

Seminararbeit: Lorawan

Tobias Sigmann

17. Mai 2019

Inhaltsverzeichnis

1	Einführung in Lora	3
2	Aufbau eines Lora-Netzwerk	4
2.1	Gateway	4
2.2	Netzwerkserver	6
2.3	Applicationsserver	6
2.4	Join-Server	7
2.5	End-Gerät	7
3	LoraWan Funktionsweise	7
3.1	Schichtenmodell	8
3.2	Netzwerkbeitritt	9
3.2.1	OTAA	9
3.2.2	ABP	10
3.3	Protokoll	11
3.4	Übertragungsart	14
3.4.1	Adaptive Data Rate	14
4	Lora Geräte Klassen	15
4.1	Klasse A	15
4.1.1	Uplink	16
4.1.2	Downlink	16
4.2	Klasse B	17
4.2.1	Klassenwechsel A nach B	18
4.2.2	Uplink	18
4.2.3	Downlink	18
4.3	Klasse C	18
5	Sicherheit	19
6	Live-Beispiel	19
7	Ausblick	19

1 Einführung in Lora

Lora ist ein Low Power, Wide Area (LPWA) Netzwerkprotokoll und somit sehr gut für batteriebetriebene kabellose Geräte geeignet. Deswegen wird Lora auch oft im Internet of Things (IoT) Bereich verwendet. Mittels der bidirektionalen Kommunikation ist es möglich Daten und Befehle über weite Strecken zu übertragen. Leider leidet darunter die Geschwindigkeit, sodass sich Lora nicht als WLAN Ersatz eignet. Trotzdem können zwischen 0.3 und 50 kbps erreicht werden. In Europa werden 863 MHz bis 870 MHz verwendet. Allerdings variiert der Frequenzbereich für andere Kontinente. Je nach Bedingungen können so bis zu 20km entfernte Endgeräte erkannt und mit diesen kommuniziert werden. Es ist sogar möglich den Standort des Gerätes zu bestimmen.

Eine Alternative zu Lora ist Sigfox, hierauf werde ich nicht weiter eingehen. LoRaWAN 1.1

[Tec15](Optimiert für Batterie Kapazität(Teilnehmer) Reichweite, Kosten mehrjährige Batterielaufzeit, kleine Datenmengen, große Reichweite, LPWAN (Low Power WAN)

Kriterien für Lora: Netzwerk Architektur, Reichweite, Batterielaufzeit, Interferenzrobustheit, Anzahl Knoten, Sicherheit, bidirektionale Kommunikation, verschiedene Anwendungsunterstützung

Orientiert für Mobile Adressierbare Endgeräte)

[AVTP⁺17](alternativen: Sigfox, Ingenu, Dash7

Klassen Kompromiss zwischen Reichweite, Performance(Latenz/ Durchsatz) und Energiebedarf

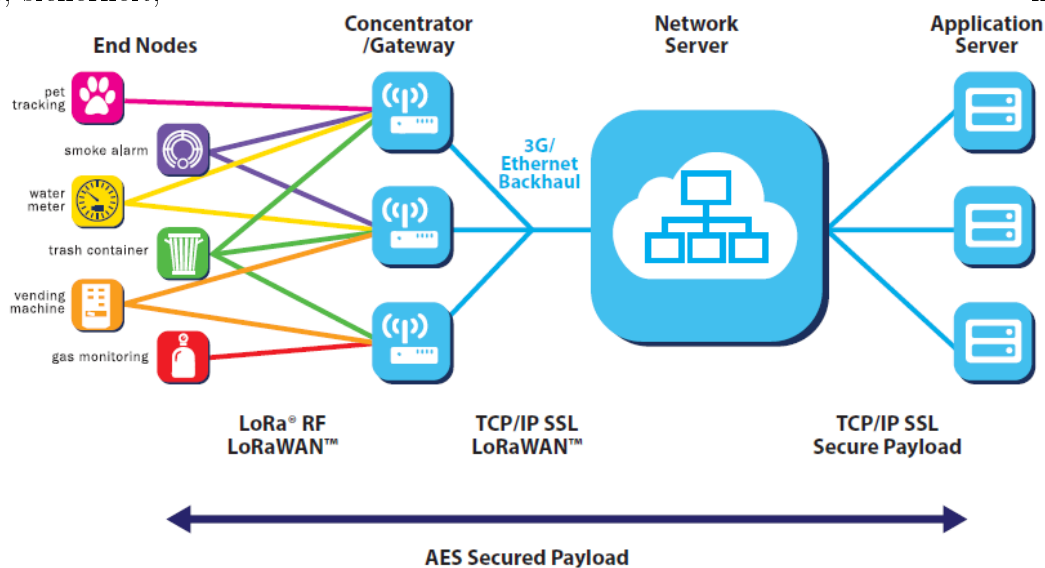
Energiesparend durch ADR (Adaptive Daten Rate))

Es wird folgen: Was ist lora, wo und wofür wird es benutzt, wie weit kann man senden und wie schnell...

