

IoT-projektin dokumentointi

IoT:sta liiketoimintaa

Harri Jäntti

Opinnäytetyö
Toukokuu 2019
Tekniikan ja liikenteen ala
Tieto- ja viestintätekniikka, Tietoverkkotekniikka

| | | |
|---|-------------------------------------|-----------------------------------|
| Tekijä(t) Jännti, Harri | Julkaisun laji Opinnäytetyö, AMK | Päivämäärä Toukokuu 2019 |
| | Sivumäärä 64 | Julkaisun kieli Suomi |
| | | Verkojulkaisulupa myönnetty: x |
| Työn nimi IoT-projektin dokumentointi IoT:sta liiketoimintaa | | |
| Tutkinto-ohjelma Tieto- ja viestintäteknikka, Tietoverkkotekniikka | | |
| Työn ohjaaja(t) Olli Väänänen, Mika Rantonen | | |
| Toimeksiantaja(t) Jyväskylän ammattikorkeakoulu | | |
| <p>Tiivistelmä</p> <p>Projektin tarkoituksena oli tutkia erilaisten laitteiden, ohjelmistojen ja yhteyksien mahdollisia käyttökohteita Internet of Things (IoT)-ympäristössä. Projektin motivaattorina toimivat trendit, jotka siirtävät erilaisia palveluja fyysisistä toteutuksista internetpohjaisiksi toteutuksiksi. Hankkeeseen osallistuvat Jyväskylän ammattikorkeakoulu (JAMK) ja Jyväskylän yliopisto (JYU) aikavälillä 01.2017 – 31.03.2019.</p> <p>Projekti toteutettiin JAMKin IoTista liiketoimintaan (IoTli) -projektiryhmän kanssa, jonka yhtenä osa-alueena tutkittiin ja toteutettiin erilaisia LoRaWAN-ratkaisuja. LoRaWAN-ratkaisuja tehtiin kaksi kappaletta, joista ainoastaan toisesta tehtiin toimiva kokonaisratkaisu.</p> <p>Projektin aikana tutkittiin LoRaWAN-laitteita, joiden avulla pystytään lähettämään antureiden keräämät tiedot LoRaWAN-yhdyskäytävälle. Näiden laitteiden avulla rakennettiin järjestelmä, joka lähettää tiedot vapaan lähdekoodin The Things Network -järjestelmään, josta tiedot pystyttiin lähettämään omaan visualisointijärjestelmään, joka toteutettiin käyttäen ThingsBoard-ohjelmistoa. LoRaWAN-yhteyksiä tutkittiin liittämällä Adeunis LoRaWAN-yhteystietojen tutkimiseen tehty laite Digitan järjestelmään, jonka toimintaa tutkittiin, mutta ei toteutettu. Projektin aikana käytiin myöskin yliopiston Digitan järjestelmälle suunniteltua visualisointitoteutusta läpi.</p> <p>LoRaWAN-ratkaisuilla voidaan tehdä pitkän kantamatkan IoT-ratkaisuja, mutta järjestelmässä uhrataan tiedonsiirtokapasiteettia. Järjestelmä soveltuu parhaiten toteutuksiin, joissa tietojen lähetystä ei tarvitse tehdä koko ajan akun virran säästämisen takia. Järjestelmät voivat vaihdella huomattavan paljon, sillä lähetettäviä laiteratkaisuja on markkinoilla paljon ja tietojen visualisointi on riippuvainen käytettävästä ohjelmistoratkaisusta.</p> | | |
| Avainsanat (asiasanat) | | |
| Muut tiedot (salassa pidettävät liitteet) | | |

| | | |
|---|--|---|
| Author(s) Jäntti, Harri | Type of publication Bachelor's thesis | Date May 2019 Language of publication: Finnish |
| | Number of pages 64 | Permission for web publication: x |
| Title of publication IoT project's documentation Business growth from IoT | | |
| Degree programme Data Network Technology | | |
| Supervisor(s) Väänänen Olli, Rantonen Mika | | |
| Assigned by Jyväskylän Ammattikorkeakoulu | | |
| Abstract <p>The purpose of this project was to explore different hardware, software and connection possibilities for their usage in an IoT environment. This project was motivated by the current trends shifting different services from a physical implementation in to an internet-based implementation. The participants of this project were from JAMK University of Applied Sciences (JAMK) and the University of Jyväskylä, and the project lasted from January 2017 to 31 March 2019.</p> <p>The project was implemented by JAMK project group. One of the technologies explored was LoRaWAN network, which was used for the implementation. Two LoRaWAN solutions were studied; however, only one of them was fully implemented.</p> <p>During the project the physical devices of LoRaWAN were investigated, which allowed the project group to gather information and send it to the group's LoRaWAN Gateway. With these devices, a system using open source was constructed. The Things Network solution was used to send it to the group's own visualization system made using ThingsBoard software. LoRaWAN connections were monitored using an Adeunis LoRaWAN tester, which allowed to monitor connectivity details. The Adeunis tester was connected to a LoRaWAN network provided by Digita, the functionality of which was studied; however, no implementation was produced with it. The university's visualization solutions for the Digita system were also explored.</p> <p>LoRaWAN can be utilized to make long range IoT solutions, which do not require much network transfer capacity. To save the battery life, the best implementations for a LoRaWAN system is the one that does not send traffic constantly. LoRaWAN solutions can vary based on the physical devices and the programs used to visualize the received information.</p> | | |
| Keywords/tags (subjects) | | |
| Miscellaneous (Confidential information) | | |

Sisältö

| | |
|--|----------|
| Lyhenteet | 5 |
| 1 Työn kuvaus | 7 |
| 1.1 Tausta | 7 |
| 1.2 Toteutukset | 7 |
| 1.3 Tutkimusmenetelmät | 7 |
| 1.4 Tutkimus | 8 |
| 2 Projektin teknologiat ja ohjelmat | 9 |
| 2.1 LoRa | 9 |
| 2.1.1 Yleistä | 9 |
| 2.1.2 Modulaatio | 9 |
| 2.1.3 Modulaation tärkeimmät ominaisuudet | 11 |
| 2.2 LoRaWAN | 13 |
| 2.2.1 Yleistä | 13 |
| 2.2.2 Verkkoarkkitehtuuri | 14 |
| 2.2.3 Pariston elinaika | 15 |
| 2.2.4 Verkon kapasiteetti | 16 |
| 2.2.5 Laitekategoriat | 17 |
| 2.2.6 Tietoturva | 18 |
| 2.2.7 Alueelliset parametrit | 22 |
| 2.3 The Things Network | 24 |
| 2.3.1 Verkkoarkkitehtuuri | 24 |
| 2.3.2 Keskeisimmät toiminallisuus | 27 |
| 2.3.3 Velvollisuuksien erottaminen | 29 |

| | | |
|----------|--|-----------|
| 2.3.4 | Yhdyskäytävän protokolla käännökset | 30 |
| 2.3.5 | Alaspäin menevän tietoliikenteen konfigurointi | 30 |
| 2.3.6 | Monistuneiden lähetysten poistaminen | 32 |
| 2.3.7 | Laitteen ja ohjelmiston etsintä | 33 |
| 2.3.8 | Kehyslaskurin tarkistus | 33 |
| 2.3.9 | Metadata keräys | 34 |
| 2.3.10 | Parhaan alaspäin menevän yhteyksien asetusten valinta..... | 34 |
| 2.3.11 | Laitte tilat ja MAC komennot..... | 35 |
| 2.3.12 | Viestien salauksen poistaminen | 35 |
| 2.3.13 | Hyötykuorman konversio | 35 |
| 2.3.14 | Alaspäin menevä liikenne | 36 |
| 2.4 | ThingsBoard..... | 37 |
| 2.5 | ThingsBoard IoT Gateway..... | 37 |
| 2.6 | LoRaWAN-laitteisto | 38 |
| 2.6.1 | LoPy4..... | 39 |
| 2.6.2 | Expansion Board ja Pysense | 39 |
| 2.6.3 | Adeunis FTD | 41 |
| 2.7 | Elasticsearch ja Grafana | 41 |
| 3 | Toteutukset ja tutkimustyö..... | 43 |
| 3.1 | The Things Network LoRaWAN -toteutus | 43 |
| 3.1.1 | The Things Network -konfigurointi lähetyksille..... | 43 |
| 3.1.2 | LoRaWAN-laitteiden käyttöönotto | 44 |
| 3.1.3 | TTN tietojen muuntaminen | 46 |
| 3.1.4 | ThingsBoard ja IoT Gateway | 47 |

| | | |
|----------|---|-----------|
| 3.1.5 | Järjestelmän toiminta kokonaisuutena | 50 |
| 3.2 | LoRaWAN Digita -tutkimus | 52 |
| 3.2.1 | Yleisesti | 52 |
| 3.2.2 | Adeunis FTD käyttöönotto | 53 |
| 3.2.3 | Digitan taustajärjestelmä | 54 |
| 3.2.4 | End Pointin pystytys | 56 |
| 4 | Pohdinta..... | 57 |
| | Lähteet | 60 |
| | Liitteet..... | 62 |
| | Liite 1. LoPy4-yhteyksien ohjelmointi | 62 |
| | Liite 2. ThingsBoard MQTT-konfigurointi | 64 |

Kuviot

| | |
|---|----|
| Kuvio 1. Chirp-signaalin havainnollistaminen | 11 |
| Kuvio 2. LoRaWAN-protokollan toiminta | 14 |
| Kuvio 3. LoRaWAN-verkon rakenne | 15 |
| Kuvio 4. Salausviestien toiminta | 20 |
| Kuvio 5. LoRaWAN paketin rakenne ja turvallisuus | 21 |
| Kuvio 6. TTN-reititys | 25 |
| Kuvio 7. TTN-komponentit | 26 |
| Kuvio 8. Komponenttien velvollisuudet | 29 |
| Kuvio 9. Järjestelmä kokonaisuudessaan | 43 |
| Kuvio 10. Nanogateway.py muokkaukset | 45 |
| Kuvio 11. Decoder-funktiot | 46 |
| Kuvio 12. Muuntamisen jälkeiset tiedot | 47 |
| Kuvio 13. Yhdyskäytävän lisääminen ThingsBoardissa | 48 |
| Kuvio 14. TTN-ohjelmisto ID ja ohjelmistotokeni..... | 49 |
| Kuvio 15. ThingsBoard-yhdyskäytävälisäykset | 50 |
| Kuvio 16. Liikennöinti yhdyskäytävässä | 51 |
| Kuvio 17. Tietojen visualisointi ThingsBoardissa..... | 52 |
| Kuvio 18. Adeuniksen lisääminen Digitan järjestelmään | 55 |

Taulukot

| | |
|--|----|
| Taulukko 1. LoRaWAN-yhteyksien alueelliset määrittelyt..... | 22 |
| Taulukko 2. Expansion Board 2.0 | 40 |
| Taulukko 3. Pysensen ominaisuudet | 40 |
| Taulukko 4. Adeunis-järjestelmän muokkauskomennot ja selitteet | 54 |

Lyhenteet

| | |
|---------------|---|
| ADR | Adaptive Data Rate |
| AMQP | Advanced Message Queuing Protocol |
| AES | Advanced Encryption Standard |
| API | Application Programming Interface |
| BGP | Border Gateway Protocol |
| BT | Bandwidth Time Product |
| CBC | Cipher Blocker Chaining |
| CMAC | Cipher-based Message Authentication Code |
| CoAP | Constrained Application Protocol |
| CSS | Chirp Spread Spectrum |
| CTR | Counter Mode Encryption |
| DSSS | Direct-Sequence Spread Spectrum |
| EUI-64 | Extended Unique Identifier-64 |
| FCC | Federal Communications Commission |
| FECs | Forward Error-correction Codes |
| FHSS | Frequency Hopping Spread Spectrum |
| FSK | Frequency Shift Keying |
| FTD | Field Test Device |
| GPS | Global Positioning System |
| GPRS | General Packet Radio Service |
| GUI | Grafical User Interface |
| HTTPS | Hypertext Transfer Protocol Secure |
| IEEE | Institute of Electrical and Electronics Engineers |
| IoT | Internet of Things |
| IoTli | IoTista liiketoimintaan |

| | |
|-----------------|--|
| IP | Internet Protocol |
| ISM | Industrial, Scientific and Medical |
| JAMK | Jyväskylän Ammattikorkeakoulu |
| JYU | Jyväskylän Yliopisto |
| LiPo | Lithium polymer |
| LoRa | Long Range |
| LoRaWAN | Long Range Wide Area Network |
| LR-WPANs | Low-Rate Wireless Personal Area Networks |
| MAC | Media Access Control |
| MD5 | Message-digest 5 |
| MQTT | Message Queuing Telemetry Transport |
| MIC | Message Integrity Code |
| OFDM | Orthogonal frequency-division multiplexing |
| OQPSK | Offset Quadrature Phase-Shift Keying |
| OSGI | Open Service Gateway Initiative |
| OTAA | Over the Air Activation |
| OUI | Organizational Unique Identifier |
| PA | Power Amplifier |
| PYH | OSI mallin fyysinen taso |
| QoS | Quality of Service |
| REST | Representational State Transfer |
| RPC | Remote Procedure Calls |
| RSSI | Received Signal Strength Indicator |
| SF | Spreading Factor |
| SNR | Signal-to-noise ratio |
| TTN | The Things Network |
| UA | Unified Architecture |
| VPN | Virtual Private Network |

1 Työn kuvaus

1.1 Tausta

Opinnäytetyössä käydään läpi IoTista liiketoimintaan -projektin tutkittuja ja toteutettuja järjestelmiä. IoTli on Jyväskylän ammattikorkeakoulun (JAMK) ja Jyväskylän yliopiston (JYU) tutkimushanke, jossa tutkittiin erilaisia Internet of Things (IoT) -toteutuksia uuden liiketoiminnan kehittämiseksi. Projektin tarkoituksena oli tutkia erilaisia antureita, sulautettuja alustoja, langatonta ja langallista tiedonsiirtoa, ohjelmistoja ja tietokantoja. Projekti toteutettiin aikavälillä 1.1.2017 – 31.3.2019, ja itse liityin projektiin mukaan 28.9.2017. (IoTli- IoT:sta uutta liiketoimintaa 2018.)

1.2 Toteutukset

Opinnäytetyössä käydään läpi vain osa projektissa tutkituista järjestelmistä aikarajoitteiden ja projektin laajuuden takia. Opinnäytetyössä käytävistä asioista toteutettiin Long Range Wide Area Network (LoRaWAN) -yhteyksillä toimiva anturiverkko, jossa käytetään avoimen lähdekoodin The Things Network (TTN) -järjestelmää viestien lähettämiseen omaan visualisointiratkaisuun. Muita järjestelmiä tutkittiin Digitan tarjoaman verkon osalta liittämällä Adeunis 868 Field Test Device (FTD) Digitan verkkoon. Digitan palveluista ei tehty toteutusta, koska projektilla ei ollut tarvetta tai aikaa tehdä samanlaista toteutusta, kuin TTN järjestelmän kanssa tehtiin, mutta järjestelmää kuitenkin tutkittiin. Projektissa käytiin myöskin läpi yliopiston tekemä visualisointiratkaisu Digitan LoRaWAN palveluille, jonka toimintaa tutkittiin.

1.3 Tutkimusmenetelmät

Opinnäytetyössä suoritetaan tutkimus ja kehitystyötä, jossa kerätään olemassa olevaa tietoa LoRaWANia käyttävistä laitteista, järjestelmistä ja niiden mahdollisista toteutuksista. Tutkimustyötä tehdään tietoja keräämällä ja toteuttamalla fyysinen järjestelmä, jonka avulla saadaan syvempi ymmärrys teknologian toimintaan. Tämän

tutkimustyön pohjalta mahdollistetaan IoT-ympäristön toteutus ja jatkokehitys LoRaWAN-yhteyksillä.

1.4 Tutkimus

Opinnäytetyössä tutkittiin ja toteutettiin IoT-ratkaisu käyttäen Pycomin LoPy4 LoRaWAN -laitteita. Toteutusta varten perehdyttiin TTN:n tarjoamiin LoRaWAN-ratkaisuihin. Projektissa tutkittiin ThingsBoard ohjelmaa, jolla voidaan hallinnoida laitteita, kerätä tietoja, prosessoida tietoja ja visualisoida tietoja. Tutkimustyötä tehtiin Digitan LoRaWAN-palveluiden osalta, jonka toiminnallisuutta testattiin liittämällä Adeunis FTD järjestelmään. Digitan palveluihin yliopisto oli tehnyt oman visualisointijärjestelmänsä, jota tutkittiin projektin aikana ja pystytettiin virtuaalisesti, mutta järjestelmää ei toteutettu kokonaisuudessaan Digitan palveluihin yhdistämällä.

2 Projektin teknologiat ja ohjelmat

2.1 LoRa

2.1.1 Yleistä

LoRa (Long Range) on pitkän kantomatkan langattomaan kommunikaatioon suunniteltu järjestelmä, joka on yhdysvaltalaisen Semtech-yrityksen yksityisomistuksessa. Järjestelmän tarkoitus on tuoda pitkällä kommunikaatio ja kestoajalla toimiva järjestelmä, joka uhraa kaistan nopeuden tämän saavuttamiseen. LoRa termillä viitataan yleisesti seuraaviin kahteen erilaiseen tasoon: fyysinen taso, joka käyttää Chirp Spread Spectrum (CSS) -radiomodulaatiotekniikkaa, tai vaihtoehtoisesti Media Access Control (MAC) -tason protokollaan LoRaWAN-yhteyksissä. (Augustin, Clausen, Kim, Townsley & Yi 2016.)

LoRan fyysisen tason tarkoitus on tarjota pitkän kantomatkan, pienen virrankulutuksen ja pienellä välityskyvyllä toimiva kommunikointijärjestelmä. Järjestelmä toimii Industrial, Scientific and Medical -taajuuksilla (ISM) 430, 433, 868 tai 915 MHz maantieteellisestä sijainnista riippuen, eli se käyttää hyväkseen lisenssivaatimuksista vapaina olevia kanavia. Tämän ansiosta LoRa-toteutuksia tehdessä ei tarvitse hankkia kalliita lisenssejä, kuten olisi tehtävä, jos järjestelmä olisi tehty käyttäen matkapuhelinverkkoja. Tässä tapauksessa, kun puhutaan pitkän kantomatkan verkoista etäisyydet, ovat yli 10 km:n luokkaa, jotka voivat olla joko julkisia tai yksityisiä verkkoja. Jokaisen lähetyksen hyötykuorma voi vaihdella 2-255 oktetin välillä ja datanopeus voi saavuttaa 50 Kbps nopeudet, mikäli kanavien yhdistäminen on käytössä. (Augustin ym. 2016.)

