, että liikennettä ei ole. LPL-näytteiden välinen aika on maksimoitu, jolloin itse kanavan tilan näytteenottoon kulutettava aika on mahdollisimman pieni. Tämä aika tosin on muutettavissa, sillä se on parametrisoitu B-MAC ratkaisussa. [21]

4.4 AI-LMAC

AI-LMAC, koko nimeltään An Adaptive, Information-centric and Lightweight Medium Access Protocol, on alankomaalaisen Twenten yliopiston tutkijaryhmän Chatterjea, van Hoesel ja Havingan ehdotelma anturiverkoille soveltuvaksi MAC-ratkaisuksi. MAC-protokollan toimintaa kuvaava artikkeli [5] on julkaistu joulukuussa 2004. Protokolla rakentuu tekijöiden aiemman, Lightweight Medium Access Control -protokollan (LMAC) varaan, jonka toiminta on esitelty luvussa 4.4.1 [31]. AI-LMAC perustuu tämän lisäksi myös ajatukselle siitä, että koska langattomat anturiverkot ovat tunnetusti hyvin sovelluskohtaisia verkkoja, tulisi myös MAC-protokollan kyetä mukautumaan erilaisiin käytössä oleviin anturiverkkosovelluksiin, ja näin ollen kyetä esimerkiksi parempaan virransäästöön verrattuna staattiseen MAC-protokollaratkaisuun.

Tässä kappaleessa käydään läpi varsinainen LMAC-osa, joka keskittyy enemmän perinteisten anturiverkkojen MAC-protokollien osa-alueisiin ja tämän jälkeen adaptiivinen sekä tietokeskeinen osa, josta muodostuu täten koko AI-LMAC-protokollaratkaisu. L-MAC osuus muodostuu kappaleista 4.4.1, 4.4.2, 4.4.3 ja 4.4.4.

4.4.1 L-MAC Protokolla

L-MAC protokollassa on pyritty huomioimaan fyysisen kerroksen vaikutukset tehokkaasti. Noodit mukautuvat vastaanottamaansa radiotaajuussignaaliin, jotta hyväksyttävän tasoinen *BER*-bittivirhetaso (*engl. Bit Error Rate)* voidaan määritellä. Tämä suoritetaan yleensä lähettämällä tietty bittikuvio, jonka mukaan vastaanottava noodi mukauttaa vastaanottoherkkyytensä ja synkronointinsa. Tällaisen *preamble*-bittikuvion lähettäminen lisää omalta osaltaan energian kulutusta. Myös vastaanottimen aktivoituminen aiheuttaa

energiankulutusta sivuvaikutuksineen. Kiteitä käyttävät vastaanotinyksiköt kytkevät oskillaattorinsa pois päältä, kun ne asetetaan alhaisen virrankulutuksen tilaan. Siirryttäessä takaisin vastaanoton tai lähetyksen mahdollistavaan tilaan, täytyy oskillaattorikide jälleen käynnistää uudelleen. Tämä kuluttaa sekä aikaa että energiaa. Tiheään tapahtuvat vastaanotin-lähetinyksikön tilanvaihdot vaikuttavat siis selkeästi anturiverkon elinikään.

L-MAC protokollassa pyritään täten minimoimaan tilanvaihtojen lukumäärä mukauttamalla unijaksojen intervallit verkossa liikkuvan datamäärän mukaan sekä pyrkimällä yksinkertaiseen toteutukseen. L-MAC on TDMA-protokolla, jossa jokaiselle noodille annetaan käyttöön yksi aikaväli, jota se itse kontrolloi. Aikavälien uudelleenkäyttö sallitaan turvallisen, ei-häiritsevän etäisyyden puitteissa. Poikkeuksena useisiin muihin TDMA-protokolliin, kyseisessä ratkaisussa ei käytetä keskitettyä aikavälin kontrollointia keskusnoodin toimesta, vaan hajautettua algoritmia [31]. Aikavälien hallinta käydään läpi seuraavassa kappaleessa.

4.4.2 Aikavälien hallinta

Oman aikavälinsä kohdalla jokainen noodi lähettää aina sanoman, joka koostuu kahdesta eri osasta: kontrollisanomasta ja datasanomasta. Kiinteän mittainen kontrollisanoma puolestaan koostuu aikaväliä kontrolloivan noodin tunnistekoodista, noodin ja *gateway*-noodin välisen etäisyyden *hop*-luvusta, vastaanottavan noodin tunnisteesta sekä dataosan pituudesta. Kontrollidataa käytetään noodien välisen synkronoinnin ylläpitoon, ja tästä syystä noodit lähettävät myös aina käyttämänsä aikavälin sekvenssinumeron.

Noodit vastaanottavat kaikki lähinaapurien lähettämät kontrollisanomat, ja mikäli eivät ole itse sanoman vastaanottajana, kytkevät noodit lähetin-vastaanotinyksikkönsä pois päältä seuraavaan aikaväliin saakka. Mikäli sanoma on osoitettu vastaanoton suorittavalle noodille, se kuuntelee lähetyksen siihen saakka kuin dataa riittää. Tämä ei välttämättä ole tarpeeksi täyttämään koko aikavälin kestoa. Tällaisessa tapauksessa sekä lähettävä että vastaanottava noodi kytkee radionsa pois päältä datalähetyksen loputtua. L-MAC sallii ainoastaan yhden sanoman lähetyksen kehyksen aikana. Sanomia voidaan tosin yhdistää, jotta lähetys voidaan suorittaa yhden kehyksen aikana ja täten pienentää viivettä. [31]

4.4.3 Verkon käyttöönotto

Noodien käynnistyessä ensimmäistä kertaa, ovat ne kaikki synkronoimattomassa tilassa. Synkronointi lähtee liikkeelle *gateway*-noodista, jonka kontrollisanomaan kaikki sen välittömässä läheisyydessä olevat ns. *one hop* -naapurit synkronoituvat. *Gateway*-noodi valitsee satunnaisesti itselleen aikavälin, jota se alkaa käyttää. Yhden kehyksen jälkeen, jokainen *gateway*-noodin välitön naapuri tietää, mitä eri aikavälejä sen välittömässä naapurustossa sijaitsevat *gateway*-noodit hallinnoivat. Seuraavaksi nämä noodit valitsevat itselleen satunnaisesti aikavälin, jota ne alkavat kontrolloida. Aikaväli valitaan luonnollisesti niistä vapaista aikaväleistä, jotka eivät ole *gateway*-noodien hallinnoimia. Jokainen noodi ylläpitää taulukkoa aikavälien käytöstä. Taulukko pitää sisällään tiedot aikaväleistä, joita noodi itse ja sen *one hop* -naapurit käsittelevät varattuina. Noodi voi valita käyttöönsä ainoastaan sellaisen vapaan aikavälin, joka ei ole käytössä yhdelläkään sen naapurilla, tai sen naapuri ei pidä aikaväliä taulukossaan. Käyttämällä tätä metodia, voidaan varmistua siitä, että aikavälien uudelleenkäyttö tapahtuu vähintään kolmen hypyn etäisyydellä, joten törmäyksiä tai häiritsevää lähetystä ei tapahdu. [31]

Verkon käyttöönottovaiheessa voi kuitenkin käydä niin, että naapurinoodit sattuvat valitsemaan saman aikavälin kontrolloitavakseen. Tällöin ne ilmaisevat naapurinoodeilleen havaitusta törmäyksestä kontrollisanomissa. Törmänneen sanoman lähettäneet noodit luopuvat tällöin aiemmin valitsemistaan aikaväleistä ja valitsevat satunnaisesti uudelleen aikavälit. Tämä valinta suoritetaan noodin identifikaatioarvoon sidotun *backoff* -ajastimen lauettua.

Noodit kontrolloivat omaa aikaväliään niin kauan, kunnes paristo loppuu tai törmäysten vuoksi niitä kehotetaan valitsemaan uusi aikaväli käyttöönsä. Aikavälejä täytyy luonnollisesti olla käytössä enemmän kuin on suurin yhtäaikaisten yhteyksien määrä noodille. L-MAC protokollassa käytetään 32 aikaväliä. [31]

4.4.4 Reititys gateway -noodeille

Jokainen noodi seuraa etäisyyttä *gateway*-noodiin *hop*-laskurin avulla. Tämä arvo sisällytetään noodin lähettämään kontrollisanomaan. Vastaan otettaessa sanoma noodi etsii

naapuritaulukostaan sellaisen naapurin, joka sijaitsee lähempänä *gateway*-noodia kuin se itse ja valitsee tämän edelleenlähetyksen kohteeksi. Mikäli taulukosta löytyy useita noodeja, jotka sijaitsevat kaikki yhtä lähellä *gateway*-noodia, valitaan näistä satunnaisesti yksi. [31]

4.4.5 Sovelluskohtainen toiminta

AI-LMAC protokolla on kehitetty erityisesti ympäristönvalvontasovelluksia silmälläpitäen. Tällaisissa anturinoodisovelluksissa noodit levitetään kattamaan laajoja maantieteellisiä alueita. On otettava huomioon, että protokollaa suunniteltaessa on yhtenä oletuksena pidetty sitä, että verkon noodien oletetaan levitysvaiheessa olevan tietyissä paikoissa ja noodien liikkuvuus on täten olematonta. Toinen oletus on se, että verkko koostuu heterogeenisistä noodeista, joilla voi olla erilaisia antureita käytössä.

Odotettu skenaario on tyypillisesti sellainen, että anturiverkkoa käyttävät yhtäaikaisesti useat tutkijat, joilla on erilaisia kiinnostuksen kohteita verkosta saatavalle datalle. Toisin sanoen anturiverkko nähdään työkaluna, jota käytetään tiedon keräämiseen. Verkkoon voidaan kohdistaa samanaikaisia datakyselyjä, jotka ovat mahdollisesti osittain päällekkäisiä verkon käytön suhteen. Seuraamus tällaisesta saattaa olla, että tietyt verkon osat ovat hyvinkin kuormitettuja datan siirron suhteen, kun taas toiset osat saattavat olla lähes käyttämättöminä. Arkkitehtuurin täytyisi kyetä mukautumaan edellä mainittuun tilanteeseen ja jatkuvasti sopeuttamaan toimintaansa verkkoon suoritettavien datan keräys pyyntöjen mukaan. Sellaisten verkon osien, jotka ovat tiheässä käytössä, tulisi toimia aktiivisemmin kuin vähemmän käytettyjen osien, jotta data voidaan välittää keskusnoodille alhaisella viiveellä. AI-LMAC korostaa dynaamisen *duty cycle* -jakson merkitystä. On tärkeää, että noodi pystyy mukauttamaan *duty cycle* -jaksonsa siirrettävän datamäärän mukaiseksi. Täten koko verkko mukautuu siitä riippuen, miten paljon ja missä osassa dataa siirretään.

Jotta arkkitehtuuri voisi mukautua dynaamisesti edellä mainittuun tapaan, täytyy tietää millaista dataa on odotettavissa siirrettäväksi tietyntyyppisen kyselyn jälkeen. AI-LMAC

käyttää seuraavassa kappaleessa läpikäytävää DMF-konseptia (*engl. data management framework)* kyseisen ongelman ratkaisuun. [5]

4.4.6 DMF-konsepti

Tässä luvussa käydään läpi DDT (*Data Distribution Table)* -taulukon toiminta, jonka päälle koko DMF-konsepti perustuu. Kyseisen taulukon avulla voidaan tehdä päätelmiä datan laadusta ja liikenteen sijainnista sen perusteella, minkä tyyppinen kysely verkolle annetaan käsiteltäväksi. Noodin vastaanottaessa datakyselyn se tutkii DDT-taulukkonsa sisällön päätelläkseen, kuinka moni sen lapsinoodeista tulee osallistumaan kyselyyn liittyvään dataliikenteeseen. Jokainen noodi ylläpitää yhtä tai useampaa DDT-taulukkoa, joista jokainen kertoo tietyn tyyppisten antureiden olemassaolon lapsinoodien listasta. Esimerkiksi jos noodilla ja sen joillakin lapsinoodeilla on sekä lämpötila- että paineanturi, ylläpitäisi noodi täten kahta eri DDT-taulukkoa kummallekin anturityypille. Taulukkoja päivitetään sen mukaan, minkälaista dataa kulkee noodien kautta. Täten DDT-taulukkojen ylläpito ei aiheuta ylimääräistä lisäkuormaa verkolle.

Kun noodi vastaanottaa havaitun anturilukeman lapsinoodiltaan, se päivittää kyseisen merkinnän asiaankuuluvassa DDT-taulukossa. Tallennetut arvot riippuvat useista muuttujista, esimerkiksi lukeman generoineen anturin tyyppi, anturin mittaama arvo ja alue, josta lukema sai alkunsa. Lisäksi DDT-taulukossa ylläpidetään tietoa pienimmästä ja suurimmasta mitatusta lukemasta sekä välittömän naapurinoodin tunnistetiedosta, jolta mitattu lukema saatiin.

DDT-taulukossa ylläpidetään siis myös lapsinoodien lukumäärää. Tämä on itse asiassa lukema, joka ilmaisee aktiivisten lapsinoodien määrän. Tällä tarkoitetaan sellaisia lapsinoodeja, jotka ovat aktiivisesti mukana jonkin datakyselyn suorittamisessa. Tämän tiedon perusteella voidaan tehdä karkean tason arvioita siitä, miten paljon siirrettävää minkäkin tyyppinen datakysely tulee aiheuttamaan. Tämän tiedon perusteella voidaan edelleen mukauttaa verkon toimintaa dynaamisesti sen mukaan, mikä tulee eri noodien todennäköinen kuormitus olemaan.

Käyttämällä useita taulukoita ja luokittelemalla vastaanotettua dataa saavutetaan useita etuja. Ohjattua hajauttamista käyttäen voidaan kuormittaa vain niitä verkon osia, jotka tulevat kuitenkin olemaan aktiivisesti datan siirrossa mukana. DDT-taulukoiden käyttö tarjoaa myös monipuolisempia vaihtoehtoja kuin ainoastaan alueen mukaan suoritettava hajautus. Käyttäjä voi esimerkiksi syöttää verkolle kyselyn, josta halutaan saada selville auringon säteilyn taso tietyltä alueelta ainoastaan silloin, kun lämpötilalukema kyseisellä alueella ylittää tietyn raja-arvon sekä ilmanpainelukemat ovat tiettyjen raja-arvojen välissä. Tämän tyyppinen kysely edellyttäisi useamman kuin yhden DDT-taulukon käyttöä - tässä tapauksessa sekä lämpötila- että ilmanpaine-DDT-taulukoista. DDT-taulukon käyttö mahdollistaa myös sisään tulevan datakyselyn optimoinnin alueen mukaan. Mikäli datakyselyllä halutaan saada selville kaikki 35 Celsius-asteen ylittävät lukemat, voidaan taulukosta tarkastaa voiko tämä lämpötila ylittyä ainoastaan tietyssä verkon osassa. Mikäli DDT-taulukon mukaan näin on, voidaan kysely kohdistaa nimenomaisen alueen noodeille ilman, että loppukäyttäjän täytyy kyselyä itse rajata.

Kuten aiemmin mainittiin, taulukossa ylläpidetään myös listaa niistä naapurinoodeista, joilta on saatu kyselyn mukaiset lukemat. Tätä tietoa käytetään jälleen datakyselyn hajauttamisessa, mikäli noodi on osallistunut datan välittämiseen kyseessä olevan datakyselytyypin mukaisesti. Koska jokaisen noodin DDT-taulukko perustuu ainoastaan välittömien (toisin sanoen lähimpien) naapurien käyttämiseen, ei taulukon koko riipu noodin syvyydestä anturipuun rakenteessa. Toisaalta DDT-taulukon koko riippuu hyvin suuresti siitä, kuinka paljon erityyppisiä antureita omaavia lapsinoodeja kyseisellä noodilla on.

DDT-taulukoiden ajantasaisuutta ylläpidetään käyttämällä ajastimia johtuen siitä, että taulukoiden sisältämä tieto ei pysy ajankohtaisena koko verkon eliniän ajan. Tästä syystä on tärkeää tietyin väliajoin suunnata datakysely koko verkolle. Täten saadaan kaikkien noodien taulukot päivitettyä taas ajan tasalle. Se, kuinka usein koko verkolle suunnattu kysely tulee suorittaa, riippuu hyvin pitkälle mitattavista fysikaalisista suureista. Täten päivitystaajuuden asettaminen on jätetty loppukäyttäjän asiantuntemuksen varaan. [5]

4.4.7 MAC-kerroksen toiminta DMF-konseptin kanssa

Toisin kuin pelkkä LMAC, AI-LMAC sallii noodin kontrolloivan useampaa kuin yhtä aikaväliä. AI-LMAC pystyy myös muuttamaan noodin kontrolloimien aikavälien määrää sen perusteella, kuinka paljon dataa kyseisen noodin läpi tulee kulkemaan. Protokollatoteutuksessa nojaudutaan siihen, että läpi koko anturiverkon jokaisella noodilla on isä-lapsi-suhde (*engl. parent-child relationship)* johonkin toiseen noodiin siten, että verkon juurinoodi on jokaisen muun noodin isänoodi. Toisin sanoen verkon kaikki noodit voivat olla yhteydessä keskenään yhden tai useamman linkin välityksellä. DDT-taulukkoa käyttämällä jokainen noodi voi päätellä, kuinka tärkeäksi se näkee lähimpien lapsinoodien aikavälitarpeen. Omaa aikavälitarvettaan noodi ei voi DDT-taulukon avulla päätellä, sillä kuten aiemmin mainittiin, sisältää tämä taulukko ainoastaan aktiivisten lapsinoodien informaation. Noodi ei myöskään omaa tietoa isänoodinsa muista lapsinoodeista, sillä ne ovat sen ulottumattomissa.

Aikavälien asettamisen suhteen AI-LMAC eroaa LMAC-ratkaisusta. Siinä vaiheessa, kun tiedetään kunkin lapsinoodin aikavälitarve, ei sitä voida AI-LMAC-ratkaisussa yhtä suoraviivaisesti asettaa lapsinoodille. Tämä johtuu siitä, että kun LMAC-ratkaisun mukaan noodi haluaa antaa aikavälejä käyttöön lapsinoodilleen, on sillä tieto omista ensimmäisen ja toisen asteen naapuruuksista. Tässä tapauksessa noodilla siis ei ole tietoa sen lapsinoodin toisen asteen naapuruuksista, jotka ovat kolmen hypyn etäisyydellä. AI-LMAC-ratkaisussa tyydytään ainoastaan neuvomaan lapsinoodia lähettämällä tieto kullekin noodille siitä, kuinka monta aikaväliä olisi kyseisen tilanteen vallitessa hyvä ottaa käyttöön. Tämän jälkeen aikavälien käyttöönotto jää lapsinoodin vastuulle, ja rajoittavana tekijänä on ainoastaan vapaiden aikavälien määrä.

