



MAC-Schicht Optimierung zur Nutzung eines auf UWB basierenden drahtlosen Kommunikationsnetzwerks innerhalb von Satelliten

von

André Lübken

Thesis zur Erlangung des akademischen Grades
Master of Science

1. Gutachter: Prof. Dr.-Ing. Carsten Bormann
2. Gutachter: Dr.-Ing. Olaf Bergmann

im

Fachbereich 3 – Mathematik / Informatik
Deutsches Zentrum für Luft und Raumfahrt (DLR)
Institut für Raumfahrtssysteme

Juni 2018

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorstehende Arbeit mit dem Titel „MAC-Schicht Optimierung zur Nutzung eines auf UWB basierenden drahtlosen Kommunikationsnetzwerks innerhalb von Satelliten“ selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der in den beigefügten Verzeichnissen angegebenen Hilfsmittel nicht bedient habe. Alle Textstellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Die eingereichte elektronische Fassung der Arbeit entspricht der eingereichten schriftlichen Fassung exakt. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Unterschrift:

Datum:

UNIVERSITÄT BREMEN

Abstract

Fachbereich 3 – Mathematik / Informatik
Deutsches Zentrum für Luft und Raumfahrt (DLR)
Institut für Raumfahrtsysteme

Master of Science

von André Lübken

Raumfahrzeuge bestehen nach gängiger Praxis aus unterschiedlichen Subsystemen, die miteinander verbunden sind und über Bussysteme, wie SpaceWire, Daten miteinander austauschen. Ein großer Kostenfaktor bei der Konstruktion ist dabei die Verkabelung, um die Kommunikation zwischen diesen Systemen sicherzustellen.

Zur Reduzierung des Verkabelungsaufwandes bietet sich die Verwendung energiesparender, drahtloser Sensornetzwerke an, wie sie auch auf der Erde bereits in vielen Anwendungsbereichen zum Einsatz kommen. Eine neue vielversprechende Übertragungstechnik in diesem Bereich stellen impulsbasierte Ultrabreitband Funkverbindungen dar. Prinzipbedingt sind diese sehr robust und bspw. nahezu immun gegen Signalabschwächung durch Mehrwegeausbreitung. Über dieser physischen Zugriffsebene definieren Standards wie der IEEE 802.15.4 auch MAC Layer, die für diese Betriebsart besonders geeignet sind und die Verteilung des Mediums zwischen den Teilnehmern organisieren. Insbesondere die jüngsten Erweiterungen dieses Standards zielen auf die Erhöhung der Robustheit dieser Netze ab, um sie z.B. in Industriesteuerungen verwenden zu können.

Diese Arbeit zeigt die Implementierung der LLDN Erweiterung des 802.15.4e Standards für Ultrabreitband Transceiver. Dieser auf einem TDMA Verfahren basierende MAC Layer zeichnet sich durch besonders geringen Protokoll-Overhead aus. In Kombination mit einer Ultrabreitbandverbindung lässt sich so ein sehr leistungsfähiges und gleichzeitig robustes Netz realisieren, das auch für kritische Prozesse an Bord eines Raumfahrzeuges eingesetzt werden kann. Die implementierte Lösung wurde mit dem weit verbreiteten TSCH MAC des Standards verglichen und zeigt neben einer signifikant geringeren Latenz auch ein deterministisches Latenzverhalten bei gleichzeitig niedrigem Energieverbrauch pro Paket. Die Kombination von LLDN und Ultrabreitband zeigt das Potential, um auf dieser Basis einen angepassten MAC Layer für die Intra-Satellitenkommunikation zu entwickeln.

Danksagung

Ich möchte an dieser Stelle zuerst meinem Betreuer am DLR Herrn Martin Drobczyk besonderen Dank aussprechen. Seine langjährige Erfahrung im Gebiet machte mir den Einstieg in ein solch komplexes Themenfeld sehr einfach. Er stand mir stets mit guten Ratschlägen und Ideen zur Seite und hat sich weit über das Maß einer üblichen Betreuung hinaus für diese Arbeit eingesetzt.

Weiterhin möchte ich auch meinem Erstgutachter Herrn Prof. Dr.-Ing. Carsten Bormann für seine Unterstützung und die zahlreichen Anregungen während der Bearbeitungszeit danken.

Außerdem danke ich meiner Freundin Frau Miriam Losch für die emotionale Unterstützung und ihr Verständnis während des letzten halben Jahres. Ihre Korrekturvorschläge tragen ungemein zur Lesbarkeit dieser Arbeit bei.

Nicht zuletzt möchte ich mich besonders bei meiner Familie bedanken, insbesondere bei meinen Eltern Holger und Manuela Lübken, die mich während meiner gesamten Studienzzeit unterstützt haben.

Inhaltsverzeichnis

Selbstständigkeitserklärung	i
Abstract	ii
Danksagung	iii
Abbildungsverzeichnis	vi
Tabellenverzeichnis	vii
1 Einführung	1
1.1 Motivation	1
1.2 Ziele und Methodik	3
1.3 Verwandte Arbeiten	4
1.4 Grundlagen einer Compsat Mission	6
1.4.1 Management von Raumfahrtprojekten	7
1.4.2 Gliederung einer Raumfahrtmission	8
1.4.3 Subsysteme eines Satelliten	9
1.4.4 WICome	10
1.5 Gliederung der Arbeit	13
2 Anforderungsanalyse	14
2.1 Anforderungen an den MAC Layer	15
2.1.1 Allgemein	15
2.1.2 Leistung	16
2.1.3 Robustheit und Zuverlässigkeit	17
2.1.4 Energie	18
2.1.5 Einbettung	19
2.2 Zusammenfassung	19
3 Technische Grundlagen	21
3.1 IEEE 802.15.4 Standard	23
3.1.1 Netzwerktopologie	24
3.1.2 Physical Layer	25
3.1.3 MAC Layer	27
3.2 Weiterentwicklungen auf Basis von 802.15.4	32
3.2.1 Nachteile des 802.15.4 MAC Layers	32
3.2.2 WirelessHART	34
3.2.3 ISA100.11a	36
3.2.4 802.15.4e	37
3.3 Ultra Wideband	46

3.4	Kapitelzusammenfassung	49
4	Umsetzung	50
4.1	Entwurf	50
4.1.1	Auswahl des MAC Layers	50
4.1.2	Auswahl eines RTOS	54
4.1.3	Architektur	56
4.2	Verwendete Hardware	59
4.3	Implementierung	62
4.3.1	Board Support Package	63
4.3.2	Simulationsumgebung	64
4.3.3	DW1000 Treiber	66
4.3.4	LLDN MAC Layer	68
4.4	Kapitelzusammenfassung	79
5	Evaluation	81
5.1	Packet Delivery Ratio	82
5.2	Latenzverhalten	83
5.3	Durchsatz	87
5.4	Energieeffizienz	88
5.5	Kapitelzusammenfassung	91
6	Fazit	93
6.1	Zusammenfassung	93
6.2	Ausblick	94
6.2.1	System	95
6.2.2	Standard	96
A	Verifikation	98
A.1	Unit Tests	98
A.2	Testfallherleitung	100
A.3	Whitebox: Statische Codeanalyse	102
	Literatur	104
	Abkürzungsverzeichnis	108

Abbildungsverzeichnis

1.1	Explosionsdarstellung Compsat	11
3.1	OSI-Modell	22
3.2	Netzwerktopologien in 802.15.4	25
3.3	Struktur einer 802.15.4 PPDU	27
3.4	Superframe Struktur des 802.14.4 MAC Layers	28
3.5	Allgemeines Format eines 802.15.4 MAC Frames	31
3.6	Einfluss der Paketrate auf die Zuverlässigkeit des klassischen 802.15.4 MAC Layers	34
3.7	TSCH Slotframe Struktur	40
3.8	Channel Hopping im DSME PAN	43
3.9	Aufbau eines Superframes in einem LLDN	45
3.10	Aufteilung eines LLDN Timeslots	46
3.11	Schmalbandige FSK Übertragung mit Trägerwelle	47
3.12	Frequenzspektrum von Ultrabreitbandübertragungen im Vergleich	48
4.1	Architektur des Netzwerkstacks	57
4.2	Verwendete Hardware – Gegenüberstellung der Hardware, die zur Ent- wicklung genutzt wurde und der letztendlichen Zielhardware	61
4.3	Architektur des der Simulationsumgebung	64
4.4	Zustandsautomat LLDN Manager	70
4.5	Chronogramm LLDN	72
4.6	Zeitsynchronisierung bei Empfang eines Paketes	75
5.1	Gegenüberstellung der Frame Zeiten mit/ohne ACK	81
5.2	Vergleich der Latenz pro Paket bei TSCH/LLDN	85
5.3	Latenz in einem LLDN bei Optimierung aller Parameter und Einreihung der Pakete in Synchronisation mit den Uplink Slots	86
5.4	Energieverbrauch der Transceiver eines Knotens und des Coordinators in einem Slot	89
5.5	Energieverbrauch des Transceivers eines Knotens während eines Superfra- mes	90
A.1	Zustandsautomat der Slotverwaltung für die Verifikation	101

Tabellenverzeichnis

2.1	Leistungsanforderungen der einzelnen Subsysteme	16
2.2	Zusammenfassung der ermittelten Anforderungen an den MAC Layer . . .	20
4.1	Entscheidungsmatrix zur Auswahl des zu implementierenden Standards .	53
4.2	Umzusetzende Features der LLDN Spezifikation	53
4.3	Entscheidungsmatrix zur Auswahl des embedded Betriebssystems	56
5.1	Packet Delivery Ratio im Vergleich	83
5.2	Vergleich des Durchsatzes verschiedener Netzkonfigurationen	88

Einführung

Drahtlose Kommunikation spielt in immer mehr Anwendungsbereichen eine wichtige Rolle. Insbesondere Technologien mit niedrigeren Datenraten und hoher Energie- und Kosteneffizienz treiben die Entwicklungen im Bereich der drahtlosen Sensor- und Aktuatornetzwerke zur Vision des allumfassenden „Internet of Things“.

Nicht zuletzt der Trend zur Verwendung von kostengünstiger, kommerzieller Serienhardware (COTS) führt dabei auch zum Einsatz in kritischeren Szenarien wie z.B. der Anwendung in Industriesteuerungen. Dabei sind Verfahren, denen der IEEE Standard 802.15.4 zugrunde liegt oft die Basis für diese Systeme.

Auch in der Raumfahrtbranche gibt es erste Bestrebungen drahtlose Sensornetze (WSN) gewinnbringend auf Raumfahrzeugen z.B. zur Intra- oder Inter-Satellitenkommunikation einzusetzen [Noh09]. Die physische Übertragung von Signalen mittels Ultrabreitband (UWB) bietet hier die Möglichkeit störungsarme und robuste Netze aufzubauen. Die geringe spektrale Leistungsdichte von UWB sorgt zudem dafür, dass es sehr verträglich mit bestehenden schmalbandigen Anwendungen koexistieren kann.

Das deutsche Zentrum für Luft- und Raumfahrt entwickelt regelmäßig mit den Compsat Minisatelliten eine Plattform für Technologieexperimente, auf der ein solches Sensornetzwerk auf Basis von UWB betrieben und evaluiert werden soll.

1.1 Motivation

Die Hauptvorteile der Nutzung von drahtlosen Sensornetzwerken auf Raumfahrzeugen sind eine signifikante Reduzierung der Masse durch nicht mehr nötige Kabelverbindungen um bis zu 10 % [LFX09], sowie die damit verbunden geringeren Montagekosten.

Da die meisten Raumfahrzeuge typischerweise Einzelstücke sind, hat auch ihre Hard- und Softwareausstattung Prototypcharakter. Seit einiger Zeit gibt es jedoch Bestrebungen zu Modularisierung und Wiederverwendung anstatt der bisherigen maßgeschneiderten Lösungen für jede Mission. Hard- und Software muss für diese aufwändig qualifiziert werden. Daher ermöglicht das Aufsetzen auf allgemeinen Standards eine Verringerung der Qualifizierungskosten und bietet Zukunftssicherheit. Dies ist einer der Gründe, weshalb das DLR neue, modulare Technologien auf den Compsat Satelliten testet. Die Einbettung

der verwendeten WSN Transceiver in das OSI Schichtenmodell erlaubt hierbei die Wiederverwendung vieler bereits implementierter Schichten auf den verwendeten Embedded Plattformen und damit von Protokollen für verschiedenste Anwendungsfälle.

Technologien auf der Basis von 802.15.4 sind kostengünstig und energieeffizient, womit sie auch für Raumfahrtanwendungen interessant sind. Die Anforderungen an die Robustheit und Zuverlässigkeit, sowohl an Hard- wie Software, in Weltraumanwendungen sind äußerst streng. Mögliche Einsatzszenarien reichen von der Übertragung wissenschaftlicher Daten z.B. von Sensoren bis hin zur Übertragung hochkritischer Informationen von und zur Lageregelung des Raumfahrzeugs. Der klassische MAC Layer von 802.15.4 reicht zur Erfüllung dieser Anforderungen nicht aus. In mehreren Untersuchungen ist hier eine zu große Störanfälligkeit festgestellt worden [ACF11]. Festgestellte Probleme sind u.A. keine Möglichkeit eine garantierte Bandbreite oder Latenz einzuhalten, sowie eine zu geringe Zuverlässigkeit. Ein weiteres Problem ist Multipath Fading, das bei den eingesetzten schmalbandigen Physical Layern eine Verschlechterung des Signals zur Folge hat.

Die Impulse Radio Ultrawideband (IR-UWB) Technologie kann hier Abhilfe schaffen. Hier werden die Informationen durch extrem kurze Pulse auf ein sehr breites Spektrum verteilt, das weitgehend immun gegen die Effekte durch Mehrwegausbreitung ist. UWB kann durch seine geringe spektrale Leistungsdichte problemlos parallel zu üblichen schmalbandigen Funkdiensten betrieben werden.