2 Aufbau eines Lora-Netzwerk

Lora wird auch deswegen gerne für IoT-Geräte verwendet, weil der Netzwerkaufbau ermöglicht die über Lora verwendeten Daten im Internet abzurufen und so ohne weiteres das Gerät mit dem Internet zu verbinden. Um die von den End-Geräten gesendeten LoRa Pakete auf IP/TCP Pakete umzusetzen wird ein Gateway benötigt, das auf der einen Seite LoRa pakete empfängt/sendet und auf der anderen Seite TCP/IP Pakete verwendet. Das Gateway implementiert aber keinerlei Logic. Hierzu ist ein Netzwerks server zuständig der durch die Gateways das Netzwerk kontrolliert und steuert. Gleichzeitig stellt er die Verbindung zu einem Applicationsserver her an den er die vom Gateway empfangenen Daten sendet und von dem auch Daten an die Endgeräte, wieder über das Gateway, gesendet werden. Diese Architektur wurde gewählt um die Laufzeit der End-Geräte, Anzahl der End-Geräte, Qualität des Signals und Sicherheit des Netzwerkes möglichst hoch zu halten.

[Tec15] (Architektur hat großen Einfluss auf Batterie, Anzahl Teilnehmer, Qualität, Sicherheit, ...)



2.1 Gateway

Das Teilnetz, das aus dem Gateway und mehreren LoRa-End-Geräten besteht, ist sternförmig aufgebaut. Jedes End-Gerät kommuniziert direkt mit dem Gateway. Diese Art der Kommunikation wird auch (Single-Hop-Connection) zu Deutsch

(Einfacher-Sprung-verbindung) genannt, da die Gesendeten Daten ohne umwege an das Gateway gesendet werden. Jedes Gateway ist mit einem Netzwerkserver verbunden.

Ein Endgeräte kann gleichzeitig an mehreren Gateways senden. Der Netzwerkserver ist zuständig die Pakete auf Dublikate zu überprüfen und nur solche nur einmalig an die Applikation zu senden. Ein weiterer Vorteil ist das kein Handover nötig ist, da alle Gateways fähig sind die Daten des Endgeräts zu verarbeiten bzw. weiterzusenden.

Durch die Sternförmige Architektur des Netzes und die Fähigkeit von allen in der Reichweite befindlichen Endgeräte Daten zu empfangen, muss ein Gateway mit vielen End-Geräten kommuniziert. Da es nicht möglich ist mit jedem Gerät nacheinander zu kommunizieren, muss dies gleichzeitig geschehen. Hierzu werden adaptive Datenraten und Mehrkanal-Multi-Modem-Transceiver verwendet um eine hohe End-Geräteanzahl zu ermöglichen. Außerdem hängt die Anzahl der Teilnehmer davon ab wie geschickt die Kanäle gewählt wurden, welche Datenraten verwendet werden und wie lange gesendet werden muss um die Daten zu senden (man spricht auch von der "Time-On-Air").

Durch die genannten Eigenschaften der Gateways wird auch eine gute Skalierbarkeit erzielt. Dadurch kann ein neues Gateway die Anzahl der Knoten um das 6 bis 8-fach erhöhen.

[Tec15] (Meistens wird ein netzförmiges Netz aufgebaut. Knoten leiten Nachrichten weiter => größere Reichweite aber kompliziert, erlaubt weniger Teilnehmer und energieaufwändig).

Lora Sternförmig => Energie-effizient, Knoten senden direkt an Gateways. Gateways senden an Server, Server muss doppelte Pakete filtern, Sicherheitscheck, ACK über bestes Gateway senden, datenrate anpassen.

Keine Handover

Gateway müssen viele Geräte handeln da Stern. erreichen durch (adaptive Datenrate, multi channel/multi modem transive) mehrere Nachrichten auch verschieden Channels gleichzeitig empfangen

Wichtige Faktoren (anz. channels, datenrate(time on air), payload länge, Sendehäufigkeit)

Skaliert sehr gut => gemacht für große Nutzerzahlen Neues gateway kann Knoten 6-8 x verbessern) [SOR17](Applikation Server -> Zentraler Server(leitend Pakete weiter) -> Gateway(wandelt lorawan in ip Pakete um) -> Endgerät/Knoten)

2.2 Netzwerkserver

Der Netzwerkserver ist das "Herzstück eines jeden Lora-Netzwerkes. Ihm fallen viele Aufgaben zu. Er kann mit mehreren Gateways und mehreren Applicationsservern verbunden sein.

Die wichtigste Aufgabe ist das Steuern des Lora-Teils des Netzwerkes. Der Server verwaltet jedes End-Gerät separat indem es mit ihm den zu verwendenden Kanal aushandelt und die datenrate adaptiv kontrolliert wenn ADR(Adaptive Data Rate) verwendet wird. Weiterhin überprüft er die empfangenen Pakete auf ihre Korrektheit und Integrität und filtert Duplikate, die durch das Empfangen des gleichen Signales an verschiedene Gateways, verursacht wurden. Dabei ermittelt er auch das Gateway das den besten Empfang zum End-Gerät hat und nutzt dieses um Daten an das Endgerät zu senden. Es ist nicht immer möglich Daten direkt zu senden. Um die Applikation-Server zu entlasten puffert der Netzwerkserver die Daten und sendet sie zum nächst möglichen Zeitpunkt zu senden. Desweiteren muss er die empfangenen Pakete "bestätigen und leitet join requests an den Joinserver weiter

Eine weitere sehr wichtige Ausgabe ist es eine API für den Applikationsserver bereitzustellen um eine einfache und schnelle Kommunikation zu ermöglichen.

