



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MIRKA JÄÄSKELÄINEN
DDS REFERENSSIARKKITEHTUURIN SOVELTAMINEN
ASIOIDEN INTERNETISSÄ

Diplomityö

Tarkastaja: professori Jouni Mattila
Tarkastaja ja aihe hyväksytty
Teknisten tieteiden tiedekuntaneu-
voston kokouksessa 17. elokuuta
2016

TIIVISTELMÄ

MIRKA JÄÄSKELÄINEN: DDS referenssiarkkitehtuurin soveltaminen asioiden internetissä

Diplomityö, 47 sivua

Marraskuu 2016

Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: Koneautomaatio

Tarkastaja: professori Jouni Mattila

Avainsanat: Asioiden internet, Data Distribution Service, referenssiarkkitehtuuri

Tämän diplomityön tarkoituksena on tutkia mitä ovat asioiden internet (IoT) ja Data Distribution Service (DDS). Tarkoituksena on myös tutkia miten DDS:n datan kirjoittajat, datan lukijat sekä julkaisija-tilaaja –malli toimivat. DDS:n toimintaa havainnollistetaan työtä varten rakennetulla Simulink-mallilla. Malli rakennettiin käyttäen Simulinkin DDS-kirjastoa sekä RTI:n DDS-kirjastoa. Käytännön osion tarkoituksena on tutkia miten DDS toimii ja kuinka datan siirto tapahtuu tietokoneelta toiselle.

IoT on älykkäiden laitteiden verkko. Laitteet ovat kytketty osaksi internetiä. Ne lähettävät dataa pilvipalveluun, josta dataa voidaan hajauttaa kohteille, jotka sitä tarvitsevat. IoT:n toiminta perustuu koneiden väliseen tiedonvälitykseen (M2M), joskin IoT on näistä kahdesta kehittyneempi ratkaisu sekä käyttää internetiä tiedon siirtoon ja jakamiseen. Ratkaisut ovat kuitenkin vasta alkutaipaleellaan siitä huolimatta, että tietoa löytyy paljon ja yritykset ovat siitä kiinnostuneita. IoT:n standardit ovat vielä melko hajanaisia ja yhtä vakiintunutta arkkitehtuuria ei ole. Kehitystä kuitenkin tapahtuu kokoajan ja tulevaisuudessa tilanne tulee olemaan parempi. Hallitsevan protokollan löytyminen edistäisi IoT:n suosiota entisestään.

DDS on datan hajauttamiseen sovellettava standardi, joka mahdollistaa kommunikaation järjestelmän fyysisen tason ja sovellustason välillä. DDS integroi systeemin komponentit yhteen ja tarjoaa sujuvaa datan siirtoa sekä luotettavuutta. Se tarjoaa myös arkkitehtuurin liiketoimintaa ja asioiden internetiä varten. DDS ei käytä toiminnassaan erilisiä palvelimia, vaan sen käsittelemä data varastoidaan globaaliin tietovarastoon, joten sen luotettavuus lisääntyy. DDS:n huono puoli on sen monimutkaisuus. Se ei siis sovi yksinkertaisiin järjestelmiin.

Kokonaisuudessaan IoT ja DDS ovat tulevaisuutta. DDS ei välttämättä ole tulevaisuudessa suosituin välikerrosohjelmisto, mutta IoT:n suosio on varmaa. Yritykset hyötyvät tällaisista ohjelmistoista paljon ja ne edistävät liiketoimintaa. Sujuva tiedonsiirto ja datan käsittely ovat nykyaikaa.

ABSTRACT

MIRKA JÄÄSKELÄINEN: Applying DDS reference architecture to Internet of Things

Tampere University of Technology

Master of Science Thesis, 47 pages

November 2016

Master's Degree Programme in Automation Engineering

Major: Machine Automation

Examiner: Professor Jouni Mattila

Keywords: Internet of Things, Data Distribution Service, reference architecture

Purpose of this Master's theses is to study what are Internet of Things and Data Distribution Service (DDS). Part of this goal is to clarify how DDS data writer, data reader and publish-subscribe model works. Operation of DDS is demonstrated by Simulink model that is built using DDS toolbox and RTI DDS library. Goal of the model is to show how DDS operates and how data is transferred from machine to another.

Internet of Things is web of intelligent devices, which are connected to Internet. Sensors send data to cloud service where data can be distributed to targets, which needs it. Operation of IoT is based on M2M communication though IoT is the more advanced one and uses Internet to transfer and share data. Internet of Things is on it's prime factor even though a lot of information is available and corporations are interested about it. IoT standards are still quite scattered. There is no one major solution for the architecture. In the future situation will be better because development happens all the time. Dominant protocol would though promote Internet of Things a lot.

Data Distribution Service (DDS) is middleware protocol that distributes data. It's a standard that enables communication between physical layer and application layer. DDS integrates components of the system together and provides fluent data sharing. DDS enables architecture for IoT and business. Operation of DDS does not rely on single servers but global data storage, which makes operation more reliable. Downside of DDS is complexity making it inefficient for small system

Outright IoT and DDS are part of the future. DDS might not be the most popular middleware software but the popularity of Internet of Things is guaranteed. Corporations get advantage out of this kind of software, which help business forward. Efficient data transfer and handling is part of modern life.

ALKUSANAT

Haluan kiittää professori Jouni Mattilaa työn tarkistuksesta ja neuvoista. Tämä työ sientöi opiskelija-ajan Hervannassa, josta jäi hyviä muistoja. Haluan kiittää myös perhettä ja ystäviä tuesta ja kannustuksesta sekä ihmisen parasta ystävää loputtomasta kärsivällisyydestä.

Hämeenlinnassa, 23.11.2016

Mirka Jääskeläinen

SISÄLLYSLUETTELO

1.	JOHDANTO	1
2.	ASIOIDEN INTERNET	3
2.1	Määritelmä	3
2.2	Perustana koneiden välinen tiedonvälitys	3
2.3	Asioiden internet	5
2.4	Arkkitehtuuri	6
2.4.1	Referenssimalli ja arkkitehtuuri	7
2.4.2	Tietoaluemalli	9
2.4.3	Informaatiomalli	10
2.4.4	Funktionaalinen malli	10
2.4.5	Kommunikaatiomalli	13
2.4.6	Turvallisuusmalli	13
2.5	Tulevaisuus	15
3.	DATA DISTRIBUTION SERVICE	19
3.1	Historia	19
3.2	Määritelmä	19
3.3	Globaali tietovarasto	21
3.4	Quality of Service	22
3.5	Data Distribution For Real-time Systems	23
3.5.1	Data-Centric Publish-Subscribe	23
3.5.2	Data Local Reconstruction Layer	24
3.6	The Real-time Publish-Subscribe Wire Protocol	25
3.7	Datakeskisyys vs serveripohjaiset ratkaisut	26
3.8	DDS – ei palvelinta	27
3.9	Miksi IoT ja DDS?	28
4.	DDS-JÄRJESTELMÄ	30
4.1	Simulointimalli	30
4.2	Järjestelmän välittämä data	37
4.3	Laskenta	40
4.4	Päätelmät	42
5.	YHTEENVETO	44
	LÄHTEET	46

LYHENTEET JA MERKINNÄT

ARM	engl. Architecture Reference Model, arkkitehtuurinen referenssimalli
DCPS	engl, Data-Centric Publish-Subscribe, datakeskeinen julkaisija-tilaaja -malli
DDS	engl, Data Distribution Service, datan jakelujärjestelmä
DLRL	engl, Data Local Reconstruction layer, datan jälleenrakennuskerros
ICT	engl, Information and Communications Technology, tieto- ja viestintäteknologia
IP	engl, Internet Protocol, internetprotokolla
IoT	engl. Internet of Things, asioiden internet
LAN	engl, Local Area Network, lähiverkko
M2M	engl. Machine to Machine, koneiden välinen tiedonvälitys
PC	engl, Personal Computer, henkilökohtainen tietokone
PIM	engl, Platform Independent Model, alusta, joka ei ole tietystä teknologiasta riippuvainen
QoS	engl. Quality of Service, palvelun laatu
RFID	engl. Radio Frequency Identification, radiotaajuinen etätunnistus
RTPS	engl, Real-Time Publish-Subscribe, reaaliaikainen julkaisija-tilaaja -malli
UDP	engl, User Datagram Protocol, yhteydetön protokolla
UML	engl, Unified Modelling Language, graafinen mallinnuskieli
WAN	engl, Wide Area Network, laajaverkko

1. JOHDANTO

Asioiden internet on tämän hetken kuumimpia puheenaiheita ja kehityksen kohteita. IoT oli esimerkiksi Embedded World –näyttelyn pääteema vuonna 2016 Nürnbergissä Saksassa. Embedded World –näyttelyn kohderyhmä on sulautettujen järjestelmien alan ammattilaiset sekä opiskelijat. Tällä kertaa esittelyn ja korostuksen kohteena oli asioiden internetin tietoturva. [1]

IoT on monen teknologiayrityksen kehityksen kohde ja puheenaihe. Se liittää laitteet, koneet, anturit, palvelut, ohjelmistot ja ihmiset yhteen internetin välityksellä. Sen tarjoamat mahdollisuudet ovat rajattomat ja teknologiaa hyödynnetään jo monessa kohteessa. Solidan digiliiketoiminnan kehittäjä Ossi Lindroos kertoo Ylen haastattelussa 19.4.2016: ”Internetin tuoma muutos on pientä siihen verrattuna, mitä esineiden internet tuo mukanaan. Me tulemme näkemään kodinkoneita, jotka tilaavat itse täydennystä. Esimerkiksi tulostin voi tilata omat musteensa tai pesukone voi tilata omat pesupulverinsa”. [2] Tämä kuvaa hyvin asioiden internetin asemaa tulevaisuudessa. IoT:n kannalta on tärkeää ymmärtää sen monimutkaisuus sekä tunnistaa alueet, joissa siitä on merkittävää hyötyä. Yritykset valmistautuvat digitaaliseen sekasortoon, jolloin päätöksentekijöiden tulisi tunnistaa ja priorisoida mahdollisuudet sekä riskit. Päätöksentekijöiden tulisi tunnistaa tärkeimpiä elementtejä, joita vaaditaan tehokkaiden sovellusten kehittämiseen. [3]

Tällä hetkellä asioiden internetin markkinat ovat hieman hajallaan ja osiltaan jopa alkutekijöissään. Ratkaisut ovat hajanaisia ja ne kohdistuvat vain tiettyyn alaan tai sovellukseen. Ratkaisujen hajanaisuutta kuvaa myös eri alustojen, protokollien ja käyttöliittymien lukumäärä. Tästä seuraa, että eri toimittajien tarjoamat sovellusten komponentit eivät välttämättä ole yhteensopivia. Tästä johtuen myös hinnat pysyvät korkeina. Standardoituja protokollia ja käyttöliittymiä kuitenkin on ja niitä kehitetään jatkuvasti uusia. Kuitenkaan ei ole syntynyt yhtä protokollan, käyttöliittymän ja alustan yhdistelmää, joka olisi ylitse muiden. Toisin kuin esimerkiksi matkapuhelinverkoissa, dominoivaa yhdistelmää ei ole löytynyt, ratkaisuilla on korkeat hinnat, hallitseva referenssiarkkitehtuuri puuttuu ja toimittajat valitsevat mielensä mukaan komponentit, asioiden internetin omaksuminen on ollut melko hidasta. [4, s. 10-11]

Asioiden internetin odotettu voimakas kasvu on siis riippuvainen yleisten standardien ja alustojen synnystä. Toisaalta niiden syntyminen riippuu siitä kehitetäänkö standardeja ja alustoja vaatimukset täyttäviin asioiden internetin tietoalueisiin. Äärimmäisessä tapauksessa tietoalueet eivät täytä vaatimuksia, jolloin sovellukset eivät jaa yleisiä standardeja ja asioiden internetin lopullinen läpilyönti voi jopa hidastua. Liiketoiminta asioiden in-

ternetin ympärillä on muodostunut tietyistä teknologiaratkaisuista, jotka usein keskittyvät tiettyihin sovellusalueisiin. Esimerkiksi RFID, jonka ratkaisuja myydään vähittäiskaupassa sekä koneiden välinen tiedonvälitys, jota käytetään etäyhteyden muodostamiseen ovat tällaisia sovellusalueita. [4, s. 10-11]

Tämän diplomityön tavoitteena on selvittää mitä ovat asioiden internet sekä Data Distribution Service (DDS) ja miten ne liittyvät toisiinsa. Asioiden internetin rooli tulevaisuudessa tulee kasvamaan ja sen mukana myös välikerrosohjelmistot kehittyvät. Diplomityön aiheeksi valittiin asioiden internetin ja DDS:n niiden ajankohtaisuuden vuoksi.

Työ on rajattu siten, että ainoa tarkasteltava välikerrosohjelmisto on DDS. Asioiden internetin kanssa voi käyttää myös muita välikerrosohjelmistoja. Työn tavoitteena on siis myös tarkastella miksi DDS sopii käytettäväksi asioiden internetin välikerrosohjelmistona. Asioiden internetin sovelluksille on saatavilla useita eri protokollia. Esimerkiksi Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP) ja Constrained Application Protocol (CoAP) ovat protokollia, joita voidaan käyttää asioiden internetin yhteydessä DDS:n tapaan. Kuitenkin edellä mainitut protokollat ovat yksinkertaisia ja käyttötapaukset ovat rajallisia. DDS puolestaan on ominaisuuksiltaan monipuolinen ja asioiden internetin tarjoaman monimutkaisen datan käsittely onnistuu siltä hyvin. [5]

Diplomityön sisältö koostuu johdannosta, teoreettisesta taustasta, teorian sovellusluvusta sekä yhteenvedosta ja päätelmistä. Luvussa 2 käsitellään asioiden internetiä ja sen taustaa. Siinä esitellään mikä asioiden internet on, mihin sitä käytetään ja mihin sen toiminta perustuu.

Luvussa 3 jatketaan taustateorian käsittelyä. DDS määritellään ja kerrotaan miten se rakentuu. Luvussa esitellään myös eri protokollia sekä havainnollistetaan DDS:n toimintaa. Teoriakappaleiden jälkeen siirrytään teorian sovelluslukuun numero 4. Teorian sovellusluvussa on esitelty Matlabin Simulinkillä toteutettu malli, joka mallintaa DDS:n toimintaa. Malli on demo, jonka tarkoituksena on havainnollistaa miten DDS:n lukijat, kirjoittajat, julkaisijat ja tilaajat toimivat. Lopuksi yhteenvedossa kootaan työn tärkeimmät asiat yhteen ja muodostetaan päätelmiä asioiden internetin ja DDS:n toiminnasta.

2. ASIOIDEN INTERNET

Termi asioiden internet (Internet of Things, IoT) tuli esille ensimmäisen kerran vuonna 1999, kun Kevin Ashton käytti termiä radiotaajuisen etätunnistuksen (Radio Frequency Identification, RFID) yhteydessä. Termin käyttö kuitenkin laajeni ja pian se oli käytössä monien eri teknologioiden yhteydessä. [6] Asioiden internet tunnetaan myös nimillä esineiden internet, Internet of Everything, teollisuuden internet ja Machine to Machine. [7, s. 4]

Tässä luvussa käsitellään asioiden internetin ominaisuuksia, terminologiaa, historiaa ja mahdollisuuksia. Luku tarjoaa myös katsauksen siihen miten koneiden välinen tiedonsiirto kehittyi lopulta asioiden internetiksi. Tässä luvussa tarkastellaan myös arkkitehtuurin ja referenssiarkkitehtuurin käsitteitä.

2.1 Määritelmä

Internet of Things (IoT) tunnetaan monilla nimillä. Tunnetuimpia ovat Internet of Objects eli esineiden internet sekä Internet of Things eli asioiden internet. Kun asioiden internetiä käytetään teollisuudessa, puhutaan usein teollisuuden internetistä. Teollisuuden internetissä yhdistyvät monet eri tekniikan osa-alueet ja tiedonsiirtoratkaisut. [8] Asioiden internet tarjoaa usean eri kokoluokan ratkaisuja ja järjestelmiä verkon ominaisuuksista riippuen. Esimerkiksi jos asioiden internetiä käytetään suljetussa ympäristössä, voidaan puhua asioiden intranetistä (Intranet of Things). Asioiden intranet on tosin melko harvoin käytetty ratkaisu. [7, s. 14]

Asioiden internetissä kuten myös teollisuuden internetissä on nimensä mukaisesti kyse näiden asioiden keskinäisestä kommunikoinnista internetin välityksellä. Asioita teollisuuden internetissä voivat olla esimerkiksi anturit ja ohjelmoitavat logiikat. [8] Asioiden internetissä sovelluksia ja palveluita on lukemattomia. Osa on jo olemassa, mutta osa on vasta suunnittelun tasolla. Sovelluksia voivat olla esimerkiksi turvallisuuteen, kustannusten pienentämiseen, liiketoiminnan optimointiin tai avustettuun asumiseen liittyvät sovellukset. Asioiden internet on eniten käytetty yrityksissä sekä teollisuudessa. [7, s. 16]

2.2 Perustana koneiden välinen tiedonvälitys

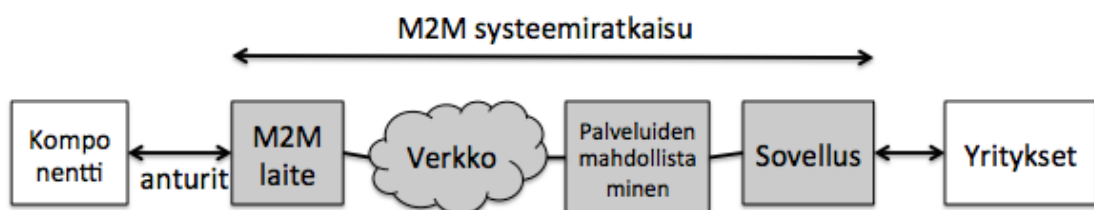
Asioiden internet käyttää suurelta osin samaa tekniikkaa kuin koneiden välinen (Machine-To-Machine, M2M) tiedonvälitys. Koneiden välisessä kommunikoinnissa yhdiste-

tään antureita ja muita laitteita tieto- ja viestintäteknologian (Information and Communication Technology, ICT) järjestelmiin joko langattomasti tai langallisesti. Asioiden internetin pohjana on siis koneiden välinen tiedonvälitys.

Sekä koneiden välinen tiedonvälitys että asioiden internet ovat viime vuosikymmenien teknologian kehityksen tuloksia. Internet protokollan (Internet Protocol, IP) nousu teki internetistä maailmanlaajuisen. Samaan aikaan antureiden ja sähköisten tunnisteiden käyttö on lisääntynyt ja kehitys on ollut nopeaa. Myös hintojen alentuminen ja internetiin pääsyn helpottuminen on ollut hyvä alku teollisen internetin kehittymiselle. Tuloksena on internetin yli kommunikoidut laitteet, joiden kustannukset ovat pienet.

Koneiden välinen tiedonvälitys tarkoittaa sovelluksia, jotka sallivat kommunikaation eri laitteiden välillä. Nämä laitteet ovat keskenään samantyyppisiä ja niissä on tietynlaiset sovellukset. Kaikki tiedonsiirto tapahtuu johtojen välityksellä tai langattomasti. Koneiden välisessä tiedonvälityksessä käyttäjä voi esimerkiksi kerätä dataa lämpötila-antureista tai nosturin tärinästä. Usein kerättyä dataa analysoidaan ja sitten käytetään turvallisuuden parantamiseen ja kustannusten vähentämiseen.