Kyseinen neuvontaprosessi alkaa juurinoodissa siitä hetkestä, kun ensimmäisen datakysely syötetään anturiverkkoon. Prosessi etenee sen jälkeen aina verkon viimeiseen lehtinoodiin saakka. Prosessin alkamisella juurinoodista varmistutaan siitä, että aikavälien käyttöönantaminen tapahtuu *fairness*-käsitteen mukaisesti. Mikäli neuvontaprosessi aloitettaisiin noodipuun keskeltä, ei näillä noodeilla olisi tietoa sisarustensa lapsinoodien aikavälitarpeista. Täten juurinoodi on ainoa, joka pystyy aloittamaan neuvontaprosessin. Tätä kutsutaan horisontaaliseksi tasapuolisuudeksi (*engl. horizontal fairness)*.

Edellisen lisäksi protokollassa on mekanismi, joka tunnetaan nimellä vertikaalinen tasapuolisuus (*engl. vertical fairness)*. Jotta voitaisiin välttyä puskurin ylivuoto-ongelmalta, tämä mekanismi varmistaa sen että välittömän lapsinoodin käyttöön ei anneta enempää aikavälejä kuin sen isänoodilla on käytössä. Tämä pienentää myös todennäköisyyttä, että datapaketteja jouduttaisiin hylkäämään kaistanleveyden puutteen vuoksi. Samalla varmistutaan siitä, että lehtinoodit eivät saa varattua itselleen ylimääräisiä aikavälejä.

Kun noodi on saanut tiedon itselleen siitä, kuinka monta aikaväliä sen olisi syytä ottaa käyttöönsä, se tarkistaa mitkä aikavälit ovat vapaina sen toisen asteen naapurustossa. Myös indikoitaessa aikavälin omistus AI-LMAC eroaa hieman LMAC-protokollasta. Siinä missä LMAC-ratkaisussa lähetetään kontrollisanoma jokaisen noodin omistaman aikavälin aluksi, AI-LMAC-protokollassa lähetetään sanoma ainoastaan noodin omistamien peräkkäisten aikavälien ensimmäisessä aikavälissä. Seuraavat aikavälit sisältävät ainoastaan dataa, ja täten voidaan vähentää *overhead*-ilmiötä. Lisäeroavaisuutena LMAC-protokollaan verrattuna kontrollisanoma sisältää myös listan kaikista niistä aikaväleistä, jotka noodi omistaa. [5]

4.5 Z-MAC

Kyseinen protokollaratkaisu on uusin työssä esitellyistä MAC-protokollista. Z-MAC-protokollaa suunniteltaessa on otettu lähtökohdaksi se, että pyritään yhdistämään samassa MAC-ratkaisussa TDMA- ja CSMA-kanavansaantimetodien vahvuudet. Lyhenne Z-MAC tulee sanoista Zebra Medium Access Control, joka kuvaa protokollan hybridiluonnetta.

CSMA-kanavanvarausmallista Z-MAC pyrkii saamaan hyötyinä tunnistettavat alhaisen viiveen, sekä korkean käyttöasteen radiotiellä. TDMA-kanavajakomenetelmän hyödyistä Z-MAC pyrkii hyödyntämään erittäin alhaisen törmäysmäärän.

Z-MAC-ratkaisun toteuttajat ovat todenneet CSMA-menetelmän hyötyjä olevan sen yksinkertaisuuden, joustavuuden ja luotettavuuden. Uudet noodit voivat vaivatta liittyä verkkoon sen eliniän aikana, samoin kuin noodien poistuminen ei aiheuta CSMA-ratkaisussa ongelmia. Myös hyvin väljät synkronointivaatimukset kuuluvat edellä mainitun

menetelmän kiistattomiin hyötyihin. Haittapuolena ratkaisussa puolestaan on nähty alttius törmäyksiin. Kuten aiemmissakin kappaleissa on mainittu, jokainen törmäys on luonnollisesti turhaan kulutettua energiaa, jonka käyttö erityisesti anturiverkkosovelluksissa tulisi minimoida. Kantoaallon tunnistuksella voidaan ehkäistä törmäysten tapahtumista, mutta tämä on tehokasta ainoastaan yhden hypyn (*engl. one hop)* naapuruussuhteessa. Kantoaallon tunnistus ei auta törmäysten välttämisessä mikäli kyseessä on kahden hypyn naapuruussuhde. Tätä ns. piilotetun terminaalin ongelmaa (*engl. hidden terminal problem)* voidaan pyrkiä minimoimaan RTS/CTS -sanomavaihdolla, mutta tämän haittapuolena on korkea *overhead* -suhde. Tämä johtuu siitä, että anturiverkkosovelluksissa kulkevat viestit ovat tyypillisesti hyvin pieniä, jolloin varsinaisen hyötykuorman suhteellinen osuus jää pieneksi käytettäessä RTS/CTS -sanomavaihtoa. [23]

TDMA-menetelmä puolestaan pystyy käsittelemään piilotetun terminaalin ongelman kärsimättä hyötykuorman osuuden pienenemisestä sanomavaihdossa. Tämä johtuu kanavajakomenetelmän perusluonteesta, eli vierekkäisten noodien lähetykset voidaan aikatauluttaa tapahtuvaksi eri aikoihin. TDMA-menetelmällä kuitenkin on hyvin monia haittapuolia, erityisesti anturiverkkosovelluksissa: [23]

1. Tehokkaan aikataulutuksen järjestäminen on hyvin hankalaa. Tämän toteuttamiseksi vaaditaan usein keskusnoodi. Rinnakkaisuutta sisältävän aikataulun muodostaminen hyvällä kanavan uudelleenkäyttösuhteella on NP-vaikea ongelma.

2. Kellosynkronoinnin ylläpito on hyvin haastavaa. Pysyminen synkronoinnin vaatimassa toleranssissa vaatii tiheää viestinvaihtoa verkossa, joka puolestaan kasvattaa *overhead*-ilmiötä ja siten lisää virrankulutusta

3. TDMA-menetelmää käytettäessä verkkotopologian muutoksilla on hyvin merkittävä vaikutus aikataulutuksen järjestämiseen. Muutos saattaa vaikuttaa koko verkon laajuisesti.

4. Häiritsevien noodien huomiointi on vaikeaa eritasoisten linkkien vuoksi. Noodit, joilla on sama aikaväli lähetyskäytössä saattavat olla muutaman hypyn päässä

toisistaan, mutta silti riittävän läheisellä etäisyydellä, jotta ne häiritsevät toistensa lähetystä.

5. Alhaisen käyttöasteen vallitessa TDMA ei pysty mukautumaan tilanteeseen, vaan kanavan käyttöaste jää hyvin alhaiseksi, ja viestin välityksen viive on hyvin suuri verrattuna CSMA-toteutukseen

Z-MAC pyrkii saamaan TDMA-toteutuksesta kuitenkin tehokkaan aikataulutuksen järjestämisen. Alhaisen käyttöasteen vallitessa siirrytään käyttämään CSMA-menetelmää, kun taas käyttöasteen kohotessa siirrytään TDMA-toteutukseen. CSMA on käytössä protokollaratkaisun perusmenetelmänä, mutta kilpailun käydessä kiivaammaksi TDMA-aikataulutusta käytetään parantamaan kilpailutilanteiden ratkaisua. Aikavälien jakaminen verkossa tapahtuu käyttöönoton yhteydessä, eli verkon eliniän alussa on paljon *overhead*-ilmiötä johtuen aikavälien jakamiseen tarvittavasta signaloinnista.

Aikavälin asettamisen jälkeen jokainen noodi käyttää omaa aikaväliään jokaisen kehysjakson aikana. Noodia, joka on asetettu kyseiselle aikavälille, kutsutaan kyseisen aikavälin omistajaksi (*engl. owner)*, ja muita noodeja ei-omistajiksi (*engl. non-owner)*. Aikavälillä saattaa olla useampia kuin yksi omistaja, sillä kyseinen aikaväli voidaan asettaa käyttöön myös muille noodeille, jotka sijaitsevat kauempana kuin kahden hypyn etäisyydellä.

Toisin kuin puhtaassa TDMA-ratkaisussa Z-MAC mahdollistaa noodin lähetystoiminnan millä tahansa aikavälillä. Kuten CSMA-protokollissa, noodi suorittaa kanavan tilan tarkistuksen ennen aiottua lähetystä ja suorittaa lähetyksen ainoastaan, mikäli kanava on vapaa. Aikavälin omistajanoodeille on kuitenkin asetettu korkeampi prioriteetti kuin sen ei-omistajille. Tämä on toteutettu pidentämällä kilpailuikkunaa siten, että aikavälin omistajat pääsevät tarvittaessa aina aikaisemmin varaamaan aikavälin käyttöönsä. Z-MAC-ratkaisun TDMA-tyyppinen osuus vaatii ainoastaan paikallisen synkronoinnin käyttöä kahden hypyn naapuruuksien sisällä. [23]

4.5.1 Käyttöönottovaihe

Verkon käyttöönottovaiheessa suoritetaan seuraavat operaatiot: naapurien etsintä (*engl. neighbour discovery)*, aikavälien asettaminen (*engl. timeslot assignment)*, paikallisen kehysjakson vaihto (*engl. local frame exchange)* ja globaali aikasynkronointi (*engl. global time synchronization)*. Nämä toimenpiteet suoritetaan otettaessa verkko käyttöön ja mikäli topologiassa tapahtuu hyvin merkittäviä muutoksia esimerkiksi fyysisen noodien uudelleensijoituksen vuoksi.

Noodin käynnistyessä se aloittaa toimintansa naapurien etsinnällä. Noodi alkaa lähettää *ping* -sanomaa välittömille, yhden hypyn etäisyydellä sijaitseville naapureilleen. Kyseinen sanoma sisältää listan noodin sen hetkisistä yhden hypyn naapureista. Jokainen noodi lähettää 30 sekunnin ajan yhden ping sanoman kerran sekunnissa. Tämä ajankohta sekunnin sisällä valitaan satunnaisesti jotta vältytään törmäyksiltä. Noodit saavat muodostettua näin naapureilta saamastaan informaatiosta myös kahden hypyn naapurilistat. Kyseistä listaa käytetään syötteenä aikavälin asetus algoritmille. Z-MAC käyttää tähän tarkoitukseen DRAND-algoritmia [24]. Kyseinen algoritmi suorittaa aikavälien jakamisen verkon jokaiselle noodille siten, että yhdelläkään noodiparilla, joka sijaitsee kahden hypyn etäisyyden sisällä toisistaan, ei ole samaa aikaväliä omistuksessaan. Tämä aikavälien jako takaa sen, että yksikään noodi ei lähetyksellään häiritse sellaista noodia joka on kahden hypyn etäisyydellä. Tämä minimoi kappaleessa 4.5 mainittua TDMA-ratkaisuille tyypillistä ongelmaa, jossa häiritsevien noodien huomiointi on vaikeaa eritasoisten linkkien vuoksi.

Sen jälkeen, kun noodi on saanut aikavälin käyttöönsä, täytyy määritellä aikajakso (*engl. time frame),* jonka aikana noodi voi käyttää aikaväliä lähetykseen. Perinteisesti TDMA-menetelmässä kehyksen määrittely on ollut kaikille laitteille sama. Jokaisella laitteella on ollut sama kehysjakso ja jokaisen laitteen samaa ensimmäistä aikaväliä 0 on käytetty synkronointiin. Tästä seurauksena koko verkossa on täytynyt olla määriteltynä suurin mahdollinen aikavälinumero MSN (*engl. Maximum slot number),* eikä verkko ole ollut joustava paikallisten muutosten suhteen. Mikäli verkkoon liittyy uusia noodeja, DRAND algoritmi suorittaa paikallisen aikavälien asettamisen ylläpitäen kuitenkin samalla voimassaolevaa aikaväliasettelua. Mikäli kyseinen muutos aiheuttaa tarpeen MSN-arvon

kasvattamiselle, täytyy tämä uusi arvo ottaa käyttöön koko verkon laajuudella. Tämä aiheuttaisi suurta energianhukkaa hyvinkin pienten, verkon topologiassa tapahtuvien, muutosten seurauksena. Z-MAC ratkaisee aikakehyksen määrittelyn siten, että jokainen noodi noudattaa paikallista aikakehystä, joka soveltuu sen naapuruston kokoon ja välttää samalla konflikteja kilpailevien noodien naapurien kanssa. Z-MAC-prokokollassa käytetään ongelman ratkaisemiseen paikallista aikavälien asettamista käyttävää *Time Frame* -menetelmää. *Time Frame* -menetelmä sallii noodien valita oman aikavälikokonsa perustuen niiden paikallisiin kahden hypyn naapurustotietoihin. Tämä menetelmä mahdollistaa sopeutumisen verkon paikallisiin muutoksiin aiheuttamatta aikataulujen vaihtumista koko verkon laajuudella. Mikäli verkossa on useita hajallaan olevia alueita, ja ainoastaan harvoja tiheästi sijaitsevia noodialueita, toimii paikallinen kehysjakson määrittely tehokkaammin kuin globaali aikavälien asettaminen. On myös huomioitava, että Z-MAC ratkaisussa käytetään CSMA-kanavansaantimenetelmää yhdessä TDMA-menetelmän kanssa. Tällöin aikavälit ovat käytettävissä noodien kilpailun kautta, joten ne eivät jää välttämättä kokonaan käyttämättömiksi. [23]

Paikallinen kehyssääntö olettaa implisiittisesti, että jokainen noodi aloittaa ensimmäisen aikavälinsä samalla ajanhetkellä. Tämän mahdollistamiseen ei tarvita ylimääräistä sanomanvaihtoa olettaen, että kellosignaali on tarkasti synkronoitu. Tämä synkronointi suoritetaan ainoastaan kerran verkon käynnistyksen yhteydessä. Myöhemmässä käyttövaiheessa noodit käyttävät vähän virtaa kuluttavaa paikallista synkronointia, joka käydään läpi myöhemmin. [23]

4.5.2 Lähetystoiminnan hallinta

Edellisessä kappaleessa kuvatun *Startup*-jakson jälkeen noodit ovat valmiita aloittamaan varsinaisen lähetystoiminnan. Z-MAC-ratkaisussa noodi voi toimia kahdessa eri tilassa: alhaisen kilpailutason LCL- (*engl. Low Contention Level)* moodissa tai korkean kilpailutason HCL- (*engl. High Contention Level)* moodissa. Noodi siirtyy HCL-tilaan, mikäli se vastaanottaa erityisen ECN-sanoman (*explicit contention notification)* aikajakson t_{ECN} - aikana naapuriltaan, joka sijaitsee kahden hypyn etäisyyden sisällä. Noodi lähettää kyseisen viestin, mikäli se havainnoi erityisen korkean kilpailutason vallitsevan siirtotiellä.

LCL-moodissa noodit voivat kilpailla pääsystä mille tahansa aikavälille, mutta HCL-tilassa noodit voivat käyttää ainoastaan niitä aikavälejä, joiden omistajiksi ne on määrätty tai mikäli ne ovat yhden hypyn naapureita aikavälin omistajanoodeille. Kuitenkin aikavälin varsinaisella omistajalla on aina korkeampi prioriteetti verrattuna ei-omistajiin. Aikavälille haluavat ei-omistajat saavat sen käyttöönsä ainoastaan, jos aikaväliä ei ole osoitettu millekään noodille tai varsinaisella omistajalla ei ole kyseisellä ajanhetkellä dataa lähetettävänään.

On myös otettava huomioon, että LCL-tilassa ollut lähetys, joka on alkanut edellisen aikavälin aikana, jatkuu seuraavan aikavälin puolelle, jossa ollaan HCL-tilassa. Tällaisessa tilanteessa tapahtuu törmäys aikaväliä lähetykseen käyttävän noodin ja aikavälin varsinaisesti omistavan noodin lähetysten välillä. Yksi tapa estää tämä tilanne olisi estää lähetyksen jatkuminen HCL-aikavälin puolelle. Tästä kuitenkin aiheutuisi huomattavaa pirstoutumista aikavälien käytön jatkuvuuteen. Z-MAC ei käytä tätä menetelmää, vaan luottaa siihen että aikavälin koon tulisi olla riittävä useamman kuin yhden paketin lähetykseen, joten varsinaisten omistajien todennäköisyys saada HCL-aikaväli käyttöönsä on hyvin suuri. Z-MAC toteuttaa LCL- ja HCL-tilat käyttämällä B-MAC-ratkaisun *backoff-,* CCA- ja LPL -palveluja. [23]

4.5.3 Tarkka kilpailutilanteen ilmoittaminen

ECN-sanomaa (*engl. Explicit Contention Notification)* käytetään ilmoittamaan kahden hypyn etäisyydellä oleville naapureille, että niiden ei tulisi ilmetä piilotettuina laitteina (*engl. hidden terminal)* aikavälin omistajanoodille tilanteessa, jossa vallitsee korkea kilpailutaso siirtotielle pääsystä. Noodi tekee päätöksen sanoman mahdollisesta lähettämisestä sen oman paikallisen tilanteen perusteella. Paikallisen, kahden hypyn etäisyyden, kilpailun arvioimiseen voidaan käyttää kahta eri menetelmää. Tilanne voidaan arvioida välittömän naapurin lähettämästä kuittausviestistä sen sisältämän *packet loss* -arvon perusteella. Koska kahden hypyn etäisyydellä olevan kilpailun kasvaminen aiheuttaa törmäyksiä, on se myös hyvin voimakkaasti yhteydessä *packet loss* -arvoon. Tämän menetelmän haittapuolena on sen vaatima sanomanvaihto. Toinen tapa on mitata radiokanavan taustakohinan tasoa. Korkean kilpailutilanteen vallitessa taustakohinan taso

kohoaa. Jotta taustakohinaa voitaisiin mitata passiivisesti, ilman periodisia näytteenottojaksoja Z-MAC laskee *noise backoff* -tapahtumien keskiarvoa. *Noise Backoff* -tapahtumassa on kyse tilanteesta, jossa noodi suorittaa kanavan varaustilanteen selvityksen ennen lähetystä, ja mittaustulos ylittää CCA-kynnyksen. Tällöin lähetystä ei luonnollisestikaan suoriteta, koska siirtotie on tulkittu varatuksi.