[SOR17](Sicherheit(zähler, ...), leitet Pakete weiter, filtert Pakete, merkt Gateway via ip verbunden, kontrolliert datenrate, Kanäle, adaptive data rate api.

2.3 Applicationsserver

Dieser Server ist zuständig den die gesendeten Nachrichten zu verarbeiten und gegebenenfalls selbst welche an die Endgeräte zu senden.

2.4 Join-Server

Ein Join-Server wird benötigt um den Beitritt mittels OTAA zu ermöglichen. Der Server kann an mehrere Netzwerkserver verbunden sein und jeder Netzwerkserver kann mehrere Join server haben. Wenn ein Endgerät dem Netzwerk beitreten möchte leitet der Netzwerkserver die anfragen an den JoinServer weiter. Dieser führt dann die nötigen schritte des Beitritts aus wie z.B. ableiten von Schlüsseln. Um dies zu tun muss ihm der NwkKey und der AppKey bekannt sein, da diese zum verschlüsseln der Nachrichten verwendet werden aber aus sicherheitsgründen nie über das Netz gesendet werden.

2.5 End-Gerät

Endgeräte sind Geräte die Informationen mittels Lora empfangen oder senden. Jedes End-Gerät ist mit einem bestimmten Applikationsserver verbunden.

Jedes Endgerät muss zur korrektur funktion mehrere wichtige Informationen Speichern.

ist das
wichtig?

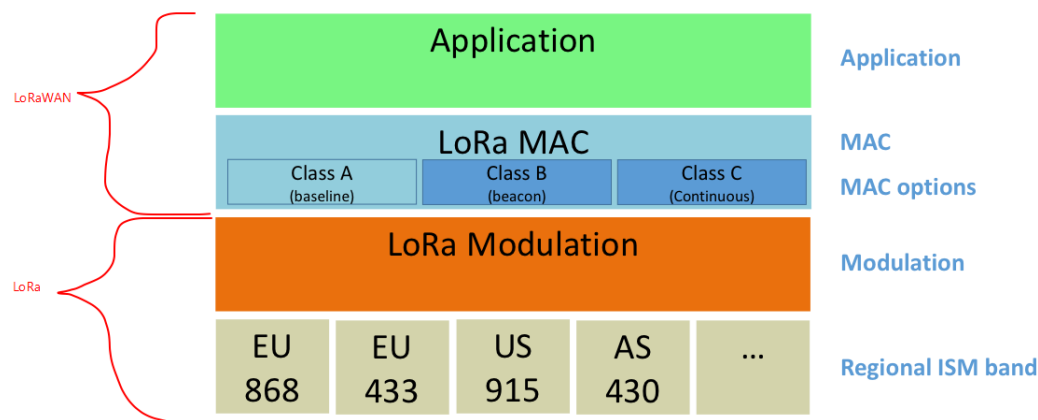
- DevEUI: Globale endgerätid die eindeutig für jedes endgerät definiert ist. (wie mac eines computers)
- JoinEUI: Globale Adresse des JoinServers an den die anfrage gehen soll. wird nur für OTAA geräte benötigt.
- NwkKey und AppKey: Werden verwendet um spätere Schlüssel abzuleiten und die kommunikation während des Joinens abzusichern. Dafür müssen sie sowohl dem Join-Server als auch dem Endgerät bekannt sein da sie nie übertragen werden.

3 LoraWan Funktionsweise

Im folgenden Kapitel wird näher auf die funktionsweise von LoRaWAN eingegangen. Speziell, liegt der fokus auf dem Netzwerkeitritt, das verwendete Protokoll und wie die Daten physikalisch Übertragen werden. [SOR17] (Geschwindigkeit ist kompromiss zwischen abstand/geschw. die untersch freuen-

zen bzw. Geschwindigkeiten beeinflussen sich nicht gegenseitig => keine Interferenz
 Die Datenrate ist einstellbar, jedoch wird die Reichweite bei höherer Datenrate gemindert. Ein Vorteil von LoRa ist, dass die einzelnen Datenraten nicht interferieren und so jedes Endgerät seine eigene Datenrate unabhängig von den anderen verwenden kann. Außerdem wird die Datenrate und die Sendeleistung für jedes Gerät separat gesteuert (ADR, Adaptive Data Rate)

3.1 Schichtenmodell



Das Schichtenmodell lässt sich in zwei Teile unterteilen. Der LoRa Teil ist der unterste und kümmert sich um die physikalische Übertragung der Pakete und der LoRaWAN Teil des Modells ist für die Steuerung des Netzwerkes, Implementierung der LoRaWAN-Klassen und das Überprüfen und Verschlüsseln der Daten zuständig.

Die unterste Schicht des LoRa Teils moduliert die verwendeten Frequenzen. In Europa muss das ISM-Band 868 verwendet werden, in den Vereinigten Staaten wird das Band 915 verwendet.