Tyypillinen koneiden välisen tiedonvälityksen ratkaisu (kuva 1) koostuu laitteista ja tietoverkosta, joka mahdollistaa etäyhteyden laitteiden välillä. Siihen kuuluvat myös sovelluksen logiikka ja esimerkiksi M2M-sovelluksen integrointi yrityksen järjestelmän prosesseihin. [7 s. 12]



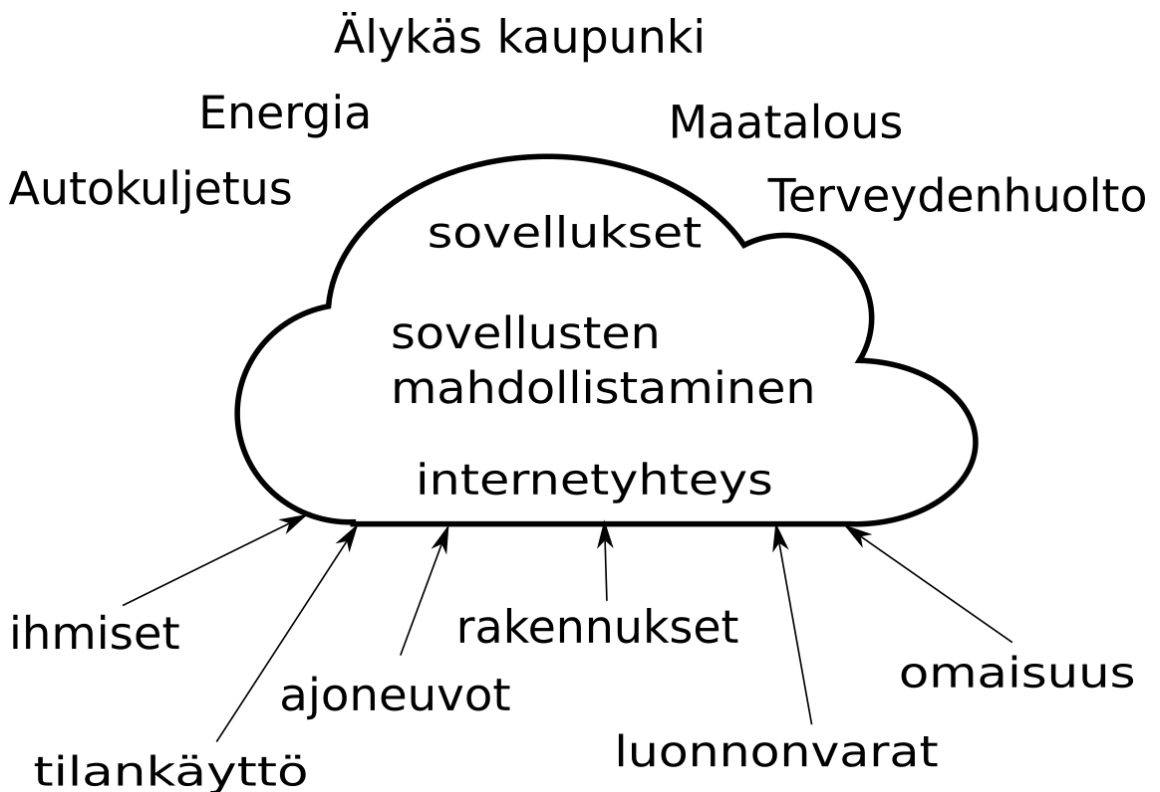
Kuva 1 M2M systeemiratkaisu [7, s. 12]

M2M systeemiratkaisua käytetään ohjaamaan yritysten komponentteja etäyhteyden avulla. Komponentit voivat olla esimerkiksi ajoneuvoja, rahtikontteja tai rakennuksia. Kuvan 1 mukaiset M2M systeemiratkaisun osat ovat M2M laite, verkko, palvelut ja sovellus. M2M laite on yhdistetty komponenttiin, josta halutaan tietoa. Laite kerää tietoa esimerkiksi anturoinnin avulla ja voi vaihdella yksinkertaisesta laitteesta hyvinkin monimutkaiseen ja älykkääseen laitteeseen. Verkko puolestaan tarjoaa etäyhteyden M2M laitteen ja sovelluspuolen serverin välille. Verkko voi olla laajaverkko (WAN, Wide Area Network) tai lähiverkko (LAN, Local Area Network). M2M systeemiratkaisuun kuuluu myös erillinen komponentti, joka mahdollistaa palvelut. Sen tarkoituksena on vähentää toteutuksen kustannuksia ja vähentää sovellusten kehittämistä tarjoamalla toiminnallisuutta, joka on yhteistä kaikille sovelluksille. Sovellus puolestaan on ohjauksen ja monitoroinnin tulos. Se on integroitu yrityksen prosesseihin. [7, s. 13]

2.3 Asioiden internet

Asioiden internet ja koneiden välinen tiedonvälitys eroavat toisistaan siten että asioiden internet viittaa antureiden ja muiden laitteiden laajempaan yhteyteen ja kommunikointiin internetin välityksellä. Enää internetissä ei ole pelkästään ihmisten ja median sisältöä, vaan siellä on myös älykkäiden koneiden ja laitteiden verkosto, joka kommunikoi ihmisten kanssa, tukee liiketoimintaa ja luo tietämystä. Teollisuuden internet ei ole oma internetinsä, vaan sen eräänlainen osa.

Kuva 2 havainnollistaa asioiden internetin käsitettä. Asioiden internet koostuu siis ratkaisuksista, jotka on yhdistetty pilveen. Pilvi on yleensä useassa datakeskuksessa toimiva joukko palvelimia. Pilvessä sijaitsevat varsinaiset sovellukset sekä internetyhteydet. Nämä ratkaisut ovat samankaltaisia koneiden välisen tiedonvälityksen sovellusten kanssa. Asioiden internetissä sovellukset voivat yhdistellä ja käyttää dataa eri lähteistä, jolloin voidaan saavuttaa tulevaisuudessa esimerkiksi älykkäitä kaupunkeja. Älykkäässä kaupungissa energiantuotanto, lämmitys, jätteet, kuljetus ja esimerkiksi vedenpuhdistus ovat kaikki integroituina toisiinsa. Tällä tavoin kaupungista saadaan tehokas ja ympäristöystävällinen asioiden internetin avulla.



Kuva 2. Asioiden internet [7 s. 15]

Myös muut sovellukset ovat mahdollisia asioiden internetissä. Mobiililaitteisiin voi ladata lukemattomia sovelluksia, joilla voi mitata sykettä, kulutettuja kaloreita, askeleita tai nopeutta. Nämä tiedot voidaan ajaa pilveen, jossa palvelin käsittelee dataa ja luo asiakkaalle esimerkiksi taulukoita ja kuvaajia mittausdatasta ja niiden yhdistelmistä.

Mahdollisuuksia on lukemattomia, mutta niillä on myös kääntöpuolensa. Valvonta voi mennä myös liiallisuuksiin.

Internetiin yhdistettyjen laitteiden määrä kasvaa jatkuvasti ja kasvaa hyvin todennäköisesti jatkossakin, koska yhä useampi ihminen omistaa enemmän kuin yhden laitteen, jossa on internetyhteys. Nämä laitteen kommunikoivat ja tarjoavat palveluja internetin yli sekä ovat yhdistettyinä toisiinsa. Tällöin on syntynyt sekä sosiaalisen ja teknisen innovaation uusi aalto. [7, s. 3-4]

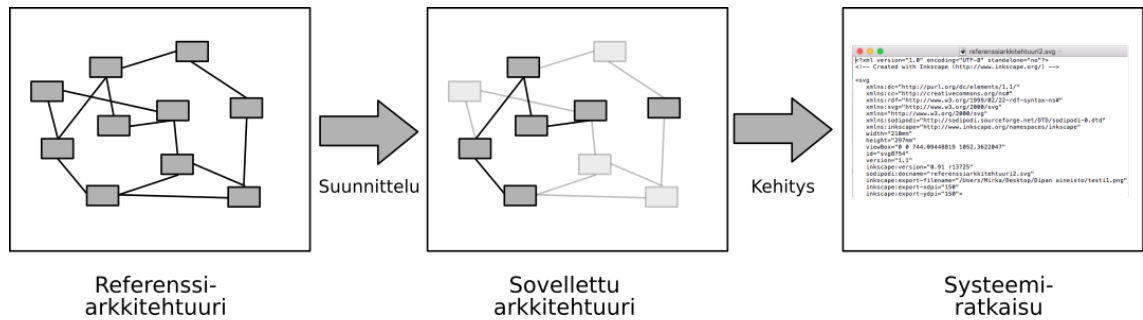
Koneiden välinen tiedonvälitys ja asioiden internet eroavat toisistaan monin tavoin, mutta niillä on myös samoja ominaisuuksia. Koneiden välisen tiedonsiirron ratkaisut keskittyvät usein ratkaisemaan tietyn ongelman, joka yrityksellä tai sidosryhmällä on. Sillä siis harvoin on laajaa näkemystä eri ongelmista ja ratkaisuista, jotka sisältävät esimerkiksi usean eri sidosryhmän. Tällöin koneiden välisen tiedonsiirron laitteet ovat usein tiettyyn käyttötarkoitukseen ja tietylle sovellukselle tehtyjä, jolloin niitä ei voi integroida muihin laitteisiin tai järjestelmiin. M2M-sovelluksilla on pitkä historia ja ratkaisut ovat usein teollisuudessa ja käytetty teknologia on sirpaloitunutta. [7, s. 36-37]

Kehitys koneiden välisestä tiedonsiirrosta asioiden internetiin tapahtui siirtymällä pois koneiden välisen tiedonsiirron suppeista käytöistä. Aloitettiin kehittää avointa, moniin käyttötarkoituksiin sopivaa innovaatiota. Asioiden internetin kehityksessä siirryttiin siis pois eristäytyneistä ratkaisuista ja kehitettiin tilalle avoin ympäristö. Tässä avoimessa ympäristössä IP on teknologian työkalulaatikko, internet on hankkeen pohja ja antureina toimivat mitä erikoisemmat asiat. [7, s. 36-37]

2.4 Arkkitehtuuri

Arkkitehtuurilla tarkoitetaan käsitteellisten osatekijöiden ja varsinaisten osatekijöiden välistä suhdetta. Käsitteellisiä osatekijöitä voivat olla esimerkiksi funktiot, data tai erilaiset palvelut. Varsinaisilla osatekijöillä puolestaan tarkoitetaan usein protokollaa. Arkkitehtuurin yhteydessä käytetty termi referenssiarkkitehtuuri viittaa yleiseen malliin, joka sisältää monipuolisen joukon osatekijöitä, jotka kuuluvat jollain tapaa asioiden internetiin.

Kun suunnitellaan sovellusta tai ratkaisua tiettyyn ongelmaan, käytetään usein referenssiarkkitehtuuria apuna luomaan varsinainen arkkitehtuuri. Varsinainen arkkitehtuuri on siis piirustus, jonka avulla ratkaistaan ongelma. Referenssiarkkitehtuurin avulla luodaan varsinainen arkkitehtuuri ja varsinaisen arkkitehtuurin avulla kehitetään ratkaisu ongelmaan tai esimerkiksi uusi sovellus kuten kuvassa 3 esitetään.



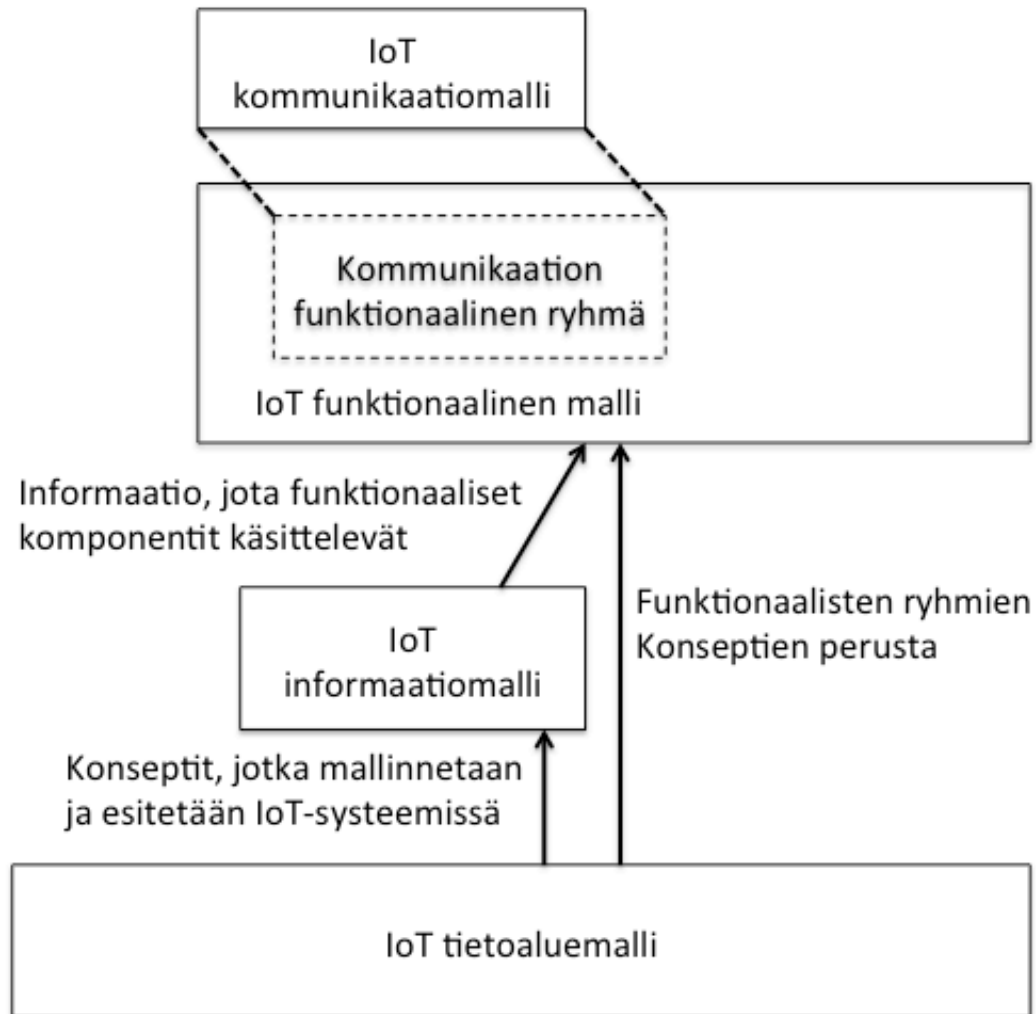
Kuva 3 Referenssiarkkitehtuurista systeemiratkaisuun [7 s. 62]

Arkkitehtuurinen referenssimalli (Architectural Reference Model, ARM) eroaa arkkitehtuurista ja referenssiarkkitehtuurista siten että siinä ei luoda yksittäistä arkkitehtuuria, vaan se on referenssimallin ja referenssiarkkitehtuurin yhdistelmä. Asioiden internetin arkkitehtuurin (IoT-A) visio on saavuttaa korkean tason yhteentoimivuus eri asioiden internetin ratkaisuiden ja järjestelmän eri tasojen tiedonvälityksen, palveluiden ja informaation välillä. Osa tätä visiota on myös arkkitehtuurinen referenssimalli. Asioiden internetin arkkitehtuuri tarjoaa valikoiman erilaisia arkkitehtuurisia näkökulmia ja esittää kuinka voi konkreettisia arkkitehtuureja, jotka perustuvat käyttötapauksiin ja vaatimuksiin. [7, s. 61-64]

Kuvasta 3 voidaan nähdä arkkitehtuurisen referenssimallin kaksi näkökulmaa. Ensimmäinen näyttää kuinka käytännössä luodaan IoT ARM ja toinen näyttää miten sitä käytetään oikean järjestelmän rakentamiseen. Jotta referenssiarkkitehtuurista päästään systeemiratkaisuun, tarvitaan suunnittelua, insinööriä, rakentamista ja testausta eli kehitystyötä. Kuvan 3 esitys toimii myös toiseen suuntaan. Systeemiratkaisu antaa palautetta ja parannusehdotuksia arkkitehtuurille. Arkkitehtuuri puolestaan välittää niistä oleelliset eteenpäin referenssiarkkitehtuurille. [7, s. 170]

2.4.1 Referenssimalli ja arkkitehtuuri

ARM eli arkkitehtuurinen referenssimalli koostuu kahdesta osasta, referenssimallista ja referenssiarkkitehtuurista. IoT referenssiarkkitehtuurin perustana on referenssimalli. Referenssimalli määrittelee tietoalueen (domain) alimallien avulla kuten kuvassa 4 on esitetty. Arkkitehtuurimallin tietoaluemalli (Domain Model) pitää hallussaan konseptin, joka kuuluu tälle kyseiselle tietoalueelle. Tässä tapauksessa konseptina ovat koneiden välinen tiedonvälitys ja asioiden internet. Tietoaluemalli lisää myös kuvauksia tai määritelmiä konseptien välille. Nämä konseptit ja mallien väliset suhteet tarjoavat pohjan informaatiomallin (Information Model) kehitykselle. Tietoaluemallissa ja informaatiomallissa operoiva systeemi sisältää konsepteja ja entiteettejä, jotka täytyy määritellä erillisessä mallissa eli funktionaalisessa mallissa (Functional Model). Nämä alimallit, jotka ovat osa IoT referenssimallia, sekä muutama muu esitellään myöhemmin kappaleessa 2.4.4. [7, s. 168-170]



Kuva 4 IoT referenssimalli [7, s.168]

Erillään referenssimallista on myös toinen ARM-systeemiin kuuluva komponentti eli referenssiarkkitehtuuri. Systeemiarkkitehtuuri on kommunikaation väline, jota käytetään sidosryhmien kesken. Kehittäjillä, liikekumppaneilla, toimittajilla ja asiakkailla on kaikilla eri näkemys järjestelmään. Näkemys määräytyy vaatimusten ja yksittäisen toimijan omien toiminta- ja käyttötapojen perusteella. Arkkitehtuuri määrittellään korkealla tasolla ja abstraktisti kun taas arkkitehtuuri oikeassa järjestelmässä ei ole niin abstrakti. Määrittely on vaikeaa, koska sidosryhmien eri näkemykset ja käsitteet ovat vaikeaselkoisia. Kuitenkin on päätetty, että tätä abstraktia käsitettä kutsutaan referenssiarkkitehtuuriksi, koska se toimii varsinaisissa järjestelmissä. Referenssiarkkitehtuuri myös ottaa haltuun arkkitehtuurin tärkeitä osia kuten suunnitteluperusteet, ohjenuorat ja entiteetit, jotta seuranta ja vuorovaikutus fyysisen maailman kanssa onnistuu. [7, s. 168-170]

2.4.2 Tietoaluemalli

Tietoaluemallin tarkoitus on kuvailla ja mallintaa reaalisen maailman entiteettejä ja niiden välisiä suhteita. Sen tarkoituksena on myös ymmärtää systeemitason vaatimuksia, tunnistaa tietoalueen entiteettejä sekä auttaa ammatinharjoittajia suunnittelemaan järjestelmiä. Järjestelmien tulee olla huollettavia ja testattavia sekä ominaisuuksien tulee täyttää kasvavat vaatimukset. Järjestelmää kehitettäessä usein syntyy hankaluuksia ongelma-alueen ja vaatimusten tulkinnan välille. Tällöin tietoaluemalli auttaa selkiyttämään tilannetta ja tarjoaa ratkaisuja ongelmiin ja tulkintahankaluuksiin. Tietoaluemalli siis visoi ratkaisun, joka on tietoalueiden joukko. Tämä joukko tekee yhteistyötä täyttääkseen systeemitason vaatimukset.

Tietoaluemalli määrittelee jonkin tietyn alueen ominaisuudet. Tässä tapauksessa kiinnostuksen kohde ja alue on tietenkin asioiden internet. Alueen ominaisuuksien oletetaan pysyvän muuttumattomina, vaikka itse ARM muokkautuu jatkuvasti ajan myötä. Tietoaluemalli pitää hallussaan tärkeimpien konseptien merkittävimmät ominaisuudet sekä näiden konseptien väliset suhteet. Se toimii myös kommunikaatioväylänä käyttäjien välillä sekä kommunikoi eri tietoalueille olevien käyttäjien kanssa. Tietoaluemalli on myös pohja muille alimalleille.

Arkkitehtuurisen referenssimallin yhteydessä käytetään termejä fyysinen entiteetti ja virtuaalinen entiteetti. Fyysinen entiteetti kuvataan digitaalisessa maailmassa virtuaaliseksi entiteetiksi. Virtuaalinen entiteetti voi olla esimerkiksi tietokantaan kirjautuminen, geologinen malli, kuva, video tai mikä tahansa muu digitaalinen asia. Yksi fyysinen entiteetti voidaan esittää usean virtuaalisen entiteetin avulla. Tällöin jokainen virtuaalinen entiteetti palvelee eri tarkoitusta esimerkiksi kuva parkkipaikasta jonkin auton ajotietokoneessa. Jokaisella virtuaalisella entiteetillä on myös ainutlaatuinen tunniste, jonka avulla se erotetaan muista digitaalisista tuotteista. Virtuaalisen entiteetin esitysmuoto ja fyysisen entiteetin varsinainen tila ovat synkronoituja toisiinsa, kun käyttäjä käyttää jompaakumpaa niistä kahdesta. Asioiden internetissä fyysiset entiteetit ovat usein antureita, sensoreita ja tunnisteita. Tunnisteiden tarkoituksena on merkitä fyysinen entiteetti, johon ne liittyvät. Oikeassa järjestelmässä tunnisteet voivat olla joko laitteita tai fyysisiä entiteettejä, eivät molempia.