Kun lähettävä noodi havaitsee korkean kilpailutilanteen vallitsevan, se lähettää yksisuuntaisen yhden hypyn ECN-sanoman sen noodin suuntaan, missä korkea kilpailutilanne on havaittu. Mikäli useampi kuin yksi kohteista on havaittu korkean kilpailutilanteen piiriin kuuluvaksi, voidaan lähettää yleislähetyssanoma (*engl. broadcast message*) joka sisältää kaikkien näiden noodien tiedot. Tyypillisesti anturiverkoissa noodilla on yksi lähetyssuunta. Tämä on kerääjänoodille, johon se lähettää tietoa. Kun esimerkiksi noodi *j* vastaanottaa yhden hypyn ECN-sanoman naapurinoodilta *i*, tarkistaa noodi *j* ensin onko se itse ECN-sanoman kohteena. Mikäli näin on, lähettää noodi *j* yleislähetyssanoman kaikille yhden hypyn etäisyydellä oleville naapureilleen. Tätä kyseistä ECN-sanomaa kutsutaan kahden hypyn etäisyyden ECN-sanomaksi. Mikäli *j* ei ole yhden hypyn ECN-sanoman kohde, se yksinkertaisesti hylkää viestin. Kun noodi vastaanottaa kahden hypyn etäisyyden ECN-sanoman, se asettaa korkean kilpailutilan ilmaisevan HCL-lipun päälle. HCL-lipun tila resetoidaan mikäli uutta kahden hypyn ECN-sanomaa ei vastaanoteta ajastimen t_{ECN} -aikana. Mikäli noodi havaitsee korkean kilpailutilanteen, on hyvin todennäköistä että myös sen naapurit tekevät samoin. Jotta ECN-sanomat eivät tukkeuttaisi verkkoa, käytetään Z-MAC:issä ylikuulemista ongelman välttämiseksi. Noodin *i* havaitessa korkean kilpailutilanteen, se odottaa satunnaisen *backoff* -ajan ennenkuin lähettää yhden hypyn ECN-sanoman. Mikäli tämän ajan kuluessa se vastaanottaa yhden hypyn ECN-sanoman, joka on osoitettu jollekin muulle noodille ja jonka ECN-sanoman lähetyksen kohteena on sama noodi kuin *i*:llä, peruuttaa se oman ECN-sanomansa lähetyksen. Mikäli t_{ECN} -jakson jälkeen kilpailutilanne on silti korkea, noodi *i* toistaa aiemman tilanteen.

ECN-sanoman käyttötarkoitus vastaa CSMA-CA-menetelmän (*engl. Carrier Sense and Multiple Access with Collision Avoidance)* RTS/CTS -sanomavaihtoa. ECN-sanomien

käyttö on kuitenkin vähemmän energiaa kuluttava, sillä kyseinen sanoma lähetetään ainoastaan silloin, kun kilpailutilanne on havaittu korkeaksi. Käyttämällä ECN-sanomien ylilähettämistä, ainoastaan pieni osa lähetetyiksi aiotuista ECN-sanomista joudutaan toteuttamaan. [23]

4.5.4 Alueellinen synkronointi

Z-MAC on muita TDMA-protokollia sallivampi synkronointivirheen suhteen, koska se käyttää CSMA-menetelmää alhaisen kilpailutilanteen vallitessa, jolloin varsinaista synkronointia ei tarvita. Ainoastaan korkean kilpailutilanteen HCL-tilassa Z-MAC protokolla vaatii synkronoinnin ylläpitoa. Tätä ylläpidetään ainoastaan paikallisesti niiden naapurien kanssa, joiden välillä ilmenee viestinvaihtoa. Synkronointiin kuluva ylimääräinen signalointi riippuu myös siitä, kuinka aktiivisesti kunkin naapurin kanssa viestinvaihtoa ilmenee. Mitä enemmän noodi lähettää dataa sitä useammin myös synkronointiin tarvittavia viestejä lähetetään. Tätä menetelmää käyttäen voidaan myös pelkästään vastaanottavien noodien synkronointi toteuttaa passiivisesti: Ne saavat tarvittavan synkronointitiedon vastaanottamistaan sanomista, eikä niiden täydy itse lähettää synkronoinnin ylläpitoon liittyviä sanomia.

Z-MAC ratkaisussa jokainen noodi rajoittaa synkronoinnin ylläpitoon käytettävän signaloinnin sen mukaan, mikä on sen lähettämä datan määrä. Jokainen noodi voi myös itsenäisesti määrätä oman synkronointisignalointikuormansa esimerkiksi käytettävissä olevan energian tai käytössä olevan kaistanleveyden mukaan. Oletuksena käytetään 1 % osuutta synkronointisignalointiin suhteessa lähetettyjen dataviestien määrään.

Paikallisessa synkronoinnin ylläpidossa jokainen dataa lähettävä noodi välittää myös periodisesti oman kelloarvonsa sisältävän synkronointiviestin. Noodin vastaanottaessa kyseisen viestin, se päivittää oman kelloarvonsa saadun tiedon perusteella. Arvoa päivitettäessä vastaanottava noodi ottaa myös huomioon oman painotetun vaellus keskiarvonsa kellolukemasta. Jotkin verkon noodit saattavat sijaita alueella, jossa viestinvälitystä esiintyy harvakseltaan. Tällaisen noodin kello saattaa vaeltaa (*engl. drift*) hyvinkin kauaksi tiuhaan synkronoitujen noodien kellosta. Kun hiljaisella liikennealueella

sijaitseva noodi haluaa lähettää viestin, se saattaa olla pois synkronoinnista. Tällaisen noodin lähettämään kelloarvoon ei tulisi luottaa eikä ottaa sitä käyttöön vastaanottavien noodien toimesta, jotka toimivat tarkasti synkronoituna.

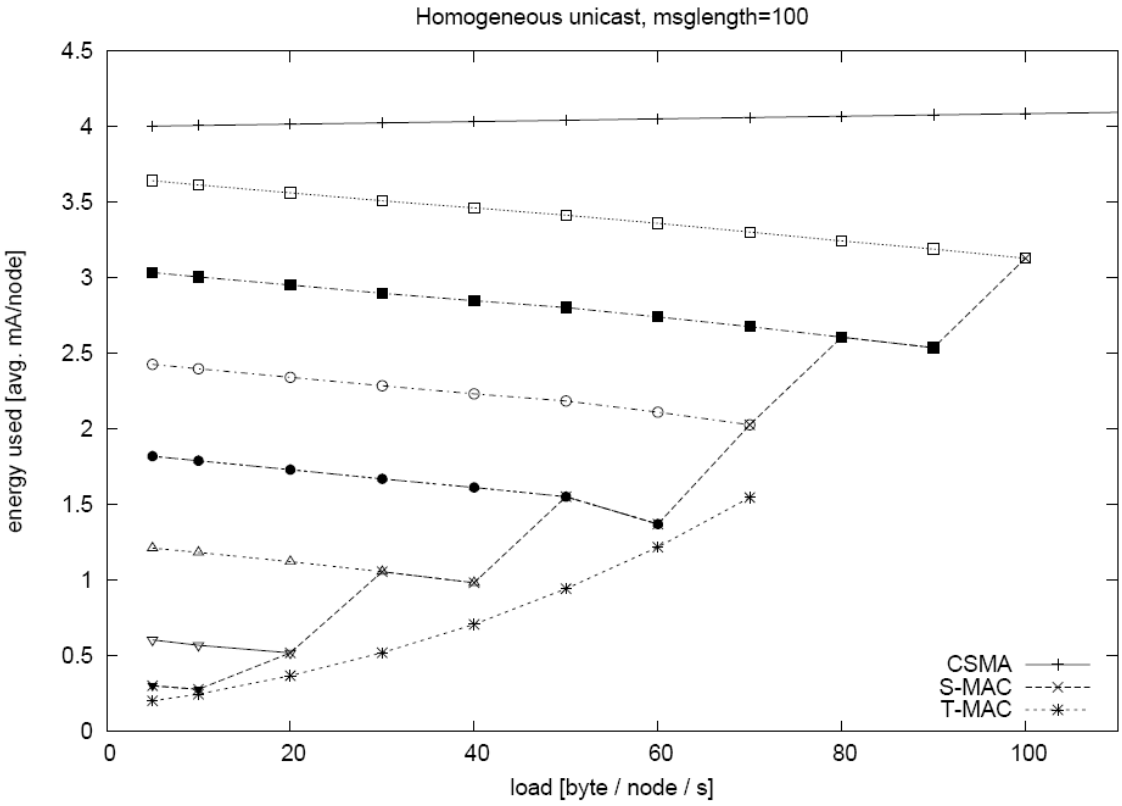
Jotta ongelmalta voitaisiin välttyä, on protokollassa otettu käyttöön luotettavuuskerroin (*engl. trust factor)* β_i mukaan keskiarvoistuspainotukseen. Tämä arvo kertoo kuinka tiheästi lähettäjät ovat synkronoituneet. Tämä mahdollistaa sen, että noodit, jotka lähettävät ja vastaanottavat synkronointiviestejä useammin omaavat yleensä korkeamman luotettavuuskertoimen. Tästä johtuen niiden lähettämät kelloarvot otetaan yleisemmin mukaan laskettaessa kelloarvoa noodeille. Tyypillisesti tällaisia ovat noodit, jotka sijaitsevat reitityspolulla, toisin sanoen ovat edelleen lähettämässä muiden noodien havaitsemaa dataa kerääjänoodille. Toisaalta noodit, jotka taas sijaitsevat reitityksen suhteen etäämmällä ja lähettävät yleensä vain itse havaitsemaansa dataa, eivät tyypillisesti omaa korkeaa luotettavuuskerrointa. Tällainen noodi saattaa olla pitkään unitilassa ilman synkronointiviestien lähetystä ja sinä aikana kelloarvo saattaa vaeltaa hyvinkin etäälle. Z-MAC-protokollaratkaisun tekijöiden mittauksissa on kuitenkin todettu, että lähetystoiminnan aktivoituessa noodi synkronoituu jopa 10 synkronointiviestin lähetyksen jälkeen tarkasti muuhun verkkoon. [23]

4.6 Vertailu ja mittaustulokset

S-MAC on sisällytetty työssä läpikäytäviin protokollaratkaisuihin siitä syystä, että sitä käytetään hyvin yleisesti vertailupohjana arvioitaessa ja mitattaessa muiden protokollaratkaisujen suorituskykyä ja ominaisuuksia. Kyseisen ratkaisun pahimpana puutteena voidaan ehkä pitää sitä, että *duty cycle* on kiinteän mittainen, eikä se täten kykene sopeutumaan kovin hyvin muuttuviin tilanteisiin verkossa. S-MAC joutuu myös ylläpitämään jatkuvasti kasvavaa synkronointitietolistaa verkon koon kasvaessa, ja uusien naapurinoodien liittyessä verkkoon. [30], [21]

T-MAC ratkaisee kiinteän *duty cycle* -ongelman lähettämällä viestit purskeessa ja siirtymällä sen jälkeen hyvin nopeasti unitilaan. Tästä seuraa kuitenkin aiemmin mainittu, ns. *early sleeping problem.* Tämä on T-MAC-protokollassa ratkaistu FRTS-menetelmällä.

T-MAC-ratkaisua on tekijöiden raportissa verrattu mittauksin S-MAC-protokollaan. T-MAC tarjoaa pienemmän energiankulutuksen mukautuvan *duty cycle* -jaksonsa ansiosta. Turha kuuntelu on pyritty minimoimaan siirtämällä noodit hyvin aggressiivisesti unitilaan lähetysjakson päättyessä.

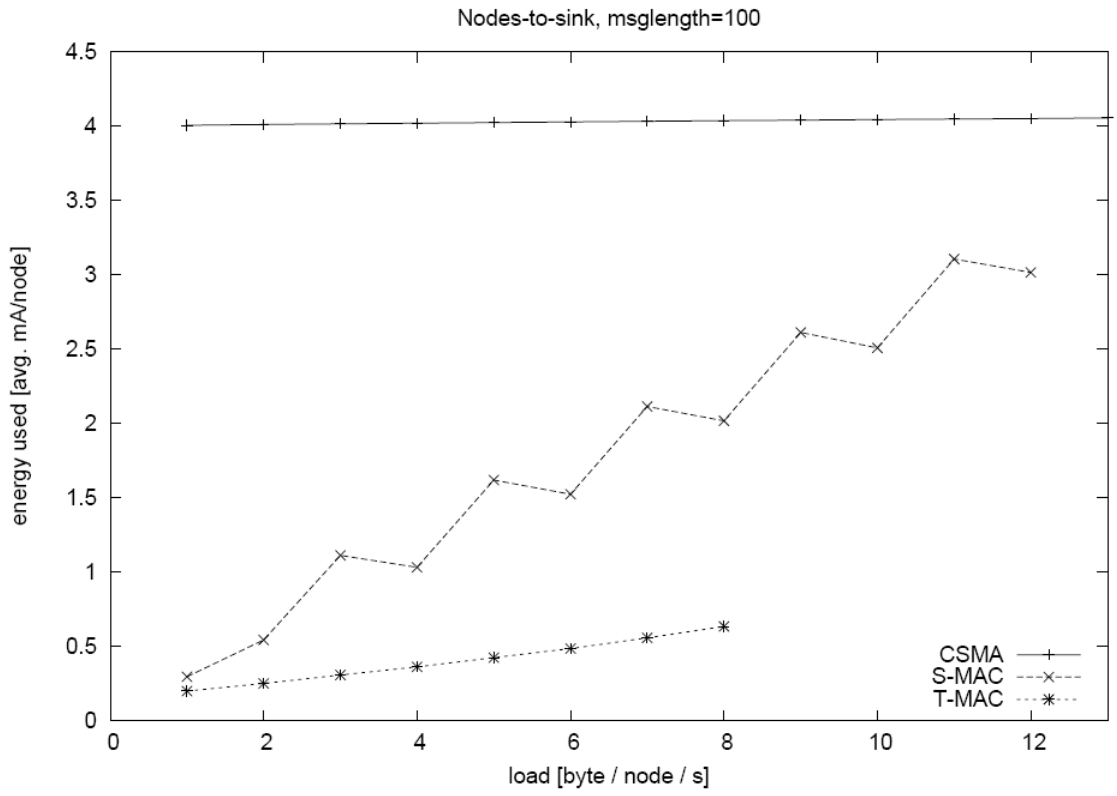


Kuva 17. Energiankulutus homogeenisessa paikallisessa yksisuuntaisessa lähetyksessä [30]

Kuvassa 17 voidaan nähdä, että T-MAC kuluttaa energiaa vähemmän jokaisessa eri variaatiossa verrattuna S-MAC-protokollaan. S-MAC-protokollalle on tehty useita eri mittauksia eri pituisilla aktiivisen jakson kestoilla. Nämä voidaan nähdä kaaviosta neliö-, ympyrä- ja kolmiomerkinnöillä havainnollistettuina. Nämä S-MAC-mittaukset on vielä kytketty toisiinsa ylimääräisellä viivalla, joka yhdistää vähiten energiaa käyttävät S-MAC-mittaukset jokaisella eritasoisella kuormalla. Kyseessä on mittaus, jossa noodit lähettävät sattumanvaraisesti valitulle naapurilleen 100 tavua varsinaista dataa sisältäviä paketteja. Tässä mittauksessa ei T-MAC-protokollan FRTS-ominaisuutta oltu aktivoitu. Yksisuuntainen homogeeninen lähetys on paras mahdollinen tilanne S-MAC-protokollalle,

koska silloin kuorma pysyy vakiona sekä ajan, että paikan suhteen. Kyseinen mittaus on suoritettu täysin simuloidussa verkossa. [30]

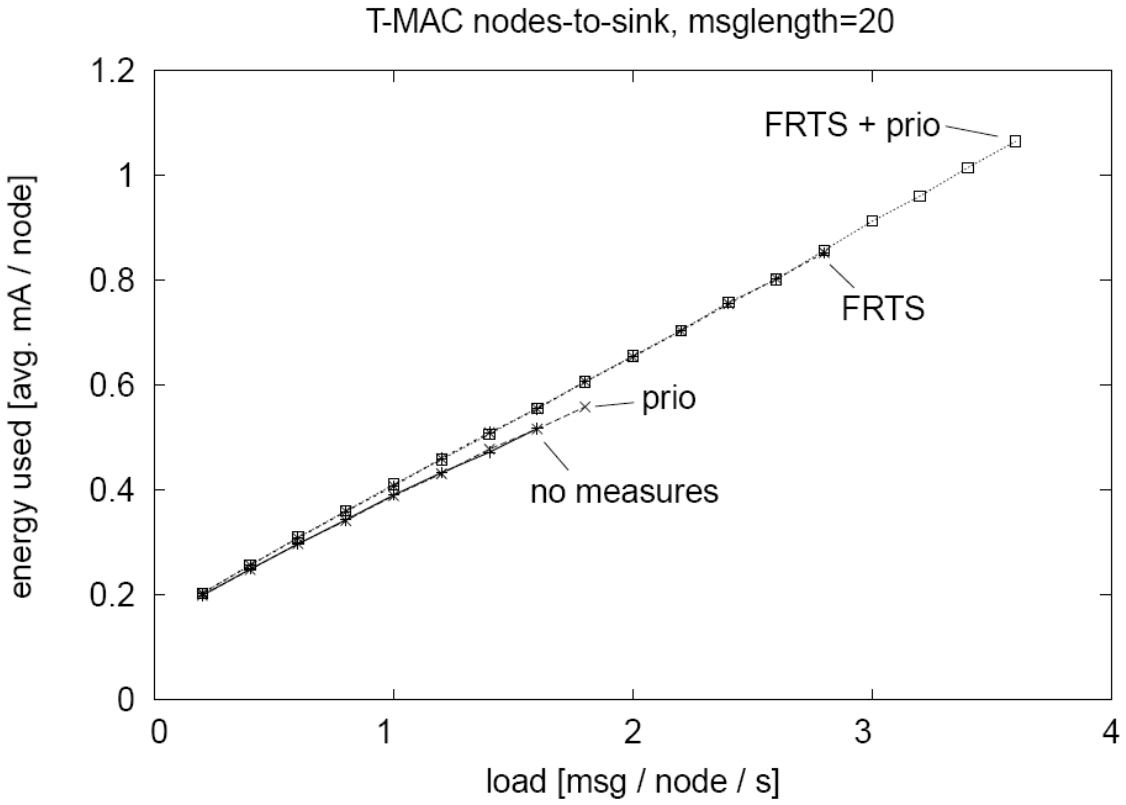
Kuvassa 18 on esitetty *node-to-sink* -lähetystilanne, joka simuloi tilannetta, jossa noodit lähettävät viestinsä kohti verkon reuna-alueella sijaitsevaa kerääjänoodia. Tässä mittauksessa oli aktivoituna T-MAC-protokollan FRTS, ylikuulemisen välttäminen sekä täyden puskurin priorisointi -ominaisuudet. Myös kyseinen mittaus on suoritettu täysin simuloidussa verkossa. Syy sille, miksi T-MAC-protokollan mittauksia ei ole tehty yhtä paljon kuin S-MAC- ja CSMA-CA-protokollien yhteydessä on, että kyseisessä simulaatiossa T-MAC on saavuttanut noodikohtaisen maksiminsa siirtonopeuden suhteen, kun taas muilla protokollaratkaisuilla on voitu siirtonopeutta vielä nostaa.