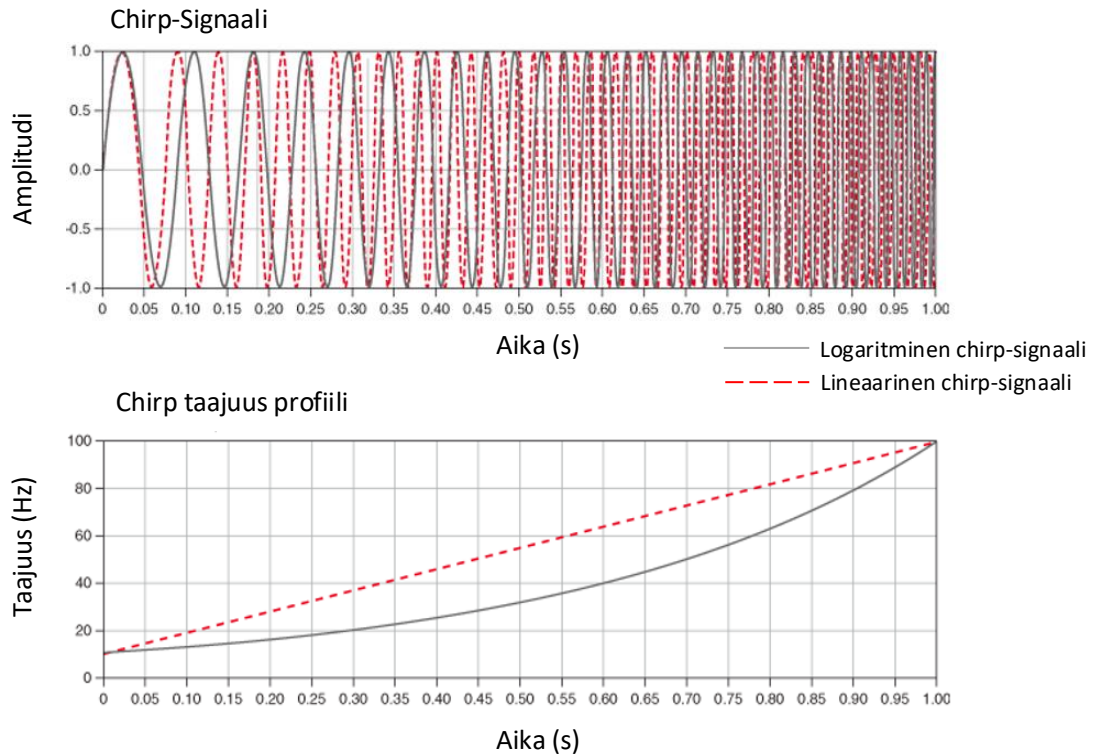
2.1.2 Modulaatio

LoRan käyttämä CSS-modulaatio kehitettiin alkuperäisesti käytettäväksi tutkiin 1940-luvulla. Perinteisesti sitä käytettiin monenlaisiin armeijan ja turvallista kommunikointia vaativiin järjestelmiin, mutta viimeisten 20 vuoden aikana tätä teknologiaa on

aloitettu käyttämään isommissa määrin tiedonsiirtojärjestelmissä, koska järjestelmällä on suhteellisen vähäiset vaatimukset lähetystehon suhteen ja sen luontainen kestävyys kanavien rappeutumista aiheuttaville mekanismeille, kuten monitie-etene-
misen aiheuttamiin signaali häiriöihin, signaalin heikentymiseen, Dopplerin ilmiöstä
aiheutuviin häiriöihin ja samalla taajuuskaistalta tuleville häiriöille. (LoRa Modulation
Basics 2015.)

CSS PHY (PHY:llä viitataan OSI mallin fyysiseen tasoon) otettiin käyttöön Institute of
Electrical and Electronics Engineersin (IEEE) toimesta sen Low-Rate Wireless Personal
Area Networks (LR-WPANS) -standardiin 802.15.4, jotta pystytään saavuttamaan pi-
demmät kantomatkat ja paremman liikuteltavuuden, kuin on mahdollista saada, jos
käytetään Offset Quadrature Phase-Shift Keying (OQPSK) Direct-Sequence Spread
Spectrum (DSSS) PHY -käytäntöjä. Semtechin LoRa-modulaatiolla käsitellään kaikkia
ongelmia, joita DSSS-järjestelmässä tulee vastaan, jotta saadaan pienellä hinnalla ja
pienellä virrankulutuksella erittäin kestävä vaihtoehto perinteiselle hajaspektritekni-
kalla toimiville kommunikointitekniikoille. (LoRa Modulation Basics 2015.)

LoRa-modulaatiossa spektrin hajautus toteutetaan luomalla chirp-signaali, jolla on
jatkuvasti vaihteleva taajuus. Tämän tavan hyöty tulee siitä, että ajoitus ja taajuussiir-
ros lähettäjän ja vastaanottajan välillä ovat samanarvoiset, mikä vähentää vastaan-
ottavan laitteen monimutkaisuutta. Chirp-signaalin taajuuskaistan leveys on saman-
arvoinen kuin spektrin kaistanleveyden signaali. Datasignaali lohkaistaan korkeilla tie-
donsiirtonopeuksilla ja moduloidaan chirp-signaaliin. Kuviossa 1 pyritään havainnoi-
listamaan chirp-signaalin toimivuutta. (LoRa Modulation Basics 2015.)



Kuvio 1. Chirp-signaalin havainnollistaminen (Continuous Frequency Sweep 2018, muokattu)

2.1.3 Modulaation tärkeimmät ominaisuudet

LoRa-modulaatio on kaistanleveydeltään ja taajuudeltaan skaalautuva. Sitä voidaan käyttää kapeakaistaisiin taajuushyppelyihin ja laajakaistaisille suorasekvenssisovelluksille. Toisin kuin muut kapeakaistaiset ja laajakaistaiset modulaatiomallit LoRa voidaan helposti muokata toimimaan kummallakin toimintamallilla tekemällä vain muutamat muutokset asetuksiin. (LoRa Modulation Basics 2015.)

LoRa on samanlainen toiminnaltaan kuin Frequency Shift Keying (FSK) olemalla jatkuvasti kuoressa oleva modulaatiomalli. Jatkuvasti kuoressa olevalla modulaatiolla tarkoitetaan sitä, että kanta-aallon amplitudi on aina vakio riippumatta modulointisignaalin vaihteluista (Peerla 2007, 2). Tämän ansiosta saadaan pienellä hinnalla ja virrankulutuksella toimivat korkeat Power Amplifier (PA)-tasot, joita voidaan uudelleen

käyttää ilman muokkauksia. Lisäksi prosessointihyödyt, joita LoRa antaa, voidaan lähettimen lähtötehoja vähentää verrattuna FSK-yhteyksiin, kun samalla pystytään ylläpitämään samantasoista tai parempaa suhdetta lähettimen vastaanottamien ja menetettyjen lähetysten välillä. (LoRa Modulation Basics 2015.)

Suuren Bandwidth Time Product (BT) arvon ($BT > 1$) ja asynkronisen luonteen takia LoRa-signaali on erittäin vastustuskykyinen kaistan sisäisten ja kaistan ulkopuolelta tulevien häiriöiden kanssa. LoRa-symbolijaksojen kesto on pitempi kuin tyypillisten Frequency Hopping Spread Spectrum (FHSS)-järjestelmien, jonka ansiosta se saa hyvän vastustuskyvyn pulssi-amplitudimodulaatio-häirintää vastaan. Virheet, jotka voivat syntyä pystytään korjaamaan Forward Error-correction Coden (FEC) avulla. FEC on häiriöiden korjaustekniikka, jolla havaitaan ja korjataan rajallinen määrä virheitä lähetysten tiedoista ilman tarvetta tehdä uudelleenlähetystä (Thakur 2018). Tyypillisesti vastaanotettavan maksimitehon suhde taajuuskaistan ulkopuolelta tulevien häiriöiden ja LoRa signaalin välillä on 90 dB. Saman taajuuskaistan tapauksessa maksimivirran suhde LoRa signaaliin on 20 dB tai parempi, jolloin se hylätään. (LoRa Modulation Basics 2015.)

LoRa signaali soveltuu hyvin kaupunki alueille, koska sillä on alhainen taajuusalue ja saa modulaation yhteydessä vastustuskykyä monitie-etenemisen aiheuttamiin signaali häiriöihin ja signaalin heikentymiseen. Edellä mainittujen tekijöiden ansiosta se soveltuu loistavasti käytettäväksi alueille, joissa on luonnon esteitä ja ihmisten rakenteita. (Comparison of LPWAN technologies n.d; LoRa Modulation Basics 2015.)

LoRa saa paremman vastaanotettujen ja hävitettyjen lähetysten suhteen lähetyksissään kuin tavanomaiset FSK järjestelmät. Yhteyksien kantomatkat saadaan nelinkertaisiksi tai vieläkin paremmiksi vastaanotettujen ja hävitettyjen lähetysten suhteen ansiosta, sekä hyvän häiriöiden vastustuskyvyn ansiosta. (LoRa Modulation Basics 2015.)

Doppler-ilmiön takia tuleva muutos aiheuttaa pienen taajuusmuutoksen LoRa pulssissa, josta tulee suhteellisen vähäpätöinen muutos kantataajuuskaistan signaalin ai-

kamuutukseen. Tämän taajuusmuutoksen ansiosta tarve tiukalle referenssi aikalähteelle pienenee. (LoRa Modulation Basics 2015.) Taajuusmuutos vastaanottavan ja lähettävän järjestelmän välillä voi nousta 20 %:iin asti kaistanleveydestä ilman että se aiheuttaa ongelmia. Tämä auttaa laskemaan LoRa lähettimien hintoja, sillä niitä ei tarvitse valmistaa erittäin suurella tarkkuudella. (Augustin ym. 2016.)

LoRa-modulaatio käyttää ortogonaalista Spreading Factor (SF) arvoja, jotka mahdollistavat useamman hajautetun signaalin lähetyksen samanaikaisesti ja myöskin samoilla kanavilla minimaalisilla signaalin herkkyyden heikentymisillä. Eri SF taajuuksilla moduloidut signaalit näkyvät meluna vastaanottavalle kohteelle, ja niitä voidaan kohdella sen mukaisesti. (LoRa Modulation Basics 2015.) SF-arvoja muokkaamalla järjestelmän toimintaa voidaan muokata, kuten pienellä SF-arvolla järjestelmä saa isomman tiedonsiirtonopeuden pienemmällä kantomatalla ja korkeammilla SF arvoilla pienemmän tiedonsiirtonopeuden pitemmällä kantomatalla.

Luontainen ominaisuus LoRa järjestelmälle on sen kyky lineaarisesti hahmoittaa eri taajuudet ja aikavirheet. LoRa on ideaalinen modulaatio tutkajärjestelmille ja sen takia se on ideaalinen järjestelmä, jos toteutetaan reaaliaikaisia paikannusjärjestelmiä. (LoRa Modulation Basics 2015.)

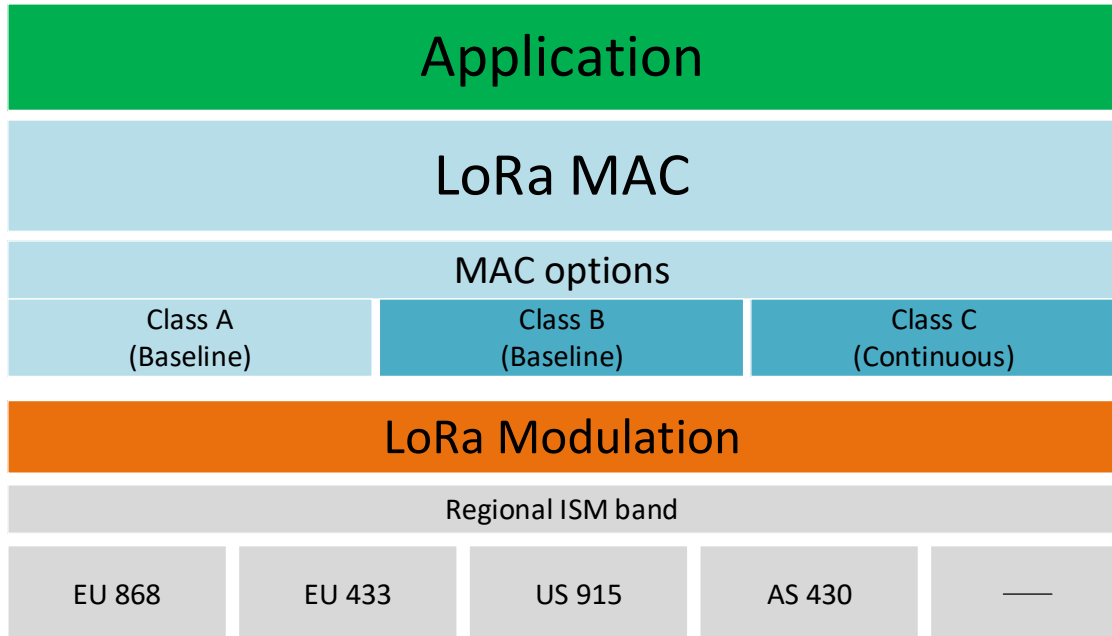
2.2 LoRaWAN

2.2.1 Yleistä

LoRaWAN tuo MAC mekaniikan järjestelmään mukaan mahdollistaen monen päätelaitteen kommunikoinnin yhdyskäytävän kanssa käyttäen LoRa-modulaatiota. MAC protokollalla viitataan OSI-mallin toiseen tasoon. LoRaWAN on avoimeen standardiin pohjautuva järjestelmä, jota LoRa Alliance kehittää, kun taas LoRa on patentoitu tekniikka. (Augustin ym. 2016.)

LoRaWAN määrittelee kommunikaatioprotokollat ja järjestelmäarkkitehtuurin verkolle, kun taas LoRan fyysinen taso mahdollistaa pitkän kantomatkan kommunikaatio

linkit. Protokolla- ja verkkoarkkitehtuurilla on suurin vaikutus, kun määritellään pariston elinaikaa, verkon kapasiteettia, Quality of Service (QoS), turvallisuutta ja minkälaisia sovelluksia verkko tarjoaa. Kuviossa 2 on havainnollistettu LoRaWAN-protokollan toimintaa. (What is LoRaWAN 2015.)

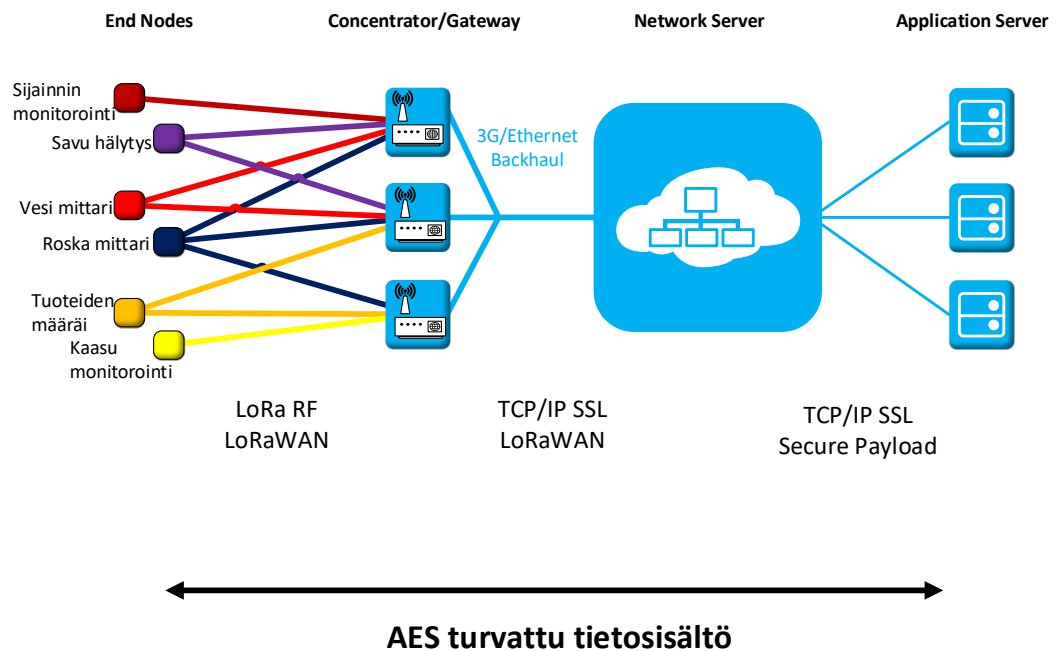


Kuvio 2. LoRaWAN-protokollan toiminta (What is LoRaWAN 2015, muokattu)

2.2.2 Verkkoarkkitehtuuri

Monet käytössä olevat verkkoratkaisut käyttävät mesh-verkkoarkkitehtuuria. Mesh-verkossa yksittäiset solmut lähettävät informaation toisille solmuille lisätäkseen verkon kommunikointikantomatkaa ja solujen kokoja. Tämä lisää kantomatkaa, mutta se myöskin lisää monimutkaisuutta, vähentää verkon kapasiteettia ja vähentää pariston elinaikaa, kun solmut vastaanottavat ja lähettävän eteenpäin tietoa muilta solmuilta, mikä voi olla epäoleellista niille. Pitkänkantomatkan tähtiarkkitehtuuri on järkevin ratkaisu, kun halutaan säilyttää pitkä pariston elinaika, kunhan pitkäkantomatka saavutetaan. (What is LoRaWAN 2015.)

LoRaWAN järjestelmässä solmut eivät ole sidottuja tiettyihin tietoverkko yhdyskäytäviiin. Lähetys LoRaWAN-verkossa tapahtuu kuvio 3 mukaisesti siten, että yksi solmu lähettää halutun datan kaikille kuuluualueella oleville yhdyskäytäviiille, jotka sitten lähettävät datan eteenpäin käyttäen runkoverkkoa. Monimutkaisuus ja älykkyys toteutetaan verkkopalvelimen puolella, joka hallinnoi verkkoa, suodattaa ylimääräiset paketit, suorittaa turvallisuustarkastukset, aikatauluttaa kuittausviestien lähetyksen optimaalisen yhdyskäytävän kautta ja suorittaa Adaptive Data Rate (ADR) määrittelyn. Mikäli solmu on liikkeessä, kuten esimerkiksi paketissa kiinni oleva Global Positioning System (GPS) sensori, yhdyskäytävii ei tarvitse tehdä siirtymistä tukiasemien välillä. (What is LoRaWAN 2015.)



Kuvio 3. LoRaWAN-verkon rakenne (What is LoRaWAN 2015, muokattu)

2.2.3 Pariston elinaika

LoRaWAN-solmut toimivat asynkronisesti, ja kommunikointi tapahtuu ainoastaan silloin, kun niillä on dataa valmiina lähetettäväksi, oli kyseessä tapahtumapohjainen tai

aikataulutettu lähetys. Tämän tyyppin protokollista voidaan myöskin käyttää Aloha-termiä. Mesh-verkossa tai synkronisessa verkossa, kuten matkapuhelinverkossa, solmujen täytyy usein "herätä" ja tarkistaa onko viestejä saapunut. Tämä synkronisointi kuluttaa huomattavat määrät virtaa ja on suurin energian kuluttaja. LoRaWAN järjestelmän pariston elinaikaodote on samalla tasolla, kuin sigfox-järjestelmän, mutta muut teknologiat eivät pärjää elinajanodotteiden kanssa. (What is LoRaWAN 2015.)