Die darüberliegende Schicht heißt LoRa Modulation und kümmert sich darum, dass die Pakete so in die Frequenz "moduliert" werden, dass der Empfänger diese korrekt und effizient empfangen kann. Mehr dazu im Kapitel Übertragungsart.

Über der LoRa Modulationsschicht liegt die erste LoRaWAN Schicht, LoRa MAC. Diese Schicht ist für die Implementierung der einzelnen Endgeräteklassen und für das Übertragen der Steuerkommandos zuständig. Mehr zu den Klassen kann im Kapitel LoRa Geräte Klassen und im Kapitel Protokoll gelesen werden.

Die oberste schiecht nennt sich Applikationsschicht und ist dafür zuständig die Nutzdaten einer NACHricht passend zu verpacken zu verschlüsseln und zu authentifizieren.

[Tec15](Application, Lora MAC, MacOptins(Classes), LoraModialtion(Regionales ISB))

3.2 Netzwerkbeitritt

End-Geräte sind immer bestimmten Netzwerken zugeordnet. Es gibt zwei wege um ein neue End-geräte zu einem bestehenden Netzwerk hinzuzufügen.

[SOR17](wei arten OTAA(Over the air activation), ABP(Activation by Personalization) Jedes gerät hat eine vorgegebene DevEUI (wie MAC adresse eines PCs), JoinEUI muss angegeben werden und adressiert den Join server

)

ABP steht für Activation by Personalization und bedeutet Wörtlich übersetzt Aktivierung durch Personalisierung.

Was
macht
der?
wohin
da-
mit?
umshreiben
!!!!

3.2.1 OTAA

Die sicherste aber auch aufwendigste methode um ein End-Gerät mit einem Netzwerk zu verbinden heißt OTAA (Over-the-Air Activation). Hierbei muss jedes mal wenn einem Netzwerk beigetreten werden soll die Join-Prozedur ausgeführt werden. Hierfür müssen folgende 4 Werte Vorgegeben werden. DevEUI, JionEUI , NwkKey, AppKey(Verschlüsselung des join requests).

Als erstes muss das End-Gerät eine join- oder rejoin-Nachricht senden. Die NACHricht besteht aus der JoinEUI, dem DevEUI und einer DevNonce. Mit der DevNonce sollen replayattacks verhindert werden. Sie ist das beim ersten Join request 0 und sollte sich bei jedem Join-Request erhöhen. Außerdem muss sie auch dann noch gespeichert werden wenn kein Strom zur verfügung speht. Falls von dem gleichen endgerät eine Join-Reguest mit einer zu kleinen DevNonce empfangen wird, wird die NACHricht ignoriert und es ist nicht möglich dem Netzwerk beizutreten.

Die Accept Nachricht besteht aus einer JoinNonce, einem NetzwerkID Net_ID,

einer Geräteadresse DevAddr, einer einstellungsfeld DLSettings , einer angabe wie lange auf eine antwort nach dem senden gewartet werden muss RxDelay und einer optionalen liste an Netzwerkparamerter CFList. Die JoinNonce wird außerdem benutzt im schlüssel wie AppSKey, ... herzuleiten. Für jedes Endgerät wird eine eigene JoinAccept nonce geführt, sie sollte sich nicht wiederholen. Jedes Endgerät merkt sich die letzte JoinNonce und tritt auch nur bei wenn diese größer ist als die letzte empfangene.

Der NWKSEKEY ist für die verschlüsselung der Datenpakete bis zu Gateway zuständig . Auch dieser Key wird vom Netzwerkserver erzeugt und muss manuell in den code eingetragen werden.

Der letzte Wert heißt APPSKEY und sichert die kommunikation vom Endgerät zu dem Applikationsserver ab. Der Schlüssel wird genau wie der NWKSEKEY vom Netzwerkserver erzeugt und verwaltet.

Mehr Informationen zu den verschiedenen Schlüsseln finden Sie in dem Kapitel Sicherheit.

Wenn der Netzwerkserver den Beitritt des Endgerätes erlaubt sendet er eine Join-Accept nachricht zurück. Das Endgerät erwartet die nachricht nach JOIN_ACCEPT_DELAY1 oder JOIN_ACCEPT_DELAY2 nach dem Senden des Request. Sollte die Join-Accept nachricht zu einem andern zeitpunkt gesendet werden, wird diese nicht empfangen weil das Endgerät nicht empfangsbereit ist. Die Nachricht enthält einstellungen für das Endgerät sowie die Id des Netzwerkes und die neue Adresse für das Endgerät. Um replayatacken zu verhindern enthält die nachricht zusätzlich eine JoinNonce. Diese wird für jedes endgerät separat geführt und muss größer sein als die zuletzt gesendete.

3.2.2 ABP

Die einfachste Art des Beitritts heist ABP was für Activation by Personalization zu deutsch Aktivierung durch Personalisierung steht. Hierbei muss lediglich vor inbetriebnahme des End-Gerätes 3 Konstanten definiert werden. Manche Hersteller "brennen" diese drei Werte fest in den chip ein, sodass er nicht geändert werden kann. Falls es nicht möglich ist dem hersteller die gewünschten

werte zukommen zu lassen, sind solche End-Geräte nicht für den Beitritt mittels ABP geeignet.