Kuva 18. Nodes-to-sink suorituskyky [30]

Kuvasta 18 voidaan nähdä, että T-MAC käyttää huomattavan paljon vähemmän energiaa verrattuna S-MAC-protokollaan. Toisaalta T-MAC kärsii heikommasta siirtonopeudesta, joka on huonoimmillaan van Damin mukaan noin 70 % S-MAC:in siirtonopeudesta. [30]

Kuvasta 19 voidaan nähdä liian aikaisen unitilaan siirtymisen ehkäisemiseen tarkoitettujen ominaisuuksien tehokkuus.

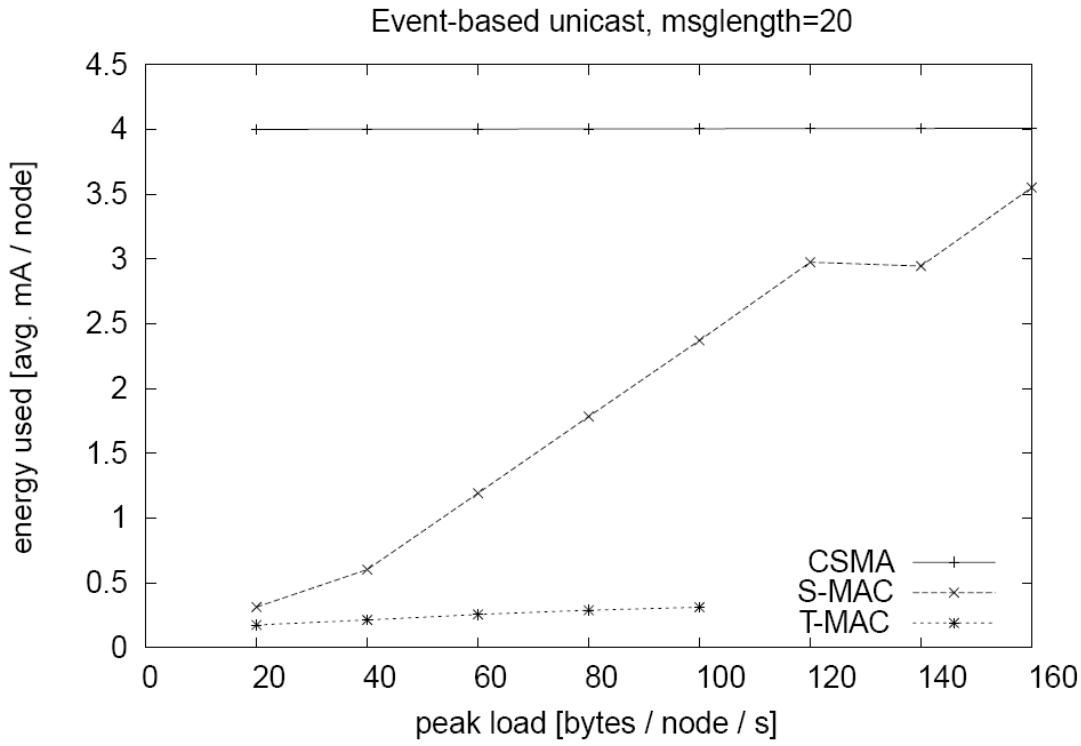


Kuva 19. T-MAC optiot *nodes-to-sink* tapauksessa

FRTS-menetelmä parantaa verkon siirtokykyä van Damin mukaan noin 70 prosentilla, kuitenkin lisäten hieman energiankulutusta. Lisääntynyt energiankulutus on seurausta ylimääräisestä viestinvaihdosta johon koko FRTS perustuu. FRTS lisättynä täyden puskurin priorisoinnilla on edelleen 30 prosenttia tehokkaampi kuin pelkkä FRTS. Pelkkä täyden puskurin priorisointi ilman FRTS-signalointia ei juurikaan lisää verkon suorituskykyä siirtonopeuden suhteen. Kyseinen mittaus on suoritettu täysin simuloidussa verkossa. [30]

Kuvassa 20 on simuloitu hyvin todennäköinen tilanne, jossa anturiverkko toimii. Kyseisessä skenaariossa verkossa ilmenee havaittava tapahtuma 10 sekunnin välein. Tapahtumien keskimääräinen kesto on 5 s, ja se havaitaan 9 noodin alueella. Nämä noodit

lähettävät yksisuuntaisia viestejä naapureilleen tapahtuman keston ajan. Sanomatiheys on kuvaajassa esitetty vaaka-akselilla



Kuva 20. Tapahtumakeskeinen skenaario: aktiiviset noodit lähettävät yksisuuntaisia viestejä

Kuvan 20 mittauksessa valinnaisista optioista ainoastaan ylikuulemisen välttäminen on ollut käytössä. Voidaan kuitenkin nähdä, että T-MAC on huomattavan paljon tehokkaampi energiankäytön suhteen kuin S-MAC. Huomattavaa on, että T-MAC ei kuitenkaan kykene käsittelemään yhtä tiheitä viestinvaihtotilanteita kuin S-MAC, johtuen juuri aikaisen unitilan ongelmasta. Sama havainto voidaan tehdä myös kuvien 19 ja 20 perusteella. T-MAC ei kykene yhtä suureen siirtonopeuteen kuin S-MAC. Tilanteissa, joissa tapahtumia verkossa on vähän, mutta niiden tapahtuessa lyhyen ajanjakson kuluessa on paljon liikennettä, on T-MAC-protokollan suorituskyky energiankulutuksen suhteen jopa 5 kertaa parempi kuin S-MAC:llä. [30]

B-MAC-protokollaratkaisun suorituskykymittaukset on suoritettu sekä S-MAC-ratkaisua, että myös T-MAC-ratkaisua vasten verrattaessa. On kuitenkin tärkeää huomata, että T-

MAC-protokolla on ainoastaan simuloitu ratkaisu, ja tätä ei ole suoritettu oikeassa anturinoodiverkossa. [21]

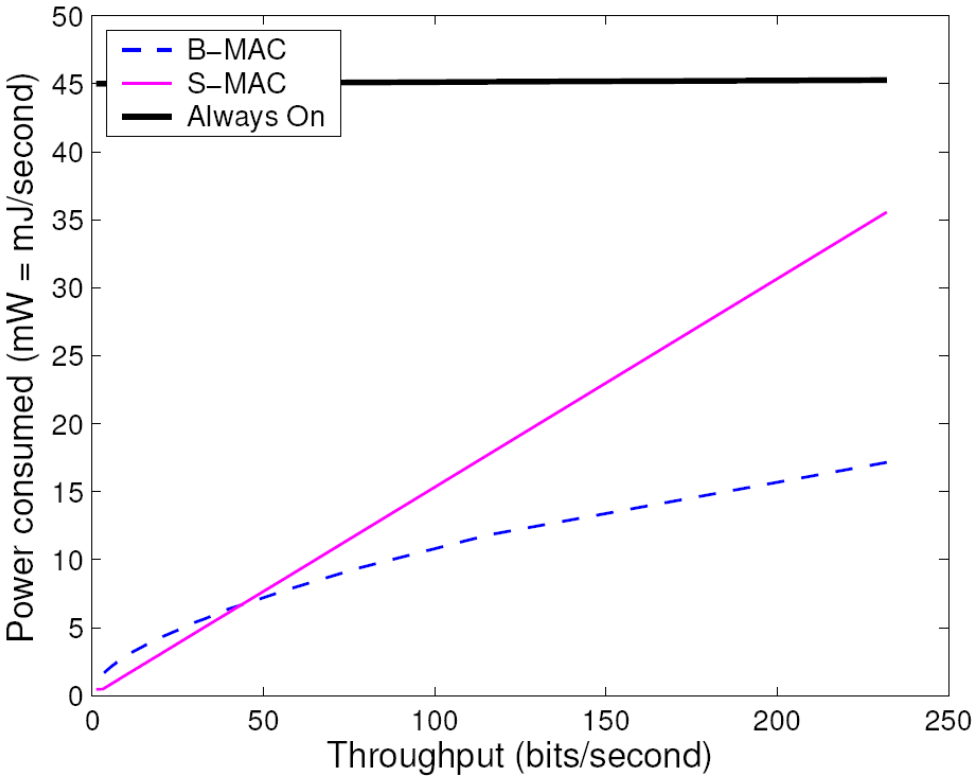
Taulukossa 2 on esitetty S-MAC- ja B-MAC-protokollaratkaisujen *preamble overhead* -arvot. Lähetettäessä saman verran dataa, on S-MAC:in *overhead* -arvo huomattavan paljon korkeampi kuin B-MAC:in. [21]

Length (bytes)	B-MAC	S-MAC
Preamble	8	18
Synchronization	2	2
Header	5	9
Footer (CRC)	2	2
Data Length	29	29
Total	46	60

Taulukko 2. Protokolla *overhead* -vertailu [21]

Vaikka B-MAC:in suorituskyky on korkeampi kuin S-MAC-protokollan, ei sitä kuitenkaan saavuteta tasavertaisuuden kustannuksella. Mittausten analysoinnilla tekijät ovat havainneet, että yksikään noodi ei saa enempää kuin 15 % paremman kanavansaantiasteen verrattuna huonoimpaan noodiin. [21]

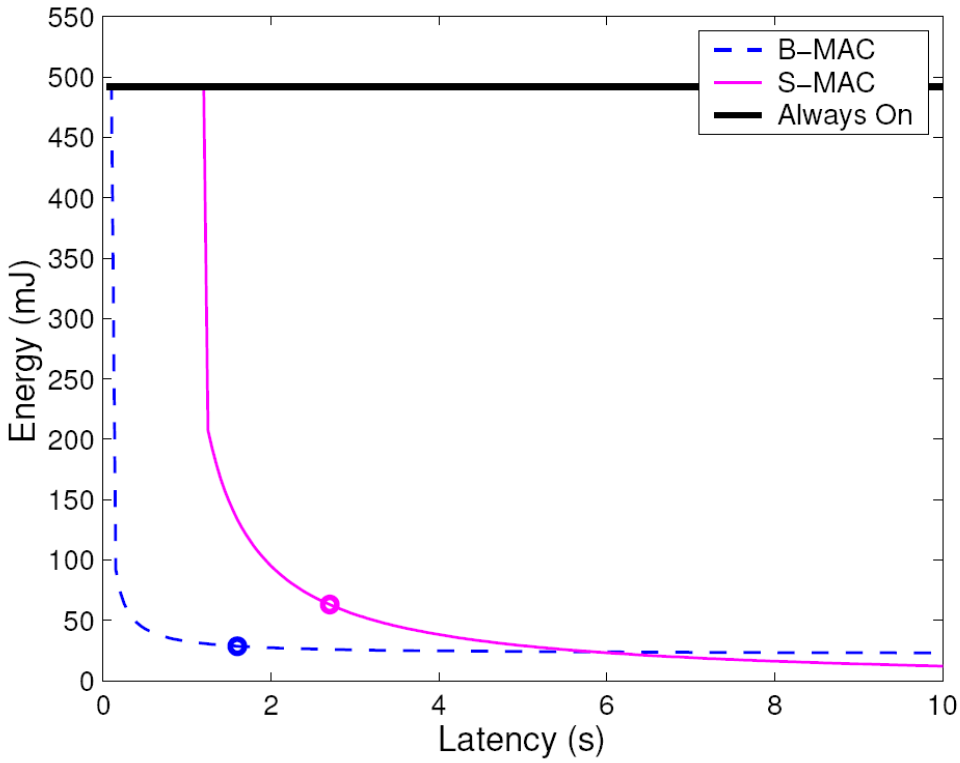
Kuvassa 21 voidaan nähdä energiankulutuksen lisääntyminen suhteessa verkossa siirrettävään datamäärään. Alhaisilla datansiirtomäärillä S-MAC voi käyttää alhaista *duty cycle* -arvoa. Siirrettävän datan lisääntyessä täytyy myös *duty cycle* -arvoa kasvattaa. Näin tehtäessä, esiintyy myös enemmän aktiivisia jaksoja, joista jokaisessa on oma SYNC-periodi. Johtuen synkronointijakson *overhead*-arvosta, kasvaa S-MAC-protokollan energiankäyttö lineaarisesti datamäärän kasvaessa.



Kuva 21: Energiankulutus suhteessa siirrettävään datamäärään [21]

Alhaisilla datamäärillä B-MAC:in energiankulutus johtuu enimmäkseen protokollan *overhead*-arvosta. Alhaisilla datamäärillä, LPL-tilassa, *overhead*-ilmiön osuus kasvaa koska käytetään pitkiä *preamble*-osioita. Datamäärän kasvaessa B-MAC kuitenkin suoriutuu energiankulutuksen suhteen paremmin kuin S-MAC. [21]

Kuvassa 22 on esitetty viiveen vaikutus energiankulutukseen. Mitä enemmän viivettä verkossa sallitaan, sitä pienemmäksi voidaan myös noodien energiankulutus asettaa. Mittauksessa on käytetty 10 reitityshypyn verkkoa, jossa siis 11 noodia. Viive on mitattu verkon päästä päähän.

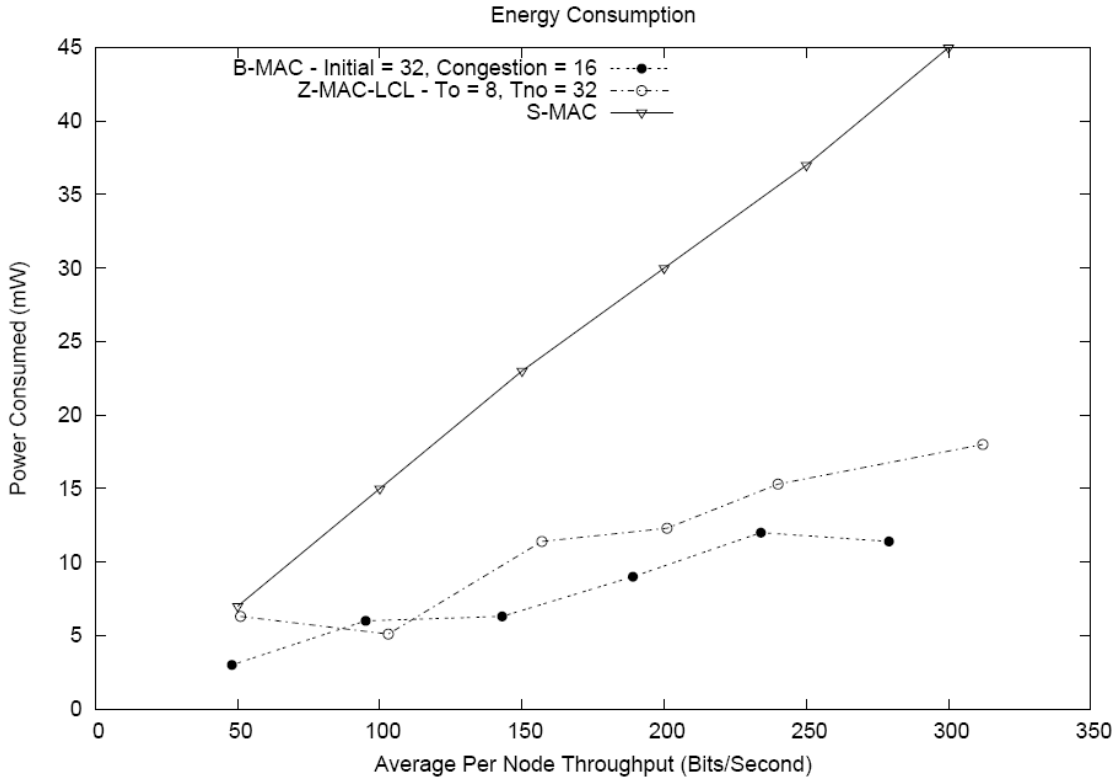


Kuva 22. Viiveen vaikutus energiankulutukseen [21]

Alle 6 sekunnin viiveillä B-MAC suoriutuu paremmin kuin S-MAC. Tätä voidaan pitää myös esimerkkinä siitä, kuinka tärkeää on protokollan konfiguroitavuus verkon eliniän aikana. B-MAC kykenee mukautumaan erilaisiin liikenneskenaarioihin, kun taas S-MAC käyttää kiinteitä arvoja. [21]

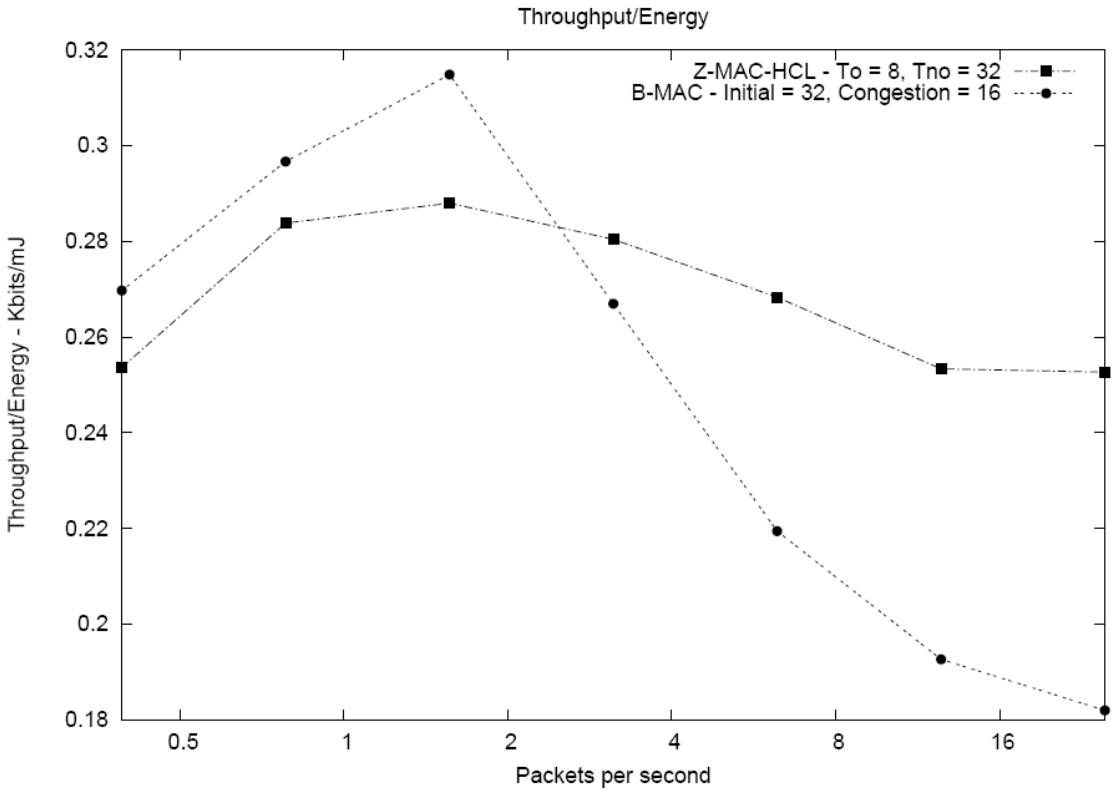
Z-MAC-protokollan mittaustuloksista ei ole saatavilla vertailua S-MAC- tai T-MAC-protokolliin. Z-MAC:in kehittäjät ovat todenneet B-MAC-vertailun olevan riittävä, sillä kyseisen protokollaratkaisun mittaustulokset ovat jo osoittaneet sen olevan parempi ratkaisu kuin S-MAC tai T-MAC energiankulutuksen ja siirtokyvyn suhteen. [21],[23]

Kuvan 23 mittauksessa on verrattu energiankulutusta siirrettävän datamäärän muuttuessa. Tässä tapauksessa verkon noodien lähettämä data on ollut vähäistä ja tiedonsiirtonopeus tästä syystä alhainen. Mittauksessa on käytetty yhden reitityshypyn verkkoa. Z-MAC ei kykene parempaan energiatehokkuuteen verrattuna B-MAC- ja S-MAC-protokollaratkaisuihin, kun verkossa liikkuva data on hyvin vähäistä.