Für die Umsetzung eines UWB basierten WSN auf dem nächsten Comsat Satelliten soll ein MAC Layer implementiert werden, der auf den Erweiterungen des 802.15.4 Standards basiert. Diese Erweiterungen wurden ursprünglich für Industrieanwendungen entwickelt, um in erster Linie mit anderen Betriebsmodi die Robustheit zu erhöhen. 802.15.4 Netze werden in den lizenzfrei nutzbaren ISM¹ Bändern betrieben, die aus oben genanntem Grund stark genutzt werden. Insbesondere im 13 cm Band bei 2,4 GHz ist durch die Verbreitung von WLAN mit ausgeprägten Störungen für WSNs zu rechnen. Viele der relevanten MAC Erweiterungen für den UWB Physical Layer sind bis jetzt nur theoretisch und durch Simulationen betrachtet worden. Vielversprechend ist bspw. eine Machbarkeitsstudie zur *Deterministic and Synchronous Multi-Channel Extension (DSME)* in Kombination mit UWB [PHI13].

Eine Implementierung und Integration dieser Erweiterungen für UWB in gängige Protokollstacks auf Embedded Betriebssystemen steht jedoch noch aus.

¹Industrial, Scientific, Medical

1.2 Ziele und Methodik

Ziel der Arbeit ist die Untersuchung von MAC Layer Erweiterungen auf Basis von 802.15.4 und ihrer Anwendbarkeit in Bezug auf IR-UWB. Geeignete Verfahren sollen auf Basis der STM32 Mikrocontroller Architektur in einem bestehenden Embedded Betriebssystem implementiert und auf ihre Leistungsfähigkeit unter verschiedenen Gesichtspunkten hin untersucht werden.

Dazu ist zuerst eine Anforderungsanalyse unter Berücksichtigung der Weltraumumgebung vonnöten. Institutionen wie die European Commission for Space Standardization (ECSS) oder das Consultative Committee for Space Data Systems (CCSDS) geben Standards und Best-Practices heraus, die hierzu unter anderem herangezogen werden können. Beispielhaft sei hier das Magenta Book der CCSDS zu drahtlosen Sensoren genannt [CCS13], das Empfehlungen für diese Art von Systemen enthält. Insbesondere wird eine Analyse der benötigten minimalen Latenz und des Datendurchsatzes durchgeführt. Auf Basis dieser wird dann eine passende Erweiterung für die Implementierung ausgewählt.

Die Implementierung des MAC Layers soll letztendlich auf STM32L151 Mikrocontrollern erfolgen. Geeignete Betriebssysteme für die Integration sind hier bspw. Contiki oder FreeRTOS. Gegebenenfalls ist die Implementierung eines Adapters oder die Portierung eines Treibers für den zu verwendenden Decawave DWM1000 UWB Chip vonnöten, um diesen an den MAC Layer des Betriebssystems anbinden zu können.

Es folgt eine Evaluation der entwickelten Lösung. Dabei ist ein Szenario zu entwickeln, das einen Vergleich der Leistungsfähigkeit des verwendeten Verfahrens mit bestehenden ermöglicht und demonstriert, ob die Anforderungen erfüllbar sind. In Anbetracht der Kritikalität bietet sich hier zusätzlich eine methodische Testfallherleitung an, die die Implementierung funktional verifiziert. Zusätzliche Whitebox Tests zur Erkennung und Beseitigung von Fehlern sind ebenfalls sinnvoll, um sicherzustellen dass diese kritische Kommunikationsschicht, auf der alle anderen aufbauen, korrekt arbeitet.

Aus dem Ziel ergeben sich die folgenden Forschungsfragen, die im Laufe der Arbeit wieder aufgegriffen und beantwortet werden sollen:

Was sind die Anforderungen an ein drahtloses Sensornetzwerk, das in Weltraumumgebungen betrieben werden muss? Diese lassen sich durch Nutzung von Standards und Best-Practices verschiedener Standardisierungsgremien herleiten.

Welche Probleme treten bei IR-UWB auf? Wie können diese ggf. durch MAC Schicht Erweiterungen gelöst werden? Auch die physische Übertragung per UWB ist nicht

optimal. Eine Untersuchung inwieweit diese durch den zu implementierenden MAC Layer gelöst werden können, ist nötig.

Welche vorhandenen MAC Erweiterungen des 802.15.4 sind mit IR-UWB sinnvoll nutzbar? Die Eigenheiten des UWB PHY Layers lassen sich u.U. nicht mit bestimmten MAC Betriebsmodi nutzen oder schaden dem Betrieb sogar.

Wie wirkt sich die Verwendung eines alternativen MAC Betriebsmodus auf den Energieverbrauch des Systems aus? Energieverbrauch ist bei den verwendeten eingebetteten Systemen ein wichtiger Faktor. Der MAC Layer hat direkten Einfluss auf den Betrieb und damit den Stromverbrauch der Funkhardware.

Wie lässt sich die Lösung in bestehende Protokollstacks integrieren? Verschiedene eingebettete Betriebssysteme verwendeten unterschiedliche architekturelle Ansätze, um die OSI Schichten zu realisieren. Eine sinnvolle Integration in eine dieser Architekturen muss durchgeführt werden.

Welche Verbesserung der Robustheit der Übertragung lässt sich erreichen? Der zu implementierende MAC Layer soll idealerweise eine Verbesserung gegenüber dem klassischen 802.15.4 MAC erreichen. Ob dies ebenfalls unter Verwendung von UWB der Fall ist, gilt es herauszufinden.

Welchen Einfluss hat die MAC Implementierung auf die Latenz? Hier gilt es ggf. Kompromisse unter Berücksichtigung der anderen Ziele zu machen.

Welchen Einfluss hat die MAC Implementierung auf den Datendurchsatz? Ob die nutzbare Bandbreite durch erweiterte MAC Layer u.U. negativ beeinflusst wird und inwieweit dies mit Bezug auf die Anforderungen akzeptabel ist, muss untersucht werden.

Wie lässt sich das in den MAC Layern verwendete CSMA/CA Verfahren mit UWB nutzen? Ultrabreitbandübertragungen zeichnen sich dadurch aus, dass sie keinen Träger nutzen, den das CSMA/CA Verfahren zur Erkennung benötigt. Geeignete Alternativen müssen evaluiert werden.

1.3 Verwandte Arbeiten

Einer der ersten Versuche zum Betrieb von drahtlosen Sensornetzwerken auf Raumfahrzeugen wurde bereits 2014 auf dem VELOX-1 Satellitenpaar mit einem ZigBee Netzwerk zur Inter-Satellitenkommunikation durchgeführt [Xie14]. Die Mission bestand aus einem

Nanosatelliten im Cubesatformat², der einen kleinen Picosatelliten bei sich führte. Im operativen Betrieb wurde dieser daraufhin mit einem Federmechanismus freigesetzt, so dass sich beide immer weiter voneinander entfernten. Zwischen beiden Satelliten wurde dann ein ZigBee Netzwerk aufgebaut, das zum Austausch der Housekeepingdaten des Picosatelliten genutzt wurde. Die Autoren konstatieren jedoch auch, dass Erweiterungen wie z.B. WirelessHART hier besser geeignet gewesen wären.

Ein Beispiel zum Einsatz von WSN in Weltraumumgebungen ist das Projekt WirelessCompose des DLR, das UWB Transceiver in einem Experiment auf der ISS einsetzt [DSP17]. Ziel des Experimentes ist die Evaluierung von UWB und die erreichbare Signalqualität in einem solchen Netz mit mehreren Sensorknoten. Da die UWB Technologie sich zudem sehr gut zur Positionsbestimmung eines Knoten in einem Netz einsetzen lässt werden die Sensoren in Experimenten zur Positionsbestimmung von Astronauten im Raum eingesetzt. Zusätzlich wird ein Energyharvesting System eingesetzt, das den batterielosen Betrieb durch Nutzung der Lichtquellen auf der ISS sicherstellen soll. Es wird zur Kommunikation jedoch kein standardisierter MAC Layer eingesetzt. Stattdessen wird die Kommunikation direkt über die physical Layer Implementierung des UWB Transceivermoduls abgewickelt. Um Kollisionen zu vermeiden wird hier die ALOHA Zugriffsmethode eingesetzt, bei der zufällig ein Zeitslot nach dem 802.15.4 Standard gewählt wird und Kollisionen durch das Ausbleiben von Bestätigungen erkannt werden. Fehlende Daten werden dann erneut übertragen.

Eine Implementierung mit Einbettung in einen Protokollstack für UWB wurde für das Contiki embedded Betriebssystem gezeigt [Cha16]. Contiki ist ein Echtzeitbetriebssystem (RTOS), das speziell für Mikrocontrollerarchitekturen entwickelt wurde, die über geringe Leistung und z.B. keine eigene Memory Management Unit (MMU) verfügen. Das Betriebssystem ist auf den Betrieb von WSN optimiert und implementiert einen entsprechenden Stack von Netzwerkprotokollen. Neben IPv4 und IPv6 implementiert es auch den für IoT Geräte wichtigen IETF Standard 6LoWPAN in einem Adaption Layer, sowie z.B. CoAP auf der Anwendungsschicht. Für die Anbindung von Transceivermodulen bietet es eine standardisierte Schnittstelle auf dessen Basis Treiber geschrieben werden können. Besonderes Herausstellungsmerkmal ist der Radio-Duty-Cycle (RDC) Layer unterhalb des MAC Layers. Dieser sorgt dafür, dass Knoten miteinander kommunizieren können, auch wenn sie den Großteil ihrer Betriebszeit schlafen. Ein Beispiel für ein solches RDC Protokoll ist ContikiMAC [Dun11]. Es nutzt keine zusätzlichen Header für die versendeten Pakete und basiert auf dem periodischen Aufwachen des Empfängers. Empfängt dieser eine Übertragung, die für ihn bestimmt ist, wird durch ein ACK der Gegenseite die weitere Empfangsbereitschaft signalisiert und der Empfänger bleibt aktiviert. Da das Protokoll jedoch auf der Messung der Empfangssignalstärke basiert und dies bei UWB durch die

²Cubesats – Kleinstsatelliten, die aus Würfeln mit 10 cm Kantenlänge zusammengesetzt sind

gepulste Übertragung schwierig ist, bedarf es weiterer Anpassungen, damit das Protokoll auch für UWB nutzbar wäre. Aktuell lässt sich UWB hier nur mit dem ursprünglichen MAC Layer des 802.15.4 ohne Anpassungen betreiben.

Eine erste Implementierung einer bestimmten Erweiterung des 802.15.4 Standards zeigt OpenWSN [Tho12b]. Ziel von OpenWSN ist die Implementierung von *Time Synchronized Channel Hopping* (TSCH) (siehe hierzu auch 3.2.4 auf Seite 37) wie es in der Erweiterung 802.15.4e des Standards spezifiziert ist. Sie ist unabhängig vom Betriebssystem und wirbt mit leichter Portierbarkeit. Aktuell ist der Stack auf dem FreeRTOS oder OpenOS Kernel für Mikrocontroller lauffähig. Die Implementierung zeigt die Vorteile des Verfahrens bei gleichzeitig sehr niedrigem Energieverbrauch der Knoten. Durch die Synchronisierung der Zeitschlitzte müssen Geräte nur dann aufwachen, wenn sie auch wirklich kommunizieren können, wodurch auch weniger Retransmits stattfinden. Beides zusammen sorgt für den niedrigen gemessenen Energieverbrauch. Einschränkung dieser Implementierung ist, dass sie lediglich den 802.15.4e Standard und hieraus nur die TSCH Erweiterung betrachtet. Sie ist somit nicht kompatibel mit den UWB PHY Erweiterungen. Wie in 4.1.1 auf Seite 50 ausgeführt, eignet sich TSCH außerdem nicht in gleichem Maße für UWB wie für die Varianten mit Träger z.B. im 2,4 GHz Band.

Eine Machbarkeitsstudie zur *Deterministic and Synchronous Multi-Channel Extension*³ (DSME), die im 802.15.4e spezifiziert ist, in Kombination mit IR-UWB wird in [PHI13] untersucht. In diesem Modus wird eine höhere Robustheit der Datenübertragung durch Mesh Networking und durch die Möglichkeit garantierte Zeitslots effizienter zu nutzen erreicht. Im Gegensatz zu TSCH muss bei Änderung der Knotenanzahl oder des Datenaufkommens keine aufwändige Neuberechnung der Slots vorgenommen werden. Der Artikel stellt bei Betrieb von DSME einen signifikant höheren Datendurchsatz sowie eine höhere Zuverlässigkeit bei der Paketempfangsrate fest. Ein Vergleich zwischen dem UWB PHY Layer mit DSME und einem konventionellen PHY Layer fehlt jedoch.

1.4 Grundlagen einer Compsat Mission

Das drahtlose Sensornetz, für das die in dieser Arbeit vorgestellten Erweiterungen entwickelt werden, soll auf dem nächsten Compsat Satelliten im Experiment WICome⁴ betrieben werden. Die besonderen Rahmenbedingungen einer Raumfahrtmission haben signifikante Auswirkungen auf die Anforderungen und die Art der Entwicklung aller

³Details zu DSME sind in 3.2.4 auf Seite 37 beschrieben

⁴Wireless Intra-Satellite Communication Experiment

beteiligten Systeme. Daher werden in diesem Abschnitt kurz die Grundlagen zur Gliederung einer Mission und der typischen Einteilung der Systeme im Allgemeinen erläutert, sowie im Besonderen die Comsat Mission.

1.4.1 Management von Raumfahrtprojekten

Die Organisation von Raumfahrtprojekten folgt, wie in anderen Branchen auch, in ihrer Vorgehensweise bestimmten Standards. Insbesondere für europäische Projekte gelten hier die Standards der European Commission for Space Standardization (ECSS), die als gemeinsames Konsortium der Industrie und der Raumfahrtagenturen der Länder ein einheitliches Rahmenwerk für die Zusammenarbeit schafft. Neben technischen Vorgehensweisen regeln die Standards der ECSS auch das Projektmanagement. So sind Projekte nach ECSS in verschiedene Phasen unterteilt, die einem dem V-Modell⁵ ähnlichen Review-Zyklus folgen.