Fyysisten entiteettien identifiointi on tärkeä osa-alue asioiden internetin järjestelmässä. Jotta käyttäjä voisi olla vuorovaikutuksessa fyysisen maailman kanssa digitaalisen maailman välityksellä, tarvitaan kaksi asiaa. Ensimmäinen on ensisijainen tunnistaminen eli käytetään fyysisen entiteetin luonnollisia ominaisuuksia. Toinen on toissijainen tunnistaminen eli käytetään tunnisteita, jotka liittyvät fyysiseen entiteettiin. Molemmat tunnistamisen tyypit on mallinnettu asioiden internetin tietoaluemalliin. [7, s. 171-182]

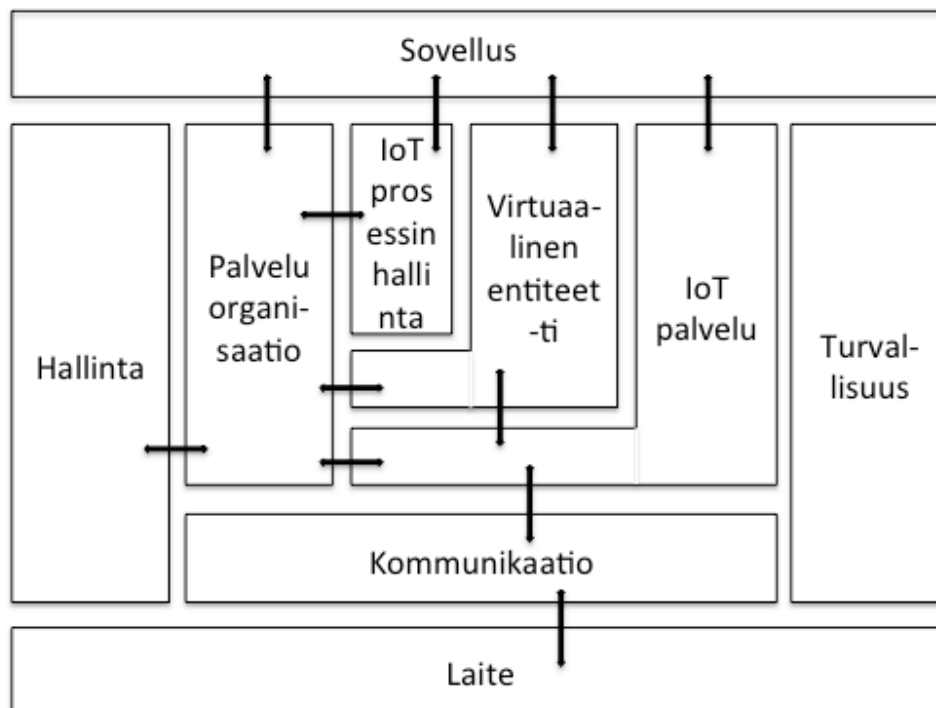
2.4.3 Informaatiomalli

Informaatiomalli sisältää virtuaalisten entiteettien tarpeellisen informaation sekä niiden ominaisuuksia. Nämä ominaisuudet voivat olla staattisia tai dynaamisia. Ne saapuvat järjestelmään manuaalisesti syötettynä datana tai virtuaaliseen entiteettiin kiinnitetyn anturin lähettämänä signaalina. Virtuaalisten entiteettien ominaisuudet voivat olla myös digitaalisesti synkronoituja kopioita anturin tilasta. Kaikkia entiteettejä ei kuitenkaan huomioida tässä. Entiteettejä, jotka voisivat sisältyä oikeaan järjestelmään, ovat esimerkiksi huoneiden nimet ja kiinteistön kerrosten numerot. Nämä ovat entiteettejä, jotka eivät liity asioiden internetin laitteisiin tai niiden toimintaan, mutta ovat kuitenkin tärkeitä itse järjestelmälle.

Informaatiomalli mallintaa kaikki tietoa-alueen käsitteet, jotka selvästi esitetään ja joita käsitellään digitaalisessa maailmassa. Informaatiomalli mallintaa siis käsitteitä näiden kahden kerroksen välillä. Se toimii myös metamallina, jonka tehtävä on luoda rakennetta informaatiolle, jota käsitellään asioiden internetin järjestelmissä. Tämä rakenne luo pohjan kaikelle järjestelmässä tapahtuvalle esittämiselle, keräykselle, prosessoinnille, varastoinnille ja informaation keräämiselle. Tietoa-alueen tärkeimpiä käsitteitä, jotka sijaitsevat asioiden internetin järjestelmässä, ovat virtuaalinen entiteetti sekä palvelu. Jälkimmäinen myös vertailee näkökantoja resurssien ja laitteiden välillä. Virtuaalinen entiteetti on puolestaan fyysisen entiteetin esitysmuoto digitaalisessa maailmassa. Informaatiomallissa ei ole muuta esitysmuotoa fyysisille entiteeteille. [7, s. 182-189]

2.4.4 Funktionaalinen malli

Funktionaalisen mallin (Functional Model) (kuva 5) tarkoitus on määrittellä funktionaaliset ryhmät (Functionality Groups) ja kuvata niiden vuorovaikutusta arkkitehtuurisen referenssimallin kanssa. Tässä kappaleessa esitellään tärkeimmät funktionaaliset ryhmät. Kuten kuvasta 5 nähdään, funktionaalisia ryhmiä on yhdeksän.



Kuva 5 Funktionaalinen malli [7]

Funktionaalinen malli (kuva 5) määrittelee joukon funktioita, jotka varmistavat järjestelmän luotettavan toiminnan. Kuvassa 5 nuolet kuvaavat funktionaalisten ryhmien välistä vuorovaikutuksia. Se myös määrittelee entiteettien väliset suhteet. Funktionaalinen malli tarjoaa pohjan ja viitekehyksen, johon voi esimerkiksi perustaa järjestelmän luotettavuusstandardit. Luotettavuusstandardit luovat vaatimukset vastuussa olevista entiteeteistä, jotka suorittavat tarvittavia funktioita tässä mallissa.

Laite (Device) on funktionaalinen ryhmä, joka sisältää kaikki mahdolliset toiminnallisuudet, joita fyysiset laitteet tarjoavat. Fyysiset laitteet ovat fyysisten entiteettien työkaluja. Tämä laite-toiminnallisuus sisältää antureita, prosessointia, varastointia ja komponenttien tunnistusta sekä oppimista, joka riippuu laitteen kapasiteetista.

Funktionaalinen ryhmä kommunikaatio (Communication) määrittelee kaikki mahdolliset kommunikointimekanismit, joita laitteet käyttävät oikeassa järjestelmässä. Tässä tapauksessa kommunikointia käytetään tiedon siirtämiseksi digitaaliseen maailmaan tai toisille laitteille. Kommunikaatioon käytettyjä teknologioita ei käytetä sovellusten ja muiden funktioiden välillä. Muita funktioita voivat olla funktiot, jotka ovat peräisin asioiden internetin palvelusta. Niiden kanssa ei käytetä näitä kommunikaatiotapoja, koska ne ovat perinteistä internet-teknologiaa.

Kolmas funktionaalinen ryhmä on asioiden internetin palvelut (Internet of Things Service). Se viestittää pääosin tietoaluemallin palvelut-luokalle. Funktionaalinen ryhmä palvelut sisältää yksittäisiä asioiden internetin palveluita, joita resurssit ovat tuoneet esille tietoverkosta tai laitteista. Asioiden internetin palvelua tukevia funktioita ovat hakemistoon liittyvät palvelut, jotka mahdollistavat palvelun löytämisen sekä voimavarojen erottelukyvyn.

Funktionaalinen ryhmä virtuaalinen entiteetti (Virtual Entity) viestittää tietoaluemallin virtuaalisille entiteettien kanssa. Se myös sisältää tarpeellisia toimintoja assosiaatioiden hallitsemiseen entiteettien itsensä sekä muiden funktionaalisten ryhmien välillä. Liitokset voivat olla staattisia tai dynaamisia riippuen fyysisten ja virtuaalisten entiteettien ominaisuuksista. Esimerkki staattisesta liitoksesta on rakennuksen käytävien, kerrosten, käytävien ja huoneiden hierarkia. Dynaaminen liitos puolestaan voi olla esimerkiksi auton liikkuminen kaupungista toiseen. Suurin ero asioiden internetin palveluiden ja virtuaalisten entiteettien palveluiden välillä ovat näiden palveluiden vaatimukset ja vastheet. Virtuaaliset entiteetit palvelu tarjoaa tiedon merkityksellisessä muodossa, jolloin ihmisten on sitä helppo lukea ja ymmärtää.

Funktionaalinen ryhmä palveluorganisaation (IoT Service Organization) tarkoitus on isännöidä kaikkia funktionaalisia komponentteja, jotka tukevat asioiden internetin palveluiden ja virtuaalisten entiteettien palveluiden muodostamista. Toisaalta tämä funktionaalinen ryhmä toimii lähinnä toimittajana muiden funktionaalisten ryhmien välillä. Esimerkiksi kun funktionaalista ryhmästä sovellus tulee vaatimus, palveluorganisaatio ohjaa sen oikeaan kohteeseen. Palveluorganisaatio siis tukee virtuaalisten entiteettien assosiaatioita, joilla on tekemistä asioiden internetin palvelun kanssa. Palveluorganisaatio sisältää myös toiminnallisuuksia esimerkiksi havaintoihin ja muodostamiseen liittyviä palveluita.

Funktionaalinen ryhmä prosessin hallinta (Internet of Things Process Management) on kokoelma eri toiminnallisuuksia. Näiden toiminnallisuuksien tarkoituksena on saada aikaan sujuva yhteistyö asioiden internetin palveluiden (IoT Services, Virtual Entity Services, Composed Services) ja liiketoiminnan välillä.

Funktionaalinen ryhmä hallinta (Management) puolestaan sisältää sellaisia toiminnallisuuksia, jotka mahdollistavat järjestelmän virheiden ja suorituskyvyn valvonnan. Myös joustavuuden mahdollistaminen on tärkeää, jotta asetuksia voidaan muuttaa ja järjestelmää voidaan muokata vastaamaan käyttäjän vaatimuksiin. Hallintaa tukevia funktioita ovat esimerkiksi omistajuus ja johto. Myös tiedon varastointi kuuluu hallinnan tehtäviin.

Funktionaalinen ryhmä turvallisuus (Security) sisältää sellaiset toiminnalliset komponentit, jotka varmistavat turvallisen toiminnan sekä järjestelmän yksityisyyden. Tämän funktionaalisen ryhmän toimintaan kuuluu yksityisyyden mekanismien hallinta

kuten anonymiteetti, kerätty data, lähteen anonymiteetti ja palveluun pääsyn rajoittaminen. Myöskään ulkopuolinen tarkkailija ei voi vetää johtopäätöksiä järjestelmän käyttäjän vaatimuksista, koska ne ovat suojattuja.

Funktionaalinen ryhmä sovellus (Application) on käytännössä vain muodollisuus, joka edustaa kaikkea tarvittavaa logiikkaa, jota tarvitaan asioiden internetin sovelluksen luomiseen. Sovellukset usein sisältävät määrättyyn tietoon räätälöityä dataa. Sovellus voi olla myös osa jotakin itseään isompaa tietotekniikkajärjestelmää, joka käyttää toiminnassaan asioiden internetiä. Tällaista käytetään usein teollisuudessa ja esimerkiksi tehtaissa. [7, s. 189-194]

2.4.5 Kommunikaatiomalli

Kommunikaatiomalli (Communication Model) koostuu järjestelmän alku- ja loppupisteiden tunnistamisesta ja yksilöinnistä. Se tunnistaa myös monilähetyksen (Multicast) ja yksilölähetyksen (Unicast) sekä teknologiat näiden vuorovaikutusten takana. Tässä kapaleessa kerrotaan miten kommunikaatiomalli tunnistaa päätepisteet tiedonvälityksen reiteiltä.

Todennäköisiä päätepisteitä tiedonvälityksessä ovat käyttäjät sekä laitteet tietoluemalissa. Käyttäjät sisältävät henkilöitä, jotka käyttävät järjestelmää, sekä digitaalisia toiminnallisuuksia kuten järjestelmän ulkoisia sovelluksia, palveluita ja järjestelmän sisäisiä komponentteja. Laitteet koneen ja ihmisen välisessä rajapinnassa toimivat vuorovaikutuksen välittäjänä käyttäjän ja fyysisen maailman välillä. Näitä laitteita voivat olla esimerkiksi näppäimistö, painikkeet sekä kosketusnäytöt. Tällöin ihminen ei ole kommunikaatiomallin päätepiste. [7, s. 194-195]

2.4.6 Turvallisuusmalli

Viimeinen funktionaalinen ryhmä liittyy turvallisuuteen, yksityisyyteen, luotettavuuteen ja varmuuteen (Safety, privacy, trust, security model). Asioiden internetiä hyödyntävä järjestelmä mahdollistaa ihmisten ja laitteiden välisen kommunikoinnin. Tällöin ihmiset ovat osa järjestelmää, jolloin mahdollisten virheiden mahdollisuus kasvaa ja huomioon täytyy ottaa myös tietoturvaluus. Tarvitaan siis järjestelmä, joka takaa myös käyttäjien yksityisyyden ja turvallisuuden.

Järjestelmän turvallisuus on sovelluskohtaista. Yleensä turvallisuus täytyy huomioida erityisesti sellaisissa järjestelmissä, jotka ovat tekemisissä ihmisten tai eläinten kanssa. Esimerkiksi hissien automaattisissa ovissa voi olla asioiden internetin järjestelmä. Turvallisuus täytyy huomioida, jotta ovien avautuminen ja sulkeutuminen eivät aiheuta ihmisille vahinkoa. Myös infrastruktuurin varjeleminen on osa turvallisuutta, koska infrastruktuurin menettäminen voi olla vahingollista ihmisille. Usein tarvitaan myös me-

kaaninen turvajärjestelmä sähköisen lisäksi. Esimerkiksi sähkökatkojen vuoksi on usein ajateltava asiaa usealta kannalta.

Vuorovaikutus fyysisen maailman kanssa vaatii myös ihmisten yksityisyyden suojaamista. Asioiden internetin yksityisyydessä otetaan huomioon seuraavat toiminnalliset komponentit: identiteetin hallinta, autenttisuus, käyttöoikeus sekä luottamus. Identiteetin hallinta esittää usean eri identiteetin samalle arkkitehtuuriselle entiteetille, jolloin ei voida tietää mikä alkuperäinen identiteetti oli. Autenttisuus puolestaan on toiminnallisuus, joka sallii identiteetin tarkistamisen oli se sitten kehitetty tai alkuperäinen identiteetti. Käyttöoikeus on toiminnallisuus, jonka tehtävänä on sallia pääsyoikeudet käyttäjille, kun käyttäjät pyrkivät vuorovaikutukseen palveluiden ja laitteiden kanssa. Luottamus puolestaan ylläpitää dynaamisia ja staattisia suhteita vuorovaikutuksessa olevien entiteettien välillä. Nämä suhteet vaikuttavat vuorovaikutaviin entiteettien käyttäytymisiin. Esimerkiksi jos jokin laite vaikuttaa epäluotettavalta (anturien mittaama data on yleisen mittausalueen ulkopuolella), toinen entiteetti voi estää laitteet tai anturin toiminnan tietyssä laitteessa.

Luottamus on tärkeää asioiden internetin järjestelmässä. Erityisesti silloin kun järjestelmä sisältää yksityistä dataa täytyy tietoturvan olla luotettava. Turvallisuusmallin tavoitteena onkin varmistaa, että järjestelmässä kaikki on niin kuin on oletettu. Poikkeaviin tilanteisiin pitää reagoida heti. Seuraavat näkökannat ovat turvallisuusmallin kannalta olennaisia:

- Luotettavuusmallin tietoalue eli määritellään entiteetit, joilla on keskenään samanlaiset luotettavuuden vaatimukset, samaan tietoalueeseen. Näin tehdään, koska toistensa kanssa vuorovaikutuksessa olevia entiteettejä on paljon. Luotettavuuden määrittely kaikkien entiteettien välille olisi kohtuutonta.
- Luotettavuuden arvioinnin mekanismit eli kuvataan esimerkiksi kuinka paljon pisteitä luotettavuus saa kunkin entiteetin kohdalla. Laskentamekanismin tulee ottaa huomioon mistä tieto on peräisin. Tiedon lähde siis vaikuttaa entiteetin saamiin luotettavuuspisteisiin. Samantapainen asioiden internetin luotettavuutta mittaava konsepti laskee laitteiden, resurssien ja palveluiden luotettavuuden taso.
- Luotettavuuden käyttäytymisen linjaus eli linjaus, joka ohjaa käyttäytymistä vuorovaikuttavien entiteettien välillä. Linjaus perustuu näiden entiteettien välillä olevaan luotettavuustasoon. Esimerkiksi miten käyttäjä voisi käyttää anturin mitaustuloksia, jotka on noudettu jostakin palvelusta, jolla on matala luotettavuuden taso.
- Luotettavuusankkuri eli entiteetti, joka on määritelty luotettavaksi sellaisten entiteettien toimesta, jotka kuuluvat samaan luotettavuusalueeseen. Luotettavuusankkuria käytetään myös määrittelemään kolmannen entiteetin luotettavuutta.

- Luotettavuuden liitto eli liitto kahden tai useamman luotettavuusmallin välillä. Se sisältää joukon sääntöjä, jotka määrittelevät miten käsitellään eri luotettavuuden suhteita. Nämä luotettavuuden suhteet ovat eri entiteettien välillä ja eri luotettavuusmalleissa. Liitto on merkityksellinen vain järjestelmissä, joissa on suuri mittakaava.

Asioiden internetin turvallisuusmalli (Security Model) koostuu kommunikoiden turvallisuudesta, joka keskittyy suurilta osin tietosuojaan sekä kokonaisuuden suojaukseen. Suojauksen vaikutuspiirissä ovat entiteetit ja funktionaaliset komponentit kuten identiteetti, autenttisuus, käyttöoikeus ja luottamus. Nämä ovat asioiden internetin arkkitehtuurisen referenssimallin tärkeimmät piirteet. [7, s. 195-197]

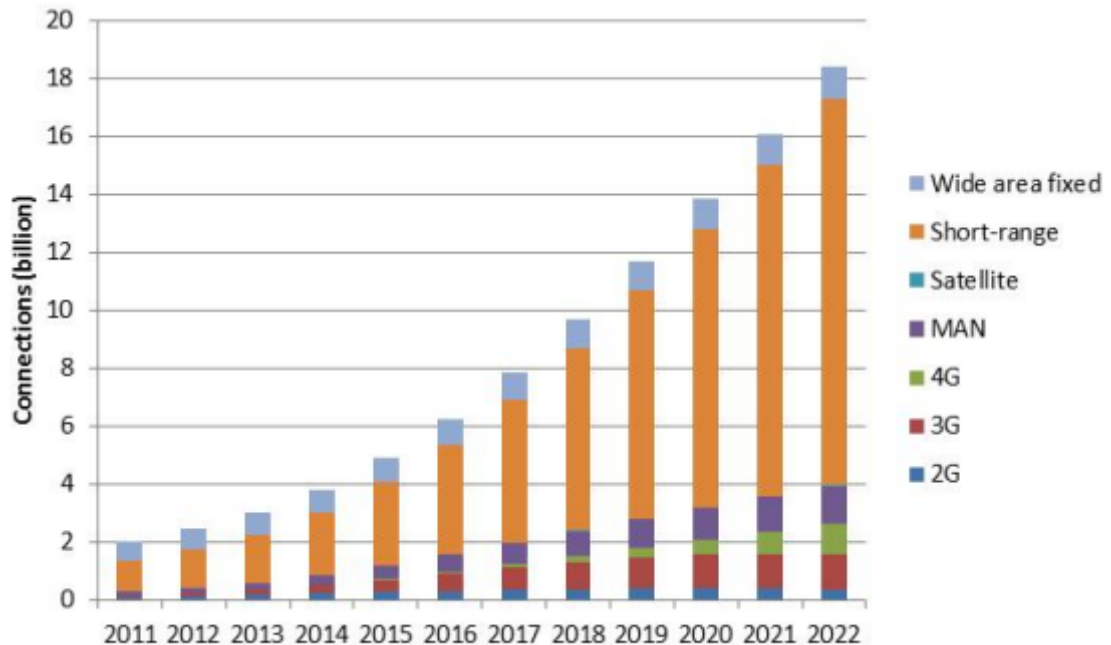
2.5 Tulevaisuus

Tällä hetkellä asioiden internetin markkinat ovat hieman hajallaan ja osiltaan jopa alkutekijöissään. Ratkaisut ovat hajanaisia ja ne kohdistuvat vain tiettyyn alaan tai sovellukseen. Ratkaisujen hajanaisuutta kuvaa myös eri alustojen, protokollien ja käyttöliittymien lukumäärä. Tästä seuraa, että eri toimittajien tarjoamat sovellusten komponentit eivät välttämättä ole yhteensopivia. Tästä johtuen myös hinnat pysyvät korkeina. Standardoituja protokollia ja käyttöliittymiä kuitenkin on ja niitä kehitetään jatkuvasti uusia. Kuitenkaan ei ole syntynyt yhtä protokollan, käyttöliittymän ja alustan yhdistelmää, joka olisi ylitse muiden. Toisin kuin esimerkiksi matkapuhelinverkoissa, dominoivaa yhdistelmää ei ole löytynyt, ratkaisuilla on korkeat hinnat, hallitseva referenssiarkkitehtuuri puuttuu ja toimittajat valitsevat mielensä mukaan komponentit, asioiden internetin omaksuminen on ollut melko hidasta. [4, s. 10-11]

Asioiden internetin kannalta on tärkeää ymmärtää sen monimutkaisuus sekä tunnistaa alueet, joissa siitä on merkittävää hyötyä. Yritykset valmistautuvat digitaaliseen sekasortoon, jolloin päätöksentekijöiden tulisi tunnistaa ja priorisoida mahdollisuudet sekä riskit. Päätöksentekijöiden tulisi tunnistaa tärkeimpiä elementtejä, joita vaaditaan tehokkaiden sovellusten kehittämiseen. Esimerkiksi tietohallintojohtajia tulisi rohkaista hankkimaan tietoa asioiden internetin sovelluksista ja niiden käytöllä saavutettavista hyödyistä. [3]

Asioiden internetin odotettu voimakas kasvu on siis riippuvainen yleisten standardien ja alustojen synnystä. Toisaalta niiden syntyminen riippuu siitä kehitetäänkö standardeja ja alustoja vaatimukset täyttäviin asioiden internetin tietoalueisiin. Äärimmäisessä tapauksessa tietoalueet eivät täytä vaatimuksia, jolloin sovellukset eivät jaa yleisiä standardeja ja asioiden internetin lopullinen läpilyönti voi jopa hidastua. Liiketoiminta asioiden internetin ympärillä on muodostunut tietyistä teknologiaratkaisuista, jotka usein keskittyvät tiettyihin sovellusalueisiin. Esimerkiksi RFID, jonka ratkaisuja myydään vähittäiskaupassa sekä koneiden välinen tiedonvälitys, jota käytetään etäyhteyden muodostamiseen ovat tällaisia sovellusalueita. [4, s. 10-11]

Vuoteen 2022 mennessä koneiden välisiä yhteyksiä tulee olemaan 18 miljardia, ennustetaan Matt Hattonin tutkimuksessa vuodelta 2012 [9]. Kuvaan 6 on hahmoteltu tätä kehitystä sekä laitteiden jakautumista teknologioiden kesken. Vuosittainen yhteyksien kasvuvauhti on noin 22%. Yli viidesosa kaikista laitteista on koneiden välisen tiedonvälityksen laitteita.