Kuva 23. Energiankulutus alhaisen datansiirron verkossa [23]

Z-MAC:in suurempi virrankulutus johtuu kahdesta seikasta: noodit heräävät pidemmäksi aikaa lähetystä varten, koska *backoff*-ikkuna on suurempi kuin B-MAC:ssä. Toinen syy on se, että Z-MAC-ratkaisussa lähetetään periodisesti synkronointiviestejä. Kyseisessä testissä lähetetyt datamäärät ovat niin pieniä, että kaikki noodit ovat LCL-moodissa. Tästä johtuen *overhead*-ilmiötä ECN-käytön vuoksi ei esiinny.



Kuva 24. Energiatehokkuus *multihop*-verkossa

Kuvan 24 mittauksessa käytössä on ollut useamman reitityshypyn verkko, jossa *duty cycle* -arvoa on vaihdeltu välillä 20 % - 60 %. Alhaisilla lähetysmäärillä B-MAC saavuttaa paremman *throughput*-arvon kuin Z-MAC. Sama pätee myös energiankulutukseen alhaisilla lähetysmäärillä, joilla B-MAC kuluttaa noin 10 % vähemmän energiaa verrattuna Z-MAC:iin. Kun lähetysmäärät kasvavat yli 3 pakettiin sekunnissa, alkaa Z-MAC:in energiankäyttö toimia selkeästi paremmin kuin B-MAC:in. Z-MAC tarvitsee myös vähemmän uudelleenlähetyksiä korkean kilpailutilanteen vallitessa kuin B-MAC. Tämä johtuu siitä, että HCL-moodin aikana, kun ECN-sanoma lähetetään kahden reitityshypyn naapurien omistamien aikavälien aikana, noodit eivät voi lähettää. [23]

5 IEEE 802.15.4 MAC

Tässä luvussa käydään läpi IEEE 802.15.4 -standardin MAC-protokollan tausta, sen toimintaa ja tärkeimmät ominaisuudet. Kappaleessa 5.1 kerrotaan standardin taustan lisäksi vertailu ominaisuuksien valossa aiemmin esiteltyihin MAC-ratkaisuihin. Protokollakerroksen yleisesti kuvattu toiminta ja ominaisuudet esitellään kappaleessa 5.2. Erilaiset kanavanvarausmenetelmät ja niihin liittyvät käsitteet käydään läpi kappaleessa 5.3. Standardin määrittelemä kehysrakenne puolestaan käydään läpi kappaleessa 5.4, ja MAC-kerroksen tarjoamat palvelut kappaleessa 5.5. Kappaleessa 5.6 on lyhyesti esitelty usein IEEE 802.15.4 standardin yhteydessä esiintyvät ZigBee ja 6LoWPAN-ratkaisut. Lopuksi kappaleessa 5.7 arvioidaan IEEE 802.15.4 MAC-ratkaisun toimintaa luvun 4 kriteerilistaa vasten. Kyseisessä kappaleessa pyritään myös vertailemaan standardin ominaisuuksia luvussa 4 esiteltyihin MAC-ratkaisuihin verrattuna.

5.1 Yleistä IEEE 802.15.4 MAC-protokollasta

Kuten aiemmin on mainittu, on IEEE 802.15.4 ainoa olemassa oleva MAC-kerroksen sisältävä standardi anturiverkkojen protokollaratkaisuksi. Kehitystyö lähti liikkeelle joulukuussa vuonna 2000, kun oli huomattu tarve uudentyyppisten langattomien lyhyen kantaman verkkojen protokollastandardille, ja päätettiin perustaa työryhmä kehittämään protokollaperheen 802 standardi 15.4. Olemassa olleet langattomat standardit, esimerkiksi IEEE 802.11 WLAN ja 802.15.1 Bluetooth, eivät soveltuneet anturiverkkojen käyttöön joko kompleksisuutensa, hintansa tai muiden teknisten rajoitusten vuoksi. [11]

IEEE 802.15.4 standardin ensimmäinen versio julkistettiin lokakuussa 2003. Tässä työssä esitettyjen MAC-protokollien esitysjärjestys on kronologinen sillä poikkeuksella, että IEEE 802.15.4 ainoana standardina on jätetty viimeiseksi. Verrattaessa aiemmin esiteltyjen MAC-ratkaisujen julkaisuvuosiin, nähdään että IEEE 802.15.4 on ollut hyvin pitkään työn alla. Tämän voidaan sanoa olevan tyypillistä standardityöskentelylle. Ensimmäinen standardin julkaisu on tapahtunut hieman S-MAC-protokollan esittelyn jälkeen. T-MAC, joka voidaan nähdä myös S-MAC:in paranneltuna versiona, on julkaistu seuraavana. B-

MAC ja AI-LMAC ovat julkaistu vuoden 2004 loppupuolella, ja tuorein työssä käsitellyistä protokollaratkaisuista, Z-MAC, siitä noin vuoden kuluttua marraskuussa 2005.

IEEE 802.15.4 MAC-ratkaisussa on myös tiettyjä samankaltaisuuksia tai suoraan samoja ratkaisuja kuin aiemmin esitellyissä tutkimustason MAC-ratkaisuissa. Näitä ovat hybridiluonne aikajakomenetelmän ja kilpailuperustaisen kanavanvarauksen suhteen, joka on käytössä myös Z-MAC-protokollassa, sekä CCA-menetelmä kanavan varaustilanteen arvioinnissa jota käytetään myös B-MAC-protokollassa.

5.2 MAC-kerroksen toiminta

IEEE 802.15.4 -standardi tarjoaa lähes jokaisesta näkökulmasta tarkasteltuna useita vaihtoehtoja toimintatapaan liittyen. Kanavaranvarausmenetelmiä voidaan sanoa olevan käytössä kaksi erilaista, sekä näiden yhdistelmä. Sama pätee myös verkkotopologioihin, joita itse standardi määrittelee kaksi erilaista. Käyttämällä MAC-kerroksen tarjoamia verkonmuodostuspalveluita voidaan kuitenkin rakentaa monimutkaisempia verkkorakenteita. Myös erilaisia laitetyyppejä, tai laitteiden rooleja, on standardin määritelmään mukaan kaksi eri tyyppiä. Näitä rooleja voidaan vaihtaa laiteyksilöiden välillä verkon toiminnan aikana.

IEEE 802.15.4 tarjoaa mahdollisuuden käyttää kahta erilaista kanavanvarausmenetelmää. Nämä ovat useissa tässä työssä esitellyistä MAC-ratkaisuissa käytössä oleva CSMA-CA-menetelmä sekä TDMA. Kilpailuperusteinen CSMA-CA-toimintatila sekä TDMA-menetelmään perustuva kanavanvaraus voivat olla myös yhtäaikaisessa käytössä. Tämä on mahdollista siten, että osa kehyksen kestosta on varattu normaalille kilpailuperusteiselle kanavanvaraukselle. Tällöin käytössä on aikaväleihin perustuva niin sanottu aikaviipaloitu CSMA-tila. Jokainen kilpailulle vapaaksi merkitty aikaväli on noodien käytettävissä, vaikka kehys on jaettukin TDMA-tyyppisesti.

Osa kehyksestä voidaan myös varata vain tiettyjen noodien käyttöön siten, että yksittäiselle noodille on varattu yksi tai useampi aikaväli ainoastaan sen käyttöön. Käytettäessä kyseistä tilaa voidaan törmäysten määrän olettaa olevan minimaalinen johtuen käytössä olevasta puhtaasta TDMA-kanavanvaraustekniikasta. Myös viiveen voidaan olettaa

pienenevän käytettäessä tietylle noodille osoitettuja aikavälejä. Mikäli kyseisellä noodilla on paljon dataa lähetettävänään, voi se mahdollisesti suorittaa lähetyksensä yhden kehysjakson aikana. Mikäli noodi joutuisi joka kerta kilpailemaan aikavälistä, lähetys kestäisi pidempään. Toisaalta noodi, jolla ei ole suoraan osoitettuja aikavälejä voi joutua odottamaan lähetysvuoroaan pidempään, sillä osa kehyksestä on tällöin varattu pelkästään muiden noodien käyttöön. Odotusjaksoa voi vielä pidentää mahdollisesti määritelty inaktiivinen periodi, jolloin verkossa ei ole liikennettä lainkaan. Edellä mainitun kanavanvarausmenetelmän edellyttämä niin sanottu superkehys käydään tarkemmin läpi kappaleessa 5.4.

Mikäli käytössä ei ole taattuja aikavälejä eikä kilpailuvapaata aikavälirakennetta muutenkaan, toimitaan puhtaassa CSMA-CA-tilassa. Kyseistä kanavanvarausmenetelmää käytettäessä törmäysten määrä luonnollisesti lisääntyy. Hidden terminal -ongelmaa vastaan protokollassa ei ole minkäänlaista suojausta, esimerkiksi RTS-CTS-kättelymekanismia. Törmäysten määrän vähentämiseksi käytetään satunnaisesti valittua lähetysajankohtaa. Nämä kaksi edellä esiteltyä CSMA-CA-kanavanvarausmenetelmää käydään tarkemmin läpi kappaleessa 5.3.

Tasapuolisuuden näkökulmasta tarkasteltuna CSMA-CA on tasapuolinen kaikille noodeille, mutta törmäysten määrän kustannuksella. Tämä on siis tyypillistä kyseiselle kanavanvarausmenetelmälle. Noodien tasapuolista kohtelua voidaan vähentää myös tarkoituksellisesti, kuten tapahtuu käytettäessä vain tietyille noodeille tarkoitettuja aikavälejä. Hyvin usein anturinoodisovelluksissa tämä onkin järkevämpää, sillä yleensä verkon kokonaissuoritus on ratkaiseva tekijä, eikä se pääseekö jokainen noodi tasapuolisesti lähettämään.

Standardi määrittelee myös kaksi erilaista laitetyyppiä ja näille erilaisia rooleja verkon toiminnassa. Laitteiden ei tarvitse välttämättä olla fyysiseltä toteutukseltaan mitenkään erilaisia, vaan yksi ja sama laitetyyppi voi toimia kahdessa eri roolissa. Täysitoimintainen FFD-laite (*engl. Full Function Device)* sisältää kaikki MAC-tason palvelut ja voi toimia verkossa joko verkon koordinaattorina, keskusnoodina tai tavallisena anturinoodina. Rajoitetun toiminnallisuuden RFD-laite (*engl. Reduced Function Device)* sisältää

rajoitetun valikoiman MAC-kerroksen palveluita ja voi toimia ainoastaan tavallisena anturinoodina. [18] RFD-roolia käytettäessä pyritään maksimoimaan noodin elinikä. RFD-noodilla ei ole kykyä reitittää sanomia, joten se voi toimia ainoastaan lehtinoodina verkossa.

Varsinaisessa anturinoodisovelluksessa noodin käyttämää roolia voidaan myös vaihtaa. Tätä ei tueta standardissa, mutta kyseinen toiminta voidaan toteuttaa MAC-kerroksen päälle. Näin voidaan välttyä tilanteelta, jossa noodin toimintamoodi rajoittaa verkon toimintaa. Esimerkiksi isoissa verkoissa voidaan pidentää verkon elinikää kierrättämällä FFD-roolia eri noodien välillä. Yksi tällainen tilanne liittyy edellä mainittuun RFD-noodin rajoitteeseen sanomien edelleen lähetyksessä. Hajautetussa tai hierarkkisessa puuverkossa jotkut noodit saattavat joutua edelleen lähettämään hyvin paljon sanomia, ja tällöin niiden virrankulutuskin on suuri. Vaihtamalla roolia noodien välillä, voidaan virrankulutusta tasata yksittäisten noodien välillä ja täten pidentää verkon kokonaiselinikää.

IEEE 802.15.4 tukee suoraan MAC-tasolla kahta eri verkkotopologiaa. Nämä ovat tähtiverkko ja hajautettu verkko. Tämän lisäksi standardi antaa mahdollisuuden käyttää klusteroituja verkkoja, vaikka tämä ei kuitenkaan ole osa standardin määrittelyä. [11] Klusteriverkkoihin liittyvä toiminnallisuus voidaan siis toteuttaa MAC-kerroksen yläpuolelle, ja näin ollen mahdollistetaan suurten anturiverkkojen muodostaminen. Standardi siis ei rajoita käytettävää verkkotopologiaa ainoastaan tähti- tai hajautettuun verkkoon, vaan mahdollistaa muidenkin topologioiden käytön. Standardi käyttää 64-bittistä osoiteavaruutta noodeille, joten tämä mahdollistaa hyvin suuret verkot laitekohtaisilla osoitteilla.

Standardissa määritellään MAC-kerrokselle neljä eri kehystyyppiä. Näitä ovat majakkakehys, datakehys, kuittauskehys ja MAC-komentokehys. Majakkakehystä käytetään synkronointiin ja superkehyksen rajojen määrittelyyn. Datakehys toimii varsinaisena hyötykuorman välittäjänä ja kuittauskehys puolestaan kuljettaa varmennustietoa onnistuneeseen sanoman välitykseen liittyen. Tämä ei kuitenkaan koske sanoman välitystä koko reitityspolun osalta, vaan ainoastaan välittömien naapurien kesken.

Komentokehystä käytetään ensisijaisesti valvontatiedon välittämiseen MAC-kerrokselle. Nämä kehystyypit ja niiden tehtävät käydään tarkemmin läpi kappaleessa 5.4.

MAC-protokolla tarjoaa ylemmille kerroksille kaksi eri palveluluokkaa. Nämä ovat MAC-data- sekä MAC-hallintapalvelut. Datapalvelut sisältävät kolme eri primitiiviä, jotka liittyvät nimensä mukaisesti datansiirtoon liittyviin toimintoihin. Hallintapalveluluokka puolestaan sisältää edelleen useita erityyppisiä palveluita, joita ovat muun muassa radiokanavien skannaus, radion tilan hallinta, assosiointi verkon koordinaattorin kanssa, GTS-aikavälien hallinta, kadonneen laitteen hallinta, synkronoinnin hallinta, majakkatilan hallinta, majakkasanomaa käyttämättömän tilan synkronointi sekä kommunikointitila. Näitä palveluja käsitellään tarkemmin kappaleessa 5.5. [11]

5.3 Kanavanvaraus

IEEE 802.15.4 -standardissa käytettävä CSMA-CA-kanavansaantimetodi poikkeaa esimerkiksi Ethernet-tekniikan kanavansaantimetodista siten, että erityistä huomiota on pyritty kiinnittämään törmäysten välttämiseen, ei pelkästään havainnoimiseen. Ethernet-tekniikassa käytetään Carrier Sense and Multiple Access with Collision Detection (CSMA-CD) -metodia, jossa törmäykset pyritään havaitsemaan tehokkaasti. Jokainen törmäys on kuitenkin turhaan käytettyä energiaa, joten langattomien anturiverkkojen luonteelle tyypillisesti tämä energian käyttö on pyritty minimoimaan.

Standardi tarjoaa kaksi erilaista toimintatapaa CSMA-CA kanavanvarauksessa. Nämä ovat majakkasanomaa käyttävä tila sekä ilman majakkasanomia toimiva CSMA-CA-tila. Molemmat toimintatavat käydään läpi kappaleessa 5.3.1. Näiden lisäksi standardi mahdollistaa osittaisen TDMA-kanavanvarausmetodin käytön, jota käsitellään superkehyksen yhteydessä kappaleessa 5.3.2.

5.3.1 CSMA

Kuuntelemalla käytettävää radiotaajuutta ennen lähetyksen aloittamista, voidaan varmistua siitä että media on sillä hetkellä vapaa. Tätä kutsutaan kantoaallon tunnistus (*engl. Carrier*

Sense) -tekniikaksi. Täyttä varmuutta törmäyksen välttämisestä kantoaallon tunnistuskaan ei tuota, mutta vähentää huomattavasti niiden todennäköisyyttä.

Poikkeuksena CSMA-CA-protokollan käytössä ovat majakka-, GTS-aikaväli- sekä kuittauslähetykset. Jokainen näistä lähetystyypeistä voidaan suorittaa ilman CSMA-CA-kanavansaantiprotokollan käyttöä.

Mikäli noodi haluaa suorittaa lähetyksen kilpailulle vapaan CAP-jakson (*engl. Contention Access Period)* aikana, käyttää se viipaloitua (*engl. slotted)* CSMA-protokollaa. Törmäysten todennäköisyyden pienentämiseksi protokollassa käytetään satunnaisviiveitä. CAP-jakson aikavälit on jaettu pienempiin aikaväleihin, joita kutsutaan backoff -jaksoiksi. Yhden backoff-jakson kesto on verrattavissa 20:een kanavasymbolin lähettämiseen vaadittavaan aikaan. Mikäli noodi haluaa lähettää viestin, mutta kantoaallon tunnistus huomaa, että radiotaajuudella on lähetys meneillään, on kyseessä niin sanottu backoff-tapahtuma.

Laite ylläpitää kolmea eri muuttujaa CSMA-CA-protokollan käyttöä varten: *NB, CW* ja *BE*. Muuttuja *NB* laskee backoff-tapahtumien lukumäärää, *CW* puolestaan kertoo senhetkisen ruuhkaikkunan koon (*congestion window)* ja *BE* on backoff-eksponentti kyseisellä ajanhetkellä. *BE*-muuttujaa käytetään laskettaessa viivearvoa lähetysyrityksen yhteydessä. Kun halutaan suorittaa uuden paketin lähetys, näiden muuttujien arvoiksi asetetaan *NB*=0, *CW*=2, *BE=macMinBE*, jossa *macMinBE* on protokollakohtainen parametri.