2.2.4 Verkon kapasiteetti

Tähtiverkon toimivuuden kannalta on yhdyskäytävällä oltava erittäin suuri kapasiteetti tai kyky vastaanottaa viestejä suurelta määrältä solmuja. Korkea verkon kapasiteetti saavutetaan käyttämällä adaptiivista tiedonsiirtonopeutta ja käyttämällä monikanavaista monimodeemi vastaanotinta yhdyskäytävänä, jotta samanaikaisia viestejä monelta kanavalta voidaan vastaanottaa. Kapasiteettiin vaikuttavat kriittisimmät tekijät ovat rinnakkaisten kanavien määrä, tiedonsiirtonopeus, tietosisällön pituus ja kuinka usein solmut välittävät tietoa. (What is LoRaWAN 2015.)

LoRa on hajaspektri pohjainen modulaatiotekniikka, joten signaalit ovat käytännössä ortogonaalisia toisiinsa verrattuna, kun erilaisia SF arvoja käytetään. Kun SF arvot vaihtuvat efektiiviset tiedonsiirtonopeudet myöskin vaihtuvat. Yhdyskäytävät myöskin käyttävät hyödykseen tätä ominaisuutta ottamalla vastaan monta eri tiedonsiirtonopeutta samalle kanavalle samanaikaisesti. Mikäli solmuilla on hyvä yhteys ja se on lähellä yhdyskäytävää, ei ole mitään syytä miksi sen pitäisi aina käyttää pienintä tiedonsiirtonopeutta ja täyttää vapaana oleva spektri pitempää kuin on pakollista. Muuttamalla tiedonsiirtonopeuksia korkeammaksi lähetyksen ilmassa oloaika lyhenee ja täten avautuu lisää potentiaalisesti tilaa muiden solmujen lähetyksille. Adaptiivinen tiedonsiirtonopeus myöskin optimoi pariston elinajan. Adaptiivinen tiedonsiirto vaatii toimiakseen symmetrisen ylöspäin ja alaspäin menevät yhteydet, joista alaspäin menevät yhteydet vaativat tarpeeksi suuren kapasiteetin. (What is LoRaWAN 2015.)

Nämä ominaisuudet mahdollistava LoRaWAN järjestelmän korkean kapasiteetin ja tekevät verkosta skaalautuvan. Verkko voidaan ottaa käyttöön minimaalisella määrällä infrastruktuuria ja ottamalla lisää yhdyskäytäviä saadaan kapasiteettia lisättyä tarpeen mukaan. Otettaessa lisää yhdyskäytäviä järjestelmään voidaan tiedonsiirtonopeutta muuttaa, jonka avulla vähennetään muihin yhdyskäytäviin kohdistuvaa ylikuulumista ja tällä saadaan skaalattua kapasiteettia 6-8 kertaisesti. (What is LoRaWAN 2015.)

2.2.5 Laitekategoriat

Päätelaitteet palvelevat erilaisia tarkoituksia ja tämän takia niiden vaatimukset ovat erilaiset. Optimoidakseen erilaisten päätelaitteiden ohjelmien profiilit LoRaWAN käyttää niitä varten erilaisia laiteluokkia. Nämä laiteluokat vaihtavat verkon downlink kommunikointi latenssia suhteessa pariston elinajan kanssa. Hallinta tai käyttölaite tyyppisissä sovelluksissa downlink kommunikoinnin latenssi on tärkeä tekijä. (What is LoRaWAN 2015.)

Laiteluokat ovat lajiteltu seuraaviin luokkiin:

- A. Paristolla toimivat sensorit ovat energiatehokkaimpia, joiden täytyy olla tuettuna kaikkien laitteiden toimesta ja alaspäin yhteydet ovat käytettävissä ainoastaan sensorin lähetyksen jälkeen.
- B. Paristolla toimivat käyttölaitteet, joka on energiatehokas, mutta menettää tehokkuuttaan, koska tämä laite saa ylimääräisen lähetyssikunaa, joka lähetetään tietynväliajoin.
- C. Päävirralla toimiva käyttölaite, joka voi virrankulutuksen puolesta kuunnella jatkuvasti ja vastaanotettavilla lähetyksillä ei ole ylimääräistä latenssia.

Luokan A laitteet ovat kaksisuuntaisesti kommunikoivia, missä jokaisen lähetyksen jälkeen seuraa kaksi lyhyttä vastaanottoikkunaa. Päätelaitteet saavat aikataulutetun lähetyspaikkansa omien kommunikointi tarpeidensa mukaan pienellä satunnaisella aikavaihtelulla. Tämän tyyppin laitteet toimivat pienimmällä mahdollisella virrankulutuksella, koska laite kommunikoii mahdollisimman vähän palvelimen kanssa ja niihin tehdyt sovellukset vaativat ainoastaan kommunikaatiota palvelin puolelta sen jäl-

keen, kun päätelaite on ensiksi lähettänyt lähetyksen. Palvelimelta lähetetyt kommunikaatiot joutuvat odottamaan kaikkina muina aikoina seuraavaa aikataulutettua lähetystä päätelaitteelta. (What is LoRaWAN 2015.)

Luokan B päätelaitteet ovat kaksisuuntaisesti kommunikoivia, jotka saavat luokan A satunnaisen lähetysikkunan lisäksi ylimääräisen vastaanottoikkunan aikataulutetuilla väliajoin. Päätelaitteen täytyy synkronisoida aikansa yhdyskäytävän kanssa aikasykronisoidulla beacon-viestillä, jotta se pystyy avaamaan vastaanottoikkunansa oikeaan aikaan. Tämän avulla palvelin tietää milloin päätelaite kuuntelee tulevia lähetyksiä. (What is LoRaWAN 2015.)

Luokan C päätelaitteet toimivat myöskin kaksisuuntaisella kommunikoinnilla ja niissä on maksimaalinen määrä vastaanottoaikoja. Tämän tyyppin päätelaitteilla on melkein jatkuvasti oleva avoin vastaanottoikkuna, joka sulkeutuu ainoastaan lähetyksen yhteydessä. (What is LoRaWAN 2015.)

2.2.6 Tietoturva

LoRaWAN käyttää kahta tasoa turvallisuudessaan, joista toinen on verkolle ja toinen ohjelmistolle. Verkkoturvallisuus varmistaa päätelaitteiden autenttisuuden verkossa, kun taas ohjelmistotason turvallisuus pitää huolen siitä, että käyttäjän tiedot eivät joudu verkon hallinnoinnista vastaavan käsiin. Advanced Encryption Standardin (AES) salausta käytetään avainten vaihdossa, joka käyttää myöskin tunnistukseen IEEE:n kehittämää Extended Unique Identifier-64 (EUI-64) järjestelmää. (What is LoRaWAN 2015.)

LoRaWAN päätelaitteet ja LoRaWAN verkko suorittavan molemminpuolisen autentikoinnin verkkoon liittymisen yhteydessä. Tämä takaa oikeiden ja autentikoitujen laitteiden liittymisen yhtä lailla oikeisiin ja autentikoituihin verkkoihin. LoRaWAN MAC ja ohjelmistojen viestien alkuperä autentikoidaan, eheys turvataan, uudelleen toistoja vastaan suojaudutaan ja suoritetaan salaus. Näillä suojauksilla, sekä molemminpuolisella autentikoinnilla varmistetaan verkon liikenteen koskemattomuus, viestien saapuminen oikeilta laitteilta, salakuuntelijat eivät pysty ymmärtämään viestien sisältöä,

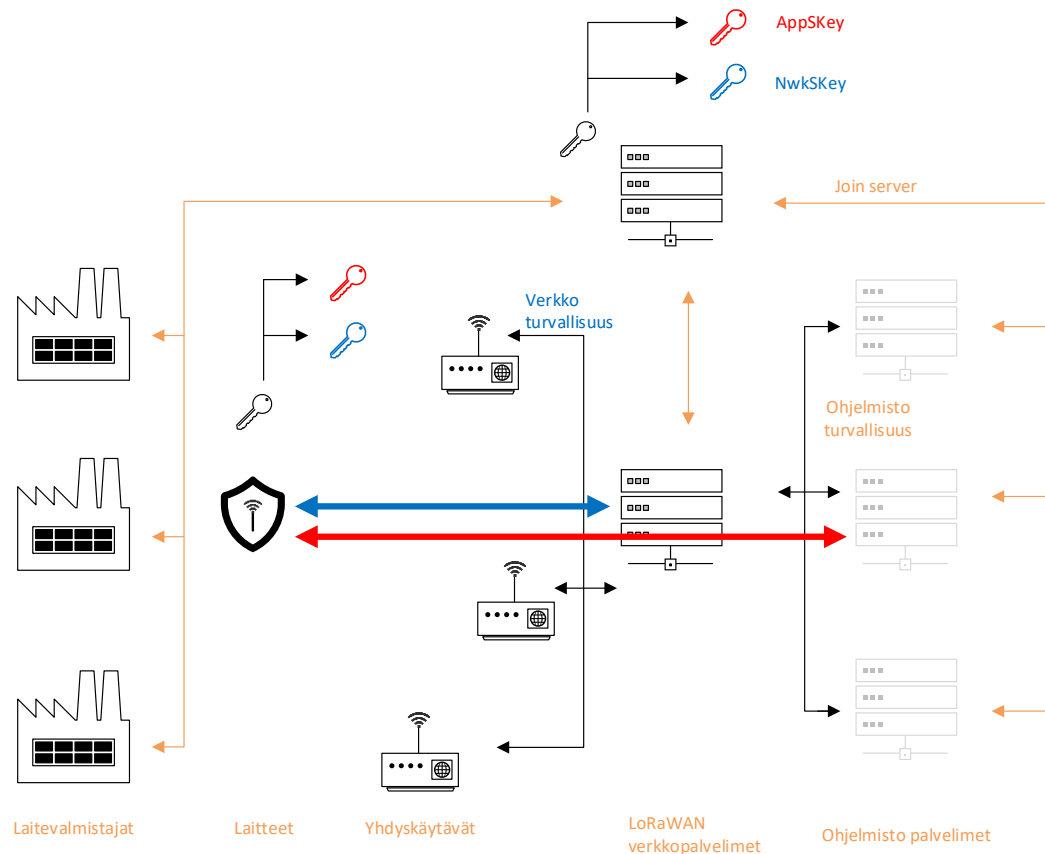
viestejä ei ole saatu kaapattua ja uudelleenlähetettyä eri toimijoiden toimesta. (LoRa Alliance Security Whitepaper 2017.)

Jokainen LoRaWAN laite on personalisoitu uniikilla 128 bittisellä AES-avaimella, jota kutsutaan AppKey nimikkeellä ja globaalilla uniikilla tunnisteella, joka on EUI-64 pohjainen DevEUI, joista molempia käytetään laitteen tunnistus prosessin aikana. EUI-64 tunnisteiden allokointiin tarvitaan Organizational Unique Identifier (OUI) IEEE rekisteröinti auktoriteetilta. Samanlaisesti LoRaWAN verkot tunnistetaan 24 bittisen globaalin uniikin tunnisteiden kautta, joita LoRa Alliance myöntää. (LoRa Alliance Security Whitepaper 2017.)

LoRaWAN ohjelmistojen tietosisällöt salataan aina päästä päähän päätelaitteen ja ohjelmistopalvelimen välillä. Pakettien eheyden suojaus tehdään jokaisen hypyn kohdalla, kuten ensimmäisen hypyn kohdalla, joka menee ilmateitse yhdyskäytävälle, jossa LoRaWAN-protokolla tuo eheyden suojauksen. Verkkopalvelimen ja ohjelmistopalvelimen välisen liikenteen voi turvata käyttäen Hypertext Transfer Protocol Secure (HTTPS) ja virtual private networkia (VPN). (LoRa Alliance Security Whitepaper 2017.)

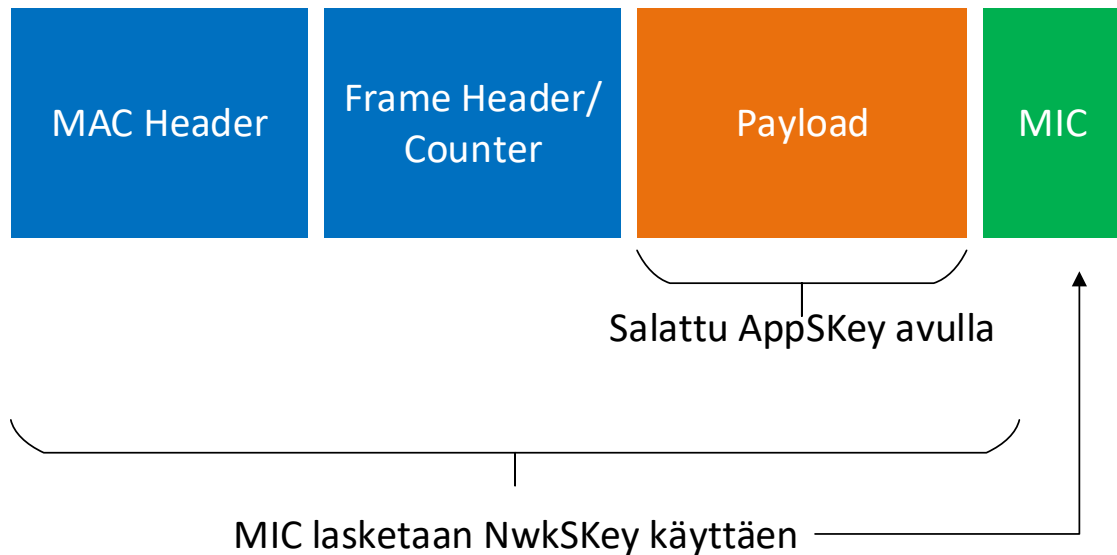
Päätelaitteen liittyessä verkkoon käytetään Over the Air Activation (OTAA) -menetelmää, jonka avulla pystytään todistamaan, että molemmilla osapuolilla on AES128 muotoinen AppKey juuriavain. Tämä todiste saadaan aikaiseksi ajamalla Cipher-based Message Authentication Code (CMAC), jossa käytetään AES-salausalgoritmia tuomaan viestien eheys ja aitous laitteen liittymispyynnön aikana ja taustajärjestelmän palvelimen vastaanottimella. Tämän avulla saadaan kaksi istuntoavainta, joista yhtä käytetään eheyden suojaamiseen ja salaamaan LoRaWAN MAC-komennot ja ohjelmiston hyötykuorma. Tästä käytetään termiä NwkSKey. Toisella avaimella salataan päästä päähän menevä ohjelmiston hyötykuorma. Tästä käytetään termiä AppSKey. NwkSKey jaetaan ympäri LoRaWAN-verkkoa, jotta saadaan todistettua/varmennettua pakettien autenttisuus ja eheys. AppSKey jaetaan sovelluspalvelimelle, jotta salaus/salauksen poisto pystytään suorittamaan ohjelmiston hyötykuormalle. AppKey

ja AppSKey voidaan piilottaa verkon hallinnoijalta ja täten estää ohjelmiston hyötykuorman salauksen poisto. Tätä prosessia havainnollistetaan kuviossa 4. (LoRa Alliance Security Whitepaper 2017.)



Kuvio 4. Salausviestien toiminta (LoRa Alliance Security Whitepaper 2017, muokattu)

Kaikki LoRaWAN-liikenne turvataan kahdella sessioavaimella. Jokainen hyötykuorma salataan käyttäen AES-Counter Mode Encryptionia (CTR), joka on AES-toimintatapa, jossa alustusvektoreiden avulla luodaan laskuri, joiden avulla salataan tietovirtoja (Xian & Tingthanathikul 2004). AES-CTR tuo kehyslaskurin mukanaan, jotta paketti-vastauksia voidaan välttää ja Message Integrity Code (MIC) saadaan järjestelmään mukaan, mikä lasketaan käyttäen AES-CMAC-menetelmää pakettien muuttamisen estämiseksi. Kuviossa 5 on havainnollistettu LoRaWAN-paketin rakennetta ja turvallisuutta. (LoRa Alliance Security Whitepaper 2017.)



Kuvio 5. LoRaWAN paketin rakenne ja turvallisuus (LoRa Alliance Security Whitepaper 2017, muokattu)

AppKey ja siitä derivoidut istunto avaimet ovat jatkuvasti varastoituna LoRa Alliance laitteessa ja niiden suojaus riippuu laitteen fyysisestä turvallisuudesta. LoRaWAN käyttää AESia standardisoidussa CTR tilassa, jonka käytössä on XOR krypto toimitoja. Käytössä on myöskin muita tiloja, kuten Cipher Blocker Chaining (CBC). Tämä vahvistaa AES algoritmin toimintaa käyttämällä uniikkeja AES avaimia, jokaisen lohkon salauksessa. (LoRa Alliance Security Whitepaper 2017.)

AppSKey ja NwkSKey luodaan samasta AppKeysta, joten on mahdollista, että LoRaWAN operaattori voisi samasta AppKeysta derivoida AppSKey arvot ja täten purkaa salauksen. Tämän tilanteen estämiseksi palvelin, joka hallinnoi AppKey varastoa, molemminpuolista todentamista ja avain derivoinnin voi suorittaa taho, joka ei ole hallinnasta vastaavien tahojen valvonnassa. LoRaWAN 1.1 määrittelyt toivat lisää joustavia ominaisuuksia operaattoreiden hallinnointi ominaisuuksien parantamiseksi. Näihin ominaisuuksiin kuuluvat kaksi itsenäistä pääsalauksavainta, joista yksi tehtiin verkolle (NwkKey) ja toinen ohjelmistoille (AppKey). (LoRa Alliance Security Whitepaper 2017.)

Taustajärjestelmän rajapinnat tuovat mukanaan hallinnointi ja tiedon signaloinnin verkkopalvelimille ja ohjelmistopalvelimille. Näiden kriittisten infrastruktuuri elementtien turvaamiseksi käytetään HTTPS ja VPN teknologioita saman tapaisesti, kuin muissakin telekommunikaatio järjestelmissä. (LoRa Alliance Security Whitepaper 2017.)

2.2.7 Alueelliset parametrit

LoRaWAN-määrittelyt vaihtelevat eri alueiden välillä, koska kaikilla alueilla ei käytetä samoja spektrien allokointeja ja säätelyvaatimukset eivät ole samanlaiset. LoRaWAN-määrittelyt Euroopan ja Pohjois-Amerikan alueille ovat määriteltyinä, mutta kaikilla alueilla ei vielä välttämättä ole määrittelyitä tehtynä, joita tekninen komitea LoRa Alliancella tekee. Taulukossa 1 kuvataan alueellisia määrittelyitä. (What is LoRaWAN 2015.)