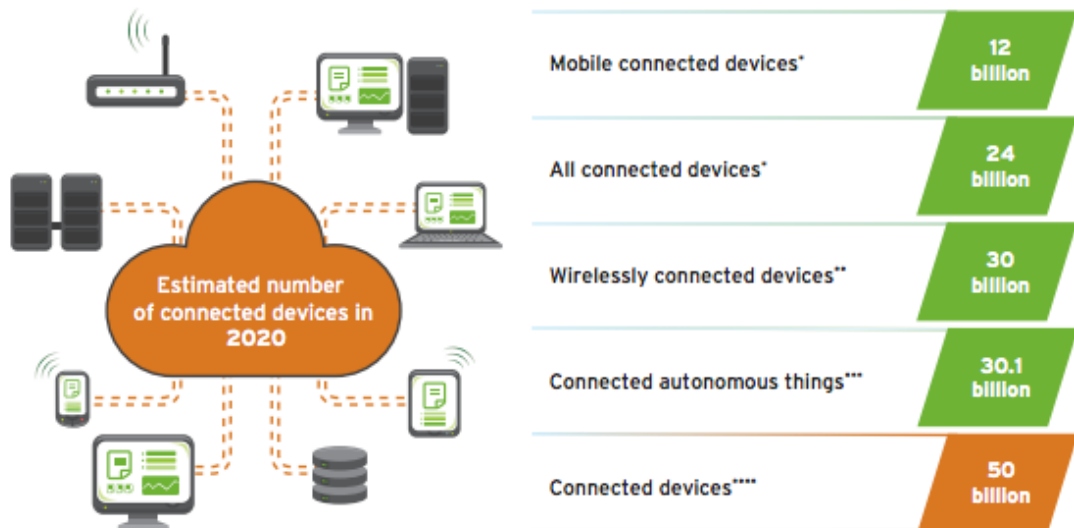


Kuva 6 Maailmanlaajuinen koneiden välinen tiedonvälitys [9]

Matt Hattonin tutkimuksessa ennustetaan myös älykkäiden rakennusten tulevan olemaan suurin asioiden internetiä hyödyntävä yksittäinen kohde käsittäen 37% yhteyksistä. Älykkäissä rakennuksissa muun muassa lämmitys, ilmastointi ja ilmanvaihto sekä turvallisuustekniikka on yhdistetty tietoverkkoon. Myös lyhyen kantaman yhteyksiä käyttävien laitteiden odotetaan valtaavan alaa M2M-tekniikalta. Suurin sektori tulee kenties olemaan rakennus- ja kiinteistöautomaatio. Koneiden välisen tiedonvälityksen kokonaisliikevaihto tulee olemaan 1,2 biljoonaa dollaria vuoteen 2022 mennessä. Noin kolmasosa liikevaihdosta tulee palveluista. Kaksi kolmasosaa liikevaihdosta puolestaan saadaan laitteista ja asennuksista. [9, s. 3-4]

Kiina ja USA tulevat olemaan suurimmat maailmanlaajuiset markkina-alueet koneiden välisen tiedonvälityksen sovelluksissa vuonna 2022. Seuraavaksi suurimpia alueita ovat Japani, Saksa ja Venäjä. Jos ajatellaan verkkoliikennettä, on vain pieni mahdollisuus, että koneiden väliset yhteydet voivat ylittää operaattorien verkkojen kapasiteetit. Koneiden välisen tiedonsiirron osuus matkapuhelinverkkojen kuormituksesta on tällä hetkellä noin kaksi prosenttia. Vuonna 2022 osuuden ennustetaan olevan alle yksi prosentti kokonaiskuormituksesta. Koneiden välisen tiedonvälityksen kasvua kutistaa laajakaistayhteyksien kasvu dataliikenteessä. [9, s. 4]

Asioiden internet on kasvanut idealistisesta konseptista realistiseksi ratkaisuksi. Asioiden internetin sovellusten suosio on monien tekijöiden summa. Asioiden internet muun muassa alentaa laitteiden kustannuksia, on laajalle levinnyt ja tarjoaa kokonaisvaltaisen yhteyden sekä keskittyy toimintojen tehokkuuteen ja tuottavuuteen. Miljardien älykkäiden laitteiden yhdistetään verkkoon lähitulevaisuudessa. Kuvassa 7 on ennustettu vuoden 2020 tilannetta sekä verkkoon yhdistettyjen laitteiden lukumäärää. [3, s. 5]



Kuva 7 Älykkäiden laitteiden ja antureiden lukumäärän kasvu [3, s. 5]

Edullisten teknologioiden ja antureiden yhdistelmä saa aikaan antureiden kustannuksien pienenemisen sekä laskenta- ja prosessoritehojen kasvun. Myös datan varastoinnin kustannukset pienenevät ja verkkoyhteydet paranevat. Nämä tekijät vaikuttavat asioiden internetin suosion kasvuun, koska käyttäjät hyötyvät sen ominaisuuksista aina vain enemmän. Esimerkiksi järjestelmät, jotka vaativat toimiakseen yhteyden, ovat edullisempia kuin ennen, koska koneiden välisen tiedonvälityksen kustannukset ovat pienentyneet. [3]

Yksi tapa edistää asioiden internetin yleistymistä on kehittää yritykselle vahvaa analyttistä kapasiteettia. Asioiden internet vaatii taitoa hallita ja analysoida suuria määriä yksinkertaista dataa, jonka kaltaista datan käsittelyä kutsutaan myös termillä big data. Perinteiset infrastruktuurit ovat yksinkertaisempia kuin asioiden internet ja vaativat vähemmän datankäsittelytaitoja. Yritysten tulee siirtää viimeisimmät kehitystyön tuloksensa pilvipalveluun ja ymmärtää datan analysoinnin merkitys, kun datan määrä ja laatu muuttuu nopeasti. Toinen vaihtoehto on rekrytoida ja kouluttaa uusia asioiden internetin osaajia. Teknologia ja kyvykkyys ovat tärkeitä, mutta tärkeää on myös hallita asioiden internetin monimutkaisuus ja suuret datamäärät. Asioiden ekosysteemit tulevat kasvamaan, joten analyttikoilla tulee olla perusteellista tietoa tietoaalueista tietyllä teollisuuden alalla. Tällöin voidaan varmistaa, että käytetään oikeaa dataa hyvien tulosten saavuttamiseksi. Tulevaisuudessa tietotekniikkaosaajille tulee olemaan entistä suurempaa kysyntää, johon ei välttämättä pystytä vastaamaan. Kolmas tapa edistää asioiden inter-

netin yleistymistä ja helpottaa sen käyttöönottoa on pyytää apua asiantuntijoilta, jolloin voidaan täyttää yrityksen kapasiteetti. Yritykset todennäköisesti rekrytoivat mielellään ulkopuolista apua, jolloin voidaan varmistaa että asioiden internetin järjestelmä toimii oikein. Ulkopuoliset asiantuntijat voivat tarjota vahvaa teollisuuden alan osaamista, teknistä tietoa, asiakaspalvelua ja apua moniin ongelmiin. Yritysten kannattaa myös hyödyntää tietoverkkokeskeisiä toimintoja. Tällöin informaatio jaetaan käyttäjille, jotka ovat ilmaisseet tarvitsevansa sitä. Kokonaisuudessaan asioiden internetistä tulee olemaan entistä parempaa hyötyä yrityksille silloin kun yrityksillä on tarvittavaa tietoa ja taitoa sen käyttöön. [3]

3. DATA DISTRIBUTION SERVICE

Data Distribution Service (DDS) on ensimmäinen kansainvälinen välikerroksen standardi, joka määrittää julkaisija-tilaaja-mallin kommunikaation ja yhteydet reaaliaikaisille ja hajautetuille järjestelmille. DDS on teknologiaratkaisu, joka on kaikkialla läsnä oleva, yhteentoimiva, turvallinen, alustasta riippumaton sekä ajan ja tilan suhteen tehokas. Se on tarkoitettu datan jakamiseen internetiin yhdistettyjen laitteiden välillä. [10]

3.1 Historia

OMG eli Object Management Group on kansainvälinen tietokonealan standardien yhtymä, joka on perustettu vuonna 1989. Se kehittää ja ylläpitää tietotekniikka-alan standardeja. Yksi yleisesti käytettyjä OMG:n määrittämiä ratkaisuja on UML (Unified Modelling Language) mallinnuskieli. OMG:n kehittämät standardit ovat moneen sovellukseen sopivia, toiselle alustalle siirrettäviä ja uudelleenkäytettäviä. Standardeja kehitetään moniin erilaisiin hajautettuihin ja vaihteleviin ympäristöihin ja ne ovat pääasiassa välikerroksen standardeja. [11]

Object Management Groupin kehittämä DDS-standardi on esitelty vuonna 2004. Se kehitettiin alun perin helpottamaan datan hajauttamisesta aiheutuvia haasteita. Haasteita esiintyi esimerkiksi kaupankäynnissä ja lentoliikenteessä ja muissa vastaavissa kohteissa, joissa käsitellään suurta määrää dataa. DDS:ää tarvittiin myös yksinkertaistamaan monimutkaisia järjestelmiä. [12]

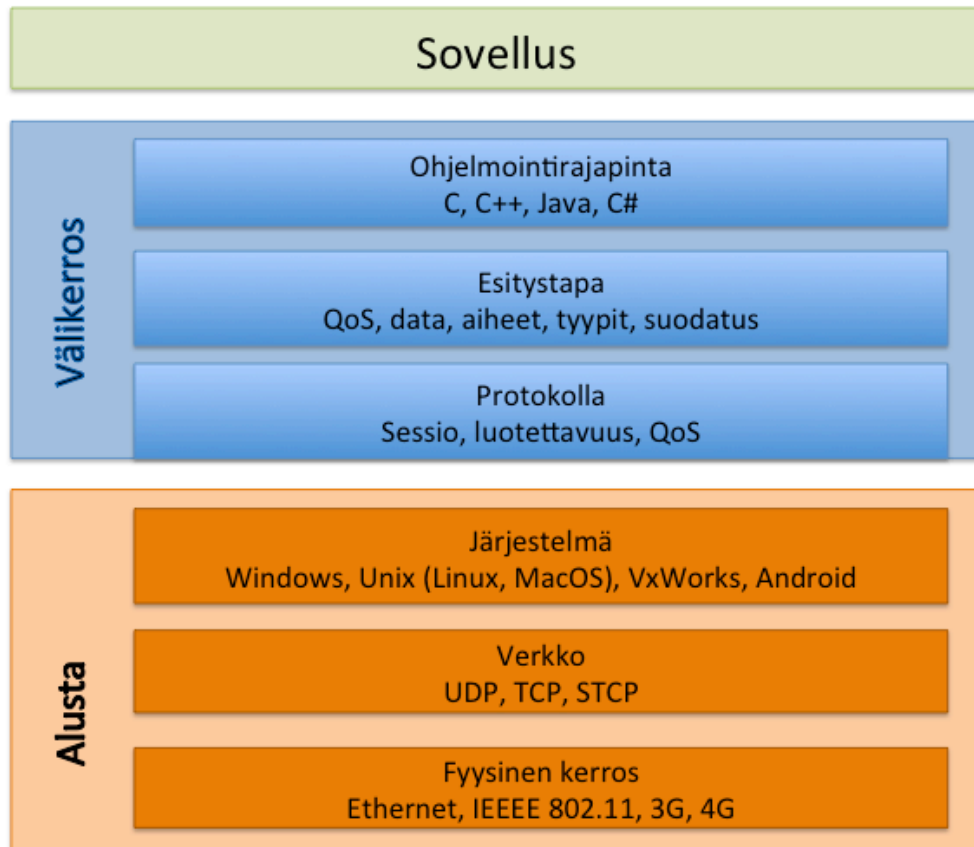
3.2 Määritelmä

Data Distribution Service eli DDS on datan hajauttamiseen sovellettava standardi, joka mahdollistaa vuorovaikutuksen ja kommunikoinnin fyysisen tason ja sovellustason välillä kuten kuvassa 8 on esitetty. DDS integroi systeemin komponentit yhteen ja tarjoaa sujuvaa datan siirtoa ja luotettavuutta. Se tarjoaa myös arkkitehtuurin liiketoimintaa tai esimerkiksi asioiden internetiä varten. [12]

Suurimmat vaatimukset DDS:lle sen kehitysvaiheessa oli luotettava suorituskyky kaikissa kohteissa sulautetuista käyttökohteista laajan skaalan käyttökohteisiin. DDS on omaksuttu käyttöön laajalti, esimerkiksi Scadan ja Telemetryn puolesta. Se mahdollista sovellusten välisen autonomisen, nimettömän ja tehokkaan datan jakamisen. [10]

Kuvassa 8 on esitetty systeemin yleinen rakenne. Ylimpänä on sovellustaso ja alimpana on alusta. Välikerros, esimerkiksi DDS, on nimensä mukaisesti näiden kahden kerrok-

sen välissä. Siihen kuuluvat ohjelmointirajapinta, esitystapa ja protokolla. Alustaan puolestaan kuuluvat itse järjestelmä, verkko sekä fyysinen kerros eli koko fyysinen laitteisto (hardware).



Kuva 8 Sovellus, välikerros ja alusta [12]

Hajautetussa järjestelmässä välikerrosohjelmisto on siis kerros operoivan järjestelmän ja sovelluskerroksen välissä. Välikerros mahdollistaa järjestelmän komponenttien välisen helpon kommunikoinnin ja datan jakamisen. Se yksinkertaistaa hajautettujen järjestelmien kehitystä antaen tietokoneohjelmistojen kehittäjien keskittyä vain sovellustensa kehittämiseen sen sijaan, että kehittäjät keskittyisivät siihen miten tieto siirtyy sovellusten ja alustojen välillä. Tällöin ohjelmistokehittäjän ei tarvitse puuttua tiedon siirron ominaisuuksiin, vaan kehittäjä voi rakentaa haluamansa ohjelman, jossa DDS vastaa tiedonsiirrosta. Tämä parantaa ohjelmoijien työn kustannustehokkuutta keskittäen heidän työnsä vain ohjelmistonkehitykseen. [12]

DDS-välikerros on ohjelmistokerros, joka erottaa sovelluksen operoitavasta järjestelmästä, verkkoliikenteestä ja alemman tason datan tallennusmuodoista. Sama konsepti on tarjolla eri ohjelmointikielille sallien sovellusten vaihtaa informaatiota kielestä ja operoivasta järjestelmästä riippumatta. Alemman tason yksityiskohdat kuten datan siir-

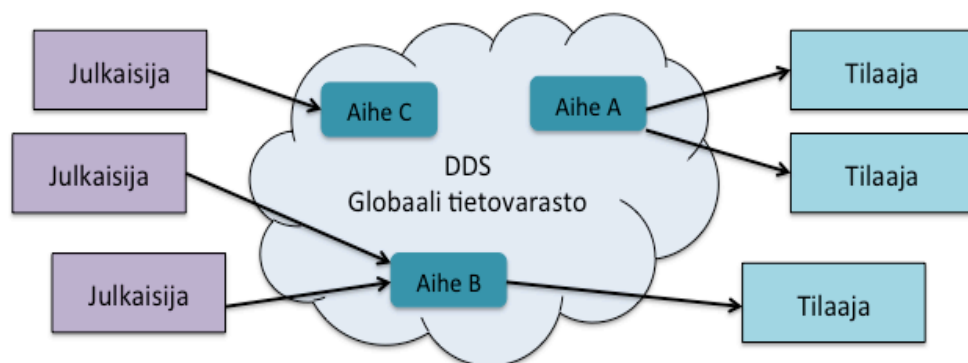
ron protokollat, yhteydet, luotettavuus, muut protokollat ja QoS ovat välikerroksen hallinnassa.

Välikerrosohjelmiston etuja ovat edustava ja korkean tason datan jakamisjärjestelmä, monikielisyys ja toimivuus eri alustoilla kuten Java, Scala, C, C++, Android, Windows, Linux ja Javascript. DDS toimii tehokkaasti pienillä kaistanleveyksillä ja siinä on minimaaliset viiveet. [10]

3.3 Globaali tietovarasto

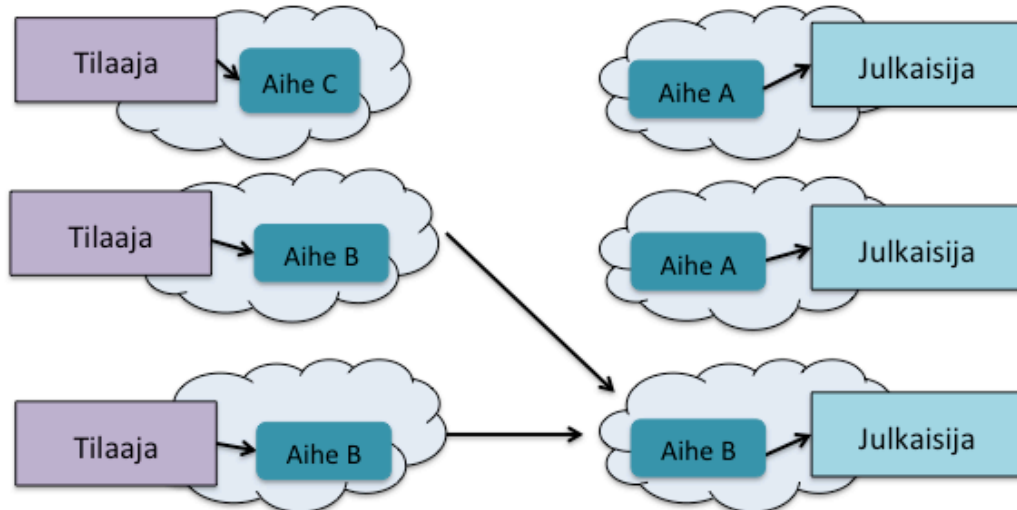
DDS:n yhteydessä käytetään termiä globaali tietovarasto (Global Data Space). Globaalia tietovarastoa kuvaillaan hajautetuksi, tehokkaaksi ja skaalautuvaksi. Se sallii sovellusten jakaa dataa anonymisti, turvallisesti ja tehokkaasti. Globaali tietovarasto on tärkeä osa DDS:n toimintaa. [10]

Globaali tietovarasto ulottuu kaikkiin sovelluksiin ja toimii kuten pilvipalvelu. DDS:n toimialue pitää sisällään globaalin tietovaraston. Globaalissa tietovarastossa oliot on merkitty aiheella (Topic) ja avaimella (Key). Globaalissa tietovarastossa julkaisijat ovat erillään tilaajista kuten kuvassa 9 on esitetty. Globaali tietovarasto on myös hajautettu. Hajautettu tietovarasto on tärkeä osa systeemiä, jotta järjestelmä toimii virheettää ja pulonkautoja ei synny. Tilaajat ja julkaisijat voivat liittyä tietovarastoon tai poistua siitä milloin tahansa. [13]



Kuva 9 Globaali tietovarasto [12]

Kun DDS on toiminnassa, tilaajat ja julkaisijat näkevät paikallisen varaston, jossa on dataa. Varaston nimi on globaali tietovarasto. Todellisuudessa DDS lähettää viestejä päivittääkseen sopivia tietovarastoja solmuissa. Tämä siis vain näyttää paikalliselta varastolta. Nämä paikalliset varastot synnyttävät illuusion siitä, että sovelluksilla on ulottuvuus kaikkeen dataan globaalissa tietovarastossa. Jokainen sovellus varastoi paikallisesti (kuva 10) tarvitsemansa vain niin kauan kuin sille on tarve.



