Seminararbeit: Lorawan

Tobias Sigmann

23. Mai 2019

Inhaltsverzeichnis

1 Einführung in Lora

Lora ist ein Low Power, Wide Area (LPWA) Netzwerkprotokoll und somit sehr gut für batteriebetriebene kabellose Geräte geeignet. Deswegen wir Lora auch of 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, sodas 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, hierrauf 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, Interreferenzrobustheit, Anzahl Konten, Sicherheit, bidirektionale Kommunikation, verschiedene Anwendungsunterstützung

Orientiert für Mobile Adressierbare Endgeräte)
[AVTP+17](alternativen: Sigfox, Ingneu, Dash7

Klassen Kompromiss zwischen Reichweite, Performance(Latzen/ 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 Logik. Hierzu ist ein Netzwerkserver zuständig der durch die Gateways das Netzwerk kontrolliert und steuert. Gleichzeitig stellt er die Verbindung zu einem Applikationsserver her, in dem er die vom Gateway empfangenen Daten Weiterleitet.

Der Applikationsserver ist zuständig den die gesendete Nachrichten zu verarbeiten und gegebenenfalls selbst welche an die Endgeräte zu senden.

Diese Architektur wurde gewählt um die Laufzeit der Akku betriebenen Endgeräte, Anzahl der Endgeräte, Qualität Signals und Sicherheit des Netzwerkes möglichst hoch zu halten. [Tec15, S. 8 ff.]

2.1 Gateway

Das Teilnetz das aus dem Gateway und mehreren LoRa-Endgeräten besteht ist Sternförmig aufbau. Jedes Endgeräten 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 mindestens einem Netzwerkserver verbunden.

Ein Endgerät kann gleichzeitig an mehreren Gateways senden. Der Netzwerkserver ist zuständig die Pakete auf Duplikate zu überprüfen und nur einmalig an die Applikationsserver zu senden. Ein weiterer Vorteil ist das kein Übergabe der Endgeräte bei Standortwechsel zu andern Gateways nötig ist. Dadurch müssen die Gateways mit vielen Endgeräten kommuniziert. Um diese hohe Endgeräteanzahl zu ermöglichen wurde darauf verzichtet mit jedem Endgerät einzelne zu kommunizieren und stattdessen auf eine Parallele Kom-

munikation gesetzt. Hierzu werden adaptive Datenraten und Mehrkanal-Multi-Modem-Transceiver verwendet.

Durch die genannt Eigenschaften der Gateways wird eine gute Skalierbarkeit erzieht. Dadurch können neue Gateways die Anzahl der Endgeräte um das 6 bis 8-fach erhöhen.vgl. [Tec15, S.10]

2.2 Netzwerkserver

Der NetzwerkServer ist das "Herzstück" eines jeden Lora-Netzwerkes. Er kann mit mehreren Gateways und mehreren Applikationsserver verbunden sein.

Die wichtigste Aufgabe des Netzwerksserver ist das Steuern des LoRa-Teils des Netzwerkes. Der Server verwaltet jedes Endgerät separat indem es mit ihm den zu verwendenden Funkkanal Aushanelt und die Datenrate kontrolliert wenn ADR(Adaptiv Data Rate) verwendet wird. Außerdem ist er bei dem Netzwerkbeitritt eines Endgerätes .beteiligt.

Weiterhin überprüft er die empfangen Pakete auf ihre Korrektheit, Integrität und filtert Duplikate, die durch das Empfangen der gleichen Übertragung von einem Endgerät an verschieden Gateways, verursacht wurden. Dabei ermittelt er auch die Gateways, die den besten empfang zu den jeweiligen Endgeräten hat und nutzt dieses um Daten an die Endgeräte zu senden.

Es ist nicht immer möglich Daten direkt zu senden, da die Endgeräte nur manchmal empfangsbereit sind. Um die Applikationsserver zu endlasten, puffert der Netzwerkserver die Daten und sendet diese zum nächst möglichem Zeitpunkten.

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

2.3 Join-Server

Der Server kann mit mehreren Netzwerkservern verbunden werden und jeder Netzwerkserver kann mehrere Join-Server haben.

Ein Join-Server wird benötigt um den Beitritt mittels OTAA zu ermöglichen. Mehr zu OTAA kann in dem Kappitel OTAA gelesen werden. Wenn ein Endgerät dem Netzwerk beitreten möchte, leitend der Netzwerkserver die Anfragen an den Join-Server weiter. Dieser führt dann die nötige schritte des Beitritts aus wie z.B. ableiten von Schlüsseln oder Senden der nötigen Einstellungen. 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 Netzt übertragen werden dürfen. [SOR17b, S. 9 f.]

2.4 End-Gerät

Endgeräte sind Geräte die Informationen mittels LoRa empfangen oder senden. Jedes Endgerät ist mit einem bestimmten Applikationsserver verbunden.

Jedes Endgerät muss zur korrekten Funktion mehrere wichtige Informationen speichern.

- DevEUI: Globale Endgeräte_ID die eindeutig für jedes Endgerät definiert ist. Vergleichbar mit der MAC-Adresse eines TCP/IP Gerätes.
- JoinEUI: Globale Adresse des Join-Servers 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 der Beitrittprozedur in ein Netzwerk abzusichern. Dafür müssen sie sowohl dem Join-Server als auch dem Endgerät bekannt sein da sie nie übertragen werden.

[SOR17c, S.47 ff.]

3 LoraWan Funktionsweise

Im folgenden Kapitel wird näher auf die Funktionsweise von LoRaWAN eingegangen. Speziell, liegt der Fokus auf dem Netzwerkebitritt, das verwendete Protokoll und wie die Daten physikalisch Übertragen werden.

