Jarmo Parkkila

Thread kotiautomaatiossa

Tietotekniikan
pro gradu -tutkielma
7. kesäkuuta 2024

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Kokkolan yliopistokeskus Chydenius

Tekijä: Jarmo Parkkila
Yhteystiedot: jarmo.h.t.parkkila@student.jyu.fi
Puhelinnumero: -
Ohjaaja: Ismo Hakala
Työn nimi: Thread kotiautomaatiossa
Title in English: Thread in Home Automation
Työ: Tietotekniikan pro gradu -tutkielma
Sivumäärä: 82
Tiivistelmä: IoT:n nopean yleistymisen myötä yhä useammassa kodissa hyödynnetään tänä päivänä kotiautomaatiota. Kotiautomaatiolla pyritään yleensä helpottamaan asumista, optimoimaan kodin resurssien kulutusta, lisäämään asumismukavuutta sekä parantamaan kodin turvallisuutta ja sillä tavoitellaan niin taloudellisia kuin ajallisiakin säästöjä. Kotiautomaatioon hyödyntämiseen liittyy kuitenkin merkittävä teknologinen haaste, sillä laitteet ja sovellukset eri valmistajilta ovat harvoin yhteensopivia keskenään. Näin ollen eri valmistajien laitteiden integroiminen osaksi yhtenäistä kotiautomaatioratkaisua on haasteellista tai jopa mahdotonta ja johtaa useiden rinnakkaisten IoT-alustojen, ekosysteemien sekä sovellusten käyttöön kodissa. Ratkaisuksi valmistajien välisiin yhteensopivuusongelmiin on kehitetty uusia teknologioita, kuten Matter ja Thread.

Tämän tutkielman on tarkoitus avata Threadia teknologiana. Tutkielmassa pyritään lisäksi tunnistamaan kotiautomaatioon yleisimmin liittyviä haasteita ja selvittämään, kuinka haasteet olisivat vältettävissä kotiautomaatioratkaisua toteutettaessa. Tutkielmassa käsitellään ensin kotiautomaation mahdollisuuksia ja haasteita sekä kotiautomaatiossa yleisimmin hyödynnettäviä lyhyen kantaman langattomia teknologioita. Teoriaosuus päättyy Threadin teknologiseen tarkasteluun, jonka yhteydessä sivutaan lyhyesti Matteria. Tutkielman empiirisessä osuudessa otetaan käyttöön Home Assistantin päälle rakentuva pienimuotoinen kotiautomaatioratkaisu. Tutkielmassa toteutettu kotiautomaatioratkaisu osoitti, että merkittävimmät kotiautomaatioon liittyvät haasteet ovat vältettävissä, kun ratkaisussa hyödynnetään avoimen lähdekoodin IoT-alustaa, soveltuvaa IEEE 802.15.4 -radiota sekä kotiautomaatiossa vielä suhteellisen uusia teknologioita Matteria ja Threadia.
Avainsanat: IoT, kotiautomaatio, älykoti, Home Assistant, Matter, BLE, Thread, WiFi, Z-Wave, Zigbee
Abstract: With the rapid growth of IoT, home automation is used in more and more homes today. Home automation is usually used to optimize resource usage,

to make living more easier, comfortable and safer and also to save money and time. When utilizing home automation, a major technological challenge arises since devices and applications from different manufacturers are rarely compatible with each other. Because of this, integrating devices from different manufacturers to a single home automation platform is challenging, if not impossible and leads to use of multiple IoT platforms, ecosystems and applications in home. To tackle these compatibility issues between manufacturers, new technologies have been developed such as Matter and Thread.

The purpose of this thesis is to give an overview of Thread as a technology. The thesis also aims to identify the most common challenges related to home automation and find out how these challenges could be avoided when setting up a home automation system. The beginning of the thesis focuses on covering opportunities and challenges related to home automation as well as the most common short-range wireless technologies used in home automation. Theoretical part ends with a technological study of Thread, which concludes with a short introduction to Matter. The empirical part of the thesis involves setting up a small-scale home automation system built on top of Home Assistant. The home automation system implemented in this thesis demonstrated, that the most significant challenges related to home automation can be avoided when utilizing open source IoT platform, appropriate IEEE 802.15.4 radio device and still relatively new technologies in home automation Matter and Thread.

Keywords: IoT, Home Automation, Smart Home, Home Assistant, BLE, Thread, WiFi, Z-Wave, Zigbee

Sanasto

6LoWPAN	IPv6 over Low-Power Wireless Personal Area Network
AES-CCM	Advanced Encryption Standard - Counter with Cipher Block Chaining-Message Authentication Code
BLE	Bluetooth Low Energy
CoAP	Constrained Application Protocol
DNS-SD	Domain Name System Service Discovery
DTLS	Datagram Transport Layer Security
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IPv6	Internet Protocol version 6
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
MAC	Medium Access Control
Matter	IPv6-pohjainen sovelluskerroksen kommunikointiprotokolla
MeshCoP	Mesh Commissioning Protocol
MLE	Mesh Link Establishment
OSI	Open Systems Interconnection
OTBR	OpenThread Border Router
PAN	Personal Area Network
PHY	Physical Layer
RLOC	Routing Locator
TCP	Transmission Control Protocol
Thread	IPv6-pohjainen mesh-verkkoprotokolla IEEE 802.15.4 -standardin IoT-laitteille
UDP	User Datagram Protocol
WiFi	IEEE 802.11 -standardikokoelman langaton tiedonsiirtoteknologia
WPAN	Wireless Personal Area Network
Z-Wave	Mesh-verkkoteknologia ITU-T G.9959 -standardin IoT-laitteille
Zigbee	Mesh-verkkoteknologia IEEE 802.15.4 -standardin IoT-laitteille

Sisällys

1 Johdanto

Kotiautomaatiomarkkinoilla on tarjolla usean eri valmistajan yksittäisiä IoT (Internet of Things) -laitteita sekä kokonaisvaltaisia kotiautomaatiojärjestelmiä, mikä on toisaalta lisännyt mahdollisuuksia toteuttaa monipuolisia kotiautomaatioratkaisuja, mutta aiheuttanut myös hajanaisuutta markkinoille ja lisännyt yhteensopivuusongelmia valmistajien välillä [128]. Kuluttajan kannalta tilanne on kaksijakoinen, sillä IoT-teknologioiden nopean kehityksen myötä kaupalliset älylaitteet ovat suhteellisen edullisia ja siten useimpien saatavilla, mutta eri valmistajien laitteet ja sovellukset kommunikoivat harvoin ongelmitta keskenään. Yksittäisten IoT-laitteiden käyttöönottaminen edellyttää tyypillisesti valmistajan oman mobiilisovelluksen ja usein myös pilvipalvelun käyttöönottamista [49]. Vastaavasti valmistajakohtaiset järjestelmät sekä ekosysteemit voivat olla riippuvaisia valmistajan markkinoimista laitteista, mikä rajoittaa haluttujen laitteiden valintaa ja saattaa turhaan nostaa kotiautomaation kustannuksia. Yhteensopivuusongelmat eri valmistajien laitteiden, IoT-alustojen sekä sovellusten välillä ilmenevät lopulta useina rinnakkaisina ratkaisuina kodissa, vaikeuttaen näin ollen myös kotiautomaation keskitettyä hallintaa.

Kotiautomaation yhteensopivuusongelmat ovat pääosin seurausta yhtenäisen standardoinnin puuttumisesta IoT-ratkaisujen toteutukseen, jonka myötä valmistajat ovat valinneet ratkaisuihinsa parhaaksi näkemänsä tiedonsiirtoteknologiat, protokollat sekä sovellustoteutukset [2]. Yksittäisten laitevalmistajien on vastaavasti täytynyt kehittää IoT-laitteista useita versioita tukemaan eri tiedonsiirtoteknologioita tai tiettyjen valmistajien ekosysteemejä, jotta kilpailu markkinoilla olisi mahdollista [49, 112]. Kotiautomaation yhteensopivuusongelmien ratkaisuksi on esitetty muun muassa Matter-kommunikointiprotokollaa, jota hyödyntämällä eri valmistajien älylaitteet, IoT-alustat sekä sovellukset voivat kommunikoida keskenään. Matter rakentuu WiFin, Ethernetin sekä BLE:n (Bluetooth Low Energy) päälle ja hyödyntää mesh-verkkoteknologiana Threadia. Thread on kotiautomaatiossa vielä suhteellisen uusi teknologia, jonka on tarkoitus lisätä yhteensopivuutta tarjoamalla laitevalmistajille IPv6 (Internet Protocol version 6) -pohjainen verkkoprotokolla IEEE (Institute of Electrical and Electronics Engineers) 802.15.4 -standardiin perustuvien IoT-laitteiden kehittämiseen.

Tutkielman pääasiallinen tutkimusongelma liittyy Thread-verkkoprotokollan teknologiseen tarkasteluun eli mikä Thread on ja mitä se tarjoaa kotiautomaatioon yhteensopivuuden näkökulmasta. Tutkielman toisessa tutkimusongelmassa pyritään tunnistamaan kotiautomaatioon yleisimmin liittyviä haasteita ja selvittämään, millä tavoin ne olisivat vältettävissä. Tutkimusongelmille luodaan pohjaa tarkastelemalla ensin kotiautomaation mahdollisuuksia ja haasteita sekä käymällä läpi yleisimpiä kotiautomaation lyhyen kantaman langattomia teknologioita. Threadin teoreettisen tarkastelun on tarkoitus avata Threadia teknologiana, jonka yhteydessä sivutaan lyhyesti myös Matteria, joka on Threadin kannalta merkityksellisin sovellusratkaisu kotiautomaatiossa. Kotiautomaatioon liittyviä haasteita ja Threadin teoreettista tarkastelua tukemaan tutkielman empiirisessä osuudessa esitellään vaihtoehto kustannustehokkaan sekä useaa tiedonsiirtoteknologiaa tukevan kotiautomaatioratkaisun toteuttamiselle.

Tutkielma jakautuu siten, että luvussa 2 luodaan ensin katsaus kotiautomaation mahdollisuuksiin sekä haasteisiin aiemman tutkimuskirjallisuuden pohjalta. Luvussa 3 tarkastellaan yleisimpiä kotiautomaation lyhyen kantaman langattomia tiedonsiirtoteknologioita ja luvussa 4 perehdytään syvällisemmin Threadiin teknologiana. Luvussa 5 esitellään tutkielmassa toteutettu pienimuotoinen kotiautomaatioratkaisu. Lopulta luku 6 sisältää yhteenvedon tutkielmasta.

2 Kotiautomaatiosta yleisesti

IoT:n nopean kehityksen myötä erilaiset IoT-ratkaisut ovat nykyisin osa arkipäivää ja yhä useampi haluaa luoda myös kotiinsa älykkään ympäristön. Kun kotiin luodaan älykäs ympäristö IoT:tä hyödyntämällä, puhutaan yleisesti älykodista. Tulkinta älykodista vaihtelee eikä sille ole olemassa selkeää määritelmää [120, s. 114], mutta älykkääksi ympäristöksi kutsuminen edellyttää ratkaisulta tietyn toiminnallisuuden täyttymistä. Älykoti voidaan käsittää ratkaisuna, jossa sensorit, aktuaattorit sekä muut kodin älykkäiksi luokiteltavat laitteet pystyvät kommunikoimaan keskenään, ovat asukkaiden hallittavissa ja muodostavat yhtenäisen älykkään kokonaisuuden tarjotakseen asukkaille hyödyllisiä palveluja [119, s. 5-6]. Älykkyyden saavuttaminen edellyttää myös sitä, että ratkaisu kerää sekä tallentaa kokonaisvaltaisesti informaatiota ympäristöstään ja analysoimalla sitä laitteet osaavat tehdä itsenäisiä päätöksiä ilman käyttäjän puuttumista ratkaisun toimintaan [57, s. 83]. Sovacool et al. [93] esittävät artikkelissaan kodin älykkyyden tulkitsemiseen seitsemän tasoa, jotka kuvaavat kodin älykkyyttä suhteessa ratkaisussa hyödynnettäviin teknologioihin. Tasot on esitetty seuraavassa mukaillen Sovacool et al. [93, s. 5-7].

- Taso 0: Koti ilman älyteknologioita tai -laitteita.
- Taso 1: Kodissa on asukkaiden ohjaamia yksinkertaisia älylaitteita, jotka eivät kuitenkaan kommunikoi keskenään.
- Taso 2: Kodin älylaitteet ovat yksinkertaisesti ohjelmoitavissa ja alkavat muodostaa kokonaisuuksia tarjoten asukkaille informaatiota kodin toiminnoista.
- Taso 3: Kodin älylaitteisiin on ohjelmoitavissa monimutkaisempia toimintoja, laitteet kommunikoivat keskenään ja suorittavat tehtäviä automatisoidusti.
- Taso 4: Kodin älyratkaisuista tulee oppivia ja ne mukautuvat kodin sekä ympäristön olosuhteisiin.
- Taso 5: Kodin erityyppiset älyratkaisut pystyvät vaihtamaan tietoa keskenään integroituakseen yhtenäiseksi kokonaisuudeksi, millä on kyky ennustaa asukkaiden sekä kodin tarpeita mahdollistaen täysin automatisoidun toiminnan.
- Taso 6: Tason 5 täyttävien kotien älyratkaisut yhdistyvät myös kodin ulkopuolisiin älyratkaisuihin, kuten älykaupungin palveluihin.

Sovacool et al. esittämistä tasoista on todettavissa, että automatisoiduilla toiminnoilla on lopulta hyvin merkittävä rooli siirryttäessä yhä älykkäämpiin ratkaisuihin. Älykoti rinnastetaankin käsitteenä usein kotiautomaatioon, mikä on toiminut pohjana modernille älykodille ja mahdollistaa itsenäisesti toimivien, älykkäiden sekä automatisoitujen ratkaisujen hyödyntämisen kodeissa [9, 120].

Seuraavissa alaluvuissa käsitellään kotiautomaation mahdollisuuksia sekä merkittävimpiä haasteita aihepiirejä käsittelevän tutkimuskirjallisuuden pohjalta. Kotiautomaation mahdollisuuksien tarkastelemisen on tarkoitus avata kotiautomaation yleisimpiä sovelluskohteita sekä sillä tavoiteltavia hyötyjä. Kotiautomaation haasteiden tarkasteleminen puolestaan tukee tutkielman tutkimusongelmien pohdintaa eli millä tavoin kotiautomaation haasteet ovat vältettävissä ja mitä Thread pyrkii ratkaisemaan kotiautomaatiossa sekä antaa näkökulmia empiirisessä osuudessa toteutetun kotiautomaatioratkaisun tueksi.

2.1 Kotiautomaation mahdollisuudet

Kotiautomaatio on laaja käsite ja sitä voidaan toteuttaa joko yksittäisillä älylaitteilla tai laajemmilla kotiautomaatiojärjestelmillä. Kotiautomaatiota hyödynnetään yleisesti kodin resurssien kulutuksen seurantaan ja hallintaan, asumisen helpottamiseen, asumismukavuuden lisäämiseen sekä kodin ja asukkaiden turvallisuuden parantamiseen liittyvissä ratkaisuissa [9, 93, 94, 120]. Kotiautomaatiolle on ominaista, että eri sovelluskohteiden ratkaisuja pyritään yhdistämään tavalla tai toisella kokonaisuudeksi, jotta kotiautomaatiolla tavoitellut hyödyt ovat saavutettavissa. Esimerkiksi kodin olosuhteiden sekä ympäristön monitoroinnin pohjalta asumisolosuhteita hallitsevia älylaitteita voidaan ohjata joko manuaalisesti tai automatisoidusti asetettujen sääntöjen perusteella ja siten lisätä asumismukavuutta, mutta tehdä asumisesta myös ekologisempaa energiaa säästämällä. Keskeisessä osassa kotiautomaatiota on etähallinta, mikä mahdollistaa kotiautomaatioratkaisun monitoroinnin ja ohjaamisen joko paikallisesti tai mistä tahansa internetin välityksellä. Kuluttajien odotukset ja näkemykset kotiautomaatiosta sekä päätös älylaitteiden hankinnasta vaihtelevat muun muassa iän, teknologisen ymmärryksen, taloudellisen tilanteen, asumismuodon sekä kotiautomaatiota kohtaan vallitsevan kiinnostuksen mukaan [30, 52, 82]. Kotiautomaatiolla tavoitellaan useimmiten taloudellisia sekä ajallisia säästöjä ja ratkaisuilta odotetaan helppoa käyttöönottoa, hallittavuutta, yhteensopivuutta, hyödyllisyyttä, luotettavuutta sekä yksityisyyttä [9, 52, 78, 82, 120].

2.1.1 Kotiautomaation sovelluskohteet

Yksi merkittävimmistä kotiautomaation tavoitteista liittyy kodin energiankulutuksen seurantaan sekä hallintaan ja sitä kautta saavutettavaan taloudelliseen säästöön [93, s. 7-8]. Kotiautomaatioratkaisuilla energiankulutusta on mahdollista seurata reaaliaikaisesti ja etähallinnan avulla yksittäisiä laitteita voidaan tarpeen mukaan ohjata internetin välityksellä. Kodin energiankulutuksen hallintaan liittyvät ratkaisut voivat olla myös täysin automatisoituja, jolloin laitteiden ohjaaminen perustuu asukkaiden läsnäolon huomiointiin tai asetettuihin sääntöihin, kuten ajastuksiin sekä sähkön hinnoittelun seuraamiseen [51, s. 12-13]. Energiankulutuksen hallintaan liittyvän kotiautomaation odotetaan siirtyvän tulevaisuudessa yhä enemmän myös kodin ulkopuolelle osaksi älykkäitä sähköverkkoja HEMS-ratkaisujen (Home Energy Management System) myötä. HEMS-ratkaisu tuo mukanaan hyötyjä niin koteihin kuin sähköyhtiöille, kun sähkönkulutusta voidaan ajoittaa paremmin kodin ja asukkaiden tarpeita, sähkön hintoja sekä sähköverkon kuormitusta seuraavaksi [56, s. 2-4; 93, s. 8].

Asumismukavuuden lisääminen sekä asumisen helpottaminen ovat myös yleisiä kotiautomaation sovelluskohteita [93, s. 7-8; 120, s. 118-119], jotka toimivat usein yhteistyössä energiankulutuksen hallintaan liittyvän kotiautomaation kanssa. Ratkaisut voivat olla esimerkiksi kodin valaistuksen ohjaamista keskitetysti erilaisilla käyttäjälaitteilla tai kodin olosuhteiden monitorointia ja havaintojen perusteella tapahtuvaa lämmityksen, jäähdytyksen sekä ilmanvaihdon ohjaamista joko automatisoidusti asetettuihin sääntöihin perustuen tai manuaalisesti etähallinnan kautta [94, s. 12-14]. Kodin asumisolosuhteiden pitäminen optimaalisina lisää asumismukavuutta, mutta luo kotiin myös terveellisemmän asumisympäristön sekä parantaa kodin energiatehokkuutta ja siten asumisen ekologisuutta. Toisaalta, vaikka asumismukavuutta lisäävät ratkaisut mahdollistavat asukkaille helpotusta kodin toimintojen hallintaan keskitettyjen sekä automatisoitujen ratkaisujen muodossa, saattavat ne tietyissä tapauksissa toimia odotusten vastaisesti lisäten kodin energiankulutusta [24, s. 14; 93, s. 11].

Kotiautomaatio käsittää usein myös erilaisia kodin sekä asukkaiden turvallisuutta parantavia ratkaisuja. Kotiautomaatioon integroituva turvallisuusratkaisu on useimmiten valvonta-/hälytinjärjestelmä, joka voi rakentua useista erityyppisistä asukkaita, kotia ja kodin ympäristöä monitoroivista sensoreista, kameroista sekä älylukituksesta [1, 9, 32, 58, 94]. Kotiautomaatioon liitetyt hälytysjärjestelmät voivat tiedottaa muun muassa epäilyttävistä havainnoista kodin ympäristössä asukkaiden

poissa ollessa, vesivuodoista, tulipalosta tai muusta vaarasta ja tehdä tarvittaessa ilmoituksen suoraan asianmukaisille viranomaisille [44, s. 271; 129, s. 4]. Kodin turvallisuusratkaisuihin on toisaalta yhdistettävissä myös niin sanottuja älyterveydenhuollon ratkaisuja, jotka mahdollistavat esimerkiksi tuetun asumisen toimintoja iäkkäille henkilöille. Morita et al. [60, s. 10] ovat määritelleet terveydenhuoltoa tukevan älykodin ratkaisuna, jossa kodin älykkäät sensorit mahdollistavat toimintojen automatisoinnin sekä ohjaamisen ja kodin sisätilojen sekä asukkaiden terveydentilan monitoroinnin. Edelleen Morita et al. mukaan tällaisen ratkaisun pääpaino tulisi olla asumismukavuuden ja -turvallisuuden ylläpitäminen sekä asukkaan elämänlaadun parantaminen. Älyterveydenhuollon ratkaisuissa voidaan hyödyntää joko kotiin asennettuja sensoreita tai puettavia IoT-laitteita [51, s. 7-8], jotka sopivaa tiedonsiirtoteknologiaa käytettäessä olisivat edelleen yhdistettävissä samaan IoT-alustaan, jolla hallitaan kodin muitakin automaatioratkaisuja [10].

2.1.2 Yleistä kotiautomaatioratkaisuista

Kotiautomaatiomarkkinoilla on tarjolla käytännössä kahdenlaisia kaupallisia kotiautomaatioratkaisuja usealta eri valmistajalta. Varsinaiset kotiautomaatiojärjestelmät ovat kokonaisvaltaisia ratkaisuja, jotka tukevat yleensä valmistajan hyväksymiä laitteita sekä tiedonsiirtoteknologioita ja ovat siten ainakin osittain suljettuja ratkaisuja [23, s. 12-13, 16]. Kotiautomaatiojärjestelmien etuna on kuitenkin se, että ne muodostavat yhtenäisen kokonaisuuden, johon sisältyy ratkaisuun soveltuvat laitteet sekä palvelut, jonka myötä hankinta ja käyttöönotto on tehty kuluttajalle helpoksi. Kaupallisiksi ratkaisuiksi voidaan luokitella myös valmistajakohtaisen IoT-alustan päälle rakentuvat ekosysteemit, joista tunnetuimpien joukkoon lukeutuvat muun muassa Amazon Alexa, Apple Home, Google Home sekä Samsung SmartThings [113]. Nämä ovat useimmiten hybridiratkaisuja [2, s. 14-15], jotka rakentuvat valmistajan markkinoimista hub-laitteista sekä pilvipalvelusta ja kaupallisuus muodostuu älylaitteiden hankkimisesta käyttäjäsovellusten ollessa ilmaisia. Valmistajakohtaiset hub-laitteet eivät välttämättä ole edellytys yksittäisten älylaitteiden käytölle, mutta ne mahdollistavat laitteiden keskitetyn hallinnan, automaatioiden hyödyntämisen sekä kotiautomaatioratkaisun säilymisen toimintakuntoisena, vaikka pilvipalvelu olisi saavuttamattomissa internet-yhteyden katkeamisesta johtuen [2, s. 15; 113]. Lisäksi useimpien valmistajien IoT-alustat sekä hub-laitteet tukevat nykyisin rajoitetusti myös muiden valmistajien älylaitteita, mikä lisää kuluttajan kannalta valinnanvaraa markkinoilla [113].

Kaupallisiin kotiautomaatioratkaisuihin liittyy kuitenkin omat haasteensa ja tästä johtuen kotiautomaatiota toteutetaan yhä useammin ilmaisen avoimen lähdekoodin IoT-alustan päälle. IoT-markkinoilla on tarjolla huomattava määrä erilaisia kotiautomaatioon soveltuvia IoT-laitteita, sensoreita, kehitysalustoja sekä muita komponentteja [23, 58, 94], jotka yhdessä avoimen lähdekoodin IoT-alustan kanssa mahdollistavat kotiautomaatioratkaisun toteuttamisen kaupallisia ratkaisuja huomattavasti kustannustehokkaammin [79]. Avoimen lähdekoodin IoT-alustojen lähtökohtana on pilvipalveluista riippumaton paikallinen kotiautomaatioratkaisu, joka tarjoaa kaupallisiin vaihtoehtoihin verrattuna enemmän muokattavuutta ja hallittavuutta käytettävien laitteiden, tiedonsiirtoteknologioiden, sovellusprotokollien, datan tallennuksen ja esittämisen sekä yksityisyyden osalta [22, s. 9-11; 79]. Avoimen lähdekoodin IoT-alustat nojaavat vahvasti kehittäjien sekä käyttäjien muodostaman yhteisön tarjoamaan tukeen, jonka myötä alustoilla voidaan hyödyntää suurta määrää erityyppisiä laitteita, mutta toisaalta alustojen keskinäiset ominaisuudet vaihtelevat merkittävästi, mikä tulisi huomioida alustaa valittaessa. Kotiautomaatioon suunnattuja avoimen lähdekoodin IoT-alustoja on tarjolla useita vaihtoehtoja, joista suosituimpien joukkoon lukeutuvat muun muassa Home Assistant, Domoticz, ioBroker sekä openHAB [79].

2.2 Kotiautomaation haasteet

Kotiautomaatio tarjoaa hyötyjä useasta näkökulmasta, mutta luotettavasti toimivan yhtenäisen kotiautomaatioratkaisun toteuttaminen, ylläpito tai laajentaminen ei ole kuitenkaan ongelmatonta. Älykotimarkkinoilla on saatavilla sekä kaupallisia kotiautomaatioratkaisuja että yksittäisiä älylaitteita usealta eri valmistajalta ja kuluttajan voi olla vaikea päättää, minkä tyyppinen ratkaisu täyttää omat tarpeet käytettävyyden, yhteensopivuuden sekä tavoiteltujen hyötyjen näkökulmasta. Kotiautomaatiolle on ominaista, että ratkaisu rakentuu vaiheittain useasta erityyppisestä laitteesta, sovelluksesta sekä palvelusta, joiden elinkaari voi vaihdella huomattavasti ja ratkaisun laajentaminen voi olla riippuvaista aiemmin hankituista laitteista tai palveluista [82, s. 247]. Kuluttajille suunnatut IoT-ratkaisut suunnitellaan tänä päivänä yhä useammin yhteensopivuutta tavoitellen, jotta kuluttajilla olisi parempi mahdollisuus valita haluamansa laitteet eri valmistajilta [2, s. 6]. Yhteensopivuuden toteutuminen valmistajien välillä on kuitenkin noussut yhdeksi merkittävimmistä teknologisista haasteista kotiautomaatiossa, sillä se toteutuu useimmiten ainoastaan

valmistajan oman ekosysteemin sisällä ja on siten johtanut rinnakkaisten sekä osin rajoittuneiden ratkaisujen käyttöön kodeissa. Yhteensopivuusongelmien ohella kotiautomaatioon liittyviä haasteita ovat muun muassa kuluttajien tietämättömyys kotiautomaation hyödyistä tai epäilevä suhtautuminen teknologisten ratkaisujen käyttöön kotona, kaupallisten ratkaisujen korkeat hankinta- ja ylläpitokustannukset, kotiautomaatioratkaisujen käyttöönoton hankaluudet ja rajoitettu muokattavuus sekä riittävän tietoturvallisuuden ja yksityisyyden saavuttaminen [9, 33, 52, 93, 120].

2.2.1 Suhtautuminen kotiautomaatioon

Vaikka erilaiset IoT-ratkaisut on yhä useammin tavalla tai toisella osa arkipäivää, suhtaudutaan kotiautomaation hyödyntämiseen edelleen epäillen. Tämä voi olla seurausta puutteellisesta käsityksestä kotiautomaation tarjoamista hyödyistä, luottamuksen puutteesta älyteknologioita sekä niitä markkinoivia tahoja kohtaan tai ajatuksesta, että kotiautomaation käyttöönottaminen on vaikeaa ja sen hyödyntäminen johtaa lopulta kodin kontrollin sekä yksityisyyden menettämiseen [30,33,52,93]. Edellä mainitut haasteet eivät välttämättä ole ainoastaan kuluttajien ennakkonäkemyksistä johtuvia. Kaaz et al. [45] tekemän tutkimuksen mukaan pelkästään yksittäisten älylaitteiden käyttöönotto saatetaan kokea haastavaksi, käyttäjäsovellukset ovat rajoittuneita sekä vaikeita käyttää ja käyttäjällä on todellisuudessa vähän vaikutusmahdollisuuksia liittyen laitteiden keräämän datan yksityisyyteen. Kotiautomaation laajamittaisen yleistymisen edellytyksenä on, että kuluttajat ymmärtävät paremmin mitä kotiautomaatiolla on saavutettavissa, onnistuvat käyttöönotossa, kokevat kotiautomaation hyödyllisenä sekä luotettavana ja hyväksyvät näin ollen älyteknologiat osaksi kodin arkea [30, 33, 52, 93].

2.2.2 Kaupallisten ratkaisujen haasteet

Kaupalliset kotiautomaatiojärjestelmät ovat kuluttajille helppo vaihtoehto toteuttaa kotiautomaatiota eri sovelluskohteisiin, sillä ne sisältävät kaiken tarvittavan yhdessä paketissa. Ratkaisujen haasteena on kuitenkin se, että ne ovat yleisesti kalliita hankkia ja ylläpitokustannukset nousevat laitteiden sekä tuettujen teknologioiden määrän kasvaessa [9; 23, s. 16]. Lisäksi ratkaisut saattavat edellyttää rakenteellisia muutoksia kotiin ja mahdolliset käytönaikaiset kulut, kuten kuukausimaksut tai lisäominaisuuksien käyttöönottaminen, vaikuttavat osaltaan ratkaisun kokonaiskustannuksiin. Kaupalliset kotiautomaatiojärjestelmät ovat usein myös ainakin osittain

valmistajan rajoittamia esimerkiksi tuettavien laitteiden sekä protokollien osalta [23, s. 12-13, 16]. Valmistajakohtaiset rajoitukset saattavatkin sitoa kuluttajan tietyn valmistajan laitteisiin, mikä ei ainoastaan rajoita haluttujen laitteiden valintaa, vaan voi myös tarpeettomasti nostaa ratkaisun kokonaiskustannuksia. Yaldaie et al. [120] tekemän tutkimuksen mukaan useimmat kuluttajat ovat valmiita maksamaan kotiautomaatiolaitteista muutamia satoja euroja, joten kustannuksilla on merkittävä vaikutus hankintapäätökselle kotiautomaatiosta sekä motivaatiolle laajentaa ratkaisua myöhemmin.

2.2.3 Teknologiset haasteet

Kotiautomaation teknologiset haasteet liittyvät useimmiten yhteensopivuusongelmiin eri valmistajien laitteiden ja sovellusten välillä sekä ratkaisujen käyttöönoton haasteisiin ja luotettavaan toimintaan [9, s. 2020; 60, s. 11; 73; 93, s. 10]. Kotiautomaation yhteensopivuusongelmat ovat osin seurausta IoT:n nopeasta kehityksestä sekä yleistymisestä, mutta etenkin yhtenäisen standardin puuttumisesta IoT-ratkaisujen kehitystyöhön sekä toteuttamiseen. IoT-ratkaisuissa hyödynnetään lähes poikkeuksetta standardisoituja teknologioita, jotka yhdessä eri organisaatioiden toteuttamien sertifiointiohjelmien myötä ovat mahdollistaneet osittaisen yhteensopivuuden saavuttamisen valmistajien välille [70, s. 183-184]. IoT-ratkaisun toteuttamiseen soveltuvia tiedonsiirtoteknologioita, verkko- ja sovellusprotokollia sekä datamuotoja on kuitenkin olemassa huomattava määrä [22, s. 3-8] ja yhtenäisen standardoinnin puuttuessa valmistajat ovat toteuttaneet IoT-laitteita sekä sovellusratkaisuja parhaaksi näkemiään teknologioita hyödyntäen [2, s. 24]. Välttääkseen yhteensopivuusongelmat kuluttaja saattaa olla sidottuna valmistajakohtaisiin laitteisiin sekä teknologioihin tai kotiautomaatioratkaisun laajentaminen on riippuvaista aiemmin hankituista järjestelmistä, mikä osaltaan rajoittaa uusien laitteiden hankintaa ja vaikeuttaa kotiautomaation keskitettyä hallintaa [9, s. 2023; 23, s. 16]. Yhteensopivuusongelmilla on toisaalta myös suora vaikutus kotiautomaatiosta aiheutuviin kustannuksiin [120, s. 122], sillä haasteet uusien laitteiden integroimisessa osaksi olemassa olevia kodin ratkaisuja voi nostaa kustannuksia tarpeettomasti.