Der Standard ECSS-M-ST-10C [Eur09, S. 19ff] definiert die folgenden Phasen:

- Phase 0 – Missionsanalyse: Hier wird das Ziel der Mission mit operationellen Einschränkungen definiert und eine erste vorläufige Anforderungsspezifikation durch den Hauptkunden erstellt.
- Phase A – Machbarkeitsstudie: Die Studie wird durch Hauptkunden und erste Zulieferer erstellt und beinhaltet erste Pläne zur Organisation und Entwicklung mit einer Analyse der technischen Machbarkeit unter anderem durch die Identifikation kritischer Technologien.
- Phase B – Grobdesign: In der vorläufigen Definitionsphase werden erste Zeitpläne, Kosten und technische Lösungen festgelegt. Eine vorläufiges Systemkonzept wird durch eine Trade-off Analyse ermittelt.
- Phase C – Feindesign: Die Detailspezifikation enthält die endgültige Definition des Designs. Alle Schnittstellen sind hier definiert, die Test- und Integrationspläne sind aufgestellt und Tests erster kritischer Komponenten finden statt.
- Phase D – Produktion und Qualifikation: Hier wird nach den zuvor ermittelten Vorgaben das Flugmodell gebaut und montiert. Die Hard- und Software des Raumfahrzeugs und des Bodensegmentes werden entwickelt und getestet. Systemtests werden durchgeführt.

⁵Im V-Modell wird den Phasen der Entwicklung jeweils eine entsprechende Qualitätssicherungsphase gegenübergestellt.

- Phase E – Einsatz: Diese Phase beinhaltet den Start, sowie den Betrieb des Raumfahrzeugs zusammen mit allen unterstützenden Aktivitäten am Boden.
- Phase F – Entsorgung: Nach dem zuvor entwickelten Entsorgungsplan wird die Mission beendet. Dies kann z.B. ein kontrollierter Wiedereintritt des Raumfahrzeugs mit anschließendem verglühen in der Erdatmosphäre sein.

Alle Phasen werden durch entsprechende Reviews abgeschlossen. Das Projekt selbst wird nach diesem Vorgehensmodell in einen Product Tree, eine Work Breakdown Structure (WBS) und einzelne Arbeitspakete / Work Packages (WP) unterteilt. Der Product Tree und die WBS unterteilen das Projekt in die einzelnen Segmente (z.B. Raum- und Bodensegment) und die entsprechenden Systeme (wie im nächsten Abschnitt erläutert), die z.B. von unterschiedlichen Zulieferern stammen können. Die einzelnen Arbeitspakete stellen dann die definierten Aufgabenbereiche in der WBS dar. Sie können beliebig klein sein, solange die Arbeit in ihnen mess- und steuerbar bleibt.

Die hier vorgestellte Compsat Mission befindet sich momentan in der Phase O/A, womit die ersten Machbarkeitsstudien abgeschlossen sind und mit der Grobspezifikation begonnen werden kann.

1.4.2 Gliederung einer Raumfahrtmission

Eine Raumfahrtmission wird in verschiedene Segmente eingeteilt, die geschlossen einer Aufgabe dienen [KWW11, vgl. S. 42 ff.]. Innerhalb der Segmente gibt es verschiedene Systeme, die der Erfüllung einer definierten Aufgabe dienen. Das Bodensegment enthält beispielsweise die Bodenstation als eines ihrer Systeme. Diese werden wiederum in weitere Subsysteme aufgeteilt. Im Beispiel der Bodenstation wäre eines das Antennensubsystem, das die Kommunikation mit dem Raumfahrzeug sicherstellt.

Üblich sind die vier Segmente *Programm-, Boden-, Start- und Raumsegment*. Im Programmsegment werden die verwaltungsspezifischen Aspekte einer Mission zusammengefasst. Dazu gehören z.B. die Planung oder auch das Qualitätsmanagement. Das Bodensegment fasst alle nötigen Systeme zum operativen Betrieb der Mission zusammen. Hier finden sich die bereits erwähnte Bodenstation, die Up- und Downlink zur Verfügung stellt, aber auch das Kontrollzentrum, in dem die Telemetrie des Raumfahrzeugs ausgewertet wird und Kommandos vorbereitet werden, die dann schließlich über die Bodenstation übertragen werden können. Ein weiterer wichtiger Teil ist hier die Datenprozessierung und Archivierung. Nutzdaten der Payload können im Regelfall nicht direkt verwendet werden und erfordern weitere Verarbeitung, um auswertbar zu sein. In der Archivierung werden schließlich nicht nur die Payloaddaten gespeichert, sondern auch die Telemetrie

und Telekommandos, die mit dem Raumfahrzeug ausgetauscht wurden. So kann z.B. im Fehlerfall aus dem Archiv exakt rekonstruiert werden, in welchem Zustand sich das Raumfahrzeug zum Zeitpunkt des Fehlers befand. Das Start- oder auch Transfersegment befasst sich mit dem Einschuss des Raumfahrzeugs in den gewünschten Orbit oder in die Transfertrajektorie. Hier werden unter anderem die organisatorischen Aspekte von Raketenstarts behandelt. Dazu gehört neben der Wahl des Startfahrzeugs und des Startplatzes z.B. auch die Planung von Kampagnen, sowie die aufwändige Vorbereitung der Starts. Im Raumsegment befindet sich schließlich das Raumfahrzeug, dessen Systeme in den Satellitenbus und die Nutzlast unterteilt werden. Da die Vernetzung der einzelnen Subsysteme auf einem Satelliten durch drahtlose Sensornetzwerke eines der denkbaren Szenarien für diese Technologie ist, folgt eine genauere Erläuterung der einzelnen Subsysteme.

1.4.3 Subsysteme eines Satelliten

Die Systeme eines Satelliten werden in den Satellitenbus und die Nutzlast unterteilt. Die Nutzlast beinhaltet die Subsysteme, die das eigentliche Missionsziel realisieren, sei es nun das Sammeln von Messwerten verschiedenster Sensoren zu Forschungszwecken oder die Bereitstellung von Kommunikationsmöglichkeiten mit dem Boden. Der Satellitenbus ermöglicht dabei als Plattform erst den Betrieb dieser Nutzlast und ist in eigene Subsysteme unterteilt, die jeweils einen ganz spezifischen Teilaspekt der Bewältigung der Weltraumumgebung realisieren. Dabei handelt es sich zusammengefasst um:

- Lageregelung / Attitude and Orbit Control System (AOCS)
- Energieversorgung / Electric Power Distribution System (EPS)
- Thermalkontrolle / Thermal Control System (TCS)
- Kommunikation (Comms)
- Struktur
- On-Bord Datenverarbeitung / Data Handling (OBDH)

Die Lageregelung kann verschiedene Sensoren und Aktuatoren beinhalten, die der Lagebestimmung des Satelliten im Raum dienen und eine Beeinflussung dieser Lage ermöglichen. Dazu gehören auf Sensorseite z.B. Sternen- oder Sonnensensoren, Magnetometer oder Gyroskope. Aktuatoren können Triebwerke oder auch Drallräder sein. Eine präzise und sichere Lageregelung ist wichtig, um z.B. Solarkollektoren optimal zur

Sonne auszurichten oder gerichtete Instrumente der Nutzlast auf das gewünschte Ziel einzustellen.

Die Energieversorgung wird durch verschiedene Arten von Generatoren sichergestellt. Das können im gewöhnlichsten Fall Sonnenkollektoren aber z.B. auch Radionuklidbatterien sein. Batterien dienen als Puffer, um bei Betrieb mit Sonnenkollektoren einen Puffer für Phasen zu haben, in denen sich das Raumfahrzeug im Schatten der Erde befindet oder die Zellen nicht gen Sonne ausgerichtet sind. Auch zum Subsystem gehören die verschiedenen Wandler und die Verkabelung, die die verschiedenen Betriebsspannungen an die anderen Subsysteme verteilen.

Die Thermalkontrolle besteht aus Komponenten zur Überwachung und Regelung der Temperatur. Ein Raumfahrzeug ist stark schwankenden Temperaturen ausgesetzt, z.B. wenn es im Orbit um die Erde zwischen Tag- und Nachtseite wechselt. Aber auch zwischen sonnenabgewandter und -zugewandter Seite gibt es starke Temperaturunterschiede. Verstärkt wird diese Tatsache dadurch, dass durch fehlende Konvektion im Weltraum kein effizienter Wärmetransport möglich ist. Gerade für Elektronik und Nutzlasten mit engen Betriebstemperaturbereichen ist dies ein Problem. Gelöst wird es durch Radiatoren und Isolierungen. Heatpipes können für einen effizienten Wärmetransport sorgen.

Im Kommunikationssystem sind alle raumfahrzeugseitigen Komponenten zur Kommunikation mit der Bodenstation untergebracht. Je nach Konfiguration werden hier entsprechende Antennen, abhängig vom verwendeten Frequenzband gewählt. Empfangs- und Sendeverstärker führen das Signal üblicherweise zu einem Modem, das die Signale in einen Bitstrom umsetzt oder umgekehrt und mit dem Bordcomputer verbunden ist.

Die Struktur eines Raumfahrzeugs fasst alle Teile zusammen, die das Raumfahrzeug mechanisch ausmachen. Auch gibt es weitere Unterteilungen, bspw. in lasttragende Strukturen oder Baugruppenträger.

Das On-Board Data Handling enthält den Bordcomputer und seine Peripherie. Hier werden die Kommandos vom Boden verarbeitet und die Telemetrie für das Senden verarbeitet. Über das OBDH werden alle anderen Subsysteme gesteuert und überwacht. Auch die Nutzlast liefert ihre Daten an den Bordcomputer, der sie für die Nutzlasttelemetrie vorbereitet.

1.4.4 WICome

Raumfahrzeuge wie Satelliten und Sonden sind insbesondere in der Forschung oft Prototypen und keine Serienprodukte. Während kommerzielle Telekommunikationssatelliten oft auf gemeinsame Elemente wie einen standardisierten Bus zugreifen, der von einem

großen Integrator wie z.B. Boeing geliefert wird, sind die meisten Forschungssatelliten in all ihren Aspekten Einzelstücke. Die hohe geforderte Qualität gepaart mit dem großen finanziellen Risiko, dass das Scheitern einer Raumfahrtmission mit sich bringt führt dazu, dass auf bewährte Systeme gesetzt wird. Vergleichbar mit der Luftfahrt werden Innovationen anders als in anderen Branchen deutlich vorsichtiger eingeführt, um Risiken für die Mission zu minimieren.

Die Beschleunigung innovativer Entwicklungen macht das Testen unter Weltraumbedingungen unabdingbar. Nur hier kann die gesamte Kette der einzelnen Segmente unter realen Bedingungen evaluiert werden. Um das Risiko zu minimieren, bieten sich hier kleinere, kostengünstige Satelliten an, die als Technologieerprobungsträger dienen.

Das Deutsche Zentrum für Luft und Raumfahrt (DLR) startet hierzu in regelmäßigen Abständen Mikro- und Minisatelliten. Der hier vorgestellte Satellit aus der Compsat Reihe wird dabei ca. alle 5-6 Jahre mit einem Gewicht von etwa 200 kg gestartet. Eine weitere Missionsreihe (S2TEP) sieht es vor, einen kleineren Mikrosatelliten mit einem Gewicht zwischen 20 und 50 kg alle 3 Jahre zu starten. Beide Missionen sollen eine DLR-eigene, modulare Satellitenplattform schaffen, die nach neuesten Methoden entwickelt wird und einfach an ein entsprechendes Missionsziel adaptiert werden kann. Bewähren sich die eingesetzten Technologien im Orbit-Einsatz auf einem Satelliten der S2TEP Reihe sollen sie in Zukunft auch auf dem größeren COMPSAT eingesetzt werden. Der aktuell kurz vor dem Start stehende Compsat trägt den Namen Eu:CROPIS. Seine größte wissenschaftliche Nutzlast wird das Pflanzenwachstum unter unterschiedlichen Gravitationsbedingungen untersuchen. Die nächste Generation des Compsat wird dann unter anderem das WICome Experiment mitführen. Für die weitere Betrachtung bietet Eu:CROPIS aber eine gute Basis, um die Anforderungen an einen solchen Minisatelliten zu spezifizieren.

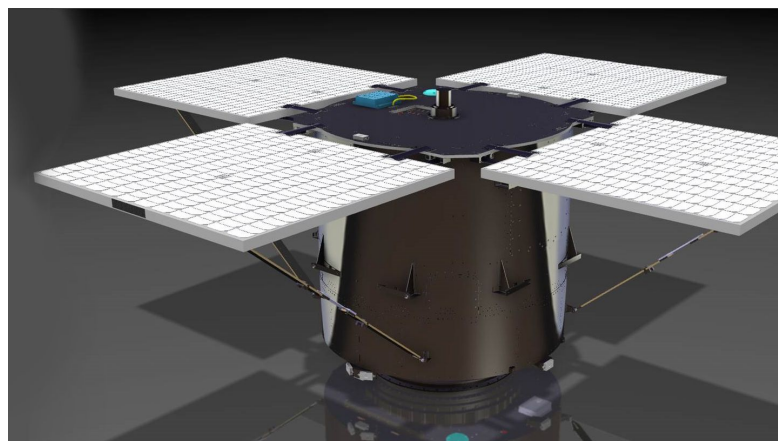


ABBILDUNG 1.1: Darstellung des Eu:CROPIS Satelliten⁶

⁶Quelle: <http://www.dlr.de/irs/desktopdefault.aspx/tabid-11288/>

Die Compsat Mission verfolgt einige innovative Ansätze, die im Rahmen des Betriebs und der Entwicklung des Satelliten evaluiert werden sollen. Grundsätzlich wird Compsat auf einer modularen Architektur basieren, die die Flexibilität erhöhen soll [Mic17]. Dies beinhaltet einen skalierbaren Bus, der je nach Bedarf aus unterschiedlichen Komponenten aufgebaut werden kann. Da auch die Payload variabel sein kann, können für den Bus z.B. andere Schnittstellen oder auch eine angepasste Rechenleistung vonnöten sein. Der Fokus der Mission im Gegensatz zum kleineren S2TEP Satelliten liegt nicht in erster Linie darin technische Systeme zu erproben, sondern auch eine wissenschaftliche Nutzlast zu betreiben. [Tho17]

Der On-Board Computer *COBC* (Compact On-Board Computer) zeichnet sich durch seine hohe Skalierbarkeit aus. Basierend auf einem Gehäuse Grunddesign lässt sich das Modul durch weitere Schnittstellen, Speicher und Prozessoren mit höherer Leistung einfach erweitern. Die Anbindung des WSN Experimentes an diesen wird dabei vermutlich über eine RS422 Schnittstelle erfolgen.