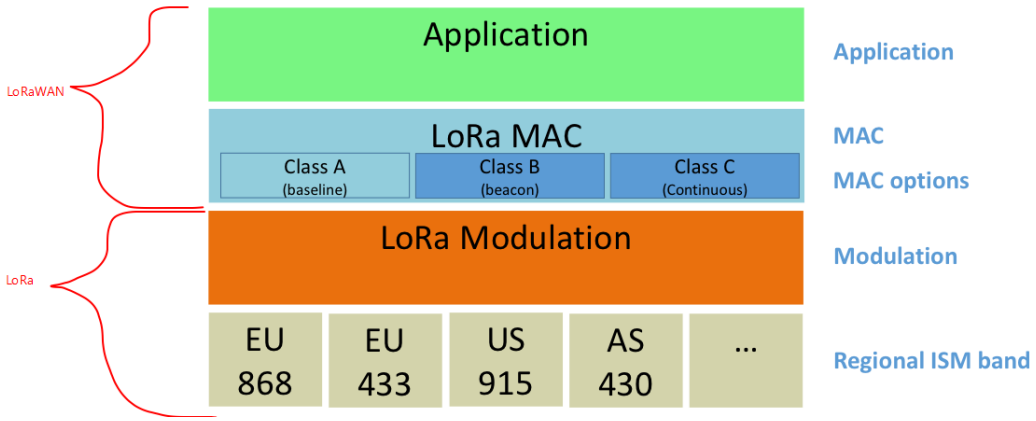

Abbildung 1: LoRaStack [Tec15, S.7]

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 Pakte und er LoRaWAN Teil des Modells ist für die Steuerung des Netzwerkes, Implementierung der LoRaWAN-Klassen und das überprüfen / verschlüsseln der Daten zuständig.

Die unterste Schicht des LoRa Teils ist für die Anwendung der Richtigen Frequenzen zuständig. In Europa muss das ISM-Band 868 verwendet werden in den Vereinigten Staaten wird das Band 915 verwendet. [Tec15, S.7]

Die darüber liegende 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 und wiederherstellen kann. Mehre dazu im Kapitel Übertragungsart

Über der LoRa Modulation Schicht liegt die erste LoRaWan Schicht, LoRa MAC. AC steht für “Media Access Protokoll”. Dieses Protokoll wird verwendet um das Loranetz zu steuern und wie der Name schon sagt, Daten zu übertragen . Diese Schicht ist außerdem für die Implementierung der einzelnen Endgeräteklassen und für das Übertragen der Steuerungskommandos zuständig. Mehr zu den Klassen kann im Kapitel namerefsec:klassen und im Kapitel Protokoll gelesen werden.

Die oberste Schicht nennt sich Applikationsschicht und ist dafür zuständig

die Nutzdaten einer Nachricht passend zu verpacken, zu verschlüsseln und zu authentifizieren.

3.2 Netzwerkbeitritt

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

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 Konstanten Vorgegeben werden. DevEUI, JionEUI , NwkKey, AppKey. erklären

Näheres zu dem DevEUI und JoinEUI kann im Kapitel **??** gefunden werden. der

Der NWKSKEY ist für die Verschlüsslung der Datenpakete bis zu Gateway zuständig. Auch dieser Key wird vom Netzwerkserver erzeugt und muss manuell in den Code eingetragen werden. [GAS17, S.3] schlüssel?

Der letzt Wert heißt APPSKEY und sichert die Kommunikation vom End-Gerät zu dem Applikationsserver ab. Der Schlüssel wird genau wie der NWKSKEY vom Netzwerkserver erzeugt und verwaltet. [GAS17, S.3]

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 Replayattack verhindert werden. Diese Nonce ist das beim ersten Join-Request 0 und sollte sich bei jedem Join-Request erhöhen. Unter anderem deswegen muss sie auch dann noch gespeichert werden wenn kein Strom zur Verfügung steht um die Nonce nicht nach jedem ausschalten zurückzusetzen. Falls von dem gleichen Endgerät eine Join-Request mit einer zu kleinen DevNonce gesendet wir, 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, einem Einstellungsfeld DLSettings , einer Zeit-

angabe wie lange zukünftig auf eine Antwort nach dem senden gewartet werden muss, hier RxDelay und einer optionalen Liste an Netzwerkparameter CFList.

Die JoinNonce wir verwendet um Replayatacken zu verhindern und muss größer sein als die zuletzt gesendete um von dem Endgerät verarbeitend zu werden. Außerdem wir die Nonce benutz um Schlüssel wie AppSKey herzuleiten. Für jedes Endgerät wird eine eigene Join Nonce geführt, sie sollte sich nicht wiederholen. Jedes Endgerät merkt sich die letzte JoinNonce und tritt auch nur bei wenn diese großer ist als die letzte empfangene.

Die Join-Accept Nachricht wird vom Endgerät nach JOIN_ACCEPT_DELAY1 oder JOIN_ACCEPT_DELAY2 nach dem Senden des Request erwarted. Sollte die Join-Accept Nachricht zu einem andern Zeitpunkt gesendet werden, wird diese nicht empfangen, da das Endgerät nicht empfangsbereit ist.

Mehr Informationen zu den Ableitungen der Schlüssel finden Sie in dem Kapitel Sicherheit. erklären

3.2.2 ABP

Die einfachste Art des Beitritts heißt 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 lasse, sind solche End-Geräte nur schlecht bis gar nicht für den Beitritt mittels ABP geeignet.

Als erstes muss die DevAdr(Geräteadresse) 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 Manuel von dort kopiert werden.

Mit Hilfe dieser 3 Werte kann die Join-Prozedur übersprungen werden. Daher kann das Endgerät direkt einem 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 dieselben Schlüssel ver-

wendet werden.