Teknologisiin haasteisiin voidaan lukea myös kotiautomaation hyödyntäminen itsessään, kun kodin eri toiminnot nojaavat yhä enemmän älyteknologioiden varaan [93, s. 10-11]. Kotiautomaation luotettava toiminta edellyttää, että ratkaisuun sisältyvät laitteet pystyvät kommunikoimaan ongelmitta keskenään ja suorittavat tehtävänsä asetettujen sääntöjen mukaisesti. IoT-laitteiden toiminta voi kuitenkin estyä

useista syistä, kuten tiedonsiirto-ongelmien, virransaannin katkeamisen tai puutteellisen tietoturvan myötä jopa hakkeroinnin seurauksena [44, s. 272-273]. Oman haasteensa kotiautomaation luotettavalle toiminnalle asettaa pilvipalveluiden hyödyntäminen. Useimpien valmistajien kotiautomaatioratkaisut nojaavat pitkälti pilvipalveluihin, sillä ne skaalautuvat hyvin suurten laite- ja datamäärien hallintaan, tukevat resurssirajoittuneiden laitteiden käyttöä IoT-ratkaisuissa sekä ovat saavutettavissa mistä tahansa internetin välityksellä [92, s. 272]. Jos kotiautomaatioratkaisulta kuitenkin edellytetään jatkuvaa yhteyttä pilvipalveluun esimerkiksi laitekomentojen välittämiseksi, johtaisi kodin internet-yhteyden katkeaminen kotiautomaatioratkaisun joko osittaiseen tai täydelliseen toimimattomuuteen [2, s. 14].

2.2.4 Tietoturva ja yksityisyys

Kodeissa hyödynnettävien IoT-laitteiden määrän nopean kasvun myötä yhdeksi merkittävimmistä kotiautomaation haasteista on noussut tietoturvallisuuden sekä yksityisyyden toteutuminen [44, 78, 93]. Kotiautomaatioratkaisulle on tyypillistä, että useiden eri valmistajien laitteet keräävät suuren määrän tietoa kodin eri toiminnoista, asukkaista sekä käyttötottumuksista ja tieto voi olla hyvinkin yksilöivää [44, s. 270-271; 93, s. 10]. Kaupallisten kotiautomaatioratkaisujen tapauksessa älylaitteet ovat useimmiten jatkuvasti yhteydessä internetiin, jotta muun muassa etähallinta ja pilvipalvelut ovat hyödynnettävissä. Etähallinta mahdollistaa kotiautomaatioratkaisun helpon monitoroinnin sekä hallinnan mistä tahansa, mutta toisaalta jatkuva internet-yhteys altistaa kodin laitteet tavallista useammin hakkeroinnille ja lisää siten riskiä datan varastamiselle tai väärinkäytölle sekä kodin verkkoon kytkeytyneiden laitteiden haitalliselle toiminnalle [44, s. 272-273].

Myös pilvipalveluiden hyödyntämiseen liittyy merkittäviä haasteita. Kodin laitteiden kommunikoidessa jatkuvasti pilvipalveluiden kanssa on luotettava siihen, että valmistajat huolehtivat laitteiden, sovellusalustojen sekä datan suojaamisesta asianmukaisesti, dataa ei jaeta tai myydä kolmansille osapuolille ja käyttäjien sekä kodin yksityisyys säilyy turvattuna [2, s. 14-15; 92, s. 279]. Schomakers et al. [78] ovat tutkimuksessaan havainneet, että kuluttajat suosivat kotiautomaatioratkaisuissa valmistajan pilvipalvelusta riippumatonta paikallista datan tallennusta, kun kyseessä on yksityiseksi luokiteltavan tiedon kerääminen ja käsitteleminen. Tämän saavuttaminen on kuitenkin haasteellista, sillä on yleistä, että jopa yksittäiset älylaitteet edellyttävät valmistajan pilvipalvelun käyttöönottamista, jonka myötä valmistajalla on pääsy kaikkeen laitteiden kodista keräämään dataan [93, s. 10].

3 Kotiautomaation tiedonsiirtoteknologiat

Kotiautomaation useat erilaiset sovelluskohteet asettavat omat vaatimuksensa ratkaisuihin valittaville tiedonsiirtoteknologioille. Kotiautomaatiossa hyödynnetään sekä langallisia että langattomia teknologioita, joiden keskinäiset ominaisuudet vaihtelevat muun muassa tiedonsiirtonopeuksien, datamäärien, virrankulutuksen, tuettavan laitemäärän, verkon laajuuden ja toimintavarmuuden sekä tietoturvan osalta [57, s. 90-92; 69]. Kotiautomaation IoT-ratkaisut perustuvat useimmiten sensoreilta kerättävän informaation monitorointiin sekä aktuaattoreiden ohjaamiseen, jolloin ratkaisuilta ei edellytetä suuria tiedonsiirtonopeuksia tai datamääriä [69, s. 14]. Yleensä tärkeämpää on laitteiden asennuksen ja käyttöönoton helppous, kustannusten pitäminen alhaisina, luotettavasti toimiva verkkoratkaisu sekä IoT-laitteiden virrankulutuksen minimointi [69, s. 23; 94, s. 2].

Kotiautomaatioratkaisuissa on jo vuosikymmenten ajan hyödynnetty langallisia teknologioita, kuten BACnet, Insteon, KNX, UPB (Universal Powerline Bus) sekä X10 [57, s. 91-92; 69, s. 3-4], jotka perustuvat Ethernet-, valokuitu-, pari- tai sähkökaapelointien hyödyntämiseen tiedonsiirrossa. Langallisilla teknologioilla voidaan toteuttaa luotettavasti toimivia sekä tarvittaessa laajojakin automaatioverkkoja, jotka eivät ole alttiita esimerkiksi langattomille teknologioille ominaisille häiriöille, tiedonsiirtonopeuden vaihtelulle tai tietoturvauhille, kuten salakuuntelulle. Teknologioiden välillä on kuitenkin merkittäviä eroja muun muassa tuettujen kaapelityyppien, laitemäärien, salausmenetelmien sekä tiedonsiirtonopeuksien ja -etäisyyksien osalta [69, s. 14-22]. Langallisten teknologioiden haasteina ovat suuri virrankulutus ja edellytys jatkuvalle virransyötölle, fyysiselle kaapeloinnille sekä kiinteille laiteasennuksille, jotka lisäävät käyttöönotto- ja ylläpitokustannuksia [57, s. 91; 69, s. 25]. Langallisten teknologioiden rinnalla hyödynnetäänkin usein myös langattomia teknologioita [69, s. 4, 19], jotta saavutetaan optimaalinen ratkaisu kustannustehokkuuden, virrankulutuksen sekä verkon rakenteen ja toimintavarmuuden näkökulmasta.

IoT:n nopea kehitys ja yleistyminen ovat osaltaan johtaneet uusien langattomien teknologioiden syntyyn sekä olemassa olevien teknologioiden kehittymiseen paremmin resurssirajoittuneille IoT-laitteille sopiviksi [63, 68, 69]. Uusien teknologioiden myötä useat valmistajat ovat tuoneet markkinoille suuren määrän langatto-

miin teknologioihin perustuvia IoT-laitteita sekä kehitysalustoja, jotka ovat yleisesti myös suhteellisen edullisia ja siten useimpien kuluttajien saatavilla. Kotiautomaatiossa hyödynnettävät langattomat teknologiat ovat useimmiten WLAN- (Wireless Local Area Network) tai WPAN (Wireless Personal Area Network) -teknologioita, joista suosituimpia ovat WiFi, BLE (Bluetooth Low Energy), Zigbee sekä Z-Wave [69]. Näiden lisäksi Thread on yleistymässä kotiautomaatiossa nopeasti, pääosin siitä syystä, että se on WiFin ohella toinen Matteriin valituista langattomista teknologioista. Seuraavissa luvuissa keskitytään tarkastelemaan WiFi- BLE-, Zigbee- sekä Z-Wave-teknologiaa ja Threadia käsitellään yksityiskohtaisemmin luvussa 4.

3.1 WiFi

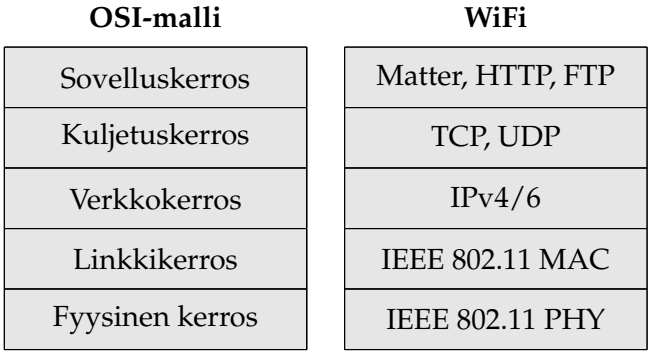
WiFi on IEEE 802.11 -standardikokoelmaan perustuva WLAN-teknologia, joka on vuosien saatossa kehittynyt useilla uusilla standardiversioilla [42]. Uudet standardit ovat tuoneet mukanaan teknologisia parannuksia muun muassa tiedonsiirtonopeuksiin, kantamaan ja tietoturvaan [11] sekä IoT-ratkaisuihin paremmin soveltuvien standardien muodossa [116]. IEEE 802.11 -standardien kehittämisestä ja julkaisusta vastaa kansainvälinen IEEE-järjestö. WiFi-laitteiden sertifioinnista vastaa Wi-Fi Alliance, jonka sertifiointiohjelmien on tarkoitus varmistaa WiFi-laitteiden keskinäinen yhteensopivuus aiempien sekä tulevien IEEE 802.11 -standardien välillä. Lisäksi sertifiointi antaa valmistajille luvan käyttää Wi-Fi Certified -logoa tuotteissaan. Taulukossa 3.1 on esitetty tunnetuimpien IEEE 802.11 -standardiversioiden ominaisuuksia.

Taulukko 3.1: IEEE 802.11 -standardeja [11, 42, 68, 116].

Standardi	**Julkaisu**	**Taajuus**	**Tiedonsiirtonopeus**
802.11 (WiFi 0)	1997	2.4 GHz	2 Mbit/s
802.11b (WiFi 1)	1999	2.4 GHz	11 Mbit/s
802.11a (WiFi 2)	1999	5 GHz	54 Mbit/s
802.11g (WiFi 3)	2003	2.4 GHz	54 Mbit/s
802.11n (WiFi 4)	2009	2.4, 5 GHz	600 Mbit/s
802.11ac (WiFi 5)	2013	5 GHz	3.5 Gbit/s
802.11ah (WiFi Halow)	2017	< 1 GHz	150 Kbit/s – 80 Mbit/s
802.11ax (WiFi 6)	2021	2.4, 5, 6 GHz	9.6 Gbit/s
802.11be (WiFi 7)	2024	2.4, 5, 6 GHz	40 Gbit/s

3.1.1 Kerrosmalli

Kuvassa 3.1 esitettyyn OSI (Open Systems Interconnection) -malliin suhteutettuna IEEE 802.11 -standardit määrittelevät kerrosmallista fyysisen kerroksen sekä linkkikerroksen ja verkko-, kuljetus- sekä sovelluskerroksella WiFin kerrokset noudattelevat TCP/IP (Transmission Control Protocol / Internet Protocol) -kerrosmallia [65].



Kuva 3.1: WiFin kerrosmalli.

Uusien standardien myötä WiFin tiedonsiirtonopeutta on onnistuttu kasvattamaan ottamalla käyttöön uusia taajuuksia sekä tehostamalla taajuuskaistan käyttöä kehittyneemmillä modulaatiomenetelmillä. IEEE 802.11a toi alun perin mukanaan OFDM (Orthogonal Frequency Division Multiplexing) -modulaation [42], mikä mahdollisti yhtäaikaisen tiedonsiirron rinnakkaisilla taajuuskanavilla. OFDM:n myötä tiedonsiirtonopeus kasvoi merkittävästi verrattuna aiempaan IEEE 802.11b -standardiin, jossa modulaatio perustui DSSS (Direct Sequence Spread Spectrum) -menetelmään. IEEE 802.11n -standardi vastaavasti mahdollisti MIMO-OFDM:n (Multiple Input Multiple Output OFDM) käyttöönottamisen [11, s. 574], jossa useiden antennien hyödyntäminen lisäsi WiFin suorituskykyä sekä kantamaa edellisiin standardiversioihin nähden [76]. IEEE 802.11ac -standardissa MIMO-tekniikka kehittyi edelleen MU-MIMO (Multi User MIMO) -menetelmällä, jonka myötä WiFi-reititin pystyy siirtämään dataa lähetyssuunnassa yhtäaikaisesti neljälle asiakaslaitteelle [84, s. 14]. Viimeisimmän julkaistun IEEE 802.11ax -standardin myötä kanavankäyttöä on edelleen tehostettu ottamalla käyttöön OFDMA (Orthogonal Frequency Division Multiple Access) -modulointimenetelmä, mikä mahdollistaa taajuuskanavien jakamisen pienempiin alikanaviin useiden asiakaslaitteiden kesken [84, s. 12]. IEEE 802.11ax -standardin hyödyntämien OFDMA- ja MU-MIMO-menetelmien myötä dataa voidaan siirtää yhtäaikaisesti WiFi-reitittimen ja kahdeksan asiakaslaitteen

välillä sekä lähetys- että vastaanottosuunnassa [65]. WiFin merkittävin haaste IoT-laitteiden kannalta on teknologian suuri virrankulutus sekä suhteellisen lyhyt kantama. IoT-ratkaisuihin paremmin soveltuvaksi WiFi-teknologiaksi julkaistiin vuonna 2017 IEEE 802.11ah -standardi, josta käytetään myös yleisnimitystä Wi-Fi HaLow [116]. Muista IEEE 802.11 -standardeista poiketen Wi-Fi HaLow toimii alle 1 GHz:n taajuusalueella ja teknologia mahdollistaa WiFi-laitteiden kantaman kasvattamisen noin 1 kilometriin hyvin alhaisella virrankulutuksella.

IEEE 802.11 -standardien linkkikerroksella hyödynnetään useita eri tekniikoita, joilla pyritään lisäämään tiedonsiirron luotettavuutta ruuhkaisella 2.4 GHz:n taajuusalueella. CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) on yleinen langattomien teknologioiden hyödyntämä menetelmä, jossa laite kuuntelee siirtotietä ennen lähetyksen aloittamista ja lähettää paketin, jos kanava todetaan vapaaksi tai muussa tapauksessa laite yrittää lähetystä uudelleen satunnaisen perääntymisajan jälkeen [111, s. 160]. Lisäksi linkkikerroksella hyödynnetään muun muassa kuittauksia datan perillemenon varmistamiseksi, RTS/CTS (Request To Send/Clear To Send) -mekanismia piilossa olevien noodien ongelmaan sekä erityisiä virransäästömenetelmiä, jotka mahdollistavat päätelaitteiden siirtymisen alhaisen virrankulutuksen tilaan [65; 111, s. 161].

3.1.2 Verkkoratkaisut ja laitetyypit

Verkkoratkaisuiltaan WiFi-laitteet muodostavat joko tähti- tai ad hoc -topologian [68, s. 18]. Tyypillisin verkkoratkaisu on WiFi-päätelaitteista sekä WiFi-reitittimestä (Access Point, AP) rakentuva topologia, johon muun muassa kodin WiFi-verkot perustuvat. Tähtitopologiassa AP-reitittimen kantaman alueella olevat päätelaitteet liittyvät samaan SSID:llä (Service Set Identifier) tunnistettavaan AP-reitittimeen, jonka kautta WiFi-verkon päätelaitteet kytkeytyvät internetiin. Ad hoc -verkossa päätelaitteet sen sijaan kommunikoivat suoraan keskenään ilman AP-reititintä. Wi-Fi Direct [117] on tyypillinen ad hoc -verkon muodostava ratkaisu, jossa esimerkiksi mobiililaite ja televisio muodostavat keskinäisen WiFi-yhteyden. Wi-Fi Alliancen julkaiseman EasyMesh-ratkaisun myötä WiFi tukee myös mesh-tyyppistä verkkoratkaisua [118]. EasyMesh mahdollistaa yhtenäisen WiFi-verkon muodostamisen useaa AP-reititintä hyödyntämällä, jotta esimerkiksi kodin WiFi-verkon kantamaa sekä toimintavarmuutta pystytään parantamaan.

3.1.3 Tietoturva

WiFin tietoturva perustuu langattoman tiedonsiirron salaamiseen laitteiden välillä. Ensimmäisten IEEE 802.11 -standardien käyttämä salausmenetelmä oli RC4 (Rivest Cipher 4) -salausalgoritmiin perustuva WEP (Wired Equivalent Privacy) -protokolla [50]. WEP todettiin kuitenkin nopeasti tietoturvattomaksi siinä havaittujen useiden haavoittuvuuksien myötä [77] ja sen korvaajaksi julkaistiin vuonna 2003 WPA (Wi-Fi Protected Access) -protokolla [31]. WPA:ssa tietoturvaa parannettiin ottamalla käyttöön jaetun avaimen yhteyden muodostus (Pre-Shared Key, PSK) ja hyödyntämällä datan salaamisessa TKIP- (Temporal Key Integrity Protocol) sekä MIC (Message Integrity Check) -protokollia [50]. WPA:ssa hyödynnettiin kuitenkin edelleen WEP-protokollan RC4-algoritmia [77], jonka myötä myös WPA todettiin lopulta haavoittuvaksi. WPA:n korvaajaksi julkaistiin vuonna 2004 WPA2, joka toi mukanaan merkittäviä parannuksia WiFin tietoturvaan AES (Advanced Encryption Standard) -algoritmia hyödyntävän CCMP (Counter Mode Cipher Block Chaining Message Authentication Code Protocol) -salausprotokollan sekä kommunikaation aloittamisessa käytettävän 4-vaiheisen kättelyn myötä [31]. WPA2 on edelleen laajasti käytössä tänä päivänä ja sen merkittävin tietoturvaongelma on PSK-menetelmä, johon voidaan kohdistaa hyökkäyksiä WiFi-verkon salasanan selvittämiseksi [59]. Wi-Fi Alliance julkaisi vuonna 2018 WPA3-protokollan [115] WPA2:n korvaajaksi. WPA3:ssa tietoturvaa ja salausavainten muodostusta on edelleen kehitetty ja PSK on korvattu SAE (Simultaneous Authentication of Equals) -protokollalla, jonka on tarkoitus parantaa suojausta heikkoa verkon salasanaa ja sen selvittämiseen kohdistuvia hyökkäyksiä vastaan [114, 115]. WPA3 ja WPA2 ovat nykyisin pakollisia toteutuksia Wi-Fi Certified -tuotteissa.

3.2 Bluetooth Low Energy

Bluetooth on lyhyen kantaman langaton teknologia, jota hyödynnetään tänä päivänä useissa erityyppisissä laitteissa, kuten älypuhelimissa, kaiuttimissa, kuulokkeissa sekä IoT-kehitysalustoissa [63, s. 67902]. Bluetoothin ensimmäinen 1.0-versio julkaistiin vuonna 1999, jonka IEEE standardisoi IEEE 802.15.1 -standardissa. Nykyisin Bluetoothin kehittämisestä, spesifikaatioiden määrittämisestä ja julkaisusta sekä laitteiden sertifioinnista vastaa Bluetooth SIG (Special Intrest Group). Bluetoothin merkittävimmät versiot on esitetty taulukossa 3.2.

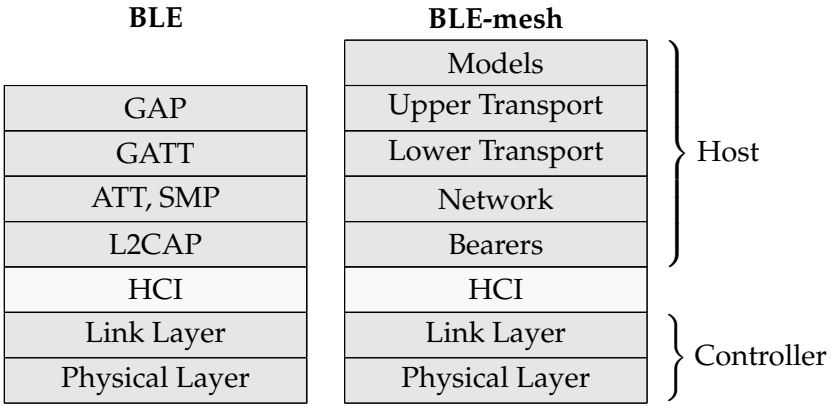
Taulukko 3.2: Bluetooth-versiot [6, 63].

Versio	Julkaisu
Bluetooth v1.0	1999
Bluetooth v2.0 (EDR)	2004
Bluetooth v3.0 (HS)	2009
Bluetooth v4.0, BLE	2010
Bluetooth v5.0	2016

Bluetooth jakautuu teknologiana BR/EDR- (Basic Rate and Enhanced Data Rate) sekä BLE (Bluetooth Low Energy) -kategorian laitteisiin [6, s. 94]. BR/EDR on perinteisempi versio Bluetoothista, joka on optimoitu jatkuvan datavirran lähettämiseen lyhyellä kantamalla mahdollisimman virtatehokkaasti [86, s. 2]. BR/EDR:lle hyvin tyypillinen sovelluskohde on äänen siirtäminen älypuhelimen ja langattomien kuulokkeiden välillä. BLE on vastaavasti IoT-laitteille suunnattu teknologia, joka tarjoaa BR/EDR-laitteisiin verrattuna alhaisemman virrankulutuksen ja mahdollistaa mesh-verkkojen toteuttamisen [8, s. 9]. Useimmat älypuhelimet tukevat tänä päivänä sekä BR/EDR- että BLE-teknologiaa, jonka myötä yhdellä laitteella voidaan hyödyntää hyvin erityyppisiä Bluetooth-laitteita. Jatkossa tässä luvussa tarkastellaan BLE-teknologiaa Bluetooth 5.0 -spesifikaation pohjalta.

3.2.1 Kerrosmalli

Kuvassa 3.2 esitetty BLE:n kerrosmalli kattaa kaikki OSI-referenssimallin mukaiset kerrokset, joten BLE ei ole teknologiana riippuvainen muista standardeista [8, s. 12].



Kuva 3.2: BLE:n kerrosmallit, muokattu [8, s. 13].

BLE:n kerrosmalli jakautuu käyttöjärjestelmä- ja laitetason määritteleviin host- sekä controller-lohkoihin, joiden välisestä kommunikaatiosta vastaa HCI (Host Controller Interface) -rajapinta [8, s. 12]. Host-lohko koostuu useista kerroksista, jotka ovat GAP- (Generic Access Profile), GATT- (Generic Attribute Profile), ATT- (Attribute Protocol), SMP- (Security Manager Protocol) sekä L2CAP (Logical Link Control and Adaptation Protocol). Host-lohkon kerroksilla on omat tehtävänsä BLE-laitteiden välisen kommunikaation mahdollistamiseksi, datan välittämiseen kerrosten välillä, BLE-laitteiden profiilien ja palveluiden määritykseen sekä tietoturvamenetelmien hallintaan [8, s. 59-76].

BLE:n kerrosmallin controller-lohkon muodostavat BLE:n fyysinen kerros sekä linkkikerros. BLE toimii 2.400 - 2.4835 GHz:n taajuusalueella, joka jakautuu 40:een 2 MHz:n kanavaan [8, s. 15]. Taajuuskanavia 37, 38 ja 39 käytetään laitteiden etsimiseen, yhteyden muodostamiseen sekä mainostamiseen ja kanavat 0 - 36 ovat käytettävissä varsinaiseen datan siirtämiseen laitteiden välillä [86, s. 4]. Mainostus on oleellinen osa BLE-laitteiden toimintaa, mikä mahdollistaa muun muassa laitteiden löytämisen, yhteyden muodostuksen sekä laitteilla olevien palveluiden mainostamisen [86, s. 5]. Bluetooth 5.0 -spesifikaation myötä kaikki 40 kanavaa ovat tarvittaessa käytettävissä BLE-laitteiden mainostamiseen. BLE:n modulaatio perustuu GFSK (Gaussian Frequency Shift Keying) -menetelmään, jossa negatiivinen tai positiivinen taajuussiirtymä keskitaajuudesta kuvaa binääriä 0 tai 1 [8, s. 15]. BLE:n fyysisellä kerroksella on käytettävissä kolme eri modulaatiomenetelmää; LE 1M PHY, LE 2M PHY sekä LE Coded PHY. Modulaatiomenetelmät kuvaavat datan lähetysnopeutta symboleina sekunnissa ja ne määrittävät myös GFSK:ssa käytettävän taajuussiirtymän [8, s. 15-16]. LE 1M PHY -menetelmä on pakollinen BLE-laitteissa ja sen symbolinopeus on 1 Msym/s sekä teoreettinen tiedonsiirtonopeus 1 Mbit/s. LE 2M PHY on vaihtoehtoinen menetelmä, jonka symbolinopeus on 2 Msym/s ja teoreettinen tiedonsiirtonopeus 2 Mbit/s. LE Coded PHY on myös vaihtoehtoinen ja sen symbolinopeus on 1 Msym/s, mutta teoreettinen tiedonsiirtonopeus ainoastaan 125 – 500 Kbit/s. Tämä johtuu menetelmässä hyödynnettävästä FEC (Forward Error Correction) -virheenkorjauskoodauksesta, jossa yksi bitti kuvataan joko kahdella tai kahdeksalla bitillä. Muihin modulaatiomenetelmiin verrattuna FEC parantaa tiedonsiirron luotettavuutta ja mahdollistaa BLE-laitteiden kantaman kasvattamisen kaksin- tai nelinkertaiseksi, mutta tekee sen hitaamman tiedonsiirtonopeuden sekä suuremman pakettikoon kustannuksella [7, s. 20; 63, s. 67907]. LE Coded PHY tunnetaan myös BLE Long Range -versiona.

BLE:n linkkikerroksella toimii tilakone, jonka tilat määrittelevät BLE-laitteiden toiminnan. BLE-laite voi olla joko standby-, advertising-, scanning-, initiating- tai connection-tilassa [6, s. 2553]. Standby-tilassa laite ei lähetä tai vastaanota dataa ja tila on virransäästöä varten [63, s. 67906]. Advertising-tilassa laite joko lähettää mainostuskanavilla tai ainoastaan kuuntelee muiden laitteiden mainostuksia. Niin ikään scanning-tilassa laite kuuntelee muiden lähettämiä mainostuksia ja tila voi olla joko passiivinen, jolloin laite ei vastaa mainostuspaketteihin tai aktiivinen, jolloin laite vastaa mainostavalle laitteelle saadakseen siltä lisäinformaatiota [63, s. 67906]. Vastaavasti initiating-tilassa laite kuuntelee tiettyjen laitteiden mainostuksia tarkoituksenaan muodostaa yhteys johonkin niistä [6, s. 2553]. Kun laitteiden on tarkoitus siirtää dataa keskenään, ne siirtyvät connection-tilaan. Kommunikaatio siirtyy tällöin mainostuskanavilta datansiirtoon tarkoitetuille kanaville ja tiedonsiirto tapahtuu master-slave-tyyppisesti [63, s. 67906]. Linkkikerros kontrolloi myös BLE:n taajuuskanavien käyttöä, jotta kommunikointi useilla kanavilla on mahdollista sekä mahdollisimman luotettavaa. Usean muun langattoman teknologian käyttämän CSMA/CA-menetelmän sijaan BLE hyödyntää adaptiivista taajuushyppelyä (Adaptive Frequency Hopping, AFH), jonka toiminta perustuu kanavanvalinta-algoritmiin [8, s. 22]. Algoritmi seuraa BLE-yhteyksien laatua ja ylläpitää kanavataulua käyttökelpoisista sekä huonoista kanavista, jonka perusteella AFH mukautuu siirtotiellä tapahtuviin muutoksiin. Linkkikerros vastaa lisäksi FEC-virheenkorjauksen koodaamisesta datapakettiin, jos käytössä on LE Coded PHY sekä 24-bittisen CRC (Cyclic Redundancy Check) -tarkistussumman lisäämisestä datapakettiin ja sen tarkistuksesta bittivirheiden varalta [8, s. 20].

3.2.2 Verkkoratkaisut ja laitetyypit

BLE tukee verkkoratkaisuina tähti-, broadcast- sekä mesh-topologiaa [5]. Tähtitopologia on tyypillinen BLE-laitteiden muodostama verkko, jossa useat BLE-laitteet voivat muodostaa yhteyden samaan keskittimenä toimivaan laitteeseen, kuten älypuhelimeen [5; 86, s. 10]. Broadcast-verkon toiminta perustuu viestin lähettämiseen yhdeltä usealle. BLE-majakat ovat esimerkki tällaisesta topologiasta, jossa yksi BLE-laite lähettää tasaisin väliajoin pieniä datapaketteja muille lähettyvillään oleville BLE-laitteille [63, s. 67906]. Bluetooth 4.0 -versiosta lähtien myös mesh-topologia on tuettuna, mutta ainoastaan BLE-laitteilla. Viestien välitys mesh-verkossa perustuu tulvamenetelmään (engl. flooding) [4, s. 25], jolloin lähettävän laitteen viestin vastaanottaa kaikki kantaman sisällä olevat laitteet. Viestin vastaanottaneet laitteet

välittävät viestin edelleen eteenpäin datapakettiin asetetun hyppyjen määrän kertovan TTL (Time To Live) -kentän mukaisesti. BLE-laitteet ylläpitävät välimuistia viimeisimmistä vastaanotetuista viesteistä, jolla estetään kertaalleen välitettyjen viestien uudelleenlähetys [4, s. 26]. Jotta kommunikaatio mesh-verkossa olisi mahdollista, BLE-laitteilla on erilaisia toimintatiloja. Kun BLE-laite on liittynyt verkkoon, se voi toimia joko relay-, friend-, LPN- (Low Power Node) tai proxy-noodina [4, s. 16]. Relay-noodin tehtävänä on välittää vastaanottamansa viestin eteenpäin verkon muille noodeille mahdollistaen usean hypyn kommunikaation verkossa. Friend- ja LPN-noodit toimivat aina yhteistyössä. LPN-noodi voi mennä virransäästötilaan säästääkseen paristoaan eikä voi tällöin vastaanottaa viestejä. Friend-noodin tehtävänä on tallentaa vastaanottamansa viestit muistiin ja välittää viestit LPN-noodille sen pyytäessä niitä. BLE:n mesh-protokollan kerrosmalli rakentuu BLE-protokollan controller-lohkon päälle [8, s. 13], mutta host-lohkon kerrokset ja toiminnallisuus ovat erilaisia mesh-toteutuksesta johtuen. Proxy-noodin tehtävänä on mahdollistaa pelkkää BLE-protokollaa tukevan laitteen, kuten älypuhelimen, kommunikointi mesh-verkkoon kuuluvien BLE-noodien kanssa [4, s. 16].