Als erstes muss die DeviceAddress angegeben werden. Diese Adresse existiert nur einmal im Netzwerk und wird verwendet um das Endgerät zu identifizieren. Die Adresse wird vom Netzwerkserver erzeugt und muss manuell von dort kopiert werden. Die verwendete Frequenz entspricht der RX1 bzw. RX2 aus dem Kapitel Klasse A.

Mit Hilfe dieser 3 Werte kann die Join-Request - Join-Accept-Prozedur übersprungen werden. Daher kann das Gerät direkt einen Lora-Netz beitreten wenn es angeschaltet wird und muss nicht erst alle Schlüssel neu ableiten und aushandeln. Allerdings ist diese Methode deswegen weniger sicher, da immer die selben Schlüssel verwendet werden.

Nach der Verbindung muss das ResetInd Mac command im FOpt-Feld gesendet werden solange bis ein ResetConf erhalten wird. Nun ist das Gerät im Netzwerk und kann unter der eingestellten Adresse und mit den eingestellten Schlüssel arbeiten.

Sobald dem Netzwerk erfolgreich beigetreten wurde werden die benötigten Schlüssel aus den vorher gesetzten Werten abgeleitet. Genaueres dazu in Kapitel Sicherheit.

3.3 Protokoll

Das LoRaWAN Protokoll ist optimiert für Batteriebetriebene Endgeräte die drahtlos kommunizieren möchten. Um energieeffizient zu sein setzt LoRa hauptsächlich auf zwei Punkte. Die Modulationstechnik und eine Adaptive Datenrate (ADR). Auch die One-Hop-Architektur trägt zur Energieeffizienz bei. Die Art wie LoRa-Signale moduliert wird in Kapitel Übertragungsart besprochen. Um die genannten Eigenschaften und das Lora-Netzwerk zu steuern werden sogenannte MAC commands verwendet. Diese werden vom Netzwerkserver oder von einem Endgerät gesendet. MAC steht hierbei für "Media Access Protokoll" und bietet die Möglichkeit die Kommunikation mit den Endgeräten, Frequenzen, Kanäle und vieles mehr zu steuern. Da die Kommandos nur für den doppeltgemoppelt

Netzwerkserver und die Endgeräte von Bedeutung sind, werden diese nicht an den Applikationsserver gesendet sondern am Netzwerkserver herausgefiltert. Im folgenden wird näher auf die MAC Kommandos und die Paketstruktur eingegangen.

Jedes Paket besteht aus grundlegend aus 2 Felder (Preamble, PHYPayload). Falls es sich um ein Uplinkpaket handelt wird noch ein CRC Code hinzugefügt (Preamble, PHYPayload, CRC). In diesem Fall spricht man von einem impliziten Paket oder von dem impliziten Modus. Impliziter Modus bedeutet, dass es kein Payload, Codierungsrate und der CRC Längenangabe gibt und diese somit eine feste zuvor definierte Länge haben. Im expliziten Modus werden noch 2 Felder hinzugefügt, PHDR und PHDR_CRC. Somit sieht ein explizites Paket folgendermaßen aus (Preamble, PHDR, PHDR_CRC, PHYPayload). Auch hier gilt, in allen Fällen eines Uplinkpaketes wird am Ende ein CRC Feld angefügt => (Preamble, PHDR, PHDR_CRC, PHYPayload, CRC).

Die Preamble ist dafür gedacht dem Empfänger mitzuteilen, dass gleich Datengesendet werden. Deswegen wird hier nur ein Signal gesendet, das ohne Informationen ist, aber von dem Empfänger wahrgenommen wird.

Da Teile des LoRaWAN Protokolls geschützt sind, finden sich über die PHDR und PHDR_CRC Felder kaum Informationen. Allerdings geht hervor, dass der PHDR die Länge des PHYPayloads und die Zieladresse beinhalten sollte. Das PHDR_CRC Feld wird benutzt, um sicherzustellen, dass die empfangenen Werte korrekt sind mittels des CRC Verfahrens.

Wie schon mehrfach erwähnt wird in Uplinknachrichten ein zusätzliches CRC Feld verwendet. CRC steht für Cyclic Redundancy Check und wird verwendet, um die Korrektheit der Nachricht zu bestätigen. PHDR, PHDR_CRC und das CRC Feld werden automatisch vom dem Funktransceiver (Modul als Empfänger und Sender) hinzugefügt.

Die bis jetzt behandelten Felder des LoRa Paketes wurden alle von der LoRa Modulationsebene erstellt.

Die darüberliegende Ebene "LoRa MAC" fügt nun das PHYPayload Feld ein. PHYPayload steht für Physikalische Payload. Es gibt 3 mögliche PHY Payloads. Entweder wird ein MACPayload eingefügt oder es werden Join-Rejoin was heißt PHYPayload?

request oder aber es wird die join accept nachricht darin transportiert. Um die Daten bzw die MAC kommandos richtig auswerten zu können und um die korrektheit überprüfen zu können werden einige Headders und zusätzliche felder benötigt. Deswegen lässt sich das Feld weiter unterteilen in (MHDR, MACPayload). Für den Fall das der MACPayload keine join-rejoin oder MacPayload nachricht ist, wir noch ein MIC feld hinzugefügt (MHDR, MACPayload, MIC). MIC steht für Message Integrity Code und wird verwendet um die korrektheit der Unterfelder MHDR | FHDR | FPort | FRMPayload festzustellen. Diese unbekannten felder werde im laufe des kapitells noch behandelt.