Laite odottaa satunnaisesti saadun kokonaisluvun *r* kertaa *backoff*-periodin määräämän aikajakson ja suorittaa kantoaallon tunnistusoperaation. IEEE 802.15.4 -standardissa tätä kutsutaan CCA-operaatioksi (*engl. Clear Channel Assessment)*. Mikäli media on vapaa, vähennetään *CW*-parametrin arvoa ja odotetaan, että seuraava *backoff*-aikajakson raja ylittyy, jonka jälkeen suoritetaan uusi CCA-operaatio. Mikäli siirtotie on edelleen vapaa, päätellään laitteen voittaneen kilpailun siirtotiestä ja lähetys voidaan aloittaa.

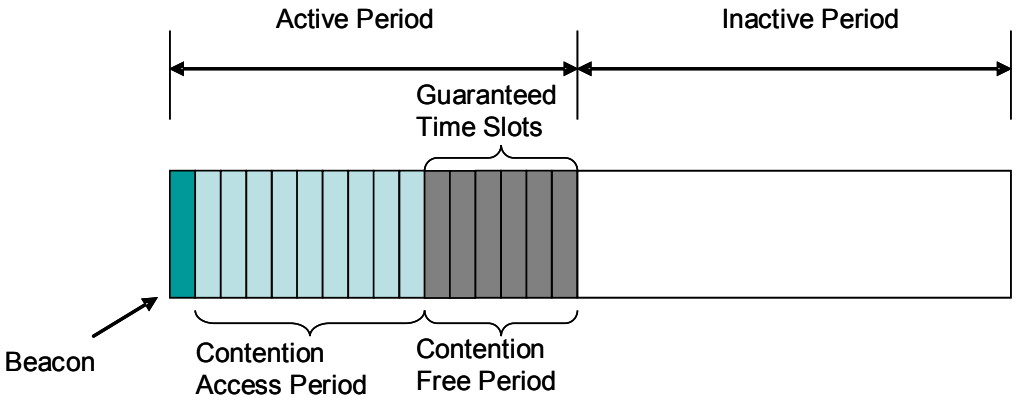
Mikäli jompikumpi CCA-operaatioista osoittaa siirtotien olevan varattu, *NB*- ja *BE*-arvoja kasvatetaan ja *CW*-arvoksi asetetaan alkuperäinen 2. Mikäli *NB* ylittää määritellyn raja-arvon, noodi hylkää kyseisen kehyksen ja raportoi lähetyksen epäonnistuneeksi. Mikäli raja-arvo ei ylity, odotetaan jälleen *r* kertaa *backoff*-arvon mukainen aika ja yritetään uudelleen.

Kuten aiemmin mainittiin, standardi tarjoaa myös toimintamoodin, jossa ei käytetä majakkasanomia lainkaan (*engl. nonbeacoded mode)*. Tässä toimintatilassa verkon koordinaattorinoodi ei siis lähetä majakkasanomia, eikä täten käytössä ole myöskään taattuja aikavälejä. Toimintamoodi on käytännössä aikavälitön, joten kaikki noodit suorittavat CCA-toiminnon ainoastaan kerran ja mikäli CCA osoittaa kanavan olevan vapaa, aloittavat noodit saman tien oman lähetystoimintansa. Koordinaattorinoodien on oltava aktiivisessa tilassa koko ajan, mutta tavalliset verkon noodit voivat noudattaa omaa uniaikatauluaan. Laitteet heräävät kahdesta eri syystä. Joko lähettääkseen sanoman keskusnoodille tai ilmoittaakseen, että ne ovat valmiina vastaanottamaan keskusnoodilta lähetettävää dataa. Tämä tapahtuu siten, että laite lähettää data request -sanoman keskusnoodille, joka tämän jälkeen aloittaa datan lähettämisen. Jokainen lähetetty sanoma kuitataan. Tiedonsiirrossa käytetään aikavälitöntä CSMA-CA-menetelmää. [18]

5.3.2 Superkehys

PAN-koordinaattori voi myös varata tietyn osan superkehyksestä yksittäisen anturinoodin käyttöön, mikäli verkon laite sitä pyytää. Näitä GTS-aikavälejä kutsutaan taatuiksi aikaväleiksi (*engl. Guaranteed Time Slot)*. Mikäli käytössä oleva anturiverkkosovelluksen vaatimuksissa on tietty kaistanleveys tai vähimmäisvaatimus sanomavälityksen viiveessä, voidaan taattuja aikavälejä käyttämällä pyrkiä täyttämään nämä vaatimukset.

Superkehystä kontrolloi PAN-koordinaattorinoodi ja sen kesto on ohjelmallisesti määriteltävissä. Superkehyksen kesto määritellään majakkasanomien avulla. Superkehys koostuu 16 aikavälistä ja ensimmäinen aikaväli alkaa PAN-koordinaattorin lähettämän majakkasanoman jälkeen. Kehyksen rakenne on nähtävissä kuvassa 25.



Kuva 25. Superkehyksen rakenne käytettäessä GTS-aikavälejä [18]

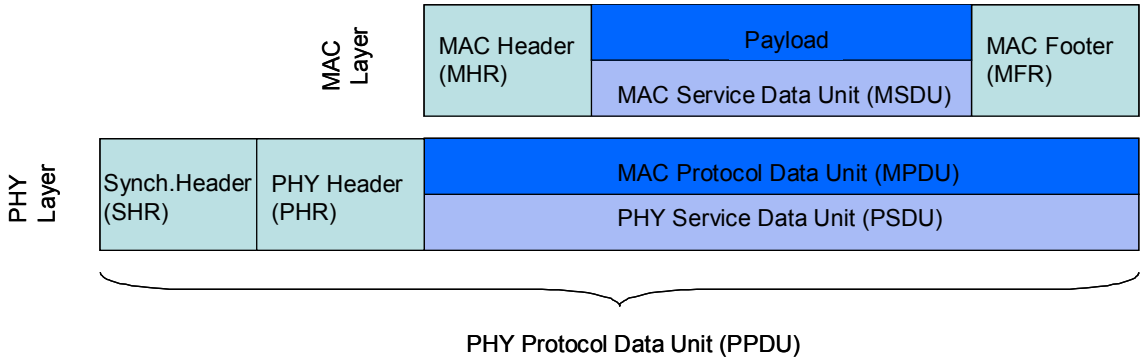
Laitteiden, jotka haluavat kommunikoida PAN-koordinaattorin kanssa, täytyy pyrkiä suorittamaan sanomanvaihto kahden onnistuneen majakkasanoman välillä, kilpailulle vapaan jakson aikana. Tätä aikajaksoa kehyksestä, jolloin kaikki noodit voivat kilpailla käytössä olevista aikaväleistä kutsutaan termillä Contention Access Period (CAP). Koko superkehys koostuu siis neljästä osasta, jotka ovat majakkasanoma, CAP, GTS-aikaväleistä koostuva Contention Free Period (CFP) sekä inaktiivinen periodi. Pystyäkseen kommunikoimaan PAN-koordinaattorin kanssa, täytyy anturinoodin saada kanava käyttöönsä edellisessä kappalessa kuvattua CSMA-CA-metodia käyttämällä.

GTS-aikavälit varataan superkehyksen lopusta ja tätä GTS-aikaväleille varattua osuutta superkehyksestä kutsutaan kilpailusta vapaaksi jaksoksi CFP (*engl. Contention Free Period)*. Majakkasanoma varaa aina superkehyksen ensimmäisen aikavälin.

Superkehys voidaan jakaa myös tilan suhteen kahteen eri osaan: aktiiviseen ja inaktiiviseen osaan. Inaktiivisen osan kohdalla kaikki verkon noodit mukaan lukien PAN-koordinaattori, voivat asettaa vastaanottimensa pois päältä ja siirtyä unitilaan. Kuten GTS-aikavälien käyttö, myös aktiivisen ja inaktiivisen tilan kesto ovat määriteltävissä sovelluksen mukaan. [11],[18]

5.4 Kehysrakenne

Fyysisen ja MAC-kerroksen välille IEEE 802.15.4 -standardi määrittelee neljä eri kehystä: majakka-, data-, kuittaus- ja MAC-komentokehyksen. Näitä kutsutaan nimellä PPDU (*PHY Protocol Data Unit*). Kuvassa 6 on esitetty kerrosten kehysrakenne. Jokainen PPDU koostuu synkronointiotsikosta SHR (*engl. Synchronization Header)*, fyysisen kerroksen otsikosta PHR (engl. *PHY Header)* sekä fyysisen kerroksen data yksiköstä, joka puolestaan koostuu MAC Protocol Data -yksiköstä MPDU (*MAC Protocol Data Unit)*. Poikkeuksena on *Ackknowledgement*-kehys, joka ei sisällä MSDU:ta. MSDU -komponentti sisältää dataa MAC-kerroksen käyttöön: superkehyksen identifikaatiotiedot ja sekvensointitiedot, osoitetiedot ja muita tietoja. Kuvassa 26 on esitetty edellä mainittu kehysrakenne. [11]



Kuva 26. IEEE 802.15.4 -standardin kehysrakenne [11]

Majakkakehyksen lähettäminen on sallittua ainoastaan IEEE 802.15.4 -standardissa määritellylle FFD-laitteelle, riippumatta käytetystä verkkotopologiasta. Majakkasanoma on MAC-kerroksen palvelu, joka välitetään protokollarajapinnan kautta fyysiselle kerrokselle. Majakkasanomaa voidaan käyttää useisiin eri tarkoituksiin. Nämä ovat superkehyksen rajan merkitseminen, kehyksen synkronointi ja yhteyden valvonta.

Superkehyksen rajan merkintään käytettäessä majakkasanoma kertoo ajoitusreferenssin, jotta superkehyksen rajat voitaisiin määritellä. Superkehys sallii kiinteän määrän kehyksiä asetettavaksi majakkasanomien väliin.

Kehyksen synkronointisignaalina majakkasanoma tarjoaa seuraavat palvelut: superkehyksen synkronoinnin tiettyyn ajanhetkeen, törmäysten estämisen, mahdollistaa vastaanottavien noodien nukkumisen silloin kun verkossa ei ole noodille tapahtumia, parantaa viestien latenssiaikaa rajoittamalla viestiliikenteen viiveen korkeintaan yhden superkehyksen mittaiseksi. Synkronisointi mahdollistaa myös tilanteen, jossa majakkasanoman lähettäneestä laitteesta tulee ajoitusreferenssi, ja laskee yksittäisen verkon noodilaitteen tarkkuusvaatimuksia ajoituksen suhteen. Jokaisella verkon noodilla voi näin ollen olla hieman epätarkempi kelloreferenssi, joka periodisesti synkronoidaan vastaamaan majakkalaitteen kelloreferenssiä. [11]

Datakehys on käytettävissä sekä RFD- että FFD-laitteille riippumatta käytettävästä verkkotopologiasta. Datakehyspalvelu antaa ylempien protokollakerrosten käyttöön varsinaisen hyötykuorman (*engl. payload)* lähettämisen ja vastaanottamisen. [11]

Kuten datakehys, on myös kuittauskehys kaikkien verkon laitteiden käytettävissä, riippumatta onko laite rajoitetun toiminnallisuuden, vai täyden toiminnallisuuden laite. Edelleen verkon topologialla ei myöskään ole vaikutusta kehyksen käytettävyyteen. Kehyksen ominaisuudet ovat muutenkin vastaavat kuin datakehyksellä, eli palvelu aikaansaadaan MAC-kerroksella ja lähetetään eteenpäin fyysiselle kerrokselle protokollarajapinnan kautta. Käyttämällä kuittauskehyksiä voidaan MAC:n yläpuolisilla kerroksilla varmistua siitä, että tiedonsiirto linkin yli on onnistunut. [11]

Data- ja kuittauskehyksen tapaan on MAC-komentokehys saatavilla kaikille verkon laitteille riippumatta topologiasta. Ja palvelu aikaansaadaan myös MAC-kerroksella, jolta se rajapinnan kautta välitetään fyysiselle kerrokselle lähetettäväksi fyysistä mediaa pitkin. Kyseinen kehys toimii ensisijaisena valvontatiedon tuottajana MAC-protokollakerrokselle. [11]

5.5 MAC-kerroksen tarjoamat palvelut

Kuten luvussa 5.2 mainittiin, standardin MAC tarjoaa kaksi palveluluokkaa ylemmille kerroksille. Data palveluluokka sisältää kolme eri primitiiviä joita käytetään tiedon välittämiseen. MAC-hallintapalveluluokka puolestaan sisältää helpommin varsinaisiksi

palveluiksi miellettäviä toimintoja. Kuvassa 27 on lueteltu kaikki MAC-hallintapalveluluokan tarjoamat palvelut.

Name	Request	Indication	Response	Confirm
MLME-ASSOCIATE	7.1.3.1	7.1.3.2♦	7.1.3.3♦	7.1.3.4
MLME-DISASSOCIATE	7.1.4.1	7.1.4.2		7.1.4.3
MLME-BEACON-NOTIFY		7.1.5.1		
MLME-GET	7.1.6.1			7.1.6.2
MLME-GTS	7.1.7.1*	7.1.7.3*		7.1.7.2*
MLME-ORPHAN		7.1.8.1♦	7.1.8.2♦	
MLME-RESET	7.1.9.1			7.1.9.2
MLME-RX-ENABLE	7.1.10.1*			7.1.10.2*
MLME-SCAN	7.1.11.1			7.1.11.2
MLME-COMM-STATUS		7.1.12.1		
MLME-SET	7.1.13.1			7.1.13.2
MLME-START	7.1.14.1♦			7.1.14.2♦
MLME-SYNC	7.1.15.1*			
MLME-SYNC-LOSS		7.1.15.2		
MLME-POLL	7.1.16.1			7.1.16.2

Kuva 27. MAC-hallintapalveluprimitiivit [15]

Primitiivien esittelyssä numero viittaa suoraan standardin [15] alalukuun, jossa se käsitellään. Vinoneliö kertoo primitiivin toteutuksen olevan optionaalinen RFD-luokan laitteelle ja asteriski puolestaan kertoo primitiivin toteutuksen olevan optionaalinen sekä FFD- että RFD-luokan laitteille.

Tässä luvussa käydään läpi tarkemmin radiokanavien skannaukseen, verkonmuodostukseen ja synkronointiin liittyviä palveluja ja käytettäviä primitiivejä. Kyseiset palvelut on valittu siitä syystä, että vastaavia asioita on käsitelty myös muiden työssä läpikäytyjen MAC-ratkaisujen osalta.

Standardi sisältää neljä erityyppistä radiokanavan skannausoperaatiota. Energiatason tunnistukseen liittyvä skannaus mittaa radiokanavan osoittaman energiatason jokaisella halutulla loogisella kanavalla. Aktiivisen kanavan skannaus etsii majakkasanomia laitteen kuuntelualueella. Laite lähettää jokaisella loogisella kanavalla *Beacon Request* -komennon, joka puolestaan aiheuttaa sen että jokainen komennon vastaanottanut PAN-koordinaattori lähettää majakkasanoman. Passiivinen kanavan skannaus puolestaan etsii majakkasanomia laitteen kuuntelualueella ilman *Beacon Request* -komennon lähettämistä. Piilotetun laitteen kanavan skannaus mahdollistaa koordinaattoriinsa yhteyden kadottaneen laitteen etsiä kyseisen PAN-verkon majakkasanomia. Tämä skannaus suoritetaan ainoastaan määritellyn kanavalistan perusteella. [11]

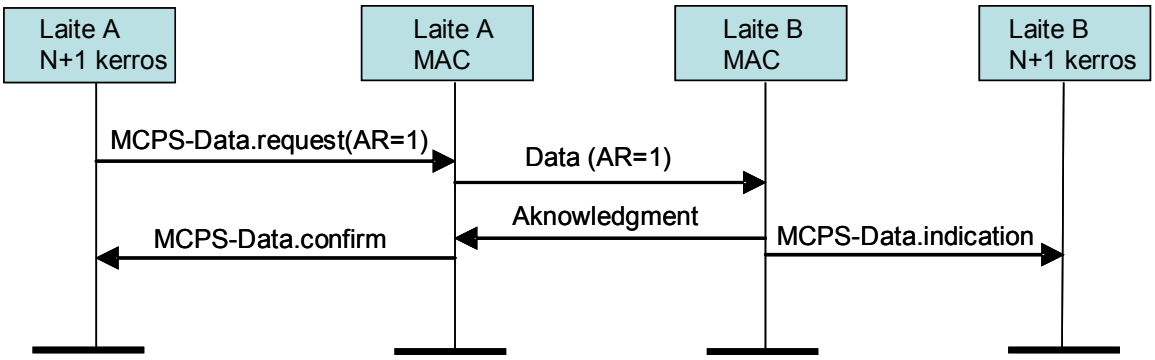
Onnistuneen kanavan skannauksen jälkeen laite voi lähettää liittymispyynnön löytämälleen PAN-koordinaattorille käyttämällä MLME.ASSOCIATE.request-primitiiviä. Tämä pyyntö voidaan tilanteesta riippuen joko hylätä tai hyväksyä käyttämällä MLME-ASSOCIATE.response-vastausta. Verkkoon liittymään pyrkivän laitteen *Assocation Request* -viestiin lähetetään kuittaussanoma PAN-koordinaattorin toimesta. Tämä kuittaussanoma ei siis kerro onko liittyminen verkkoon hyväksytty vai ei. Se ainoastaan vahvistaa sen, että verkkoon liittyminen on tunnistettu ja tähän liittyvä sanoma on vastaanotettu onnistuneesti. Sanoman vastaanottamisen jälkeen PAN-koordinaattorin täytyy päättää onko sillä tarpeeksi resursseja hyväksyäkseen uuden laitteen verkkoon liittymisen. Kyseisen toimenpiteen yhteydessä laite voi myös pyytää 16-bittistä osoitetta normaalin 64-bittisen sijaan. Tämä pienentää pakettien kokoa, ja siten myös virrankulutusta. [11]

Verkosta poistuminen voidaan suorittaa MLME-DISASSOCIATE-primitiivejä käyttämällä. Tämä toimenpide voidaan suorittaa sekä PAN-koordinaattorin että normaalin laitteen toimesta. [11]