Auch die Software Plattform *OUTPOST* (Open modular softWare PlatfOrm for Spacecraft), die auf dem COBC betrieben wird, ist modular ausgelegt. Sie stellt Schichten zur Abstrahierung der Hardware und den unterschiedlichen, einsetzbaren Betriebssystemen zur Verfügung. Hinzu kommen Implementierungen von Logging, Middleware und verschiedenen im Raumfahrtbereich standardisierten Protokollen wie Space Packets oder auch des Packet Utilization Standard (PUS).

Die Entwicklung der Software wird modellbasiert durchgeführt, d.h. ein Großteil der Implementierung wird aus zuvor erstellten Modellen automatisch generiert. Unter anderem sind so Systemtests wesentlich einfacher durchzuführen. Dies ist auch Voraussetzung für die durchgeführte automatische Verifikation.

Auch die nötigen Kommunikationseinrichtungen werden durch den Einsatz von Software Defined Radio (SDR) flexibler. Bis auf die analogen Stufen der Transceiver kann so die gesamte Modulation/Demodulation und Filterung in Software realisiert werden. Dadurch ist auch dieses System zukünftig leichter anpass- und damit wiederverwendbar.

Das WICome Experiment soll die Einsetzbarkeit der UWB Technologie im Bereich der Intra-Satellitenkommunikation evaluieren und wird einen Verbund von Knoten betreiben, die zu Testzwecken Daten austauschen werden.

1.5 Gliederung der Arbeit

Die verbleibende Arbeit ist wie folgt gegliedert: Basierend auf den in dieser Einleitung vorgestellten Zielen und weiteren Quellen werden in der Anforderungsanalyse (siehe Kapitel 2 auf der nächsten Seite) die nötigen Anforderungen an das zu entwickelnde System aufgestellt und bewertet.

Im nächsten Kapitel (siehe Kapitel 3 auf Seite 21) werden dann die technischen Grundlagen von 802.15.4 erläutert und eine Auswahl von möglichen und bezüglich der Anforderungen relevanten Weiterentwicklungen beschrieben, die für die Umsetzung implementiert werden können.

Es folgt die Umsetzung (siehe Kapitel 4 auf Seite 50), in der auf Basis der Anforderungen und der beschriebenen Erweiterungen zu 802.15.4 ein Entwurf erfolgt, der eine sowohl technisch für UWB sinnvolle, wie zu den Anforderungen passende Erweiterung auswählt. Hier wird auch auf die verwendete Hardwareplattform und das ausgewählte Betriebssystem eingegangen.

Die implementierte Lösung wird anschließend im nächsten Kapitel evaluiert (siehe Kapitel 5 auf Seite 81). Es werden die Methoden zur Verifikation und deren Ergebnisse erläutert. Außerdem wird ein Evaluationsszenario beschrieben, das die umgesetzte Lösung in Beziehung zu vorhandenen Implementierungen setzt. Die Auswertung wird anschließend diskutiert.

Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick auf zukünftig mögliche Weiterentwicklungen (siehe Kapitel 6 auf Seite 93) ab.

Anforderungsanalyse

Der Einsatz von LRWPANs auf Raumfahrzeugen unterliegt besonderen Anforderungen, die nicht zwangsläufig denen aus den üblichen Anwendungsfeldern von drahtlosen Netzwerken des 802.15.4 entsprechen. Am vergleichbarsten sind hier Einsatzszenarien aus der Anlagensteuerung- und Überwachung in der Industrie für die bereits einige Modifikationen des Standards vorgenommen wurden, um den zuverlässigen Betrieb zu gewährleisten.

Die in diesem Kapitel ermittelten Anforderungen beziehen sich allein auf den MAC Layer, der eine wichtige Rolle für den Betrieb eines WSN insgesamt spielt. Ihr Fokus liegt auf Robustheit und Zuverlässigkeit beim Betrieb in räumlich begrenzter Ausdehnung des Netzes. Als Anhaltspunkt hierfür dienen unter anderem Experteninterviews, die in einer früheren Arbeit bereits geführt wurden [Hau16]. Hier wurden die Verantwortlichen der unterschiedlichen Subsysteme des Eu:CROPIS Satelliten (siehe 1.4.3 auf Seite 9) zu ihren Anforderungen an ein mögliches WSN befragt.

Eine weitere Quelle für Anforderungen ist das Magenta Book der CCSDS zum Einsatz von WSN auf Raumfahrzeugen [CCS13]. In diesem Band werden einige Empfehlungen zu Betriebsparametern gegeben, die für den Betrieb von drahtlosen Sensornetzwerken auf Raumfahrzeugen wichtig sind und welche Probleme bei der Auslegung betrachtet werden müssen. Das grundlegende Ziel ist das Ersetzen von Kabeln im Raumfahrzeug, die zur Vernetzung der verschiedenen Komponenten verwendet werden. Ein in vielen Satelliten eingesetztes kabelgebundenes Bussystem ist *SpaceWire*, das von der ECSS standardisiert wurde [Eur08]. Es bietet eine hohe Übertragungssicherheit, geringe Latenz und ein deterministisches Übertragungsverhalten bei gleichzeitig geringer Komplexität der Hardware und benötigter Software.

Drahtlose Netzwerke auf einem Satelliten, die die Subsysteme anbinden sollen, müssen diese Anforderungen ebenfalls erfüllen, wenn sie in Zukunft zumindest Teile der Kabelinfrastruktur ersetzen sollen. Die folgenden Abschnitte liefern nun eine Beschreibung und Begründung der einzelnen Anforderungen im Detail.

2.1 Anforderungen an den MAC Layer

Der MAC Layer entscheidet über grundlegende Eigenschaften des Netzes und ist für die Sicherstellung der Übertragung verantwortlich. Zusätzlich entscheidet sein Design in nicht geringem Maße über den Energieverbrauch der Sende- und Empfangshardware. Ebenso wichtig ist die Einbettung des Layers in Betriebssystem und Netzwerkstack. Die Anforderungen zu diesen Bereichen sind im Folgenden in diese Gruppen eingeteilt worden, wobei für jede Gruppe die funktionalen und nicht-funktionalen Anforderungen aufgeführt sind. Jeder Gruppe ist eine Begründung vorangestellt, die die Wahl der spezifischen Anforderungen kurz darlegt.

2.1.1 Allgemein

Auf einem Satelliten existiert üblicherweise ein zentraler Bordcomputer, der sowohl die Telekommandos von der Erde empfängt und an die entsprechenden Subsysteme weiterleitet, als auch Telemetrie der Systeme und der Nutzlast sammelt, um sie an die Bodenstation weiterzuleiten. Für ein drahtloses Sensornetz bietet sich daher eine Sterntopologie an mit einem zentralen Verwaltungsknoten¹, der in der Nähe des OBC montiert ist, um von dort mit möglichst kurzer Verkabelung an einen Datenbus und die Stromversorgung angeschlossen zu werden. Die mögliche Ausprägung als Mesh Netzwerk, bei dem Übertragungen über mehrere Knoten geleitet werden ist auf einem Raumfahrzeug eher ungeeignet. Zum einen, weil dies oft nicht-deterministische Latenzen zur Folge hat und zum Anderen, da die Entfernungen, die ein solches Netz überbrücken muss auf einem Satelliten höchstens wenige Meter betragen.

Räumlich gruppierte Sensoren (z.B. Sonnen-, oder Drehratensensoren) können über einen einzelnen Knoten an das Netz angebunden werden. Bei der Vielzahl an möglichen Systemen, die angebunden werden sollte es aber trotzdem möglich sein, mindestens 20 Knoten in einem Netz anbinden zu können.

REQ-01 Sterntopologie Das WSN muss die Möglichkeit bieten eine Sterntopologie mit einem zentralen Verwaltungsknoten zu bilden.

REQ-02 Knotenanzahl Der MAC Layer des WSN muss in der Lage sein, mindestens 20 Knoten sicher zu verwalten.

¹in 802.15.4 *Coordinator* genannt

2.1.2 Leistung

Die definierenden Leistungsparameter in einem drahtlosen Sensornetzwerk sind Latenz und Datendurchsatz. Der MAC Layer fügt durch seine Funktionsweise immer einen gewissen Overhead ein. Je nach gewünschter Funktionalität kann die Leistung darunter leiden. In [Hau16] wurden bereits Interviews mit Experten der einzelnen Subsysteme für das COMPSAT System Eu:CROPIS zur möglichen Einsetzbarkeit von WSN allgemein geführt, die eine Aufschlüsselung über die mögliche Gesamtanzahl und ihre Verwendung gibt. Einige zusätzliche Befragungen der Experten wurden im Rahmen dieser Arbeit durchgeführt, um die Latenz- und Durchsatzanforderungen genauer quantifizieren zu können.

In Tabelle 2.1 sind die Verbindungsanforderungen der einzelnen Subsysteme, für die WSN in Frage kommen, aufgeschlüsselt. Zu beachten ist, dass die Anzahl der Sensoren hier nicht der Anzahl der einzusetzenden WSN Knoten entspricht. Viele der gleichartigen, wie Sonnen- oder Magnetfeldsensoren sind oft räumlich nah beieinander angebracht, so dass sie über einen WSN Knoten in das drahtlose Netz eingebunden werden können. Der Großteil der eingesetzten Sensoren wird zurzeit über Bussysteme wie RS-485 an den OBC angebunden. Wird diese Verbindung durch WSN ersetzt müssen lediglich die Nutzdaten der Sensoren übertragen werden. Es ist danach für dieses Beispiel mit einer minimalen Gesamtdurchsatzrate von 1,1 kB/s netto zu rechnen. Um Retransmits, Overhead und andere unvorhergesehene Übertragungen aufnehmen zu können, sollten mindestens 3 kB/s zur Verfügung stehen.

Subsystem	Sensortyp	Anzahl	Echtzeit	Rate	Datenrate
ACS	Sonnensensor	10	ja	10 Hz	600 B/s
ACS	Magnetfeldsensor	2	ja	5 Hz	160 B/s
ACS	Drehratensensor	4	ja	5 Hz	160 B/s
ACS	GPS	2	ja	1 Hz	150 B/s
TCS	Temperatursensor	60	nein	0,1 Hz	24 B/s
EPS	Spannungs- / Stromsensor	20	ja	0,1 Hz	16 B/s

TABELLE 2.1: Leistungsanforderungen der einzelnen Subsysteme

Besonders hervorzuheben ist die Echtzeitanforderung des Lageregelungssystems. Um z.B. während eines gesteuerten Manövers die Kontrolle zu behalten bedarf es einer maximalen Verzögerung von 10 ms. Dabei handelt es sich um die Zeit, nach der die Daten der Anwendungsschicht zur Verfügung stehen müssen, also beim Empfang z.B. bereits durch den Netzwerkstack bis hin zur obersten Schicht verarbeitet sein müssen. Der sendende Knoten nutzt diese Zeit, um die Daten durch den Stack bis hinunter zum PHY Layer zu verarbeiten, wohingegen der empfangende Knoten die Zeit zur Dekodierung der Pakete bis hinauf zur Anwendungsschicht benötigt. Dabei entfällt ein Großteil auf

die Synchronisierung mit den Sendezeiten im MAC Layer. Die Verarbeitung in den hier verwendeten übergeordneten Schichten liegt deutlich unter 1 ms. Aufgerundet bleiben 8 ms für die Verarbeitung von MAC zu MAC übrig. Dabei handelt es sich um eine harte Echtzeitanforderung, deren Latenz zu jeder Zeit erreicht werden muss, da die Daten sonst unbrauchbar wären, oder aufwändig interpoliert werden müssten. Diese Latenz muss deterministisch sein und darf z.B. nicht durch Pufferung unbekannter Länge zur Ermöglichung einer Funktion variabel sein. Trotz dieser Einschränkung ist es weiterhin nötig, auch nicht vorhersehbare variable Datenströme zu verarbeiten, z.B. wenn nach Abfrage Payloaddaten übertragen werden müssen.

REQ-03 *Durchsatz* Das WSN muss einen Gesamtdurchsatz von mindestens 3 kB/s an Nutzdaten ermöglichen.

REQ-04 *Latenz* Pakete, die über den MAC Layer gesendet werden dürfen eine Latenz von 10 ms nicht überschreiten.

REQ-05 *Determinismus* Eine etablierte Latenz darf durch Funktionen des MAC Layers nicht variiert werden.

REQ-06 *Variable Datenrate* Der MAC Layer muss Dienste erlauben, die eine wechselnde Datenrate aufweisen.

2.1.3 Robustheit und Zuverlässigkeit

Neben den Leistungsparametern spielt die Zuverlässigkeit und die Fehlertoleranz des Netzes eine wichtige Rolle. Daher muss ein MAC Layer in der Lage sein, Paketbestätigungen und ggf. schnelle Neuversendungen von Paketen durchzuführen, die verloren gegangen sind. Insbesondere für die Versendung zeitkritischer Messwerte, wie dies z.B. beim Lageregelungssystem der Fall ist, ist dies wichtig.