Nach Beitritt muss das ResetInd Mac Kommando im FOpt Feld gesendet werden gesendet werden solange bis ein ResetConf Kommando erhalten wird. Nun ist das Gerät im Netzwerk und kann unter der eingestellten Adresse und mit dem eigestellten Schlüssel arbeiten. [SOR17c, S. 64]

3.3 Protokoll

Das LoRaWAN Protokoll ist optimiert für Batteriebetrieben Endgeräte die drahtlos kommunizieren möchten. Um energieeffizent zu sein setzt LoRa hauptsächlich auf zwei Punkte. Die Modulationstechnick und eine Adaptive Dtanenrate (ADR). Auch die Öne-Hopärchitektur trät zur energieeffizenz bei. Die Art wie LoRa siganle Moduliert wird in Kapittel Übertragungsart besprochen. [CB+17, S,1 f]

Damit der Netzwerkserver das LoRa-Netzt steuern kann wurden, wurden Mac-Kommandos eingesetzt. Mit diesen Kommandos lassen sich wie schon gesehen, dem Netzwerk beitreten, mit dem Endgerät kommunizieren und Frequenzen, Kanäle und vieles mehr zu steuern. Da die Kommandos nur für den Netzwerkserver und die Endgeräte von Bedeutung sind, werden diese nicht an den Applikationsserver gesendet sonder vom Netzwerkserver herausgefiltert. Im Folgenden wird näher auf die MAC-Kommandos und die Paketstruktur eingegangen. Ein Uplink ist ein Paket dass vom Endgerät, das Bildlich gesprochen "unter" dem Gateway sitzt, an das Gateway gesendet wir. Ein Downlink ist dem entsprächet ein Paket das vom Gateway an den Server gesendet wird.

3.3.1 MAC-Kommandos

3.3.2 LoRa-Paketstruktur

Die Paketstruktur kommt wie beim ISO/OSI Schichtenmodel durch das "durchlaufen" des Stacks zustande. Da im Folgenden die Paketstruktur vom Groben ins Feine Betrachtend wird, werden hier als erstes die Felder der Modulationsschicht betrachtet.

Jedes Pakete besteht aus grundlegend aus 2 Felder Präambel und PHYPalyload Falls es sich um einen Uplink-Paket handelt wird noch ein CRC Code hinzugefügt: Preamble, PHYPalyload, CRC. In diesem Fall spricht man von einem implizit Paket oder von dem implizitem Modus. Implizit Modus bedeute dass es kein Payload Header gibt, der Felderlängen oder CRC längenangebe angibt. Diese sind somit feste zuvor definierte. Im expliziten Modus werden noch 2 Felder hinzugefügt, PHDR und PHDR_CRC. Somit sieht ein expliziteste Paket folgendermaßen aus: Preamble,PHDR, PHDR_CRC, PHYPalyload. Auch hier gilt, im Falle eines Uplink-Paketes wird am Ende ein CRC Feld angefügt. Somit ergibt sich folgende Paketstruktur: Preamble, PHDR, PHDR_CRC, PHYPalyload, 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 werden kann.

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 korket sind. Dies wird mittels des CRC Verfahrens überprüft.

Wie schon mehrfach erwähnt wird in Uplink-Nachrichten ein zusätzliches CRC Feld verwendet. CRC steht für Cyclisch Redundanz Check und wird verwendet um die Korrektheit der Nachricht zu bestätigen. PHDR, PHDR_CRC und das CRC Feld werden automatisch vom dem Funktransceiver (Modul aus Empfänger und Sender) hinzugefügt.

Die LoRa MAC ebene fügt nun das PHYPalyload Feld ein. PHYPalyload steht für Physikalische Payload. Es gibt 3 Mögliche PHYPayloads Entweder wird ein MACPaylod eingefügt, Join-Rejon-Request oder aber es werden 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 Headers und zusätzliche Felder benötigt. Deswegen lässt sich das Feld PHYPalyload weiter unterteilen in MHDR und MACPayload. Für den Fall das der MACPaylod eine Join-rejoin oder MACPayload Nach-

richt ist, wir noch ein MIC Feld hinzugefügt(MHDR, MACPayload, MIC). MIC steht für Message Integrity Code und wird verwendet um die Korrektheit der des MACPayloads und des MHDR festzustellen.

Das MHDR Feld beschreibt wie die Daten im MACPayload Feld zu deuten sind. Wieder wird dieses Feld in Unterfelder Unterteilt. MType, 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 Usag" zu Deutsch "für zukünftige verwendeung reservier".Daher kann dieses Feld in der version 1.1 und niedriger ignoriert werden. Das Major Unterfeld wird verwendet um das LoRa-Version 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 dem 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. Daten die gesendet werden sollen befinden sich in dem FRMPayload Feld. Wenn keine Daten gesendet werden, kann das FRMPayload Feld auch MAC-Kommandos enthalten. In dem Feld FPorts wird angegeben an welchen Port und somit an welche Teilapplikation des Applikationsserver die Daten geleitet werden sollen. Es gibt einige feste Ports. Port 0 ist reserviert um MAC-Kommandos im FRMPayload Feld entgegenzunehmen. Die Ports 0x01 bis 0xDF sind Anwendungsspezifische Ports und Port 0xE0 ist für das LoRaWAN Test Layer Protokoll reserviert. Falls ein anderer Port als die geraden genannten angegeben wird, wird die Nachricht verworfen.

Erneut kann der FHDR "Frame Headerïn einzelne Felder unterteilt werden in DevAddr, FCtrl, FCnt, Fopts. In dem Feld DevAddr wird die Zieladresse der Nachricht vermerkt. Im Feld FCnt (Frame Counter) wird der jeweilige counterwert für die bisher gezählten Nachrichten übermittelt. Hamit schützt man sich vor Replayattacks. Mehr zu den Counter kann im Kapitel Sicherheit gelesen werden. Im FOpt Feld können bis zu 5 MAC Kommandos parallel zu Daten übermittelt werden. Die Anzahl kommt auf die Menge der mitgelieferten

Variablen an.

Das Letzte Feld das in Unterfelder unterhielt werden kann ist das FCtrl Feld. Hier wird das Verhalten des Gerätes gestiert sowie Nachrichten bestätigt. Es gibt leichte unterschiede für ein Uplink und für Downlink Nachrichten. Beide Nachrichtentypen haben ein ADR, ein ACK und ein FOptsLen Feld. Im ADR wird definiert ob der Sendende bereit ist im Modus Ädaptive Data Rate"Daten zu senden, siehe KapitelAdaptive Data Rate. Mit dem ACK Feld können empfangene Nachrichten bestätigt werden. Ob Nachrichter bestätigt werden müssen steht im MType Feld(Confirmed Data). In dem FOptsLen Feld wird die Länge des FOpts Feldes mitsamt des Headers eingetragen Wenn Das FOptsLen 0 ist, ist kein FOpts Feld vorhanden.