3.2.3 Tietoturva

BLE:n tietoturva perustuu laiteparin muodostamiseen ja sen yhteydessä luotavien verkko-, sovellus- sekä laiteavainten vaihtamiseen [4, s. 23; 86, s. 17]. Laiteparin muodostamiseen on käytettävissä neljä menetelmää: Numeric Comparison, Just Works, OOB (Out of Band) sekä Passkey Entry [6, s. 244-246, 248]. Numeric Comparison -menetelmää käytetään, kun laiteparin molemmat laitteet pystyvät esittämään kuusinumeroisen liittämiskoodin näytöllä ja käyttäjä hyväksyy laiteparin muodostuksen molemmissa laitteissa. Just Works -menetelmä perustuu Numeric Comparison -menetelmään ja se on käytössä sellaisten BLE-laitteiden laiteparin muodostuksessa, kun liittämiskoodin syöttäminen ei ole mahdollista liitettävässä laitteessa. Käyttäjältä kysytään tällöin ainoastaan hyväksyntä laiteparin muodostamiseksi. OOB-menetelmässä laiteparia ei muodosteta BLE-yhteydellä, vaan tyypillisesti NFC (Near Field Communication) -teknologiaa hyödyntämällä. NFC:tä käytettäessä liitettävät laitteet viedään lähelle toisiaan ja laiteparin muodostaminen varmistetaan käyttäjältä. Passkey Entry -menetelmä on vastaavasti käytettävissä silloin, kun laiteparin muodostamiseksi kuusinumeroinen liittämiskoodi voidaan esittää käyttäjälle toisessa laitteessa, mutta liitettävässä laitteessa koodi on ainoastaan käyttäjän syötettävissä.

BLE-laitteiden välillä siirrettävän datan muuttumattomuuden varmistamiseksi kaikki paketit salataan linkkikerroksella 128-bittisellä AES-CCM (AES - Counter with Cipher Block Chaining-Message Authentication Code) -salauksella [86, s. 19]. BLE-laitteilla on lisäksi käytettävissä erityinen yksityisyysominaisuus, jolla pyritään estämään laitteiden seuranta verkossa. Ominaisuutta hyödyntävä BLE-laite vaihtaa säännöllisesti julkista laiteosoitettaan, josta ainoastaan aiemmin laiteparin muodostaneet laitteet pystyvät purkamaan kyseisen laitteen yksityisen laiteosoitteen ja edelleen kommunikoimaan laitteen kanssa [86, s. 19].

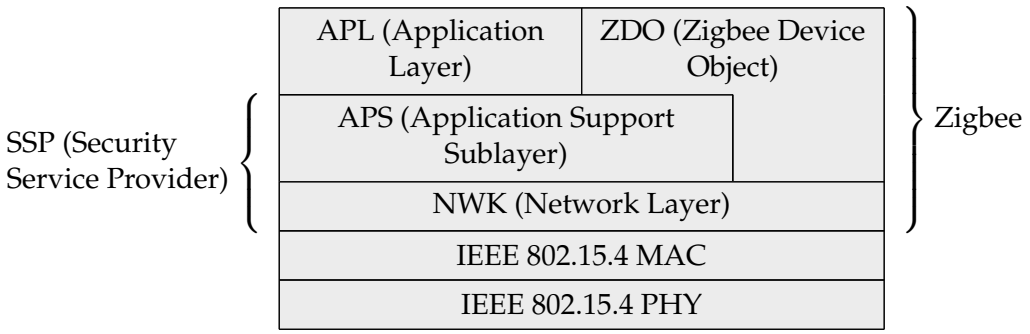
3.3 Zigbee

Zigbee on vuosien saatossa noussut yhdeksi suosituimmista kotiautomaatiossa hyödynnettävistä langattomista teknologioista [69]. IEEE 802.15.4 -standardin päälle rakentuvana ja mesh-pohjaisia verkkoratkaisuja tukevana teknologiana Zigbee soveltuu erinomaisesti paristokäyttöisten IoT-laitteiden teknologiaksi, jonka sovelluskohteet eivät rajoitu ainoastaan kotiautomaatioon [90, s. 3]. Zigbee luokitellaan avoimeksi standardiksi, joten teknologia on siten kenen tahansa hyödynnettävissä useiden eri valmistajien piiri- sekä alustaratkaisuilla. Zigbeen kehittämisestä, spesifikaatioiden määrityksestä ja julkaisusta sekä laitesertifioinnista vastaa CSA (Connectivity Standards Alliance) yhdessä jäsenorganisaatioidensa kanssa [18].

Zigbee 1.0 -versio julkaistiin vuonna 2004, jonka jälkeen Zigbee on päivittynyt muun muassa Zigbee 2006-, Zigbee 2007- (Zigbee PRO), Zigbee 3.0- sekä Zigbee PRO 2023 -versiolla [19,20,90]. Ensimmäiset Zigbee-spesifikaatiot määrittelivät useita erillisiä Zigbee-laiteprofiileja muun muassa kotiautomaatioon, älymittaukseen sekä terveydenhuoltoon. Vuonna 2016 julkaistun Zigbee 3.0:n merkittävin uudistus oli laiteprofiilien yhdistäminen BDB (Base Device Behavior) -mallin alle [90, s. 28], mikä lisäsi yhteensopivuutta eri valmistajien välillä. Vuonna 2023 julkaistu Zigbee PRO 2023 toi mukanaan tietoturvaparannuksia Zigbee-laitteiden käyttöönottoon ja salausavainten käsittelyyn, laajensi Zigbeen tuen alle 1 GHz:n taajuuksille sekä esitteli uutena ominaisuutena Zigbee Directin [19]. Zigbee Directin on tarkoitus helpottaa Zigbee-laitteiden käyttöönottoa ja tehdä Zigbee-verkoista paremmin saavutettavia BLE-teknologiaa hyödyntämällä. Esimerkiksi älypuhelin voi Zigbee Directin myötä kommunikoida suoraan Zigbee-verkon laitteiden kanssa BLE- ja Zigbee-protokollat toteuttavan Zigbee-laitteen kautta [83].

3.3.1 Kerrosmalli

OSI-referenssimalliin suhteutettuna Zigbee rakentuu fyysisellä kerroksella ja linkkikerroksella IEEE 802.15.4 -standardin päälle [20, s. 1]. Zigbee-spesifikaatiot vastaavasti määrittelevät kuvassa 3.3 esitetyn kerrosmallin mukaiset verkko- sekä sovelluskerrokset, joilla määritellään Zigbee-laitteiden varsinainen toiminnallisuus.



Kuva 3.3: Zigbeen kerrosmalli, muokattu [20, s. 2].

Zigbeen verkkokerroksen (Network Layer, NWK) tehtävänä on vastata verkon muodostuksesta, reitityksestä, laitteiden liittämisestä sekä osoitteistuksesta [20, s. 234]. Lisäksi NWK-kerroksella voidaan tarvittaessa hyödyntää uudelleenlähetystä, jos linkkikerroksen uudelleenlähetykset epäonnistuvat [90, s. 13]. APS (Application Support Sublayer) -kerros toimii rajapintana NWK-kerroksen sekä sovelluskerroksen (Application Layer, APL) välillä ja se vastaa päästä päähän viestin välityksestä, salauksesta sekä kuittauksista [20, s. 14, 16; 91, s. 5]. APL-sovelluskerroksella määritellään Zigbee-laitteiden varsinainen toiminnallisuus. Zigbee-laitteella voi sen rakenteesta riippuen olla useita toiminnallisia sovelluksia, jotka kuvataan 1–254 numeroituina päätepisteinä [20, s. 16]. Päätepiste koostuu yhdestä tai useammasta Zigbee-klusterista, joka sisältää päätepisteellä olevat attribuutit sekä komennot eli laitteeseen sisällytetyt toiminnot. Klusterilla on joko asiakas- tai palvelinrajapinta, joiden välillä Zigbee-laitteet kommunikoivat keskenään [90, s. 20]. Zigbee-klusterit ja niillä olevat attribuutit sekä komennot on määritelty ZCL (Zigbee Cluster Library) -spesifikaatiossa. APS- ja APL-kerrosten yhteydessä toimii lisäksi ZDO (Zigbee Device Object), joka on jokaiseen Zigbee-laitteeseen sisällytettävä ylläpitosovellus [67, s. 46]. ZDO:lle on varattu päätepisteen tunniste 0 ja sen tehtävänä on määritellä laiteroolit, alustaa APS- ja NWK-kerrokset sekä tietoturvamenetelmät määrittelevä SSP (Security Service Provider) ja mahdollistaa laitteiden sekä palveluiden löytäminen verkossa [20, s. 17, 201].

Zigbeen fyysinen kerros sekä linkkikerros perustuvat IEEE 802.15.4 -standardiin, joka on kehitetty alhaisen virrankulutuksen IoT-laitteille, joiden toiminta ei edellytä suuria datamääriä tai tiedonsiirtonopeuksia [90, s. 3]. IEEE 802.15.4 -standardin ominaisuudet vaihtelevat taajuusalueesta riippuen taulukossa 3.3 esitetyn mukaisesti.

Taulukko 3.3: IEEE 802.15.4 -standardin ominaisuuksia [63, s. 67896].

Taajuus	**Tiedonsiirtonopeus**	**Modulaatio**
868 MHz (EU)	20 Kbit/s	BPSK
915 MHz (US)	40 Kbit/s	BPSK
2.4 GHz	250 Kbit/s	O-QPSK

IEEE 802.15.4 -standardin laitteet toimivat joko 868/915 MHz:n tai 2.4 GHz:n lisenssivapailla taajuuksilla. Käytettävästä taajuusalueesta riippuen tiedonsiirtonopeus vaihtelee välillä 20–250 Kbit/s ja standardi määrittelee datapaketin maksimikooksi 127 tavua [111, s. 152, 154]. Standardin fyysinen kerros vastaa useista tehtävistä, kuten laitteen radion hallinnasta, modulaatioista, yhteyden laadun ja vastaanotetun signaalin voimakkuuden mittaamisesta sekä kanavan tilan selvittämisestä [68, s. 11]. IEEE 802.15.4 -pohjaiset IoT-laitteet ovat useimmiten paristokäyttöisiä, jolloin laitteen radion hallinta mahdollistaa lepotilojen hyödyntämisen [63, s. 67897] ja siten virrankulutuksen minimoinnin. Modulaatio perustuu käytettävästä taajuusalueesta riippuen joko BPSK- (Binary Phase Shift Keying) tai O-QPSK (Offset Quadrature Phase Shift Keying) -menetelmään. Modulaatiossa hyödynnetään lisäksi DSSS (Direct-Sequence Spread Spectrum) -menetelmää, jolla pyritään parantamaan signaalin häiriönsietokykyä usean teknologian jakaessa sama taajuusalue [63, s. 67897]. Yhteyden laatua kuvaavaa LQI (Link Quality Indicator) -arvoa hyödynnetään yleisimmin selvittäessä linkin kustannuksia naapurinoodeihin. Zigbee-verkossa muun muassa reitittävät laitteet hyödyntävät LQI-arvoa reititystietojen yhteydessä, jotta niillä on tiedossa optimaalisin ja luotettavin yhden hypyn linkki verkossa [90, s. 8].

IEEE 802.15.4 -standardin linkkikerroksen tehtävänä on mahdollistaa luotettava yhden hypyn tiedonsiirto verkossa. Törmäysten välttämiseksi linkkikerroksella käytetään CCA (Clear Channel Assessment) -menetelmää. CCA perustuu fyysisen kerroksen suorittamaan signaalin voimakkuuden mittaukseen (Energy Detection), josta saatua tulosta verrataan asetettuun raja-arvoon [111, s. 155-156]. Jos CCA:n myötä kanavalla havaitaan toisen laitteen lähetys, linkkikerros yrittää paketin lähetystä uudelleen perääntymisajan jälkeen. Linkkikerros vastaa lisäksi vastaanotettujen pakettien kuittauksesta sekä CRC (Cyclic Redundancy Check) -tarkistussumman

lisäämisestä jokaiseen pakettiin ja pakettivirheiden tarkistuksesta vastaanottajalla [111, s. 156]. Linkkikerroksella hyödynnetään myös uudelleenlähetystä, jos lähetettyyn pakettiin ei saada kuittausta tai CCA-tarkistus ei ole todennut kanavaa vapaaksi. Linkkikerroksella suoritettavat kuittaukset toimivat aina yhden hypyn päähän ja mahdolliset uudelleenlähetykset suoritetaan viisi kertaa hyvin nopeassa syklissä [90, s. 12].

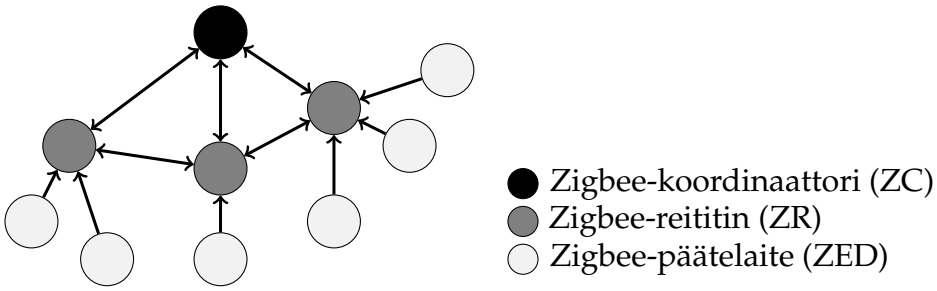
3.3.2 Verkkoratkaisut ja laitetyypit

Zigbeen laitetyypit noudattelevat IEEE 802.15.4 -standardia, jossa on määritelty yleisesti FFD- (Fully Functional Device) sekä RFD (Reduced Functional Device) -tyypin laitteet [111, s. 153]. FFD-tyyppinen laite on RFD-laitetta suorituskykyisempi, yleensä verkon koordinaattori tai reititin, joka on kytkettynä verkkovirtaan [63, s. 97896]. RFD-laitteet ovat vastaavasti vähävirtaisia, useimmiten paristokäyttöisiä laitteita, joilla on mahdollisuus hyödyntää lepotilaa. Zigbee-spesifikaatio määrittelee edelleen kolmentyyppisiä laitteita: koordinaattorin (Zigbee Coordinator, ZC), reitittimen (Zigbee Router, ZR) sekä päätelaitteen (Zigbee End Device, ZED) [111, s. 296]. Näistä ZC sekä ZR ovat FFD-tyyppisiä laitteita ja ZED-laitteet RFD-tyyppisiä, jotka voivat kommunikoida ainoastaan ZC:n tai ZR:n kanssa.

Jokaisessa Zigbee-verkossa on aina yksi ZC-koordinaattori. ZC vastaa verkon alustamisesta, mahdollistaa laitteiden liittymisen verkkoon sekä toimii verkon keskitettynä turvallisuuslaitteena (Trust Center, TC), joka vastaa muun muassa verkkoavainten jakamisesta ZR- ja ZED-laitteille [90, s. 6]. ZR-reitittimien tehtävänä on ylläpitää reititystietoja, välittää muiden laitteiden paketteja verkossa sekä toimia ZED-laitteiden liityntäpisteenä verkkoon. ZR-reitittimet mahdollistavat verkon laajuuden kasvattamisen ja siten mesh-verkkojen toteutumisen Zigbeessä. ZED voi olla joko paristokäyttöinen lepotilaa hyödyntävä laite tai radiotaan päällä pitävä verkkovirtaan kytketty laite. ZED-laitteet eivät voi välittää muiden laitteiden paketteja verkossa ja ne kommunikoivat aina oman ZR-reitittimensä kanssa [111, s. 297].

IEEE 802.15.4 -standardin tukemat verkkotopologiat ovat tähti, puu sekä mesh [63, s. 97896], jotka ovat käytettävissä myös Zigbeessä. Tähtitopologiassa ZR ja ZED kommunikoivat suoraan ZC-koordinaattorin kanssa, eikä verkossa ylläpidetä tai vaihdeta reititystietoja. Puutopologia voi muodostua, jos Zigbee-laitteiden kantama ei mahdollista mesh-verkon muodostumista [66]. Puutopologiassa yksittäiset ZR-reitittimet voivat kommunikoida keskenään välittääkseen muiden laitteiden viestejä verkossa. Kuvassa 3.4 esitetty mesh-verkko on tyypillisin Zigbee-laitteiden

muodostama topologia, jossa kaikki samaan verkkoon kytkeytyneet ZR:t pystyvät kommunikoimaan keskenään ja yhdellä ZR:llä voi olla useita ZED-laitteita hallinnassaan. Kommunikaatio Zigbee-verkossa perustuu joko reititystaulun hyödyntämiseen yhdellä hypyllä, yleislähetykseen, ryhmälähetykseen tai monilähetykseen (engl. many-to-one) [90, s. 7]. ZR-reitittimien ylläpitämien reititystaulujen hyödyntäminen on tyypillisin kommunikaatiomenetelmä esimerkiksi kahden ZED-laitteen välillä. Reittien selvittämiseksi ZR:t lähettävät verkkoon yleislähetyksenä selvitysviestin, johon saapuvan vastauksen myötä reittitiedot päivittyvät myös muihin reitillä oleviin reitittimiin [67, s. 48]. Ryhmälähetystä Zigbee-laitteet käyttävät silloin, kun sama viesti on tarkoitus lähettää usealle samaan ryhmään kuuluvalle laitteelle, kuten esimerkiksi ryhmälle Zigbee-lamppuja [90, s. 7]. Monilähetystä Zigbee-laitteet voivat hyödyntää, jos kaikkien verkossa olevien laitteiden täytyy lähettää viesti yhdelle keskittimenä toimivalle Zigbee-laitteelle.



Kuva 3.4: Zigbeen mesh-topologia, muokattu [111, s. 296].

Zigbee-verkon muodostamisesta vastaa aina ZC-koordinaattori. Verkkoa alustettaessa ZC skannaa käytettävissä olevat kanavat ja valitsee Zigbee-verkolle vähiten häiriöisen kanavan, joka pysyy samana niin kauan, kuin Zigbee-verkko on olemassa [111, s. 300]. Verkon alustuksen yhteydessä ZC määrittelee verkolle myös 16-bittisen PAN ID (Personal Area Network Identifier) -osoitteen sekä 64-bittisen EPID (Extended PAN ID) -osoitteen [67, s. 40]. Verkon muodostuksen jälkeen ZR- ja ZED-laitteet voivat pyytää liittymistä verkkoon. Uusien laitteiden liittyminen verkkoon on mahdollista joko ZC:n tai ZR:n kautta, jolta Zigbee-laite saa tietoonsa käytettävän kanavan, verkon PAN ID- ja EPID-osoitteen, laitteelle satunnaisesti muodostetun 16-bittisen verkko-osoitteen sekä datapakettien salauksessa käytettävän verkkoavaimen [67, s. 38, 41; 91, s. 4].

3.3.3 Tietoturva

Tietoturva Zigbeessä perustuu APS-kerroksen linkkiavaimen sekä NWK-kerroksen verkkoavaimen hyödyntämiseen [20, s. 408]. Zigbeen NWK-kerroksella kaikki datapaketit salataan 128-bittisellä NWK-avaimella, joka on kaikkien samassa verkossa toimivien Zigbee-laitteiden tiedossa. Salaus perustuu AES-CCM (AES - Counter with Cipher Block Chaining-Message Authentication Code) -menetelmään, jonka yhteydessä pakettiin lisätään myös MIC (Message Integrity Code) -tarkistuskoodi paketin muuttumattomuuden varmistamiseksi [91, s. 4]. NWK-kerroksella hyödynnetään lisäksi kehyslaskuria, jolla pyritään suojautumaan muun muassa toistohyökkäyksiä (engl. replay attack) vastaan [91, s. 4].

Zigbee-verkossa on aina yksi turvallisuuslaitteena toimiva laite (Trust Center, TC), jonka tehtävänä on hyväksyä uusien laitteiden liittyminen verkkoon sekä muodostaa ja jakaa NWK-verkkoavaimet laitteille [20, s. 411]. Yleensä Zigbee-verkon koordinaattori toimii TC-laitteena [46, s. 6]. Kun uusi ZR tai ZED pyrkii liittymään verkkoon, välittyy tästä tieto TC:lle. TC päättää hylätäänkö laitteen liittymispyyntö, vai sallitaanko laitteen liittyminen verkkoon, jolloin laitteelle palautetaan verkon NWK-avain [91, s. 4]. TC:n tehtävänä on lisäksi uusia NWK-avain tietyn ajan kuluttua, jotta laitteiden kehyslaskurit nollaantuvat ja verkosta poistuneiden laitteiden palaaminen verkkoon voidaan estää [91, s. 12]. Jotta Zigbee-verkko säilyisi toimintakuntoisena, voi aiemmin verkkoon liittynyt laite palata takaisin verkkoon minkä tahansa ZR:n kautta, jos sillä on hallussaan verkossa edelleen käytettävä NWK-avain [91, s. 13]. Muussa tapauksessa takaisin verkkoon palaavan laitteen on aloitettava liittymisprosessi uudelleen TC:n kautta.

Zigbeen APS-kerroksen salaus on vastaavasti päästä päähän -salausta, jolloin ainoastaan kahdella keskenään kommunikoivalla laitteella on tiedossaan tarvittava linkkiavain salauksen purkamiseen [91, s. 6]. APS-kerroksen salausta hyödynnetään muun muassa silloin, kun TC toimittaa NWK-avaimen verkkoon liittyvälle uudelle laitteelle. Jos kaksi verkkoon kuuluvaa laitetta haluavat hyödyntää APS-kerroksen salausta, vastaa TC linkkiavaimen muodostuksesta ja toimittamisesta laitteille. Merkittävin tietoturvahaaste salausavainten käsittelyssä on liittynyt NWK-avaimen jakamiseen, sillä valmistajat ovat käyttäneet Zigbee-laitteiden linkkiavaimina yleisesti tiedossa olevia avaimia [46, s. 6]. Zigbeen 3.0-versiossa NWK-avaimen jakamista on kehitetty tietoturvallisempaa suuntaan, jonka myötä TC:n ja Zigbee-verkkoon liittyvän laitteen välille muodostetaan yhteinen salausavain, joka on johdettu liittyvän laitteen laitetunnisteesta [91, s. 8].

3.4 Z-Wave

Z-Wave on kotiautomaatioon suunnattu alhaisen virrankulutuksen ja tiedonsiirtovaatimuksen langaton teknologia, joka on korkean tietoturvavaatimuksensa myötä erittäin suosittu etenkin kodin turvallisuuteen liittyvissä IoT-ratkaisuissa. Z-Waven ensimmäinen versio julkaistiin 2000-luvun alussa ja kotiautomaatiomarkkinoilla on nykyään saatavilla tuhansia sertifioituja Z-Wave-laitteita muun muassa kodin valaistuksen, lukituksen ja lämmityksen ohjaamiseen sekä erityyppisiin sensoreista koostuviin ratkaisuihin [123]. Z-Waven kehittämisestä, spesifikaatioiden julkaistusta sekä laitteiden sertifioinnista vastaa Z-Wave Alliance. Z-Wave-laitteiden sertifiointi takaa, että kaikki sertifioinnin läpäisseet ja Z-Wave-logon saaneet laitteet ovat keskenään yhteensopivia valmistajasta riippumatta [127]. Z-Wave on siitä poikkeuksellinen teknologia, että ainoana piirivalmistajana toimii Silicon Labs [88].

Z-Wavesta on julkaistu Z-Wave Plus (yleisesti Z-Wave) sekä Z-Wave LR (Long Range) -versiot [88], joiden ominaisuuksia on esitetty taulukossa 3.4. Muista kotiautomaation WPAN-teknologioista poiketen Z-Wave toimii alle 1 GHz:n taajuuksilla ja rakentuu ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) G.9959 -standardin päälle [126]. Z-Wave tukee yhden verkon osalta maksimissaan 232 laitetta ja hyödyntää mesh-reititystä muodostaakseen toimintavarman laiteverkon. Z-Wave LR on suunnattu pitkän kantaman teknologiaksi, joka tukee 4000 laitetta yhdessä verkossa ja mahdollistaa yli 1 km:n kantaman saavuttamisen [122].

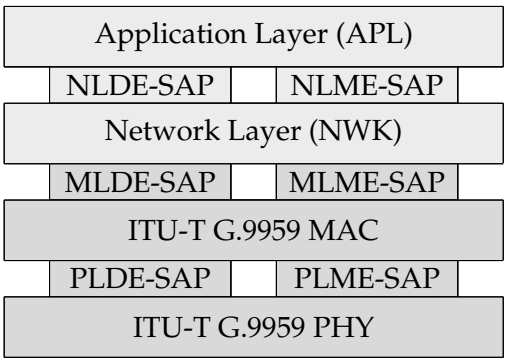
Taulukko 3.4: Z-Waven versiot [89, 122].

Versio	**Taajuus**	**Nopeus**	**Kantama**	**Laitemäärä**	**Topologia**
Z-Wave	865.2 – 926.3 MHz	9.6 – 100 Kbit/s	100 m	232	Mesh
Z-Wave LR	912 – 920 MHz	100 Kbit/s	> 1000 m	4000	Tähti

3.4.1 Kerrosmalli

Z-Waven fyysinen kerros sekä linkkikerros rakentuu ITU-T G.9959 -standardin päälle [95, s. 68] ja Z-Wave-spesifikaatiot määrittelevät verkko- sekä sovelluskerroksen [126, s. 13]. Kuvassa 3.5 esitetyn kerrosmallin jokaisen kerroksen välissä toimii lisäksi kaksi SAP (Service Access Point) -palvelurajapintaa, joiden kautta kerrokset välittävät vastaanottamansa ja käsittelemänsä informaation seuraavalle kerrokselle. Z-Wave toimii alle 1 GHz:n lisenssivapailla taajuusalueilla, jotka vaihtelevat eri

puolilla maailmaa 865.2 – 926.3 MHz:n välillä. ITU-T G.9959 -standardi määrittelee taajuusalueille taulukon 3.5 mukaiset datanopeusluokat, joissa modulaatio perustuu joko FSK- (Frequency Shift Keying) tai GFSK (Gaussian FSK) -menetelmiin ja datan koodauksessa käytetään Manchester- tai NRZ (Non Return to Zero) -koodausta.



Kuva 3.5: Z-Waven kerrosmalli, muokattu [126, s. 13].

Taulukko 3.5: ITU-T G.9959 -standardin ominaisuuksia [95, s. 11-12].

Datanopeus	**Tiedonsiirtonopeus**	**Modulaatio**	**Koodaus**
R1	9.6 Kbit/s	FSK	Manchester
R2	40 Kbit/s	FSK	NRZ
R3	100 Kbit/s	GFSK	NRZ

Taajuusvalinnan sekä datanopeusluokkiin liittyvien määritysten ohella fyysinen kerros vastaa laitteen radion hallinnasta, datan vastaanotosta ja lähetyksestä, kanavan tilan selvittämistä (Clear Channel Assessment, CCA) sekä linkin laadun mittaamisesta [95, s. 8]. ITU-T G.9959 -standardin linkkikerroksen tehtävänä on verkko- ja laitetunnisteiden määritteleminen sekä varmistaa luotettavan tiedonsiirron toteutuminen verkossa [95, s. 27-29]. Linkkikerros vastaa kanavalle pääsystä fyysisen kerroksen CCA-tarkistuksen perusteella, kuittauksista datan onnistuneen vastaanoton ja muuttumattomuuden todentamiseksi sekä tarvittaessa paketin uudelleenlähetyksestä. Datan muuttumattomuus todennetaan datanopeuksilla R1 ja R2 8-bittisellä FCS (Frame Check Sequence) -tarkistussummalla ja datanopeudella R3 16-bittisellä CRC (Cyclic Redundancy Code) -tarkistussummalla [95, s. 46-47].

Varsinaiset Z-Wave-spesifikaatiot määrittelevät verkko- (NWK) sekä sovelluskerroksen (APL). NWK-kerros vastaa Z-Wave-verkon muodostamiseen ja ylläpitoon liittyvistä toimista sekä reititystietojen määrityksestä ja hallinnasta, jotta laitteet voivat saavuttaa muita laitteita usean hypyn kommunikaatiolla [126, s. 14, 20].

APL-kerros määrittelee Z-Wave-laitteiden ominaisuudet ja toiminnot sekä turvallisuusluokan, jonka perusteella laitteiden välinen kommunikaatio salataan. Z-Waven laitespesifikaatio [124] määrittelee useita laitetyyppejä eri sovelluskohteisiin, joiden on tarkoitus varmistaa yhteensopivuus eri valmistajien Z-Wave-laitteiden välillä. Laiteluokat jakautuvat basic-, generic- sekä specific-luokkiin, jotka määrittelevät laitteen roolin verkossa sekä sillä olevat komentoluokat [124, s. 3-6]. Komentoluokat ovat kokoelmia laitteen omista sekä muiden laitteiden ohjaamiseen käytettävistä komennoista, joiden perusteella Z-Wave-laitteet kommunikoivat keskenään.

3.4.2 Verkkoratkaisut ja laitetyypit

Z-Waven verkkokerroksen spesifikaatio määrittelee laitteille kolme vastaanottotilaa: AL (Always Listening), FL (Frequenly Listening) sekä NL (Non-Listening) [126, s. 18-19]. AL-tyyppinen laite pitää radiotaan jatkuvasti päällä ja on verkossa reitittämiseen osallistuva laite. FL-tyyppinen laite voi tarvittaessa sulkea radionsa ja viettää suurimman osan ajastaan lepotilassa säästääkseen virtaa eikä siten osallistu reititykseen verkossa. NL-tilassa laite vastaavasti lähettää määritellyin väliajoin dataa tietylle verkon laitteelle. Z-Wave-verkon laitteet ovat joko ohjauskomentoja välittäviä kontrollereita tai slave-tyyppisiä komentoja suorittavia sekä havaintojaan lähettäviä päätelaitteita. Z-Wave-verkossa voi olla useita kontrollereita, joista yksi on ensisijainen kontrolleri ja muut toissijaisia [126, s. 17-18]. Ensisijainen kontrolleri vastaa verkon muodostuksesta ja sen ylläpidosta, uusien laitteiden liittämisestä sekä reititystietojen jakamisesta toissijaisille kontrollereille. Toissijaisten kontrollerien tehtävänä on reitittää informaatiota verkossa ja tarvittaessa liittää uusia laitteita verkkoon ensisijaisen kontrollerin ohjaamana. Kontrollerit voivat olla verkon topologian suhteen joko liikkuvia tai paikallaan olevia laitteita [124, s. 16-17]. Ensisijainen kontrolleri on hyvin usein paikallaan oleva Z-Wave-hub, joka toimii keskitettynä kommunikaatiopisteenä muille verkon laitteille ja on saavutettavissa internetin välityksellä. Päätelaitteiden tehtävänä on suorittaa kontrollereilta vastaanottamiaan komentoja tai ne voivat tarvittaessa lähettää havaintojaan rajalliselle määrälle muita verkon laitteita [124, s. 19-20]. Päätelaitteet ovat joko verkkovirtaan kytkettyjä reitittäviä laitteita tai yksinkertaisempia reitittämiseen kykenemättömiä paristokäyttöisiä laitteita.