Ein weiteres Problem stellt die Rekonfiguration des Netzes dar. Neue Knoten müssen sich im Normalfall bei einem zentralen Verwaltungsknoten anmelden. Viele sehr einfach konstruierte oder auf geringen Overhead optimierte MAC Layer bieten eine solche Möglichkeit nicht, was eine komplette, zeitintensive Neuinitialisierung des Netzes zur Folge hat. Nun ist die Konfiguration und Anzahl der Netzwerkknoten auf einem Raumfahrzeug für gewöhnlich festgelegt. Es ist jedoch üblich, dass Geräte redundant ausgelegt sind, um bei Ausfällen direkt einen Ersatz aktivieren zu können. Häufigste Ursache für den Ausfall ist Strahlung im Weltraum, die verschiedene negative Effekte auf Elektronik hat. Sie kann von kurzzeitigen, reversiblen Störungen (*transient / upset*) bis zur Zerstörung von integrierten Schaltkreisen führen (*latchups / Total Ionizing Dose*). Da die verwendete Hardware zumeist aus CoTS² Komponenten besteht, die nicht gesondert

²Commercial off the Shelf

strahlungsgehärtet wurde, ist sie besonders von dieser Art des Ausfalls betroffen. Ein drahtloses Netzwerk muss daher in der Lage sein, sich selbst bei Veränderung der Knoten zu rekonfigurieren. Zudem darf die Aufnahme eines Ersatzknotens in das Netz und die Rekonfiguration der Parameter nicht mehr als 2 s in Anspruch nehmen, da dies die festgelegte Maximalzeit ist, für die der OBC für kritische Subsysteme Daten extrapolieren kann.

REQ-07 ACKs und Retransmits Der MAC Layer muss die Möglichkeit zum Bestätigen und zur Wiederholung von Nachrichten bieten.

REQ-08 Rekonfiguration Der ausgewählte Standard soll die Möglichkeit zur dynamischen Rekonfiguration bieten.

REQ-09 Rekonfigurationsdauer Muss eine Rekonfiguration vorgenommen werden, so darf diese nicht mehr als 2 s in Anspruch nehmen, um den Betrieb nicht zu gefährden.

REQ-10 Knotenausfall Der Ausfall eines Knotens und die eventuelle Wiederaufnahme der Kommunikation in gleicher Konfiguration darf den Betrieb des restlichen Netzes z.B. durch eine nicht steuerbare, automatische Rekonfiguration nicht beeinträchtigen.

2.1.4 Energie

Die Energieversorgung spielt besonders für die einfachen Knoten im Netzwerk eine wichtige Rolle. Der zentrale Coordinator benötigt mehr Energie, da er Netzwerkverwaltungsaufgaben übernimmt und ist im Normalfall kabelgebunden mit Strom versorgt. Die eigentlichen Sensorknoten sind oft auf Batteriebetrieb ausgelegt und könnten damit sowohl ohne Datenkabel als auch ohne Kabel zur Stromversorgung betrieben werden. Bei Knoten, die an der Außenhülle des Satelliten angebracht sind ist z.B. eine autarke Versorgung mit kleinen Solarzellen denkbar. Den Energieverbrauch auf ein Minimum zu reduzieren muss dabei Ziel der Kombination aus Soft- und Hardware sein.

Dem MAC Layer kommt dabei besondere Bedeutung zu, denn er kommandiert den Physical Layer und damit direkt die Funkhardware. Ist sie aktiv und sendet oder lauscht, verbraucht sie einen Großteil der Energie. Hinzu kommt, dass auch der steuernde Controller sich nicht im Schlafmodus befinden kann und weiter Energie verbraucht. Die Zuweisung möglichst kurzer Aktivitätsphasen an die Knoten bei gleichzeitigem Erhalt der Kommunikationsflexibilität muss daher vom MAC Layer übernommen werden.

REQ-11 Slot Scheduling Der MAC Layer soll durch Verteilung von Sende- und Empfangszeiten für die einzelnen Knoten möglichst lange Inaktivitätsphasen und damit das energiesparende Schlafen des Gesamtsystems fördern.

REQ-12 Energieeffizienz Der Energieverbrauch aller Knoten des WSN darf für das WICome Experiment 595 mW im Durchschnitt nicht überschreiten.

2.1.5 Einbettung

Damit sich der MAC Layer in einem System flexibel verwenden lässt, sollte er in einem vorhandenen OSI Stack eingebettet werden. Dies erlaubt Applikationen auf höherer Ebene auf standardisierte und portable Weise miteinander zu kommunizieren. Auch für die Implementierung des MACs selbst ist es von Vorteil auf Standards aufzubauen, sodass zukünftig ggf. unterschiedliche Geräte am gleichen Netz teilnehmen können.

Viele der Applikationen, die auf Raumfahrzeugen eingesetzt werden, müssen Daten z.B. zur Regelung von Prozessen in Echtzeit verarbeiten. Daher ist es nötig, dass auch das Basisbetriebssystem des Sensornetzes echtzeitfähig ist. Hinzu kommt, dass die verwendete Mikrocontrollerplattform stark begrenzte Ressourcen zur Verfügung stellt. Diese Limitierungen dürfen nicht überschritten werden.

REQ-13 *OSI Stack Integration* Der zu entwickelnde MAC Layer soll in einen vorhandenen OSI Stack integriert werden.

REQ-14 *MAC Standard* Der MAC Layer soll auf einem Standard basieren, der die zukünftige Interoperabilität sicherstellt.

REQ-15 *Echtzeitbetriebssystem* Eine Integration in ein echtzeitfähiges Betriebssystem, dass die verwendete Hardware unterstützt, muss vorgenommen werden.

REQ-16 *Verwendete Ressourcen* Der verwendete STM32L151CC Mikrocontroller bietet lediglich 256 kB an Flash und 32 kB an RAM. Die Implementierung muss diese Einschränkungen einhalten, um auf dem Controller lauffähig zu sein.

2.2 Zusammenfassung

Die in diesem Kapitel vorgestellten Anforderungen beschreiben einen MAC Layer für ein sicheres, deterministisches Sensornetz mit geringen Latenzen. Trotz der hohen geforderten Übertragungssicherheit bei oftmals festgelegten periodischem Datenaufkommen soll er trotzdem flexibel auf variables Datenaufkommen reagieren können. Ein weiterer wichtiger Punkt ist der Einsatz von anerkannten Standards und die Einbettung in etablierte Systeme zur einfachen Nutzung. Aufgrund der tiefen Einbettung eines MAC Layers in den Gesamtkontext eines Netzwerksystems berühren die Anforderungen auch viele allgemeine Aspekte, wie z.B. die Echtzeitfähigkeit.

Die Evaluation des Systems wird mit Bezug auf die Anforderungen durchgeführt und ist in Kapitel 5 auf Seite 81 genauer beschrieben. Mit der in diesem Kapitel durchgeführten Analyse lässt sich somit auch bereits die erste Forschungsfrage: »Was sind die Anforderungen an ein drahtloses Sensornetzwerk, das in Weltraumumgebungen betrieben werden muss?« beantworten. Tabelle 2.2 gibt einen Überblick über die ermittelten Anforderungen und die Verifikationsmethode, die angewandt werden soll.

Kürzel	Anforderung	Verifikation
REQ-01	Sterntopologie	Review
REQ-02	Knotenanzahl	Test
REQ-03	Durchsatz	Test
REQ-04	Latenz	Test
REQ-05	Determinismus	Review / Test
REQ-06	Variable Datenrate	Test
REQ-07	ACKs und Retransmits	Test
REQ-08	Rekonfiguration	Test
REQ-09	Rekonfigurationsdauer	Test
REQ-10	Knotenausfall	Test
REQ-11	Slot Scheduling	Review
REQ-12	Energieeffizienz	Test
REQ-13	OSI Stack Integration	Review / Test
REQ-14	MAC Standard	Review
REQ-15	Echtzeitbetriebssystem	Review
REQ-16	Verwendete Ressourcen	Review

TABELLE 2.2: Zusammenfassung der ermittelten Anforderungen an den MAC Layer

Technische Grundlagen

Die Basis der in dieser Arbeit entwickelten MAC Erweiterung bildet der IEEE Standard 802.15.4. Populär ist dieser geworden durch Anwendungen wie z.B. Zigbee und MiWi, die im Bereich der Heimautomatisierung und Unterhaltungselektronik mittlerweile eine weite Verbreitung gefunden haben.

Ebenfalls interessante neue Standards im Bereich der drahtlosen Sensornetze wie 802.15.4g oder der LORA PHY Layer sind aufgrund ihrer Reichweite von 20-80 km eher für sog. SUNs (Service Utility Networks) ausgelegt und daher nicht Fokus dieser Untersuchung.

Auf Basis von 802.15.4 gab es seit Einführung der ersten Revision des Standards 2003 eine Reihe von verschiedenen Erweiterungen. Insbesondere für die Kombination der alternativen Bitübertragungsschicht für Ultrabreitband Übertragungen mit Features aus den 802.15.4 MAC Spezifikationen soll in diesem Kapitel eine Grundlage geschaffen werden.

802.15.4 wurde für diese Arbeit ausgewählt, da der Standard Zuverlässigkeit und eine geringe Komplexität mit der für Sensornetze oft vernachlässigbaren Einschränkung eines geringen Datendurchsatzes bietet.

Die in diesem Kapitel vorgestellten Standards nach 802.15.4 definieren im Wesentlichen den physischen Medienzugang und die Sicherungsschicht nach dem OSI Modell. Das OSI Modell definiert Schichten, die für genau einen funktionalen Abschnitt einer Netzwerkarchitektur zuständig sind. Die einzelnen Schichten werden üblicherweise so designt, dass sie wohldefinierte Schnittstellen zu den angrenzenden Schichten haben (siehe Abb. [3.1 auf der nächsten Seite](#)). So stellt bspw. die Sicherungsschicht eine Schnittstelle zur Verfügung, mit der die Vermittlungsschicht eine möglichst fehlerfreie Übertragung z.B. für das Routing zur Verfügung hat. Der MAC Layer ist die Schnittstelle zu erweiterten Features der höheren Schichten.

Standards auf Basis des OSI Modells sind wichtig, da sie Interoperabilität fördern. In der Regel lassen sich die Hardwaremodule, die in WSN eingesetzt werden einfach als Bytepuffer nutzen, über den Daten geschrieben werden, sodass nur die Bitübertragungsschicht genutzt wird. Dies ist bei Kommunikation von wenigen Funkmodulen z.B. bei einer einfachen Punkt-zu-Punkt Verbindung relativ unproblematisch, führt jedoch bei größeren

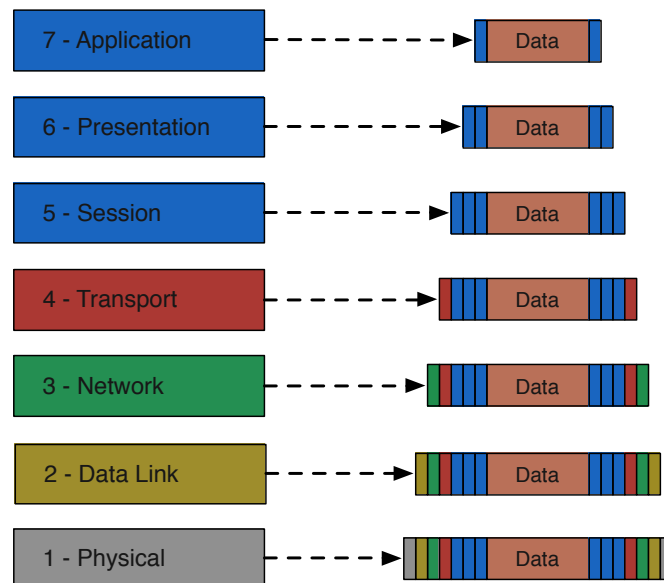


ABBILDUNG 3.1: OSI-Schichtenmodell für Netzwerkprotokolle

Installationen, insbesondere mit höheren Robustheitsanforderungen schnell zu Problemen.

Die Einbettung solcher Module in das Ökosystem der OSI Layer bietet viele Vorteile. So sind insbesondere die höheren Layer auf vielen Plattformen, wie den oft in Kombination mit den nötigen Funkmodulen eingesetzten Mikrocontrollern, bereits vorhanden. Sie werden auf breiter Front eingesetzt und sind erschöpfend getestet worden. Durch die Implementierung eines MAC Layers lassen sich diese also einfach anbinden und so erweiterte Fähigkeiten, wie Adressierung, Routing und Transportkontrolle bis hin zu Anwendungsprotokollen wie CoAP oder HTTP nutzen. Üblicherweise werden die Schichten so spezifiziert, dass sie klare Dienste zur jeweils darüber liegenden Schicht anbieten, die diese nutzt, um Daten zu empfangen oder zu übertragen.

Die aktuell am häufigsten eingesetzten Technologien im Bereich der WSN sind Bluetooth Low Energy (BLE) und 802.15.4. Die Autoren von [MPT13] sehen in ihrer Untersuchung in Bezug auf Energieeffizienz und reinen Durchsatz beim Vergleich der beiden Technologien BLE in beiden Punkten als überlegen an. Es wurde jedoch festgestellt, dass die Implementierung eines BLE Stacks ca. das Vierfache des Programmspeichers einer 802.15.4 Implementierung benötigt [MPT13, S. 603]. Zu den weiteren Nachteilen in Bezug auf die Anwendung in Sensornetzwerken gehören dabei die Beschränkung von BLE auf ein einzelnes Netz ohne die Möglichkeit von mehreren Hops zwischen den Knoten. Die Autoren konstatieren, dass diese Topologieprobleme die Flexibilität von BLE zu stark einschränken, um in anderen Bereichen als menschenbezogenen Anwendungen (personal area networks) eingesetzt zu werden.