Kuva 10 Globaalin tietovaraston oikea toteutus [10]

Globaali tietovarasto on käsite, joka on vain kokoelma paikallisia varastoja. Jokainen sovellus, kirjoitettuna millä ohjelmointikielellä tahansa, näkee paikallisen muistin sen alkuperäisessä tallennusmuodossa. Globaalia tietovarastoa voi lukea mistä tahansa järjestelmästä millä tahansa ohjelmointikielellä. Kaiken lisäksi globaali tietovarasto toimii todella pienellä vasteella. DDS:n toimialueen sisällä informaatiota jakavia yksiköitä ovat dataoliot aiheiden sisällä. Aihe on määritelty sen nimen perusteella ja dataolio joidenkin tiettyjen piirteiden perusteella. [12]

3.4 Quality of Service

DDS:n yhteydessä käytettäviä käsitteitä ovat siis toimialue (Domain), kirjoittaja (Data Writer), julkaisija (Publisher), lukija (Data Reader), tilaaja (Subscriber) ja aihe (Topic). Julkaisija on vastuussa julkaisuiden levittämisestä ja omistaa kirjoittajan. Kirjoittaja puolestaan yhdistetään vain yhteen julkaisijaan ja yhteen aiheeseen. Julkaisija puolestaan voi kuulua vain yhteen toimialueeseen. Tilaaaja on vastuussa datasta, jota se vastaanottaa tilauksilta. Tilaaaja myös omistaa lukijan. Palvelun laadusta vastaa Quality of Service (QoS), joka toimii lukijan, julkaisijan ja aiheen kanssa. Kuten julkaisija, myös tilaaaja voi kuulua vain yhteen toimialueeseen.

Palvelun laadun mekanismeja (QoS) käytetään ohjaamaan DDS:n käsitteiden ominaisuuksia kuten tärkeys, kesto ja saatavuus. Jokainen julkaisija-tilaaja -pari sopii itsenäisesti palvelun laadusta. Silti DDS-mallit voivat tukea äärimmäisen monimutkaisia ja joustavia datan siirto vaatimuksia. [14]

QoS on toisaalta myös mekanismi, joka mahdollistaa tietoliikenteen resurssien kontrolloinnin. Dataa voidaan jakaa myös siten että siihen sisältyy QoS-vaatimuksia. Vaatimuksia ovat esimerkiksi luotettavuus, systeemin toimintanopeus ja turvallisuus. Käytös-

sä olevissa järjestelmissä, kaikki päätepisteet eivät tarvitse kaikkea dataa kaikista paikallisista varastoista. Tällöin DDS lähettää vain datan mitä tarvitaan. Jos viestit eivät saavuta oikeaa määränpäättä, järjestelmä tuo luotettavuutta sinne missä ongelma sijaitsee. Kun järjestelmä muuttuu, DDS osaa selvittää minne lähetetään mitään dataa sekä osaa välittää tiedon muutoksista sovelluksille. Välikerrosohjelmisto osaa myös suodattaa dataa. Esimerkiksi jos datan määrä on suuri, DDS lähettää jokaiselle päätepisteelle vain dataa, jota ne todella tarvitsevat. Myöskään nopeat päivitykset eivät tuota ongelmia. DDS selviää niistä lähettämällä monilähetysviestejä päivittääkseen useita sovelluksia yhtä aikaa. DDS osaa myös pitää kirjaa uusista tiedostomuodoista, jotka tulevat käyttöön järjestelmässä. Kun uusia tiedostomuotoja kehitty, DDS kääntää ne automaattisesti. Myös sellaisissa sovelluksissa, joissa turvallisuus on tärkeää, DDS on mukana säatelemässä niihin pääsyä. DDS:n ominaisuudet tulevat siis parhaiten esille, kun asioita tehdään yhtä aikaa erittäin suurilla nopeuksilla vaativissa ja arvaamattomissa ympäristöissä. [12]

Suurin ero DDS:n ja vastaavien välikerrosohjelmistojen välillä on siis QoS. Tämän palvelun laatua ohjaavan mekanismin 21 parametria mahdollistavat dynaamisen ja muunneltavan reaaliaikaisen verkon. Kuitenkin huonona puolena on asetusten mahdollinen sekoittuminen. Aina ei ole selvää millä asetuksilla saavutetaan paras lopputulos. Usein DDS:n toteutuksissa on oletuksena jotkin tietyt asetukset, joten tutkimustyötä asetusten parissa riittää. [15]

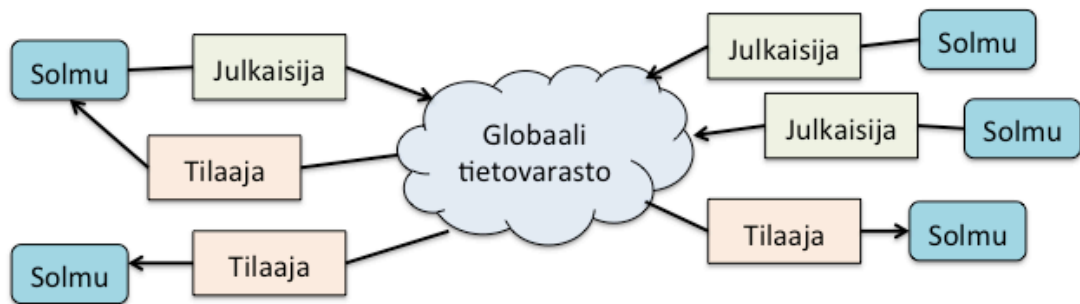
3.5 Data Distribution For Real-time Systems

Data Distribution For Real-time Systems on OMG:n standardi, joka määrittelee kaksi rajapintaa, jotka ovat eri tasoisia. Data-Centric Publish-Subscribe (DCPS) on alemman tason ohjelmointirajapinta, jonka tarkoituksena on toimittaa tehokkaasti oikea informaatio oikeille vastaanottajille. Data Local Reconstruction Layer (DLRL) puolestaan on ylemmän tason ohjelmointirajapinta. DLRL toisin kuin DCPS ei ole pakollinen DDS:n toteutuksessa, mutta se mahdollistaa yksinkertaisen integraation palvelusta sovelluserokselle. [5]

3.5.1 Data-Centric Publish-Subscribe

Data-Centric Publish-Subscribe eli DCPS-rajapinnan tiedonsiirto perustuu julkaisija-tilaaja-malliin (Publish-Subscribe Model). Julkaisija-tilaaja –malli koostuu nimensä mukaisesti julkaisijoista ja tilaajista. Kuuntelijat (Listeners) tilaavat dataa, jota ne lähettävät eteenpäin kaikille rekisteröityneille vastaanottajille. Nämä voivat olla olioita samassa välityspalvelimessa, vaikka vastaanottajat ovat eri komponenteissa tai solmuissa kuten kuvassa 10 on esitetty. Lähettäjät (Senders) vastaanottavat dataa sovelluksen komponenteilta tai välikerrokselta ja lähettävät sitä eteenpäin yhteydessä oleville kuuntelijoille. Tärkein syy lähettäjien ja kuuntelijoiden käyttöön on kytkeä kommunikoivat

komponentit siten että tekniset ongelmat, jotka liittyvät datan hajauttamiseen ovat yhdessä paikassa, jolloin asetusten muuttaminen on helpompaa. [14]

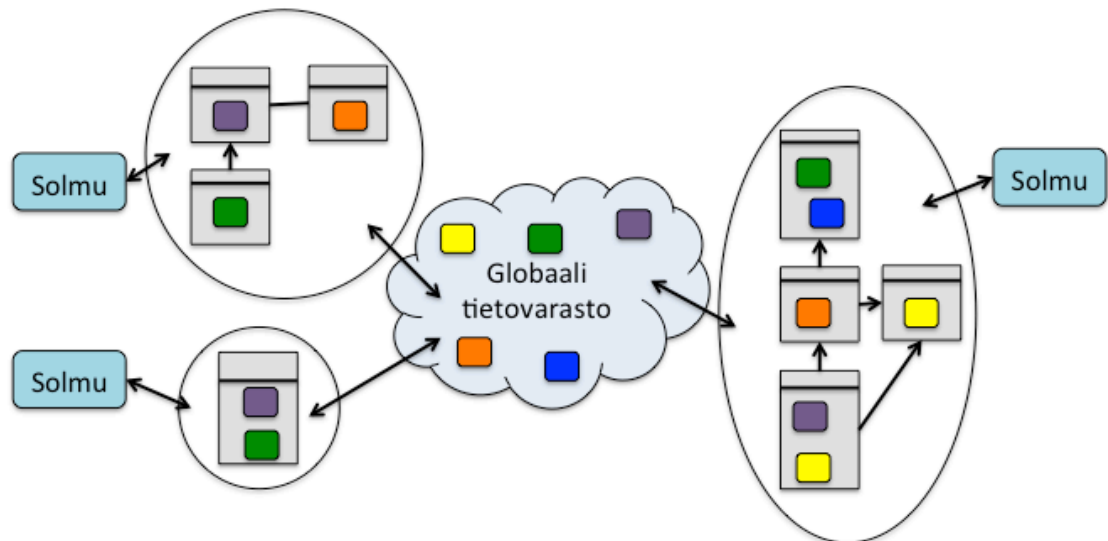


Kuva 11 DCPS [14]

DCPS-mallista on tullut suosittu reaaliaikaisissa sovelluksissa. Suosio oli mahdollista saavuttaa, koska ennen malli jota käytettiin datan hajauttamiseen perustui hajautettuun jaettuun muistiin. Tämän kaltainen ratkaisu ei ole joustava eikä skaalautuva ja se on vaikea saada toimimaan hyvin internetin ylitse. DCPS puolestaan on ratkaisu, joka perustuu globaalin tietovarastoon. Kuten aiemmin on mainittu, globaali tietovarasto on kaikkien sovellusten saavutettavissa. Sovellukset, jotka haluavat olla osa tietovaraston dataa, ilmoittavat haluavansa olla julkaisijoita. Tilaajiksi puolestaan ilmoittautuvat sellaiset sovellukset, jotka haluavat päästä käsiksi tietovaraston dataan. Kun julkaisijat julkaisevat uutta dataa tietovarastossa, välikerrosohjelmisto levittää tiedon kaikille kiinnostuneille tilaajille. [16]

3.5.2 Data Local Reconstruction Layer

DLRL-rajapinta eli Data Local Reconstruction Layer mahdollistaa sovelluksen saavuttaa datan aivan kuin data olisi paikallista (kuva 11). Tässä tapauksessa välikerrosohjelmisto päivittää paikallisen kopion datasta sen levittämisen lisäksi. DLRL siis uudelleen rakentaa dataa paikallisesti. [16]



Kuva 12 DLRL

DLRL rajapinnan tarkoitus on tarjota suora pääsy jaetun datan luo. Se on suunniteltu siten että sovelluksen kehittäjän sallitaan käyttää DLRL:n alla olevia DDS:n toimintoja. Tämä ei kuitenkaan aina onnistu, koska DLRL rajapinnan päämääränä on saumattomasti integroitua sovellukseen. Tällöin joitakin DDS:n toimintoja ei voi käyttää DLRL:n kautta, vaan ne on saavutettavissa vain DDS:n kautta. Kuitenkin DLRL auttaa sovelluksen suunnittelijaa monin tavoin. Se esimerkiksi mahdollistaa kohteiden luokittelun muun muassa datakentän perusteella. Datakentät voidaan sitten liittää DCPS:n entiteetteihin. Näitä entiteettejä voidaan lukea, poistaa, kirjoittaa tai luoda. [17]

On olemassa kaupallisia tuotteita, jotka toteuttavat DCPS:n kokonaisuudessaan ja DLRL:n vain osin. Kuitenkin nämä tuotteet ovat yksityisomistuksessa eivätkä sisällä standardoitua käyttöliittymää. [16]

3.6 The Real-time Publish-Subscribe Wire Protocol

Real-Time Publish-Subscribe Wire Protocol (RTPS) on alemman tason protokolla, jota käytetään DDS:n eri toteutusten välisen kommunikoinnin mahdollistamiseen. Tämä DDS:n eritelmä määrittelee siis sovelluskerroksen rajapinnan ja DDS:n käyttäytymisen, jotka tukevat DCPS protokollaa reaaliaikajärjestelmissä. Koska DDS:n toteutukset voivat olla eri toimittajilta, niiden välille on taattava yhteensopivuus. RTPS-protokollalla on juuret teollisuusautomaatiossa ja se on otettu käyttöön laajalti erilaisissa teollisuuden laitteissa. [18]

RTPS kehitettiin erityisesti tukemaan datan hajautuksen tuomia vaatimuksia. Teollisuusautomaatioyhteisö määritteli vaatimukset publish-subscribe wire –protokollalle. Nämä vaatimukset ovat lähellä DDS:n vaatimuksia. Tuloksena on tiivis yhteys DDS:n ja RTPS wire –protokollan välillä. [18]

RTPS protokolla on suunniteltu käytettäväksi internet protokollan (IP) kanssa. RTPS on suunniteltu siten, että se voi toimia monilähetyksen ja esimerkiksi UDP/IP-protokollan yli. Sen tärkeimpiä ominaisuuksia ovat suorituskyky ja QoS ominaisuudet, jotka mahdollistavat luotettavan julkaisija-tilaaja yhteyden reaaliaikaisille sovelluksille käyttäen IP standardia. Viansietokyky sallii verkkoyhteyksien luonnin ilman virheitä. RTPS on helppokäyttöinen ja uudet sovellukset ja palvelut voidaan yhdistää tai irrottaa verkosta milloin tahansa. Muita tärkeitä ominaisuuksia ovat joustavuus, muunneltavuus, modulaarisuus ja skaalautuvuus. Joustavuus sallii protokollalle laajennuksia ja parannuksia uusien sovellusten muodossa ilman että yhteensopivuus kärsii. Muunneltavuus sallii luotettavuuden ja ajantasaisuuden vaatimusten tasapainottamisen jokaiselle datan toimitukselle. Modulaarisuus mahdollistaa yksinkertaisten laitteiden toteuttaa protokollan osajoukon ja silti saada yhteys verkkoon. Skaalautuvuus puolestaan mahdollistaa systemin toimivuuden hyvin laajoissa julkaisija-tilaaja –verkoissa. [18]

Platform Independent Model (PIM) on malli, joka mahdollistaa RTPS-protokollan määrittämisen ilman tietoliikenneprotokollien rajoitteita. Nimensä mukaisesti, PIM on standardi, joka ei ole toteutuskielestä riippuvainen. Platform Specific Model (PMS) puolestaan määrittelee tarkan esityksen biteistä, tavuista tai mistä tahansa informaatiosta, joka on alustalle ominainen. PMS on rakennettu PIM:n päälle ja sen avulla liitetään PIM UDP/IP protokollaan. [18]

3.7 Datakeskisyys vs serveripohjaiset ratkaisut

Välikerrosohjelmistoja on olemassa viestikeskeisiä ja datakeskeisiä. Yksinkertaiset ratkaisut ovat usein viestikeskeisiä. Useimmat välikerrosohjelmistot lähettävät tietoa sovellusten ja järjestelmien välillä. DDS on datakeskeinen ratkaisu, joka mahdollistaa kehittyneen viestien sisällön. Viestit sisältävät tietoa, jota sovellus tarvitsee ymmärtääkseen dataa, jota se vastaanottaa. Datakeskisyydelle tyypillistä on, että DDS tietää mitä dataa se varastoi ja hallinnoi datan jakamista. Ohjelmoijien, jotka käyttävät perinteistä viestikeskeistä välikerrosohjelmistoa, täytyy kirjoittaa ohjelma, joka lähettää viestejä. Ohjelmoijien, jotka käyttävät datakeskeistä välikerrosohjelmistoa, täytyy kirjoittaa ohjelmisto, joka erittelee miten ja milloin dataa jaetaan ja sitten jakavat datan eteenpäin. DDS tekee tämän kaiken ilman, että ohjelmoijan tarvitsee kirjoittaa monimutkaista ohjelmaa sovellukselle. DDS siis tarjoaa kontrolloitua, hallittua ja turvallista dataa käyttäjilleen. [12]

Hajautetut järjestelmät toimivat siis lähettämällä tietoa itsenäiseltä sovellukselta toiselle. Perinteisesti tämä tiedonsiirto on tapahtunut viestikeskeisesti. Viestikeskeinen tiedonsiirto voi olla yksinkertaista tai monimutkaista. Yksinkertaisimmillaan tiedonsiirto on suoraa sovellukselta sovellukselle ja monimutkaisimmillaan se koostuu viestijonoista ja eri transaktioista. Tämän kaltainen järjestelmä luulee, että viesti itsessään on tiedonsiirron tarkoitus eikä järjestelmä ole tietoinen viestin sisällöstä. Sovelluksen kehittäjät suunnittelevat siis ohjelmistoja, jotka lähettävät viestejä vastaanottajien välillä. Viesti-

keskeisessä välikerrosohjelmistossa lähetetään siis vain dataa ilman, että tiedetään ketä se kiinnostaa. [19]

Datakeskeinen infrastruktuuri puolestaan päivittää dataa, jonka sisällön se tuntee. Niimensä mukaisesti datakeskeisen järjestelmän tarkoituksena on datan sisältö. Datakeskeisen infrastruktuurin on määriteltävä data, jota se hallinnoi. Tällöin se voi muodostaa säännöt kuinka data on rakennettu ja miten se saavutetaan. Ohjelmistosuunnittelijat kehittävät siis sovelluksia, jotka lukevat ja päivittävät sisäänpääsyjä tietoalueisiin. Datakeskeisessä välikerrosohjelmistossa puolestaan tiedetään kuka on tilannut mitään dataa, jolloin data osataan lähettää oikeille vastaanottajille. [19]

Datakeskeinen ja viestikeskeinen eivät kuitenkaan ole toistensa täysiä vastakohtia. Jotkin järjestelmät jopa yhdistävät näiden kahden välikerrosohjelmistojen eri ominaisuuksia. Esimerkiksi viestikeskeiset järjestelmät voivat käyttää tietokantoja ja muita vastaavia datakeskeisille järjestelmille tyypillisiä ominaisuuksia. Toisaalta käytettävä välikerrosohjelmisto täytyy aina valita järjestelmän vaatimusten mukaan. [19]

Muita merkittäviä eroja datakeskeisen välikerrosohjelmiston ja muiden toteutusten välillä ovat palvelun laadun mekanismit (Quality of Service). Näitä mekanismeja ovat esimerkiksi dynaamisuus ja skaalautuvuus. [15] Datakeskisyys on kuitenkin melko ideaalinen ratkaisu, kun DDS:ää käytetään asioiden internetin yhteydessä. [12]

3.8 DDS – ei palvelinta

DDS toimii ilman erillisiä palvelimia. Tämä ominaisuus yhdessä joustavuuden ja skaalautuvuuden kanssa erottavat sen muista välikerrosohjelmistoista ja muista vastaavista ratkaisuista. DDS:n käyttäjät eivät sido itseään tiettyyn toimittajaan, vaan standardiin. Tällöin käyttäjät voivat vaihtaa ja muuttaa DDS:n toimittajia esimerkiksi kesken prosessin kehityksen. DDS on siis vertaisverkkomalli, joka ei vaadi yhdyskäytäviä, palvelimia tai taustalla jatkuvasti pyörivää palvelinohjelmistoa.

Muut välikerrosohjelmistot käyttävät palvelimia. Esimerkiksi TCP ja CORBA vaativat toimiakseen palvelimen. Niiden toiminta perustuu asiakas-palvelin –malliin, jolloin asiakkaat ja serverit tietävät toistensa olinpaikat voidakseen kommunikoida. Toimintaperiaate on siis erilainen verrattuna DDS:n julkaisija-tilaaja –malliin. DDS käsittelee yksityiskohtia kuten datan suodattamista, luotettavuutta, datahistorian tallennusta ja tiedon siirron nopeuden muutoksia. DDS voi keskittyä funktionaalisiin vaatimuksiin.