Taulukko 1. LoRaWAN-yhteyksien alueelliset määrittelyt (What is LoRaWAN 2015)

| | Eurooppa | Pohjois-Amerikka | Kiina | Korea | Japani | Intia |
|-----------------------------------|-----------------|-----------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| Taajuuskaista | 867-869MHz | 470-510MHz | 470-510MHz | 920-925MHz | 920-925MHz | 865-867MHz |
| Kanavat | 10 | 64 + 8 + 8 | Teknisen komitean määrittelyt | Teknisen komitean määrittelyt | Teknisen komitean määrittelyt | Teknisen komitean määrittelyt |
| Kanavan kaistanleveys Ylös | 125/250kHz | 125/500kHz | | | | |
| Kanavan kaistanleveys Alas | 125kHz | 500kHz | | | | |
| TX virta ylös | +14dBm | +20dBm (+30dBm sallittu) | | | | |
| TX virta alas | +14dBm | +27dBm | | | | |
| SF ylös | 7-12 | 7-10 | | | | |
| Datan nopeus | 250bps-50kbps | 980bps-21.9kbps | | | | |
| Hyöty häviö suhde ylös | 155dB | 154dB | | | | |
| Hyöty häviö suhde alas | 155dB | 157dB | | | | |

Euroopassa LoRaWAN määrittelee kymmenen kanavaa, joista kahdeksan ovat monitiedonsiirto kanavia, joiden nopeudet vaihtelevat 250bps – 5.5 kbps välillä, yksi kanava on varattu korkealle tiedonsiirtonopeudelle, joka toimii 11kbps nopeudella ja viimeinen on varattu FSK kanava 50kbps nopeudella. European Telecommunications Standards Institute (ETSI) sallii maksimissaan +14dBm virran huipputehon, jossa on poikkeuksena G3 taajuuden +27dBm virran huipputeho. ETSI määrittelee käyttöjaksosoihin rajoituksia, mutta enimmäislähetys- tai kanavien viiveaikoihin rajoituksia ei ole. (What is LoRaWAN 2015.)

ISM-taajuudet Pohjois-Amerikassa ovat 902-928 MHz alueella. LoRaWAN määrittelee 64, 125 kHz uplink kanavaa, jotka alkavat 902.3 MHz ja jatkuvat 914.9 MHz asti, 200 kHz korotusten välein. Tämän lisäksi on kahdeksan 500 KHz ylöspäin menevää tiedonsiirtokanavaa, jotka toimivat 1.6 MHz korotusten välein alkaen 903 MHz ja loppuvat 914.9 MHz. Kahdeksan alaspäin menevää tiedonsiirtokanavaa on jaettu 500 kHz välein alkaen 923.3 MHz jatkuen 927.5 MHz asti. Maksimaalinen virran huipputeho Pohjois-Amerikassa 902-928 MHz taajuuskaistalla on +30dBm, mutta useimmille laitteille riittää +20dBm. Federal Communications Commission (FCC) laatimien sääntöjen mukaan käyttöjaksorajoituksia ei ole, mutta kanavien viiveajat eivät saa ylittää 400msec yhtä kanavaa kohden. (What is LoRaWAN 2015.)

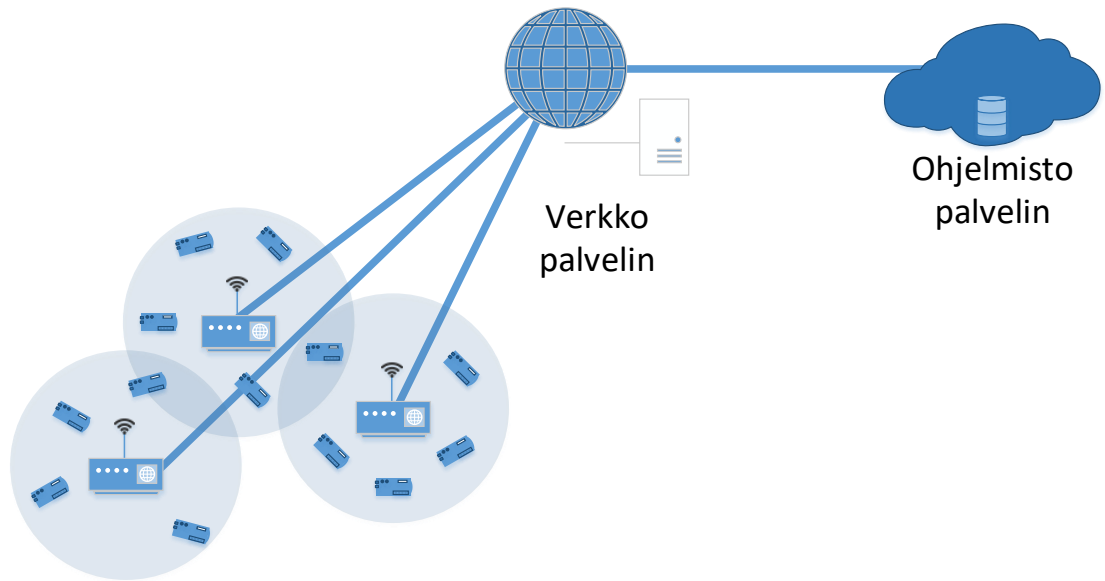
FCC määrittelee taajuushyppelyn vaatimukset, jossa vaaditaan yli 50 kanavan käyttöä tasaisesti ISM-taajuuskaistalla. LoRaWAN on määritelty käyttämään yli 50 kanavaa, jotta se voi käyttää hyväkseen käytettävissä olevaa spektriä ja sallia maksimaalisen huipputehon. LoRa-modulaatio kelpuutetaan digitaaliseksi modulaatiotekniikaksi, joten sen ei tarvitse noudattaa kaikkia taajuushyppelyn vaatimuksia, joita FCC Hybridi moodi vaatii. Hybridi moodissa maksimaalinen huipputeho on rajoitettu +21dBm ja ainoastaan kahdeksan kanavaa 64 ylöspäin menevästä tiedonsiirtokanavasta ovat käytössä Hybridi tilassa. (What is LoRaWAN 2015.)

2.3 The Things Network

The Things Network (TTN) on avoimen lähdekoodin periaatteisiin pohjautuva infrastruktuuri ratkaisu LoRaWAN toteutuksille, jonka tarkoituksena on tarjota ilmainen LoRaWAN taustajärjestelmä, johon LoRaWAN yhdyskäytävät ja ohjelmistot liitetään. TTN perustettiin vuonna 2015 Wienke Giezeman ja Johan Stokking toimesta. TTN projektia kehitetään vapaaehtoisilla lisäyksillä ympäri maailman. (The Thing Network n.d.)

2.3.1 Verkkoarkkitehtuuri

TTN tarjoaa taustajärjestelmän, joka on vastuussa laitteiden ja ohjelmistojen IoT tietojen reitityksestä. Tyypillinen IoT-verkko vaatii yhdyskäytävän sillaksi tietyn radio protokollan ja internetin välille. Tapauksissa missä laite itsessään tukee Internet Protokolla (IP) pinoa yhdyskäytävä joutuu ainoastaan välittämään eteenpäin paketit internettiin. Ei IP protokollaa käyttävät järjestelmät, kuten LoRaWAN vaativat jonkinlaisen reitityksen ja prosessoinnin ennen kuin viestit voidaan lähettää vastaanottavalla ohjelmalle. TTN toimii yhdyskäytävien ja ohjelmistojen välissä, pitäen huolen reitityksestä ja prosessointi vaiheista. Tätä havainnollistetaan kuviossa 6. (Network Architecture n.d.)



Kuvio 6. TTN-reititys (Network Architecture n.d, muokattu)

TTN visiona on ollut reititysjärjestelmä, joka toimii hajautetulla ja jaetulla tavalla. Kaikkien kiinnostuneiden ryhmien pitäisi pystyä pystyttämään omat tietoliikenneverkot ja oman osan taustajärjestelmää, sallien heidän osallistumisensa globaaliin kommunikaatioverkkoon. Verkko on hajautettu moneen komponenttiin, jota havainnollistetaan kuviossa 7. Asian selkeyttämiseksi komponentit esitetään vain kerran, vaikka on mahdollista olla yksi moneen tai monta moneen yhteyksiä komponenttien välillä. (Network Architecture n.d.)

TTN tavoittelee joustavuutta käyttöönottovaihtoehtojensa suhteen. Suositelluin tapa on ottaa yhteys julkiseen yhteisöverkkoon, jonka tarjoaa The Things Network Foundation ja heidän partnerinsa. Tässä tapauksessa ohjelmat ottavat yhteyttä julkisen yhteisöverkon käsittelijään, normaalisti käyttäen Message Queuing Telemetry Transport (MQTT) Application Programming Interfacea (API). (Network Architecture n.d.)

Oletus käsittelijän toteutuksessa julkaistaan JSON representaation ylöspäin menevässä viestistä MQTT välittäjän aiheeseen. Tämän avulla ohjelmiston tarvitsee vain tilata sama MQTT aihe ja prosessoida tiedot sitä kautta. Yksityisten verkkojen käyttöönotto on myöskin mahdollista käyttämällä kaikkia näitä samoja komponentteja omassa ympäristössään. Tällä tavalla kaikki tiedot pysyvät oman yksityisen ympäristön sisällä, mutta on kuitenkin mahdollista käyttää TTN tarjoamaa tilienhallintapalvelinta autentikointiin ja auktorisointiin. (Network Architecture n.d.)

Hybridi mallien käyttöönotot mahdollistuvat tulevaisuudessa. Helpoin toteutus tälle tavalle on käyttää omaa käsittelijä antaen sen hoitaa salauksen ja salauksenpoistamisen viesteissä. Monimutkaisempi toteutus tällaiselle järjestelmälle olisi käyttää yksityistä verkkoa, joka jakaa tietoja julkisen verkon kanssa. Tämän toteutuksen toimintaan saamiseksi tarvittaisiin yksityinen reititin, jonka täytyy ottaa yhteyttä julkiseen välittäjään ja toisinpäin. Näissä tapauksissa yksityinen verkko voi purkaa julkisen liikenteen yhteisöverkkoon ja käyttää julkista yhteisöverkkoa takaamaan yhteyksien toimivuus. Edellä mainitut menetelmät eivät ole vielä mahdollisia 3.3.2019 voimassa olevalla toteutuksella taustajärjestelmästä. (Network Architecture n.d.)

2.3.2 Keskeisimmät toiminnallisuus

TTN tarjoaa LoRaWAN-verkkopalvelimen, koska LoRaWAN suurin toiminnallisuus tapahtuu verkon puolella laitteiden ollessa yksinkertaisia ja minimaalisia lähestymistävaltaan. Taustajärjestelmät, joita myöskin kutsutaan verkkopalvelimiksi ovat vastuussa suurimmasta osasta logiikasta, jota järjestelmässä on. LoRaWAN suunniteltiin

keskitetyksi arkkitehtuuriksi teleoperaattoreille, jotta pystyttäisiin käyttämään jaettua infrastruktuuri järjestelmää, siihen täytyi lisätä muutamia asioista. TTN taustajärjestelmästä pystytään erottelemaan erilaisia ydintoiminnallisuuksia. (Network Architecture n.d.)

Ensimmäiseksi sieltä löytyvät yhdyskäytäviin liittyvät toiminnallisuudet, kuten aikataulutukset ja yhdyskäytävien käyttöaikojen hallinnointi. Aikatauluja tarvitaan, koska yhdyskäytävät voivat tehdä yhden lähetyksen samanaikaisesti. Käyttötietojen avulla pystytään jakamaan kuormitus tasaisesti eri yhdyskäytävälle ja tämän avulla noudatetaan Euroopan käyttöjaksovaatimuksia. Yhdyskäytävien tilanteen monitorointi on myös yksi tärkeä ominaisuus. (Network Architecture n.d.)

Toinen tärkeä asia on laitteisiin liittyvät toiminnallisuudet, jotka hallinnoivat laitteiden tiloja verkossa. Laitteosoitteet eivät ole uniikkeja, joten verkon täytyy seurata laitteiden osoitteita, jotta järjestelmä pystyy kartoittamaan oikeat viestit oikeille laitteille ja ohjelmistoille. Lisäksi verkon täytyy seurata turvallisuusavaimia ja kehyslaskureita. Tulevaisuudessa pitäisi tulla verkon käytön seuranta solmuille. (Network Architecture n.d.)

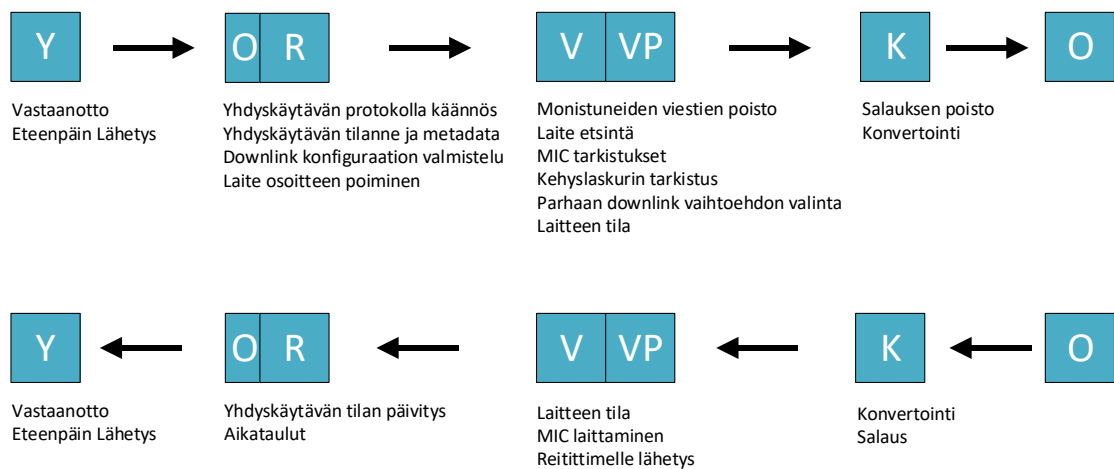
Ohjelmistoon liittyviä toiminnallisuuksia on hyvä havainnollistaa käyttäen välittäjiä ja käsittelijöitä, sillä niiden täytyy tietää mihin palvelinliikenne täytyy lähettää tietyille ohjelmistoille. Käsittelijöiden täytyy tietää, kuinka tulkita binääritietoja ja toimia silta korkeamman tason protokollille, kuten Advanced Message Queuing Protocol (AMQP) ja MQTT tapauksissa. (Network Architecture n.d.)

Viimeisenä päästään kaikista tärkeimpään toiminnallisuuteen, joka on TTN toimivuus jaettuna verkkona ja sitä tukeville toiminnallisuuksille. Palveluiden löytämisen toiminnallisuus auttaa komponentteja määrittelemään mihin liikenne pitäisi reitittää. Tämänhetkinen toteutus on tehty keskitetyllä Discovery-palvelimella antaen Things Network Foundationille hallinnan siitä mitkä komponentit pääsevät julistamaan tiettyjä palveluita. (Network Architecture n.d.)

2.3.3 Velvollisuuksien erottaminen

Saadakseen erilaiset taustajärjestelmän komponentit mahdollisimman irralleen toisistaan tehdään selvät erottelut vastuualueiden suhteen jokaiselle komponentille. Yleinen idea tässä on reitittimen vastuu kaikesta yhdyskäytävän liittyvistä toiminnallisuuksista ja aluekohtaisista tiedoista. Välittäjä käsittelee useita laiteosoitteita ja on vastuussa oikean käsittelijän löytämisestä, jolle lähettää viestit. Verkkopalvelin on vastuussa yksittäisten laitteiden tilan pitämisestä. Käsittelijä on vastuussa salauksesta, salauksen poistosta, viestien muuntamisesta ja viestien lähettämisestä eteenpäin laitteille. (Network Architecture n.d.)

TTN taustajärjestelmä on toteutettu edellä mainittujen velvollisuuksien erottamisen pohjautuvana. Kaikilla komponenteilla on suuri määrä korkean tason velvollisuuksia ja sen on suoritettava useita tehtäviä prosessoidessaan ylöspäin/alaspäin menevät tietoliikenne viestit. Yleiskatsaus tästä prosessista on nähtävä alla olevassa kuviossa 8. (Network Architecture n.d.)



Y = Yhdyskäytävä
 O = Ohjelmisto
 R = Reititin
 V = Välittäjä
 K = Käsittelijä

Kuvio 8. Komponenttien velvollisuudet (Network Architecture n.d, muokattu)

2.3.4 Yhdyskäytävän protokolla käännökset

Yhdyskäytävän saadessa viestin LoRa lähetykseltä se tiivistetään ja lähetetään eteenpäin TTN järjestelmään internetin ylitse. Moni yhdyskäytävä käyttää samaa referenssi yhdyskäytävä protokollaa, mutta vaihtoehtoisia protokollia on kehitetty tietyille taustajärjestelmän ratkaisuille. TTN on myöskin kehittämässä omaa yhdyskäytävä protokollaa, joka sopii paremmin TTN järjestelmälle, kuin referenssi protokolla turvallisuuden ja pääsyn hallinnan suhteen. (Network Architecture n.d.)

Monilla yhdyskäytävä protokollilla on sama rakenne. Yhden tai useamman viestin saadessaan niiden binäärilasti lähetetään eteenpäin taustajärjestelmälle yhdessä metadatan kanssa, kuten vastaanotetun signaalin voimakkuus (RSSI) ja signaali-kohinasuhde (SNR). Säännöllisin väliajoin yhdyskäytävä myöskin lähettää tilapäivityksiä yhdyskäytävästä itsestään, kuten Global GPS koordinaatit, vastaanotettujen ja lähetettyjen pakettien määrän. (Network Architecture n.d.)

TTN järjestelmään pystyy yhdistämään monen eri valmistajan yhdyskäytävillä, joissa käytetään eri protokollia. Tässä onnistuakseen taustajärjestelmässä tehdään mahdollisimman generiseksi siltoja hyväksikäyttämällä, jotka kääntävät eri valmistajien yhdyskäytävissä olevat protokollat siihen protokollaan mitä sisäisesti käytetään taustajärjestelmässä. (Network Architecture n.d.)

Yhdyskäytävien GPS koordinaatisto voi olla erittäin tärkeää ohjelmistoille. Tämän takia taustajärjestelmä varastoi viimeisimmät tilannepäivitys viestit yhdyskäytävältä ja lisää GPS tiedot metadataan jokaiselle ylöspäin menevälle viestille. (Network Architecture n.d.)

2.3.5 Alaspäin menevän tietoliikenteen konfigurointi

LoRaWAN yhdyskäytävien lähettämät vastaukset saatuihin viesteihin vaihtelevat maantieteellisten alueiden mukaan. Reititin on vastuussa kaikesta yhdyskäytävään liittyvistä asioista ja aluekohtaiset parametreista, joten reitittimen täytyy määritellä, kuinka vastausviestit voidaan lähettää laitteelle. Jokaisen ylöspäin menevän viestin

jälkeen tulee 2 vastaanottojaksoa, joista yksi tulee yhden sekunnin jälkeen saapuvasta viestistä ja toinen kahden sekunnin jälkeen. Tämän takia jokaista yhdyskäytävää kohden, jotka vastaanottavat viestin joutuu reititin rakentamaan alaspäin menevälle liikenteelle kaksi konfiguraatiota. (Network Architecture n.d.)