Die augenscheinlichen Vorteile in Bezug auf Durchsatz und Energieeffizienz sind bei BLE vor allen Dingen beim Einsatz im Consumer Bereich interessant. Auf Basis von 802.15.4 sind dagegen einige Erweiterungen und Nachfolgestandards entstanden, die z.B. die für Industrieanwendungen geforderte Zuverlässigkeit liefern.

Standards auf Basis von 802.15.4 sind für die Anbindung von Sensoren und Aktoren ideal. Sie sind dafür ausgelegt worden, geringe Datenraten bei hoher Energieeffizienz zu liefern. Viele Erweiterungen zielen außerdem darauf ab, die Zuverlässigkeit der Datenübertragungen zu optimieren. Auch die Begrenzung der Latenz der Übertragungen ist hier oft ein wichtiges Designkriterium der MAC und PHY Layer. Geringe Latenzen sind z.B. in Industriesteuerungsanwendungen mit extrem zeitkritischen Regelschleifen notwendig. Übertragbar ist dies natürlich auch auf Raumfahrtanwendungen. Die Steuerung des Lageregelungssystems eines Raumfahrzeugs erfordert z.B. die sehr zeitgenaue Lieferung von Sensordaten und entsprechend ebenso genau übertragene Befehle an die Aktoren des Systems. Dabei muss die Übertragung entsprechend zuverlässig erfolgen. So dürfen z.B. nicht unbemerkt Datenpakete verloren gehen.

Die verwendeten Transceiver für die physical Layer von 802.15.4 weisen typischerweise eine geringe elektronische Komplexität auf. Dies ist gerade für Elektronik, die in Raumfahrtumgebungen eingesetzt werden soll ein wichtiger Faktor. Neben geringerem Energie- und Platzverbrauch lassen sich weniger komplexe Transceiver einfacher und günstiger gegen Strahlungseinflüsse schützen.

Im Folgenden werden die Grundlagen des Standards 802.15.4 und seiner Erweiterungen erläutert.

3.1 IEEE 802.15.4 Standard

Die IEEE hat im Jahr 2003 mit dem Standard 802.15.4 den ersten Schritt zur Standardisierung der Kommunikation in den aufkommenden Sensor- und Aktornetzwerken geschaffen. Innerhalb der Arbeitsgruppe 802.15, die sich *Wireless Personal Area Networks* (WPAN) verschrieben hat, existieren verschiedene Arbeitsgruppen, von denen die vermutlich bekannteste mit 802.15.1 die Standards für Bluetooth definiert. Die entsprechende Task Group 4 kümmert sich dabei um die Standardisierung von sog. Low Rate WPANs (LR-WPAN), die eine geringe Datenübertragungsrate bei gleichzeitig geringer Komplexität und niedrigem Energieverbrauch bieten.

802.15.4 definiert die unteren beiden Layer des OSI Schichtenmodells, die Sicherungsschicht (MAC), sowie die Bitübertragungsschicht (Physical/PHY). Aufbauend auf diesen

gibt es eine Reihe von Anwendungsspezifikationen, die höhere Schichten definieren, um Sensornetzwerke sinnvoll aufbauen zu können.

Zu den bekannteren Anwendungen, die den Standard nutzen gehört *ZigBee*, eine Spezifikation von höheren Layern, die eine Routing-, Sicherheits- und eine Anwendungsschicht definieren. Hauptsächlich genutzt werden ZigBee Module besonders im Bereich der Heimautomatisierung und der Unterhaltungselektronik.

3.1.1 Netzwerktopologie

In einem 802.15.4 Netz kann es zwei unterschiedliche Arten von Geräten geben:

- FFD – Full Function Device. Ein Gerät, das in der Lage ist neben der Funktion als normaler Netzknoten auch als sog. Personal Area Network Coordinator (PAN Coordinator) oder einfacher Coordinator aufzutreten.
- RFD – Reduced Function Device. Ein Gerät, das üblicherweise batteriebetrieben ist und nur über geringe Ressourcen verfügt. Es kann nur mit FFDs kommunizieren.

Ein PAN Coordinator wird benötigt, um ein 802.15.4 Netz aufzubauen. Dieser wählt eine PAN ID, die in der Umgebung noch nicht verwendet wird, wodurch nun FFDs und RFDs dem Netzwerk beitreten können. Er ist auch für die Initiierung des Konfliktlösungsmechanismus zuständig, wenn eine gleiche PAN ID im Empfangsbereich des Netzwerks auftritt.

Grundsätzlich definiert der Standard[IEE16] zwei verschiedene Netzwerktopologien, die in einem solchen LR-WPAN möglich sind (siehe Abb. 3.2 auf der nächsten Seite):

Zum einen die Stern Topologie, bei der alle Knoten direkt mit dem PAN Coordinator kommunizieren und dieser eine etwaige Kommunikation zwischen den Knoten ermöglicht.

Zum anderen die Peer-to-Peer Topologie, bei der FFDs auch direkt untereinander kommunizieren dürfen. Da RFDs auch in dieser Topologie weiterhin nur mit FFDs kommunizieren können, gibt es zwischen ihnen keine Peer-to-Peer Verbindungen. Um Daten über weite Distanzen zu übertragen gibt es hier die Möglichkeit sog. Cluster-Tree Netze aufzubauen. Bei diesen bilden die RFDs die Blätter einer Baumstruktur. Sie verbinden sich mit FFDs, die die Coordinator Rolle einnehmen, also Verbindungen von anderen Geräten entgegennehmen können. Diese können sich nun in einer Baumstruktur zu weiteren Koordinatoren verbinden. Die Wurzel des Baumes bildet dann ein Knoten, der die Rolle des übergeordneten PAN-Coordinators einnimmt.

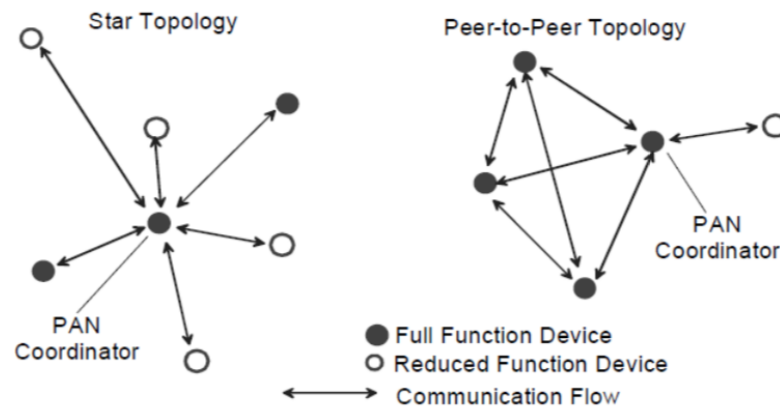


ABBILDUNG 3.2: Mögliche Netzwerktopologien in einem 802.15.4 Netz
[IEE16, vgl. S. 46]

Routing in solchen Mesh Netzwerken ist nicht Teil des Standards und wird auf höheren Schichten realisiert. Dies kann z.B. durch RPL¹ oder auch durch den Network Layer in ZigBee realisiert werden.

3.1.2 Physical Layer

Der Physical (PHY) Layer oder auch Bitübertragungsschicht nach dem OSI Referenzmodell definiert den physischen Übertragungsweg für die Daten. Im Fall der hier vorliegenden drahtlosen Übertragung werden also Frequenzbänder und Modulationsarten vorgegeben, die es Transceivern, die den Standard implementieren erlauben miteinander zu kommunizieren.

Der Standard enthält in der aktuellen Version sehr viele physische Übertragungswege, die für verschiedene Anwendungsfälle konzipiert und als Erweiterung über die Jahre eingebracht wurden. Grundsätzlich werden jedoch an alle PHYs dieselben Anforderungen gestellt [IEE16, S. 381]:

- Alle PHYs nutzen zwei Oktette als Frame Check Sequence (FCS), also eine angehängte Prüfsumme an die Frames
- Die Aktivierung und Deaktivierung des Transceivers muss möglich sein.
- Energy Detection – Der Transceiver muss in der Lage sein, festzustellen, ob auf dem gegebenen Kanal momentan ein Signal anliegt, das einen bestimmten Schwellwert überschreitet.

¹Routing Protocol for Low Power and Lossy Networks – Ein von der IETF spezifiziertes Distanzvektor Routing Protokoll für Sensornetze, das Routing in Mesh Netzwerken unterstützt

- Link Quality Indicator – Eine Metrik muss geliefert werden, die die Qualität des empfangenen Signals angibt, also wie leicht sich dieses demodulieren lässt. Verwandt ist die Received Signal Strength Indication (RSSI), die Auskunft über die Stärke des Signals gibt, aber keine Aussage über die Qualität trifft.
- Clear Channel Assessment (CCA) – Für das CSMA/CA wird neben Energy Detection auch die Erkennung eines Signals, das der aktuellen Konfiguration des PHYs entspricht verwendet.
- Wahl der Kanalfrequenz
- Übertragung und Empfang von Daten

Der ursprüngliche Standard aus dem Jahre 2003 definiert zwei mögliche Physical Layer. Der erste verwendet im Frequenzbereich von 868 Mhz (Europa) / 915 Mhz (USA) bei Bandbreiten von 300, respektive 600 kHz eine binäre Phasenmodulation (BPSK). Der zweite mögliche Layer liegt im weltweit zugänglichen ISM Band bei mittleren 2450 Mhz und nutzt bei 2 Mhz Kanalbandbreite eine Quadratur Phasenmodulation (QPSK).

Beide setzen das Frequenzspreizverfahren Direct Sequence Spread Spectrum (DSSS) ein. Bei der Frequenzspreizung wird das Nutzsignal mit einem weiteren Signal multipliziert, sodass sich das übertragene Signal auf eine größere Bandbreite verteilt. Auf der Empfängerseite wird das Signal wieder durch nochmalige Multiplikation mit dem Spreizsignal entspreizt, bevor es weiterverarbeitet wird. Etwaige Störungen, die auf dem Signalweg eingetragen werden, werden durch die Entspreizung des Signals stark gedämpft, da sie nur einer einmaligen Multiplikation unterliegen. Dies macht das Signal wesentlich unanfälliger gegenüber schmalbandigen Störungen, Echos und Interferenzen, führt aber auch zu einer Verteilung der Sendeenergie über das breitere Spektrum, wodurch andere Dienste weniger gestört werden. Gerade im ISM Band ist die Verträglichkeit der unterschiedlichen Technologien eines der Designziele der IEEE bei ihren Standards. Bei der hier eingesetzten Variante des *Direct Sequence* Spreadings basiert die Spreizung des Signals auf einem Code, der sich aus dem Nutzsignal ergibt. Im Fall von 802.15.4 wird jedes Datensymbol auf eine Kette von vorgegebenen Symbolen abgebildet, den sog. Chips. Die Anzahl der Chips mit denen ein Symbol ersetzt wird gibt die Chiprate vor. Je höher die Chiprate ist, desto mehr Bandbreite wird benötigt, um die Symbole zu übertragen, die Spreizung ist somit größer. [Die06, vgl. S.10-12]

Mit dieser Spreizung werden im 868 Mhz Band mit 300 kchips/s 20 kBit/s an Nutzdatenrate erreicht, respektive 40 kBit/s bei einer Chiprate von 600 kchip/s im 915 Mhz Band. Im ISM Band sind durch die höhere Bandbreite 250 kBit/s möglich. Neuere Erweiterungen im Standard können mit anderen Spreizverfahren auch auf den für BPSK genutzten Bändern eine Übertragungsrate von 250 kBit/s erreichen.

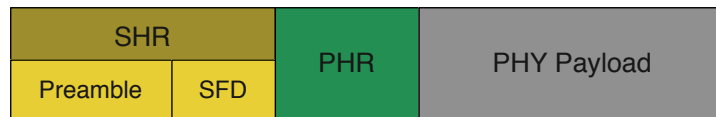


ABBILDUNG 3.3: Struktur einer 802.15.4 PPDU [IEE16, nach S. 411]

Der Standard definiert eine *Physical Protocol Data Unit* (PPDU), die den eigentlichen Nutzdaten zwei Header voranstellt (siehe Abb. 3.3). Der Synchronization Header (SHR) enthält hierbei eine feste Präambel und den Start Frame Delimiter (SFD), mit denen ein Empfänger sich synchronisieren und auf den Empfang vorbereiten kann. Der folgende Physical Header (PHR) enthält die Länge des folgenden Nutzdatenstroms in Byte. Es können maximal 127 Byte an Payload Daten pro PPDU übertragen werden. Die Fragmentierung größerer Datenblöcke muss daher auf höheren Ebenen erfolgen, z.B. wie in WSN oft üblich über 6LowPAN.

Die PPDU ist die letzte Schicht, die eine reine bitweise Repräsentation darstellt. Es folgt die Konvertierung entsprechend der Modulationsart zu Symbolen in QPSK oder auf eine Differenzcodierung, anschließend die Aufspreizung durch Ersetzung der Daten durch Chips und schließlich die Modulation des Signals, um es zu versenden.

3.1.3 MAC Layer

Im Gegensatz zu anderen Standards, in denen die Sicherungsschicht in die obere Logical Link Control (LLC) und die untere Media Access Control (MAC) Schicht unterteilt wird, wird in 802.15.4 aus Gründen der Komplexitätsreduktion auf den LLC Layer verzichtet. Darüber liegende Implementierungen können einen LLC oder andere Verfahren des Zugriffs auf den MAC Layer nutzen.

Wie in 3.1.1 auf Seite 24 beschrieben kommunizieren Reduced Function und Full Function Devices in einem PAN. Ein solches Netz kann in zwei grundsätzlichen Konfigurationen betrieben werden: In einem *beacon-enabled* und einem *nonbeacon-enabled* Modus.

Im *beacon-enabled* Modus wird eine *superframe* genannte Struktur zur Kommunikation zwischen den Geräten verwendet (siehe Abb. 3.4 auf der nächsten Seite). Dabei sendet der PAN-Coordinator in regelmäßigen Abständen Beacons aus. Der Zeitraum zwischen diesen beiden Beacons ist der Superframe. Dieser lässt sich optional in einen aktiven und einen inaktiven Teil unterteilen, wobei Geräte während des inaktiven Teils schlafen und somit Energie sparen können.