Välikerrosohjelmisto voi myös perustua julkaisija-tilaaja –malliin ja silti käyttää palvelimia toiminnassaan. Tällainen on JMS (Java Messaging Service). JMS käyttää palvelimia tiedonvälitykseen, joka on konfiguroitava käytettävien aiheiden kanssa. Jokainen sovellus, joka on mukana JMS:n tiedonvälityksessä, on yhteydessä JMS-palvelimeen, kun taas julkaisijat ja tilaajat ovat toisiinsa löyhästi kytkettyjä. JMS-palvelin käsittelee

julkaisijoiden ja tilaajien yhdistämiseen liittyvät yksityiskohdat. DDS ei vaadi tällaisia palvelimia tai taustalla pyörivää palvelinohjelmistoa, koska DDS hallitsee pääteipisteitään sovelluksien sisällä.

DDS:n käyttö poistaa siis palvelimien tarpeen. Palvelimet ovat mahdollisia vikaantumisen kohteita. Myös verkot ovat yksinkertaisempia kuin sellaiset verkot, joissa käytetään palvelimia. Jos verkon palvelin kaatuu, koko järjestelmästä tulee hyödytön. DDS on melkoisen luotettava juuri siksi, että palvelimia ei tarvita. Palvelin voi myös kuormittua, jolloin viive kasvaa. DDS:n toimintaa palvelin ei hidasta. [20]

3.9 Miksi IoT ja DDS?

Monet asioiden internetin sovellukset ovat luonteeltaan datakeskeisiä. Dataa kerätään, varastoidaan, analysoidaan ja jaetaan. Dataa kerätään eri laitteilta ja järjestelmiltä ja siirretään julkiseen tai yksityiseen pilvipalveluun. Pilvipalvelusta dataa voidaan analysoida, luokitella ja siitä voidaan tehdä erilaisia kuvaajia ja päätelmiä. [10]

Asioiden internetissä kommunikointi ja tiedonsiirto tapahtuu monin eri reitein. Tiedonsiirto voi olla koneiden välistä, koneen ja pilven välistä sekä eri pilvipalvelujen välillä tapahtuvaa datan siirtoa. Koneiden välinen tiedonsiirto voi muodostaa verkon, jossa jokainen verkkoon kytketty taho toimii palvelimena ja asiakkaana verkon muille tahoille. Puhutaan siis vertaisverkosta (peer to peer). Vertaisverkossa laitteistot voivat olla hyvin erilaisia ja verkkotyöskentelyn kyvykkyys eri tasoista. Joissakin käyttökohteissa, esimerkiksi pilvipalvelun sisällä, pieni viive ja korkea suoritusteho ovat oleellisia. Jotta koneiden välinen kommunikointi mahdollistaa asioiden internetin avoimuuden ja yhteentoimivuuden, koneiden välisen tiedonsiirron tulee perustua standardiprotokolliin. Koneen tai laitteen ja pilvipalvelun välisen tiedonsiirron luonne riippuu sovelluksen ominaisuuksista. Ominaisuuksia voivat olla satunnainen datan päivitys tai reaaliaikainen datan päivitys, rajattu kaistanleveys, katkonainen yhteys ja tietoturva. [10]

Asioiden internetin sovelluksille on saatavilla useita eri protokollia. Esimerkiksi Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP) ja Constrained Application Protocol (CoAP) ovat protokollia, joita voidaan käyttää asioiden internetin yhteydessä DDS:n tapaan. Kuitenkin edellä mainitut protokollat ovat yksinkertaisia ja käyttötapaukset ovat rajallisia. DDS puolestaan on ominaisuuksiltaan monipuolinen ja asioiden internetin tarjoaman monimutkaisen datan käsittely onnistuu siltä hyvin. [5]

Monet järjestelmät sisältävät laitteita, palvelimia ja internetiin yhdistettyjä laitteita. Näillä kaikilla on erilaiset tavat kommunikoida, joten on helpointa käyttää yhtä ajatusmallia niiden kaikkien kesken. Järjestelmien suunnittelijoiden tulisi päättää käytettävät protokollat, jotka kohtaavat halutut sovellukset. Tämän jälkeen laajennettaisiin valinta kaikkialle järjestelmään. Esimerkiksi mobiililaitteen datan käyttö ja jonkin toisen lait-

teen datan keräys ovat erilaisia käyttötapauksia. Siis vaatimukset toiminnoille ovat erilaisia. Kaiken kaikkiaan DDS on monikäyttöinen verrattuna muihin vastaaviin protokolleihin. Se voi hallita datan keruuta yksinkertaisilta laitteilta sekä yhdistää suuria ja tehokkaita laitekokonaisuuksia verkkoon ja lukea dataa pilvipalvelusta. DDS:n kommunikointi perustuu vertaisverkkomalliin kuten jo aikaisemmin mainittiin. Tällöin voidaan minimoida palvelimien ja viestien häiriöiden lukumäärää, jolloin toimintavarmuus paranee. DDS:n sanotaan olevan ideaalinen asioiden internetin sovelluksiin, jotka vaativat kestäväää ja luotettavaa arkkitehtuuria. [5]

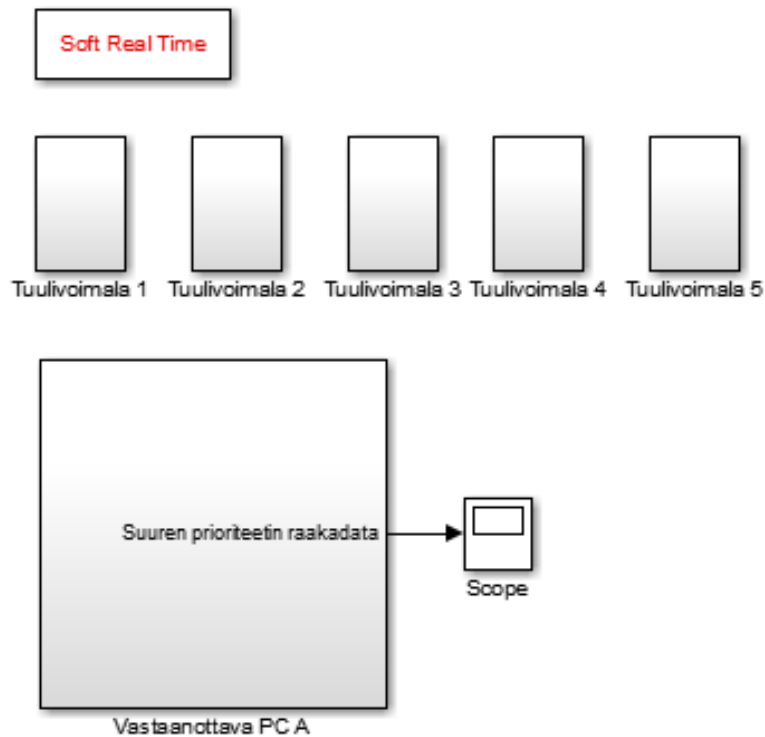
4. DDS-JÄRJESTELMÄ

Tässä luvussa esitellään järjestelmä, jonka tarkoituksena on simuloida miten Data Distribution Servicen lähetys, vastaanotto ja kuuntelu toimivat. Simulointimalli rakennettiin käyttäen Mathworksin Matlabia, Simulinkia, DDS-toolboxia sekä RTI connext DDS -ohjelmistoa. Aluksi kappaleessa 4.1 esitellään mallin rakenne ja toiminta, kappaleessa 4.2 tarkastellaan järjestelmän välittämää dataa ja lopuksi esitellään mallin laskentaominaisuuksia sekä tehdyt päätelmät.

4.1 Simulointimalli

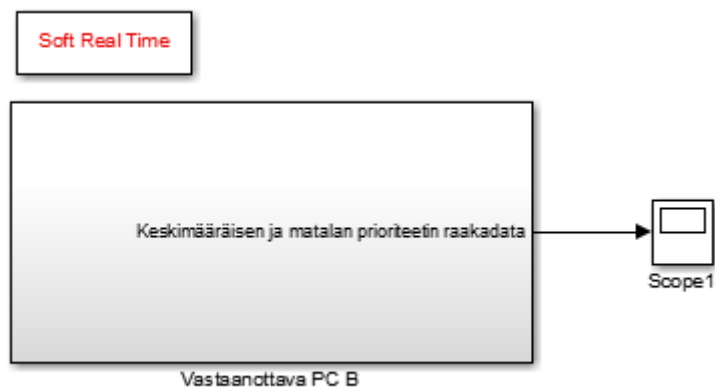
Matlabin Simulink-toolboxia käyttäen rakennettiin Data Distribution Serviceä hyödyntävä malli. Mallissa on viisi tuulivoimalaa, jotka lähettävät dataa (kuva 13) sekä kaksi tietokonetta, PC A ja PC B. Tuulivoimalat edustavat globaalia tietovarastoa ja sijaitsevat PC A:n kanssa samassa mallissa. Tuulivoimalat sijaitsevat PC:llä A, vaikka ne voisivat sijaita omalla tietokoneellaan tai kokonaan erillisessä Matlab-mallissa.

Soft real time –lohkon avulla Simulink-mallia on mahdollista ajaa lähes reaaliajassa. Jos mallissa ei olisi soft real time –lohkoa, Matlab ajaisi Simulink-mallin niin nopeasti kuin tietokoneen suorituskyvyn puitteissa pystyisi. PC A ja PC B kuitenkin kommunikoivat keskenään, joten mallit on saatava toimimaan yhtä aikaa. Soft real time –lohko sisältää funktion, jonka ajaminen kestää noin yhden sekunnin ajan, jolloin Simulink-mallin laskentasyklin ajaminen kestää yhden sekunnin. Tällöin esimerkkijärjestelmä suoritetaan lähes reaaliajassa ja mallien välinen kommunikointi on mahdollista.



Kuva 13 PC A

Vastaanottavia tietokoneita mallissa on siis kaksi, PC A ja PC B. PC B (kuva 14) sijaitsee eri Matlab-mallissa kuin PC A. Näitä tietokoneille voitaisiin ajaa myös fyysisesti eri tietokoneilla, vaikka tässä demonstraatiossa molempia malleja ajetaan samalla tietokoneella. Myös PC B:n kanssa samassa mallissa on soft real time –lohko, joka mahdollistaa mallien yhtäaikaisen toiminnan.

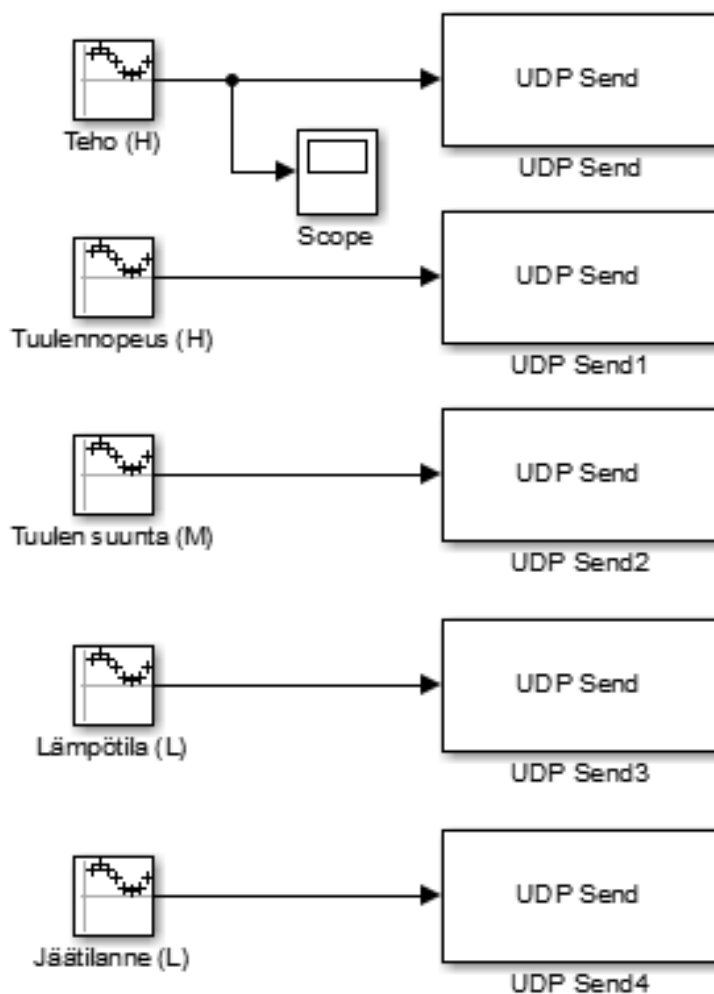


Kuva 14 PC B

PC A -mallissa sijaitsevia tuulivoimaloita on viisi. Yhdessä tuulivoimalaa simuloivassa mallissa (kuva 15) on viisi kuvitteellista anturia, jotka mittaavat dataa. Tässä järjestelmässä anturien lähettämää dataa simuloidaan siniaalloilla. Signaalit simuloivat tuulivoimalan tuulivoimalan tehon tuottoa, tuulen nopeutta, tuulen suuntaa, lämpötilaa sekä

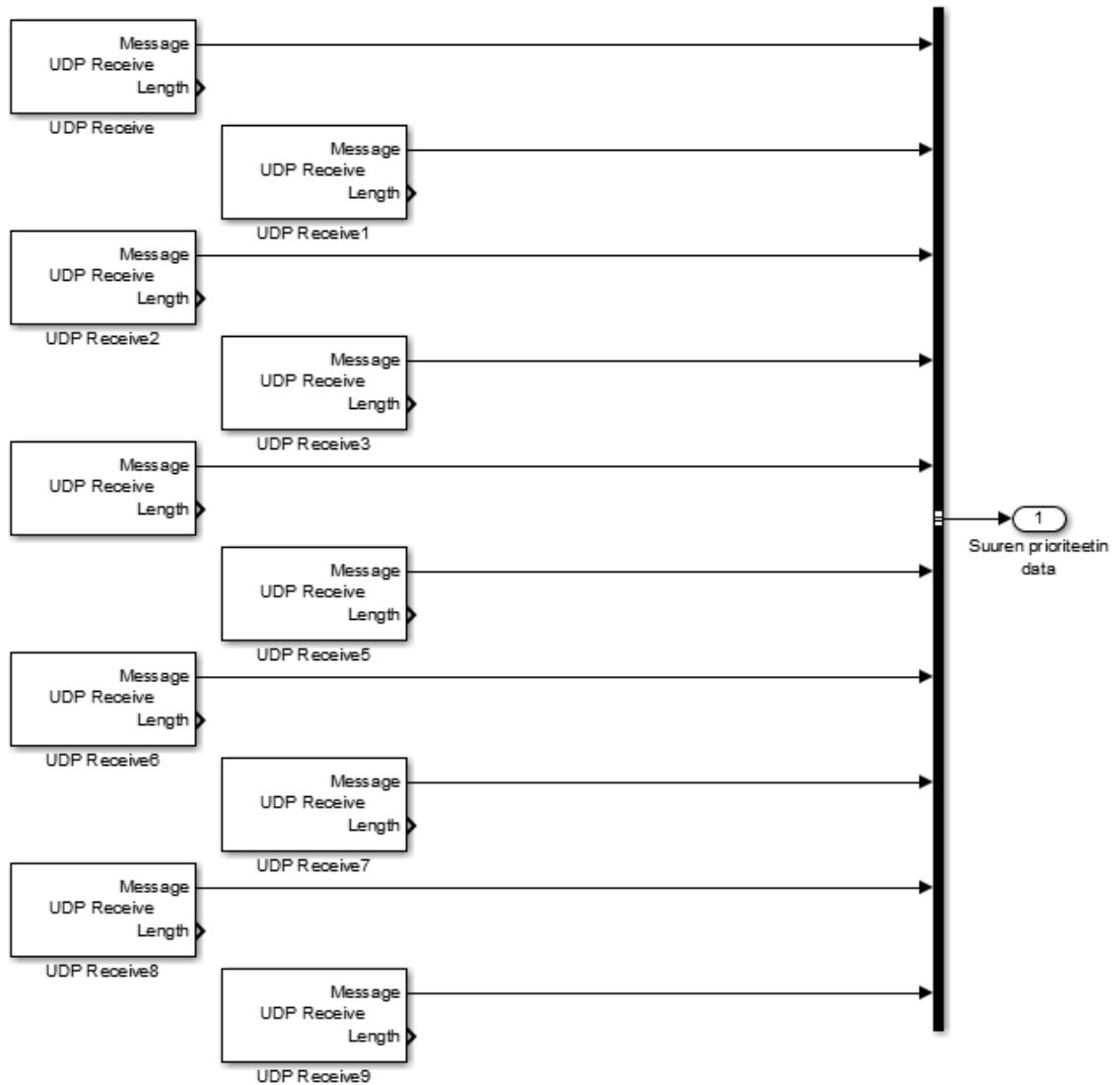
antavat jäähälytyksen tarvittaessa. Antureihin on merkitty onko niiden lähettämä data suuren prioriteetin (H) dataa, keskimääräisen prioriteetin (M) dataa vai matalan prioriteetin (L) dataa. Teho sekä tuulennopeus ovat suuren prioriteetin dataa, tuulen suunta on keskimääräisen prioriteetin dataa ja lämpötila sekä jäähälytys ovat matalan prioriteetin dataa.

Tuulivoimalat lähettävät dataa UDP:n (User Datagram Protocol) välityksellä. UDP on protokolla, jonka toimintaan ei kuulu tarkistaa menikö data perille eli se on yhteydetön protokolla. [19] Tuulivoimaloita on yhteensä viisi ja niissä kaikissa on samanlaiset viisi simuloitua anturia, jotka kaikki lähettävät hieman eri dataa kuin tuulivoimala 1. Jokaisen tuulivoimalan jokainen anturi lähettää dataa UDP:n välityksellä eteenpäin.



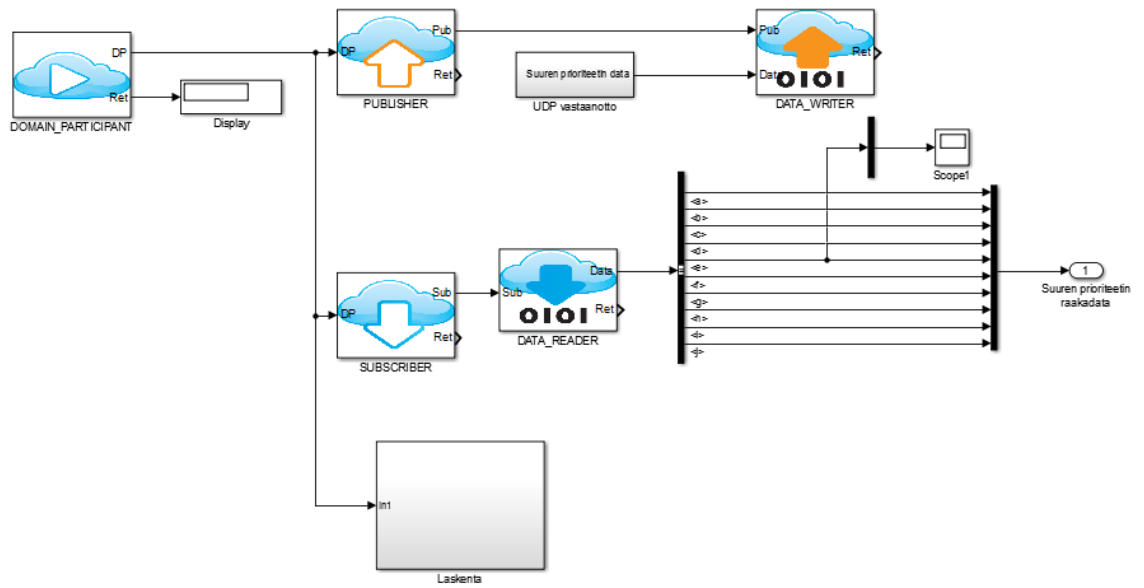
Kuva 15 Tuulivoimala 1

UDP send –lohkot siis lähettävät anturien dataa ja UDP receive –lohkot vastaanottavat sitä (kuva 12). PC:lle A lähetetään vain suuren prioriteetin dataa. Kuvassa 16 on siis kaksi signaalia jokaiselta tuulivoimalalta. Kaikkien voimaloiden tehot ja tuulen nopeudet lähetetään UDP:n välityksellä PC:lle A.