Parhaan vaihtoehdon valitsemisen takia reititin joutuu lisäksi laskemaan pisteytyksen molemmille vaihtoehdoille. Tähän pisteytykseen vaikuttaa muutama tekijä. Tällä hetkellä siihen vaikuttavat ilmassa oloaika, signaalin voimakkuus, yhdyskäytävän käyttöaste ja entuudestaan varatut lähetyksen. Mikäli alaspäin menevät viestit varataan yhdyskäytävälle, jolla on parempi signaalin voimakkuus, saadaan paremmat mahdollisuudet sille, että solmu saa alaspäin tulevan lähetyksen oikein. (Network Architecture n.d.)

Viestien ilmassa oloajan ja yhdyskäytävän käyttöasteen yhdistelmää käytetään optimoimaan verkkoa kokonaisuudessaan. Jokainen lähetyksen estää yhdyskäytävän vastaanottajat joksikin aikaa, joten on parempi lähettää viestit lyhyemmässä ajassa. Tähän pystyäkseen alaspäin lähetettävissä viesteissä suositaan nopeampia tiedonsiirtonopeuksia. Yhdyskäytävän käytöllä ilmaistaan kuinka paljon aikaa yhdyskäytävä käyttää viestien vastaanottamiseen. Yhdyskäytävät, jotka saavat paljon käyttöä, ovat verkolle paljon tärkeämpiä kuin vähemmällä käytöllä olevat yhdyskäytävät. Sen takia pienemmällä käytöllä olevia yhdyskäytäviä suositaan alaspäin menevien viestien lähetyksissä ja isommalla käytöllä olevat yhdyskäytävät voivat keskittyä ylöspäin menevien viestien vastaanottamiseen. (Network Architecture n.d.)

Ensimmäinen asia mitä reitittäessä pakettia tarvitaan, on laitteen osoite. Tästä käytetään termiä DevAddr ja se on ei uniikki 32-bit osoite, joista 25 bittiä voidaan määrittellä verkko-operaattorin toimesta. TTN on valinnut liikenteen jakamiseensa laiteosoite prefiksin pohjalta. Jokainen välittäjä julkaisee tietyn määrän laiteosoite prefiksejä löytöpalvelun kanssa. Nämä prefiksit toimivat samalla tavalla, kuin IP-osoite alueiden julkaisu Border Gateway Protocol (BGP) tapauksessa. Verkkokomponentit voi-

vat ajoittain hakea listan välittäjistä ja niiden julkaisemista prefikseistä. Viestit lähetetään kaikille välittäjille, jotka julkaisevat prefiksin, joka sopii viestissä tulleeeseen laiteosoitteeseen. (Network Architecture n.d.)

2.3.6 Monistuneiden lähetysten poistaminen

LoRaWAN on pitkän kantomatkan radio protokolla, joten on erittäin todennäköistä, että lähetettävät viestit tulevat vastaanotetuiksi monessa yhdyskäytävässä. Tämä tarkoittaa sitä, että taustajärjestelmän täytyy suorittaa jonkinlainen monistuneiden viestien poisto pystyäkseen lähettämään viestit sitä haluavalle ohjelmalle vain kerran. Tämä ei tarkoita sitä, etteivätkö monistuneet lähetykset olisi tärkeitä. Näissä lähetyksissä voi olla tärkeitä metadata tietoja. Esimerkiksi näistä lähetyksistä voidaan saada viestin vastaanottaneen yhdyskäytävän sijainti, vastaanottoaika ja signaalin voimakkuus, joiden avulla on mahdollista määritellä viestin lähettäneen laitteen sijainti. (Network Architecture n.d.)

Monistuneiden lähetysten poistaminen ei ole helppoa jaetussa verkossa, mutta sitä asiaa voi helpottaa, mikäli käytössä on vain yksi palvelin. Sen takia TTN päätyi tekemään abstraktion monistuneisuuden poisto implementaation päälle. Tämän avulla on helppoa vaihtaa implementaatio sopivammaksi käyttötapaus kohtaisesti samalla pitäen samat rajapinnat. (Network Architecture n.d.)

Nykyinen implementaatio taustajärjestelmästä poistaa monistuneet ylöspäin menevät viestit paikallisesti käyttäen message-digest 5 (MD5) algoritmin tarkistus summaa saaduista hyötykuormista. Viestien hyötykuormat kaikilla monistuneilla viesteillä on sama ja mahdollisuudet sille, että poistaessa monistuneita lähetysä tulisi eri viesti, jolla olisi sama hash arvo on erittäin pieni. (Network Architecture n.d.)

Yhdyskäytävät voidaan yhdistää mihinkä tahansa liityntäverkkoon. Liityntäverkot ovat käyttäjä verkkoja, jotka yhdistävät tilaajat tiettyihin palveluntarjoajiin ja puhelinoperaattoreiden verkon kautta muihin verkkoihin, kuten internettiin (Rouse 2014). Yleisempiä esimerkkejä ovat langallinen Ethernet yhteys, WiFi tai jopa General Packet Radio Service (GPRS) yhteys, joilla saadaan yhteydet internettiin. Vaikka radioaallot

kulkevat valonnopeudella verkon viive aiheuttaa sen, että monistuneet viestit eivät saavu välittäjälle samanaikaisesti. Kerätäkseen jokaisen yhdyskäytävän lisäämät metadatat saapuneista viesteistä, välittäjä joutuu hetken pitämään monistuneita viestejä puskurimuistissa. Tässä aikajaksossa pyritään keräämään mahdollisimman monta monistunutta viestiä ilman, että vastaus viestin lähettävä ohjelmisto menettäisi vastaanotto ikkunansa, joka on auki yhden sekunnin jälkeen lähetyksestä. (Network Architecture n.d.)

TTN tekemät mittaukset ovat näyttäneet, että keskimääräinen viive ensimmäinen ja viimeisen monistuneen viestin välillä ovat hieman alle 100 ms, kun suurin viive on noin 300 ms. Näiden tutkimusten perusteella monistuneiden viestien poistamiselle on määritelty 200ms. Mittaukset joidenka pohjalta nämä määrittelyt on tehty ovat aikajaksolta, milloin yhdyskäytäviä ei vielä ollut tiheästi laitettu, joten tulevaisuudessa nämä määrittelyt voivat muuttua. (Network Architecture n.d.)

2.3.7 Laitteen ja ohjelmiston etsintä

Koska laiteosoitteet eivät ole uniikkeja on pakko määritellä tarkalleen mikä laite viestin lähetti ja mihinkä ohjelmaan se kuuluu. Tässä onnistuakseen taustajärjestelmän on suoritettava salausviestien eheyskoodien (MIC) tarkistukset, jokaiselle samalla laiteosoitteella olevalle laitteelle. Tämä saadaan tehtyä välittäjällä, joka pyytää listan laitteista osoitteineen verkkopalvelimelta ja tarkistaa voidaanko MIC varmistaa käyttäen verkko session avainta. Mikäli nämä eivät täsmää viesti tiputetaan. (Network Architecture n.d.)

2.3.8 Kehyslaskurin tarkistus

Kehyslaskuri LoRaWAN viesteissä on turvallisuus toimenpide, jota käytetään huomamaan toistohyökkäykset. MIC varmennuksen jälkeen välittäjä tarkistaa kehyslaskurin paikkansapitävyyden. Kehysten laskettu määrä voi ainoastaan kasvaa, joten mikäli viesteissä tulee pienempi laskettu kehysten määrä vastaan, kuin viimeisin tiedetty

kehysten määrä, täytyy viesti tiputtaa. Lisäksi välittäjän täytyy varmistaa, että viimeimmän tiedetyn kehysten määrän ja viestissä tuleva kehysten määrän välillä ei ole liian isoa väliä. (Network Architecture n.d.)

LoRaWAN tukee molempia 16-bit ja 32-bit kehyslaskureita. Viestin otsikko kenttään kuitenkin laitetaan vain 16 vähiten merkitsevää bittiä. Tämän takia taustajärjestelmän täytyy pitää silmällä koko 32-bit kehyslaskuria ja käyttää sitä 16-bit laskurin sijaan, joka tulee viestin mukana. (Network Architecture n.d.)

2.3.9 Metadata keräys

Kaikkien tarkistusten mentyä läpi välittäjä voi aloittaa viestin prosessoinnin. Ensimmäisenä välittäjä yhdistää monistuneet viestit, jotka ovat saapuneet eri reitittimiltä ja yhdyskäytäviltä. On tärkeää tehdä ero metadatan kanssa, jotka ovat samat jokaiselle yhdyskäytävälle, jotka vastaanottavat monistuneet viestin ja jokaiselle vastaanotolle omat uniikit metadatat. Esimerkiksi taajuus, modulaatio ja tiedonsiirtonopeus ovat samoja kaikilla yhdyskäytävillä, joten nämä täytyy lähettää eteenpäin ainoastaan kerran. Toisaalta signaalinvoimakkuus, vastaanotto aika ja GPS koordinaatit jokaiselle yhdyskäytävälle pitää myöskin olla mukana viestiä lähettäessä eteenpäin. Tässä tapauksessa myöskin eri alaspäin menevien yhteyksien konfiguraatio tiedot pitää myöskin kerätä, jotta seuraavat toimenpiteet voidaan suorittaa optimaalisesti. (Network Architecture n.d.)

2.3.10 Parhaan alaspäin menevän yhteyksien asetusten valinta

Välittäjän täytyy valita parhaat asetukset alaspäin menevään viestiin vastatessaan siihen. Välittäjällä ei ole tietoja yhdyskäytävästä, joka vastaanotti kyseisen viestin, joten tässä onnistuminen on erittäin vaikea tehtävä. Tämän takia reititin on jo valmiiksi laskenut pisteet jokaiselle alalinkki konfiguraatiolle. Mikäli tämä piste laskelma on tehty standardisoidulla tavalla, välittäjän täytyy vain järjestellä kaikki mahdolliset alalinkki vaihtoehdot ja käyttää niistä parasta. (Network Architecture n.d.)

2.3.11 Laite tilat ja MAC komennot

Ennen ylöspäin menevän viestin lähettämistä eteenpäin käsittelijälle se täytyy lähettää verkkopalvelimelle, jotta laitteen tila voidaan päivittää. Verkkopalvelin myöskin lisää alaspäin meneviin viesteihin mallipohjan. Tätä mallipohjaa voi käsittelijä käyttää lähettäessään alaspäin meneviä viestejä takaisin laitteelle. Se sisältää kaikki vaaditut arvot, kuten kehyslaskurin, viestin tyyppin ja asetus liput, jotta käsittelijän täytyy ainoastaan lisätä ohjelmisto kuorma viestiin mukaan. Lisäksi tämän avulla verkkopalvelin saa mahdollisuuden lisätä MAC komentoja viestii. Tätä voi havainnollistaa esimerkiksi, jossa viestin vastaanottaneiden yhdyskäytävien määrä ja niiden signaalinvaihtamisen perusteella verkkopalvelin voi listata MAC komentoja, jotka ohjeistavat laitetta lähettämään korkeammalla tiedonsiirtonopeudella. (Network Architecture n.d.)

2.3.12 Viestien salauksen poistaminen

Viestit ovat päästä päähän salattuja, joten taustajärjestelmä on myöskin vastuussa viestien salauksen poistosta. Joissakin tapauksissa ohjelmiston omistaja ei halua TTN olevan vastuussa tästä. Tämän takia viestien salauksen poistaminen on määritelty erilliselle komponentin käsittelijälle antaen ohjelmiston omistajalle mahdollisuuden käyttää omaa käsittelijää yksityisessä ympäristössään. (Network Architecture n.d.)

2.3.13 Hyötykuorman konversio

Salauksen poiston jälkeen käsittelijän on mahdollista dekoddata ja muuntaa hyötykuorma formaattiin, johonka ohjelmisto pääsee helposti käsiksi. Oletus käsittelijän implementaatio sisältää nämä niin sanotut hyötykuorma funktiot. Nämä funktiot ovat yksinkertaisia JavaScript funktioita, joita voidaan käyttää dekoddaamaan, muuntamaan ja tietojen varmentamiseen. Dekooderin avulla dekoddataan binaariset hyötykuormat sopivampaan muotoon. (Network Architecture n.d.)

Vaihtoehtoinen kääntäjä voi muuntaa arvot dekoodatusta JSON objektista. Tämä voi esimerkiksi olla muunnos jännitteestä oikeaksi arvoksi tai lämpötilan mittaukset Celsius arvoista Fahrenheit arvoihin. Vaihtoehtoista validaattoria voidaan käyttää tarkistamaan tietojen aitoutta ja poikkeukselliset tiputukset. TTN on tulevaisuudessa tuomassa hyötykuorma funktioita, jotka voivat muuttaa käyttäytymistään viestin FPort kentässä ja tämän avulla TTN yhteisö pystyy määrittelemään standardisoidun tiedon salaus formaatin jokaiselle FPortille. Esimerkkinä tästä voidaan tehdä standardisoitu formaatti sääasemien tietojen lähetyksille. (Network Architecture n.d.)

2.3.14 Alaspäin menevä liikenne

Kun ylöspäin menevä viesti on julkaistu MQTT:lle käsittelijä määrittelee, onko tarpeen lähettää vastaus laitteelle alaspäin menevällä viestillä. On olemassa kolme tilannetta milloin alaspäin menevä viesti pitää lähettää. Ensimmäinen tapaus on silloin, kun ohjelmistolla on hyötykuorma valmiina lähetettäväksi laitteelle. Tässä tapauksessa hyötykuorma lisätään vastaus mallipohjaan, jonka verkkopalvelin loi. Toinen tapaus on silloin, kun ylöspäin menevä viesti vaatii varmistuksen. Huolimatta alaspäin menevän hyötykuorman valmiudesta vastaus viesti täytyy lähettää. Kolmannessa tapauksessa verkkopalvelimen täytyy lähettää MAC komento laitteelle. Mikäli tämä huomataan käsittelijä voi päättää, totellaanko verkkopalvelinta, vaikka nykyinen toteutus tottelee verkkopalvelimen pyyntöjä. Samalla tavalla, kuin ylöspäin menevissä viesteissä käsittelijä on vastuussa viestien hyötykuorman salauksesta. (Network Architecture n.d.)

Mikäli alaspäin menevä hyötykuorma ei ole saatavilla käsittelijä voi valita odottaa pieni hetki, että ohjelmisto saa valmisteltua alaspäin menevän viestin, jonka se tekee juuri saadun ylöspäin tulleen viestin perusteella. Määräajan kuluttua käsittelijän täytyy lähettää alaspäin menevä viesti takaisin välittäjälle. (Network Architecture n.d.)

Sen jälkeen kuin välittäjä on vastaanottanut alhaaltapäin tulevan viestin käsittelijältä se lähettää viestin verkkopalvelimelle, mikä päivittää laitteen tilan tietokannassa ja

luo viestille MIC koodin. Tämän jälkeen välittäjä lähettää alhaaltapäin tulleen viestin eteenpäin reitittimelle, joka on vastuussa yhdyskäytävästä, joka on lähettänyt alhaaltapäin tulleen viestin. (Network Architecture n.d.)

Reititin on vastuussa yhdyskäytävän aikataulun hallinnasta. Koska useimmilla yhdyskäytävillä on tilaa vain yhdelle alaspäin menevälle viestille, puskurimuistissaan reitittimen täytyy laittaa suunnitellut viestit puskurimuistiinsa viimeiseen hetkeen asti ja lähettää jokainen viesti juuri ajoissa yhdyskäytävälle. (Network Architecture n.d.)

2.4 ThingsBoard

ThingsBoard on avoimeen lähdekoodin periaatteisiin pohjautuva IoT-alusta, jolla tietoja kerätään, prosessoidaan, visualisoidaan ja laitteita hallinnoidaan. Tällä mahdollistetaan laitteiden yhteydet standardisoiduilla IoT-protokollilla, jotka tässä tapauksessa ovat MQTT, Constrained Application Protocol (CoAP) ja HTTP. Alusta myöskin tukee pilviratkaisuja ja paikallisia ratkaisuja. (What is ThingsBoard? n.d.)

Järjestelmän keskeisimmät ominaisuudet:

- Määrittelee laitteiden, voimavarojen ja asiakkaiden suhteet toistensa kanssa.
- Kerää ja visualisoi tietoja laitteilta ja muilta voimavaroilta.
- Analysoi tulevat telemetriatiedot ja määrittelee hälytykset niiden pohjalta.
- Laitteiden kontrollointi etänä käyttäen Remote Procedure Callsia (RPC).
- Työnkulkujen rakentaminen käyttäen laitteiden elinaika tapahtumia, Representational State Transfer (REST) API tapahtumia ja RPC pyyntöjä.
- Dynaamisten ja reagoivien kojelautojen suunnittelu, joissa pystyy näyttämään telemetriatietoja ja laitteidentietoja.
- Sääntöketjujen tekeminen, joiden avulla pystyy määrittelemään käyttökohteiden mukaiset säännöt.
- Laitteiden tietojen puskeminen toiseen järjestelmään. (What is ThingsBoard? n.d.)

2.5 ThingsBoard IoT Gateway

ThingsBoard IoT Gateway on laitteiden integraatioon tehty ohjelmisto, jonka avulla pystytään yhdistämään IoT-laitteita ThingsBoard järjestelmään. Järjestelmä on Java

pohjainen, mutta eroaa kuitenkin samanlaisista projekteista, joissa on Open Service Gateway Iniativen (OSGI) teknologioita käytössä. Järjestelmän idea on jotakuinkin samanlainen kuin microservices arkkitehtuurissa. Ohjelmisto kehittäjät voivat valita Pythonin, Go, C/C++ ja muiden ohjelmointikielien väliltä käyttötarkoituksiinsa sopivan vaihtoehdon. Tämä järjestelmä tukee MQTT-protokollaa, jonka avulla on mahdollista hallinnoida, konfiguroida ja kerätä tietoja IoT-laitteilta, jotka ovat yhdistyneet ulkopuoliseen MQTT välittäjään. (What is ThingsBoard IoT Gateway? n.d.)

Järjestelmän tärkeimmät ominaisuudet:

- OPC organisaation Unified Architecture (UA) lisäosa tarjoaa mahdollisuuden kerätä tietoja IoT-laitteilta, jotka ovat yhdistettyjä OPC-UA palvelimiin.
- Tuki tietojenkeräykselle Sigfoxin taustajärjestelmälle.
- Tuetaan Modbus protokolla tietojen keräyksessä.
- Pysyvästi tietojen keräyksellä varmistetaan tietojen lähetys, jopa verkko tai laiteongelmien ilmetessä.
- Automaattinen uudelleenyhdistäminen ThingsBoard klusteriin.
- Saapuvien tietojen ja viestien kartoitus yhtenäiseen muotoon. (What is ThingsBoard IoT Gateway? n.d.)