Z-Wave perustuu verkkoratkaisuiltaan mesh-topologiaan [126]. Ensisijainen kontrolleri vastaa Z-Wave-verkon muodostuksesta ja luo verkolle satunnaisen 32-bittisen HomeID-verkkotunnisteen sekä jokaiselle verkkoon liittyvälle laitteelle

yksilöllisen 8-bittisen NodeID-tunnisteen [95, s. 70]. Samaan verkkoon kuuluvat laitteet jakavat saman HomeID-tunnisteen, jonka alla voi olla maksimissaan 232 Z-Wave-laitetta [126, s. 16]. Kommunikaatio Z-Wave-verkossa tapahtuu joko suoraan naapurien välillä NodeID-tunnisteiden perusteella tai useamman hypyn reititystä hyödyntämällä kontrollerien ylläpitämien reititystaulujen mukaisesti. Kaikki Z-Wave-verkon kontrollerit ylläpitävät listausta naapureistaan ja voivat tarvittaessa jakaa verkon topologiatietoja keskenään tai aloittaa reitin selvityksen, jos verkossa havaitaan reititysongelmia [124, s. 16-17; 126, s. 17].

3.4.3 Tietoturva

Z-Wave-laitteille on määritelty neljä turvallisuusluokkaa; S2 Access Control, S2 Authenticated, S2 Unauthenticated sekä S0 [125, s. 875]. S2 Access Control -luokka on turvallisin ja sitä edellytetään käytettäväksi kodin lukituslaitteissa sekä autotallin ovenavaajissa. S2 Authenticated on seuraavaksi turvallisin luokka, jota käytetään muun muassa kodin turvallisuusratkaisujen sensorilaitteissa, joiden käyttöönottaminen edellyttää laitteiden todennusta käyttäjän toimesta. S2 Unauthenticated on tarkoitettu yksinkertaisimmille Z-Wave-laitteille, kuten lampuille, joissa ei voida esittää laitteen todennuksessa tarvittavaa DSK (Device Specific Key) -laiteavainta. S0-luokka on ollut käytössä ennen S2-luokkien julkaisua markkinoille tuoduissa Z-Wave-laitteissa. S0-luokkaa käyttävät Z-Wave-laitteet ovat edelleen tuettuina S2 Unauthenticated -luokan kautta, jotta taaksepäin yhteensopivuus toteutuu myös aikaisempien Z-Wave-laitteiden osalta [46, s. 7-8].

Z-Wavessa datan salaus perustuu 128-bittiseen AES-CCM-menetelmään, jossa salausavaimena käytetään laitteelle liittymisprosessin yhteydessä luovutettua verkkoavainta [125, s. 849-850]. Verkkoavaimet ovat S2-turvallisuusluokkakohtaisia, jolla varmistetaan, että laitteet eivät voi purkaa korkeamman turvallisuusluokan laitteille tarkoitettuja viestejä [46, s. 8]. Laitteen käyttämä S2-turvallisuusluokka määritellään Z-Waven sovelluskerroksella, jonka myötä data säilyy päästä päähän salattuna. Yleensä Z-Wave-verkon ensisijainen kontrolleri vastaa verkkoavaimen luovuttamisesta uudelle laitteelle. Verkkoavaimen vaihtaminen laitteiden välillä perustuu ECDH (Elliptic Curve Diffie Hellman) -avaimenvaihtoon, jossa laitteet luovat toistensa julkisesta avaimesta johdetun väliaikaisavaimen salatakseen pysyvän verkkoavaimen siirron [125, s. 849]. S2 Access Control- sekä S2 Authenticated -luokkia käyttävien laitteiden osalta käyttäjän on annettava osa liitettävän laitteen julkisesta DSK-avaimesta kontrolleriin, joka vastaa laitteen liittämisestä verkkoon. DSK-avain

voi olla joko numerosarja tai QR (Quick Response) -koodi, jonka käytön on tarkoitus todentaa kontrollerille, että kyseessä on varmasti laite, jota käyttäjä on liittämässä verkkoon [125, s. 847-848, 1161]. S2 Unauthenticated -laitteiden osalta DSK-avain siirretään suoraan Z-Wave-laitteiden välillä ilman käyttäjän puuttumista liittymisprosessiin.

3.5 Yhteenveto WPAN-teknologioista

WPAN-teknologioiden tarjoamat ominaisuudet, kuten tiedonsiirtonopeudet, käytettävät taajuusalueet, tuetut verkkoratkaisut ja laitemäärät sekä laitteiden virrankulutus, tietoturvallisuus ja hinta, vaihtelevat jonkin verran eri teknologioiden välillä [69, 75]. Teknologioilla on näin ollen hyvät sekä huonot puolensa ja ne soveltuvat paremmin tietyn tyyppisiin tarkoituksiin. Taulukossa 3.6 on esitetty kotiautomaatioon soveltuvien WPAN-teknologioiden muutamia teknisiä ominaisuuksia.

Taulukko 3.6: Vähävirtaisten WPAN-teknologioiden vertailu [5, 63, 68, 89, 116].

	WiFi	**Bluetooth Low Energy**	**Zigbee**	**Z-Wave**	**Thread**
Standardi	IEEE 802.11ah	BLE	IEEE 802.15.4	ITU-T G.9959	IEEE 802.15.4
Taajuus	< 1 GHz	2.4 GHz	2.4 GHz, 868 ja 928 MHz	865 – 926 MHz	2.4 GHz
Salaus	WPA2/3	AES-128	AES-128	AES-128	AES-128
Tiedonsiirtonopeus	150 Kbit/s – 80 Mbit/s	125 Kbit/s – 2 Mbit/s	20 – 250 Kbit/s	9.6 – 100 Kbit/s	20 – 250 Kbit/s
Kantama	10 – 1000 m	10 – 400 m	10 – 100 m	30 – 100 m	10 – 100 m
Topologia	Tähti	Broadcast, Mesh, Tähti	Mesh, Puu, Tähti	Mesh, Tähti	Mesh
Laitemäärä verkossa	8191	32767	> 65000	232	250

Kotiautomaation langattomille teknologioille on yhteistä, että niihin perustuvat IoT-laitteet ovat vähän virtaa kuluttavia, yleensä paristokäyttöisiä sekä pienikokoisia laitteita ja siten helposti asennettavissa erilaisiin kohteisiin kotona. Laitteiden siirreltävyys ja useimpien WPAN-teknologioiden tuki mesh-pohjaiselle verkkoratkaisulle mahdollistavat osaltaan joustavien sekä hyvin skaalautuvien kotiautomaatioratkaisujen toteuttamisen eri tarpeisiin [57, s. 92]. WPAN-teknologiat jakavat samat langattomalle tiedonsiirrolle ominaiset haasteet, jotka ovat seurausta samalla taajuusalueella toimivien laitteiden toisilleen aiheuttamista häiriöistä sekä signaalin

etenemiseen ja vaimenemiseen vaikuttavista tekijöistä, kuten toimintataajuudesta, fyysisistä esteistä sekä heijastuksista [69, s. 19, 22]. Mahdolliset ongelmat langattomassa tiedonsiirrossa vaikuttavat IoT-laitteiden kantamaan sekä verkon toimintavarmuuteen ja näkyvät lopulta käyttäjälle IoT-ratkaisun epäluotettavana toimintana sekä palvelun laadun heikentymisenä.

WiFi-pohjaisten IoT-laitteiden sekä kehitysalustojen määrä on kasvanut markkinoilla nopeasti ja ne ovat nykyisin kaikkien kuluttajien saatavilla suhteellisen edullisesti [94]. Kuluttajan kannalta WiFin merkittävin etu on IoT-laitteiden käyttöönoton helppous ilman erillislaitteiden hankintaa, sillä kodin WiFi-verkkoon liitettävät IoT-laitteet ovat hallittavissa suoraan esimerkiksi älypuhelimella. Teknologiana WiFi on suhteellisen tietoturvallinen ja tarjoaa korkean tiedonsiirtonopeuden soveltuen siten enemmän kaistaa vaativiin ratkaisuihin, kuten äänen ja videon siirtoon [69, 75]. WiFin merkittävimmät haasteet kotiautomaation näkökulmasta ovat teknologian suuri virrankulutus, rajoittunut kantama, alhainen laitemäärä yksittäisessä verkossa sekä IoT-laitteiden osalta puuttuva tuki mesh-verkoille [69, 94]. Toisaalta IEEE 802.11ah -standardin mahdollistaman alhaisemman virrankulutuksen, suuremman laitemäärän sekä pidemmän kantaman myötä WiFi soveltuu tulevaisuudessa paremmin myös kotiautomaatioon [63, s. 67912].

Bluetoothin BR/EDR- sekä BLE-teknologiat ovat nykyisin lähes poikkeuksetta tuettuina markkinoilla olevissa mobiililaitteissa, jonka myötä Bluetooth-laitteet ovat nopeasti ja helposti käyttöönotettavissa [121, s. 19]. BR/EDR-teknologian merkittävimmät haasteet IoT-ratkaisuissa ovat johtuneet tähtitopologian huonosta skaalautuvuudesta, lyhyestä kantamasta sekä tuesta ainoastaan kahdeksalle laitteelle yhdessä verkossa [63, s. 67905]. BLE-meshin myötä tähän on kuitenkin tullut muutos, sillä teknologia mahdollistaa tuhansista vähävirtaisista BLE-laitteista rakentuvien ja kantamaltaan laajojen laiteverkkojen hyödyntämisen IoT-ratkaisuissa. Muista mesh-pohjaisista WPAN-teknologioista poiketen BLE-mesh hyödyntää taajuushyppelyä tiedonsiirron luotettavuuden parantamiseksi sekä tulvamenetelmää viestien välittämiseen verkossa. Tulvamenetelmän myötä BLE ei edellytä reititystietojen ylläpitoa, mikä yksinkertaistaa laitteiden välistä kommunikaatiota [121, s. 19], mutta saattaa toisaalta lisätä törmäysten määrää verkossa ja aiheuttaa siten pakettihäviöitä [72, s. 2094].

Zigbee ja Z-Wave ovat ominaisuuksiensa puolesta hyvin samankaltaisia teknologioita. Kummallakin on takanaan pitkä kehityshistoria ja kotiautomaatiomarkkinoilla onkin tarjolla erittäin paljon molempiin teknologioihin perustuvia IoT-laitteita

useisiin sovelluskohteisiin. Zigbee ja Z-Wave soveltuvat erinomaisesti sensoridatan keräämiseen sekä IoT-laitteiden ohjaamiseen, sillä ne ovat vähävirtaisia, nojaavat pienten datamäärien käsittelyyn ja alhaiseen tiedonsiirtonopeuteen sekä tukevat mesh-reititystä muodostaakseen luotettavasti toimivan laiteverkon [63, 69]. Teknisestä näkökulmasta yksi Z-Waven eduista on toimiminen alle 1 GHz:n taajuuksilla, jonka myötä verkon toimintavarmuutta ja kantamaa voidaan parantaa, kun signaali etenee materiaalien läpi korkeita taajuuksia paremmin. Toisaalta Z-Waven ei myöskään tarvitse kilpailla kaistasta ruuhkaisella 2.4 GHz:n taajuusalueella WiFin, BLE:n, Zigbeen sekä Threadin kanssa. Z-Wave-laitteiden tuominen markkinoille edellyttää aina sertifiointia, mikä näkyy kuluttajalle laitteiden korkeampana hintana verrattuna Zigbee-laitteisiin [69, s. 24], mutta toisaalta sertifiointi varmistaa yhteensopivuuden kaikkien Z-Wave-laitteiden välillä. Zigbeestä on vastaavasti ajan saatossa julkaistu useita eri spesifikaatioita sekä laiteprofiileja, mikä on aiheuttanut yhteensopivuushaasteita etenkin ennen Zigbee 3.0 -spesifikaatiota markkinoille tuotujen laitteiden välillä [75]. Kotiautomaatioratkaisuissa Zigbee ja Z-Wave edellyttävät aina erillisen reunalaitteen/hubin käyttämistä, sillä teknologiat eivät ole tuettuina esimerkiksi mobiililaitteissa, eivätkä ne tue natiivisti IP-pohjaista tiedonsiirtoa [46, s. 67915; 111, s. 302].

4 Thread

Thread on IEEE 802.15.4 -standardin päälle rakentuville IoT-laitteille suunniteltu alhaisen virrankulutuksen mesh-pohjainen verkkoprotokolla. Threadin kehittäminen on lähtenyt ajatuksesta, että tarvitaan helposti käyttöönotettava, toimintavarma, hyvin skaalautuva, tietoturvallinen sekä täysin IP-pohjainen verkkoteknologia, joka on yhteensopiva minkä tahansa sovelluskerroksen toteutuksen kanssa [105]. Thread ei ole Zigbeen tai Z-Waven tapaan uuden standardin määrittelevä teknologia, vaan Threadissa hyödynnetään tunnettuja IEEE- ja IETF (Internet Engineering Task Force) -standardeja [106]. Threadin kehittämisestä, spesifikaatioiden julkaisemisesta sekä sertifiointiohjelmista vastaa teknologiayritysten yhteenliittymä Thread Group, jonka perustivat alun perin ARM, Big Ass Fans, Freescale Semiconductor, Nest Labs (Google), Samsung Electronics, Silicon Labs sekä Yale Security [104]. Thread-laitteiden tuominen markkinoille edellyttää liittymistä Thread Groupin jäseneksi sekä hyväksyttyä laitesertifiointia. Kuluttajalle Thread näkyy kotiautomaatiomarkkinoilla tänä päivänä kahdentyyppisinä Thread-laitteina, jotka on merkitty kuvassa 4.1 esitetyillä logoilla. Logojen on tarkoitus viestiä kuluttajalle laitteiden olevan Thread-sertifioituja sekä yhteensopivia muiden valmistajien Thread-laitteiden kanssa ja tuoda selkeästi ilmi minkä tyyppinen Thread-laite on kyseessä.

Kuva 4.1: Thread-logot [101, s. 11].

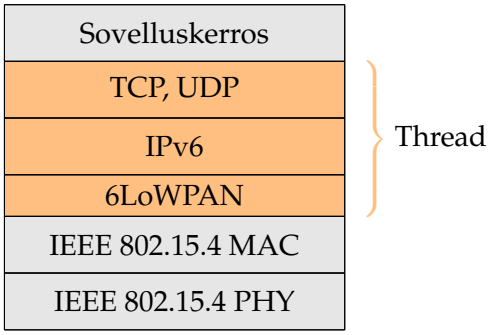
Thread Group julkisti ensimmäisen Thread 1.0 -version vuonna 2014. Thread 1.0 esiteltiin kotiautomaatioon suunnattuna vähävirtaisena mesh-verkkoprotokollana, joka rakentuu laitetasolla IEEE 802.15.4 -standardin päälle, hyödyntää 6LoWPAN (IPv6 over Low-Power WPAN) -sovituskerrosta natiivin IPv6-tuen saavuttamiseksi ja antaa mahdollisuuden käyttää sovelluskerroksella mitä tahansa sovellusratkaisua [106]. Thread 1.2 -versio julkaistiin vuonna 2019 ja se toi mukanaan parannuksia Thread-laitteiden virrankulutukseen optimoimalla nukkuvien laitteiden lähetystehoa sekä esitteli uuden laitetyypin SSED:n (Synchronized Sleepy End Device) [97].

Lisäksi 1.2-versio mahdollisti Bluetoothin käyttämisen vaihtoehtoisena kommunikointimenetelmänä Thread-laitteiden kanssa sekä rinnakkaisten Thread-verkkojen hyödyntämisen laajemmissa rakennusautomaation IoT-ratkaisuissa [98]. Threadin 1.3-versio julkaistiin vuonna 2022, joka toi mukanaan yhteensopivuuden Matterille, standardisoi Thread-reunareitittimien toteutuksen Thread-laitteiden saavutettavuuden parantamiseksi ja lisäsi Thread-laitteisiin tuen TCP (Transmission Control Protocol) -protokollalle tehostamaan laiteohjelmistojen päivitystä sekä parantamaan tiedonsiirron luotettavuutta [99].

Google on ollut alusta asti mukana Threadin kehittämisessä ja nopeuttaakseen Threadin leviämistä, Google julkaisi vuonna 2016 OpenThreadin. OpenThread [25] on avoimen lähdekoodin versio Threadista ja se mahdollistaa kehittäjille helpon tavan päästä testaamaan Threadia useiden eri valmistajien piiriratkaisuilla sekä kehitysalustoilla [47]. OpenThread toteuttaa Threadin virrallisen spesifikaation määritykset, joten siihen pohjautuvat laitteet ovat tuotavissa markkinoille sellaisenaan, mutta valmistajan on joka tapauksessa liityttävä Thread Groupin jäseneksi ja sertifioitava myös OpenThread-pohjaiset laitteet [103].

4.1 Threadin kerrosmalli

Kuvassa 4.2 on esitetty Threadin kerrosmalli. Thread rakentuu tunnettuista standardeista sekä protokollista jättäen sovelluskerroksen kokonaan vapaaksi [105]. Kuljetuskerroksella Thread hyödyntää UDP (User Datagram Protocol) -protokollaa ja viestien välitys Thread-laitteiden välillä perustuu CoAP (Constrained Application Protocol) -protokollaan, joka on varta vasten resurssirajoittuneille IoT-laitteille kehitetty viestinvälitysprotokolla. Threadin 1.3-version myötä Thread-laitteet tukevat myös TCP-protokollaa, jonka on tarkoitus parantaa tiedonsiirron luotettavuutta sitä edellytettävissä sovellusratkaisuissa. Verkkokerroksella Thread nojaa täysin IPv6-pohjaiseen tiedonsiirtoon, jonka myötä Thread-laitteet ovat tarvittaessa saavutettavissa myös internetistä. IPv6-tuki on Threadissa toteutettu resurssirajoittuneille IoT-laitteille standardisoidulla 6LoWPAN-sovituskerroksella [108]. Fyysisellä kerroksella sekä linkkikerroksella Thread rakentuu IEEE 802.15.4 -standardin päälle, mikä mahdollistaa Threadin hyödyntämisen useiden eri valmistajien markkinoille tuomilla piireillä sekä kehitysalustoilla.



Kuva 4.2: Threadin kerrosmalli, muokattu [105, s. 5].

4.1.1 Fyysinen kerros ja linkkikerros

IEEE 802.15.4 -standardin päälle rakentuvana protokollana Thread jakaa Zigbeen kanssa samat menetelmät kanavalle pääsyn, kuittausten sekä uudelleenlähetysten osalta, joita käytiin läpi luvussa 3.3.1. Vaikka IEEE 802.15.4 -standardi tukee useita taajuusalueita sekä tiedonsiirtonopeuksia, Thread-laitteet toimivat ainoastaan 2.4 GHz:n taajuusalueella, jonka myötä tiedonsiirtonopeus on 250 Kbit/s [106]. Linkkikerroksella Thread hyödyntää muutamia erityisiä tekniikoita paristokäyttöisten lepotilaa hyödyntävien Thread-laitteiden (Sleepy End Device, SED) virrankulutuksen alentamiseksi sekä saavutettavuuden parantamiseksi [97]. Yksi näistä on EFP (Enhanced Frame Pending), jolla voidaan vähentää tiedonsiirtoa SED-laitteen ja reitittimen (parent) välillä. EFP:n myötä SED-laitteen ei tarvitse erikseen kysyä parent-reitittimeltään onko sillä viestejä laitteelle, vaan SED:n lähettäessä muuta dataa reitittimelle, se saa tiedon mahdollisista viesteistä reitittimen palauttamassa kuittauksessa. Toinen SED-laitteiden virrankulutusta alentava menetelmä on IEEE 802.15.4 -standardiin sisältyvä CSL (Coordinated Sampled Listening), joka mahdollistaa SED:n toimimisen aikasynkronissa parent-reitittimensä kanssa. Aikasynkronoinnin myötä SSED (Synchronized SED) -laite voi herätä määritellyin aikavälein vastaanottamaan parent-reitittimeltään sille osoitetut viestit ja olla muun ajan lepotilassa. Aikasynkronointi on erityisen hyödyllinen esimerkiksi aktuaattori-laitteissa, joiden on vastaanotettava komentoja säännöllisesti, mutta laitteen virrankulutus on kuitenkin pyrittävä pitämään mahdollisimman alhaisena [97, s. 5].

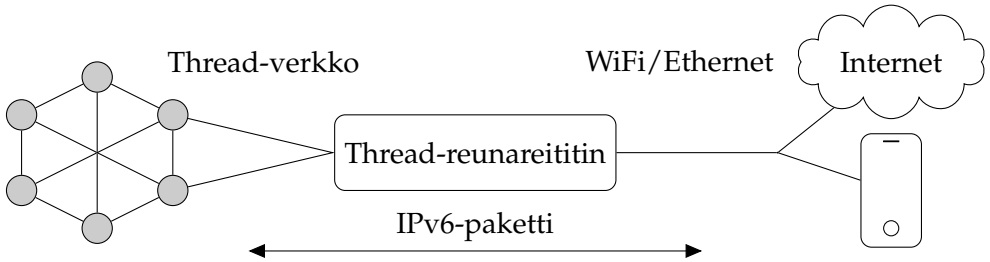
Thread-laitteet hyödyntävät myös useita erilaisia linkkimittareita laitteiden välisten linkkien laadun selvittämiseen. Linkkimittarien on tarkoitus parantaa laitteiden pariston kestoa, yhteyksien luotettavuutta sekä koko Thread-verkon toimintavarmuutta. Linkkimittareina hyödynnetään joko viestimääriä, linkkikustannuksia

tai RSSI- (Received Signal Strength Indicator) ja LQI (Link Quality Indicator) -arvoja [97, s. 6]. Linkkimittarien perusteella Thread-laitteet pystyvät optimoimaan lähetystehoaan, jonka alentaminen parantaa laitteen pariston kestoa sekä vähentää piilossa olevien noodien ongelmaa, kun laitteet eivät turhaan häiritse toistensa kommunikaatiota verkossa.

Threadissa datan salaus perustuu IEEE 802.15.4 -linkkikerroksella 128-bittiseen AES-CCM-menetelmään ja laitekomissioinnin yhteydessä Thread-laitteille jaettuun verkkoavaimeen [102]. Thread edellyttää kaiken Thread-laitteiden välisen kommunikaation salaamista, minkä lisäksi linkkikerroksella hyödynnetään kehyslaskureita naapurilaitteiden välillä [46, s. 10]. Linkkikerroksen suojausmenetelmillä pyritään suojautumaan muun muassa salakuuntelua sekä toistohyökkäyksiä vastaan eikä verkkoavainta tietämätön laite voi esiintyä Thread-verkossa luotettuna laitteena [46, s. 10]. Threadin muista tietoturvamenetelmistä, laitekomissioinnista sekä verkkoavaimen jakamisesta on kerrottu enemmän luvussa 4.3.

4.1.2 Verkko- ja kuljetuskerros

Verkkokerroksella Thread perustuu IPv6-protokollaan [108], jonka myötä Thread on täysin yhteensopiva muiden IPv6-pohjaisten tiedonsiirtoteknologioiden kanssa. Kuvassa 4.3 on esitetty esimerkinomaisesti IP-pohjaisuuden hyöty Threadissa. IEEE 802.15.4 -standardiin perustuvana teknologiana Thread edellyttää vähintään yhden Thread-reunareitittimen käyttämistä, jotta kommunikaatio Thread-verkkoon sekä toisen teknologian IP-verkkoon ja tarvittaessa internetiin on mahdollista [101]. Threadin IP-pohjaisuus mahdollistaa kuitenkin sen, ettei Thread-reunareitittimissä tarvitse tehdä Zigbeen ja Z-Waven tapaista protokollamuunnosta tai pakettien salauksen purkamista, mikä alentaa kommunikaatioviivettä sekä mahdollistaa päästä päähän -salauksen toteutumisen.

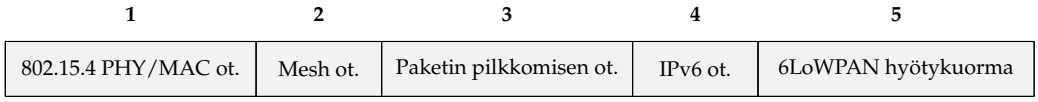


Kuva 4.3: Threadin IP-pohjaisuus, muokattu [107, s. 8].

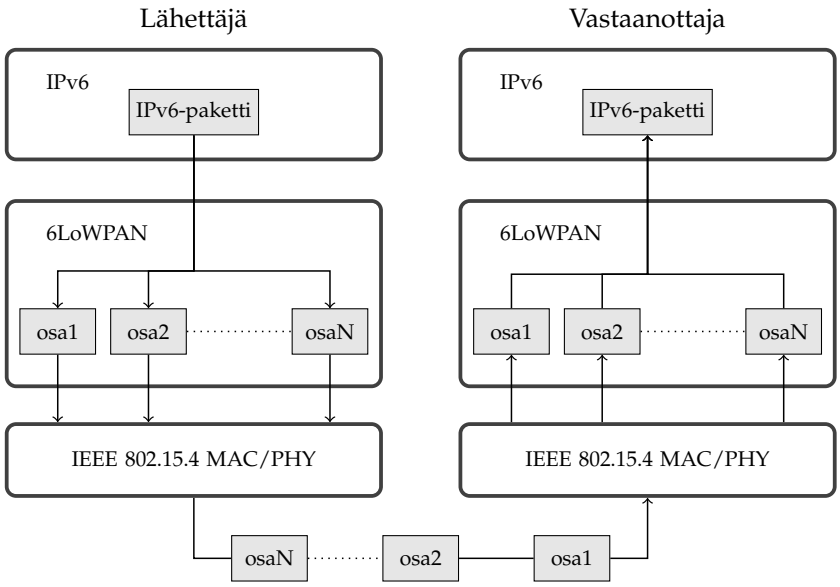
Kuljetuskerroksella Thread käyttää yhteydetöntä UDP-protokollaa, jonka luotettavuutta voidaan parantaa linkki- ja sovelluskerroksen uudelleenlähetyksillä [106, s. 15]. Viestien välityksessä Thread-laitteiden kesken UDP:n päällä hyödynnetään resurssirajoittuneille IoT-laitteille soveltuvaa CoAP-protokollaa. CoAP-viestejä käytetään muun muassa Thread-verkon muodostuksessa ja ylläpidossa, laitekomissioinnissa sekä reititystietojen vaihtamisessa [46, s. 9; 102]. Lisäksi UDP:n päällä hyödynnetään DTLS (Datagram Transport Layer Security) -protokollaa, jota käytetään muodostamaan suojattu istunto rekisteröitäessä Thread-verkon komissioija sekä komissioitaessa uusia laitteita Thread-verkkoon [102]. Thread tukee UDP:n rinnalla myös TCP-protokollaa, jonka on tarkoitus parantaa tiedonsiirron luotettavuutta ja mahdollistaa Thread-laitteiden ohjelmistojen päivittäminen tehokkaammin [99]. TCP-tuki on Threadissa toteutettu TCPlp (TCP Low Power) -protokollalla, joka on IEEE 802.15.4 -standardiin perustuville laitteille kehitetty kevyempi versio perinteisestä TCP-protokollasta [48].

IP-pohjaisuus on yksi Threadin merkittävimmistä eduista verrattaessa sitä muihin alhaisen virrankulutuksen WPAN-teknologioihin. Thread-laitteissa ei ole kuitenkaan mahdollista ottaa IPv6-protokollaa käyttöön sellaisenaan, sillä IPv6 edellyttää siirtoyksikön (Maximum Transmission Unit, MTU) pakettikooksi 1280 tavua ja IEEE 802.15.4 -standardi tukee ainoastaan 127 tavun paketteja [81, s. 28]. Jotta IPv6-protokollan hyödyntäminen resurssirajoittuneilla IoT-laitteilla olisi mahdollista, on tarkoitukseen kehitetty 6LoWPAN-sovituskerros [81, s. 16], joka toimii IEEE 802.15.4 -linkkikerroksen ja IPv6-verkkokerroksen välillä. 6LoWPAN-sovituskerros mahdollistaa IPv6- ja UDP-otsikkotietojen pakkaamisen, IPv6-paketin muodostamisen, IPv6-paketin pilkkomisen lähettäjällä ja sen koostamisen vastaanottajalla sekä IPv6-pakettien välittämisen useamman hypyn mesh-verkossa [108]. Kuvassa 4.4 on esitetty esimerkki 6LoWPAN-paketin rakenteesta. Paketti alkaa (1) IEEE 802.15.4 -otsikkotiedoilla, joihin sisältyy muun muassa verkon PAN ID, lähde- ja kohdelaitteen osoite sekä kehyslaskurit. Paketin mesh-otsikkotiedot (2) sisältävät tiedot paketin välittämisestä mesh-verkon sisällä useamman hypyn kommunikaatiolla IEEE 802.15.4 -linkkikerroksella [81, s. 38-40]. Mesh-otsikkotietoihin sisältyy jäljellä olevien hyppyjen määrä sekä paketin lähettäjän ja lopullisen vastaanottajan osoite. Mesh-otsikkotietoja seuraa IPv6-paketin pilkkomisen otsikkotiedot (3), joiden perusteella IPv6-paketti voidaan koostaa oikein vastaanottajalla [108, s. 12]. IPv6-otsikkotiedot (4) sisältää pakatut IPv6- sekä UDP-otsikkotiedot [108, s. 6] ja paketin loppuosa (5) muodostuu IPv6-paketin hyötykuormasta. Kuvassa 4.5 on

esitetty 6LoWPAN-protokollan toimintaperiaate. 6LoWPAN-sovituskerros pilkkoo verkkokerrokselta vastaanottamansa IPv6-paketin osiin lähettäjällä ja koostaa sen vastaanottajalla. Ensimmäinen 6LoWPAN-paketti (*osa1*) rakentuu kuvan 4.4 mukaisesti sisältäen osan IPv6-paketin hyötykuormasta. Muut osat (*osa2/N*) sisältävät loput IPv6-paketin hyötykuormasta ja pakettien rakenne vastaa muuten ensimmäistä pakettia (*osa1*), mutta paketit eivät sisällä kuvan 4.4 mukaisia IPv6-otsikkotietoja (4).



Kuva 4.4: 6LoWPAN-paketin rakenne, muokattu [108, s. 5].



Kuva 4.5: 6LoWPAN-protokollan toimintaperiaate, muokattu [108, s. 11].

4.2 Verkon arkkitehtuuri ja laitetyypit

Thread-verkon topologia riippuu reitittimien määrästä verkossa. Jos verkossa on ainoastaan yksi Thread-reititin, verkko muodostaa tähtitopologian ja useammalla reitittimellä aina mesh-topologian [106, s. 8]. Thread-verkossa voi toimia samanaikaisesti 32 aktiivista reititintä [105, s. 12] ja teoriassa jokaiseen verkon sisäiseen reitittimeen voi olla liittyneenä 511 päätelaitetta [28]. Mesh-pohjaisena ratkaisuna Thread-verkko on itsemuodostuva ja itsekorjautuva, mikä toteutuu dynaamisesti

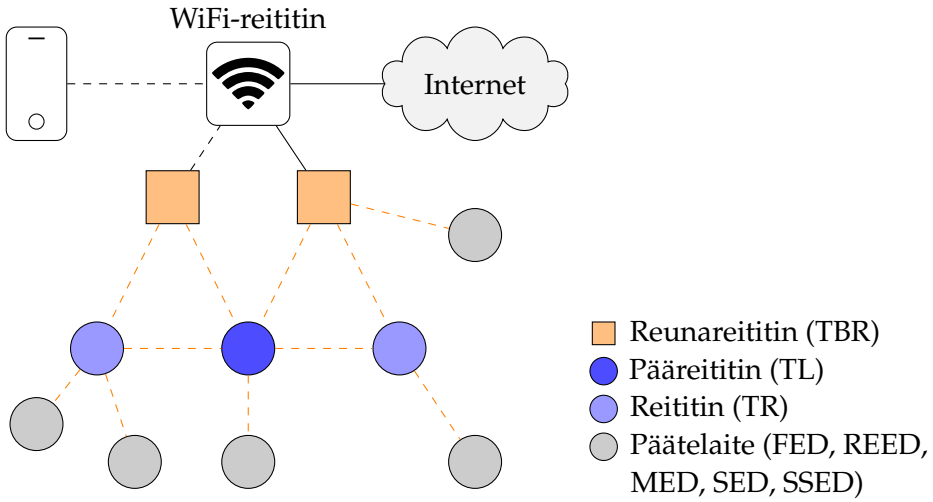
tilaansa vaihtavien reitittimien sekä tehokkaan reitityksen myötä. Thread-verkossa jokainen reititin ylläpitää reititystaulua ja reitittimet vaihtavat säännöllisesti reititystietoja keskenään, jotta verkko säilyy kytkettynä ja toimintakuntoisena myös topologian muuttuessa [105, s. 13]. Reitittimillä on lisäksi kyky säilyttää muistissaan lepotilassa oleville päätelaitteille osoitettuja viestejä [74, s. 292], sillä alhaisen virrankulutuksen teknologiana Thread-verkko koostuu merkittäviltä osin myös lepotilaa hyödyntävistä paristokäyttöisistä päätelaitteista.