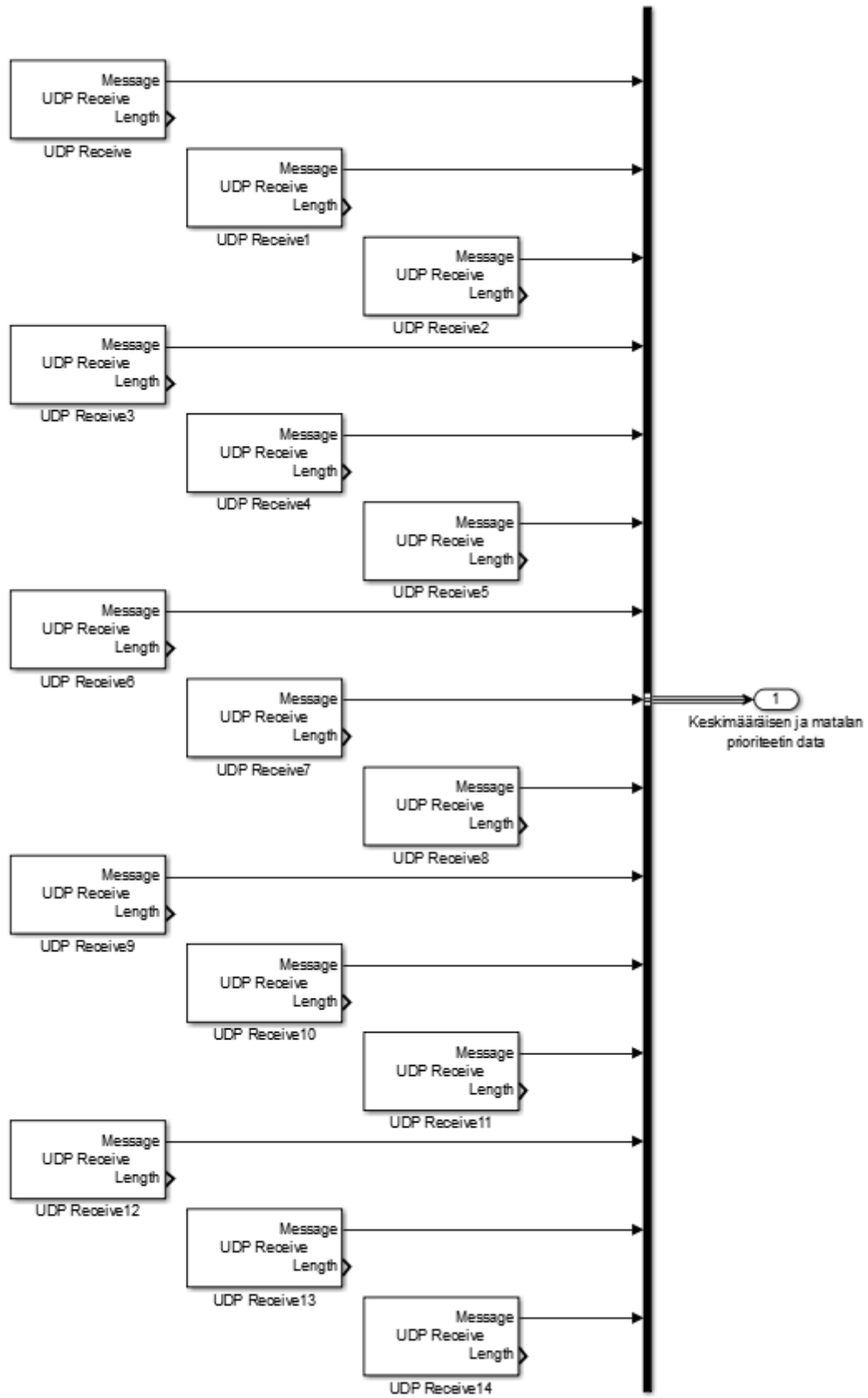
Kuva 16 Suuren prioriteetin data

Suuren prioriteetin data saapuu vastaanottavassa mallissa PC A sijaitsevalle datan kirjoittajalle eli DDS Data Writerille. Kirjoittaja lukee suuren prioriteetin datan ja kirjoittaa datan pilveen. Kun data on kirjoitettu pilveen, julkaisija kutsuu kirjoittajaa. Julkaisija lähettää tuulivoimaloiden dataa eteenpäin sitä haluaville. DDS Reader eli lukija puolestaan noutaa datan toimialueen tietovarastosta. Koko vastaanottava PC A on saman toimialueen eli domainin alla. PC A -mallissa on myös alimalli nimeltä laskenta. Se on esitelty myöhemmin kappaleessa 4.3. Kuvasta 17 nähdään myös miten kirjoittaja lähettää ulos samat kymmenen signaalia, jotka syötettiin datan kirjoittajalle.



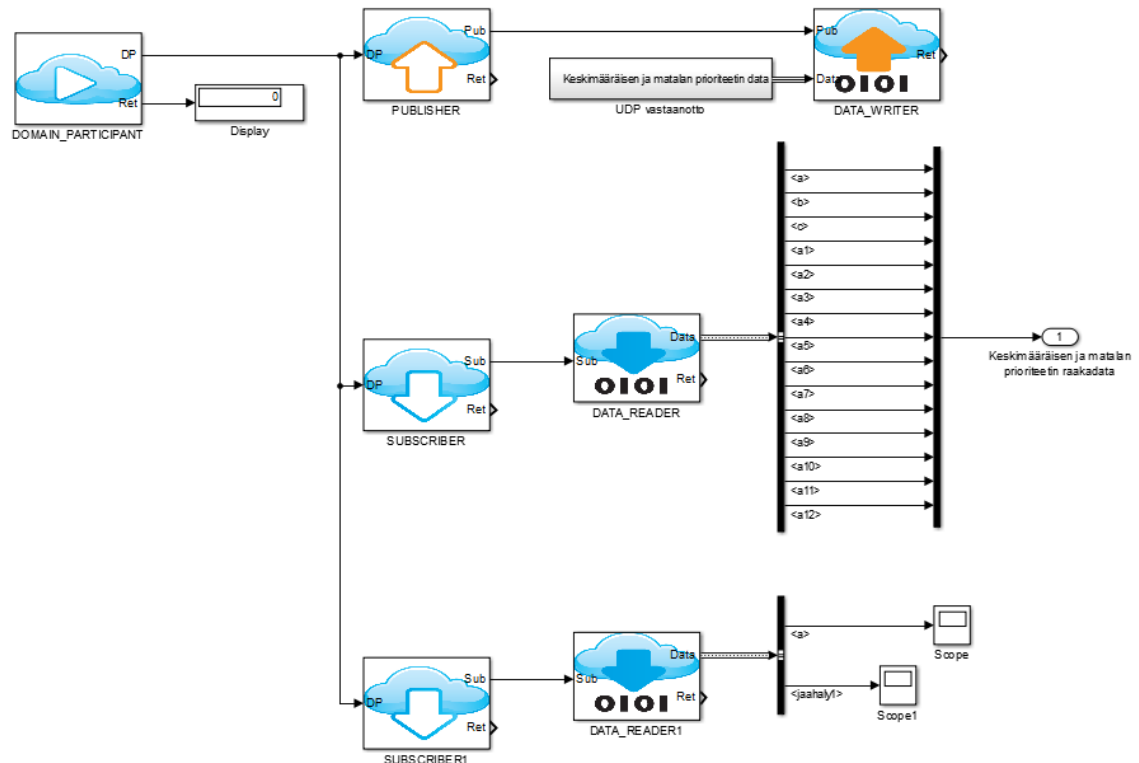
Kuva 17 Vastaanottava PC A

Kuvassa 18 ovat viisitoista keskimääräisen ja matalan prioriteetin datan signaalia. Jälleen kaikki viisi tuulivoimalaa lähettävät määrättyä dataa UDP:n välityksellä. Tällä kertaa ne tuovat dataa tuulen suunnasta ja lämpötilasta sekä tietoa jäätilanteesta. Vastaanottavalle vastaanottavalle mallille PC B lähetetään siis sekä keskimääräisen että matalan prioriteetin dataa.



Kuva 18 Keskimääräisen ja matalan prioriteetin raakadata

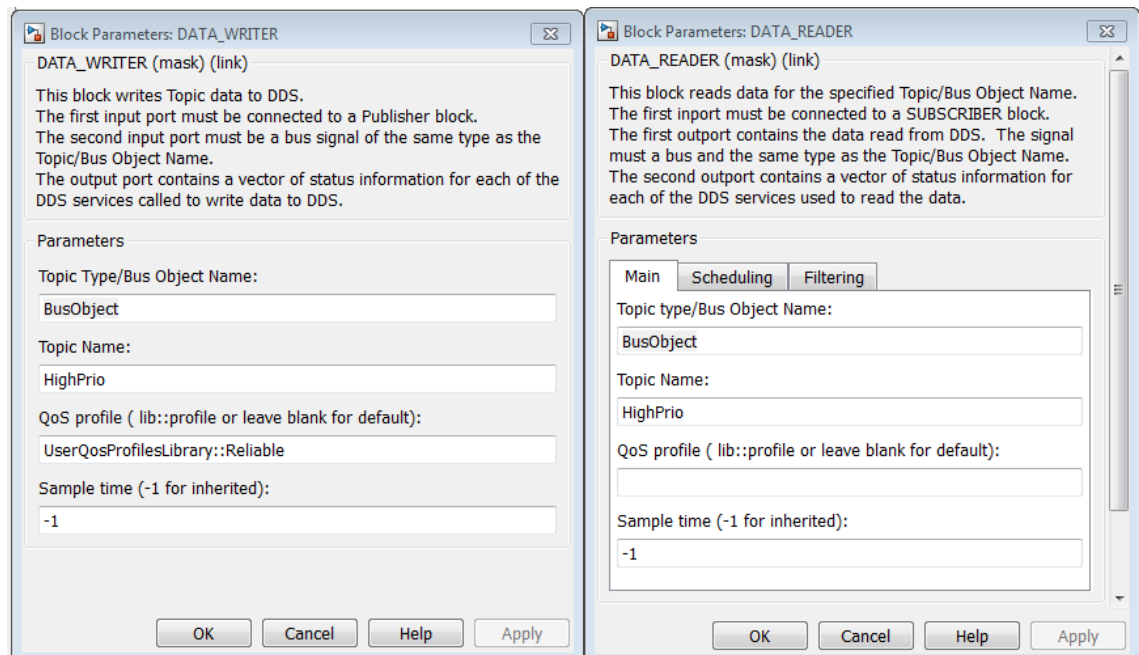
PC B (kuva 19) on rakenteeltaan vastaava kuin PC A. Suurin ero näiden kahden mallin välillä on tulevien signaalien lukumäärä. PC B:lle saapuvat sekä keskimääräisen prioriteetin että matalan prioriteetin datat. PC B:llä on myös PC A:lla tapahtuvan laskenta-alimallin välittämän datan vastaanotto. Sitä käsitellään myöhemmin luvussa 3.3.



Kuva 19 Vastaanottava PC B

Näiden kahden dataa vastaanottavan mallin tarkoituksena on näyttää miten DDS käytännössä toimii. Tämän vuoksi tuulivoimaloiden lähettämä data on jaettu siten että siitä osa menee PC:lle A ja osa PC:lle B. Tällöin dataa voidaan käsitellä ja sitä voidaan lähettää DDS:n välityksellä PC:ltä toiselle.

Kuva 20 on esimerkki datan kirjoittajan ja datan lukijan asetuksista. Asetuksissa on määritelty aihe eli topic sekä busobject, joka on Matlabin sisäinen väylä datan välitykseen. Molemmilla lohkoilla on samat asetukset, koska ne toimivat samalla mallin tasolla ja niiden käyttäjälle näkyvän toiminnan erona on vain datan välityssuunta.



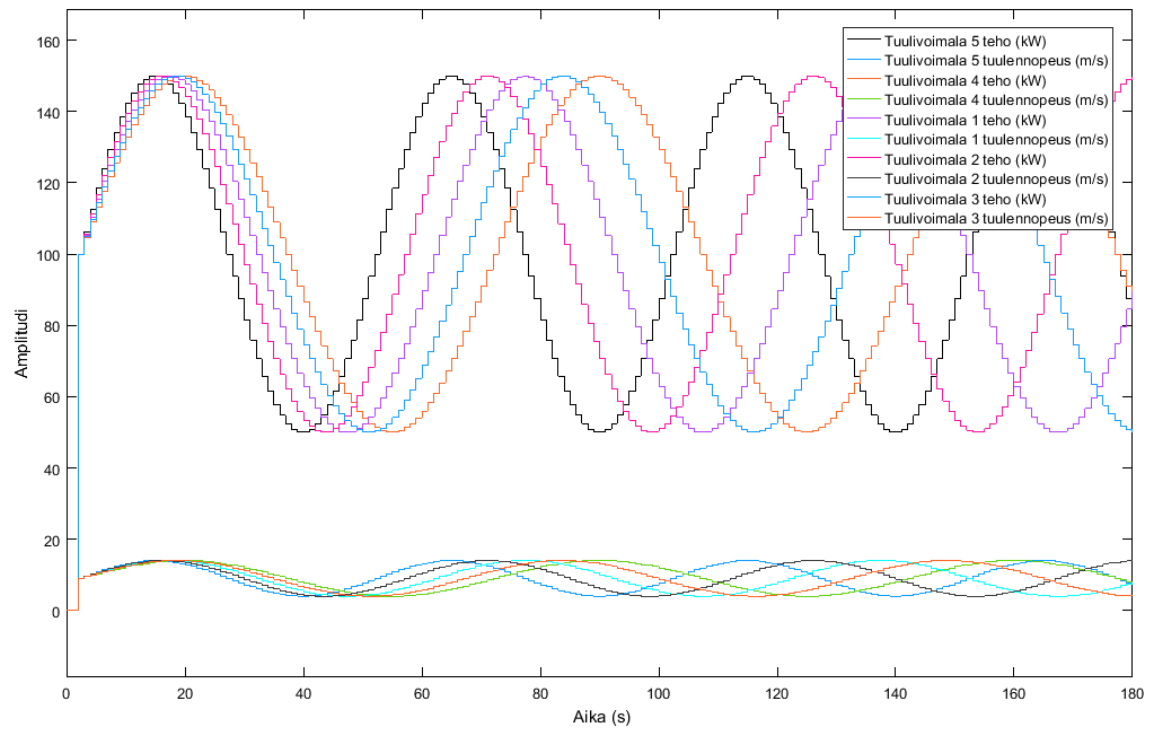
Kuva 20 Esimerkki datan kirjoittajan ja datan lukijan asetuksista

Vastaanottavan tietokoneen mallissa on myös julkaisija ja tilaaja, joille avautuu myös asetussivut. Lohkoille voi valita asetuksista toimivatko ne julkaisijana vai tilaajana. Niiden liittäminen tietoaalueeseen on määritelty graafisesti, joten lohkon asetuksista ei valita erillistä tietoaaluetta. Sama pätee myös datan kirjoittajan ja datan lukijaan, joiden liittäminen julkaisijaan ja tilaajaan määritellään Simulink-mallissa graafisesti.

4.2 Järjestelmän välittämä data

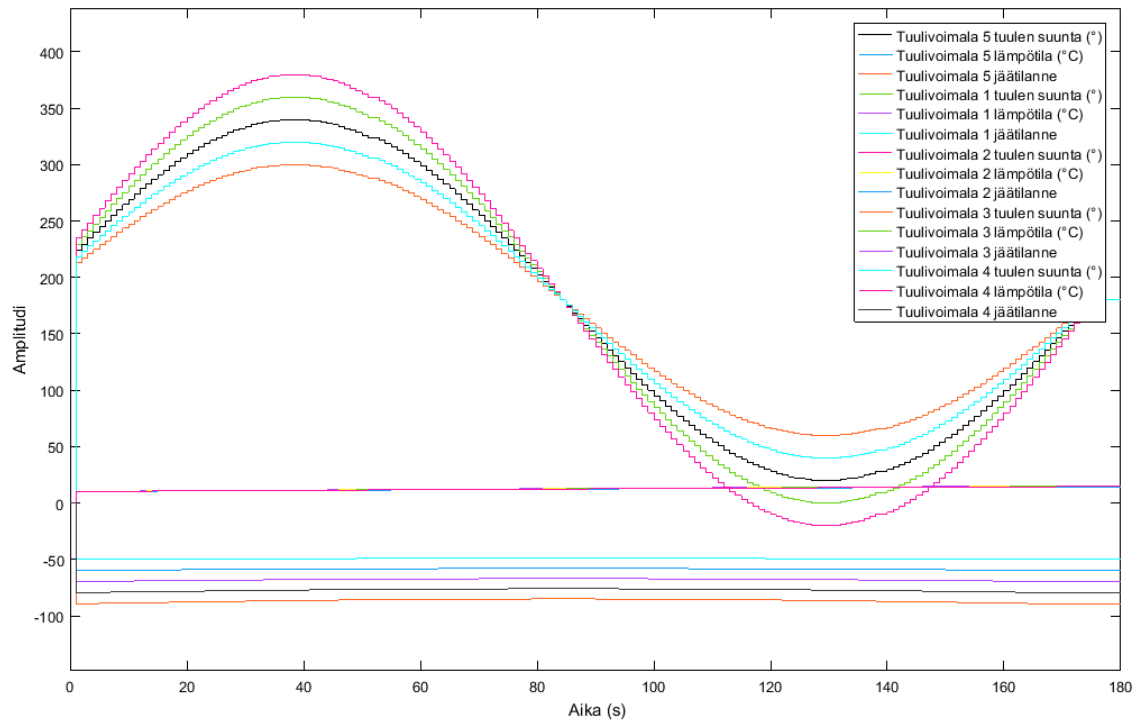
Tuulivoimaloihin simuloitujen anturien mittaamat signaalit on esitelty tässä luvussa. Signaalien suuruusluokka on realistinen pienellä tuulivoimalalla. Signaaleja mallinetaan siniaalloilla, joten ne eivät muilta ominaisuuksiltaan ole realistisia. Signaaleille on määritelty kuvissa selkeästi näkyvät arvot.

Kuvassa 21 on esitetty PC A:n välittämä suuren prioriteetin raakadata. Tässä y-akselilla on siniaallon amplitudi ja x-akselilla simulointiaika. Simulointiajaksi määritettiin kolme minuuttia eli 180 sekuntia. Suuren amplitudin signaalit ovat tuulivoimalan tehon arvoja (kW) ja pienen amplitudin signaalit ovat tuulen nopeuksia (m/s).



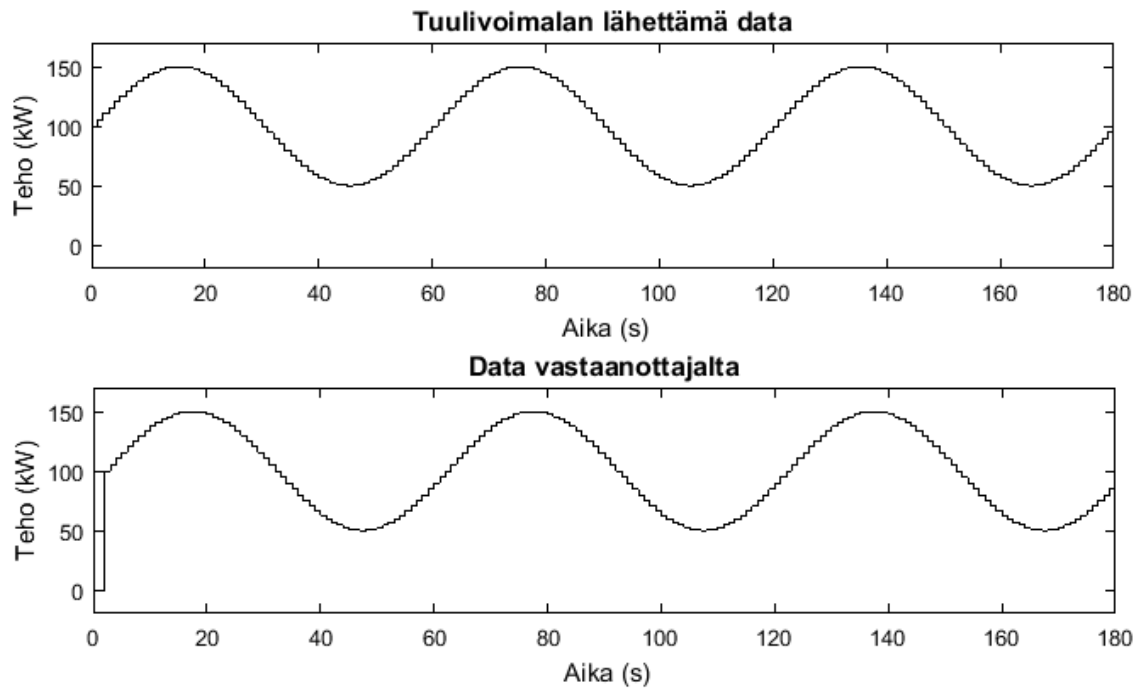
Kuva 21 Suuren prioriteetin raakadata

Kuvassa 22 on PC:ltä B välittyvä keskimääräisen ja matalan prioriteetin data. Suurin amplitudi on tuulen suunnalla. Kaikki lämpötilasignaalit ovat toistensa kanssa päällytysten, koska myös todellisuudessa lämpötila muuttuu kolmen minuutin aikana hyvin vähän. Kuvassa alhaalla olevat signaalit kuvaavat tuulivoimalan jäätilanetta. Koska signaalit ovat keskenään samanlaisia, niiden sijaintia on kuvaajassa siirretty. Absoluuttinen arvo ei ole kuvaajassa tärkeä, joten käyriä siirrettiin kuvaajassa toisistaan erottuville korkeuksille.