Die aktive Zeit wiederum ist in gleich lange Zeitabschnitte, die Slots unterteilt. Von diesen bildet ein Teil eine Contention Access Period (CAP) und der restliche Teil eine

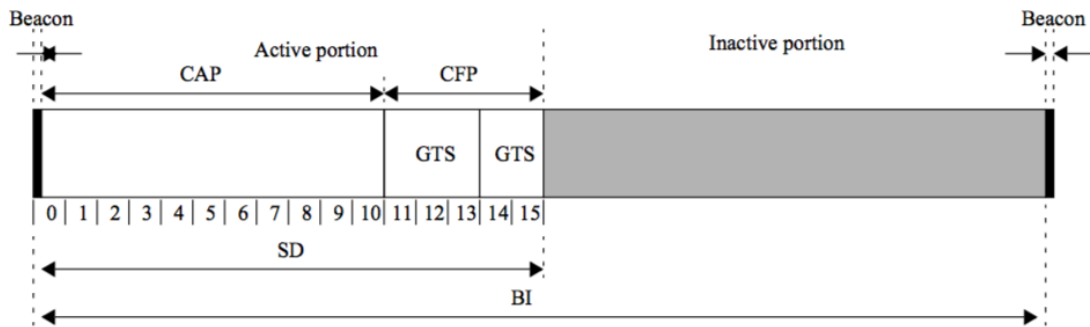


ABBILDUNG 3.4: Beispiel einer Superframe Struktur [IEE16, S. 58]

Contention-free Period (CFP). Während der Slots können Daten übertragen werden. Um der Verarbeitung im MAC Layer genügend Zeit zu geben, wird ein *Inter Frame Space* (IFS) zwischen den Frames eingehalten.

In der CAP konkurrieren die Geräte mit einem Wunsch zur Kommunikation um die verfügbaren Slots mittels des *Carrier Sense Multiple Access / Collision Avoidance* (CSMA / CA) Verfahrens. Dabei werden Kollisionen durch das gleichzeitige Senden von mehreren Stationen dadurch vermieden, dass eine Station vor dem Sendevorgang auf dem Übertragungsmedium hört, ob dieses belegt ist. Ist dies der Fall wird eine *backoff-period* genannte Zeitperiode abgewartet, bevor ein neuer Übertragungsversuch stattfindet. Werden Beacons genutzt, wird die Variante *slotted CSMA/CA* verwendet. Dabei sind die backoff-periods auf die Slots zwischen den Beacons synchronisiert. Das CSMA/CA Verfahren wird nicht für das Senden der Beacons und die ACKs eingesetzt, da letztere unmittelbar nach Empfang einer Nachricht folgen und dies für Beacons nicht nötig ist, da sie in der Basisvariante nur vom Coordinator ausgesendet werden.

In der Contention-free Period soll es keine solchen konkurrierenden Übertragungen geben. Sie ist unterteilt in bis zu sieben *guaranteed time slots* (GTS), die auch mehrere der Superframe Slots belegen dürfen. Diese Zeitslots werden Geräten zugesichert, sodass diese entweder Daten empfangen oder senden können. Die Verwaltung der GTS übernimmt der PAN-Coordinator. Bei diesem müssen sich Geräte mit einer Anfrage für einen GTS anmelden. Der PAN-Coordinator vergibt daraufhin die Slots an die anfragenden Geräte auf einer *first-come-first-serve* Basis.

Neben diesem Slot basierten Verfahren gibt es auch den einfacheren non-beacon enabled Betriebsmodus. Hierbei nutzen die Stationen immer CSMA/CA, bevor sie auf dem Medium Daten übertragen. Hierbei wird die unslotted Variante der Kollisionsvermeidung verwendet, die mit normalen back-off Zeiten arbeitet und nicht auf Slotgrenzen synchronisiert. Der PAN-Coordinator übernimmt in diesem Modus also keine koordinierenden Aufgaben.

Bei beiden Verfahren (beacon und non-beacon enabled) gibt es grundsätzlich drei verschiedene Möglichkeiten des Datentransfers:

- Transfer zu einem Coordinator – In einem beacon-enabled PAN wartet das Gerät auf den Beacon und synchronisiert sich mit der Superframe Struktur, um anschließend seinen Frame zu senden. In einem non-beacon enabled PAN werden die Daten einfach direkt an den Coordinator gesendet, sobald der Kanals frei ist.
- Transfer von einem Coordinator – Ein beacon-enabled Coordinator überträgt Daten an ein Gerät, indem er im Beacon signalisiert, dass eine Nachricht wartet. Stellt ein Gerät dies fest, sendet es in der nächsten CAP eine Anfrage, woraufhin der Coordinator die Nachricht ausliefert. Befindet sich das PAN im non-beacon enabled Modus muss das Gerät selbstständig Anfragen stellen und bekommt die Verfügbarkeit einer Nachricht nicht explizit mitgeteilt.
- Transfer zwischen zwei Geräten (Peer-to-Peer) – Bei dieser Art des Transfers greifen die kommunizierenden Geräte auf das Medium zu, wenn es frei ist. Da es keine Koordinierung gibt, müssen beide Receiver durchgängig empfangen, um keine Übertragung zu verpassen. Eine energiesparende Synchronisation der Geräte auf eine gemeinsam ausgehandelte Kommunikationszeit ist hierbei nicht Teil des Standards und muss daher auf einem höheren Layer erfolgen.

Eine Bestätigung von Nachrichten (Acknowledgement) ist grundsätzlich optional. Ist eine Nachricht nicht mit einem Acknowledge Request (AR) markiert, sendet die Gegenseite beim Empfang der Nachricht keine Bestätigung aus. Der MAC Layer des sendenden Gerätes geht in diesem Fall davon aus, dass die Nachricht erfolgreich zugestellt wurde. In einem Szenario, bei dem das Fehlen von einigen Messwerten eines Sensors z.B. kein Problem darstellt, kann das Auslassen der ACKs zusätzlich Energie sparen. Ist das AR Feld gesetzt wird eine Bestätigung von der Gegenseite erwartet. Nach Empfang eines Frames wird nach einem festgelegten Pausenintervall ein ACK Frame geschickt. Das sendende Gerät wartet auf diese Bestätigung und prüft die angegebene Sequenznummer auf Korrektheit. Wird kein ACK nach einer festgelegten Zeitspanne empfangen, sendet das Gerät den Frame erneut, bis eine konfigurierbare Anzahl von Wiederholungen erreicht wurde. Ein Sonderfall tritt ein, wenn die Übertragung indirekt, also über einen Coordinator erfolgen soll. Dabei wiederholt der Coordinator im Fehlerfall die Aussendung des Frames nicht direkt, sondern wartet zuerst auf eine neue Datenanfrage des Gerätes.

Um ein PAN aufzubauen führen die Geräte eine Energy Detection und einen aktiven oder passiven Scan der verfügbaren Kanäle aus. Bei den aktiven und passiven Scans wird auf bereits vorhandene Beacon Aussendungen gelauscht. Beim aktiven Scan wird dazu

zusätzlich ein *beacon request* abgesetzt, der andere Coordinator eines non-beacon enabled PANs auf dem Kanal dazu auffordert einen Beacon auszusenden. Die RFDs haben diese Möglichkeit nicht. Sie nutzen die *orphan notification*, um den PAN Coordinator zu finden. Dabei wird nacheinander auf den zur Verfügung stehenden Kanälen diese Mitteilung abgesetzt, woraufhin der Coordinator die Quelladresse des Gerätes prüft. Wird diese gefunden, wird ein *coordinator realignment* durchgeführt. Dabei sendet der Coordinator die Konfigurationsparameter des PANs erneut aus und das RFD kann sich entsprechend einrichten und mit dem Netz assoziieren.

Beim typischen Start eines PANs beginnt der PAN-Coordinator mit dem oben beschriebenen Scan und wählt einen Kanal, eine PAN-ID, die noch nicht vergeben ist, sowie eine kurze MAC Adresse (MAC-Short-Address) von 2 Byte Länge. Danach werden dann mittels Aussenden des Coordinator Realignment Kommandos alle Parameter des PANs, wie z.B. die PAN-ID und die Kanalnummer bekannt gemacht. Anschließend beginnt der Coordinator mit der Aussendung von Beacons.

Geräte können sich nun mit dem PAN assoziieren, indem sie sich durch Senden eines *association request* an den PAN-Coordinator bei diesem anmelden. Sie bekommen bei Erfolg ebenfalls eine Short-Address zugewiesen. Diese wird bis auf einige Ausnahmen anschließend für die gesamte Kommunikation verwendet. Ansonsten verwendet das Gerät seine wesentlich längere EUI-64 basierte MAC Adresse (*macExtendedAddress*). Besonders ist die Broadcast Short-Address $0 \times \text{FFFF}$, die für Broadcasts, also das adressieren aller Teilnehmer verwendet wird. Dies ist z.B. bei der Verwaltung der GTS der Fall, da alle Geräte über die Vergabe der garantierten Zeitslots unterrichtet werden müssen, die diese nutzen wollen.

Die eigentliche Übertragung von Daten kann nun auf zwei Wegen erfolgen. Im einfachsten Fall ist diese direkt, also von Gerät zu Gerät. Dabei sendet ein Gerät einen Frame aus, den das empfangende Gerät anhand der Ziel Adresse erkennt und auswertet. Eine weitere Möglichkeit und für RFDs die Hauptmethode der Kommunikation ist die indirekte Übertragung von Daten. Ein RFD schläft aus Effizienzgründen die meiste Zeit. Der Coordinator gibt in den Beacon Frames an, ob und für welche Adressen Nachrichten vorliegen, die in der nachfolgenden CAP abgeholt werden können.

Der MAC Layer erfüllt zu höheren Schichten auch eine Filteraufgabe. Der Transceiver des PHY Layers empfängt und meldet alle PPDU's an den MAC Layer, die er physisch empfangen kann, also von Geräten ausgesendet wurden, die dem Standard entsprechen oder durch Störungen beeinflusst wurden. Daher werden zuerst alle Frames verworfen, deren Prüfsumme in der Frame Check Sequence (FCS) nicht zu den Daten des restlichen Frames passt. Weitere Filterstufen hängen davon ab, ob der MAC Layer im *promiscuous mode* konfiguriert ist. Ist dies der Fall, werden alle Nachrichten an den nächst höheren

Layer weitergereicht. Falls nicht, folgen weitere Filterstufen, wie z.B. die Prüfung der Übereinstimmung der Ziel PAN ID oder der Ziel Adresse mit der eigenen Geräteadresse.

Ein weiteres optionales Feature des MAC Layers ist Security. Dabei geht darum die Vertraulichkeit und Authentizität der Nachrichten sicherzustellen, sowie einen Schutz vor Replay Attacks zu liefern. Replay Angriffe zielen darauf ab, legitime Nachrichten aufzuzeichnen und nochmals abzuspielen, um ein System zu beeinflussen. Beispielsweise sei hier die Möglichkeit des Aufzeichnens und wieder Abspielens der Öffnungskommandos eines Funkschlüssels für ein Fahrzeug genannt. Der Schutz vor solchen Angriffen in 802.15.4 wird durch die Speicherung eines verschlüsselten Frame Zählers erreicht. Verschiedene Securitylevel sind konfigurierbar, die einen Einfluss z.B. auf den verwendeten Algorithmus haben. Die Verschlüsselung erfolgt generell symmetrisch. Das bedeutet, dass auf allen Geräten das Schlüsselmaterial verteilt werden muss. Ist dies nicht gewünscht oder möglich, kann eine Verschlüsselung auch auf höheren Schichten erfolgen.

Zusammenfassend zeigt Abbildung 3.5 eine allgemeine Form der verschiedenen möglichen MAC Framestrukturen für allgemeine MAC Frames. Daneben gibt es verschiedene Sonderformen, z.B. für ACKs. Wie in der oberen Zeile zu sehen, kann ein solcher Frame je nach Konfiguration viele Informationen weglassen oder verkürzt darstellen. Im Header enthält das Feld *Frame Control* Informationen über den Aufbau des Frames. Die weiteren Felder enthalten die in der Adressierung bereits behandelten Variablen.

Octets: 1/2	0/1	0/2	0/2/8	0/2	0/2/8	variable	variable		variable	2/4
Frame Control	Sequence Number	Destination PAN ID	Destination Address	Source PAN ID	Source Address	Auxiliary Security Header	IE		Frame Payload	FCS
		Addressing fields					Header IEs	Payload IEs		
MHR							MAC Payload		MFR	

ABBILDUNG 3.5: Allgemeines Format eines 802.15.4 MAC Frames [IEE16, S. 151]

Es folgen die *Information Elements* (IEs). Diese gehören bereits zu den Erweiterungen des 802.15.4e und werden in 3.3 auf Seite 46 genauer erläutert.

Der Footer schließt den Frame mit einer Frame Check Sequence (FCS) ab, die einen CRC Wert enthält, der von der MAC Schicht bei Empfang geprüft wird.

Dieser Abschnitt gibt nur einen groben Überblick über den gesamten Standard. Viele Details, die auch durch die zahlreichen Erweiterungen Einzug gehalten haben, überschreiten

den Umfang dieses Überblicks und sind daher nicht beschrieben worden. Die Erweiterungen 802.15.4a und e werden jedoch in [3.3 auf Seite 46](#) und [3.2.4 auf Seite 37](#) ausführlich behandelt.

3.2 Weiterentwicklungen auf Basis von 802.15.4

Vor allem im Anwendungsfeld der Industriesteuerungen haben sich einige weitere Standards etabliert, die als Weiterentwicklungen von 802.15.4 gelten. Sie sollen hier kurz mit ihren Besonderheiten vorgestellt werden. ISA100.11a und WirelessHART gelten hier als die Industriestandards, die die weiteren MAC Betriebsmodi in den Erweiterungen des Standards, insbesondere des 802.15.4e, stark inspiriert haben.