2.6 LoRaWAN-laitteisto

Käyttämämme fyysiset LoRaWAN laitteet ostimme Pycom yritykseltä, koska heidän järjestelmänsä käyttää MicroPython ohjelmointikieltä normaalisti käytössä olevan Arduinon C/C++ kielen sijaan. Pycom markkinoi ja valmistaa kommunikaatio laitteita. Pycom Oy tunnettiin aikaisemmin nimellä WiPy Oy. Yhtiö perustettiin vuonna 2015 ja toimii Cranleighissa Iso-Britanniassa. Yrityksen perustivat Bettina Rubek Slater ja Fred de Haro. (Company Overview of Pycom Ltd 2019.)

Toteutuksessa käytimme Pycomin myymiä LoPy4 moduuleja, Expansion Boardeja ja antureilla varustettua kehitysalustaa Pysenseä. LoPy4 on LoRa alusta, jolla suoritamme tiedonsiirron Pysensen sensoreista yhdyskäytävällemme. Yhdyskäytävämme

koostuu LoPy4 moduulista ja Expansion Boardista, joiden avulla yhdistämme runkoverkkoon WiFi:n avulla. Jokaiselle laitteelle lisättiin vielä ylimääräinen antenni, jotta järjestelmä toimii ilman ongelmia.

2.6.1 LoPy4

LoPy4 on MicroPythonilla toimiva kehitysalusta, jossa on LoRa, Sigfox, Wifi ja Bluetooth tiedonsiirtomenetelmät. LoPy4 voidaan käyttää tuomaan antureille LoRa tiedonsiirto mahdollisuudet ja rakentaa siitä yksi kanavaisen yhdyskäytävän, josta Pycom käyttää termiä "nano gateway". LoPy4 on myöskin mahdollista käyttää raaka LoRa tilaa, jolla paketit voidaan lähettää suoraan LoPy4 laitteiden välillä. (LoPy4 specsheet n.d.)

2.6.2 Expansion Board ja Pysense

Pycom Expansion Boardin avulla saadaan lisäominaisuuksia yhdistämällä se LoPy4 kanssa. Tärkeimmät ominaisuudet mitä Expansion Boardin avulla saadaan, ovat USB yhteydellä ja Lithium polymer (LiPo) paristolla virrat järjestelmään. Expansion Boardissa ei tule erillisiä antureita mukana, joten käytimme sitä alustana yksikanavaisten yhdyskäytävien toteutuksissa. Expansion Board 2.0 ominaisuudet löytyvät taulukossa 2. Pycomin Expansion Board on yhteensopiva LoPy, SiPy, FiPy ja WiPy 2.0 kanssa. (Expansion Board 2.0 specsheet 2017.)

Taulukko 2. Expansion Board 2.0 (Expansion Board 2.0 specsheet 2017)

| | Expansion Board 2.0 |
|----------------------------------|---|
| Mitat | 65 x 50 x 8mm |
| Virransaanti | USB yhteydellä ja LiPo paristolla |
| MicroSD kortti paikka | Kyllä |
| Virran latausmahdollisuudet | LiPo BQ24040 ja lisäksi 100mA ja 450mA lataus virrat |
| USB muunto serial liitântään | Tehty käyttäen FT234XD |
| Tulovalitsin | TPS2115A estojännitteen suojauksella |
| Paristoliitin | JST tyylinen |
| LED valot | Virran tilan määrittelyyn ja käyttötarkoituksiin toinen |
| Pariston jännitteen monitorointi | WiPy ADC kautta |

Pysense toimii samalla tavalla, kuin Expansion Board, mutta siinä on enemmän sensoreita. Pysensen mukana tulevat samat virtavaihtoehdot, kuin Expansion Boardissa, mutta koska tätä laitetta voidaan käyttää päätelaitteena, jonka tarkoituksena on lähettää pitkän aikaa, voidaan se laittaa ultra pienelle virrankulutukselle, joka tarkoittaa noin 1uA. Alla olevassa taulukossa 3 löytyvät laitteen tärkeimmät ominaisuudet. (Pysense specsheet 2017.)

Taulukko 3. Pysensen ominaisuudet (Pysense specsheet 2017)

| | Pysense |
|--------------------------|-----------------------------------|
| Koko | 55mm x 35mm x 10mm |
| Käyttölämpötila | Min: -45°C Max: +85°C |
| Paino | 11g |
| Virransaanti | USB yhteydellä ja LiPo paristolla |
| MicroSD kortti paikka | Kyllä |
| Tuetut paikannus menetit | Galileo ja GZSS |

Seuraavat arvoja voidaan mitata laitteessa mukana tulevilla sensoreilla:

- Ympäristövalaistus
- Ilmanpaine
- Ilmankosteus
- Lämpötila
- Kiihtyvyyttä (Pysense specsheet 2017.)

2.6.3 Adeunis FTD

Adeunis FTD on luokan A LoRaWAN yhteensopiva laite, jonka avulla voidaan tutkia LoRaWAN yhteyksien toimivuutta ja kantomatkaa. Laite ei tue Point-to-Point (PPP) protokollaa, joten sitä pitää käyttää yksityisessä tai julkisessa operaattoriverkossa. Laitteella voi tutkia miten käytössä oleva verkko toimii, kuten saamalla selville verkon uplink, downlink, SF ja viestien virhesuhde (PER) arvot nykyisessä laitteen sijainnissa. Laitteella pystyy myöskin tarkastelemaan sensoreiden lähettämää liikennettä, kuten lämpötilaa, GPS koordinaatteja ja akun virran määrää. Erityisesti järjestelmä on suunniteltu sovellettavaksi erilaisten ohjelmien vahvistamiseen. (Adeunis Field Test Device User Guide n.d.)

2.7 Elasticsearch ja Grafana

Elasticsearch on Apache Lucene pohjainen hakupalvelin. Sen kehitti Shay Banon ja sen julkaisu tapahtui vuonna 2010. Nykyään järjestelmää ylläpitää Elasticsearch BV ja viimeisin versio on 2.1.0. Elasticsearch on reaaliaikainen hajautettu ja avoimeen lähdekoodin pohjautuva tekstihaku ja analyttinen järjestelmä. Käsiksi järjestelmään pääseen RESTful verkkoselain pohjaisen käyttöliittymän kautta ja järjestelmä käyttää Schemaless JSON dokumentteja tietojen säilytykseen. Järjestelmä on rakennettu Java ohjelmointikielellä, jonka ansiosta Elasticsearchia voidaan käyttää useilla alustoilla. Järjestelmän nopeuden ansiosta käyttäjät pääset tutkimaan suuria määriä tietoja korkeilla nopeuksilla. (Elasticsearch – Basic Concepts n.d.)

Elasticsearchin pääasialliset ominaisuudet:

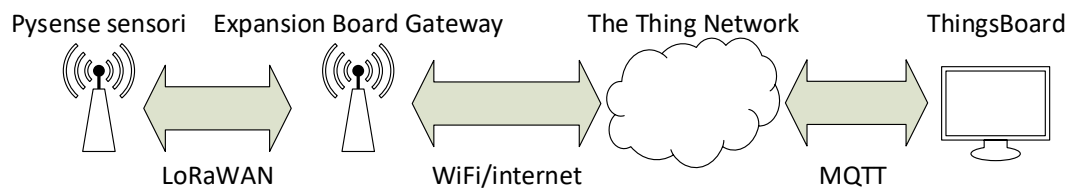
- Skaalautuva rakenteellisten ja rakeenteettomien tietojen kanssa petabyteihin asti
- Soveltuu dokumenttien varastointiin samoilla tavoilla, kuin MongoDB ja RavenDB tietokanta ratkaisut
- Käyttää ei normalisoitua strategiaa tietokanta hakujensa nopeuden parantamiseen (Elasticsearch – Basic Concepts n.d.)

Grafana on avoimen lähdekoodin tietojen visualisointi, monitorointi ja analysointiin tehty työkalu, johon pystyy määrittelemään hälytykset saaduille tiedoille ja on yhteensopiva monen tietokantaratkaisun kanssa. Grafanan avulla käyttäjät voivat visualoida saadut tiedot lämpökartoilla, kaavioilla, taulukoilla tai tehdä yhteenvedon saaduista tiedoista. Järjestelmää voi myöskin muokata lataamalla Grafana yhteisön tekemiä kojelautoja tai laajennuksia. Järjestelmässä voi todentaa käyttäjien oikeuksia LDAP, Google Authentication ja Githubia käyttäen. (What is Grafana? n.d.)

3 Toteutukset ja tutkimustyö

3.1 The Things Network LoRaWAN -toteutus

LoRaWAN toteutuksemme voidaan jakaa kolmeen osa-alueeseen. Ensimmäisenä ovat fyysiset LoRaWAN laitteet, jossa käytettiin Pycom LoRaWAN laitteita tekemään anturi mittauksia ja toimimaan yhdyskäytävinä. Toisena ovat TTN tarjoama taustajärjestelmä ratkaisu, jonka avulla pystyttiin vastaanottamaan lähettämämme liikenne, muuntamaan se haluamaamme muotoon ja lähettämään omaan järjestelmäämme. Viimeisenä on oma järjestelmämme, joka toimii ThingsBoard alustalla ja visualisoi lähettävämme tiedot. Tätä järjestelmän rakennetta havainnollistetaan kuviossa 9.



Kuvio 9. Järjestelmä kokonaisuudessaan

3.1.1 The Things Network -konfigurointi lähetyksille

TTN konfigurointi täytyy tehdä ensimmäisenä toimenpiteenä yhdyskäytävillä ja anturilaitteiden lisäämisen kannalta, koska näin saadaan tarvittavat tiedot laitteiden ohjelmointiin. Rekisteröidessään oman yhdyskäytävän TTN järjestelmään täytyy laitteen EUI ottaa talteen, jotta järjestelmä pystyy tunnistamaan laitteen. Taajuus suunnitelma määriteltiin Euroopan alueella, joka on 868MHz ja määrittelimme reitittimen sijainnin, joka Euroopan alueella on eu.thethings.network sijainnissa. Pycomin Nano-Gateway lasketaan vanhaksi pakettien lähetysjärjestelmäksi, joten se pitää määrittellä rekisteröidessä yhdyskäytävää.

Ohjelmiston pystytyksessä tarvitsee vain määritellä ohjelmisto ID, joka voi olla mitä käyttäjä itse haluaa laittaa. Tämän jälkeen järjestelmä luo tarvittavat ohjelmisto EUI osoitteet ja pääsyavaimet (Access keys). Tämän jälkeen laitteen lisääminen onnistuu järjestelmään, jonka tekemisessä tarvitsee määritellä omavalinnainen laite ID ja LoPy4 laitteessa mukana tuleva EUI.

Pysense tarvitsee oikeat pääsytiedot TTN järjestelmästä, jotta se pystyy lähettämään keräämäänsä telemetriatiedot. Nämä tarvittavat tiedot TTN antaa käyttäjälle ohjelmistojen laitehallinnan kautta, jotta lähettävät laitteet pystyvät suorittamaan OTAA metodilla suoritettavan yhdistämisen ja avainten vaihdon. Tämän prosessin jälkeen saadaan tarvittavat avaimet Pysense laitteelle, jotta se voi suorittaa onnistuneen OTAA metodilla yhdistämisen.

3.1.2 LoRaWAN-laitteiden käyttöönotto

LoRaWAN lähettäjänä toimii LoPy4, johonka lisätään antureita sisältävä Pysense. Pysense antureiden mittaukset lähetetään LoPy4 LoRaWAN yhteyden kautta yhdyskäytävälle, joka koostuu LoPy4 ja Expansion Boardista. Yhdyskäytävä vastaanotettuaan lähetyksen lähettää sen WiFi yhteydellä runkoverkkoon, jonka jälkeen se päätyy TTN järjestelmään. Laitteiden hallinnoinnissa käytetään Pycomin tekemää lisäohjelmaa nimeltään pymkr, jonka avulla Atom tekstieditori saa yhteyden laitteeseen ja pystyy muokkaamaan tarvittavia tiedostoja.

LoPy4 laiteohjelmistoissa oli aikaisemmissa versioissa jaettuna Euroopan ja Pohjois-Amerikan taajuudet, mutta 1.16.0 päivityksessä ne yhdistettiin. Tämä näytti aiheuttavan ongelmia Pycomin Nano-Gatewayn ohjelmiston kanssa, koska ohjelmisto ei pystynyt käsittelemään verkkoon liittymispyyntöä kunnolla. Tätä tehdessä huomattiin kuitenkin, että mikäli laite saa kunnollisen yhdyskäytävän (kahdeksan kanavainen yhdyskäytävä) kanssa kättelyn tehtyä uusin laiteohjelmisto toimii loistavasti. Demo järjestelmämme toimii 1.15.0 laiteohjelmisto versiolla näiden syiden takia.

Pycom Nano-Gatewayn pystytyksessä käytetään LoPy4 vastaanottamaan päätelaitteiden lähettämät tiedot, lähetetään tarvittaessa vastaukset takaisin päätelaitteille,

kuten OTAA yhteyden varmennuksen kanssa ja lopuksi otetaan WiFi yhteys, jonka avulla saadaan lähettyä/vastaanotettua tiedot TTN taustajärjestelmältä. Ainoat muokkaukset mitä tarvitsi tehdä Pycomin tarjoamiin yhdyskäytävä tiedostoihin, jotka löytyvät Pycomin ylläpitämästä Github repositoriosta oli vaihtaa alla olevia kuviossa 10 määreitä, pienentääksemme OTAA aktivointiin kuluva aikaa.

```
#old  
t_us = tmst - utime.ticks_cpu() - 15000  
  
#new  
t_us = tmst - utime.ticks_us() - 20000
```

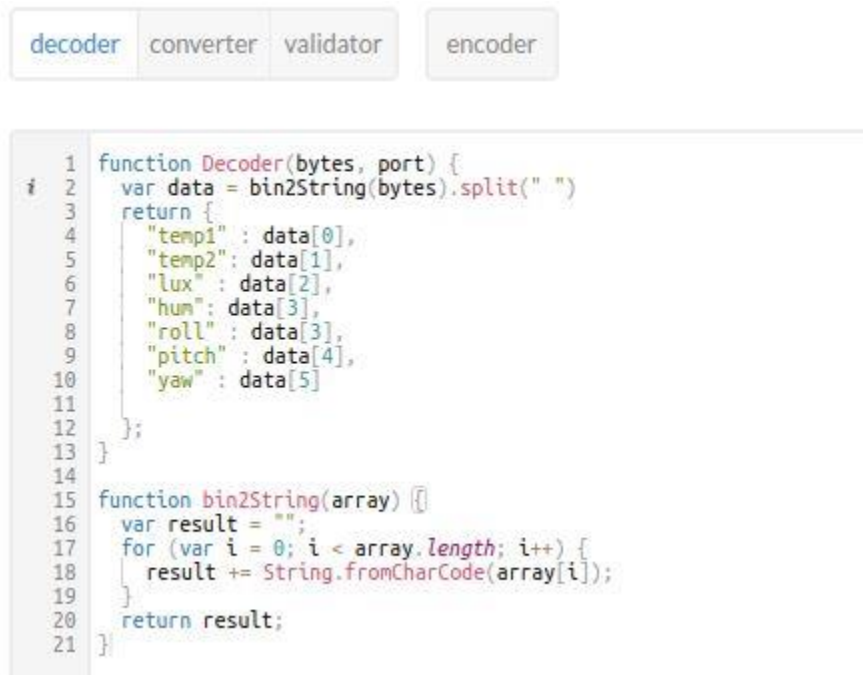
Kuvio 10. Nanogateway.py muokkaukset

Konfigurointimuutokset mitä järjestelmä vaatii, pitää tehdä config.py tiedostossa, kuten NTP osoite ja WiFi tiedot. Muutoksia Euroopan alueelta ei tarvinnut tehdä paljoa, koska se oli alun perin tehty Euroopan tiedoilla. Ainoa asia mitä muutimme, oli laittaa oikeat WiFi tiedot, jotta järjestelmä pystyy lähettämään tietoja eteenpäin.

Pysense sensoreiden tiedonlähetyks vaatii OTAA aktivoinnin, joka vaatii laite ja ohjelmisto avaimet TTN laitehallinnasta. Näiden avainten avulla järjestelmä osaa lähettää keräämänsä tiedot oikeaan paikkaan. Toimiakseen oikein järjestelmä vaatii Pycomin tarjoamat viralliset kirjastot sensoreiden toiminnalle, jotka löytyvät Pycomin julkisesta Github repositoriosta. Näiden kirjastojen lisäksi muokkasimme järjestelmän koodia toimimaan OTAA metodilla, joka löytyy liitteestä 1. Tämä tiedosto lisätään laitteen, jonka jälkeen se aloittaa yhteyden muodostamisen ja tietojen lähetyksen uudelleenkäynnistämisen jälkeen. Muokkasimme tiedonlähetyksen tapahtumaan 30 sekunnin välein. Tiedot lähetetään väleillä erotetuilla merkkijonona TTN järjestelmään, jossa se sitten muutetaan JSON (JavaScript Object Notation) muotoon.

3.1.3 TTN tietojen muuntaminen

Saapuvat bitit joudutaan muokkaamaan luettavaan muotoon käyttäen TTN muunninta (Decoder). Tämän muuntamisen ansiosta saadaan vastaanotetut hyötykuormat luettavaan muotoon ilman, että tätä prosessia tarvitsisi tehdä jokaisessa ohjelmassa erikseen. Muuntamisessa käytetty funktio on kuvion 11 mukainen ja muuntamisen jälkeen tiedot näkyvät kuviossa 12 näytetyllä tavalla.



```
1 function Decoder(bytes, port) {
2   var data = bin2String(bytes).split(" ")
3   return {
4     "temp1" : data[0],
5     "temp2" : data[1],
6     "lux" : data[2],
7     "hum" : data[3],
8     "roll" : data[3],
9     "pitch" : data[4],
10    "yaw" : data[5]
11  };
12 }
13 }
14
15 function bin2String(array) {
16   var result = "";
17   for (var i = 0; i < array.length; i++) {
18     result += String.fromCharCode(array[i]);
19   }
20   return result;
21 }
```

Kuvio 11. Decoder-funktiot

Uplink

Payload

```
32 39 2E 37 35 20 33 32 2E 32 37 20 31 32 37 2E 39 39 20 31 38 2E 36 38 20 31 2E 31 36 20 2D 31 2E 33 36
```

Fields

```
{
  "hum": "18.68",
  "lux": "127.99",
  "pitch": "1.16",
  "roll": "18.68",
  "temp1": "29.75",
  "temp2": "32.27",
  "yaw": "-1.36"
}
```

Metadata

```
{
  "time": "2018-04-16T12:33:37.136052964Z",
  "frequency": 867.3,
  "modulation": "LORA",
  "data_rate": "SF7BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eui-b827ebfffe787a5e",
      "timestamp": 2211043787,
      "time": "",
      "channel": 4,
      "rssi": -120,
      "snr": -6.8,
      "latitude": 62.25952,
      "longitude": 25.68433,
      "altitude": 270,
      "location_source": "registry"
    }
  ]
}
```

Kuvio 12. Muuntamisen jälkeiset tiedot

3.1.4 ThingsBoard ja IoT Gateway

ThingsBoard asennettiin Ubuntu 16.04 alustalle Java 8 kanssa. ThingsBoard on tehty käyttämään Java 8, joten asennuksen yhteydessä tämäkin asennettiin. ThingsBoard voi käyttää SQL tai NoSQL tietokantoja, mutta on suositeltavaa käyttää PostgreSQL tietokantaa, joten päädyimme käyttämään sitä.