Kuva 22 Keskimääräisen ja matalan prioriteetin raakadata

Kuvassa 23 on vertailu tuulivoimalan lähettämästä ja vastaanotetusta datasta. Esimerkiksi signaaliksi tähän on valittu tuulivoimalan 1 lähettämä tehon arvo. Kuvassa ylempi kuvaaja on teho, joka välitetään vastaanottajalle. Tämä signaali on sama tehon signaali kuin kuvassa 16 on ja se on piirretty PC A:n lukijan välittämästä signaalista. Alempi kuvaaja puolestaan on anturin välittämä data. Matlabin scope-lohko, josta piirretään alkuperäinen data, nähdään tuulivoimalaa kuvaavassa kuvassa 15.

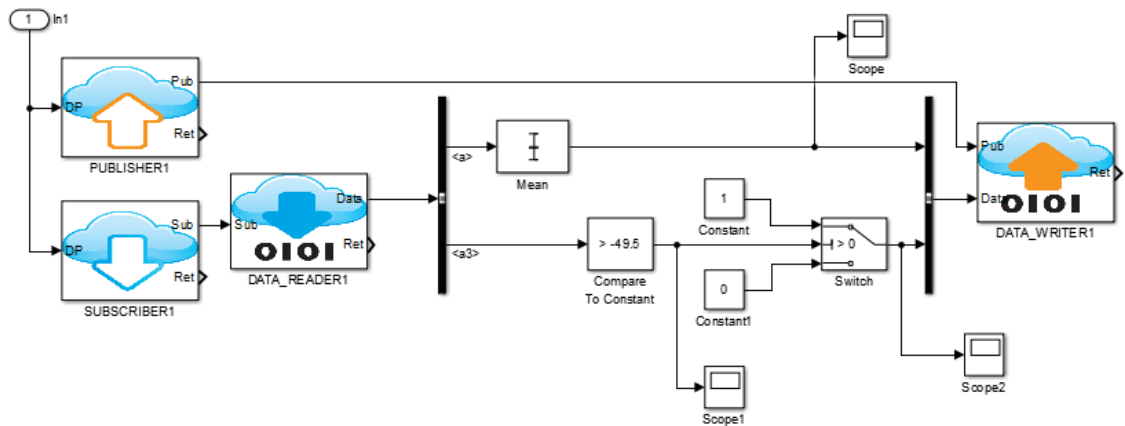


Kuva 23 Vastaanotettu ja lähetetty data

Kuvasta 23 voidaan todeta, että vastaanotettu ja lähetetty teho ovat samoja ja DDS välittää datan alkuperäisen kaltaisena. Lähetettävässä datassa on kuitenkin viive, joka johtuu siitä että DDS-ympäristön käynnistyminen vie aikaa. Data myös kirjoitetaan globaaliin datavarastoon ja luetaan sieltä toisessa kohdassa ohjelmaa. Jokainen operaatio lisää datan siirron viivettä. Myös UDP:llä on osuutensa viiveen synnyssä, joten voidaan katsoa että viive syntyy sekä UDP:n että DDS:n vaikutuksesta. Tehon signaali etenee vain mallissa PC A, joten viive ei johdu siitä, että dataa siirrettäisiin järjestelmästä toiseen ja esimerkiksi PC:lle B. Datan siirron käynnistymisen jälkeen vastaanotettu data ja lähetetty data ovat identtisiä.

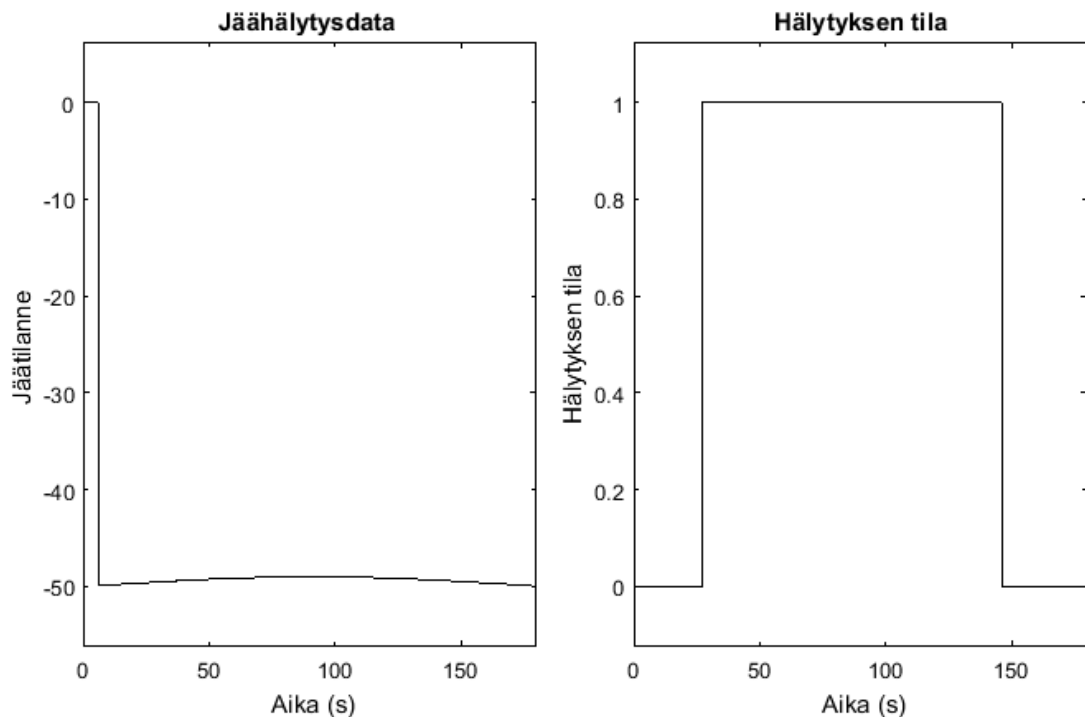
4.3 Laskenta

Alimallissa (kuva 24), joka sijaitsee PC:llä A, suoritetaan laskentaa tuulivoimaloiden datalle. Alimallin tarkoituksena on osoittaa, että DDS-järjestelmän avulla dataa on helppo siirtää paikasta toiseen ja sitä voidaan vastaanottaa missä vain paikassa, josta on pääsy samaan globaaliin datavarastoon.



Kuva 24 Laskenta

PC B lukee jäätilan tiedon sekä tuulen suunnan tuulivoimaloilta ja kirjoittaa ne globaaliin datavarastoon. Kuvan 24 esimerkissä PC A hakee keskimääräisen ja matalan prioriteetin dataa globaalista datavarastosta. Tilaa tilaa tuulen suunnan sekä jäätilan tuulivoimalalta viisi. Tuulen suunnalle lasketaan liukuva keskiarvo. Jäätilan numeerisesta arvosta asetetaan vertailemalla binäärinen hälytystieto päälle, jos jäätilan arvo ylittää määritellyn rajan (Kuva 25). Rajaksi on määritetty $-49,5$ yksikköä. Jäähälytyksen mittausdatasta nähdään selvästi mittausdatan arvo, joka pysyy nollassa kunnes DDS:n datansiirto käynnistyy.



Kuva 25 Jäähälytys

Jäähälytys on voimassa, kun funktion arvo on 1. Jos arvo on 0, jäähälytys ei ole voimassa. Järjestelmät käynnistettiin tässä hieman eri aikaan, joten voimassa oleva jäähälytys

ei näy kuvan 25 oikean puoleisessa kuvaajassa. Oikean puoleinen kuvaaja on piirretty datasta, joka on välitetty DDS:n läpi. Tällöin jäähälytyksen arvo pysyy nollassa, vaikka mittausdatan arvon perusteella jäähälytyksen pitäisi olla aluksi päällä. Jäätilannetiedoista kynnystetty arvo kirjoitetaan globaaliin datavarastoon.

DDS:n avulla dataa voidaan siis vastaanottaa useassa eri paikassa ja siirrellä tietokoneelta toiselle vaivattomasti. Laskenta-alimallissa on esitetty kuinka PC A hakee PC B:n datavarastoon kirjoittamaa keskimääräisen ja matalan prioriteetin dataa suorittaen tälle yllä mainitut laskutoimitukset ja kirjoittaa laskutoimitusten tulokset globaaliin datavarastoon. Vastaavien laskentojen lisääminen datavarastosta löytyville datoilta on tulevaisuudessakin erittäin helppoa.

4.4 Päätelmät

DDS:ää hyödyntävän Simulink-mallin rakentaminen oli aluksi haastavaa, mutta loppujen lopuksi kuitenkin sujuvaa. Simulinkin DDS-toolbox on helppokäyttöinen ja ohjeita on saatavilla. Lohkojen asetukset oli helpohko määrittellä ja DDS-toolbox on käyttäjäystävällinen. Tällaisella systeemillä DDS:n toimintaperiaatteita on helppo mallintaa ja havainnollistaa. Datan siirto paikasta toiseen sujuu vaivattomasti ja mallin avulla on helppo osoittaa miten dataa voidaan lukea missä vaan, koska kaikki on kirjoitettu globaaliin datavarastoon.

Mallin tavalla rakennetun järjestelmän ongelmana voi olla DDS datan kirjoittajien sijainti. Jos PC A tai PC B kaatuu, myöskään DDS ei toimi kokonaisuudessaan. Vain toimivan tietokoneen suoraan UDP-liikenteen kautta vastaanottamat datat ovat saatavilla. Jos datan kirjoitus globaaliin datavarastoon tapahtuisi tuulivoimalassa, vastaanottavien tietokoneiden toimivuus ei olisi niin kriittistä. Tosin datan käsittely tapahtuu tietokoneilla, jolloin pelkkä raakadatan olemassaolo pilvessä ei välttämättä riitä.

DDS on myös hieman monimutkainen. Julkaisija-tilaaja –malli tuo rakenteeseen hankaluuksia, joten DDS-järjestelmää ei kannata rakentaa pienten ja yksinkertaisten datojen välittämiseen. Esimerkiksi siinä tapauksessa, jos vastaanottajia ja lähettäjiä on molempia vain yksi. Jos datan vastaanottajia on useita, tarvitsisi perinteisesti rakentaa dataa jakava palvelin. Tällöin DDS käyttö on perusteltua, koska se välittää datan luotettavasti ja yksinkertaisesti vastaanottajalle.

Kuten aiemmin todettiin kappaleessa 4.2, järjestelmässä syntyy viive DDS:n vuoksi. Viive syntyy, koska välikerrosohjelmiston käynnistyminen vie aikaa. Tämä ongelma ilmenee vain Simulink-mallissa, sillä todellisuudessa DDS on jatkuvassa toiminnassa. Myös datan kirjoittaminen pilveen vie aikaa, jolloin lukeminen voidaan aloittaa vasta kun tietyn hetken data on jo kirjoitettu. Tässä Simulink-mallissa viivettä lisää myös käytössä oleva UDP. Data voidaan siis kirjoittaa pilveen, vasta kun PC vastaanottaa datan UDP:n kautta. Todellisessa järjestelmässä tulee mukaan myös tietoliikenneverkkojen

viive. Esimerkiksi mobiilidatan viive voi olla satoja millisekunteja ja langattoman lähiverkon viive voi olla kymmeniä millisekunteja.

Kaupallisten toimijoiden DDS-lisenssit ovat kalliita. Esimerkiksi RTI:n lisenssi maksaa useita tuhansia dollareita. Ilmaisiakin vaihtoehtoja on, mutta niiden ohjelmistotuki on heikompaa kuin RTI:llä. Esimerkiksi Matlabin Simulinkiin ei ole moduuleita simulointia varten yliopistoille ilmaisissa ohjelmistoissa.

5. YHTEENVETO

Tämän diplomityön tarkoituksena oli tutkia mitä ovat asioiden internet ja Data Distribution Service (DDS). Tarkoituksena oli myös tutkia miten DDS:n julkaisija-tilaaja – malli toimii. Tässä työssä tutkittiin mitä asioiden internetin ja DDS:n taustalla on ja mistä niiden toiminta koostuu. DDS:n toimintaa havainnollistettiin Simulink-mallilla, joka rakennettiin käyttäen Simulinkin DDS-kirjastoa sekä RTI:n DDS-kirjastoa. Käytännön osion tarkoituksena oli siis tutkia miten DDS toimii ja kuinka datan siirto tapahtuu tietokoneelta toiselle.

Asioiden internet on älykkäiden laitteiden verkko. Laitteita voivat olla esimerkiksi anturit, jotka voivat sijaita nostureissa, trukeissa tai älytalon lämmitysjärjestelmässä. Anturit lähettävät dataa pilvipalveluun, josta dataa voidaan hajauttaa kohteille, jotka sitä tarvitsevat. Asioiden internetin toiminta perustuu koneiden väliseen tiedonvälitykseen (M2M), joskin IoT on näistä kahdesta kehittyneempi ratkaisu sekä käyttää internetiä tiedon siirtoon ja jakamiseen.

Asioiden internet on kuitenkin vasta alkutaipaleellaan siitä huolimatta, että tietoa löytyy paljon ja yritykset ovat siitä kiinnostuneita. Asioiden internetin ratkaisut ovat vielä melko hajanaisia ja ratkaisua selkeälle arkkitehtuurille ei ole. Kehitystä kuitenkin tapahtuu kokoajan ja tulevaisuudessa tilanne tulee olemaan parempi. Hallitsevan protokollan löytyminen edistäisi asioiden internetin suosiota entisestään. Liiketoiminta asioiden internetin ympärillä on muodostunut tietyistä teknologiaratkaisuista, jotka usein keskittyvät tiettyihin sovellusalueisiin. Esimerkiksi RFID, jonka ratkaisuja myydään vähittäiskaupassa sekä koneiden välinen tiedonvälitys, jota käytetään etäyhteyden muodostamiseen ovat tällaisia sovellusalueita.

Asioiden internetin yhteydessä käytetään myös termiä arkkitehtuurinen referenssimalli eli ARM. Arkkitehtuurinen referenssimalli tarjoaa pohjan asioiden internetin järjestelmien suunnittelulle. Tämä malli koostuu kolmesta osasta, joita ovat asioiden referenssimalli, referenssiarkkitehtuuri sekä ohjenuorat. Nämä osat koostuvat vielä useista alimalleista, jotka ohjaavat asioiden internetin järjestelmän suunnittelutyötä.

DDS on datan hajauttamiseen sovellettava standardi, joka mahdollistaa kommunikaation järjestelmän fyysisen tason ja sovellustason välillä. DDS integroi systeemin komponentit yhteen ja tarjoaa sujuvaa datan siirtoa sekä luotettavuutta. Se tarjoaa myös arkkitehtuurin liiketoimintaa ja asioiden internetiä varten. DDS ei käytä toiminnassaan erilisiä palvelimia, vaan sen käsittelemä data varastoidaan globaaliin tietovarastoon, jolloin sen luotettavuus lisääntyy. Data Distribution Service tarjoaa myös arkkitehtuurin asioi-

den internetiä varten. Muut protokollat esimerkiksi MQTT, AMQP ja CoAP ovat yksinkertaisia ja käyttötapaukset ovat rajallisia. DDS puolestaan on ominaisuuksiltaan monipuolinen ja asioiden internetin tarjoaman monimutkaisen datan käsittely onnistuu siltä hyvin.

Lopuksi diplomityössä on esitelty Mathworksin Simulinkia käyttäen rakennettu malli, jonka tarkoituksena on havainnollistaa DDS:n toimintaa. Mallissa on viisi tuulivoimaa, jotka lähettävät dataa sekä kaksi tietokonetta. Tietokoneet vastaanottavat ja käsittelevät dataa sekä siirtävät sitä toisilleen. Mallin rakennus onnistui kohtuullisessa ajassa, koska ohjeita on saatavilla. Malli myös havainnollistaa DDS:n toimintaa hyvin sekä esittelee datan siirron ominaisuudet.

Kokonaisuudessaan asioiden internet ja DDS ovat tulevaisuutta. DDS ei välttämättä ole tulevaisuudessa se suosituin välikerrosohjelmisto, mutta asioiden internetin suosio on varmaa. Yritykset hyötyvät tällaisesta kustannustehokkaasta ohjelmistojen kehityksestä, joka edistää liiketoimintaa. Sujuva tiedonsiirto ja datan käsittely ovat nykyaikaa.

LÄHTEET

- [1] Krister Wikström, Tietoturva tärkeää teollisessa internetissä, Sähkö ja tele -lehti, Sähköinsinööriliitto ry, 3/2016, s.16-18
- [2] Panu Vatanen, Yle, 19.4.2016 <http://yle.fi/uutiset/3-8821146> Viitattu 21.11.2016
- [3] Reaping the Benefits of the Internet of Things, 5/2014, 10 s.
<https://www.cognizant.com/InsightsWhitepapers/Reaping-the-Benefits-of-the-Internet-of-Things.pdf> Viitattu 21.11.2016
- [4] Mazhelis, Warma, Leminen, Ahokangas, Pussinen, Rajahonka, Siuruainen, Okkonen, Shveykovskiy, Myllykoski, Internet of Things Market, Value Networks and Business Models: State of the Art Report, 95 s.
<http://internetofthings.fi/extras/IoTSOTAReport2013.pdf> Viitattu 21.11.2016
- [5] How Does DDS Compare to Other Technologies,
<http://portals.omg.org/dds/features-benefits/> Viitattu 21.11.2016
- [6] Ashton K., That ‘Internet of Things’ Thing, RFID Journal, Jun 22, 2009.
<http://www.rfidjournal.com/articles/view?4986> Viitattu 21.11.2016.
- [7] Höller J., Tisatsis V., Mulligan C., Avesand S., Karnouskos S., Boyle D., From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, 1st ed. Academic Press, UK, 2014, 352 p.
- [8] Feng Xial, Laurence T. Yang, Lizhe Wang, Alexey Vinel, International journal of communication systems Int. J. Commun. Syst. 2012, Published online in Wiley Online Library.
<https://www.homeworkmarket.com/sites/default/files/q5/04/07/danainfo.acppwiszgmk2n0u279qu76contentserver.pdf> viitattu 21.11.2016
- [9] Matt Hatton, 1/2013, 8 s.
http://www.telecomengine.com/sites/default/files/temp/CEBIT_M2M_WhitePaper_2012_01_11.pdf Viitattu 21.11.2016
- [10] Angelo Corsaro, Building the Internet of Things with DDS,
<http://www.omg.org/news/meetings/tc/nj-13/special-events/iot-pdfs/corsaro.pdf> Viitattu 21.11.2016.

- [11] IoT-A, Deliverable D1.5 – Final architectural reference model for the IoT v3.0, Iot-A, 2013, 482 s
- [12] OMG, <http://portals.omg.org/dds/what-is-dds-3/> Viitattu 21.11.2016
- [13] Angelo Corsaro, The DDS Tutorial, 173 s. http://www.laas.fr/files/SLides-A_Corsaro.pdf Viitattu 21.11.2016
- [14] P. Santhi Thilagam, Alwyn Roshan Pais K., Chandrasekaran N., Balakrishnan, Advanced Computing, Networking, Security, International Conference, AD-CONS 2011, Surathkal, India, December 16-18, 2011 Revised Selected Papers.
- [15] Joseph M. Schlesselman, Gerardo Pardo-Castellote, Bert Farabaugh, OMG Data Distribution Service (DDS): Architectural Update, 2 s. <https://www.semanticscholar.org/paper/Omg-Data-distribution-Service-dds-Architectural-Schlesselman-Clara/e25a9c9dc8b09b7717598db29b3ccbf427539065/pdf> Viitattu 21.11.2016
- [16] Data Distribution Service for Real-time Systems Versio 1.2, OMG Available Specification Formal 07-01-01, 260 s.
- [17] DDS Data Local Reconstruction Layer (DDS-DLRL) Version 1.4., OMG Document Number: formal/2015-04-12, formal 84 s.
- [18] The Real-time Publish-Subscribe Protocol (RTPS) DDS Interoperability Wire Protocol Specification Version 2.2, OMG Document Number: formal/2014-09-01, OMG Document Number: formal/2014-09-01, formal, 210 s.
- [19] Stan Scheider, What's the Difference Between Message-centric and Data-centric Middleware, Electronic Design, 6.7.2012 <http://electronicdesign.com/embedded/whats-difference-between-message-centric-and-data-centric-middleware> Viitattu 21.11.2016
- [20] Twinoaks Computing, What can DDS do for you? 16 s., http://www.omg.org/hot-topics/documents/dds/CoreDX_DDS_Why_Use_DDS.pdf Viitattu 16.11.2016