Ein Downlinkpaket hat zusätzlich ein RFU Feld das nicht verwendet wird und ein FPending Feld. In diesem Feld kann das Gateway bzw. der Netzwerkserver dem Endgerät mitteilen, dass noch mehr Daten zu senden sind.

Dahingegen hat ein Uplinkpaket ein ClassB Feld indem das Endgerät dem Gateway mitteilt, dass es gerne auf Funktionsklasse B wechseln würde und ein ADRACKReq Feld. Dieses Feld wird verwendet um zu überprüfen ob das Netzwerk noch antwortet. Die genaue Funktionsweise ist im Kapitel Adaptive Data Rate erklärt.

Eine noch genauere Darlegung der LoRa-Paketstruktur kann in den LoRaWA 1.1 Specification [SOR17c] gefunden werden.

Tabellen einfügen, belegen maccomandos einfgen zuerset

3.4 Übertragungsart

Um die Entstandenen Pakete in Signale umzusetzen und diese effizient und gleichzeitig übertragen zu können nutzt LoRa Chirp-Spread-Spectrum (CSS). Hierbei werden die Frequenz über eine gewisse Zeit hinweg veränder. Durch messen in welche richtig, ansteigen oder abfalle, die Frequenz verändert wird, können 1 und 0 Codiert werden. Man spricht bei einem Bit von einem Chrip-Impuls. Durch aneinanderreihe der verschiedenen Impulsen ist es möglich mehrere Bits nacheinander zu übertragen. Das entstandene Signal wird auch al Sub-Chrip bezeichnet. Durch verwenden von Unterschiedlichen ansteigezeiten

und abfallzeiten ist es möglich mehre Signale auf der selben Frequenz zu übertragen ohne das die Signale sich gegenseitig stören. Dies nennt man auch Spreading Factor. Außerdem kann die Parallelität durch verschiedene Frequenzbereiche verbessert werden. CSS ist besonders für große Reichweiten geeignet und somit auch bestens für Lora. Am besten ist das Signal wenn das Endgerät nahe am Gateway ist. Je weiter es entfernt desto schlechter wird das Signal. Um die Kommunikation trozdem zu ermöglichen wird der "Spreading Factor" erhöht. Dies hat auch den Vorteil dass der Energieaufwand gering gehalten werden kann. Analog wie Menschen auf einer Party nicht immer versuchen lauter sonder besonders langsam und deutlich sprechen. [SOR17a]

Diese Aufteilung durch den Spreading Factor und die Frequenz werden Channels erzeugt. Channels können beliebig benutzt werde. es gibt allerdings mpssen zwei regeln zu beachtet werden:

1. Channels werden per Pseudozufallszahl geändert
2. Sendezeit erfüllt die Regionalen Bestimmungen

Das Aloha Protokoll wird verwendet um festzustellen wann gesendet werden soll. Dabei wird einfach gesendet wenn Daten zum Senden vorhanden sind. Wenn nun zwei Sender gleichzeitig auf dem selben Channel senden möchten kommt es zu einer Kollision. Dadurch kann das Gateway die empfangenen Daten nicht mehr auswerden und die Daten müssen erneut übertragen werden. Deswegen warten beide Endgeräte eine zufällige, unterschiedliche Zeit ab bist sie erneut senden.

frequenzy hopping

3.4.1 Adaptive Data Rate

Adaprive Data Rate oder kurz ADR wird verwendet um immer die optimalste Senderate und die optimale Sendepower für das Endgerät zu finden und so schnellstmöglich die Daten zu senden. ADR kann nur verwendet werden wenn im FHDR Feld des LoRa-Paketes das ADR Bit gesetzt ist, siehe Protokoll.Die Steuerung durch ADR findet durch den Netzwerkserver statt. Sobald der Netzwerkserver bereit ist, setzt er das Bit im Downlink-Paket. Ist das

Endgerät ebenfalls bereit setzt es ebenfalls das Bit und ADR kann verwendet werden. Falls es nicht möglich sein sollte ADR zu verwenden sollte es durch das Applikationsslayer gesteuert werden.

Die Steuerung findet durch spezielle MAC-Kommandos statt. Standardgemäß wird die höchste Übertragungsstärke verwendet und die geringste Übertragungsrate. Falls diese gedrosselt werde soll wird vom Netzwerkserver das LinkADRReq MAC –Kommando benutzt. Mit diesem wird das Endgerät informiert, dass es die Übertragungsstärke, Übertragungsrate oder den Übertragungskanal ändern soll. Die Werte sind in den Parameter codiert. Sobald die Werte geändert wurden, muss periodisch überprüft werden ob das Netzwerk die Nachrichten noch bekommt. Deswegen wird jedes Mal wenn der wenn ein Uplink empfangen wird, wird der ADR_ACK_CNT Zähler erhöht. Wenn dieser Zähler ein gewissen schwellenwert (ADR_ACK_Limit) überschreitet, wird das ADRACKReq Bit im Uplink gesetzt. Dieses signalisiert den Netzwerkserver das er mit einem ein Nachricht senden muss um die Verbindung zu bestätigen. Falls dieser Downlink nicht in ADR_ACK_Delay Frames empfangen wird, wird zuerst die Übertragungsstärke auf das Maximum gesetzt. Falls möglich wird außerdem die Datenrate verringert um die Reichweite zu erhöhen. Die Datenrate wird solange weiter, jede ADR_ACK_Delay Frames, verringert bis diese minimal ist. Falls diese schon minimal ist müssen alle Kanäle benutzt werden. Dies wird solange probiert biss eine Verbindung hergestellt werden kann. [SOR17c, S.19 f]

4 Lora Geräte Klassen

Um maximal energie zu sparen aber trozdem die möglichkeit dass die endgeräte agiel Daten empfangen können wurden die Geräteklassen eingeführt. Das Hauptmerkmal der Klassen sind die unterschiedlichen empfangsmodien. Es gibt 3 Klassen, A, B und C. Die Klasse A muss standartgemäß von jedem Endgerät implementiert werden. B und C sind Optional und müssen nicht vorhanden sein . Alle Geräte die mehr als A können werden als "high class End-Devices"genannt.