Das MHDR Feld beschreibt wie die Daten im MACPayload Feld zu deuten sind. Wieder wird dieses Feld in Unterfelder Unterteilt. MTType, RFU und Major heissen die Unterfelder. Das MType feld beschreibt die Art der Nachricht. z.B: kann hier angegeben werden ob es sich um Datennachrichten, Join-Nachrichten, ... handelt. RFU steht für "Reserved for Future Usage" Deutsch "für zukünftige verwendung reserviert". Daher kann dieses Feld in der version 1.1 und niedriger ignoriert werden. Im Major Unterfeld wird verwendet um das Format der Nachricht zu definieren. Momentan ist nur der wert 0 Definiert. 0 Steht für loRaWan R1. Die restlichen werde sind für zukünftige updates reserviert.

Mit der Unterteilung des MACPayload springen wir in den LoRaStack noch eine ebene höher in die Applikationsschicht. Enthalten im MacPayload feld sind der Frameheader (FHDR), der Frame Port (FPort) und der Frame payload (FRMPayload).

[SOR17](Mac commands werden benutzt um geräte zu steuern => frequenzen zu ändern, ... Application wird diese nie erhalten, läuft zwischen netzwerkserver und lora gerät ab. Verschlüsselt hier oder da. aufbau: 1byte command, x byte extra data. müssen vom empfanher acknolaged werden. Reihenfolge ist zu beachten. Alle nachrichten in einem frame müssen auch in einem frame ack werden. => Macbuffer ermöglicht dies. Wenn buffer überleuft werden die ältesten ack. Aloha

CRC usw wurden von sender erstellt und eingefügt.)

[CB⁺17](Um energieeffizient zu sein setzt LoRa hauptsächlich auf zwei

(Was passiert mit dem rest?

Punkte. Die Modulationstechnik und eine Adaptive Datenrate (ADR))

3.4 Übertragungsart

[SOR17](Knoten können zu jeder Zeit, auf beliebigen Kanälen, beliebig schnell, beliebig lange senden, solange folgende regeln befolgt werden.

- Channels werden per Pseudozufallszahl geändert
- Sendezeit erfüllt die Regionalen Bestimmungen

)

[AVTP⁺17](Chirp Signal => Zeitliche Änderung in Trägerfrequenz(höhere Frequenz als Datenrate)(positiv chirp/negativ chirp)

Datensignal wird in Chirp Signal moduliert. Resultierende Signal ist breitbandiger als Datensignal. Maximale Datenrate auch mit Rauschen erreichbar.

Durch orthogonale SSspread Factor"mehrere Signale auf einem Chanel) [Tec15](normal FSK, schon sehr effizient. Lora "chirp spread spectrum modulation". Ist wie FSK aber größere Reichweite, robuster. Stammt aus dem Militär/raumfahrt.Lora als erstes für kommerziellen billigen Einsatz.

Spread spectrum => signale sind Orthogonal für versch. spreizraten, faktoreliert mit datenrate => verschiedene Datenraten auf einem Kanal

Nähere Geräte sind schneller => höhere Datenrate => kürzere Übertragungsdauer und lassen somit mehr Zeit für andere, => bessere Batterielaufzeit. Deswegen sind symmetrische up/downlinks nötig.) Frequenzhopping, spread spectrum, code-channels

3.4.1 Adaptive Data Rate

[AVTP⁺17](Datenrate und Funkfrequenz(RF) werde passend zum Abstand angepasst

nahe Knoten => hohe Datenrate => kurze Sendezeit => weniger RF-Power kann nach Bedarf geändert werden

=> immer möglichst schnell senden => weniger Energie

) [CB⁺17](chirp signal)

4 Lora Geräte Klassen

[SOR17](GGeräte müssen mindestens A können, alle die mehr können werden auch "high class End-Devices" genannt) Vielleicht zu klein => in anderes Kapitel stopfen. Bei mehrfacher Übertragung wird nicht erhöht

Die Endgeräte sind je nach Kommunikationsart/Protokoll Art in drei Klassen (A, B und C) unterteilt.

Jede Klasse hat 3 Counter FCntUP(Pro uplink ++), FCNTDown(pro downlink außer port 0 => mach), AFCntDown(port ungleich 0 dann ++) (nur beschreiben wie diese grob funktionieren) Zähler sollen nicht flüchtig sein (Batteriewechseln kein reset) bei neuverbinden müssen alle Counter auf 0 gesetzt werden. Counter müssen auf beiden Seiten gleich gehalten werden (Synchron geführt) Wenn Nachricht empfangen ist muss der darin enthaltene Counter größer sein als der eigene.

die Counter Werte sollen so weit wie möglich nur einmal verwendet werden.