Kuvassa 4.6 on esitetty Thread-verkon arkkitehtuuri sekä Thread-verkossa toimivat laitetyypit. Threadin laitetyypit jakautuvat kahteen pääluokkaan: Full Thread Device (FTD) sekä Minimal Thread Device (MTD) [105, s. 7]. FTD-laitteet ovat monipuolisimpia sekä suorituskykyisimpiä Thread-laitteita, jotka pitävät radiotaan jatkuvasti päällä ja ovat näin ollen verkkovirtaan kytkettyjä. Thread-verkossa kaikki reitittämiseen kykenevät laitteet ovat FTD-tyyppisiä, joiden tehtävänä on ylläpitää ja vaihtaa reititystietoja, vastaanottaa verkon sisäisiä ryhmälähetysviestejä, laajentaa mesh-verkkoa sekä toimia parent-reitittimenä päätelaitteille (child-laite) [28,64]. Thread-verkossa FTD-laitteita ovat

- reunareitin (Thread Border Router, TBR)
- pääreititin (Thread Leader, TL)
- verkon sisäinen reititin (Thread Router, TR)
- reitittävä päätelaite (Router Eligible End Device, REED)
- reitittämiseen kykenemätön päätelaite (Full End Device, FED).

MTD-tyyppiset Thread-laitteet ovat vastaavasti yleensä paristokäyttöisiä päätelaitteita, kuten sensoreita tai aktuaattoreita. MTD-laitteet ovatkin pääosin lepotilassa ja ne voivat kommunikoida ainoastaan FTD-tyyppisen parent-reitittimensä kanssa [28,64]. Thread-verkon MTD-tyyppisiä laitteita ovat

- radiotaan aina päällä pitävä päätelaite (Minimal End Device, MED)
- nukkuva päätelaite (Sleepy End Device, SED)
- aikasynkronissa oleva nukkuva päätelaite (Synchronized SED, SSED).



Kuva 4.6: Thread-verkon arkkitehtuuri, muokattu [105, s. 12].

4.2.1 Thread-reunareititin

Thread-verkko liittyy kodin WiFi- tai Ethernet-verkkoon Thread-reunareitittimen (TBR) kautta, joka toimii rajapintalaitteena eri verkkoteknologioiden välillä [101]. Zigbeestä ja Z-Wavesta poiketen Thread ei edellytä keskitettyä hub-laitetta, vaan Thread tukee usean TBR:n käyttöä yhdessä Thread-verkossa, jotta älykotiverkon toimintavarmuutta voidaan parantaa eikä Thread-verkkoon muodostuisi yhtä vikaherkkää pistettä [110, s. 162]. Threadin ollessa täysin IPv6-pohjainen teknologia, ei TBR-reitittimissä tarvitse tehdä protokollamuunnoksia tai purkaa pakettien salausta, vaan TBR voi välittää IPv6-paketin sellaisenaan läpi. TBR:n toiminnallisuus on näin ollen toteutettavissa hyvin erityyppisiin kodin laitteisiin, kuten televisioihin tai älykaiuttimiin [96, s. 5], kunhan laite sisältää asianmukaiset verkkoyhteydet.

Toiminnallisesti TBR vastaa reitityksestä IP-verkkojen välillä, palvelurekisterien ylläpidosta ja palveluiden mainostamisesta [105, s. 8] sekä ulkoisen komissioijalaitteen rekisteröimisestä Thread-verkkoon [102]. IPv6-reitityksen osalta TBR:n tehtävänä on muodostaa OMR (Off-Mesh Routable) -etuliite Thread-verkolle, jonka se saa DHCPv6 (Dynamic Host Configuration Protocol version 6) -protokollalla rinnakkaisen verkon reitittimiltä tai TBR muodostaa OMR-etuliitteen Thread-verkon XPAN ID (Extended PAN ID) -tunnistetta hyödyntämällä [99, s. 6-7]. TBR mainostaa Thread-verkon OMR-etuliitettä rinnakkaiseen verkkoon, jonka perusteella rinnakkaisen verkon laitteet tietävät minkä TBR:n kautta kyseisen IPv6-etuliitteen omaavat Thread-laitteet ovat saavutettavissa [99, s. 7-8]. Vastaavasti Thread-laitteet

voivat saavuttaa rinnakkaisen verkon laitteita TBR-reitittimien Thread-verkkoon jakamien IPv6-etuliitteiden kautta [99, s. 8]. Reitityksessä käytettävien osoitteiden määrittämisen lisäksi TBR-reitittimien tulee toteuttaa NAT64-toiminnallisuus, jotta Thread-laitteet voivat kommunikoida tarvittaessa myös IPv4-verkon laitteiden kanssa [98, s.19]. Thread käyttää laitteilla olevien palveluiden mainostamiseen ja etsimiseen DNS-SD (Domain Name System Service Discovery) -protokollaa [99, s. 8]. Palveluiden etsiminen esimerkiksi WiFi-verkossa perustuu yleisesti mDNS (multicast DNS) -kyselyn lähettämiseen kaikille verkon laitteille. Thread-verkon laitteet eivät kuitenkaan suoraan hyödynnä mDNS-kyselyjä, sillä niiden välittäminen ryhmälähetysviesteinä olisi Thread-verkon sisällä tiedonsiirron kannalta liian kallista, eivätkä lepotilaa hyödyntävät Thread-laitteet ole jatkuvasti saavutettavissa. Tästä johtuen TBR-reitittimet ylläpitävät SRP (Service Registration Protocol) -rekisteriä, johon Thread-verkon laitteilla olevat palvelut rekisteröidään [99, s. 9]. TBR toimii DNS-SD-välityspalvelimena verkkojen välillä ja vastaa Thread-laitteiden puolesta rinnakkaisesta verkosta tuleviin mDNS-kyselyihin. Vastaavasti Thread-laitteen etsiessä rinnakkaisen verkon palveluita, se lähettää DNS-SD-kyselyn unicast-viestinä TBR:lle, joka välittää pyynnön mDNS-kyselynä eteenpäin ja palauttaa vastauksen unicast-viestinä Thread-laitteelle [100, s. 24-25].

4.2.2 Reitittävät Thread-laitteet

Thread-verkon sisäisten reitittimien (Thread Router, TR) tehtävänä on ylläpitää reittejä verkossa, laajentaa mesh-verkkoa, toimia parent-reitittimenä päätelaitteille sekä mahdollistaa uusien laitteiden liittyminen Thread-verkkoon [47, s. 6]. Jokaisessa Thread-verkossa on aina yksi pääreititin (Thread Leader, TL), joka vastaa reititinroolien ylläpidosta, verkkomääritysten jakamisesta TR-reitittimille sekä komissioijan hyväksymisestä verkkoon [105, s. 6-7; 110, s. 165]. Lisäksi pääreitittimen tehtävänä on muodostaa linkkikerroksen salauksessa käytettävä verkkoavain (Master Key, MK) [46, s. 10]. Pääreititin valitaan automaattisesti verkon muodostuksen yhteydessä ja jos se putoaa pois verkosta, ottaa jokin muu reititin pääreitittimen tehtävän verkossa [106, s. 5]. Tämä on mahdollista, sillä jokaisen Thread-verkon reitittimen tulee toteuttaa sama toiminnallisuus, jonka myötä niillä on tiedossaan samat verkon konfigurointitiedot. Verkon topologiasta riippuen pääreitittimenä voi toimia Thread-reunareititin, joka voi myös olla Thread-verkon ainoa reititin, johon päätelaitteet yhdistyvät.

FTD-tyyppisistä päätelaitteista REED (Router Eligible End Device) on reitittämiseen kykenevä laite, mutta se liittyy Thread-verkkoon aina ensin tavallisena päätelaitteena. REED voi tarvittaessa päivittää tilansa TR-reitittimeksi, jos esimerkiksi uudella verkkoon liittyvällä laitteella ei ole reititintä kantamansa alueella [105, s. 17]. Thread pyrkii pitämään aktiivisten reitittimien määrän 16 – 23 välillä ja jos REED:n liittyessä verkkoon reitittimiä on vähemmän kuin 16, päivittää REED tilansa TR-reitittimeksi [28]. Vastaavasti TR-reititin voi alentaa tilansa REED-laitteeksi [105, s. 7], jos sillä ei ole child-laitteita ja verkossa on topologian kannalta optimaalinen määrä reitittimiä. REED:n ja TR:n tilan vaihtamisesta päättää kuitenkin aina Thread-verkon pääreititin [106, s. 14], jotta vältetään reitittimien päällekkäinen osoitteistus verkossa.

4.2.3 Thread-päätelaitteet

Thread-verkossa päätelaitteet ovat aina child-laitteita, jotka voivat kommunikoida ainoastaan oman parent-reitittimensä kanssa eikä niillä näin ollen ole kykyä välittää muiden laitteiden viestejä verkossa [106, s. 9]. FTD-tyyppisistä päätelaitteista FED (Full End Device) on REED-laitteen kaltainen reititystietoja ylläpitävä laite, mutta se ei voi koskaan toimia reitittimenä. Reititystietoja ylläpitävänä FED pystyy kuitenkin selvittämään reittejä Thread-verkossa ja vähentää siten parent-reitittimensä kuormaa [64]. MTD-tyyppisistä päätelaitteista MED (Minimal End Device) on laitetyyppi, joka pitää radiotaan jatkuvasti päällä. Kommunikointi parent-reitittimen kanssa on näin ollen kaksisuuntaista ja viiveetöntä, ilman erillisiä yhteyden avauksia tai aikasynkronointeja. Muita MTD-tyyppisiä päätelaitteita ovat paristokäyttöiset SED (Sleepy End Device) sekä SSED (Synchronized SED) [105, s. 8]. SED on pääosin lepotilassa ja herää määritellyin väliajoin kysymään parent-reitittimeltään, onko sillä muistissaan laitteelle osoitettuja viestejä. Toinen vaihtoehto viestien vastaanottamiseen parent-reitittimeltä on luvussa 4.1.1 sivuttu EFP-menetelmä, jota hyödyntämällä SED saa tiedon saapuneista viesteistä parent-reitittimen lähettämässä kuittauksessa. SSED on vastaavasti aikasynkronissa parent-reitittimensä kanssa ja vastaanottaa sille saapuneet viestit yhteisen aikaikkunan mukaisesti, mikä alentaa viivettä sekä virrankulutusta verrattuna SED-laitteeseen [64]. Jotta Thread-verkon itsekorjautuvuus toteutuu myös päätelaitteiden osalta, päätelaite pyrkii automaattisesti löytämään verkosta uuden reitittimen itselleen, jos se menettää yhteyden omaan parent-reitittimeensä [46, s. 9].

4.3 Thread-verkosta yleisesti

Thread-verkon itsemuodostuvuus ja -korjautuvuus sekä usean hypyn kommunikaatio toteutuvat reitittimien vaihtamien tilojen ja automaattisesti reitittimien välillä päivittyvien reititystietojen myötä. Thread-verkon osioituminen on myös yksi menetelmä, jolla verkon itsemuodostuvuus toteutuu. Thread-verkko voi osioitua, jos osa verkon reitittimistä menettää yhteyden muihin reitittimiin tai pääreititin putoaa verkosta [110, s. 164]. Jos Thread-verkko osioituu, on osioilla aina oma pääreititin sekä omat verkon tunnistetiedot, mutta osiot jakavat edelleen muun muassa alkuperäisen Thread-verkon verkkoavaimen. Osiot voivat näin ollen yhdistyä myöhemmin yhdeksi Thread-verkoksi, jos reitittimien yhteys osioiden välillä palautuu [28].

4.3.1 Verkon muodostus ja reititys

Samassa Thread-verkossa toimivat laitteet jakavat yhteisesti muun muassa saman IEEE 802.15.4 -kanavan, mesh-verkon sisäisen IPv6-etuliitteen sekä salauksessa käytettävän verkkoavaimen [34, s. 4; 102]. Lisäksi laitteet jakavat samat Thread-verkon tunnistetiedot, kuten 16-bittisen PAN ID:n, 64-bittisen XPAN ID:n (Extended PAN ID) sekä selkokielisen verkkonimen, joiden perusteella Thread-verkot erotetaan toisistaan [27]. Uuden Thread-laitteen liittyminen verkkoon alkaa olemassa olevien Thread-verkkojen etsimisellä, jota seuraa laitekomissiointi ja lopulta yhteyden muodostus parent-reitittimeen sekä liittyminen verkkoon [105, s. 16-17]. Thread-laite aloittaa liittymisen Thread-verkkoon lähettämällä liittymispyyntöjä jokaisella käytettävissä olevalla IEEE 802.15.4 -kanavalla [27; 34, s. 3]. Laite saa liittymispyynnön kuulleilta Thread-verkon reitittimiltä vastauksena kyseisen Thread-verkon tunnistetiedot, jonka jälkeen laite päättää mihin verkkoon se haluaa liittyä. Jos uusi Thread-laite on reitittämiseen kykenevä, se voi tarvittaessa muodostaa myös uuden Thread-verkon ja valita itsensä verkon pääreitittimeksi. Jokainen Thread-laite liittyy olemassa olevaan verkkoon kuitenkin aina ensin päätelaitteena, jonka jälkeen reitittävä laite voi pyytää pääreitittimeltä lupaa vaihtaa tilansa varsinaiseksi reitittimeksi [105, s. 17]. Sopivan verkon löytämisen jälkeen uuden Thread-laitteen on läpäistävä komissiointiprosessi, jonka yhteydessä laitteelle luovutetaan salauksessa käytettävä verkkoavain. Onnistuneen komissioinnin jälkeen liittyvän laitteen täytyy vielä muodostaa yhteys parent-reitittimeen, jotta laite saa tietoonsa muun muassa verkon IPv6-etuliitteet, pääreitittimen osoitteen ja sille muodostuu omat IPv6-osoitteet [27].

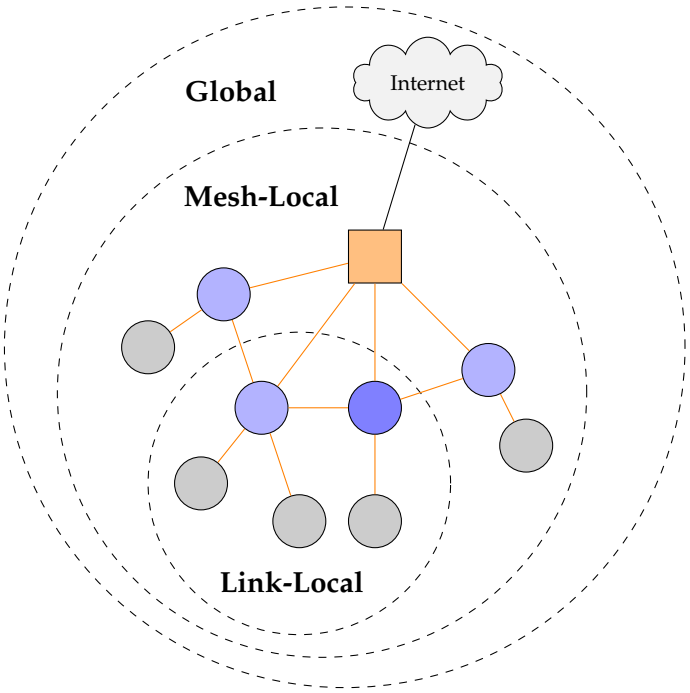
Thread-verkon sisäiseen konfigurointiin ja ylläpitoon liittyvässä kommunikaatiossa hyödynnetään MLE (Mesh Link Establishment) -protokollaa [53, s. 4116]. MLE-protokollan mukaisia MLE-viestejä käytetään muun muassa laitteen liittämisen yhteydessä sekä osoite- ja reititystietojen, linkin laadun ja kustannusten, salausavainten sekä kehyslaskurien jakamiseen ja päivittämiseen Thread-laitteiden välillä [27, 105]. Kaikki MLE-viestit salataan Thread-verkkoon komissioitujen laitteiden välillä vastaavalla AES-CCM-menetelmällä kuin linkkikerroksellakin, mutta erillisellä salausavaimella [53, s. 4118]. MLE-viestien yhteydessä hyödynnetään myös MPL (Multicast Protocol for Low-Power and Lossy Networks) -protokollaa ryhmälähetykseen Thread-laitteiden välillä esimerkiksi silloin, kun reitittimet haluavat saada tietoonsa reittejä naapurireitittimiin sekä jaettaessa verkon konfigurointitietoja muille reitittimille [29; 105, s. 13].

Reititystietojen muodostaminen Thread-verkossa perustuu etäisyysvektoriprotokollaan, jossa usean hypyn reittikustannus muodostuu RSSI-arvosta johdetusta linkkikustannuksesta naapurireitittimien välillä [105, s. 13-14]. Thread-laitteet hyödyntävät aina parasta mahdollista reittiä kohdelaitteeseen, jonka yhteenlaskettu kustannus on alhaisin. Thread-verkon reitittimet vaihtavat reititystietoja säännöllisesti keskenään ilman erillisiä pyyntöjä, jotta niillä on tiedossaan optimaaliset reitit verkossa ja verkon topologian muutoksiin voidaan reagoida nopeasti. Reitittimien toisilleen mainostamat MLE-viestit sisältävät tiedon yhden hypyn linkkikustannuksesta naapureiden välillä sekä reititystiedot muihin verkon reitittimiin, joita reitittimet ylläpitävät naapuri- sekä reititystauluissaan [106, s. 13-14].

4.3.2 Osoitteistus

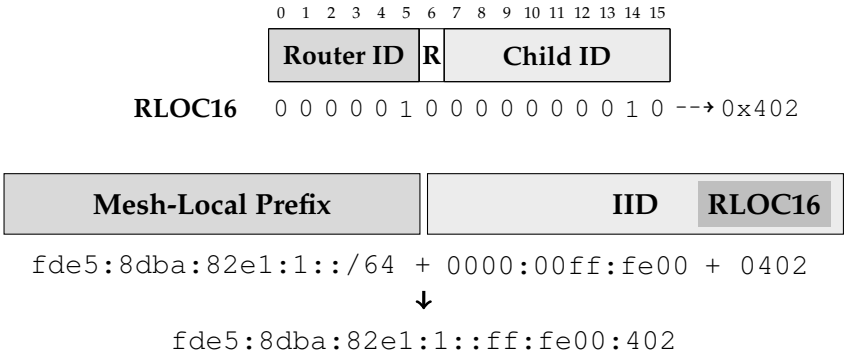
IPv6-protokollan myötä Thread-laitteilla on useita unicast-tyyppisiä IPv6-osoitteita. Kuvassa 4.7 on esitetty kolme Threadissa käytössä olevaa IPv6-osoitteiden ulottuvuutta. Global-osoitteet (Global Unicast Address, GUA) ovat Thread-verkon ulkopuolisia julkisia osoitteita, joiden kautta Thread-laitteet ovat saavutettavissa muista IPv6-verkoista [26]. GUA-osoitteella on aina etuliite `2000::/3` ja Thread-laite muodostaa GUA-osoitteensa Thread-reunareitittimien jakamien IPv6-etuliitteiden perusteella, joko SLAAC (Stateless Address Autoconfiguration) -menetelmää tai DHCPv6-protokollaa hyödyntämällä [105, s. 10]. Mesh-Local-osoitteet ovat vastaavasti Thread-verkon sisällä saavutettavia osoitteita, jotka alkavat aina etuliitteellä `fd00::/8`. Thread-laitteella on kaksi Mesh-Local-osoitetta: topologiasta riippuvainen RLOC (Routing Locator) sekä muuttumaton ML-EID (Mesh-Local Endpoint

Identifier). Link-Local-osoitteet (Link-Local Address, LLA) ovat käytössä yhden hypyn kommunikaatiossa ja osoitteet alkavat etuliitteellä `fe80::/16`. LLA-osoitteet eivät ole reititettäviä osoitteita ja niitä käytetään esimerkiksi naapureiden etsimiseen sekä reititystietojen jakamiseen [26].



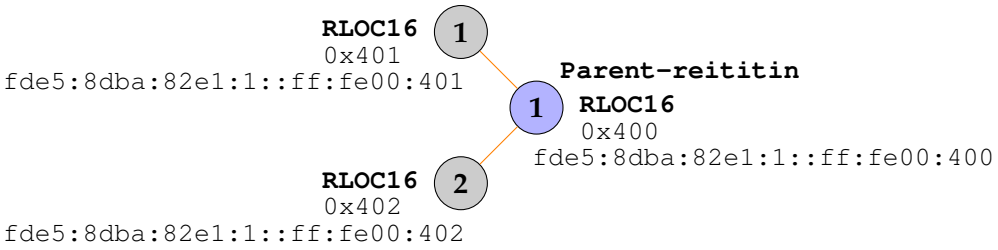
Kuva 4.7: Thread-verkon IPv6-ulottuvuudet, muokattu [26].

IPv6-osoitteet ovat 128-bittisiä ja koostuvat Threadissa 64-bittisestä etuliitteestä (IPv6 prefix) sekä 64-bittisestä IID (Interface Identifier) -tunnisteesta. IP-reititys Thread-verkossa perustuu reititystauluun, joka sisältää jokaisen verkon reitittimen osoitteen pakatussa muodossa (RLOC16) sekä tiedon osoitetta vastaavasta seuraavan hypyn reitittimestä [105, s. 15]. Thread-laitteen liittyessä verkkoon sille muodostetaan mesh-verkon sisäinen 128-bittinen RLOC-osoite, joka kuvaa laitteen sijaintia suhteessa Thread-verkon topologiaan [26]. RLOC-osoite koostuu 64-bittisestä IPv6-etuliitteestä (Mesh-Local Prefix), jonka Thread-verkon kaikki laitteet yhteisesti jakavat sekä 64-bittisestä IID-tunnisteesta, joka sisältää 16-bittisen RLOC16-osoitteen ja on aina muotoa `0000:00ff:fe00:RLOC16` [26]. Jokaisella Thread-laitteella on yksilöllinen RLOC16-lyhytosoite, joka koostuu reitittimen tunnuksesta (Router ID) sekä päätelaitteen tunnuksesta (Child ID) [105, s. 10]. Kuvassa 4.8 on esitetty esimerkki päätelaitteen RLOC16-osoitteen muodostumisesta ja sen sisältymisestä varsinaiseen RLOC-osoitteeseen.



Kuva 4.8: Päätelaitteen RLOC-osoitteen muodostuminen, muokattu [26].

Reitittimien RLOC16-osoitteen määrityksestä vastaa Thread-verkon pääreititin ja päätelaitteiden RLOC16-osoitteesta parent-reititin, jotka seuraavat myös osoitteiden päällekkäisyyksiä verkossa [46, s. 9]. Reitittimillä RLOC16-osoitteen Child ID:n muodostavat 9:n bittiä ovat aina nollia, jonka perusteella osoitteen tiedetään kuuluvan reitittimelle [105, s. 10]. Päätelaitteen RLOC16 muodostuu parent-reitittimen Router ID:stä sekä päätelaitteelle määritellystä Child ID:stä, joka on aina muuta kuin nollia. Kun Thread-verkon reitittimillä on tiedossaan kaikkien muiden reitittimien RLOC16-osoitteet, ne osaavat reitittää paketteja Router ID -bittien perusteella kohdelaitteelle. Jos kyseessä on päätelaitteelle osoitettu paketti, välittää parent-reititin paketin päätelaitteelle ylläpitämänsä child-taulun mukaisesti. RLOC16-osoitteesta on nähtävissä, että Thread-verkossa voisi olla 63 reititintä ($2^6 - 1$), mutta reitittimien määrä on spesifikaatiossa rajattu 32:een, jonka myötä reititystaulu saadaan pidettyä yksinkertaisena. Vastaavasti Child ID:n 9 bitin myötä yhdellä reitittimellä voi teoriassa olla 511 ($2^9 - 1$) päätelaitetta. Kuvassa 4.9 on vielä esimerkki osoitteiden määräytymisestä Thread-verkossa.



Kuva 4.9: Thread-laitteiden RLOC-osoitteistus, muokattu [26].

Koska Thread-verkko on itsekorjautuva, voi päätelaite muodostaa yhteyden uuteen reitittimeen, jos sen oma parent-reititin ei ole enää saavutettavissa [105, s. 6]. Reitittimen vaihtumisen seurauksena päätelaitteen RLOC16 muuttuisi vastaamaan

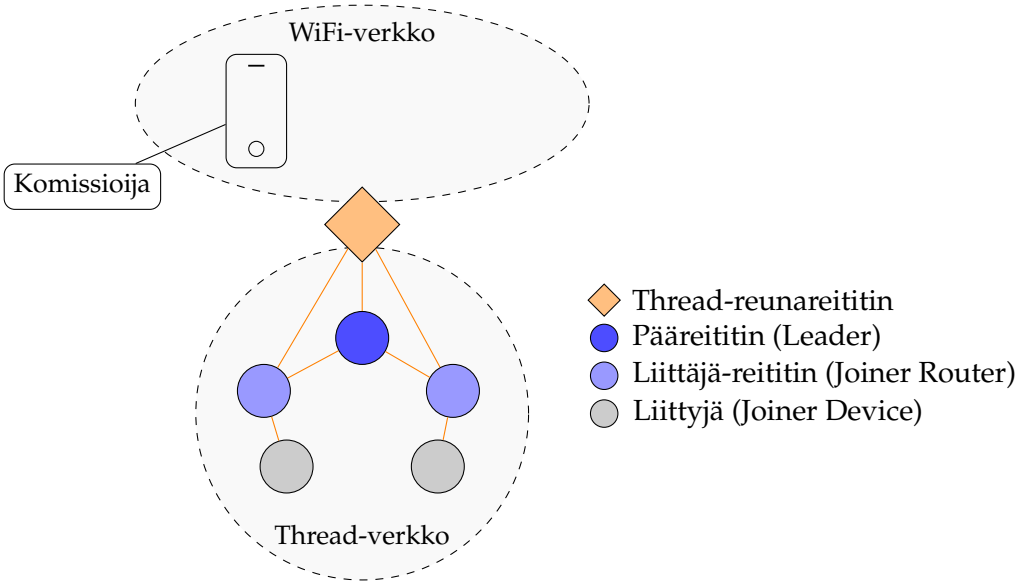
uuden parent-reitittimen osoitetta. Sovelluksissa ei näin ollen hyödynnetä RLOC-osoitteita, vaan laitteilla on vielä erillinen muuttumaton ML-EID-osoite. ML-EID-osoitteella on sama IPv6-etuliite kuin RLOC-osoitteella, mutta loput 64 bittiä muodostetaan satunnaisesti laitekomissioinnin jälkeen [26]. Reitityksessä verkkokerroksen tehtävänä on kohdistaa laitteen ML-EID-osoite sitä vastaavaan RLOC16-osoitteeseen. Tiettyjen Thread-verkon sisäisten konfiguraatioiden suorittamiseksi (kts. luku 4.3.1) osa Thread-laitteista vastaanottaa myös ryhmälähetysviestejä, johon on käytettävissä taulukon 4.1 mukaiset osoitteet. Lisäksi Thread-verkossa on käytettävissä spesifikaatiossa määriteltyjä anycast-osoitteita (Anycast Locator, ALOC) silloin, kun kohdelaitteen RLOC-osoite ei ole saavutettavissa tai se halutaan selvittää [26]. Esimerkiksi Thread-verkon pääreitittimelle on määritelty oma anycast-osoite, josta laite on aina saavutettavissa [105, s. 9].

Taulukko 4.1: Threadin ryhmälähetysosoitteet [26].

IPv6-osoite	**Ulottuvuus**	**Vastaanottajat**
ff02::1	Link-Local	FTD:t ja MED:t
ff02::2	Link-Local	FTD:t
ff03::1	Mesh-Local	FTD:t ja MED:t
ff03::2	Mesh-Local	FTD:t

4.3.3 Thread-laitteen komissiointi

Luvussa 4.3.1 sivuttiin lyhyesti vaiheita, jotka laitteen on käytävä läpi liittyessään Thread-verkkoon. Merkittävä osa Threadin tietoturvasta nojaa laitekomissiointiin ja sen myötä uudelle laitteelle luovutettavaan verkkoavaimeen, josta johdetaan erillinen salausavain linkkikerrokselle sekä MLE-viestien salaukseen [53]. Laitekomissiointi jaetaan kahteen menetelmään: sisäiseen ja ulkoiseen [102]. Sisäinen komissiointi tapahtuu Thread-verkon sisäisten laitteiden kesken, jolloin komissioiva laite on jo liittyneenä Thread-verkkoon. Ulkoisessa komissioinnissa komissioijana toimii esimerkiksi WiFi-verkkoon liittynyt älypuhelin, joka kommunikoi Thread-verkkoon reunareitittimen kautta. Kotiautomaatiossa tyypillinen komissiointimenetelmä perustuu juuri ulkoiseen komissiointiin. Esimerkiksi Matter hyödyntää tätä menetelmää, jonka myötä Thread-laitteiden liittämisestä pyritään tekemään käyttäjälle mahdollisimman yksinkertaista [101, s. 10]. Ulkoisessa komissioinnissa osallisena olevat laitteet on esitetty kuvassa 4.10.



Kuva 4.10: Thread-laitteen ulkoinen komissiointi, muokattu [102, s. 6].

Thread hyödyntää laitteiden komissioinnissa CoAP-protokollaan perustuvaa MeshCoP (Mesh Commissioning Protocol) -protokollaa sekä komissioinnin aikana laitteiden välille muodostettavia suojattuja DTLS (Datagram Transport Layer Security) -istuntoja [110, s. 165]. Lisäksi komissioinnin yhteydessä käytettävien avainten vaihtaminen suojataan J-PAKE (Password-Authenticated Key Exchange with Juggling) -menetelmällä [46, s. 11]. Varsinainen laitekomissiointi koostuu kahdesta vaiheesta: komissioijaehdokkaan hyväksymisestä sekä uuden laitteen komissioinnista Thread-verkkoon. Seuraavassa tarkastellaan hyvin yleisellä tasolla ainoastaan kuvan 4.10 mukaista ulkoista komissiointiprosessia.

Ennen kuin mikään laite voi toimia komissioijana, täytyy Thread-verkon pääreitittimen hyväksyä ulkoinen komissioijaehdokas verkon ainoaksi komissioijaksi [53, s. 4116]. Komissioijaehdokkaan, kuten älypuhelimen, on ensin muodostettava suojattu DTLS-istunto Thread-reunareitittimen kanssa käyttämällä yhteistä komissiointitunnusta (Commissioning Credential) [102, s. 2, 10]. Komissiointitunnuksena toimii komissioijaehdokkaan 6 – 255 tavuinen selkokielinen tunnus, josta johdetaan DTLS-istunnossa käytettävä PSKc (Pre-Shared Key for Commissioner) -avain [46, s. 10]. Kun reunareititin on todentanut komissioijaehdokkaan, se välittää pyynnön edelleen Thread-verkon pääreitittimelle, joka joko hyväksyy tai hylkää pyynnön [102, s. 14]. Jos pääreititin hyväksyy komissioijaehdokkaan verkon komissioijaksi, se välittää Thread-verkon reitittimille tiedon reunareitittimestä, jonka kautta kyseinen komissioija on saavutettavissa.