Joinen nur in A beschrieben? wohin mit

[SOR17c](Geräte müssen mindestens A können, alle die mehr können werden auch "high class End-Devices"genannt) Vieleicht zu klein => in anderes Kapitel stopfen. Bei mehrfacher übertrageung 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 auser port 0 => mach), AFCntDOwn(port ungleich 0 dann ++) (nur beschreiben wie diese grob funktionieren) Zähler sollen nich flüchtig sein(Batteriewechseln kein reset) bei neuverbinden müssen alle counter auf uf 0 gesetzt werden. counter müssen auf beiden seiten glich gehalten werden(Synchron geführt) Wenn nachricht empfagen ist muss der darin enthaltenene counter größer sein als der eigene.

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

) [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 wird auch (All end-Devicec) gennnt undzeichnet sich durch sehr geringer Stromverbrauch aus. Die Kommunikation kann bidirektionalen Stadtfinden, allerdings muss die Kommunikation von dem Endgeräte gestartet werden. Das bieted die möglichkeit das das Endgerät, wenn keine Daten gesendet werden müssen, in einen sehr sparsamen Schlafmodus wechselt. Um das Endgeräte nicht zum äufwachenżwingen zu müssen, wurde auf einen Hardbeat oder ähnliches verzichtet. Dadurch kann das Endgerät so lange ßchlafen"wie es möchte. Somit ist die Klasse A auch die potenziell Stromsparende Endgeräteklasse.

Die Klasse A erlaubpt auserdem das das Endgerät andere Protokolle schickt solange es keine LoRa Daten sendet oder empfängt.

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). Die Downlinkfenster werden RX1 und RX2 genannt. Da die Kommunikation asynchron Stadtfinden, muss das endgerät wartet bis die uplinkphase abgeschlossen ist.

Die empfangsfenserr RX1 und RX2 mussen mindestens solange geöffnet bleiben das sie eine beginnende Übertrgung feststellen können. Flass keine übertragung epfangen wird, wird das fenseter weider geschlossen. Anderenfalls werden die Dtaen empfangen. Das Empfangsfenster RX1 wird nach RECIEV_DELAY1 zeiteiheiten +/- 20msec nach beendigung des Upliks geöffnet. Es wird die selbe freuqenz und Datenrate verwendet die auch be den uplik verwendet wurde. Wenn festgestellt in RX1 festgestellt wurde das keine Weiteren Daten mehr epfangen werden müssen kann auf das öffnen des RX2 fensters auch verzichted werden. RX2 wird wird nach RECIEV_DELAY2 zeiteiheiten +/- 20msec nach beendigung des Upliks geöffnet. Allerdings ist die Datenrate und freuenz fest. Nur Mittels spezieller MAC commands kann dies verändert werden.

Für alle join / rejoin aktivitäten wird immer die Klasse A verdendet . Schon erklärt oder woanders

[SOR17c](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 Reciever_Delay1 / Reciever_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 erlabut andere protokole zu sprechen wenn ncht gesendet oder gehört wird.<==) [SOR17c](Frequenz abhänig von Uplinkfrequenz, Datenrate abhängig on Uplinkdatenrate, wird nach Reciever_Delay 1 +/- 20 msec erwarted, Datenrate auch abhänig von Regionalen regeln, Standart: Datenrate = Uplinkdatenrate) [SOR17c](feste Frequenz/Dtanetnrate, nach Delay2 +/- 20 msec, Frequenz/Datenrate mittels

MAC änderbar) [SOR17c](Öffnungslänge muss für Preamble ausreichen, nach RX1 + MIC und autentigitätscheck muss nicht zwingen RX2 geöffnet werden, Sender muss in einem der beiden Fenster stattfinden, Falls Downlink über beide Fenster => feames müsen gleich sein. Knoten dürfen nich während empfangen/ zwischen RX1 und RX2 senden, ender Protokolle dürfen gesprochen werden wenn gesendet werden darr)

4.2 Klasse B

Die Klasse B (B für BEACON) bietet bidirektionale Kommunikation mit einer deterministischem 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 einens periodischem "beacon"die zu festgelegten. DIeser BAcon wird regelmäßig vom GAteway gesendet und dint der syncronisation der Endgeräte. Zeitpunkten gesendet werde realisiert. Die Latenz ist einstellbar und kann bis zu 128 Sekunden. Die Endgeräte öffnen in regälmasigen ubständen ein empfangsfenster das pingslot genannt wird. Ein Downlink der in einem Pingslor gesendet wird wird ping genannt. Da dimmer Das gateway mit dem besten empfang die Daten an das GAteway sendet, muss das Endgerät selbständig feststellen wenn es einen Bacon mit einer unbekannten ID bekommt und durhch eien uplink dem server mitteilen das es in ierne neuen Umgebung ist. Dadurch lernt der seerver wo sich das Entgerät befindet und kann das Gatewa mit dem besten empfang wählen.

Obwohl das Endgerät durch die periodischen "beacons"nicht ßchalfen"kann, ist die Klasse B für den Batteriebetrieb gedacht.

[SOR17c](wird verwendet wenn mehr bedarf für empfangsfenster ist. Hierzu ist ein synchronsignal nötig=> zu bestimmten zeiten kann damit empfangen werden Gateway sended Beacon für synchrinsation. Um daten empfangen zu werden werden empfangsslots => pingslots verwendet, werden periodisch geöffneto und mittels beacon synchronisiert. Normalerweise werde diese schnell geschlossen außer es wird etwas empfagne. Gateway dessen beacon benutzt

wird, wird nach empfangsqualität ausgewählt. Wenn neuer/unbekanter Beachon von einem anderen GAteway empfangen wird, wird der netzwerkserver benachrichted und dieser entscheidet welcher verwendet wird()passt rozte an).