Verkon synkronointia ylläpidetään eri tavoin riippuen toimintatilasta. Majakkasanomaa käyttävässä tilassa laitteet synkronoituvat majakkasanoman avulla. Majakkasanomaa käyttämättömässä tilassa puolestaan FFD-laitteet voivat kuitenkin lähettää majakkasanomia, joiden avulla muut verkon laitteet aloittavat tunnistuksen. Mikäli

toimitaan majakkasanomia käyttävässä tilassa, synkronointia verkossa ylläpidetään PAN-koordinaattorin kanssa MLME-SYNC- ja MLME-SYNC-LOSS-primitiivien avulla. Ensin mainitun primitiivin avulla noodi voi etsiä majakkasanomia. Majakkasanomien etsiminen aloitetaan MLME-SYNC.request-sanomaa käyttämällä. Käytännössä tämä tarkoittaa sitä, että radion vastaanotin aktivoidaan ja odotetaan ennalta määrätyn ajan verran PAN-koordinaattorin lähettämää majakkasanomaa. Majakkasanomia käyttävässä verkossa majakkasanomien etsintä voidaan suorittaa kahdella eri tavalla. Noodi voi tarkkailla jatkuvasti PAN-koordinaattorin lähettämiä majakkasanomia tai tämä voidaan tehdä ainoastaan kerran. Jatkuvaa tarkkailua käytettäessä noodi vastaanottaa ensimmäisen majakkasanoman, jonka perusteella se saa tiedon superkehyksen rakenteesta ja tietää myös milloin sen täytyy herätä vastaanottaakseen seuraavan majakkasanoman. Lähettääkseen dataa laite aktivoituu hieman ennen majakkasanoman saapumista. Mikäli käytössä on yhteen majakkasanoman vastaanottamiseen perustuva toimintatila, laite aktivoituu ja kuuntelee ennakoidun ajanjakson löytääkseen majakkasanoman ja voidakseen aloittaa lähetyksen. [11],[19]

Mikäli verkossa käytetään *beaconless*-toimintatilaa, MLME-SYNC.request-tapahtuma aiheuttaa sen, että laite generoi datapyyntösanoman pollatakseen PAN-koordinaattoria MLME-POLL-primitiivin avulla. PAN-koordinaattori vastaa kyseiseen sanomaan kuittausviestillä, joka samalla indikoi onko koordinaattorilla dataa lähetettävänä kyseiselle noodille. Kuittaussanoman avulla laite voi ylläpitää synkronointiaan. Yksinkertaistettuna synkronointi toteutetaan majakkasanomien avulla, jos verkko on *beacon*-tilassa. Mikäli verkko on *beaconless*-tilassa, synkronointi toteutetaan pollaamalla PAN-koordinaattoria. [11]



Kuva 28. Onnistunut datansiirto kuittaussanomaa käyttäen [14]

Varsinaiseen datan siirtoon standardissa on määritelty kolme eri MAC-kerroksen primitiiviä. Nämä ovat request, confirm ja indication. Kuvassa 28 on esitetty primitiivien nimet sisältävä signalointikaavio onnistuneesta sanomanvaihdosta datansiirtotapahtumaan liittyen. Kyseisessä skenariossa on käytetty optionaalista sanoman kuittausta. Mikäli kuittaussanoma ei ole käytössä, laite A lähettää *confirm*-sanoman ylemmälle kerrokselle välittömästi kun se on siirtänyt datapaketin laitteelle B.

Standardi tarjoaa myös tietoturvaan liittyviä palveluja. MAC-kerros voi toimia kolmessa eri tilassa tietoturvan kannalta katsottuna. Nämä ovat turvaamaton tila, oikeuslistan kontrollointi ACL-tila (*engl. Access List Control mode)* sekä turvattu tila. Turvaamaton tila on nimensä mukaisesti täysin salaamaton ja tarkoitettu sovelluksiin, joissa tietoturvalle ei ole tarvetta. ACL-tilassa ylläpidetään osoitelistaa noodeista, joiden kanssa kommunikointi on sallittua. Vastaanotettu viesti lähetetään ylemmille kerroksille yhdistettynä tietoon, oliko lähettävä laite sallittujen noodien listalla vai ei. Tämän jälkeen ylempien kerrosten tehtävänä on päättää hylätäänkö vastaanotettu tieto vai ei.

Turvatussa toimintatilassa laite voi tarjota palveluina edellä käsiteltyä ACL-pääsylistan käyttöä, viestien salausta, eheyden varmistamista sekä viestin tuoreuden varmentamista. Sanomien salaus toteutetaan käyttämällä AES-salausalgoritmia (*Advanced Encryption Standard)*. Viestin eheyden varmistaminen on palvelu, jolla voidaan varmistua siitä, että vastaanotettua viestiä ei ole muokattu sellaisen tahon toimesta, jolla tähän ei ole oikeutta.

Tämä suoritetaan lisäämällä sanomaan erityinen MIC-koodi (*engl. Message Integrity Code)*. Viestin tuoreuden varmistamisella pyritään välttämään tilanne, jossa verkossa vaihdettuja sanomia voidaan käyttää niin sanottuihin palvelunestohyökkäyksiin (*engl. Denial Of Service, DOS)* tallentamalla niitä ja lähettämällä samaa viestiä monia kertoja uudelleen. Sanomiin voidaan lisätä järjestysnumero ja vastaanotettaessa viestiä verrataan kyseistä numeroa edelliseen. Jos järjestysnumero on suurempi, voidaan uusi sanoma hyväksyä. [11]

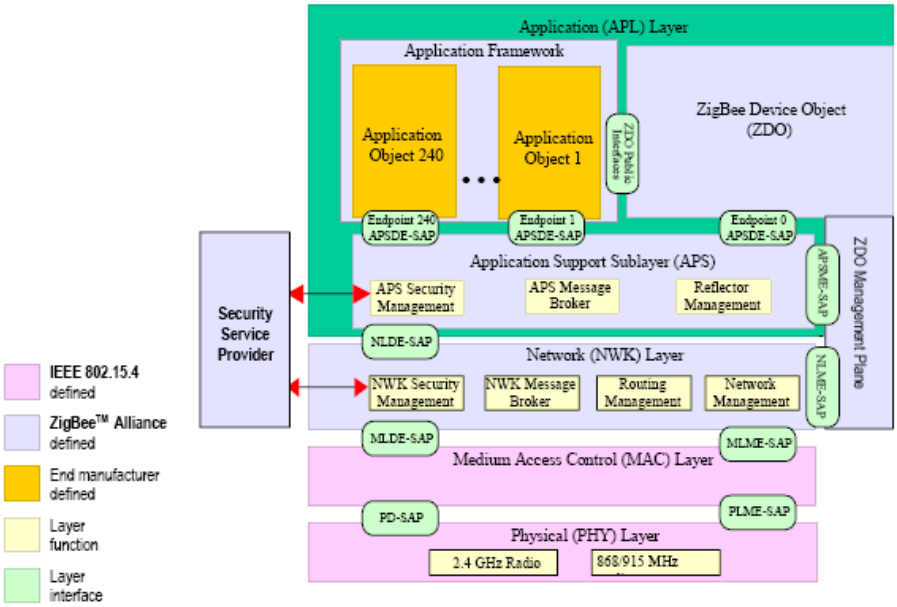
Käytettävät turvallisuuspalvelut voidaan valita standardin tarjoamista valikoimista (*engl. Security Suite)*. Valikoimia on kolmea perustyyppiä, AES-CTR tarjoaa pääsylistan käytön ja salauksen, mutta ei viestin eheyden varmistamista. Tuoreden vamistus on tässä moodissa optionaalinen. AES-CCM-moodit sisältävät pääsylistojen käytön, salauksen, eheyden tarkistuksen sekä optionaalisen tuoreuden tarkistuksen. AES-CBC-MAC-moodit tarjoavat pääsylistan käytön ja eheyden varmistuksen. AES-CCM- ja AES-CBC-MAC-moodeissa on vaihtoehtoina joko 32-, 64- tai 128-bittiseen salaukseen perustuvat vaihtoehdot. Vähimmäisvaatimuksena turvatun tilan käyttöön standardissa on määritelty AES-CCM-moodin käyttö 64-bittisessä tilassa. [11]

5.6 ZigBee ja 6LoWPAN

Puhuttaessa langattomista anturiverkoista tai edullisista ja yksinkertaisista anturilaitteista mainitaan usein nimi ZigBee. Hyvin usein myös törmää virheelliseen kuvitelmaan, että IEEE 802.15.4 -standardi ja ZigBee ovat yksi ja sama asia. ZigBee on kaupallisten yritysten muodostama ryhmittymä, jonka tarkoituksena on kehittää luotettavia, edullisia, vähän virtaa käyttäviä tuotteita, jotka ovat langattomasti yhteydessä toisiinsa ja joita käytetään havainnointiin ja kontrollointiin. [40]

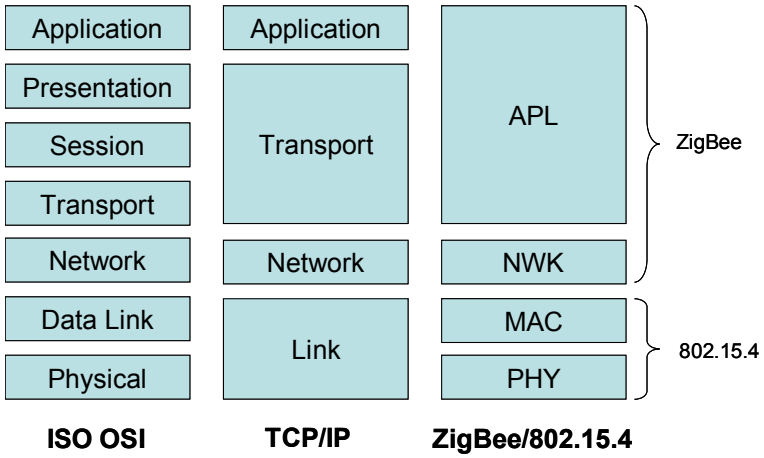
Sovellusalueita, joissa ZigBee-ryhmittymän mukaisia tuotteita tullaan näkemään, ovat kuluttajaelektroniikassa, koti- ja rakennusautomaatiossa, teollisten sovellusten ohjaamisessa, PC-laitteissa, lääketieteellisessä käytössä olevissa anturisovelluksissa sekä leluissa ja peleissä. On arvioitu, että vuoden 2009 loppuun mennessä standardin mukaisten toimitettujen radiolaitteiden määrä olisi jopa 750 miljoonaa kappaletta [13].

ZigBee-ryhmittymän protokollapino rakentuu kuitenkin IEEE 802.15.4 -standardin päälle. ZigBee määrittelee verkko- ja sovelluskerroksen mukaiset protokollakerrokset ja pyrkii mahdollistamaan helpon sovelluskehityksen anturiverkkosovellusalueelle tarjoamalla valmiin sovellusrajapinnan. Kuvassa 29 on esitetty ZigBee-protokollapinon yksityiskohtainen rakenne.



Kuva 29. ZigBee protokollapinon rakenne [40]

ZigBee ei kuitenkaan ole itsessään yhteensopiva nykyään jatkuvasti merkitystään lisäävän TCP/IP-protokollapinon kanssa [33]. Kuvassa 30 on esitetty OSI-mallin, TCP/IP-pinon ja ZigBee/IEEE 802.15.4 - yhdistelmän protokollakerrosten vastaavuus.



Kuva 30. Protokollapinojen vastaavuus

Suoraviivaisesti ajateltuna IPv6-pino voitaisiin ottaa suoraan käyttöön IEEE 802.15.4 MAC-kerroksen päälle, mutta tällöin törmätään ongelmiin standardien erikokoisten pakettikokojen kanssa. IEEE 802.15.4 -standardin mukaan fyysisen kerroksen palvelu datayksikön PSDU maksimikoko on 127 tavua. Mikäli käytössä on standardin tarjoama turvallisuuspalvelu, jää IPv6-datalle ainoastaan 81 tavua jäljelle. Tämä on hyvin vähän, sillä pelkästään otsikkotiedot IPv6-paketille varaavat 40 tavua. [33]

Kyseiseen puutteeseen pyrkii vastaamaan 6LoWPAN (*IPv6 over Low power WPAN*)-teknologia. Tämä on Internet Engingeering Task Force -työryhmän kehityksen alla oleva standardi, josta on kirjoitushetkellä saatavilla ehdotelmatasoinen versio [17]. Kyseisessä ratkaisussa IEEE 802.15.4 MAC-kerroksen päälle on rakennettu adaptaatiokerros, joka muuntaa IEEE 802.15.4 pakettikehystyksen IPv6-protokollan mukaiseksi ja päinvastoin. Adaptaatiokerrosta käytetään pilkkomaan IPv6 paketteja useampiin IEEE 802.15.4 kehyksiin lähetystä varten. Tällöin ZigBee-pinon verkkokerros on poistettu kokonaan käytöstä ja korvattu 6LoWPAN adaptaatiokerroksella. [33]

5.7 Arviointi

Tässä kappaleessa pyritään arvioimaan IEEE 802.15.4 MAC:ä jokaisen luvussa 4 esitetyn antuiverkoille tarkoitetun MAC-protokollan ominaisuuden suhteen. Lopuksi on myös esitetty johtopäätöksiä protokollan soveltuvuudesta anturiverkoille.

Käytettäessä majakkaviestejä tukevaa tilaa ja superkehyksen GTS-aikavälejä, voidaan törmäysten määrän olettaa olevan minimaalinen johtuen käytössä olevasta TDMA-

kanavanvaraustekniikasta. CSMA-CA-menetelmää käytettäessä törmäysten määrä luonnollisesti lisääntyy. Hidden terminal -ongelmaa vastaan protokollassa ei ole minkäänlaista suojausta, esimerkiksi RTS-CTS-kättelymekanismia. Törmäysten määrän vähentämiseksi käytetään satunnaisesti valittua lähetysajankohtaa.

Majakkasanomaa käytettäessä tavalliset RFD-laitteet voivat siirtyä unitilaan kilpailujakson aikana. Kuitenkin koordinaattorinoodien täytyy olla aina aktiivisena kyseisen jakson aikana. Ilman majakkasanomaa toimittaessa koordinaattorinoodin on pysyttävä aktiivisessa tilassa jatkuvasti. Mikäli kyseisessä roolissa toimivan FFD-noodin energiansaanti on ainoastaan pariston varassa, on selvää että sen elinikä on huomattavan lyhyt. Tämä rajoite voidaan tosin kiertää toteuttamalla vaadittava noodien synkronointi MAC-kerroksen yläpuolelle. Tähän standardi tarjoaa riittävät palvelut MAC-kerroksella. Tässä toimintatilassa energiaa hukkaantuu myös siitä syystä, että noodin on pysyttävä aktiivisessa tilassa Data Request -sanoman lähettämisestä aina siihen saakka, kunnes sanoma on vastaanotettu. Eräs energiaa tuhlaava seikka on se, että GTS-aikavälillä suoritettavan lähetyksen aikana noodin täytyy odottaa koko aikavälin kesto riippumatta lähetettävän datapaketin koosta. GTS-aikavälejä käyttävän noodin täytyy myös herätä joka kerta CFP-periodin aikana. Tämä aiheuttaa turhaa energiankulutusta, mikäli siirrettävää dataa ei kyseisen kehyksen aikana ole [19]. Myös majakkasanomaa käyttävien synkronointitilojen välillä on eroja energiankulutuksen suhteen. Käytettäessä periodista majakkasanoman avulla tapahtuvaa synkronointia noodi voi siirtyä unitilaan, koska se tietää milloin herätä seuraavaa majakkasanomaa varten. Kertaluonteiseen majakkasanoman vastaanottoon perustuvassa synkronoinnissa noodi voi säästää energiaa, koska sen ei tarvitse herätä vastaanottamaan jokaisen kehyksen majakkasanomaa. Kun noodi haluaa lähettää, sen täytyy siirtyä aktiiviseen tilaan välittömästi odottaakseen seuraavaa majakkasanomaa. Mikäli inaktiivinen periodi on määritelty hyvin suureksi, voi energiaa kulua huomattava määrä, mikäli majakkasanoma on lähetetty juuri ennen noodin heräämistä. Lun tulosten mukaan jatkuvaa majakkasanoman seurantaa käyttävä toimintatila on energiatehokkaampi. [19]

Verkon skaalautuvuuden suhteen eräs rajoittava tekijä on noodien tyyppi ja sijainti verkossa. Ainoastaan FFD-noodit voivat edelleen lähettää sanomia, joten verkon

topologiassa RFD-noodi voi toimia ainoastaan lehtinoodina. Majakkasanomaa käytettäessä topologia rajoittuu ainoastaan tähtiverkkoon. Kyseisen topologian perusongelmana on sen joustamattomuus. Noodeja ei voida viedä kovin kauaksi keskusnoodista. Standardi kuitenkin mahdollistaa monimutkaisempien verkkotopologioiden käytön toteuttamalla vaadittavat toiminnot MAC-kerroksen yläpuolelle, esimerkiksi klusteroitujen hierarkkisten verkkojen muodostamisen.

Tilanteessa, jossa käytetään majakkaviestejä ja useat noodit lähettävät tiheään hyvin pieniä paketteja, jää kanavan käyttöaste matalaksi. Jokainen CAP-jakson aikaväli käytetään vain osittain ja näin aikavälin hyötyaste ja koko siirtotien käyttöaste jää pieneksi. Tällä ominaisuudella ei tosin anturiverkoissa ole yleensä kovin suurta painoarvoa. Jos käytetään GTS-aikavälejä ja näitä omaavalla noodilla ei ole liikennettä kehyksen aikana, käytetään kyseiset aikavälit turhaan.

Asettamalla useita GTS-aikavälejä noodin käyttöön, jolla on paljon lähetettävää dataa, voidaan oletettavasti viivettä pienentää verrattuna tilanteeseen, että noodi joutuisi kilpailemaan jokaisesta yksittäisestä aikavälistä. Havainto on todettu paikkansapitäväksi myös standardin MAC-ratkaisua käsittelevässä suorituskykyanalyysissä. [19]

Majakkasanomaa käytettäessä noodi voi joutua odottamaan CFP-periodin sekä inaktiivisen periodin päättymiseen saakka, kunnes se voi aloittaa datansiirron yrittämisen. Aikavälitöntä CSMA-CA-toimintatilaa käytettäessä lähetys voidaan aloittaa välittömästi, kun kanavan on havaittu olevan vapaa.

Protokollan teoreettiset siirtonopeudet esitettiin kappaleessa 3.3. Todellisessa käytössä saavutettavat lukemat riippuvat käytettävästä topologiasta, toimintamoodista sekä verkon kilpailutilanteesta. Siirtonopeuteen vaikuttaa myös kulloisenkin käytössä olevan kehyksen pituus, johon puolestaan vaikuttaa esimerkiksi osoitteiston pituus. Pyrittäessä säästämään energiankulutusta alhaisen *duty cycle* -arvon avulla, lisätään samalla kuitenkin viivettä ja rajoitetaan kaistanleveyttä [19]. Toinen seikka, jolla on vaikutusta siirtonopeuteen, on käytettävä turvallisuusluokka. Kehyksen eheystarkistus sekä tuoreusvarmistus lisäävät siirrettävien oktettien määrää jokaisen lähetettävän kehyksen yhteydessä. Kehykset, jotka ovat hyvin pieniä, lisäävät tällöin myös virrankulutuksen suhdetta huomattavasti [11].