) [CB⁺17] [Tec15] (Asynchrone Knoten wegen Batterie => Event/Scheduler gesteuert verwendet ALOHA

Normal Netze müssen sich synchronisieren und Nachrichten abrufen. Lora partiell nicht => laut GSMA 3 bis 5 fach effizienter)

zur besseren Anpassung/ Anpassung an Batterie

EU: 10 Kanäle (8: 250bps bis 5.5kbps) (1: FSK 50kbps) (high rate Lora 114kbps)

)

4.1 Klasse A

Klasse A zeichnet sich durch sehr geringer Stromverbrauch aus. Die Kommunikation kann bidirektional sein, allerdings muss die Kommunikation von dem Endgerät gestartet werden. Das bietet die Möglichkeit das das Endgerät, wenn keine Daten gesendet werden müssen, in einen sehr sparsamen Schlafmodus wechselt. Um das Endgerät nicht zum Aufwachen zwingen zu müssen, wurde auf einen "Hardreset" ähnliches verzichtet. Das Endgerät kann so lange schlafen wie es möchte. Dadurch ist die Klasse A auch die

potenziell Stromsparende Endgeräteklasse.

Das Endgerät startet die Kommunikation in dem es Daten an das Gateway sendet(uplink). Daraufhin hat das Gateway die Möglichkeit 2 mal Daten zum Endgeräte senden(downlink). Da die Kommunikation asynchron stattfindet wartet das Endgeräte bis es beide uplinks empfangen hat.gewünscht.

Um zu ermitteln, wann gesendet werden darf, wird das ALOHA-Protokoll verwendet. Da das Gateway nicht immer Daten an die Endgeräte senden kann, muss es diese zwischenspeichern um diese bei der Nächsten Kommunikation zum senden. [SOR17](Es wird zwischen up- / downlink unterschieden)

4.1.1 Uplink

[SOR17] (vom Knoten zum Gateway(1..n), Lora radio packet mode, (Preamble, PHDR, PHDR_CRC, PHYPayload, CRC(cyklische Redundanz Prüfung))))

4.1.2 Downlink

[SOR17] (radio packet explicit mode, vom Gateway(1) zum Knoten(1), ausgelöst vom Netzwerkserver, auch multikasts möglich, (Preamble, PHDR, PHDR_CRC, PHYPayload) Um Nachricht kurz zu halten kein CRC am ende, nach Receiver_Delay1 / Receiver_Delay2 kann empfangen werden (rx1, rx2)

Fenster müssen lange genug für Preamble auf bleiben=> wenn erkannt wird empfangen wenn nicht fenster weider zu. Es darf nur gesendet werden wenn beide fenster zu sind. Es ist auch erlaubt andere protokole zu sprechen wenn nicht gesendet oder gehört wird.)

4.1.2.1 Rx1 [SOR17](Frequenz abhängig von Uplinkfrequenz, Datenrate abhängig on Uplinkdatenrate, wird nach Receiver_Delay 1 +/- 20 msec erwartet, Datenrate auch abhängig von Regionalen regeln, Standart: Datenrate = Uplinkdatenrate)

4.1.2.2 Rx2 [SOR17](feste Frequenz/Datenrate, nach Delay2 +/- 20 msec, Frequenz/Datenrate mittels MAC änderbar) [SOR17](Öffnungslänge

muss für Preamble ausreichen, nach RX1 + MIC und authentigitätscheck muss nicht zwingen RX2 geöffnet werden, Sender muss in einem der beiden Fenster stattfinden, Falls Downlink über beide Fenster => frames müssen gleich sein. Knoten dürfen nicht während empfangen/ zwischen RX1 und RX2 senden, ander Protokolle dürfen gesprochen werden wenn gesendet werden darf
)

4.2 Klasse B

Die Klasse B bietet bidirektionale Kommunikation mit einer deterministischen downlink Latenz. Um diese Latenz zu gewährleisten, muss die Kommunikation Synchron ablaufen. Außerdem muss festgestellt werden, ob das Endgerät bzw. das Gateway noch in Reichweite ist. Dies wird mittels eines periodischen "beacon" zu festgelegten Zeitpunkten gesendet werden realisiert.

Die Latenz ist einstellbar und kann bis zu 128 Sekunden.

Obwohl das Endgerät durch die periodischen "beacons" nicht schlafen kann, ist die Klasse B für den Batteriebetrieb gedacht. [SOR17] (wird verwendet wenn mehr Bedarf für Empfangsfenster ist. Hierzu ist ein Synchronsignal nötig => zu bestimmten Zeiten kann damit empfangen werden Gateway sendet Beacon für Synchronisation. Um Daten empfangen zu werden werden Empfangsslots => Pingslots verwendet, werden periodisch geöffnet und mittels Beacon synchronisiert. Normalerweise werden diese schnell geschlossen außer es wird etwas empfangen. Gateway dessen Beacon benutzt wird, wird nach Empfangsqualität ausgewählt. Wenn neuer/unbekannter Beacon von einem anderen Gateway empfangen wird, wird der Netzwerkservers benachrichtet und dieser entscheidet welcher verwendet wird (passt rote an).

Das Netzwerk muss die Standard Ping-slot Periode, Datenrate und Kanal kennen.

Um ein Gerät auf Klasse B zu kommen muss erst von Klasse A gewechselt werden.