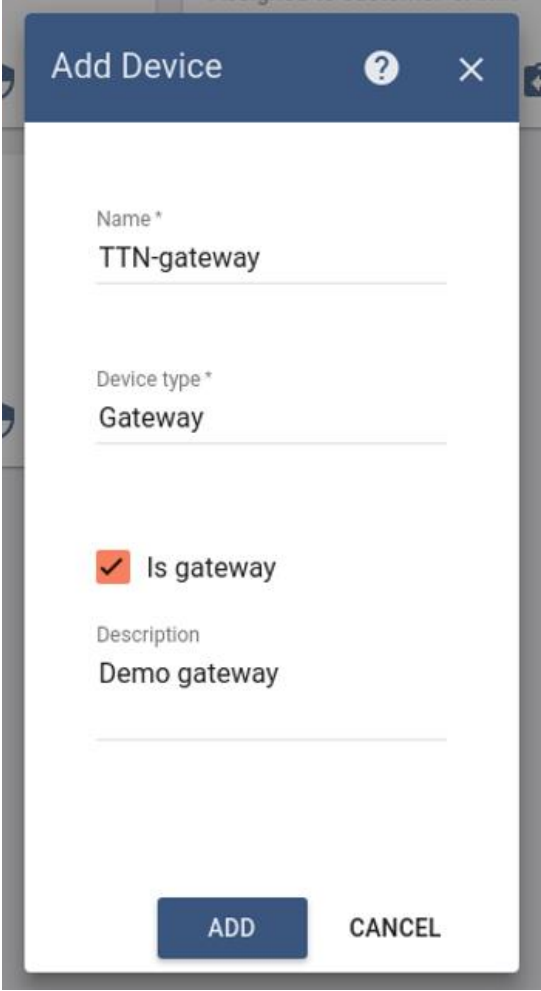
Saadaksemme LoRa laitteiden lähettävät tiedot visualisoitua ThingsBoardissa täytyy ThingsBoard IoT Gateway asentaa myöskin. ThingsBoard IoT gateway toimii siltana TTN MQTT palvelimen ja ThingsBoard alustan välillä. Tämän avulla TTN

vastaanottamat tiedot LoRa antureilta saadaan lähetettyä ThingsBoardiin ja visualisoitua.

Järjestelmän konfigurointia varten tarvitaan seuraavat tiedot:

- ThingsBoard palvelimen IP-osoite tai domain nimi.
- ThingsBoard yhdyskäytävän laite tokeni.
- TTN ohjelmisto ID.
- TTN ohjelmisto tokeni.

Asennettuumme ThingsBoard IoT Gatewayn pystyimme hallinnoimaan järjestelmää menemällä verkkoselaimella ThingsBoardin käyttämään IP-osoitteeseen. Täällä loimme uuden laitteen ja muokkasimme sen toimimaan yhdyskäytävänä kuvion 13 mukaisesti.



The image shows a mobile-style dialog box titled "Add Device". It contains the following fields and controls:

- Name***: TTN-gateway
- Device type***: Gateway
- Is gateway**
- Description**: Demo gateway

At the bottom, there are two buttons: "ADD" and "CANCEL".

Kuvio 13. Yhdyskäytävän lisääminen ThingsBoardissa

Laitteen luonnin jälkeen pystytään tarkastelemaan sen tietoja details välilehden alta. Tämän välilehden alta löytyvät viimeisimmät vastaanotetut telemetriatiedot, joista löytyvät tiedot, kuten yhdistyneiden laitteiden määrä, hälytykset, asiakas ja palvelin määrittelyt. Tämän jälkeen saatiin pääsy tokeni, jota tarvitaan ThingsBoard yhdyskäytävän konfiguroinnissa.

Viimeisimmät konfigurointi tiedot löytyvät TTN hallinnointi paneelista. Ohjelmisto ID ja ohjelmisto tokenin saadaan yleisestä ohjelmistojen hallinnointi paneelista, joka näkyy kuviossa 14. Tässä kuvion kohdassa yksi saadaan laitteen ohjelmisto ID ja kohdassa kaksi saadaan ohjelmisto tokeni.

The screenshot shows the ThingsBoard Console interface for a specific application named 'lora'. The interface is divided into several sections:

- APPLICATION OVERVIEW:** This section contains key information about the application:
 - Application ID:** 'lora' (highlighted with a red circle '1').
 - Description:** (empty field).
 - Created:** '11 days ago'.
 - Handler:** 'ttn-handler-eu (current handler)'.
- APPLICATION EUIs:** A section for managing EUIs, currently showing a list of EUIs.
- DEVICES:** A section showing '3 registered devices'.
- COLLABORATORS:** A section showing the user 'konte' with options to manage collaborators.
- ACCESS KEYS:** A section for managing access keys. It shows a 'default key' with a 'devices' tab selected. The key value is partially visible as '..... base64' and is highlighted with a red circle '2'.

Kuvio 14. TTN-ohjelmisto ID ja ohjelmistotokeni

Näillä tiedoilla saadaan ThingsBoard IoT Gateway ja MQTT konfiguraatiot tehtyä. Grafical User Interfacea (GUI) käyttämällä järjestelmää ei saatu toimimaan, joten konfiguraatio muutokset tehtiin palvelimella oleviin tiedostoihin. Ensimmäisenä yhdyskäytävän konfiguraatio tiedosto `tb-gateway.yml` päivitettiin lisäämällä siihen alla olevassa kuvio 15 olevat tiedot.

```
gateway:
  connection:
    host: "ThingsBoard; Server IP or Domain"
  security:
    accessToken: "ThingsBoard; Gateway Device Token"

mqtt:
  enabled: true
```

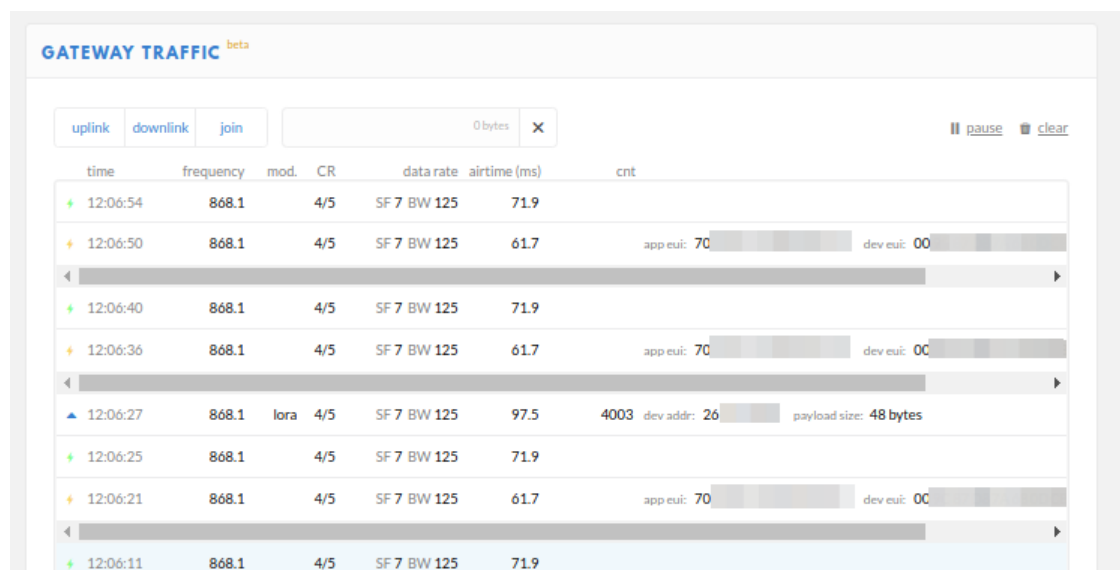
Kuvio 15. ThingsBoard-yhdyskäytävälisäykset

MQTT yhteyksien saamiseksi ThingsBoard yhdyskäytävän ja TTN MQTT palvelimen välillä toimimaan liittessä 2 olevat konfiguraatiot tiedot lisätään `mqtt-config.json` tiedostoon.

3.1.5 Järjestelmän toiminta kokonaisuutena

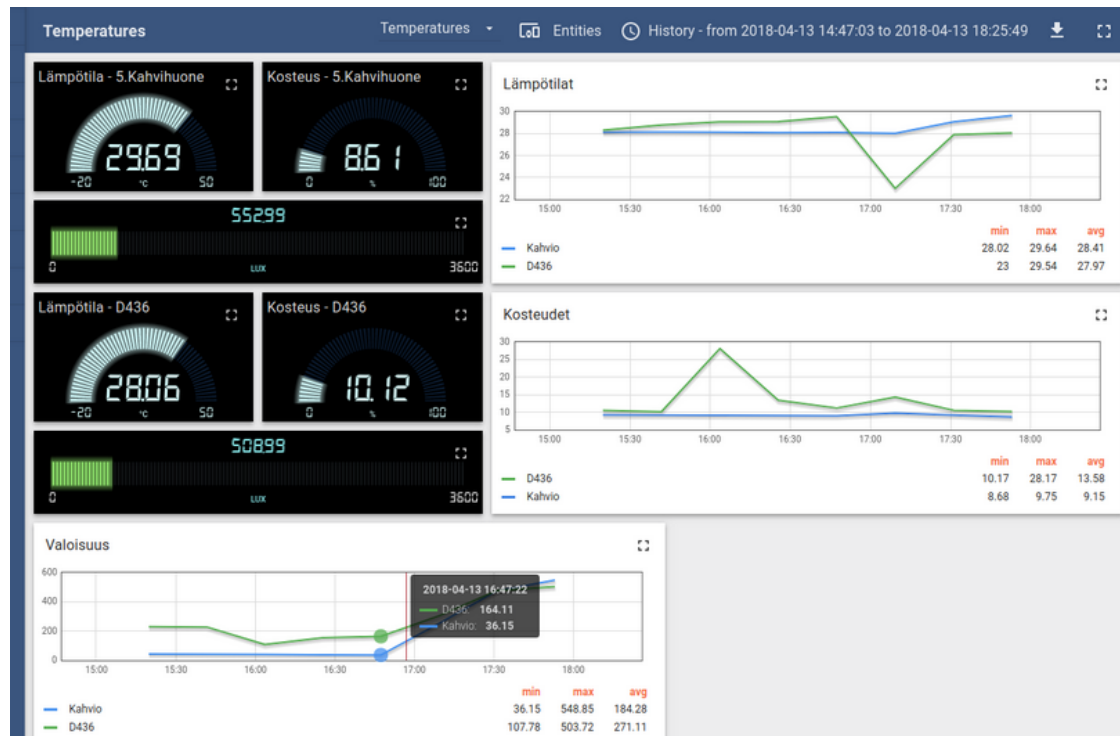
Järjestelmän pystytyksen ja konfiguroinnin jälkeen aloitetaan PySense laitteella OTAA prosessi. Tämän prosessin etenemisen seuraamiseksi ohjelmoimme LED valot näyttämään käyttäjälle, onko yhteyden muodostaminen onnistunut. Yhteyden muodostamiseen kuluva aika vaihtelee saataville olevien yhdyskäytävien mukaan. Käyttäessämme NanoGateway yhdyskäytäviä huomasimme järjestelmän olevan altis ongelmille aktivointiprosessin aikana, johtuen lähetettävien viestien aikaeroista, mutta kunhan yhteydet muodostuvat järjestelmä toimii ongelmitta. Yhteyden muodostamisen jälkeen sensoreiden tiedot lähetetään 868MHz taajuudella eteenpäin yhdyskäytävälle.

LoRaWAN yhdyskäytävät vastaanottavat paketit ja lähettävät ne eteenpäin TTN:lle. On tärkeää muistaa, että lähetykset LoPy laitteelta voivat näkyä monella tai ei yhdelläkään yhdyskäytävällä. TTN yhdistää ja poistaa tuplaantuneet viestit mistä tahansa vastaanottamastaan liikenteestä. LoRaWAN liikenne on salattu, joten vastaanotettujen viestien hyötykuorman sisältöä ei voida tarkastella vastaanottavalla yhdyskäytävällä. Tämä on tärkeää, koska yhdyskäytävät voivat välittää paketteja eteenpäin ja takaisin muiltakin LoRa käyttäjiltä. Alla olevasta kuviosta 16 nähdään miltä liikennöinti näyttää LoPy NanoGateway yhdyskäytävässämme.



Kuvio 16. Liikennöinti yhdyskäytävässä

Vastaanotettua liikenteen TTN järjestelmä muokkaa sen haluamaamme muotoon. Tämän jälkeen vastaanotettu telemetria lähetetään eteenpäin ThingsBoard alustallemme visualisoitavaksi TTN tarjoaman MQTT palvelun avulla. Alla olevasta kuviosta 17 nähdään miltä visualisoitu tiedot näyttävät.



Kuvio 17. Tietojen visualisointi ThingsBoardissa

3.2 LoRaWAN Digita -tutkimus

3.2.1 Yleisesti

Projektissa tutkittiin LoRaWAN toteutusta käyttäen Digitan tarjoamaa taustajärjestelmää. Digita tarjoaa asiakkailleen mahdollisuutta heidän taustajärjestelmänsä ja pääsyä heidän ylläpitämiinsä yhdyskäytäviin. Digita on pystyttänyt yhdyskäytävänsä kaikkiin Suomen suurimpiin kaupunkeihin, ja peitealue kattaa suurimman osan Suomesta (Digitan IoT LoRaWAN-verkon peittokartta n.d).

Digita toimii radio- ja tv-ohjelmien välittäjänä koko Suomen alueella. Digita myöskin kehittää ja tarjoaa monipuolisia internet-tv- ja radiopalveluita. Digitan palveluvalikoimaan kuuluvat myöskin IoT- ja DataCenter-palvelut. Asiakkaisiin kuuluvat mediayhtiöt sekä mobiili- ja laajakaistaoperaattorit. (Digita yhtiönä n.d.)

Digitan taustajärjestelmää testattiin ainoastaan yhteyksien osalta, sillä tämän järjestelmän koko ketjun rakentaminen olisi vaatinut pysyvät ”end point” ratkaisun pystyttämistä, johonka projektin suhteen ei ollut tarvetta. Kuitenkin tässä projektissa tutkittiin yliopiston toteuttamaa ratkaisua end point järjestelmästä, jonka toteutus tehtiin käyttäen Elasticsearchia tietokantana, Grafanaa tietojen visualisointiin ja lotli-parseria tietojen saantiin tietokantaan. Järjestelmän toteutus olisi toiminut teoria tasolla siten, että lähetävä LoRa laite olisi yhdistänyt Digitan yhdyskäytävään, josta se olisi Digitan taustajärjestelmä ratkaisusta lähetetty end pointille. Digitan yhteyksiä testattiin yhdistämällä Adeunis 868 Field Test Device (FTD) järjestelmän, jonka avulla pystytään tutkimaan LoRa yhteyksien toimivuutta.

3.2.2 Adeunis FTD käyttöönotto

Laitteen käyttöönotossa laite yhdistettiin tietokoneeseen USB-B piuhalla ja ladattiin Adeunin suosittelema Hercules sarjaportti terminaali ohjelmisto, jolla pystyttiin muokkaamaan laitteen järjestelmäasetuksia. Hercules ohjelmiston avulla järjestelmä voidaan laittaa käyttämään OTAA menetelmää yhteyksien ottamisen kanssa ja määrittellä sen käyttämät osoitteet. Taulukossa 4 käydään läpi käytetyt komennot ja selitetään niiden tarkoitus. Järjestelmään syötetyillä tiedoilla ja laitteen takana olevalla DevEUn avulla saadaan järjestelmä yhdistettyä Digitan verkkoon.

Taulukko 4. Adeunis-järjestelmän muokkauskomennot ja selitteet

| Syntaksi | Selitys |
|--------------------------------|--|
| FF\$FF\$FF\$FF\$FF\$2B\$2B\$2B | Komentojen syöttö tilaan siirtyminen |
| ATT63 PROVIDER <CR><LF> | AppKeyn muokkaamisen salliminen |
| ATS221=1 <CR><LF> | OTAA tilan käyttöönotto |
| ATS216=B6C32323 <CR><LF> | LoRa AppKeyn muokkaaminen tapahtuu 4 tavun välein, joka tapahtuu ATS216-ATS219 välillä |
| ATS217=EAB23319 <CR><LF> | |
| ATS218=3A3251CA <CR><LF> | |
| ATS219=DE5393AB <CR><LF> | |
| ATS214=16653133 <CR><LF> | LoRa AppEUIn muokkaaminen |
| ATS215=A9CDD1B3 <CR><LF> | |
| AT&W <CR><LF> | Muokkausten tallentaminen |
| ATO <CR><LF> | Muokkaustilasta poistuminen |

3.2.3 Digitan taustajärjestelmä

Digitan taustajärjestelmä on Actilityn tekemä verkkoselaimessa toimiva GUI, jossa pystyy lisäämään LoRaWAN laitteita ja tarkastelemaan liikennettä. Liikenteen tarkastelussa nähdään jokaisen lähtevän paketin yhteyksien tiedot, kuten laitteen DevEUI, SNR arvo ja MIC arvot. Laitteiden hallinnassa LoRaWAN laitteita voidaan lisätä tietty määrä, joka riippuu siitä, minkälainen sopimus on Digitan kanssa solmittu. Laitteiden hallinnassa on mahdollista määritellä omat ohjelmisto palvelimet, mutta sitä ei tämän projektin puitteissa tehty. Järjestelmässä voi myöskin tarkastella verkon toimivuutta kartan kautta ja hallinnoiden verkon toimintoja, mutta meillä ei ollut oikeuksia niihin, koska ne ovat Digitan ylläpidon alla.

Adeuniksien lisääminen Digitan taustajärjestelmään onnistuu laitteidenhallinnan kautta. Laitteen lisäämiseen tarvitaan laitteen valmistaja, malli, aktivointi tapa ja laitteen osoitetiedot. Tätä prosessia näytetään kuviossa 18.

New device

Administrative data

Device name: Adeunis LoRaWAN tester

Marker: * Change marker

Administrative info:

Administrative location: * 62° 13' 41" N 25° 45' 28" E Change location

Motion indicator: Device profile settings

Device identification

Manufacturer: * Adeunis

Model: * Field Test Device - class C - EU eu868

Device activation: Over The Air Activation (OTAA)

DevEUI: * 00-18 [REDACTED]

AppEUI: * 16-6D [REDACTED]

AppKey: * B6-C3 [REDACTED]

Network parameters

Connectivity plan: DIGITA Connectivity Supplier / ADRv2 (9)

DevAddr: * Allocated by the network server

Application layer handling

Application server routing profile: DIGITA RP

Kuvio 18. Adeunixin lisääminen Digitan järjestelmään

Laitteen lisäämisen jälkeen saadaan yhteystiedot laitteen näkyville, kuten lähetettyjen ja vastaanotettujen pakettien määrät. Laitteen lisäämisen jälkeen sitä pystyttiin käyttämään LoRaWAN-yhteyksien mittaamiseen ja se lähetti saadut tiedot Digitan

järjestelmään, mutta koska projektissa ei suoritettu end point ratkaisua, niin näitä tietoja ei visualisoitu.