Kun uusi laite (Joiner Device, JD) haluaa liittyä Thread-verkkoon, on sen ensin muodostettava yhteys reitittimeen (Joiner Router, JR), jonka kautta laite on saanut alustavasti tietoonsa kyseisen Thread-verkon tunnistetiedot. JD avaa tämän jälkeen DTLS-kättelyn JR:n kanssa, joka välittää DTLS-kättelyn tiedot reunareitittimelle ja se edelleen komissioijalle [102, s.18]. Thread-verkon topologiasta ja reitittimien määrästä riippuen, JR-reitittimenä voi toimia myös pääreititin tai reunareititin, joka voi olla myös Thread-verkon pääreititin. Oleellista on kuitenkin se, että kommunikaatio kulkee aina reunareitittimen kautta ja komissioija ei toimita JD:lle verkkoavainta tai muita tunnistetietoja, vaan siitä vastaa aina JR. Suojatun DTLS-istunnon muodostamiseksi komissioija todentaa JD:n sen PSKd (Pre-Shared Key for Device) -avaimesta, jonka komissioija saa tietoonsa esimerkiksi laitteen QR-koodin skannaamisen myötä [53, s. 4118]. DTLS-istunnon muodostamisen jälkeen JD:n ja komissioijan välille muodostetaan yhteinen KEK (Key Encryption Key) -avain, jolla suojataan varsinaisen verkkoavaimen (Master Key, MK) toimitus JD:lle. KEK-avain on kertakäyttöinen ja komissiointikohtainen, jolla pyritään parantamaan komissioinnin tietoturvaa [53, s. 4118]. Komissioijan tehtävänä on toimittaa KEK-avain JR:lle, jonka jälkeen JR voi luovuttaa KEK-avaimella salatun MK-avaimen JD:lle [46, s. 11]. Kun JD on onnistuneesti komissioitu Thread-verkkoon, laite tallentaa JR-reitittimeltä saamansa MK-avaimen sekä verkon tunnistetiedot pysyväismuistiin, jotta laite voi liittyä samaan Thread-verkkoon ilman uudelleenkomissiointia [46, s. 10; 105, s. 17].

4.4 Thread osana Matteria

Yksi Threadin eroavaisuuksista esimerkiksi Zigbee- ja Z-Wave-teknologioihin on sovelluskerroksen jättäminen vapaaksi, mikä yhdessä Threadin IPv6-pohjaisuuden kanssa mahdollistaa erilaisten koti- ja rakennusautomaation sovellusratkaisujen käyttämisen Threadin päällä [98, s. 11-12]. Vaikka Thread soveltuukin ominaisuuksiensa ja teknologioidensa puolesta niin älykodin, rakennusautomaation kuin älykaupungin IoT-sovelluksiin [47, 74], on sen hyödyntäminen ollut vähäistä. Osin tähän on ollut syynä Threadin IP-pohjaisuus sekä rajoittuminen verkkokerroksen protokollaksi, jonka myötä Threadin edellyttämän reunareitittimen toiminnallisuuden toteuttaminen kaupallisiin laitteisiin on ollut hidasta [34, s. 2]. Toisaalta Zigbeen ja Z-Waven kaltaisten teknologioiden suosio etenkin kotiautomaatiossa ei ole antanut valmistajille syytä uuden teknologian hyödyntämiselle.

Kotiautomaatiossa Thread on alkanut yleistymään laajemmin vasta viime vuosina Matter-kommunikointiprotokollan julkaisun jälkeen. Matterin näkökulmasta Thread on ideaali ratkaisu vähävirtaisten mesh-laiteverkkojen teknologiaksi sen IEEE 802.15.4 -pohjaisuuden, natiivin IPv6-tuen sekä vapaan sovelluskerroksen myötä [96]. Matterista onkin tullut Threadin kannalta merkityksellisin sovelluskerroksen ratkaisu kotiautomaatiossa, sillä markkinoille tuotavat Thread-laitteet ovat Thread-sertifioinnin ohella myös Matter-sertifioituja, jonka myötä Thread-laitteiden komissiointiprosessi ja kommunikointimalli perustuvat Matteriin.

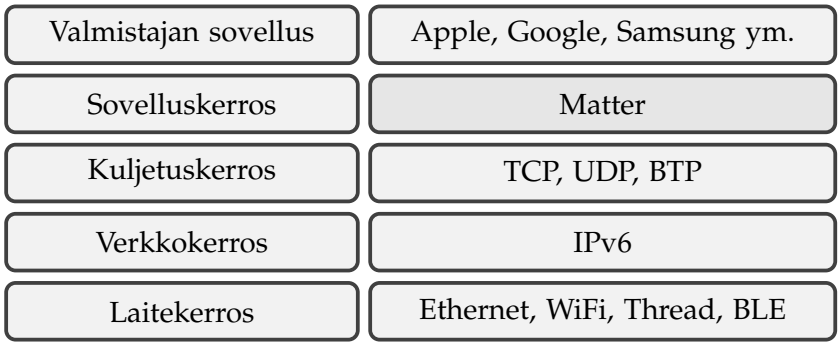
4.4.1 Matter yleisesti

Matterin on tarkoitus vihdoin ratkaista älykodin laitteiden ja sovellusten väliset yhteensopivuusongelmat tuomalla älykotimarkkinoille standardisoitu ratkaisu, jonka myötä älykodin laitteiden hankinta ja käyttöönotto selkeytyy sekä kehitystyö yhtenäistyy [17]. Matter on avoimeen lähdekoodiin [16] perustuva sovelluskerroksen kommunikointiprotokolla, jota hyödyntämällä älykodin laitteet pystyvät kommunikoimaan keskenään saman ekosysteemin sisällä riippumatta siitä, minkä valmistajan sovelluksista tai laitteista ratkaisu koostuu. Matterin kehitystyö alkoi vuonna 2019 nimellä CHIP (Connected Home over IP) ja sen kehittämisestä vastaa teknologiayritysten yhteenliittymä, jossa ovat mukana muun muassa Amazon, Apple, Google, Samsung sekä CSA [17]. Yhteenliittymän toimintaa ohjaa CSA, joka vastaa myös markkinoille tuotavien Matter-laitteiden sertifioinnista. Sertifioinnin läpäisseet laitteet merkitään Matter-logolla, jonka on tarkoitus olla lupaus siitä, että laitteet ovat yhteensopivia keskenään ja liitettävissä mihin tahansa Matteria tukevaan älykotisalustaan [17, 49].

Matterin myötä kuluttajat eivät ole enää sidottuja valmistajakohtaisiin laitteisiin tai ekosysteemeihin, vaan voivat jatkossa valita vapaammin älykotiin hankittavat laitteet sekä käytettävät sovellukset [112]. Matterin on osaltaan myös tarkoitus yksinkertaistaa, yhtenäistää sekä nopeuttaa älykodin IoT-laitteiden kehitystyötä ja markkinoille tuloa. Matterin myötä laitevalmistajien ei enää tarvitse toteuttaa samasta laitteesta usean eri tiedonsiirtoteknologian versioita, vaan valmistajat voivat jatkossa hyödyntää Matterin teknologioita toteuttaakseen Matter-sertifioidun laitteen, joiden tiedetään keskustelevan ongelmitta keskenään ja ovat liitettävissä minkä tahansa valmistajan Matteria tukevaan IoT-alustaan [80, 112].

4.4.2 Matterin kerrosmalli

Matter on täysin IPv6-pohjainen ja se rakentuu tunnettujen verkkoteknologioiden, kuten WiFin, Ethernetin sekä Bluetooth LE:n päälle ja hyödyntää vähävirtaisten mesh-verkkojen teknologiana Threadia [14]. Kuvassa 4.11 on esitetty Matterin kerrosmalli ja sen tukemat teknologiat. Matter tukee kuljetuskerroksella TCP-, UDP- sekä BTP (Bluetooth Transport Protocol) -protokollia [21, s. 79-80] ja verkkokerroksella on tuettuna ainoastaan IPv6-protokolla. Tiedonsiirtoteknologioista WiFi ja Ethernet toimivat Matterissa älykodin paikallisverkkona ja mahdollistavat enemmän kaistaa vaativien ratkaisujen hyödyntämisen osana Matteria sekä tarvittaessa laiteyhteydet internetiin. Mesh-laiteverkkojen teknologiana Matter tukee suoraan ainoastaan Threadia. BLE-teknologiaa Matterissa hyödynnetään Matter-laitteiden komissiointiprosessin aikana, mutta muilta osin BLE ei ole käytössä Matterissa.



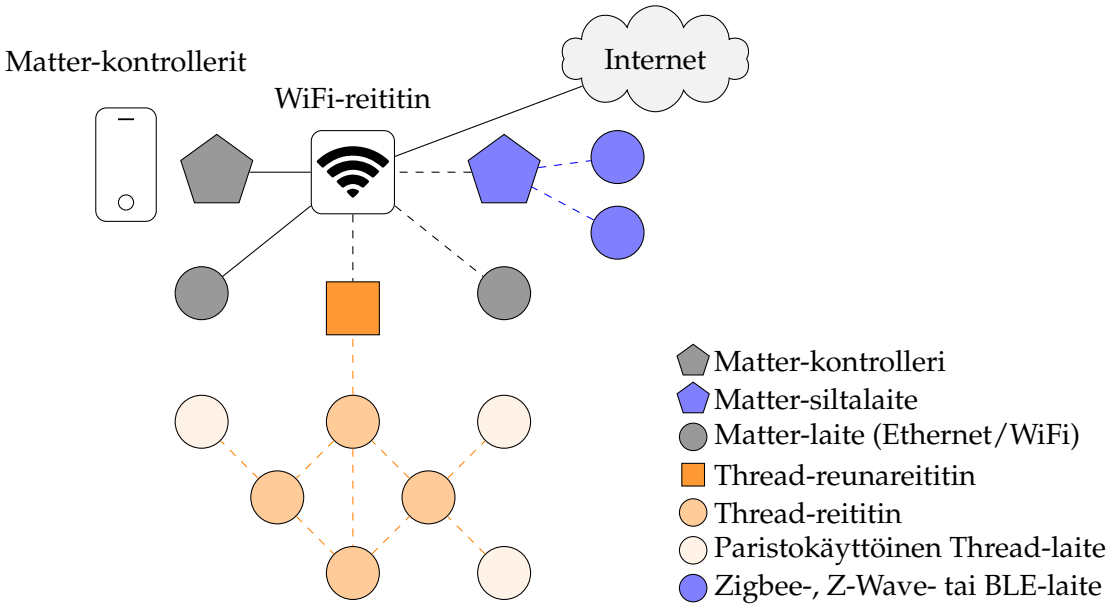
Kuva 4.11: Matterin kerrosmalli, muokattu [3, s. 3].

4.4.3 Matterin arkkitehtuuri

Matter on suunniteltu paikalliseksi verkkoratkaisuksi, jossa eri verkkoteknologioihin perustuvat IPv6-pohjaiset laitteet voivat keskustella keskenään ilman edellytystä internet-yhteydelle tai pilvipalveluille [21, s. 47-48]. Matterin paikallisen verkkoratkaisun myötä laitekomennot välittyvät lähes viiveettä ja älykotiverkon toimintavarmuus sekä yksityisyys paranevat. Internet-yhteyttä edellytetään kuitenkin esimerkiksi etähallintaan, ohjelmistopäivityksiin sekä Matter-laitteiden aitouden tarkistamiseen komissiointiprosessin yhteydessä CSA:n DCL (Distributed Compliance Ledger) -palvelusta [15; 80, s. 3]. Lisäksi edistyneempien laiteasetusten määritys saattaa edellyttää valmistajien omien sovellusratkaisujen käyttämistä [49, s. 61].

Matter yhdistää WiFi-, Ethernet- tai Thread-verkkoon liittyneet Matteria tukevat laitteet Matter-kokoelmaksi. Matter-kokoelmasta käytetään nimitystä *Matter Fabric* ja kokoelmia voi olla käytössä useita samanaikaisesti [21, s. 817]. Matter-kokoelmien rinnakkainen toiminta on mahdollista Matterin Multi-Admin-ominaisuuden myötä, mikä antaa käyttäjille mahdollisuuden liittää samat Matter-laitteet useaan älykotialustaan riippumatta sovellusten tai laitteiden valmistajasta [12]. Matter-laite liitetään Matter-kokoelmaan erillisellä komissiointiprosessilla, jossa hyödynnetään mobiililaitetta sekä laitteen QR-koodia tai laitetunnistetta. Komissiointi tapahtuu BLE- tai WiFi-yhteydellä riippuen Matter-laitteen tukemasta tiedonsiirtoteknologiasta [21, s. 238]. Matter-kokoelmaan voi liittää ainoastaan sertifioituja laitteita, millä pyritään parantamaan älykotiverkon tietoturvaa sekä yksityisyyttä [15].

Kuvassa 4.12 on esitetty eri verkkoteknologioista koostuva Matter-arkkitehtuuri. Matter edellyttää aina paikallista Matter-kontrolleria, jonka kautta Matter-laitteet liitetään IoT-alustaan ja joka vastaa automaatioista, laitteiden välisistä yhteyksistä sekä etähallinnasta [3, s. 8; 85]. Jos Matter-kokoelmaan liitetään Thread-laitteita, on kotiverkossa oltava myös vähintään yksi Thread-reunareitin. Thread-reunareititin voi olla integroituna Matter-kontrolleriin, jonka myötä kotiautomaatioratkaisun laitekantaa voidaan vähentää. Matter ei ole millään tapaa muita tiedonsiirtoteknologioita poissulkeva ratkaisu, vaan esimerkiksi Zigbee-, Z-Wave- tai BLE-laitteita voi liittää osaksi Matter-kokoelmaa Matteria tukevan siltalaitteen kautta [21, s. 475].



Kuva 4.12: Matterin arkkitehtuuri, muokattu [101, s. 7].

5 Kotiautomaatioratkaisun toteutus

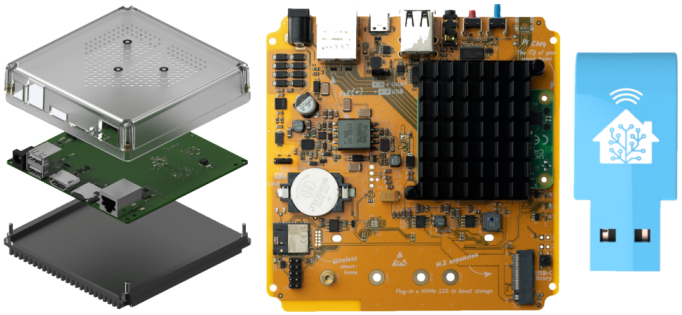
Threadin teoreettinen tarkastelu vastaa osaltaan tutkielman tutkimusongelmaan eli mitä Threadilla pyritään ratkaisemaan kotiautomaatiossa. Jotta Threadin käytännönläheinen tarkastelu olisi tutkielman kontekstiin soveltuvaa, oli tarpeen toteuttaa pienimuotoinen Threadia hyödyntävä kotiautomaatioratkaisu. Lisäksi empiirisen osuuden tavoitteena oli tutkia luvussa 2.2 tarkasteltuihin haasteisiin peilaten, kuinka usean tiedonsiirtoteknologian hyödyntäminen olisi keskitetyssä kotiautomaatioratkaisussa mahdollista. Ratkaisu toteutettiin kohteeseen, jossa ei aiemmin hyödynnetty kotiautomaatiota, joten perusajatuksena oli kustannustehokas alustaratkaisu, joka kuitenkin antaisi hyvät lähtökohdat kotiautomaation laajentamiselle tulevaisuudessa. Näiden vaatimusten pohjalta IoT-alustaksi valikoitui avoimen lähdekoodin Home Assistant [37], jonka todettiin tukevan useaa tiedonsiirtoteknologia sekä kattavasti eri valmistajien IoT-laitteita ja ennen kaikkea vähintään kehitysasteella Matteria sekä Threadia [39, 41].

5.1 Home Assistant

Home Assistant on erityisesti kotiautomaatioon suunnattu avoimen lähdekoodin IoT-alusta. Täysin paikallisena ja pilvipalveluista riippumattomana alustaratkaisuna Home Assistant mahdollistaa kodin laitteiden keskitetyn hallinnan ja automatisoinnin sekä datan yksityisyyden säilyttämisen [37]. Home Assistant tukee useiden valmistajien eri tiedonsiirtoteknologioihin perustuvien IoT-laitteiden sekä palveluiden liittämistä tuhansien sovellusintegraatioiden kautta [36]. Home Assistantin toiminnallisuutta on lisäksi mahdollista laajentaa erilaisilla lisäosilla, kuten tietokannan hallintaan liittyvillä työkaluilla. Home Assistant on suunniteltu käytettäväksi selainkäyttöliittymällä, mutta käytettävissä on myös mobiililaitteille suunnattu Companion-sovellus [35]. Companion-sovellus mahdollistaa Home Assistantiin lisättyjen laitteiden helpon ohjaamisen sekä sovellusilmoitusten vastaanottamisen Home Assistantista mobiililaitteeseen. Home Assistant on etähallittavissa internetin yli ottamalla käyttöön maksullisen Home Assistant Cloud -pilvipalvelun tai toteuttamalla etäyhteyden kotiverkkoon omilla menetelmillä.

Home Assistantin päänäkymänä toimii kojelauta, jonka kautta kodin laitteiden ohjaaminen ja niiden keräämän informaation esittäminen on mahdollista. Kojelaudat ovat hyvin vapaasti käyttäjän muokattavissa erilaisilla korteilla, jotka on jaettu kategorioihin, kuten laitteiden ohjaamiseen tai sensorien keräämän datan visualisointiin. Esimerkki Home Assistantin kojelaudasta on esitetty myöhemmin luvussa 5.2.3. Home Assistantissa laitteilla sekä palveluilla olevia toimintoja tai attribuutteja kutsutaan entiteetiksi. Entiteetit voivat olla esimerkiksi laitteen ohjauskomentoja tai sen tarjoamia attribuutteja, kuten sensoriarvoja, joita voidaan esittää kojelaudassa sekä hyödyntää automaatioissa. Home Assistantin automaatiot perustuvat entiteetin tilan seuraamiseen ja sen perusteella suoritettaviin toimintoihin, kuten toisen entiteetin tilan muuttamiseen. Entiteettien tilojen seuraamisesta vastaa tilakone, joka välittää tiedon tilan muutoksesta tapahtumakäsittelijälle toiminnon suorittamiseksi [79, s. 167337]. Automaatioita voi luoda joko graafisen käyttöliittymän avulla tai suoraan YAML (YAML Ain't Markup Language) -merkintäkielellä ja automaatiot mahdollistavat monimutkaistenkin ehtorakenteiden luomisen sekä useiden laitteiden ohjaamisen.

HAOS-käyttöjärjestelmä (Home Assistant Operating System) on otettavissa käyttöön useilla eri alustaratkaisuilla, kuten esimerkiksi Raspberry Pi -alustalla [36]. Home Assistant markkinoi myös omia Green- sekä Yellow-laitealustoja [37], jotka on esitetty kuvassa 5.1. Alustojen on tarkoitus mahdollistaa Home Assistantin käytön aloittaminen mahdollisimman helposti, sillä ne ovat plug-and-play-laitteita ja sisältävät HAOS-käyttöjärjestelmän valmiiksi asennettuna. Alustat on suunniteltu liitettäväksi kotiverkkoon Ethernet-kaapelilla eivätkä ne sisällä tukea esimerkiksi WiFi-yhteydelle. Edullisempi Green-alusta ei sisällä tukea millekään langattomalle teknologialle ja tuki niille on toteutettava sopivaa USB-donglea hyödyntämällä. Tarkoitukseen soveltuu esimerkiksi Home Assistantin markkinoima SkyConnect (kts. kuva 5.1), joka on Silicon Labsin EFR32MG21-piiriin perustuva pienikokoinen Zigbeetä sekä Threadia tukeva USB-dongle [37]. Yellow-alusta rakentuu Raspberry Pi CM4 -moduulin ympärille ja alusta tukee Silicon Labs MGM210P -moduulin myötä Zigbeetä sekä Threadia. Yellow-alustaa on saatavilla joko valmiiksi koottuna tai itse koottavina versioina, jotka mahdollistavat monipuolisemmilla ominaisuuksilla varustetun Raspberry Pi CM4 -moduulin käyttämisen ja vaihtoehtoisesti alustan virransyötön PoE (Power-over-Ethernet) -tekniikalla.



Kuva 5.1: Home Assistant Green, Yellow sekä SkyConnect [37].

5.2 Toteutus

Tässä luvussa esitellään toteutetun ratkaisun rakennetta, käytettyjä laitteita sekä vaatimuksia Threadin hyödyntämiselle Home Assistantissa. Tarkoituksena ei ole käydä yksityiskohtaisesti läpi Home Assistantin käyttöönottoa tai sovellettuja asetuksia, vaan kertoa pääpiirteittäin hyödynnetyt menetelmät sekä vaatimukset toimivan ratkaisun toteuttamiseksi. Toteutetun ratkaisun osalta on huomioitava, että tässä luvussa esitettävät menetelmät ja vaatimukset pätevät tutkielman aikaan saatavilla olleeseen tietoon sekä työkaluihin.

Vaikka dokumentaation perusteella Home Assistantin todettiin tukevan Threadia, on sen hyödyntäminen edelleen tutkielman aikaan varhaisessa kehitysvaiheessa. Näin ollen Threadin toimivuudesta ei ollut varmuutta ja ratkaisuun hankittiin vain yksittäisiä Matter- ja Thread-sertifioituja laitteita. Jos laitteiden liittäminen Home Assistanttiin onnistuisi, voidaan myös muiden sertifioitujen laitteiden olettaa liittyvän ongelmitta. Ratkaisuun hankittiin lisäksi muutama Zigbee-laite, jotta usean tiedonsiirtoteknologian yhtäaikainen toiminta olisi todennettavissa. Alustaratkaisun kustannustehokkuutta tavoiteltaessa ajatuksena oli pitää usean tiedonsiirtoteknologian hyödyntämiseen vaadittava laitteisto mahdollisimman vähäisenä. Home Assistantin Thread-dokumentaation [41] mukaan Home Assistant voi toimia ainoana Thread-reunareitittimenä hyödyntämällä OTBR (OpenThread Border Router) -integraatiota sekä IEEE 802.15.4 -radion sisältävää laitetta. Tämän myötä ratkaisussa olisi mahdollista välttää kolmansien osapuolien Matter-kontrollerien sekä Thread-reunareitittimien hankinta ja siten alentaa merkittävästi alustaratkaisun kustannuksia. On kuitenkin huomattava, että käytettäessä Home Assistantia ainoana Thread-reunareitittimenä, on Thread-laitteiden komissiointi mahdollista ainoastaan Android-pohjaisella älypuhelimella [41].

5.2.1 Ratkaisun arkkitehtuuri ja käytetty laitteisto

Taulukossa 5.1 on lueteltuna kaikki ratkaisussa hyödynnetyt laitteet ja ratkaisun arkkitehtuuri on esitetty kuvassa 5.2. Home Assistantin HAOS-käyttöjärjestelmä asennettiin aiemmin hankitulle Raspberry Pi 3B+ -alustalle, joten Zigbeetä ja Threadia varten oli hankittava IEEE 802.15.4 -radion sisältävä USB-dongle. Alun perin ajatuksena oli hankkia Home Assistantin SkyConnect, sillä sen sisältämä Silicon Labsin EFR32MG21-piiri mahdollistaa Zigbeetä ja Threadia yhtäaikaisesti tukevan laiteohjelmiston hyödyntämisen donglessa. SkyConnectin saatavuudessa oli kuitenkin haasteita ja lopulta dongleksi valikoitui Sonoff Zigbee 3.0 USB Dongle Plus-E, joka rakentuu niin ikään EFR32MG21-piiristä. Sonoffin donglessa oli saapuessaan laiteohjelmistona suhteellisen vanha Zigbee-koordinaattori ja koska tavoitteena oli hyödyntää Zigbeetä sekä Threadia, dongleen vaihdettiin Silicon Labsin MultiPAN RCP (Radio-Co-Processor) -ohjelmisto. Donglen uudelleenohjelmoinnissa hyödynnettiin kehittäjän toteuttamaa web-pohjaista työkalua [1], jolla laiteohjelmistoksi on vaihdettavissa seuraavat vaihtoehdot:

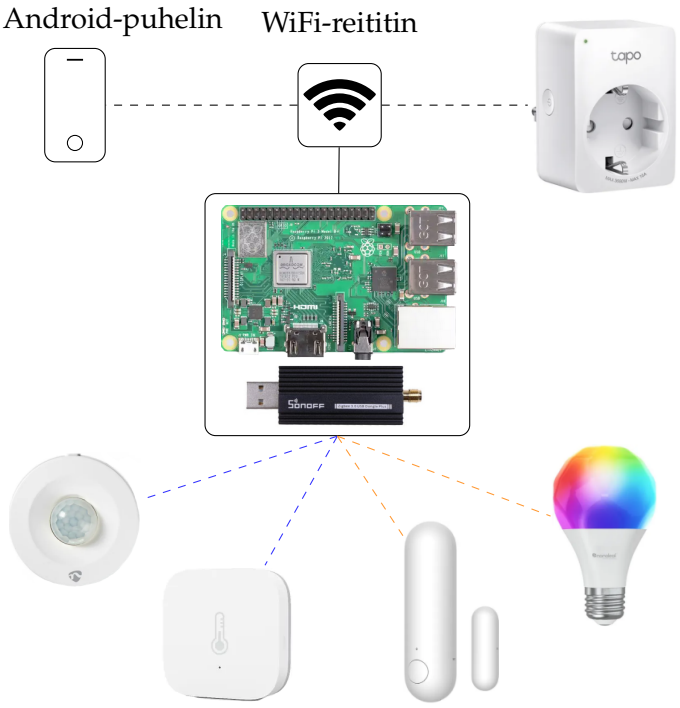
- EZSP (EmberZNet Serial Protocol) Zigbee
- MultiPAN RCP Zigbee + Thread
- OpenThread RCP.

Silicon Labs MultiPAN RCP -laiteohjelmisto on moniprotokolla-ohjelmisto, jolla EFR32MG21-piirin sisältävä laite tukee yhtäaikaisesti Zigbeetä sekä Threadia. Moniprotokolla-ohjelmiston hyödyntäminen EFR32MG21-piiriin perustuvalla laitteella edellyttää Home Assistantin tapauksessa sitä, että Zigbee- ja Thread-verkko määritellään käyttämään samaa IEEE 802.15.4 -kanavaa [38]. Verkkojen kommunikaatio erotetaan toisistaan suodattamalla vastaanotetut paketit verkon PAN ID:n perusteella ja välittämällä paketit sen mukaan oikealle verkkokerrokselle [87, s. 5]. Haasteena moniprotokolla-ohjelmistoa käytettäessä on se, että Zigbeen ja Threadin toimiessa samalla kanavalla, laitteet jakavat myös käytettävissä olevan kaistan keskenään. Yhteisen kaistan jakaminen voikin rajoittaa Zigbee- sekä Thread-laitteiden määrää kotiautomaatiossa, sillä laitemäärän kasvaessa myös tiedonsiirto-ongelmat todennäköisimmin lisääntyvät.

[1] https://github.com/darkxst/silabs-firmware-builder

Taulukko 5.1: Ratkaisussa hyödynnetyt laitteet.

Raspberry Pi 3 Model B+ [71]
Sonoff Zigbee 3.0 USB Dongle Plus-E [43]
Nedis SmartLife ZBSM10WT, liiketunnistin (Zigbee) [62]
Aqara Temperature and Humidity Sensor T1, olosuhdesensori (Zigbee) [55]
Aqara Door and Window Sensor P2, ovi-/ikkunasensori (Thread+Matter) [54]
Nanoleaf Essentials Matter Smart Bulb, älylamppu (Thread+Matter) [61]
TP-Link Tapo P110M, älypistorasia (WiFi+Matter) [109]



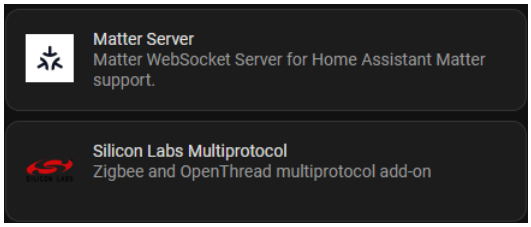
Kuva 5.2: Ratkaisun arkkitehtuuri, laitekuvat [43, 54, 55, 61, 62, 71, 109].

5.2.2 Laitteiden käyttöönoton vaatimukset

Zigbeen ja Threadin hyödyntäminen edellyttää tiettyjen lisäosien sekä integraatioiden asentamista ja määrittämistä Home Assistantiin. Kaupalliset kotiautomaatioon suunnatut Thread-laitteet ovat useimmiten myös Matter-sertifioituja ja näin ollen Thread-laitteiden komissiointi sekä laitteiden välinen kommunikaatio noudattaa Matter-spesifikaatiota. Kuten tutkielman teoriaosuudessa Matterin osalta todettiin,

täytyy kotiverkossa olla Matter-kontrollerina toimiva laite, joka yhdistää laitteet samaan Matter-kokoelmaan. Thread vastaavasti edellyttää aina Thread-reunareitintä, joka toimii rajapintana Thread-verkon sekä WiFi-/Ethernet-verkon välillä. Home Assistant tarjoaa kattavan dokumentaation mitä Matterin ja Threadin hyödyntäminen edellyttää Home Assistantissa [39, 41], joten tässä luvussa vaatimuksia käsitellään ainoastaan yleisellä tasolla.

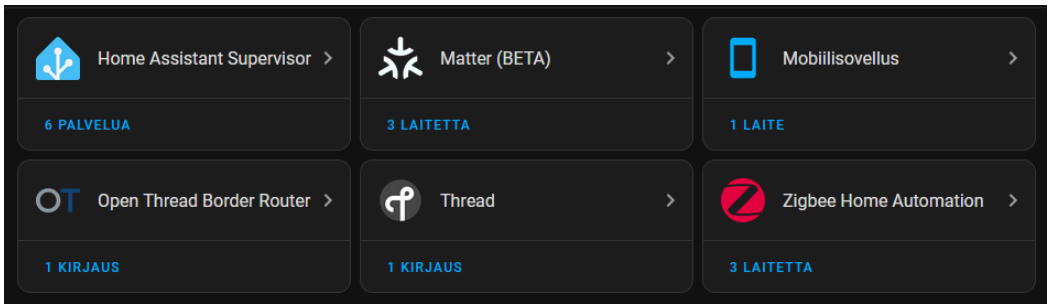
Tutkielman toteutuksessa tarvittavat Home Assistantin lisäosat on esitetty kuvassa 5.3. Jotta moniprotokolla-ohjelmistoa käyttävä IEEE 802.15.4 -dongle on hyödynnettävissä, täytyy Home Assistantiin asentaa Silicon Labs Multiprotocol -lisäosa. Lisäosa toimii sovellusrajapintana donglen sekä Home Assistantin integraatioiden välillä ja siihen sisältyy Zigbee-protokollapino sekä OTBR (OpenThread Border Router) -toteutus, jota hyödyntämällä Home Assistant toimii Thread-reunareititttimenä. Matter Server -lisäosa toteuttaa Matter-kontrollerin toiminnallisuuden Home Assistantiin. Lisäosa mahdollistaa Matter-kokoelman luomisen sekä laitteiden lisäämisen kokoelmaan ja vastaa Matter-laitteiden välisestä kommunikaatiosta. Edellä mainittujen lisäosien myötä Home Assistant saadaan toimimaan sekä Matter-kontrollerina että Thread-reunareititttimenä.