Entgeräte müssen Netzwerkservers über Position informieren. Dies kann über eine leere Nachricht passieren oder eine normale (uplink).

Das beacon und die enthaltenen daten werden an die applikation geschickt. Der server kann den beacon auswerten. zwischen beacon und uplink wird random time verwendet um kolisionen zu verhindern . änderungen an pingslot- periode .. muss mitgeteilt werden. Hierzu ist klasse A nötig => wechel zu A, wechel zu B.

Beacon wird genutzt um clockdrift auszugleichen. Wenn kein beacon empfanen wird => Bacenless mode. Dieser wird bis zu 2 stunden beibehalten. Reines verlassen auf interne Uhr. Wenn beacon empfagne wird, wird zeit zurückgesetzt.)

Nachschuen
wie ge-
nau
das
funk-
tio-
niert

4.2.1 Klassenwechsel A nach B

[SOR17](Endgerät fart LoRaWAN layer an. Layer sucht beacon. Mac command DeviceTimeReq um schneller bacon zu bekommen nutzen. Danach wird das ClassB feld auf 1 gesetzt. Bei den geöffneten fenstern wrd der maximal mögliche clockdrift berücksichtigt. Downlink läuft wie bei A ab.)

4.2.2 Uplink

[SOR17](Wie bei)

4.2.3 Downlink

[SOR17](wie bei A, frequenzplan kann sich unterschieden. AUch Multikasts möglich,)

4.2.3.1 Singelcast [SOR17]()

4.2.3.2 Multicast [SOR17](sepearate Adresse für Multicast Festgelegt durch layer oder manuell für gruppenmulticast Nicht führ MAC geeignet,)

4.3 Klasse C

Um eine möglichst geringe/keine Latzen zu erzielen ist die Klasse C gemacht. Dies bedeutet aber auch das der Stromverbrauch am höchsten ist und somit

nicht für den Batteriebetrieb geeignet. Das Gateway kann immer Daten senden außer wenn das Endgerät gerade Daten sendet. Hier sind Geschwindigkeit von bis zu 50mb möglich.

Es ist auch möglich während des Betriebes eines Endgerätes die Klasse zu wechseln. Dies wird am häufigste zwischen A und B getan/ ist nur zwischen A und B möglich. [SOR17](öffnet RX1 und RX2 fenster wie in Klasse A. Immer wenn nicht gesendet wird oder RX1 offen ist, ist RX2 offen. Multicast ist auch möglich.)

5 Sicherheit

Lora bietet die end-to-end Sicherheit an, indem es die Signale zweimal verschlüsselt.

Die erste Verschlüsselung dient dazu die gesendeten Daten vor eventuellen Mithörern zu verschlüsseln. Die Verschlüsselung geschieht mit einem 128-bit Network-Session-Key.

Die zweite Verschlüsselung wird bis zur endgültigen Weiterverarbeitung der Daten auf z.B. einen Server verwendet und ist ein 128 bit Application-Session-Key.

Das zur Verschlüsselung verwendete Protokoll ist AES. Auch zu Authentifizierung und zur Überprüfung der Integrität wird AES verwendet. [GAS17] [Tec15](Applikationsverschlüsselung(schutz der Daten for mitlesen) Netzwerk(Autentiizierung der Knoten) AFS, Key Exnage IEEE EU164) [SOR17](symetrischer Schlüssel => nur einer benötigt, Sessionkey ist abgeleited von Knoten-rootkey. JoinServer setllt verbindung der Keys her.)

6 Live-Beispiel

wenn vorhanden.

7 Ausblick

Literatur

- [AVTP⁺17] ADELANTADO, FERRAN, XAVIER VILAJOSANA, PERE TUSET-PEIRO, BORJA MARTINEZ, JOAN MELIÀ-SEGUÍ und THOMAS WATTEYNE: *Understanding the Limits of LoRaWAN*. <https://ieeexplore.ieee.org/abstract/document/8030482>, September 2017. Eingesehen am 09.04.2019.
- [CB⁺17] CHEONG, PHUI SAN, JOHAN BERGS, , CHRIS HAWINKEL und JEROEN FAMAHEY: *Comparison of LoRaWAN Classes and their Power Consumption*. <https://ieeexplore.ieee.org/abstract/document/8240313>, November 2017. Eingesehen am 09.04.2019.
- [GAS17] GEMALTO, ACTILITY und SEMTECH: *LoRaWANTM SECURITY WHITE PAPER PREPARED FOR THE LoRa ALLIANCETM*. <https://lora-alliance.org/resource-hub/lora-alliance-security-whitepaper>, Februar 2017. Eingesehen am 09.04.2019.
- [SOR17] SORNIN, N. (Herausgeber): *LoRaWANTM 1.1 Specification*. Lora-Alliance, <https://tools.ietf.org/pdf/rfc8376.pdf>, 1.1 Auflage, Oktober 2017. Eingesehen am 09.04.2019.
- [Tec15] TECHNICALMARKETINGWORKGROUP1: *A technical overview of LoRa® and LoRaWANTM*. <https://lora-alliance.org/resource-hub/what-lorawantm>, November 2015. Eingesehen am 09.04.2019.