3.2.4 End Pointin pystytys

Järjestelmän pystytys tehtiin Ubuntu 16.04 alustalle, jolle asennettiin NodeJs, Elasticsearch, Java 8, Grafana ja kloonattiin yliopiston tekemä lotli-parser, joka teoriassa vastaanottaa ja muuttaa saapuvat tiedot oikeaan muotoon. Järjestelmän toiminnan takaamiseksi uudelleenkäynnistämisen yhteydessä asennettiin pm2 ohjelmisto ja konfiguroitiin se aloittamaan lotli-parser uudestaan järjestelmän käynnistyessä uudestaan. Grafanan asennuksessa suoritettiin virallisten Ubuntu ohjeiden mukaisesti. Elasticsearchin kanssa tehtiin yhden solmun (single node) asennus, koska koko järjestelmää testattiin yhdellä työasemalla.

Asennusten jälkeen käytetään Grafanan verkkoselaimella avattavaa GUI, jolla tiedot pystyisi visualisoimaan käyttäjän haluamalla tavoilla. Paneeleja lisäämällä käyttäjät voivat valita haluamansa näköisen representaation saapuvista tiedoista. Paneelia muokkaamalla on mahdollista valita haluttu tietolähde, jonka jälkeen käyttäjät voivat valita millä kriteereillä tietoja valitaan.

4 Pohdinta

Projektin päämääränä oli tutkia ja kehittää erilaisia IoT-ratkaisuja, joista LoRa oli yksi teknologia, jonka mahdollisuuksia tutkittiin. TTN toteutuksen tavoitteena oli tutkia LoRa teknologian toimintaa ja erilaisia markkinoilla olevia LoRa lähettämiä. Näiden tutkimusten pohjalta aloitettiin järjestelmän rakentaminen, jolla saataisiin lähetettyä sensori tietoja LoRa lähettimen avulla omalle palvelimellemme.

LoRa järjestelmiä tutkittaessa todettiin, että suurin osa niistä on Arduinon C kielellä toimiva järjestelmiä, joka olisi tuottanut projektiryhmän jäsenille lisätöitä sen opetteluun. Etsimme myöskin mahdollisimman helppokäyttöistä järjestelmää, joka ei vaatisi paljoa elektroniikka osaamista järjestelmän kasaamisessa. Näillä perusteluilla päädyimme käyttämään Pycomin LoRaWAN laitteita ja toteuttamaan järjestelmän käyttäen TTN järjestelmää, josta pystyimme lähettämään tiedot eteenpäin projektissa aikaisemmin käyttöönotettuun ThingsBoard järjestelmään. Tämän työn perusteella saatiin kehitettyä toimiva IoT-järjestelmä, jonka avulla pystyttiin perehtymään tarkemmin LoRaWAN järjestelmien toimintaan.

Valitut laitteet soveltuivat hyvin LoRaWAN järjestelmien toiminnallisuuteen perehtymisessä ja soveltuivat hyvin demo järjestelmän pystyttämiseen, vaikka yhdyskäytävät toimivatkin vain yhdellä kanavalla. Pycomin laitteiden käyttöönotossa ainoa ongelma tuli vastaan järjestelmäpäivityksen 1.16.0 myötä, jossa 868 MHz ja 915 MHz taajuuDET yhdistettiin. Tämän takia Pycom yhdyskäytäviin yhdistäminen ei onnistunut, joten korjasimme asian käyttämällä 1.15.0 versiosta järjestelmästä. Saapuvien viestien muuntaminen TTN puolella tuotti ongelmia oman osaamattomuuden takia, mutta onneksi sain tukea projektin muilta jäseniltä, jotka osasivat hoitaa tämän ohjelmoinnin paremmin.

Toteutuksemme tehtiin käyttäen MicroPython ohjelmointikieltä, jonka takia toteutuksemme eroaa suurimmasta osasta muista LoRaWAN toteutuksista, koska yleisesti ne toteutetaan käyttäen Arduinon ohjelmointikieltä. Järjestelmän toteutusta ei ole

suositeltavaa tehdä samoilla yhdyskäytävillä, sillä ne ovat yksikanavaisia yhdyskäytäviä, jonka takia niiden toiminallisuus on paljon rajoittuneempaa. Saadessamme sensori mittauksien tuloksia huomattiin saatujen arvojen olevan epätarkkoja, joten ei ole suositeltavaa käyttää näitä laitteita, mikäli tarvitaan tarkkoja mittaustuloksia.

Kokonaisuudessaan järjestelmä hyvin kuvaa LoRaWAN järjestelmän toimintaa ja miten se täytyy rakentaa. Tämän toteutuksen pohjalta voidaan rakentaa LoRaWAN toteutuksia IoT-projekteihin, joissa tiedonsiirtoon ei tarvita suurta kapasiteettia. Projektissa käytetyt avoimen lähdekoodin ohjelmistot myöskin mahdollistavan toteutusten tekemisen ilman suuria panostuksia maksullisiin ohjelmiin LoRaWAN ratkaisuisissa.

Järjestelmässä käytetyt fyysiset laitteet toimivat hyvin ympäristön demo tarkoitukseen, mutta tarkkojen anturi mittausten saamiseksi kannattaa suunnitella omat LoRaWAN anturi laitteet tarpeiden mukaan. Yhdyskäytävä toimi meidän tarkoituksiimme riittävän hyvin, mutta kahdeksan kanavaisen yhdyskäytävän käyttö on suositeltavaa, mikäli järjestelmä halutaan ottaa teollisuuskäyttöön. Paremmalla yhdyskäytävällä saadaan varmemmat yhteydet ja suurempi määrä laitteita yhdistymään järjestelmään. Järjestelmäkehitystä voidaan toteuttaa laitteiden antamien tietojen perusteella, kuten GPS koordinaattien avulla toteutettu visualisointi kartalla, jossa näytetään laitteiden antamat tiedot. Mahdollisuudet erilaisille toteutuksille ovat valtavat, sillä ne ovat täysin riippuvaisia käytettyjen laitteiden ja ohjelmistojen ominaisuuksista.

Digitan LoRaWAN järjestelmä oli tarkoitus tutkia ensiksi, mutta viivästysten takia päädyttiin toteuttamaan avoimen lähdekoodilla toteutettu TTN ratkaisu ensiksi. Tämän takia Digitan toteutuksen tarkoitus oli enemmän tutkia järjestelmän toimintaa ja testata miten järjestelmä toimii. Tähän lisäksi tutkittiin yliopiston toteutusta end point järjestelmästä.

Digitan ratkaisussa saatiin yhdistettyä Adeunis testeri järjestelmään, jonka avulla pystyttiin tutkimaan LoRaWAN yhteyksien kuuluvuutta. Taustajärjestelmä ratkaisussa oli paljon monitorointiin tarkoitettuja ominaisuuksia, kuten kartta ominaisuus mistä lait-

teiden sijaintia pystyi monitoroimaan. Digitan taustajärjestelmä tarjoaa paljon enemmän monitorointi mahdollisuuksia, kuin TTN järjestelmä, mutta kärsii isojen yritysten mukanaan tuomasta byrokratiasta. Yliopiston järjestelmän pystytys tapahtui helposti ja oli erittäin monipuolinen visualisointi ratkaisu. Näiden tutkimusten tärkein tulos oli saada tutkittua järjestelmiä ja käytössä olleille järjestelmille kirjoitettua tietopohjaa.

Digitan verkkoon saatiin laitteet yhdistettyä ongelmitta ja järjestelmä sai yhteyden Adeunis laitteeseen ongelmitta. Järjestelmälle olisi pitänyt pystyttää pysyvä end point järjestelmä, jonka avulla olisimme saaneet testailtua yhteyksien toimintaa tarkemmin, mutta tiedonpuutteen ja aikarajoitteiden takia tätä ei tehty. Tämän takia Digitan toteutus jäi minimalistiseksi. Yliopiston järjestelmään olisi päässyt paremmin perehtymään, mikäli koko ketju olisi pystytetty.

Tämän toteutuksen osalta oli paljonkin ongelmia saada tietoa minkälainen järjestelmä olisi tarvittu end point toteutukseen. Järjestelmän käyttöönotossakin kesti paljon aikaa, kun kommunikointi Digitan kanssa ei onnistunut sujuvasti. Digitan hallussa olevasta järjestelmästä ei hirveästi tahtonut tietoa löytyä ja tähän olisi tarvinnut heidän osaltaan enemmän opastusta. Tässä vielä enemmän sekoitti päätä se mitkä ominaisuudet meillä olivat käytössä ja mitä kaikkea niillä pystyi tekemään.

Näitä tutkimuksia voidaan tulevaisuudessa käyttää LoRaWAN toteutuksissa, jotka käyttävät Digitan taustajärjestelmää ja yhdyskäytävä palveluita. Tämän järjestelmän kehittämiseen tarvitaan pysyvä end point järjestelmä, jossa voidaan käyttää samantyylistä järjestelmää mitä yliopisto on käyttänyt. Tämän järjestelmän toteutus on helppompi tehdä käyttäen TTN palveluita, sillä heidän järjestelmästään saa helpommin tietoa ja käyttäjät pystyvät itse määrittelemään minne vastaanotetut tiedot lähetetään.

Lähteet

- Adeunis Field Test Device User Guide. N.d. Adeunis käyttöönnotto dokumentti. Viitattu 10.5.2019. https://www.adeunis.com/wp-content/uploads/2017/08/FTD_LoRaWAN_EU863-870_UG_FR_GB_V1.2.2-1.pdf.
- Augustin, A., Clausen, T., Kim, D., Townsley, W. & Yi, J. 2016. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. PMC artikkeli. Viitattu 8.5.2019. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5038744/>.
- Company Overview of Pycom Ltd. 2019. Bloombergin sivuilta löytyvä S&P Global Market Intelligencen tekemä yrityskuvaus. Viitattu 17.4.2019. <https://www.bloomberg.com/research/stocks/private/snapshot.asp?privcapId=336464922>.
- Comparison of LPWAN technologies. N.d. Artikkelin Waviot sivustolla. Viitattu 8.5.2019. <https://waviot.com/technology/waviot-lpwan-technology-comparison>.
- Continuous Frequency Sweep. 2018. Artikkelin National Instrumentsin sivuilta. Viitattu 17.5.2019. http://zone.ni.com/reference/en-XX/help/372416L-01/svtconcepts/lvac_continuous_sweep/.
- Digita-yhtiönä. N.d. Digitaalisen yrityskuvaus yrityksen kotisivuilta. Viitattu 10.5.2019. <https://www.digita.fi/yhtio>.
- Digitaalinen IoT LoRaWAN-verkon peittokartta. N.d. Digitaalisen verkkosivut. Viitattu 10.5.2019. https://www.digita.fi/yrityksille/iot/iot_lorawan-verkon_peittokartta.
- Elasticsearch – Basic Concepts. N.d. Artikkelin Tutorialspoint-sivustolta. Viitattu 10.5.2019. https://www.tutorialspoint.com/elasticsearch/elasticsearch_basic_concepts.htm.
- Expansion Board 2.0 specsheet. 2017. Pycomin tekninen dokumentaatio Expansion Board 2.0 laitteesta. Viitattu 10.5.2019. <https://docs.pycom.io/.gitbook/assets/expansion2-specsheet.pdf>.
- IoTli- IoT:sta uutta liiketoimintaa. 2018. Projektin kuvaus yliopiston sivuilta. Viitattu 14.5.2019. <https://www.jyu.fi/it/fi/tutkimus/tutkimushankkeet/rakennerahasto/iot>.
- LoPy4 specsheet. 2017. Pycomin tekninen dokumentaatio LoPy4 laitteesta. Viitattu 10.5.2019. <https://pycom.io/wp-content/uploads/2017/11/lopy4Specsheet17.pdf>.
- LoRa Alliance Security Whitepaper. 2017. LoRaWANin turvallisuuteen perehtyvä dokumentti. Viitattu 9.5.2019. https://lora-alliance.org/sites/default/files/2018-04/lora_alliance_security_whitepaper.pdf.
- LoRa Modulation Basics. 2015. Semtech-yrityksen tekninen dokumentaatio LoRa modulaatiosta. Viitattu 8.5.2019. <https://www.semtech.com/uploads/documents/an1200.22.pdf>.

- Network Architecture. N.d. The Things Networkin tekemä verkkoarkkitehtuuridokumentaatio. Viitattu 9.5.2019. <https://www.thethingsnetwork.org/docs/network/architecture.html>.
- Peerla, R. 2007. Constant Envelope Modulation Techniques. PowerPoint esitelmä. Viitattu 8.5.2019. <https://www.scribd.com/doc/97348481/Constant-Envelope-Modulation-Techniques>.
- Pysense specsheet. 2017. Pycomin tekninen dokumentaatio Pysense laitteesta. Viitattu 10.5.2019. <https://docs.pycom.io/.gitbook/assets/pysense-specsheet.pdf>.
- Rouse, M. 2014. access network. Viitattu 9.5.2019. <https://whatis.techtarget.com/definition/access-network>.
- Thakur, N. 2018. Forward Error Correction (FEC). Viitattu 8.5.2019. <https://www.tutorialspoint.com/forward-error-correction-fec>.
- The Things Network. N.d. Yritys kuvaus. Viitattu 9.5.2019. <https://angel.co/the-things-network>.
- What is Grafana?. N.d. Grafanan sivuilla oleva järjestelmänkuvaus. Viitattu 10.5.2019. <https://grafana.com/grafana>.
- What is LoRaWAN. 2015. Tekninen yleiskatsaus LoRa ja LoRaWAN järjestelmistä. Viitattu 9.5.2019. <https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>.
- What is ThingsBoard?. N.d. ThingsBoardin tekemä dokumentaatio järjestelmästä. Viitattu 10.5.2019. <https://thingsboard.io/docs/getting-started-guides/what-is-thingsboard/>.
- What is ThingsBoard IoT Gateway?. N.d. ThingsBoardin tekemä dokumentaatio. Viitattu 10.5.2019. <https://thingsboard.io/docs/iot-gateway/what-is-iot-gateway/>.
- Xian, L & Tingthanathikul, W. 2004. Advanced Encryption Standard (AES) in Counter Mode. Viitattu 9.5.2019. <https://pdfs.semanticscholar.org/7e2f/3c9060256bd7b0a4749c39cf6032736037f5.pdf>.

Liitteet

Liite 1. LoPy4-yhteyksien ohjelmointi

```

""" OTAA Node example compatible with the LoPy Nano Gateway """

from network import LoRa
import socket
import binascii
import struct
import time
import pycom
import config

from lib.MPL3115A2 import MPL3115A2
from lib.LTR329ALS01 import LTR329ALS01
from lib.SI7006A20 import SI7006A20
from lib.LIS2HH12 import LIS2HH12

pycom.heartbeat(False)
pycom.rgbled(0x00007f) # blue

# initialize LoRa in LORAWAN mode.
# Europe = LoRa.EU868
lora = LoRa(mode=LoRa.LORAWAN)

# create an OTA authentication params
app_eui = binascii.unhexlify('70XXXXXXXXXXXXXX')
app_key = binascii.unhexlify('DA1XXXXXXXXXXXXXXXXXXXXXXXXXXXX')
dev_eui = binascii.unhexlify('00XXXXXXXXXXXXXX')

for i in range(0, 2):
    lora.remove_channel(i)

# set the 3 default channels to the same frequency (must be before sending the OTAA join request)
lora.add_channel(0, frequency=config.LORA_FREQUENCY, dr_min=0, dr_max=5)
lora.add_channel(1, frequency=config.LORA_FREQUENCY, dr_min=0, dr_max=5)
lora.add_channel(2, frequency=config.LORA_FREQUENCY, dr_min=0, dr_max=5)

# join a network using OTAA
lora.join(activation=LoRa.OTAA, auth=(dev_eui, app_eui, app_key), timeout=0, dr=config.LORA_NODE_DR)

```

```

# wait until the module has joined the network
join_wait = 0
while True:
    time.sleep(2.5)
    if not lora.has_joined():
        pycom.rgbled(0x7f7f00) # yellow
        time.sleep(0.25)
        print('Not joined yet...')
        pycom.rgbled(0x7f0000) # red
        time.sleep(0.1)
        join_wait += 1
        if join_wait == 5:
            pycom.rgbled(0x00007f) # blue
            time.sleep(0.1)
            print('Join retry')
            lora.join(activation=LoRa.OTAA, auth=(dev_eui, app_eui, app_key), timeout=0, dr=config.LORA_NODE_DR)
            join_wait = 0
        else:
            break

print('Network joined!')
pycom.rgbled(0x007f00) # green

# create a LoRa socket
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
# set the LoRaWAN data rate
s.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)
# make the socket non-blocking
s.setblocking(False)

# init the libraries
temp = MPL3115A2() # Barometric Pressure Sensor with Altimeter
lux = LTR329ALS01() # Digital Ambient Light Sensor
multi = SI7006A20() # Humidity and Temperature Sensor
accel = LIS2HH12() # 3-Axis Accelerometer

while True:
    data = "%.2f %.2f %.2f %.2f %.2f %.2f" % (temp.temperature(), multi.temperature(), lux.light()[0], multi.humidity(), accel.roll(),
    accel.pitch())
    print("Sending %s" % data)
    s.send(data)
    pycom.rgbled(0x00007f) # blink as blue for 0.25s when sending
    time.sleep(0.25)
    pycom.rgbled(0x007f00) # green
    time.sleep(30)

```

Liite 2. ThingsBoard MQTT-konfigurointi

```

{
  "brokers": [{
    "host": "eu.thethings.network",
    "port": 1883,
    "ssl": false,
    "retryInterval": 3000,
    "credentials": {
      "type": "basic",
      "username": "The Things Network; Application Id",
      "password": "The Things Network; Application Token"
    }
  },
  "mapping": [{
    "topicFilter": "+/devices+/up",
    "converter": {
      "attributes": [],
      "timeseries": [{
        "type": "double",
        "key": "temp1",
        "value": "${$.payload_fields.temp1}"
      }, {
        "type": "double",
        "key": "temp2",
        "value": "${$.payload_fields.temp2}"
      }, {
        "type": "double",
        "key": "hum",
        "value": "${$.payload_fields.hum}"
      }, {
        "type": "double",
        "key": "lux",
        "value": "${$.payload_fields.lux}"
      }
    ],
    "type": "json",
    "deviceNameJsonExpression": "${$.hardware_serial}",
    "filterExpression": ""
  }
  ]
},
"connectRequests": [{
  "topicFilter": "+/devices+/events/activations",
  "deviceNameJsonExpression": "${$.dev_eui}"
}],
"disconnectRequests": [],
"attributeRequests": [],
"attributeUpdates": [],
"serverSideRpc": []
}]
}

```