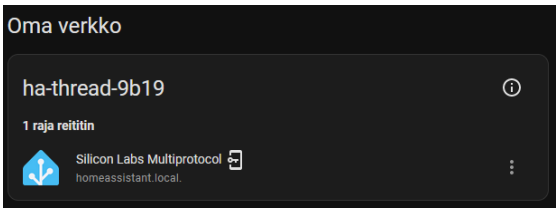
Kuva 5.3: Ratkaisun edellyttämät lisäosat.

Ratkaisun kannalta välttämättömät Home Assistantin integraatiot on esitetty kuvassa 5.4. Home Assistant Supervisor on HAOS-käyttöjärjestelmän sisäinen integraatio, joka vastaa HAOS:n ylläpidosta ja päivityksestä. Matter (BETA) -integraatio kommunikoi Matter Server -lisäosan kanssa ja mahdollistaa Matter-kokoelmaan liitettyjen laitteiden ohjaamisen Home Assistantista. Mobiilisovellus-integraatio asentuu automaattisesti, kun Companion-sovelluksella kirjaudutaan Home Assistantiin. Integraatio mahdollistaa ilmoitusten vastaanottamisen Home Assistantista puhelimeen sekä puhelimen sensorien hyödyntämisen entiteetteinä Home Assistantissa. Open Thread Border Router -integraatio mahdollistaa Home Assistantin määritellä ja kontrolloida OTBR-reunareitintä sen tarjoaman REST API:n (REpresentational State Transfer Application Programming Interface) kautta [41]. Thread-integraatiolla

vastaavasti hallitaan käytettävissä olevia Thread-verkkoja ja niiden tunnistietoja sekä määritellään Home Assistantin ensisijaisesti käyttämä Thread-verkko, josta esimerkki kuvassa 5.5. Zigbee Home Automation (ZHA) -integraatiota käytetään luomaan Zigbee-verkko. Zigbee-verkon luominen on ZHA:lla hyvin johdonmukaista, mutta jos Home Assistantissa hyödynnetään Silicon Labsin Multiprotocol -lisäosaa, täytyy ZHA:n laitepoluksi määrittää donglen sijaan lisäosan tarjoama polku (esim. `socket://core-silabs-multiprotocol:9999`). Home Assistantin muodostaman Zigbee- sekä Thread-verkon osalta on lisäksi varmistettava, että verkoille on määritelty sama IEEE 802.15.4 -kanava, jotta kommunikaatio kummankin verkon laitteisiin voi toimia. WiFi-verkkojen mahdollisesti aiheuttamien häiriöiden minimoimiseksi Zigbee- ja Thread-verkkojen muodostuksessa kanavaksi valitaan oletuksena joko 15, 20 tai 25, jotka eivät mene päällekkäin yleisimmin käytettävien WiFi-kanavien 1, 6 sekä 11 kanssa [38].

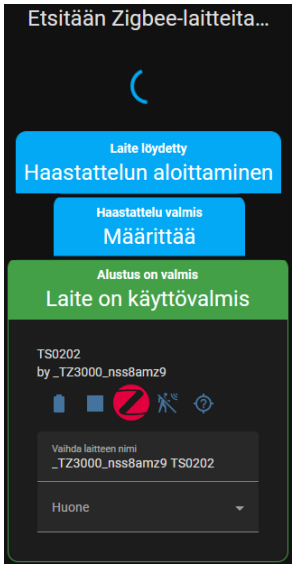


Kuva 5.4: Ratkaisun edellyttämät integraatiot.



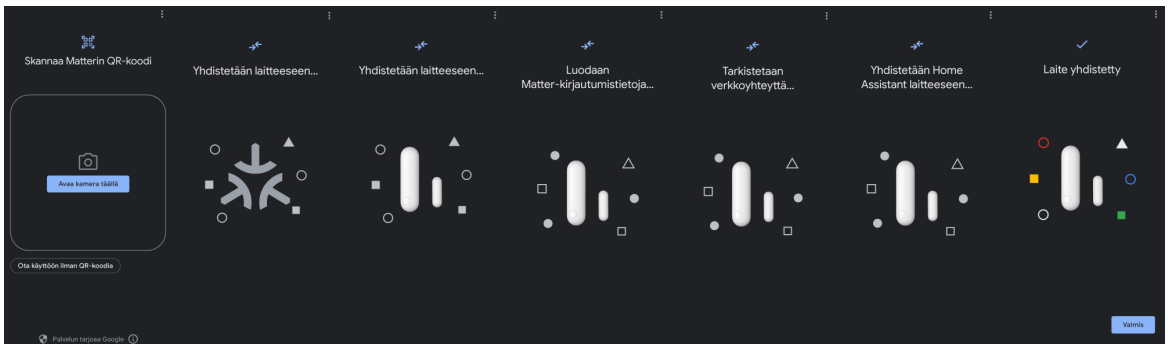
Kuva 5.5: Home Assistantin Thread-verkko.

Kun moniprotokollan, Matterin sekä Threadin edellyttämät lisäosat ja integraatiot on asennettu ja määritelty, voi laitteita liittää Home Assistantiin. Zigbee-laitteen liittäminen on käyttäjälle hyvin suoraviivainen prosessi. Käyttäjän tarvitsee ainoastaan laittaa Zigbee-laitteiden haku päälle ja painaa Zigbee-laitteessa olevaa nappia, jonka jälkeen laite liittyy ZHA:n muodostamaan Zigbee-verkkoon. Kuvassa 5.6 on esitetty typistetysti kolme vaihetta, jotka suoritetaan Zigbee-laitetta liitettäessä.



Kuva 5.6: Zigbee-liiketunnistimen liittäminen Home Assistantiin.

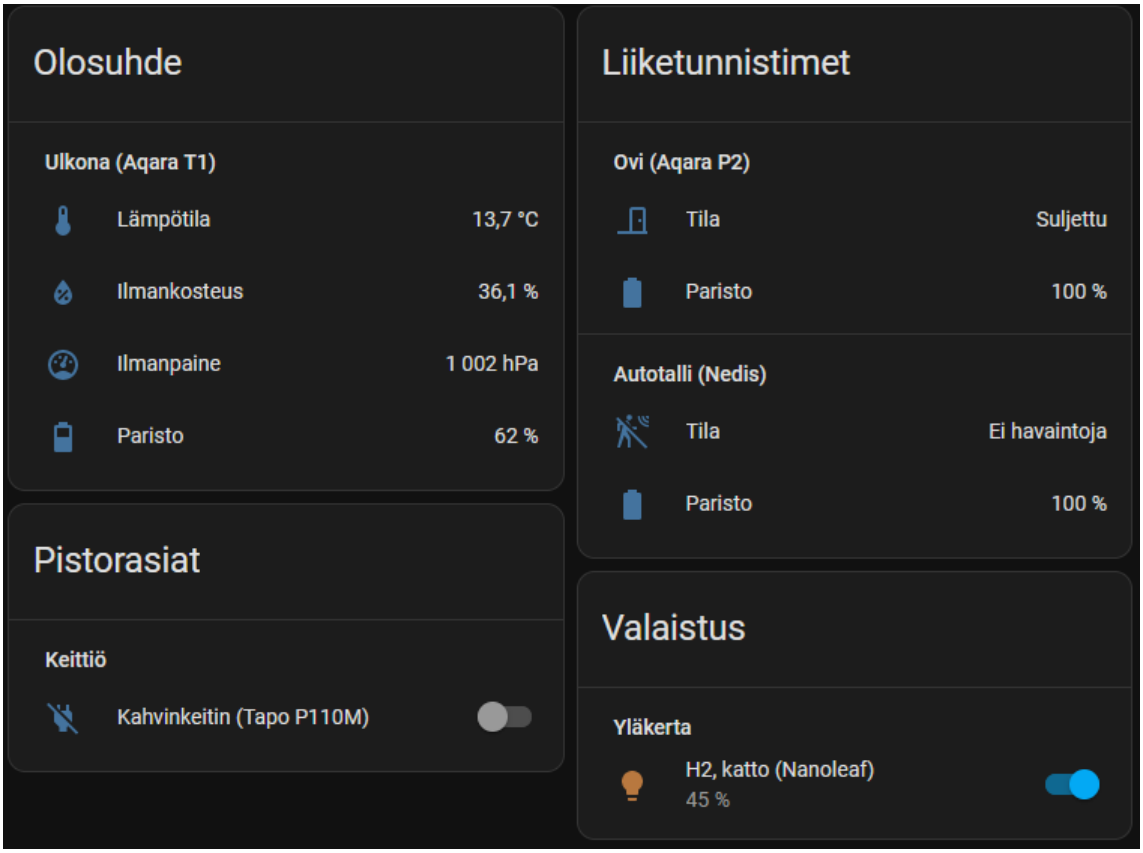
Matter-laitteet lisätään Home Assistantiin Companion-mobiilisovelluksella, sillä laitekomissioinnissa hyödynnetään BLE-yhteyttä. WiFi-pohjaisten Matter-laitteiden komissioinnin voi aloittaa heti Companion-sovelluksen käyttöönottamisen jälkeen. Ennen Thread-laitteiden komissiointia on kuitenkin välttämätöntä synkronoida Home Assistantin muodostaman Thread-verkon tunnistetiedot puhelimeen. Synkronointi tehdään Companion-sovelluksella ja menetelmä on tutkielman aikaan tuettuna ainoastaan Android-laitteilla [41]. Matter-laite lisätään Home Assistantiin aloittamalla Companion-sovelluksessa uuden Matter-laitteen lisääminen ja QR-koodin skannaamisen tai laitetunnisteen syöttämisen jälkeen käyttäjältä ei edellytetä toimenpiteitä, vaan komissiointiprosessi etenee automaattisesti. Kuvassa 5.7 on esitettynä kaikki vaiheet komissioitaessa uutta Thread-laitetta Matter-kokoelmaan.



Kuva 5.7: Thread-laitteen lisääminen Home Assistantiin.

5.2.3 Ratkaisun toiminnallisuus ja automaatiot

Kuvassa 5.8 on esitetty toteutetun ratkaisun kojelauta. Home Assistantiin on mahdollista luoda useita kojelautoja ja niihin on valittavissa hyvinkin informatiivisia näkymiä sekä erilaisia nappeja ja säätimiä laitteiden ohjaamiseen. Mobiilikäyttöä ajatellen kojelauta pyrittiin pitämään mahdollisimman selkeänä ja jäsenneltynä siten, että siihen on helppo lisätä uusia laitteita. Thread-pohjaista älylamppua ohjataan kojelaudasta yksinkertaisella liukusäätimellä, jonka yhteydessä voidaan esittää myös muutamia hyödyllisiä tietoja lampun tilasta. Lampun muut asetukset, kuten värin, värilämpötilan sekä kirkkauden säädöt ovat käytettävissä lampun ”takaa” avautuvasta näkymästä, mutta esimerkiksi vaihtuvat valoefektit eivät ole käytettävissä. WiFi-pohjaisella Matter-älypistorasialla ei ole käytettävissä muita entiteettejä, kuin päälle-/poiskytkentä. Esimerkiksi energiankulutustiedot eivät ole käytettävissä. Zigbee- ja Thread-sensorit tarjoavat laitetyypeille ominaiset attribuutit esitettäväksi kojelaudassa.



Kuva 5.8: Home Assistantin kojelauta.

Ohessa muutama yksinkertainen automaatio havainnollistamaan, kuinka Home Assistantilla voi luoda automaatioita. Home Assistant mahdollistaa todella monipuolisten automaatioiden toteuttamisen, mutta tutkielman kannalta oleellista oli testata teknologioiden toimivuutta, mikä on saavutettavissa yksinkertaisillakin automaatioilla. Ensimmäinen automaatio 5.1 laittaa kahvinkeittimen päälle asetettuna kellonaikana ja lähettää puhelimeen ilmoituksen, johon sisältyy myös ulkolämpötila. Toinen automaatio 5.2 seuraa ulkolämpötilan muuttumista ja jos terassin ovi on auki ulkolämpötilan ollessa alle 15 astetta, automaatio lähettää ilmoituksen puhelimeen. Kolmannessa automaatiossa 5.3 Zigbee-liiketunnistimella ohjataan Thread-älylamppua ovikellon tavoin. Automaatio vaihtaa lampun värin punaiseksi kahden sekunnin ajaksi, kun liiketunnistin havaitsee liikettä viiden sekunnin ajan ja lamppu on päällä. Ennen lampun värin vaihtamista lampun nykyinen tila otetaan talteen, joka palautetaan värin vaihtamisen jälkeen.

Listaus 5.1: Automaatio kahvinkeittimelle.

```
alias: kahvinkeitin
description: Kahvinkeitin päälle ja puhelimeen ilmoitus, jossa mukana ulkolämpötila.
trigger:
  - platform: time
    at: "07:00:00"
action:
  - service: switch.turn_on
    target:
      entity_id: switch.smart_wi_fi_plug
    data: {}
  - service: notify.mobile_app_sm_s21fe
    metadata: {}
    data:
      message: >-
        Kahvinkeitin laitettu päälle. Ulkolämpötila:
        {{states('sensor.lumi_lumi_sensor_ht_agl02_lampotila')}}°C
mode: single
```

Listaus 5.2: Automaatio oven tarkistukselle.

```
alias: terassin_ovi
description: Ilmoitus, jos ulkolämpötila < 15 ja ovi on auki.
trigger:
  - platform: state
    entity_id: sensor.lumi_lumi_sensor_ht_agl02_lampotila
condition:
  - condition: and
    conditions:
      - condition: numeric_state
```

```
      entity_id: sensor.lumi_lumi_sensor_ht_agl02_lampotila
      below: 15
    - condition: state
      entity_id: binary_sensor.aqara_door_and_window_sensor_p2
      state: "on"
action:
  - service: notify.mobile_app_sm_s21fe
    data:
     message: >-
      Terassin ovi auki. Lämpötila:
      {{states('sensor.lumi_lumi_sensor_ht_agl02_lampotila')}}°C
mode: single
```

Listaus 5.3: Automaatio liiketunnistimelle.

```
alias: liiketunnistin_lamppu
description: Lampun värin vaihtaminen 2s ajaksi, jos lamppu päällä ja liiketunnistin
    havaitsee liikettä 5s.
trigger:
  - platform: state
    entity_id: binary_sensor.tz3000_nss8amz9_ts0202_liike
    from: "off"
    to: "on"
    for:
     seconds: 5
condition:
  - condition: and
    conditions:
     - condition: state
       entity_id: light.essentials_a19_a60
       state: "on"
action:
  - service: scene.create
    data:
     scene_id: before
     snapshot_entities:
       - light.essentials_a19_a60
  - service: light.turn_on
    target:
     entity_id: light.essentials_a19_a60
    data:
     brightness_pct: 40
     rgb_color:
       - 255
       - 0
       - 0
  - delay:
     seconds: 2
  - service: scene.turn_on
    target:
     entity_id: scene.before
    data: {}
mode: single
```

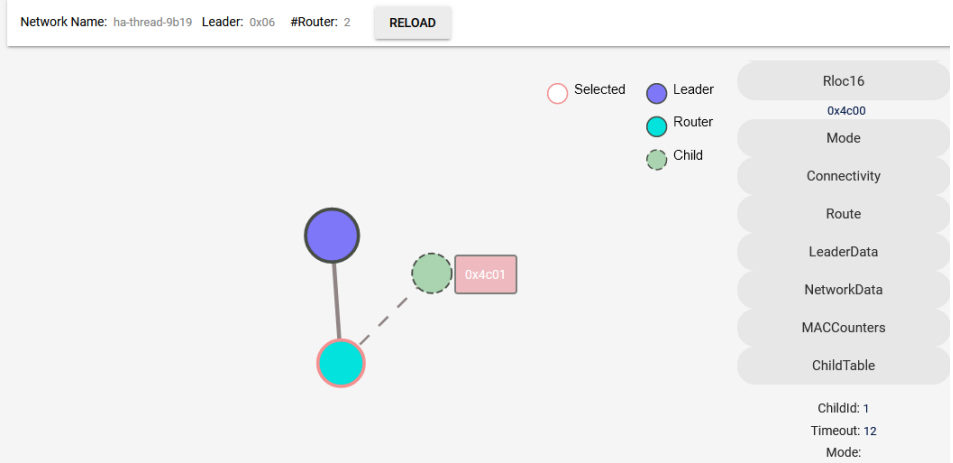
5.3 Johtopäätökset toteutuksesta

Toteutettu ratkaisu on hyvin suppea esimerkki Home Assistantin tarjoamista mahdollisuuksista kotiautomaatioon. Käyttäjän näkökulmasta Home Assistant vaatii perehtymistä käyttöönottoon liittyvissä toimissa sekä laitteiden edellyttämien integraatioiden ja lisäosien määrittämisessä. Merkittävä osa ajasta kuluukin erinäisiin alkuvalmisteluihin, mutta kattavan dokumentaation myötä Home Assistantin käytössä pääsee kuitenkin nopeasti alkuun. Ennen Home Assistantin käyttöönottamista kannattaa kuitenkin pohtia, mitä tiedonsiirtoteknologioita ratkaisussa aikoo käyttää ja minkälaisella laitteistolla teknologiat ovat hyödynnettävissä. Kuten tutkielman toteutuskin osoittaa, Home Assistant tarjoaa lopulta kattavan tuen usealle eri tiedonsiirtoteknologialle sekä laitevalmistajalle ja mahdollistaa laitteiden keskitetyn hallinnan yhdellä IoT-alustalla. Lisäksi yksinkertaisten, mutta kuitenkin hyödyllisten automaatioiden toteuttaminen onnistuu suhteellisen vähäisellä vaivalla. Kaiken kaikkiaan Home Assistantia voi pitää erittäin varteenotettava vaihtoehtona ensimmäiseksi ja ainoaksi kotiautomaatioalustaksi, kunhan on valmis käyttämään jonkin verran aikaa tarvittaviin alkuvalmisteluihin.

Tutkielmassa toteutettua alustaratkaisua voi pitää hyvinkin kustannustehokkaana. Jos HAOS-käyttöjärjestelmä on asennettavissa olemassa olevalle laitealustalle, edellyttää ratkaisu ainoastaan soveltuvan IEEE 802.15.4 -donglen hankkimista, joiden hinnat liikkuvat yleisesti muutamissa kymmenissä euroissa. Muiden laitteiden osalta tilanne on jokseenkin kaksijakoinen. Zigbee-laitteita on ollut IoT-markkinoilla jo vuosien ajan ja laitteet ovat tänä päivänä hyvinkin edullisia. Thread on vastaavasti kotiautomaatiossa suhteellisen uusi teknologia ja laitteiden tuominen markkinoille edellyttää aina liittymistä Thread Groupin jäseneksi sekä pakollista laitesertifiointia, mikä näkyy kuluttajalle laitteiden korkeahkona hintana. Kotimaisilla markkinoilla Thread-laite voi olla jopa kolme kertaa kalliimpi, kuin vastaava Zigbee-laite. Ratkaisun kokonaiskustannustehokkuuden arvioiminen on suhteellista, sillä jokaisella on todennäköisesti yksiköllinen näkemys kotiautomaatioon käytettävästä rahamäärästä suhteessa laitteiden tarjoamaan hyödyllisyyteen.

Home Assistantin Silicon Labs Multiprotocol -lisäosan OTBR-toteutus mahdollistaa web-käyttöliittymän hyödyntämisen Thread-verkon analysointiin. Käyttöliittymän kautta voi tarkastella muun muassa verkon topologiaa, IPv6-osoitteita sekä verkkoon liittyneiden laitteiden reititystietoja. Kuvassa 5.9 on esitetty tutkielman ratkaisun Thread-laitteiden muodostama topologia. Vähäisestä laitemäärästä johtuen mesh-verkon muodostumisesta ei voi puhua, mutta topologiasta voi todeta,

että Thread-laitteet toimivat verkossa laitetyyppien mukaisesti. Thread-verkon on muodostanut OTBR-reunareititin, joten se toimii myös verkon pääreitittimenä. Kahden muun laitteen osalta on nähtävissä, että Thread-lampun RLOC16-osoite on `0x4c00` ja laite toimii parent-reitittimenä paristokäyttöiselle Thread-ovisensorille, jonka RLOC16-osoite `0x4c01` on muodostunut parent-reitittimen Router ID:stä sekä laitteen omasta Child ID:stä (1). Jos lamppu ei ollut enää ovisensorin saavutettavissa, ovisensorin parent-reitittimeksi vaihtui reunareititin. Vastaavasti lampun palatessa takaisin verkkoon, lamppu liittyi ensin reunareitittimen child-laitteeksi ja sen tila vaihtui hetkeä myöhemmin varsinaiseksi reitittimeksi. Thread-verkon itsemuodostuvuus ja itsekorjautuvuus oli näin ollen todennettavissa laitteiden osalta.



Kuva 5.9: Thread-verkon topologia.

Luvussa 5.2.2 käytiin lyhyesti läpi Zigbee- ja Matter-laitteiden liittämisprosesseja, jotka ovat helppoudeltaan käyttäjälle hyvin samankaltaista. Zigbee-laitteet ovat kuitenkin huomattavasti nopeammin hyödynnettävissä Home Assistantissa, sillä Threadin käyttöönottaminen vaatii valituista laiteratkaisuista riippuen enemmän alkuvalmisteluja. Jos ajatuksena on hyödyntää WiFi-pohjaisia Matter-laitteita, näiden lisääminen Home Assistantiin vaatii ainoastaan Matterin edellyttämän lisäosan ja integraation asentamisen sekä Companion-sovelluksen käyttöönottamisen. Laitteiden lisääminen Home Assistantiin oli hyvin yksinkertaista, sillä käyttäjän ei tarvitse antaa verkkojen tunnistetietoja tai salasanoja, vaan niin Zigbee-laitteiden kuin Matter-laitteidenkin liittämisprosessit etenivät automaattisesti. Matter-laitteen komissiointi osoittautui kuitenkin erityisen hitaaksi prosessiksi, jossa aikaa kului useita minuutteja. Komissiointi päättyi myös useimmiten virheeseen, mutta laite saattoi

kuitenkin olla liittyneenä Home Assistantiin. Verrokkina Zigbee-laitteet liittyivät Home Assistantiin kymmenissä sekunneissa ja aina onnistuneesti.

Tutkielmassa toteutettu ratkaisu osoitti, että Zigbeen ja Threadin hyödyntäminen on mahdollista myös yhdellä moniprotokolla-laiteohjelmistoa tukevalla IEEE 802.15.4 -donglella. Moniprotokolla-ratkaisua ei kuitenkaan lähtökohtaisesti suositella käytettäväksi, sillä ratkaisu on kokeellinen [40]. Suosituksena on käyttää joko kaupallista Thread-reunareititintä tai erillistä donglea Zigbeelle ja Threadille. Tutkielman ratkaisussa ei havaittu tiedonsiirto-ongelmia Zigbee- tai Thread-laitteiden osalta, mutta on huomioitava, että ratkaisun laitemäärä on vähäinen ja mahdolliset ongelmat voivat ilmetä laitemäärän kasvaessa. Ratkaisun toimivuutta sekä laitteiden ja Home Assistantin välistä kommunikaatiota testattiin esimerkiksi luvussa 5.2.3 esitetyillä yksinkertaisilla automaatioilla. Toimintojen viiveettömyyttä arvioitiin ainoastaan silmämääräisesti. Ohjattaessa WiFi-pistorasiaa tai Thread-lamppua Home Assistantin kojelaudasta, toiminta oli käyttäjän näkökulmasta viiveetöntä. Zigbee- tai Thread-laitteen kommunikaatioviiveessä Home Assistantiin ei ollut havaittavissa eroa. Kokonaisuutena ratkaisun toiminta oli automaatioitakin hyödyntämällä niin viiveetöntä, ettei toimintaan kiinnittänyt mitään huomiota.

Matter on vielä uusi teknologia kotiautomaatiossa, mikä ilmeni Matter-laitteiden osin puutteellisina ominaisuuksina. Esimerkiksi Thread-älylamppu ei tue valoefektejä tai vaihtuvia tiloja, eivätkä WiFi-älypistorasian energiankulutustiedot olleet hyödynnettävissä. Puutteet johtuvat laitteissa käytetystä Matter-versiosta, joka ei tue kyseisten laiteominaisuuksien välittämistä Matterin yli. Juuri tutkielman aikaan julkaistun Matter 1.3 -spesifikaation myötä, Matter tukee jatkossa myös edellä mainittuja ominaisuuksia [13], mutta riippuu täysin laitevalmistajista, milloin uuden spesifikaation määritykset ovat tuettuina. Home Assistantin tapaisessa alustaratkaisussa Matterin hyödyllisyys ei tullut esille parhaalla mahdollisella tavalla, sillä Home Assistant on lähtökohtaisesti paikallinen ratkaisu ja se tukee kattavasti useiden eri valmistajien laitteita lisäosien sekä integraatioiden kautta. Joka tapauksessa Matterin valmistajariippumattomuus sekä laitteiden helppo liittäminen toteutuivat myös tutkielman ratkaisussa. Lisäksi Threadin ja Matterin eduksi on sanottava se, että laitteiden saavutettavuus oli Zigbee-laitteita parempi. Matter-laitteet olivat aina käytettävissä HAOS:n käynnistymisen jälkeen ja jos laitteiden virta katkaistiin ja palautettiin myöhemmin, laitteet olivat muutamissa sekunneissa jälleen aktiivisena Home Assistantissa. Zigbee-laitteet sen sijaan toimivat vaihtelevasti ja entiteettien päivittymisessä oli havaittavissa ongelmia etenkin HAOS:n käynnistymisen jälkeen.

6 Yhteenveto

Tutkielmassa luotiin yleiskatsaus kotiautomaatioon tarkastelemalla sen tarjoamia mahdollisuuksia, tyypillisimpiä sovelluskohteita sekä sen hyödyntämiseen liittyviä merkittävimpiä haasteita. Lisäksi tutkielmassa perehdyttiin muutamaan suosituimpaan kotiautomaation langattomaan tiedonsiirtoteknologiaan. Kotiautomaatio mahdollistaa helpolla tavalla kodin resurssien, asumisolosuhteiden sekä ympäristön monitoroinnin reaaliaikaisesti, jonka pohjalta kodin laitteita voidaan ohjata joko automatisoidusti asetettujen sääntöjen perusteella tai manuaalisesti etähallinnan kautta. Kotiautomaatiolla tavoitellaan niin taloudellisia kuin ajallisiakin säästöjä sekä pyritään lisäämään asumismukavuutta, parantamaan kodin energiatehokkuutta ja siten asumisen ekologisuutta sekä luomaan kodista turvallisempi asumisympäristö. Kotiautomaation merkittävimpinä haasteina nähdään kaupallisten ratkaisujen korkeat hankinta- ja ylläpitokustannukset, ratkaisujen käyttöönoton hankaluudet, riittävän yksityisyyden saavuttaminen sekä yhteensopivuusongelmat eri valmistajien laitteiden ja sovellusratkaisujen välillä. Kotiautomaatioratkaisuissa hyödynnetään tyypillisimmin lyhyen kantaman langattomia teknologioita, kuten WiFiä, BLE:tä, Zigbeetä tai Z-Wavea. WiFi on tänä päivänä yleinen kodin langaton verkkoteknologia ja BLE laajasti tuettuna erilaisissa käyttäjälaitteissa, jonka myötä teknologioihin pohjautuvien IoT-laitteiden käyttöönotto on nopeaa ja helppoa ilman erillislaitteiden hankintaa. Zigbee ja Z-Wave sen sijaan edellyttävät aina teknologiaa tukevan hub-laitteen hyödyntämistä, mutta vastapainoksi ne tarjoavat alhaisen virrankulutuksen mesh-pohjaisina teknologioina luotettavaa tiedonsiirtoa ja tukevat suurta määrää paristokäyttöisiä laitteita yhdessä verkossa soveltuen siten erinomaisesti kotiautomaation sensoriverkkoratkaisuihin.

Tutkielman teoriaosuuden pääpaino oli Thread-verkkoprotokollan käsittelyssä, mikä osaltaan vastaa tutkielmalle asetettuun tutkimusongelmaan, mikä Thread on ja mitä se tuo kotiautomaatioon yhteensopivuuden näkökulmasta. Thread on suunniteltu tarjoamaan laitevalmistajille standardisoitu IPv6-pohjainen verkkoprotokolla, jota hyödyntämällä IoT-laitteiden kehitystyö helpottuu sekä yhtenäistyy eri valmistajien välillä. Mesh-pohjaisena ja laitetasolla IEEE 802.15.4 -standardin päälle rakentuvana teknologiana Thread jakaa paljon samankaltaisuutta Zigbeen ja osin

myös Z-Waven kanssa. Threadin merkittävimmät erot ja samalla sen edut ovat natiivi tuki IPv6-protokollalle sekä sovelluskerroksen jättäminen vapaaksi. Zigbeen ja Z-Waven tavoin myös Thread edellyttää IoT-reunalaitteen käyttöä, jonka kautta Thread-verkot liittyvät kodin WiFi- tai Ethernet-verkkoon. Zigbeen ja Z-Waven edellyttämät hub-laitteet ovat useimmiten ainakin jollain tasolla valmistajakohtaisia, jotta ne tukevat valmistajan sovellusratkaisua. Thread-reunareitittimet ovat sen sijaan täysin valmistajariippumattomia ja reunareitittimen toiminnallisuus on toteutettavissa hyvin erityyppisiin kodin älylaitteisiin, sillä IPv6-pohjaisuuden myötä reunareitittimissä ei edellytetä protokollamuunnoksia tai paketin salauksen purkamista. Lisäksi Thread tukee useiden reunareitittimien käyttöä, mikä lisää kotiautomaation toimintavarmuutta, kun Thread-verkkoon ei muodostu yhtä vikaherkkää pistettä. Kotiautomaatiossa Threadin sovellusratkaisuna toimii Matter. Siinä missä Thread tarjoaa laitevalmistajille yhtenäisen tavan toteuttaa alhaisen virrankulutuksen IoT-laitteita sekä Thread-reunareitittimiä, on Matterin tarkoitus standardisoida eri valmistajien laitteiden ja sovellusratkaisujen kommunikointi sovelluskerroksella. Matter on kotiautomaatiossa vielä suhteellisen uusi teknologia, mutta yhdessä Threadin kanssa sen odotetaan olevan seuraava edistysaskel kohti yhtenäistä ja valmistajariippumatonta kotiautomaatiota.