Das Netzwerk muss die standart ping-slot periode Datenrate und kanal kennen.

Um ein gerät auf klasse B zu kommen muss erst von Klasse A gewächselt werden.

Entgeräte müssen Netzwerkserver über position nformieren. Dies kann über eine leere nachricht passieren oder eine normale(uplink).

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

Nachschuen wie genau

Beacon wird genutzt um clockdrivt 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.)

das funktioniert

4.2.1 Klassenwechsel A nach B

Um einen Wechel überhaupt zu ermöglichen muss der Netzwerkserber die devault ping-slot periodem die pingslot datenrate und den Pingslot channel kennen.

Ale endgeräte treten in Klasse A dem Netzwerk bei. Das wecheln in die klasse B wird durch folgenden Prozess realisiet.

Als erstes muss das Programm des ENdgerätes beim LoRaWAN layer anfrogen ob es möglich ist in klasse B zu wecheln. Der LAyer sucht nun nach einem baecon. Wird ein backen entdeckt, wird die BEACON_LOCKED Servisprimitive zurückgeliefert. Wenn kein BAckoun empfangen wurde wir die BEACON_NOT_FOUND primitive zurückgegeben. Um diesen proess zu beschleinigen kann das DeviceTimeReq MAC kommando verwendet werden. Da-

erklären

mit wird das GAteway aufgefordert eien bAcon zu senden. Nun kann das endgerät in den modus B wächseln.

Als Zeites setzt der MAC Layer des engerätes das Class B BIt im FCtrl feld Des Upliks auf 1. Dadurch ist er auch verandwortlich die Ping slots und für die Beacons zu öffen. Dabei muss mit der größt möglichen abweiching der Internen Uhr gerechnet werden und demensprechend die Epfangsfenster angepasst werden. DIese darf pro Beacon nicht mehr als +/- 1.3msec liegen. Der Inhalt der Empfangenenen Beacons wird mit der Signalsterke and das Programm des Endgerätes zur weiteren Verarbeitung gesendet. Dmait kann z.B. dem LoRaWan layer angewiesen werden die Uhr nachzustellen.

[SOR17c](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 geöfneten fenstern wrd der maximal mögliche clockdrift berücksichtigt. Downlink läuft wie bei A ab.

)

4.2.2 Betrieb

Damit der NEtzwerkserver dem Endgerät mittielen kann dass die pingslots frequen und/oder die Datnerate geändert werden soll gibt es den PingSlotChannelReq Mac kommando. DIe werden sind in den argumenten enthalten.

Das Endgerät kann die Periode der Pingslots zu einer beliebigen Ziet ändern. Ist dies der Fall, so muss das ENdgerät in Köasse A wechseln mit mittels dem MAc kommando PingSlotChannelReq die geämderte periode MItteilen . Danach kann zuück in Kalsse B gewechselt werden.

Falls einige länger als 2 Stunden kein Beacon empfangen wird, kann die syncronistion mit de Netzwerk verloren gehen. Dadruch funktoiniert die Kommunikation in Klasse B noicht merh und es wird in Klasse A gewechselt. Da sich nun die Kommunikationsstrategie verändet muss mit einem Uplink in dem das CLassB Fled 0 ist, der Netzwerkserver informiert werden. Nun kann versucht werden eine verbindung mit der Klasse A aufzubauen. Das Programm des Endgeätes kann versuchen wieder in Klasse B zu wecheln. DIeser prozess

wird andere gespeicher 1/s => hz

kann sich immer weider wiederholen.

Um auch innerhalb der maimal 2 Stnden in den kein Beacon empfangen wurde einen kommunikatio zu ermöglichen wird jedes mal wen ein Beacon verloren geht in den beacon-less modus gewechselt. Dieser Modus orierntiert sich ausschlieslich an der internetn Uhr. Um den Drift auszugleichen werden die EMpfangsfenter immer früher begonnen und immer später beendet. Das bedeutet einen höheren Energieverbracuh aber auch eine höhere Warschelinlichkeit noich Daten zu empfangen obwohl die Uhren des GAtways und des des ENdgerätes auseinanderlaufen. drift?

4.2.3 Singel / Multicast

Die Downlink der Klasse B unterschidene sich nicht von denen der Klasse B. allerding kann sich derFreqenzplan untersdchedien.

In Klasse B können die NAchrichten als Singelcast oder als Multicast nachrichten verwendet werden. Eine Singelcast nachricht wird an des geröt das im DevAffr fled der NAhcricht codiert ist gesendet. Im Multicastmodus wird das paket an alle ENdgeräte gesendet. Damit die möglich ist müssen sich die geräte die selbe multicas Adresse und die dazugehörigen schlüssel teilen. Durch verschiedene Multicstadressen ist es möglich soganannte multicas gruppen zu erzeugen die nciht alle sonder nur ein Teil aller entgeräte beinhalten. LoRaWan git allerding keine Methode vor wie die adressen und Schlüssel verteilt werden. Diese Aufgabe muss laso in der Applikationsebene sprich im Programm der ENtgeräte oder Direkt bei der Personlaisierung (Programerung) erledigt werden.