Noodeilla on tasavertainen pääsy mediaan CAP-jakson aikana. Tarvittaessa noodit voivat saada käyttöönsä GTS-aikavälejä, jolloin niillä on luonnollisesti taattu mahdollisuus saada enemmän resursseja käyttöönsä siirtotiellä. Tällä pystytään vaikuttamaan hyvin voimakkaasti siirtonopeuteen. Muuta mekanismia protokolla ei tarjoa enemmän sanomia lähettävien noodien käyttöön.

Majakkasanomaa käyttävä tila voi olla käytössä ainoastaan tähtiverkkotopologiaa käytettäessä. Tilan etuihin kuuluu muun muassa se, että koordinaattorinoodi voi kommunikoida muiden noodien kanssa välittömästi niin halutessaan. Huono puoli majakkasanomaa käyttävässä moodissa on kuitenkin se, että noodien täytyy herätä säännöllisesti vastaanottamaan majakkasanoma. [29]

Majakkasanomaa käyttämättömässä tilassa noodit voivat lähettää vapaasti sanomia koordinaattorinoodille käyttämällä CSMA-CA-protokollaa. Datan vastaanottaminen on puolestaan hieman mutkikkaampaa. Mikäli noodi haluaa vastaanottaa keskusnoodilta sanomia, sen täytyy herätä lepotilasta ja kysyä data request -sanomaa käyttäen, onko keskusnoodilla sille lähetettävää dataa. Noodien ei kuitenkaan tarvitse herätä säännöllisesti vastaanottamaan majakkasanomaa. Energiankulutusta ajatellen tämä on etu verrattuna majakkasanomaa käyttävään tilaan. Tämä täytyy arvioida tavoitellun verkon eliniän ja odotettavan datansiirtomäärän suhteen. [29]

IEEE 802.15.4 MAC-ratkaisussa on hieman samankaltainen hybridiluonne kuin Z-MAC-protokollassa. Kummassakin on mahdollisuus käyttää myös TDMA-menetelmää. Kuitenkin Z-MAC on dynaamisempi ja kykenee päättelemään milloin on parempi käyttää TDMA-metodia, milloin taas on CSMA-CA-metodia. Protokollaratkaisussa voidaan tilaa myös vaihtaa joustavasti tilanteen mukaan. GTS-aikavälien käyttö asettaa myös haasteita energiankulutuksen suhteen. Noodin täytyy pysyä hereillä koko GTS-aikavälin ajan, riippumatta siitä mikä on varsinaisen datapaketin kesto. Mikäli GTS-aikaväli on arvioitu liian suureksi, kuluu energiaa hukkaan turhan kuuntelun muodossa [29].

IEEE 802.15.4 MAC-protokolla soveltunee parhaiten tietyille sovellusalueille, esimerkiksi kotiautomaatioon ja muihin tilanteisiin, joissa noodien liikkuvuus ei ole suurta. Koordinaattorinoodien virrankulutus asettaa topologiamuutoksille suuria haasteita, sillä

kyseisen FFD-laitteen virrankulutus on väistämättä suuri. RFD-laitteiden kyvyttömyys välittää liikennettä eteenpäin voidaan nähdä toisena rajoitteena topologiamuutosten suhteen. Mikäli koordinaattorina toimiva FFD-noodi vikaantuisi, saattaa olla että osa RFD-noodeista jäisi verkon ulkopuolelle. Tämä täytyy ottaa huomioon MAC-kerroksen yläpuolella, jotta laitteet voivat itse valita toimintamoodinsa verkossa.

Energiankulutuksen suhteen ratkaisu tuskin kykenee kilpailemaan uudempien MAC-ratkaisujen kanssa. Valitettavasti kirjoitushetkellä aiheeseen liittyviä vertailutuloksia IEEE 802.15.4 ja muiden työssä esiteltyjen MAC-ratkaisujen välillä ei ollut saatavilla. Työssä käytetty Lu'n suorituskykyanalyysi on myös suoritettu ainoastaan majakkasanomaa käyttävässä tilassa.

6 Yhteenveto

Merkittävin alue, jossa energiaa voidaan säästää, on turhan kuuntelun minimoiminen. Kuitenkin myös protokollan mukautuminen erilaisiin verkon kuormitustilanteisiin on varsin hyödyllistä energiankäytön optimoinnin suhteen. Jotkin työssä esitellyistä MAC-ratkaisuista suoriutuvat paremmin tietyntyyppisissä liikennemäärä- ja verkkorakenneskenaarioissa. Optimaalisen MAC-ratkaisun valinta riippuneekin siitä, kuinka hyvin verkon toiminta on kyetty analysoimaan ennen MAC-ratkaisun valintaa. Kaikkia MAC-ratkaisuja ei ole testattu oikeassa anturiverkossa lainkaan ja tämän lisäksi ei ole saatavilla mittaustuloksia, joissa kaikkia ratkaisuja olisi simuloitu vastaavilla verkkorakenteilla ja liikennekuormilla.

IEEE 802.15.4 -protokollaa vasten suoritettuja mittaustuloksia ei yksikään protokollaratkaisun raportti tarjonnut. Oletettavaa kuitenkin on, että perinteinen CSMA-CA-ratkaisu ei kykene kilpailemaan uudempien protokollaratkaisujen kanssa energiankulutuksen suhteen. Z-MAC:in tapaan standardin MAC-ratkaisu tarjoaa myös TDMA-tyyppisen ratkaisun superkehystoimintaa käytettäessä, mutta tämä on mahdollista ainoastaan tähtiverkkotopologiassa. Kuten aiemmin on mainittu, tästä rajoituksesta voidaan kuitenkin päästä eroon toteuttamalla vaaditun topologian tuki MAC-kerroksen yläpuolelle. Tämä toimintamoodi ei kuitenkaan kykene mukautumaan vaihteleviin liikennemääriin yhtä dynaamisesti kuin Z-MAC. Liikennemäärään sopeutuvaa toimintaa varten voitaneen kuitenkin toteuttaa MAC:in yläpuolisille kerroksille vaadittavia ratkaisuja.

Z-MAC näyttäisi tuottavan parhaat tulokset silloin, kun liikenne verkossa on hyvin aktiivista. Alemmilla kuormilla se ei aivan yllä B-MAC:in tasolle. Täten olisi hyvin tärkeää ymmärtää käyttöönotettavan verkon toimintaa ja pyrkiä ennakoimaan tyypillisin tilanne, jossa noodit joutuvat toimimaan. Z-MAC pystyy yhdistämään TDMA- ja kilpailuperusteisen kanavanvarauksen hyvät puolet. TDMA-protokollien yleisen ongelman, kellosynkronoinnin ylläpidon, kyseinen ratkaisu toteuttaa protokollatasolla.

Esitellyistä ratkaisuista AI-LMAC on selkeästi suunniteltu tietyntyyppisten anturiverkkosovellusten käyttöön ja on hyvin vaikea arvioida, miten se soveltuisi täysin

toisentyyppiseen sovellusympäristöön. Kyseisen MAC-ratkaisun suunnittelun lähtökohtana on ollut ympäristöntarkkailuun tarkoitettu anturiverkkosovellus.

Tässä työssä on keskitytty enimmäkseen energiankulutuksen minimointiin MAC-ratkaisujen avulla. Joissakin sovelluksissa saattavat tietyt muut osa-alueet olla tärkeitä, esimerkiksi viiveen minimointi tai siirtonopeuden maksimointi verkossa. Tällöin täytyy myös MAC-ratkaisun valinnassa pyrkiä kompromissiin, jossa tärkeimmiksi nähdyt kriteerit pystytään täyttämään mahdollisimman hyvällä energiansäästötasolla.

Vaikka jokin tietty protokolla saattaa olla hyvin energiatehokas, on muiden tekijöiden, kuten muistinkulutuksen, toteutuksen kompleksisuuden sekä protokollan vaatiman prosessoinnin vaikutuksia mahdoton arvioida kokonaisuuden kannalta. MAC-protokollaa valittaessa ja mahdollisesti toteutettaessa, on nämäkin tekijät toki otettava huomioon.

Energiansäästön avaimina MAC-kerroksen protokollaratkaisussa ovat turhan kuuntelun minimointi sekä mahdollisimman joustava toiminta kulloisenkin sovelluksen ja liikennetilanteen mukaan. Z-MAC-protokollan käyttämä dynaaminen kanavansaantimetodin valinta liikennemäärän mukaan lienee selkeä vahvuus.

Ainoana standardina anturiverkoille olemassa oleva IEEE 802.15.4 MAC-ratkaisu soveltunee sellaisenaan parhaiten yksinkertaisille sovelluksille käytettynä, varsinkin mikäli noodien sijainti tiedetään etukäteen hyvin ja topologiamuutokset ovat vähäisiä. Standardin etuihin kuuluu kiistämättä kuitenkin se tosiseikka, että vaikka itse MAC-tasolla tiettyjä ominaisuuksia ei ole tuettu, pystytään MAC-kerroksen tarjoamien palveluiden avulla toteuttamaan ominaisuuksia yläpuolisiin kerroksiin. Yleensä ottaen standardien etuihin kuuluu myös lähes poikkeuksetta itse standardistatuksen aiheuttama momentti. Standardin päälle toteutettaneen hyvin paljon erilaisia palveluita, kuten voidaan jo todeta ZigBee ja IP-protokolla-adaptaatioista.

Mielenkiintoinen tutkimusalue on myöskin AI-LMAC-protokollan suunnittelussa nähty suuntautuneisuus tietyn sovellusalueen suuntaan. MAC-protokollaa parametrisoimalla suorituskyky voitaisiin pyrkiä virittämään, ei ainoastaan sovellusalueen mukaan, vaan jopa yksittäisen käyttökohteen omaispiirteisiin mahdollisimman hyvin mukautuen.

Lähteet

[1] Abd-El-Barr M. I., Youssef M.A.M., Al-Otaibi M.M., *Wireless Sensor Networks - Part I: Topology and Design Issues*, IEEE Canadian Conference on Electrical and Computer Engineering, toukokuu 2005, sivut 1165-1168.

[2] Akyildiz I.F., Su W., Sankarasubramaniam Y. ja Cayirci E., *Wireless Sensor networks: a survey*, Computer Networks, vol. 38, no 12, maaliskuu 2002, sivut 393-422.

[3] Akyildiz I.F., Weilian S., Sankarasubramaniam Y. ja Cayirci E., *A survey on sensor networks*, IEEE Communications magazine, vol 40, no 8, elokuu 2002, sivut 102-114.

[4] Anttila Aki, ”TCP/IP tekniikka”, Helsinki Media, Juva, 2000

[5] Chatterjea S., van Hoesel L.F.W., Havinga P.J.M., *AI-LMAC: An Adaptive, Information-centric and Lightweight MAC protocol for Wireless Sensor Networks*, Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. joulukuu 2004, sivut 381-388.

[6] Chong C-Y ja Kumar S.P., *Sensor Networks: Evolution, Opportunities, and Challenges*, Proceedings of the IEEE, vol. 91, no. 8, elokuu 2003, sivut 1247-1256.

[7] Chydenius Instituutti, tietoliikennelaboratorio, *Ad hoc ja sensoriverkot*, saatavilla WWW-muodossa <http://rf.chydenius.fi/seminaari/ADHOC_JA _SENSORIVERKOT.ppt>, viitattu 10.08.2007.

[8] Cook D.J., Das S.K. ja Wiley J., *Wireless Sensor Networks*, Smart Enviroments: Technologies, Protocols, and Applications, New York 2004.

[9] Crossbow Berkeley Mote MICA2 tekniset tiedot. Saatavilla WWW - muodossa <http://www.xbow.com/Products/productdetails.aspx?sid=174>, viitattu 10.08.2007.

[10] Goldsmith A.J. ja Wicker S.B, *Design challenges for energy-constrained ad hoc wireless networks*, IEEE Wireless Communications, vol. 9, no 4, elokuu 2002, sivut 8-27.

[11] Gutiérrez J., Callaway E.H. ja Barrett R., “Low-Rate Wireless Personal Area Networks, Enabling Wireless Sensors with IEEE 802.15.4” IEEE Standards Wireless Network Series, New York 2003.

[12] Gutierrez J.A., Naeve M., Callaway E., Bourgeois M., Mitter V., ja Heile B., *IEEE 802.15.4: A Developing Standard for Low-Power Low-Cost Wireless Personal Area Networks,* IEEE Network, Vol. 15, No. 5, syyskuu/lokakuu 2001, sivut 12-19.

[13] Hirvonen J., Sallinen M., Maula H., Suojanen M., *Sensor Networks Roadmap*, Saatavilla WWW-muodossa <URL: http://www.vtt.fi/inf/pdf/tiedotteet/2007/T2381.pdf>, VTT Tiedotteita - Research Notes 2381, Espoo 2007

[14] IEEE Std 802.15.4-2006. Saatavilla WWW-muodossa <http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>. Viitattu 02.06.2007

[15] IEEE Std 802.16-2004. Saatavilla WWW-muodossa

<http://standards.ieee.org/getieee802/802.16.html> viitattu 4.1.2008.

[16] IEEE, Std 802, *IEEE Standards for Local and Metropolitan Networks*, saatavilla WWW-muodossa < http://www.ieee802.org/> viitattu 5.1.2008.

[17] IETF 6LoWPAN draft. Saatavilla WWW-muodossa <http://www3.ietf.org/html.charters/6lowpan-charter.html> viitattu 26.01.2008

[18] Karl H. ja Willig A., ”Protocols and Architectures for Wireless Sensor Networks“, John Wiley et. Sons Ltd, West Sussex 2005.

[19] Lu G., Krishnamachari B., Raghavendra C., *Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks*, IEEE International Conference on Performance, Computing, and Communications, 2004.

[20] Noury, Herve N., Rialle T., Virone V., Mercier G., Morey E., Moro G., Porcheron A., *Monitoring behavior in home using a smart fall sensor and position sensors,*

Microtechnologies in Medicine and Biology, 1st Annual International, lokakuu 2000, sivut 607-610.

[21] Polastre J., Hill J., Culler D., *Versatile Low Power Media Access for Wireless Sensor Networks,* Proceedings of the 2nd international conference on Embedded networked sensor systems, marraskuu 2004, sivut 95-107.

[22] Postari J., *Reititys langattomassa anturiverkossa*, Pro Gradu -tutkielma, Jyväskylän Yliopisto, Kokkolan Yliopistokeskus Chydenius-Instituutti 2006.

[23] Rhee I., Warrier A.C., Aja M., Min J., *Z-MAC: a Hybrid MAC for Wireless Sensor Networks,* Proceedings of the 3rd international conference on Embedded networked sensor systems, marraskuu 2005, sivut 90-101.

[24] Rhee I., Warrier A.C., Xu L., *Randomized Dining Philosophers to TDMA Scheduling in Wireless Sensor Networks,* Technical Report TR-2005-21, Department of Computer Science, North Carolina State University, huhtikuu 2004.

[25] Sankarasubramaniam Y., Akan Ö.B. ja Akyildiz I.F., *Event -to-Sink Reliable Transport in Wireless Sensor Networks*, Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, kesäkuu 2003, sivut 177-188.

[26] Shrestha A. ja Xing L., *A Performance Comparison of Different Topologies for Wireless Sensor Networks*, IEEE Conference on Technologies for Homeland Security, toukokuu 2007, sivut 280-285.

[27] Singh S. ja Raghavendra C.S., *PAMAS - Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks,* ACM Sigcomm Computer Communication Review, heinäkuu 1998.

[28] Srivastava M., Muntz R. ja Potkonjak M., *Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments*, Proceedings of the 7th annual international conference on Mobile computing and networking, 2001, sivut 132-138.

[29] Timmons N.F. ja Scanlon W.G., *Analysis of the Performance of IEEE 802.15.4 for Medical Sensor Body Area Networking*, IEEE SECON Sensor and Ad Hoc Communications and Networks, lokakuu 2004, sivut 16-24.

[30] van Dam T. ja Langendoen K., *An adaptive energy-efficient MAC protocol for wireless Sensor networks,* Proceedings of the 1st international conference on Embedded networked sensor systems, marraskuu 2003, sivut 171-180.

[31] van Hoesel L.F.W. ja Havinga P.J.M., *A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches,* Proceedings of First International Conference on Networked Sensing Systems, lokakuu 2004.

[32] Wang Q., Shin W., Liu X., Zheng Z., Oh C., Bedoor K. AlShebli, Caccamo M., Carl A. G., Gunter E., Hou J., Karahalios K., ja Lui S., *I-Living: An Open System Architecture for Assisted Living*, IEEE International Conference on Systems, Man, and Cybernetics (SMC), lokakuu 2006.

[33] Wang R-C., Chang R-S. ja Chao H-C, *Internetworking Between ZigBee/802.15.4 and IPv6/802.3 Network*, SIGCOMM Workshop "IPv6 and the Future of the Internet", elokuu 2007.

[34] Warneke B., Last M., Liebowitz B. ja Pister K.S.J., *Smart Dust: communicating with a cubic-millimeter computer*, IEEE Computer Society, tammikuu 2001, sivut 44-51.

[35] Werner-Allen G., Lorincz K., Welsh M., Marcillo O., Johnson J., Ruiz M. ja Lees J., *Deploying a wireless sensor network on an active volcano*, IEEE Internet Computing, vol. 10, no. 2, maaliskuu/huhtikuu 2006, sivut 18-25.

[36] Wikström M, Jyväskylän Yliopisto, *luentomateriaali: Tietoliikenneprotokollat*, Saatavilla WWW-muodossa <http://www.mit.jyu.fi/wikstrom/opetus/tiea322/luennot/tiea322-2.pdf>. Viitattu 10.08.2007

[37] Wood A.D. ja Stankovic J.A., *Denial of Service in Sensor Networks*, IEEE Computer Society, lokakuu 2002, sivut 54-62.

[38] Ye W. ja Heidemann J., *Medium Access Control in Wireless Sensor Networks,* Technical Report ISI-TR-580, USC/Information Sciences Institute, lokakuu 2003.

[39] Ye W., Heidemann J., Estrin D., *Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks*, IEEE/ACM Transactions on Networking (TON), kesäkuu 2004, sivut 493-506.

[40] ZigBee Home Page, saatavilla WWW-muodossa <http://www.zigbee.org> , ZigBee Alliance.

[41] Zimmermann H., *OSI Reference Model--The ISO Model of Architecture for Open Systems Interconnection,* IEEE Transactions on Communications, Vol. 28, No. 4, huhtikuu 1980, sivut 425-432.