Tutkielman empiirisessä osuudessa toteutetun kotiautomaatioratkaisun avulla pyrittiin löytämään vastauksia tutkielman toiseen tutkimusongelmaan eli miten kotiautomaatioon yleisimmin liittyvät haasteet olisivat vältettävissä. Toteutettu ratkaisu osoitti, että kotiautomaation käytön aloittaminen on mahdollista suhteellisen vähäisellä vaivalla ja alhaisilla kustannuksilla, kun ratkaisu toteutetaan Home Assistantin kaltaisen avoimen lähdekoodin IoT-alustan päälle. Vaikka Home Assistantin käyttöönotto, sopivan oheislaitteiston selvittäminen ja automaatioiden toteutus vaativatkin syvällisempää perehtymistä, tarjoaa Home Assistant lopulta valmiin kotiautomaatioalustan, joka tukee useaa tiedonsiirtoteknologiaa sekä eri valmistajien IoT-laitteita ja mahdollistaa paikallisena ratkaisuna kotiautomaation täydellisen kontrollin säilyttämisen sekä yksityisyyden saavuttamisen. Tutkielman toteutuksessa Home Assistantin todettiin tukevan Matteria sekä Threadia, mikä osaltaan helpottaa ratkaisun laajentamista tulevaisuudessa, kun IoT-laitteita voi valita vapaammin eri valmistajilta. Tutkielman ratkaisussa Home Assistantissa hyödynnettiin moniprotokolla-toteutusta ja sen myötä samaa IEEE 802.15.4 -radiota Zigbeelle sekä Threadille. Moniprotokolla-toteutuksen todettiin toimivan ongelmitta eikä esimerkiksi Zigbee- tai Thread-laitteiden kommunikaatioviiveessä havaittu käytettä-

vyyden kannalta eroja. Threadin osalta ainoana rajoittavana tekijänä voidaan pitää laitteiden korkeaa hintaa, jonka myötä Zigbee säilyttänee edelleen suosionsa kotiautomaatiossa. Matter ja Thread kuitenkin tarjoavat sertifioitujen laitteiden myötä täydellistä yhteensopivuutta valmistajasta riippumatta, mikä on merkittävä etu myös avoimen lähdekoodin IoT-alustan päälle rakentuvassa kotiautomaatiossa. Tutkielmassa myös havaittiin, että Matter-laitteiden saavutettavuus Home Assistantissa oli merkittävästi parempi kuin Zigbee-laitteiden, mikä parantaa yleisesti kotiautomaation käytettävyyttä.

Tutkielmassa esitetty kotiautomaatioratkaisu on vain yksi esimerkki, kuinka kotiautomaatioon yleisimmin liittyvät haasteet olisivat vältettävissä. Toteutetun ratkaisun IoT-alusta sekä käytetyt laitteet ja teknologiat valittiin tarkoin, jotta muun muassa Thread olisi hyödynnettävissä ratkaisussa. Lisäksi alhaisesta laitemäärästä johtuen Zigbeen ja Threadin osalta hyödynnetyn moniprotokolla-ratkaisun toimivuus oli todennettavissa luotettavasti ainoastaan tutkielman laajuudessa. Tutkielmaan liittyy näin ollen selkeitä rajoituksia ja esitettyihin tuloksiin on suhtauduttava tietyllä kriittisyydellä eikä niiden pohjalta voi tehdä liian yleistäviä johtopäätöksiä.

Jatkotutkimusaiheista päällimmäiseksi nousee Matter, jota sivuttiin tutkielmassa vain yleisellä tasolla. Matter kuitenkin yleistyy kotiautomaatiossa nopeasti ja se on saanut taakseen merkittävimmät IoT-teknologiayritykset. Matterin teknologinen tarkastelu auttaisi paremmin ymmärtämään, mihin Matterin kommunikaatiomalli perustuu ja mitä Matter tuo kotiautomaatioon yhteensopivuuden sekä tietoturvan näkökulmasta. Matterin tutkiminen on mahdollista myös laitetasolla, sillä avoimen lähdekoodin teknologiana Matter [1] on vapaasti hyödynnettävissä ja kokeiltavissa useilla eri alustaratkaisuilla. Niin ikään avoimen lähdekoodin teknologiana OpenThread [2] mahdollistaisi Threadin laiteläheisen tutkimisen. Threadin tutkiminen vaihtoehtoisena teknologiana esimerkiksi sensoriverkkoratkaisuihin voisi tarjota uusia näkökulmia Threadin soveltuvuudesta ja hyödyistä myös kotiautomaation ulkopuolelle, kuten laajemman rakennusautomaation tai älykaupungin IoT-ratkaisuihin.

[1] https://github.com/project-chip/connectedhomeip

[2] https://github.com/openthread/openthread

Lähteet

[1] AKHMETZHANOV, B. K., GAZIZULY, O. A., NURLAN, Z., JA ZHAKIYEV, N. Integration of a Video Surveillance System Into a Smart Home Using the Home Assistant Platform. Julkaisusarjassa *2022 International Conference on Smart Information Systems and Technologies (SIST)* (2022), 1–5.

[2] BABUN, L., DENNEY, K., CELIK, Z. B., MCDANIEL, P., JA ULUAGAC, S. A survey on IoT platforms: Communication, security, and privacy perspectives. *Computer Networks 192* (03 2021), 108040/1–52.

[3] BELLI, D., BARSOCCHI, P., JA PALUMBO, F. Connectivity Standards Alliance Matter: State of the art and opportunities. *Internet of Things 25* (2024), 101005/1–27.

[4] BLUETOOTH SIG, INC. Bluetooth Mesh Networking. URL `https://www.bluetooth.com/wp-content/uploads/2019/03/Mesh-Technology-Overview.pdf`, viitattu 16.03.2024.

[5] BLUETOOTH SIG, INC. Learn About Bluetooth Topology Options. URL `https://www.bluetooth.com/learn-about-bluetooth/topology-options/`, viitattu 14.03.2024.

[6] BLUETOOTH SIG, INC. *Core Specification 5.0*, 2016. URL `https://www.bluetooth.com/specifications/specs/core-specification-5-0/`.

[7] BLUETOOTH SIG, INC. Understanding Reliability in Bluetooth Technology, 2020. URL `https://www.bluetooth.com/bluetooth-resources/understanding-reliability-in-bluetooth-technology/`, viitattu 15.03.2024.

[8] BLUETOOTH SIG, INC. The Bluetooth Low Energy Primer, 2022. URL `https://www.bluetooth.com/wp-content/uploads/2022/05/Bluetooth_LE_Primer_Paper.pdf`, viitattu 16.03.2024.

[9] BRUSH, A. B., LEE, B., MAHAJAN, R., AGARWAL, S., SAROIU, S., JA DIXON, C. Home automation in the wild: challenges and opportunities. Julkaisusarjassa *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), CHI '11, Association for Computing Machinery, 2115–2124.

[10] CHAUDHARY, S., YOUSUFF, S., MEGHANA, N., T S, A., JA GUDDETI, R. R. A Multi-Protocol Home Automation System Using Smart Gateway. *Wireless Personal Communications 116* (02 2021), 2367–2390.

[11] CHEN, D. A Survey of IEEE 802.11 Protocols: Comparison and Prospective. Julkaisusarjassa *5th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2017)* (01 2017), vol. 141 of *Advances in Engineering Research*, 569–578.

[12] CONNECTIVITY STANDARDS ALLIANCE. All about choice: Multi-Admin in Matter gives users more flexibility and unlocks smart home innovation, julkaistu 27.04.2022. URL https://csa-iot.org/newsroom/all-about-choice-multi-admin-in-matter-gives-users-more-flexibility-and-unlocks-smart-home-innovation/, viitattu 19.04.2024.

[13] CONNECTIVITY STANDARDS ALLIANCE. Matter 1.3 Specification announced, enabling energy reporting for Matter devices, and support for water and energy management, electric vehicle chargers, and new major appliances, julkaistu 08.05.2024. URL https://csa-iot.org/newsroom/matter-1-3-specification-released/, viitattu 11.05.2024.

[14] CONNECTIVITY STANDARDS ALLIANCE. Matter Arrives Bringing A More Interoperable, Simple And Secure Internet Of Things to Life, julkaistu 04.10.2022. URL https://csa-iot.org/newsroom/matter-arrives/, viitattu 19.04.2024.

[15] CONNECTIVITY STANDARDS ALLIANCE. Matter FAQ. URL https://csa-iot.org/all-solutions/matter/matter-faq/, viitattu 19.04.2024.

[16] CONNECTIVITY STANDARDS ALLIANCE. Matter Github. URL https://github.com/project-chip/connectedhomeip, viitattu 19.04.2024.

[17] CONNECTIVITY STANDARDS ALLIANCE. The Connectivity Standards Alliance Unveils Matter, Formerly Known as Project CHIP, julkaistu 11.05.2021. URL https://csa-iot.org/newsroom/chip-is-now-matter/, viitattu 19.04.2024.

[18] CONNECTIVITY STANDARDS ALLIANCE. Zigbee. URL https://csa-iot.org/all-solutions/zigbee/, viitattu 19.03.2024.

[19] CONNECTIVITY STANDARDS ALLIANCE. Zigbee PRO 2023 Improves Overall Security While Simplifying Experience, julkaistu 12.04.2023. URL https://csa-iot.org/newsroom/zigbee-pro-2023-improves-overall-security-while-simplifying-experience/, viitattu 23.03.2024.

[20] CONNECTIVITY STANDARDS ALLIANCE. *Zigbee Specification, R22 1.0*, 2017. URL https://csa-iot.org/wp-content/uploads/2022/01/docs-05-3474-22-0csg-zigbee-specification-1.pdf.

[21] CONNECTIVITY STANDARDS ALLIANCE. *Matter Specification Version 1.0*, 2022.

[22] DOMÍNGUEZ-BOLAÑO, T., CAMPOS, O., BARRAL, V., ESCUDERO, C. J., JA GARCÍA-NAYA, J. A. An overview of IoT architectures, technologies, and existing open-source projects. *Internet of Things 20* (2022), 100626/1–15.

[23] FROIZ-MÍGUEZ, I., FERNÁNDEZ-CARAMÉS, T., FRAGA-LAMAS, P., JA CASTEDO, L. Design, Implementation and Practical Evaluation of an IoT Home Automation System for Fog Computing Applications Based on MQTT and ZigBee-WiFi Sensor Nodes. *Sensors 18(8)* (08 2018), 1–42.

[24] FURSZYFER DEL RIO, D., SOVACOOL, B., JA GRIFFITHS, S. Culture, energy and climate sustainability, and smart home technologies: A mixed methods comparison of four countries. *Energy and Climate Change 2* (12 2021), 100035/1–19.

[25] GOOGLE. OpenThread. URL https://openthread.io/, viitattu 10.04.2024.

[26] GOOGLE. OpenThread - IPv6 Addressing, päivitetty 07.09.2023. URL https://openthread.io/guides/thread-primer/ipv6-addressing, viitattu 28.04.2024.

[27] GOOGLE. OpenThread - Network Discovery and Formation, päivitetty 07.09.2023. URL https://openthread.io/guides/thread-primer/network-discovery, viitattu 27.04.2024.

[28] GOOGLE. OpenThread - Node Roles and Types, päivitetty 07.09.2023. URL https://openthread.io/guides/thread-primer/node-roles-and-types, viitattu 25.04.2024.

[29] GOOGLE. OpenThread - Router Selection, päivitetty 07.09.2023. URL https://openthread.io/guides/thread-primer/router-selection, viitattu 27.04.2024.

[30] GØTHESEN, S., HADDARA, M., JA KUMAR, K. N. Empowering homes with intelligence: An investigation of smart home technology adoption and usage. *Internet of Things 24* (2023), 100944/1–21.

[31] HALBOUNI, A., ONG, L.-Y., JA CHEW, L. Wireless Security Protocols WPA3: A Systematic Literature Review. *IEEE Access 11* (2023), 112438–112450.

[32] HAMZAH, A. S., JA ABDUL-RAHAIM, L. A. Smart Home Automation System Using Cloud Computing Based Enhancement Security and Environment. Julkaisusarjassa *2022 2nd International Conference on Advances in Engineering Science and Technology (AEST)* (2022), 334–339.

[33] HARGREAVES, T., WILSON, C., JA HAUXWELL-BALDWIN, R. Learning to live in a smart home. *Building Research & Information 46* (02 2017), 1–13.

[34] HERRERA, T., JA NÚÑEZ, F. Design and Prototyping of a Thread Border Router Based on a Non Network-Co-Processor Architecture. *IEEE Access 8* (2020), 60613–60625.

[35] HOME ASSISTANT. Companion App. URL https://companion.home-assistant.io/, viitattu 07.05.2024.

[36] HOME ASSISTANT. Getting Started. URL https://www.home-assistant.io/getting-started/, viitattu 06.05.2024.

[37] HOME ASSISTANT. Home Assistant. URL https://www.home-assistant.io/, viitattu 06.05.2024.

[38] HOME ASSISTANT. Home Assistant SkyConnect, About multiprotocol issues. URL https://skyconnect.home-assistant.io/about-multiprotocol/, viitattu 09.05.2024.

[39] HOME ASSISTANT. Matter (BETA). URL https://www.home-assistant.io/integrations/matter/, viitattu 07.05.2024.

[40] HOME ASSISTANT. The State of Matter, julkaistu 25.01.2024. URL https://www.home-assistant.io/blog/2024/01/25/matter-livestream-blog/, viitattu 12.05.2024.

[41] HOME ASSISTANT. Thread. URL https://www.home-assistant.io/integrations/thread/, viitattu 07.05.2024.

[42] IEEE. The Evolution of Wi-Fi Technology and Standards, julkaistu 16.05.2023. URL https://standards.ieee.org/beyond-standards/the-evolution-of-wi-fi-technology-and-standards/, viitattu 09.03.2024.

[43] ITEAD INTELLIGENT SYSTEMS CO.,LTD. Sonoff Zigbee 3.0 USB Dongle Plus-E. URL https://itead.cc/product/zigbee-3-0-usb-dongle/, viitattu 08.05.2024.

[44] JOSE, A., JA MALEKIAN, R. Smart Home Automation Security: A Literature Review. *The Smart Computing Review* (08 2015), 269–285.

[45] KAAZ, K. J., HOFFER, A., SAEIDI, M., SARMA, A., JA BOBBA, R. B. Understanding user perceptions of privacy, and configuration challenges in home automation. Julkaisusarjassa *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (2017), 297–301.

[46] KAMBOURAKIS, G., KOLIAS, C., GENEIATAKIS, D., KAROPOULOS, G., MAKRAKIS, G. M., JA KOUNELIS, I. A State-of-the-Art Review on the Security of Mainstream IoT Wireless PAN Protocol Stacks. *Symmetry 12* (04 2020), 579/1–29.

[47] KHATTAK, S., NASRALLA, M., FARMAN, H., JA CHOUDHURY, N. Performance Evaluation of an IEEE 802.15.4-Based Thread Network for Efficient Internet of Things Communications in Smart Cities. *Applied Sciences 13* (06 2023), 1–23.

[48] KUMAR, S., ANDERSEN, M., KIM, H.-S., JA CULLER, D. Performant TCP for Low-Power Wireless Networks. Julkaisusarjassa *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI '20)* (02 2020), 911–932.

[49] LAMB, H. Smart home matters. *Engineering & Technology 18*, 1 (2023), 58–61.

[50] LASHKARI, A. H., DANESH, M. M. S., JA SAMADI, B. A survey on wireless security protocols (WEP, WPA and WPA2/802.11i). Julkaisusarjassa *2nd IEEE International Conference on Computer Science and Information Technology* (2009), 48–52.

[51] LASHKARI, B., CHEN, Y., JA MUSILEK, P. Energy Management for Smart Homes — State of the Art. *Applied Sciences 9* (2019), 3459/1–23.

[52] LI, W., YIGITCANLAR, T., EROL, I., JA LIU, A. Motivations, barriers and risks of smart home adoption: From systematic literature review to conceptual framework. *Energy Research & Social Science 80* (10 2021), 102211/1–29.

[53] LIU, Y., PANG, Z., DÁN, G., LAN, D., JA GONG, S. A Taxonomy for the Security Assessment of IP-Based Building Automation Systems: The Case of Thread. *IEEE Transactions on Industrial Informatics 14*, 9 (2018), 4113–4123.

[54] LUMI UNITED TECHNOLOGY CO., LTD. Aqara Door And Window Sensor P2. URL `https://www.aqara.com/eu/product/door-and-window-sensor-p2`, viitattu 08.05.2024.

[55] LUMI UNITED TECHNOLOGY CO., LTD. Aqara Temperature and Humidity Sensor T1. URL `https://www.aqara.com/en/temperature-and-humidity-sensor-t1`, viitattu 08.05.2024.

[56] MAKHADMEH, S. N., KHADER, A. T., AL-BETAR, M., NAIM, S., ABASI, A., JA ALYASSERI, Z. Optimization methods for power scheduling problems in smart home: Survey. *Renewable and Sustainable Energy Reviews 115* (2019), 109362/1–15.

[57] MOCRII, D., CHEN, Y., JA MUSILEK, P. IoT-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet of Things 1-2* (09 2018), 81–98.

[58] MOHAN, S., J, N. A., JA SITHARTHAN, R. Enhanced Home Automation and Security Using IoT Architecture. Julkaisusarjassa *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)* (2022), 1–6.

[59] MOISSINAC, K., RAMOS, D., RENDON, G., JA ELLEITHY, A. Wireless Encryption and WPA2 Weaknesses. Julkaisusarjassa *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)* (2021), 1007–1015.

[60] MORITA, P., SUNDAR SAHU, K., JA OETOMO, A. Health Monitoring Using Smart Home Technologies: Scoping Review. *JMIR mhealth and uhealth 11* (04 2023), 1–16.

[61] NANOLEAF. Nanoleaf Essentials Matter Smart Bulb. URL https://nanoleaf.me/en-EU/products/essentials/bulbs/?category=A60-E27&standard=matter&size=each, viitattu 08.05.2024.

[62] NEDIS. SmartLife Liiketunnistin. URL https://nedis.fi/fi-fi/product/turvallisuus-ja-valvonta/varoittimet/liiketunnistin/550726145/smartlife-liiketunnistin-zigbee-30-paristokayttoinen-ip20-tunnistuskulma-120-0-tunnistusalue-5-m-maks-akunkesto-12-kuukautta-valkoinen, viitattu 08.05.2024.

[63] NIKOUKAR, A., RAZA, S., POOLE, A., GÜNEŞ, M., JA DEZFOULI, B. Low-Power Wireless for the Internet of Things: Standards and Applications. *IEEE Access 6* (2018), 67893–67926.

[64] NORDIC SEMICONDUCTOR. Thread device types, päivitetty 19.04.2024. URL https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/device_types.html, viitattu 22.04.2024.

[65] NORDIC SEMICONDUCTOR. Wi-Fi overview, päivitetty 18.12.2023. URL https://developer.nordicsemi.com/nRF_Connect_SDK/doc/2.5.1/nrf/protocols/wifi/wifi.html, viitattu 10.03.2024.

[66] NORDIC SEMICONDUCTOR. Zigbee quick start guide. URL https://developer.nordicsemi.com/nRF_Connect_SDK/doc/2.5.1/nrf/protocols/zigbee/qsg.html, viitattu 22.03.2024.

[67] NXP SEMICONDUCTORS. ZigBee 3.0 Stack User Guide, päivitetty 11.09.2018. URL https://www.nxp.com/docs/en/user-guide/JN-UG-3113.pdf, viitattu 22.03.2024.

[68] OLIVEIRA, L., RODRIGUES, J. J., KOZLOV, S. A., RABÊLO, R. A. L., JA ALBUQUERQUE, V. H. C. MAC Layer Protocols for Internet of Things: A Survey. *Future Internet 11*, 1 (2019), 16/1–42.

[69] ORFANOS, V. A., KAMINARIS, S. D., PAPAGEORGAS, P., PIROMALIS, D., JA KANDRIS, D. A Comprehensive Review of IoT Networking Technologies for Smart Home Automation Applications. *Journal of Sensor and Actuator Networks 12*, 2 (2023), 30/1–31.

[70] RAGHAVENDRAN, D. C. V. Internet of Things – A Big Challenge in getting the Right Protocol. *International Advanced Research Journal in Science, Engineering and Technology 4* (07 2017), 180–192.

[71] RASPBERRY PI. Raspberry Pi 3 Model B+. URL https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus, viitattu 08.05.2024.

[72] RONDÓN, R., MAHMOOD, A., GRIMALDI, S., JA GIDLUND, M. Understanding the Performance of Bluetooth Mesh: Reliability, Delay, and Scalability Analysis. *IEEE Internet of Things Journal 7*, 3 (2020), 2089–2101.

[73] ROSCIA, M., DANCU, V., JA LAZAROIU, G. C. Smart Home Survey Analysis. Julkaisusarjassa *2023 IEEE International Smart Cities Conference (ISC2)* (2023), 1–5.

[74] RZEPECKI, W., JA RYBA, P. IoTSP: Thread Mesh vs Other Widely used Wireless Protocols – Comparison and use Cases Study. Julkaisusarjassa *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)* (2019), 291–295.

[75] SAMUEL, S. S. I. A review of connectivity challenges in IoT-smart home. Julkaisusarjassa *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)* (2016), 1–4.

[76] SANGOLLI, S., JA THYAGARAJAN, J. TCP Throughput Measurement and Comparison of IEEE 802.11 Legacy, IEEE 802.11n and IEEE 802.11ac Standards. *Indian Journal of Science and Technology 8* (08 2015), 1–8.

[77] SARMIENTO, O., GUERRERO, F., JA ARGOTE, D. Basic security measures for IEEE 802.11 wireless networks. *Ingeniería e Investigación 28* (08 2008), 89–96.

[78] SCHOMAKERS, E.-M., BIERMANN, H., JA ZIEFLE, M. Users' Preferences for Smart Home Automation – Investigating Aspects of Privacy and Trust. *Telematics and Informatics 64* (07 2021), 101689/1–16.

[79] SETZ, B., GRAEF, S., IVANOVA, D., TIESSEN, A., JA AIELLO, M. A Comparison of Open-Source Home Automation Systems. *IEEE Access 9* (2021), 167332–167352.

[80] SHASHWAT, K., HAHN, F., OU, X., JA SINGHAL, A. Security Analysis of Trust on the Controller in the Matter Protocol Specification. Julkaisusarjassa *2023 IEEE Conference on Communications and Network Security (CNS)* (2023), 1–6.

[81] SHELBY, Z., JA BORMANN, C. *6LoWPAN: The Wireless Embedded Internet*. John Wiley and Sons, Ltd, 2009.

[82] SHIN, J., PARK, Y., JA LEE, D. Who will be smart home users? An analysis of adoption and diffusion of smart homes. *Technological Forecasting and Social Change 134* (2018), 246–253.

[83] SILICON LABORATORIES. Introducing Zigbee Direct. URL `https://docs.silabs.com/zigbee/7.4.1/zigbee-direct/`, viitattu 19.03.2024.

[84] SILICON LABORATORIES. Key Benefits of Wi-Fi 6, julkaistu 8/2023. URL `https://www.silabs.com/documents/public/presentations/wf-101-key-benefits-of-wi-fi-6-what-comes-next.pdf`, viitattu 10.03.2024.

[85] SILICON LABORATORIES. Matter Connectivity Standard FAQ: Volume 2. URL `https://pages.silabs.com/rs/634-SLU-379/images/Matter-Connectivity-Standard-FAQ-V2.pdf`, viitattu 19.04.2024.

[86] SILICON LABORATORIES. UG103.14: Bluetooth LE Fundamentals. URL `https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf`, viitattu 14.03.2024.

[87] SILICON LABORATORIES. UG103.16: Multiprotocol Fundamentals Rev. 0.4, julkaistu 2023. URL `https://www.silabs.com/documents/public/user-guides/ug103-16-multiprotocol-fundamentals.pdf`, viitattu 08.05.2024.

[88] SILICON LABORATORIES. Z-Wave. URL https://www.silabs.com/wireless/z-wave, viitattu 31.03.2024.

[89] SILICON LABORATORIES. Z-Wave Global Regions. URL https://www.silabs.com/wireless/z-wave/global-regions, viitattu 27.03.2024.

[90] SILICON LABORATORIES. UG103.2: Zigbee Fundamentals, 2021. URL https://www.silabs.com/documents/public/user-guides/ug103-02-fundamentals-zigbee.pdf, viitattu 19.03.2024.

[91] SILICON LABORATORIES. AN1233: Zigbee Security, 2023. URL https://www.silabs.com/documents/public/application-notes/an1233-zigbee-security.pdf, viitattu 22.03.2024.

[92] SINGH, J., PASQUIER, T., BACON, J., KO, H., JA EYERS, D. Twenty Security Considerations for Cloud-Supported Internet of Things. *IEEE Internet of Things Journal 3*, 3 (2016), 269–284.

[93] SOVACOOL, B. K., JA FURSZYFER DEL RIO, D. D. Smart home technologies in Europe: A critical review of concepts, benefits, risks and policies. *Renewable and Sustainable Energy Reviews 120* (2020), 109663/1–20.

[94] STOLOJESCU-CRISAN, C., CRISAN, C., JA BUTUNOI, B.-P. An IoT-Based Smart Home Automation System. *Sensors 21* (05 2021), 3784/1–23.

[95] THE INTERNATIONAL TELECOMMUNICATION UNION. *ITU-T G.9959*, 2015. URL https://www.itu.int/rec/T-REC-G.9959-201501-I.

[96] THREAD GROUP. Elegantly Connecting Your Smart Home Network, julkaistu 9/2023. URL https://www.threadgroup.org/Portals/0/documents/ElegantlyConnectingYourSmartHomeNetworkWhitePaper_4431_1.pdf, viitattu 15.04.2024.

[97] THREAD GROUP. Thread 1.2 Base Features, julkaistu 06/2019. URL https://www.threadgroup.org/Portals/0/documents/support/Thread%201.2%20Base%20Features.pdf, viitattu 06.04.2024.

[98] THREAD GROUP. Thread 1.2 in Commercial White Paper, julkaistu 09/2019. URL https://www.threadgroup.org/Portals/0/documents/support/ThreadInCommercialWhitePaper_2542_1.pdf, viitattu 08.04.2024.

[99] THREAD GROUP. Thread 1.3.0 Features White Paper, julkaistu 19.07.2022. URL https://www.threadgroup.org/Portals/0/documents/support/Thread1.3.0WhitePaper_07192022_3990_1.pdf, viitattu 06.04.2024.

[100] THREAD GROUP. Thread 1.3.0 Webinar, julkaistu 29.09.2022. URL https://portal.threadgroup.org/DesktopModules/Inventures_Document/FileDownload.aspx?ContentID=4076, viitattu 05.05.2024.

[101] THREAD GROUP. Thread Border Router White Paper, julkaistu 19.07.2022. URL https://www.threadgroup.org/Portals/0/documents/support/ThreadBorderRouterWhitePaper_07192022_4001_1.pdf, viitattu 18.04.2024.

[102] THREAD GROUP. Thread Commissioning, julkaistu 13.07.2015. URL https://www.threadgroup.org/Portals/0/documents/support/CommissioningWhitePaper_658_2.pdf, viitattu 16.04.2024.

[103] THREAD GROUP. Thread FAQ. URL https://www.threadgroup.org/support, viitattu 06.04.2024.

[104] THREAD GROUP. Thread Introduction, julkaistu 15.07.2014. URL https://www.threadgroup.org/portals/0/documents/thread_introduction_website_7-15-14.pdf, viitattu 06.04.2024.

[105] THREAD GROUP. Thread Network Fundamentals, julkaistu 28.09.2022. URL https://portal.threadgroup.org/DesktopModules/Inventures_Document/FileDownload.aspx?ContentID=633, viitattu 01.04.2024.

[106] THREAD GROUP. Thread Stack Fundamentals, julkaistu 13.07.2015. URL https://www.threadgroup.org/Portals/0/documents/support/ThreadOverview_633_2.pdf, viitattu 08.04.2024.

[107] THREAD GROUP. Thread Technical Overview, julkaistu 05.10.2015. URL https://www.threadgroup.org/portals/0/documents/resources/Thread_Technical_Overview.pdf, viitattu 06.04.2024.

[108] THREAD GROUP. Thread Usage of 6LoWPAN, julkaistu 13.07.2015. URL https://www.threadgroup.org/Portals/0/documents/support/6LoWPANUsage_632_2.pdf, viitattu 15.04.2024.

[109] TP-LINK CORPORATION PTE. LTD. TP-Link Tapo P110M. URL https://www.tp-link.com/fi/home-networking/smart-plug/tapo-p110m/, viitattu 08.05.2024.

[110] UNWALA, I., TAQVI, Z., JA LU, J. Thread: An IoT Protocol. Julkaisusarjassa *2018 IEEE Green Technologies Conference (GreenTech)* (2018), 161–167.

[111] VASSEUR, J.-P., JA DUNKELS, A. *Interconnecting Smart Objects with IP The Next Internet*. Elsevier Inc, 2010.

[112] VOX MEDIA, LLC. Matter: what you need to know, julkaistu 06.10.2022. *The Verge* (2022). URL https://www.theverge.com/23390726/matter-smart-home-faq-questions-answers, viitattu 05.05.2024.

[113] VOX MEDIA, LLC. How to pick a smart home platform, julkaistu 12.06.2023. *The Verge* (2023). URL https://www.theverge.com/23751295/smart-home-platform-google-amazon-apple-samsung, viitattu 18.02.2024.

[114] WI-FI ALLIANCE. Security. URL https://www.wi-fi.org/discover-wi-fi/security, viitattu 09.03.2024.

[115] WI-FI ALLIANCE. Wi-Fi Alliance introduces Wi-Fi CERTIFIED WPA3 security. URL https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-wi-fi-certified-wpa3-security, viitattu 09.03.2024.

[116] WI-FI ALLIANCE. Wi-Fi CERTIFIED HaLow: Long range, low power Wi-Fi. URL https://www.wi-fi.org/system/files/Wi-Fi_CERTIFIED_HaLow_Highlights.pdf, viitattu 09.03.2024.

[117] WI-FI ALLIANCE. Wi-Fi Direct. URL https://www.wi-fi.org/discover-wi-fi/wi-fi-direct, viitattu 09.03.2024.

[118] WI-FI ALLIANCE. Wi-Fi EasyMesh. URL https://www.wi-fi.org/discover-wi-fi/wi-fi-easymesh, viitattu 09.03.2024.

[119] WU, D., FENG, W., LI, T., JA YANG, Z. Evaluating the intelligence capability of smart homes: A conceptual modeling approach. *Data & Knowledge Engineering 148* (2023), 102218/1–18.

[120] YALDAIE, A., PORRAS, J., JA DRÖGEHORN, O. Who are Smart Home Users and What do they Want? – Insights from an International Survey. *Applied Computer Systems 28* (08 2023), 114–124.

[121] YIN, J., YANG, Z., CAO, H., LIU, T., ZHOU, Z., JA WU, C. A Survey on Bluetooth 5.0 and Mesh: New Milestones of IoT. *ACM Transactions on Sensor Networks 15*, 3 (05 2019), 1–29.

[122] Z-WAVE ALLIANCE. What is Z-Wave Long Range and How Does it Differ from Z-Wave? URL https://z-wavealliance.org/what-is-z-wave-long-range-how-does-it-differ-from-z-wave/, viitattu 27.03.2024.

[123] Z-WAVE ALLIANCE. Z-Wave Compatible Products. URL https://www.z-wave.com/shop-z-wave-smart-home-products, viitattu 31.03.2024.

[124] Z-WAVE ALLIANCE. *Z-Wave Device Class Specification*, 2021. URL https://z-wavealliance.org/development-resources-overview/specification-for-developers/, viitattu 29.03.2024.

[125] Z-WAVE ALLIANCE. *Application Work Group Z-Wave Specifcations Release 2023B*, 2023. URL https://z-wavealliance.org/development-resources-overview/specification-for-developers/.

[126] Z-WAVE ALLIANCE. *Z-Wave and Z-Wave Long Range Network Layer Specification*, 2023. URL https://z-wavealliance.org/development-resources-overview/specification-for-developers/, viitattu 29.03.2024.

[127] Z-WAVE ALLIANCE, INC. Z-Wave Certification: The Key To Interoperability. URL https://z-wavealliance.org/interoperability/, viitattu 28.02.2024.

[128] ZEGEYE, W., JEMAL, A., JA KORNEGAY, K. Connected Smart Home over Matter Protocol. Julkaisusarjassa *2023 IEEE International Conference on Consumer Electronics (ICCE)* (2023), 1–7.

[129] ÇETINKAYA, O., JA AKAN, O. *Use of Wireless Sensor Networks in Smart Homes*. Taylor & Francis Group, 04 2016, ss. 233–258.