In Mlticastadressen sind keine MAc kommandows erlaubt. Nur Daten dürfen als Multicastbnachricht übertragen weden. DIes wurde eingeführt da Multicastnachrichten nicht die selbe robutheit wie SIngelcastnachrichten haben. Die NAchrichten dürfen nicht acknloged werden . Das Fpending zeigt an das mehr Multicasnachrichten zu senden sind. [SOR17c](sepearate Adresse für Multicast Festgelegt durch layer oder manuell für gruppenmulticast Nicht führ MAC geeignet,) unconfirmed / confirmed

4.2.4 Beacon

Wie schon erwähnt wird der Beackon verwendet um das ENdgerät mit dem Netzwerek zu synchronsisiern. Deswegen wird dieser Periodisch gesendet. Die Zeit zwischen zwei Beackons wir BEACKON_Period genannt. Die Endgeräte öffnen Enmpfangsfenster um diese Beacons zu empfangen. Ein BAcok zu übertragen dauert BEACON_RESERVED lange. Das beaon wird Beakon_GUARD früher geöfnnet um sicher zu stellen das BEac auch wirklich zu empfangen. Während versucht wird ein Beacon zu empfangen kann kein pingslot geöffnet werden. Auserdem wird die Beakon_GUARD benutzt um sicherzustellen das kein Ping slott mehr geöffnet ist. Deswegen muss diese Beakon_GUARD mindestens so lang sein wie ein maximaler pingslot. Ein weiterer vorteil ist, dass nicht daraf geachth werden muss wann ein pingslot geöffnet wird, da er soweiso im zweifelsfall fertig ist befor ein beakon empfangen wird. vorlesungsbezug?

Um snychronsisierungen druch die beacons zu vermeiden, wie alle entgeräde wollen sofort nach den beakon senden wollen, wird mittels zufälliger warte, pingslot zeiten und zufälliger pingslotanzahlen verhindert.

Beacons haben ihr eigenes Paketformat. DIese PAkete sind immer gleich lang. Dadurch kann auf header verzichted werden was auch der Geschiwndigkeit der verarbetug zu gute kommt. Wie auch ein Normales LoRaPaket, so besteht auch das erste Feld des Beakonpaketes aus der Preable nur das die des Beakonpaketes länger dauert was ein bemerken der übertragung warscheinlicher macht. Danach folgt nur noch der BCNPayload. Der BCNPayload lässt sich untertielen in RFU, Time, CRC, GWSpecific, RFU, CRC. DIe zwei CRC Felder weisen schon auf die logische unterteilung in zwei hälten hin. Der erste Teil enthält beacon speziische informationen (time und CRC). In dem Timefeld ist die zeit seit 00:00:00, Sunday 6th of January 1980 (start of the GPS epoch) modulo 2$\hat{3}$2 enthalten. das CRC feld wird verwendet um die korrektheut des Zeit und des RFU Fledes zu versichern. DIe andere hälfte ist GAtewayspezifisch. Sie enthält das GwSpecific fled und ein RFU fled das auch dirch ein zweites CRC feld abgesichert ist. Das GwSpezific feld lässt sich unterteilen in InfoDesc und Info felder. Das InfoDesc gibt an auf was sich das Infofeld be-

zieht.0 GPS coordinate of the gateway first antenna 1 GPS coordinate of the gateway second antenna 2 GPS coordinate of the gateway third antenna 3:127 RFU 128:255 Reserved for custom network specific broadcasts. Sonlage sich im infofled coordinaten enthalten kann dieses unterteilt werden in Längen unt breitengrad.

Auch Klasse A kann den beacon somit nutzen um herauszufinden von welchem gateway es gerde Datenempfängt und um somit eventuelle standortwechel festzustellen.

In Europa werden die Beacons auf einer festen frewuen übetragen die sich nicht endert auser über das MAC kommando PingSlotChannelReq. Auf anderen Kontinenten kann es sein das frequenzyhopping angewendet wird.

regionale parameter erwähnt?

4.3 Klasse C

C steht für CONTINUOUSLY Listening. Wie der Name schon sagt wird hier unaufhörlich ein empfängssender geöffnet. Dadur wird es ermöglist fast Latzenfrei zu übertragen. 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.

Geräte die Klasse C implementieren sollen aus nicht die Klasse B implementieren das es sonst zu fejlern kommen kann.

Diese Klasse verwendet die Gleichen empfangsfenster mit den gleichen Funktione wie in Klasse A. Der große unterscheid besteht allerdings darin das RX2 immer dan geöffnet ist wenn nicht gerade daten an das Gatway gesendet werden oder RX1 geöggnet ist. Also auch Während . Auserdem stehen die gleichan MAC kommandus und zwei zusätzzliche zu verfügubng .

suche normal net?

Auch in Klasse C ist es, wie in B, möglich Multicastnachrichten zu senden. hierbe gelten die gleichen Regeln wie bei B.

[SOR17c](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.)

4.3.1 Wechsel von A nach C

Da es kein ClassC Fled in einem LoRapaket gibt, wurde für das umschalten in ClassC mode MAc kommandos eingeführt. Das endgerät sendet das DeviceModeInd commando. ALs parameter kann es 0 für Klasse A und 2 Für klasse C angeben. Der Netzerkserver kann mit DeviceModeConf welches den wert der klasse enthät indas gewächselt wurde.

5 Sicherheit

Weche felder was ist wie verschlüsselt?

SIcherheit in netzwerkfägigen Systemen ist ein sher wichtiges und heiß dikutiertes thema. Da LoRa daten Üver die Lpft überträgt, ist es extrem wichtig sich und die Dtaen zu schützen. Da Luft als Medium benutzt wird könnten alle in der Nähe befindlichen geräte die gesendeten Daten mithören. Aber genauso kann ein Endgerät sich als ein andres ausgeben und in seinem Namen Daten an ein Fremden Server send. Um zu verhindern das Gesnedene Daten mitgelsen werden müssen diese verschlüsselt werde. Um zu verhindern das jemand anderst so tut als wäre er das engerät mussen die Dtane Autentifiziert werden. Slest jetz könnten z.B. Join-request mitgeschnitten werden und von dem (böße̊n) endgerät wiederholt werden um das (gut) endgerät daran zu hindern aktiv dem Netzwerk beizutreten. Deswegen wurden zähler eingebaut. Im folgenden wird sich nächer damit beschäftigt welche mechanismen es gibt die gennten probleme zu umgehen.

Oberflächlich gesehen bietet Lora eine end-to-end Sicherheit an, indem es die Signale zweimal verschlüsselt. Die erste Verschlüsselung dient dazu die gesendeten Daten vor eventuellen Mithörern zu verschlüsseln, also pm Endgerät bis zu Gateway zu verschlüsslen. 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.

Nächer betrachtd benutzt Lore eine ganze Reihe an Schlüssel und Zähler verwendet um die Kommunikation abzusichern. Da die verwendeten Schlüssel

bei OTAA-Aktivation variieren wird hier eine viel höhere Sicherit geboten als bei ABP-Activation wo alle Schlüssel von anfangan vorgegeben werden. Infolgedessen wird im folgenden Text auf die SIcherheit unter verwendung von OTAA bezogen.

Jesed enggerät hat seine eigenen NwkKey (Netzwerkschlüssel) und AppKey (applikation schlüssel). Sobald einem Netzwerk begetreten wurde wird aus dem NwkKeay der FNwkSIntKey , SNwkSIntKey und NwkSEncKey abgeleited. AUs dem AppKey iwrd zusätzlich der AppSKey abgeleited. Die schlüssel müssen so gespeichert werden, dass es nicht mölgich ist diese auf irgendeiener weise aus dem Speicher zu holen außer für das endgerät selber. Zusätzlich werden Join-Keys abgeleited. JSInitKEy und JSEncKey.

Der FNwkSIntKey ist einzigartig für ein Endgerät und heist Forwarding Network session integrity key. Der Schlüssel wird verwendet um ganze oder Teile der MIC felder in den LoRapaketen zu berechen . mic

Serving Network session integrity key heißt abgekützt SNwkSIntKey. Dieser Schlüssel wird verwedent um die Integrität des MIC codes zu überprüfen. Zusätzlich wird er auch verwendet um Teile des MIC codes zu berechen. Dieser Schlüssel ist spezifisch für ein entgerät.

NwkSEncKey oder lang Network session encryption key, ist für jede Netzwerksitzung einzigartig und wurd verwendet um empfnagen order gesendete Mac kommandos zu ent- oder verschlüsseln.

Der AppSKey wird auch Application session key und wird einem Endgerät zugeordent. Er wird vom Gateway und vom Endgerät verwendet um Daten die zum Applikationsserver geschickt werden sollen zu verschlüsseln.

pad fügt do viele 0en das die länge an vielfaches von 16 ist AppSKey = aes128_encrypt(NwkKey, 0x02 | JoinNonce | NetID | DevNonce | pad16) FNwkSIntKey = aes128_encrypt(NwkKey, 0x01 | JoinNonce | NetID | DevNonce | pad16) SNwkSIntKey = NwkSEncKey = FNwkSIntKey.

Jedes Gerät hat 3 verscheidene Frame counter um die Anzahl der gesendeten und empfangenen Frams mitzuzählen der FCntUP counter zählt die uplikframes, der NFCNTDown zählt die MAC-downlinkframes und der AF-

CntDown welcher alle downlinkframes zählt die Nutzdaen enthalten.

Wenn ein gerät dem Netzwerk beitritt, werden zuerst die Counter auf 0 gesetzt. Beide seiten einer Kommunikation halten die zähler gleich. Beim senden wird der Aktulle counterwert in das FCnt feld eingetragen. Werden übertragungen wiederholt so wird der counter nicht erhöht weiderholung

Durch das verwerfen von NAchrichten mit zu kleinem Counterwert, wird schon verhindert das Pakete von einem Angreifer aufgenommen und zu einem späteren Zeitpunkt wiederabgespielt werden. drin?

Jauch bei den Join oder Accept NAchrichten besteht die gefahr eine Replayattac. Da hier dem Netzwerk noch nicht begetreten wurde, können die zähler niht verwendet werden. Hier wird eine Nonce in die Join-Pakete coderit . DIese Nonce zählt auf die gleiche weise hoch wie die counter. Die gegnerische Seiter der Kommunkikation muss die Nonce tracken und darf nur pakete mint einer Nonce akzeptieren die höher ist als die letzt Nonce. wie sieht paket aus.

[GAS17](Netzwerkserver hat AppKey daraus werden AppSKeay und NwkSkey erzeugt) [Tec15](Applikationsverschlüsselung(schutz der Daten for mitlesen) Netzwerk(Autentiizierung der Knoten) AFS, Key Exnage IEEE EU164) [SOR17c](symetrischer Schlüssel => nur einer benötigt, Sessionkey ist abgeleited von Knoten-rootkey. JoinServer setllt verbindung der Keys her.) frequncyhopping

6 Live-Beispiel

wenn vorhanden.

7 Ausblick

Die verwendete frequenz entspricht der RX1 bzw RX2 aus dem kapitel Klasse A. wo kammt das her gehe ich zu weit?

Sobald dem Netzwerk erfolreich beigetreten wurde werden ein die benötigten schlüssel aud dem vorher gesetzetn Werten abgeleited. genauers dazu in kapitel Sicherheit.

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 Famaey: *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: *LoRaWAN™SECURITYA WHITE PAPER PREPARED FOR THE LoRa ALLIANCE™*. https://lora-alliance.org/resource-hub/lora-alliance-security-whitepaper, Februar 2017. Eingesehen am 09.04.2019.

[SOR17a] SORNIN, N. (Herausgeber): *Exploring LoRa and LoRaWAN(todo)*. Lora-Alliance, https://tools.ietf.org/pdf/rfc8376.pdf, 1.1 Auflage, Oktober 2017. Eingesehen am 09.04.2019.

[SOR17b] SORNIN, N. (Herausgeber): *LoRaWAN™ 1.1 Backend(todo2*. Lora-Alliance, https://tools.ietf.org/pdf/rfc8376.pdf, 1.1 Auflage, Oktober 2017. Eingesehen am 09.04.2019.

[SOR17c] SORNIN, N. (Herausgeber): *LoRaWAN™ 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 LoRaWAN™*. https://lora-alliance.org/

resource-hub/what-lorawantm, November 2015. Eingesehen am 09.04.2019.