Zuerst folgt jedoch eine Betrachtung der Nachteile des MAC Layers des ursprünglichen Standards, die diese neuen Betriebsmodi überhaupt erst notwendig zu machen scheinen.

3.2.1 Nachteile des 802.15.4 MAC Layers

Kernprobleme von drahtlosen Übertragungsverfahren im Vergleich zu kabelgebundenen sind bei allen eingesetzten Funkstandards die Zuverlässigkeit der Übertragung, der erreichbare Datendurchsatz und die maximale Latenz einer Übertragung. Die Wahl des entsprechenden Physical Layers trägt viel zur Vermeidung dieser Probleme für das gewählte Anwendungsfeld bei. Der im Rahmen dieser Arbeit verwendete IR-UWB Layer bietet hier zusätzlich auch Lösungen für diverse Probleme, die auf der physischen Seite auftreten können, wie bspw. Multipath-Fading. Er ist in [3.3 auf Seite 46](#) genauer beschrieben. Generell besteht ein Unterschied zwischen Übertragungsfehlern, die durch den Physical Layer ausgelöst werden, und solchen, die auf MAC Ebene ihren Ursprung haben. Der PHY ist bspw. verantwortlich für nicht empfangene Pakete, die durch zu geringe SNR am Empfänger ausgelöst werden, also im einfachsten Fall durch zu große Entfernung der Knoten voneinander. Der MAC hingegen regelt den Zugriff der Knoten auf das Medium. Es können Kollisionen auftreten, die durch falsche Parameterwahl verursacht werden und dem MAC zuzuschreiben sind.

Der MAC Layer muss aber all diesen Problemen begegnen und z.B. erneute Übertragungen einleiten, falls Pakete verloren gehen. Besonders im 802.15.4 Standard ist dies von Bedeutung, denn es findet keine dynamische Anpassung der Modulationsparameter bei schlechteren Ausbreitungsbedingungen statt. Die Standards für WLAN sehen hier bspw. die Nutzung eines *Modulation and Coding Schemes* (MCS) vor. Dieses sorgt bei schlechterer Signal to Noise Ratio (SNR) dafür, dass die Modulation gewechselt wird, sodass

bei schlechten Empfangsbedingungen auf ein leichter zu demodulierendes Verfahren mit geringerem Durchsatz gewechselt wird. Für 802.15.4 ist dies nicht vorgesehen, da hier hauptsächlich in ihrer Leistung beschränkte Geräte eingesetzt werden. Eine solche Implementierung würde neben komplexerer Hardware den Verwaltungsaufwand erhöhen, was einen höheren Energieverbrauch zur Folge hat und nicht zuletzt einen größeren Code Footprint besitzt, der ein Gerät mit mehr Speicher voraussetzt.

Nicht zuletzt aufgrund dieser nötigen Einfachheit spielt der MAC Layer in drahtlosen Sensornetzen eine besondere Rolle bei der Sicherstellung der Zuverlässigkeit seiner Übertragungen. Exemplarisch für die vielen Untersuchungen zur Leistungsfähigkeit sei hier die Arbeit [ACF11] genannt. Die Autoren haben hier eine ausführliche Analyse der Zuverlässigkeit von Übertragungen in 802.15.4 Netzen vorgelegt. Sie stellen dabei in der Standardkonfiguration eine sehr schlechte *Packet Delivery Ratio* (PDI) fest. Es wird in den Tests also nur ein kleiner Prozentsatz der gesendeten Pakete auch tatsächlich vom Ziel empfangen. Dabei fällt die PDI bei größerer Anzahl der Knoten schnell unter 40 % bei lediglich 15 Knoten im PAN (siehe Abb. 3.6 auf der nächsten Seite). Als Hauptgrund für diese schlechten Werte wird der CSMA/CA Algorithmus ausgemacht, wie er in 3.1.3 beschrieben ist. Die gewählten Standardwerte für die Parametrisierung des Verfahrens sorgen dafür, dass die CAP, in der alle Knoten um die Nutzung des Übertragungsmediums kämpfen nicht optimal genutzt wird. Die Delivery Ratio lässt sich jedoch in nahezu allen Fällen auf 100 % erhöhen, wenn diese Parameter angepasst werden. Dies hat jedoch stark negative Auswirkungen auf die Latenz. Dies deckt sich auch mit den Untersuchungen in [Pol08], in denen mittels eines analytischen Modells und einer Simulation die Ergebnisse zu diesem spezifischen CSMA/CA Verfahren bereits theoretisch ermittelt worden sind.

Der Durchsatz als weiterer wichtiger Faktor ist durch die Physical Layer fest vorgegeben und nicht wie bereits ausgeführt dynamisch anpassbar. Die Packet Delivery Ratio ist also ausschlaggebend für die Nettodatenrate. Damit ist eine höhere Zuverlässigkeit gleichbedeutend mit höherem Datendurchsatz. Die Anpassung der CSMA/CA Parameter enthält z.B. neben der Konfiguration der Backoff-Period, die die Länge und Anzahl der Wartefenster vor erneuten Sendeversuchen bestimmt, auch die maximale Anzahl der Wiederholungen von Zustellversuchen. Letztere haben einen signifikanten Einfluss auf die Nettodatenrate, womit ein Netz mit hoher zeitlicher Auslastung des Mediums durch bspw. eine große Anzahl von Knoten trotz optimierter MAC Parameter einen geringeren Durchsatz hat als ein entsprechend kleineres Netz.

Die Latenz als letzter der genannten Faktoren spielt je nach Anwendung eine ebenso wichtige Rolle. Grundsätzlich ist die Latenz durch Verwendung des CSMA/CA Verfahrens nicht beschränkt. Die Autoren der Analyse konnten durch ihre Modifikationen der CSMA/CA Parameter, die im optimalen Fall nicht mehr standardkonform war, zwar eine

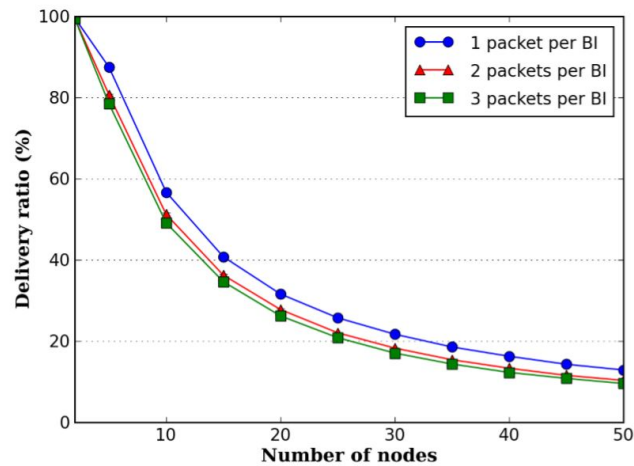


ABBILDUNG 3.6: Einfluss der Paketrate auf die Zuverlässigkeit des klassischen 802.15.4 MAC Layers ([ACF11, vgl. S. 11])

höhere Zuverlässigkeit und einen größeren Durchsatz erzielen, jedoch nur auf Kosten einer stark gesteigerten mittleren Latenz der Übertragung. In Industrieanwendungen kann dies für die Regelung von Anlagen ein Problem sein, denn es wird neben hoher Zuverlässigkeit gleichzeitig eine geringe Latenz benötigt.

Mit den klassischen MAC Layern ist dieses Problem nicht anders zu lösen. Die in den nächsten Abschnitten vorgestellten Erweiterungen versuchen diesen Problemen mit anderen MAC sowie PHY Layern zu begegnen.

3.2.2 WirelessHART

WirelessHART ist eines der Verfahren, das den 802.15.4e inspiriert hat. Es wird als die drahtlose Variante des *HART* (Highway Addressable Remote Transducer) Standards beworben, einem der weltweit am häufigsten eingesetzten Kommunikationsprotokolle für Industrieanwendungen. HART wird zum Aufbau von Feldbussen auf Stromschleifenbasis in der Anlagensteuerung eingesetzt. Definiert sind auch die oberen Schichten des OSI Modells. Neben spezifischen Schichten für die physische Übertragung von Signalen über Stromschleifen im 4 mA / 20 mA Standard und einen MAC, der den rohen Bitstrom versendet gibt es noch eine Transport- und eine Applikationsschicht. Die Transportschicht bietet dabei eine verbindungslose, aber zuverlässige Art der Kommunikation für die Applikationsschicht, die auf optionalen ACKs aufbaut. Die Applikationsschicht stellt das eigentliche HART Protokoll zur Verfügung. Pakete dieses Protokolls sind auf die Anlagensteuerung ausgelegt. Es finden sich hier u.A. Felder für das auszuführende Kommando, ein Datenfeld und Adressierungsdaten.

WirelessHART Geräte können durch dieses Anwendungsprotokoll einfach in bestehende HART Infrastrukturen integriert werden und umgekehrt. Auch Kabelgebundene HART Geräte können so über einen Drahtlosadapter für ein WirelessHART Netzwerk konvertiert werden. Die Bitübertragungsschicht ist aus dem 802.15.4 entnommen und festgelegt auf den klassischen 2,4 GHz PHY mit DSSS in der Offset-QPSK Modulation, wie sie schon in der ersten Version des Standards definiert wurde.

Der MAC Layer weist dagegen einige Besonderheiten auf, die ihn vom ursprünglichen IEEE Standard abheben. Wichtig ist, dass die MAC Services, die von diesem angeboten werden weiterhin kompatibel zu 802.15.4 sind. Die Schichten, die innerhalb dieser Standards definiert werden erhalten auch standardisierte Schnittstellen (Services genannt) nach außen, sodass andere Schichten, die sie nutzen leicht austauschbar sind. Die Verfahrensweisen innerhalb des WirelessHART MACs weichen jedoch ab. Es werden Time-Slots definiert, die auf fixe 10 ms festgelegt sind. Diese werden in einem TDMA Verfahren verwendet, um den Zugriff auf das Medium zu steuern. Jedem Gerät wird ein entsprechender Slot zugewiesen, den es für Datenübertragungen nutzen kann. Zusätzlich wird ein Frequency Hopping Verfahren auf den 16 verfügbaren Kanälen angewendet, um Störungen zu minimieren. Gibt es konstant Interferenzen auf einem bestimmten Kanal, bietet das System die Möglichkeit, diese zu blacklisten, womit sie aus der Liste der Frequenzsprünge ausgeschlossen werden. Jedes Gerät verwaltet hierzu eine Tabelle der aktiven Kanäle. [NR12, vgl. S. 8 f.]

WirelessHART besitzt im Vergleich zum kabelgebundenen HART zusätzlich noch einen Network Layer, der die Möglichkeit zum Betrieb von Mesh Netzwerken zur Verfügung stellt. Jeder Knoten hält dafür eine Liste der Verbindungen zu seinen Nachbarn vor. Die Netzwerkschicht kümmert sich dann um das Routing der Pakete innerhalb dieses Graphen. Zusätzlich ist auf dieser Schicht noch die Sicherung der Datenintegrität und der Vertraulichkeit implementiert, sowie ein Verfahren, dass Knoten bestimmte Zeiten für die Übertragung von größeren Datenmengen zuweisen kann.

WirelessHART ist durch integrierte Netzwerk und Anwendungsschicht ausgelegt auf Endanwender und wird von einigen Firmen in fertig zu verwendenden Produkten zur Integration angeboten. Im Gegensatz zu ISA100.11a ist es weitaus weniger konfigurierbar. So gibt es in WirelessHART bspw. nur einen Algorithmus zum Channel Hopping, um die Interoperabilität zwischen Geräten zu vereinfachen. Gleiches gilt für die fixen Zeitslots von 10 ms. Es bietet keine IP Adressen in der Netzwerkschicht. Trotzdem ist es im Gegensatz zu ZigBee in den Bereichen Zuverlässigkeit und Koexistenz mit anderen Funkdiensten im 2,4 GHz Band leistungsfähiger und damit für industrielle Anwendungen wesentlich besser geeignet [LSH08].

3.2.3 ISA100.11a

Ein weiterer Protokoll Stack, der bei den Weiterentwicklungen des IEEE Standards für die Verbesserungen herangezogen wurde ist ISA100.11a der International Society of Automation. Ziele bei der Entwicklung dieses Standards waren hohe Flexibilität durch Unterstützung verschiedener Protokolle, die Nutzung offener Standards und wie bei WirelessHART auch eine hohe Zuverlässigkeit, deterministische Zustellung von Daten durch QoS und Sicherheit der übertragenen Daten.

Auch ISA100.11a definiert einen Protokollstack, ähnlich wie WirelessHART oder auch ZigBee. Es stellt jedoch keinen Application Layer für die Prozessautomatisierung zur Verfügung, wie dies bei WirelessHART der Fall ist. Unterstützte Netzwerktopologien sind Stern, Mesh oder eine Kombination aus beiden. Eine Besonderheit ist das Routing, das durch 6LoWPAN über den Data Link Layer stattfinden kann, wenn innerhalb eines Meshes geroutet wird (mesh-under) oder über Backbone Router zwischen einzelnen Meshes auch IP basiert ablaufen kann (route-over).

Der Physical Layer in einem ISA100.11a Gerät basiert ebenfalls auf 802.15.4 DSSS mit O-QPSK und operiert auch im überfüllten 2,4 GHz Band, was weitere Anpassungen zur Erreichung der geforderten Robustheit nötig macht.

Der Data Link Layer, der diese Aufgabe übernehmen muss ist eine Abwandlung des 802.15.4 MACs und nicht kompatibel zu diesem. Er wurde erweitert, um link-local Adressierung zu ermöglichen, sowie die Nachrichtenweiterleitung im Mesh Routing sicherzustellen. Wie bei WirelessHART enthält dieser auch einen Mechanismus zum Channel Hopping und braucht daher Möglichkeiten zur Synchronisation. Auch hier werden Nachrichten in Time Slots von typischerweise 10 ms Länge kommuniziert, wobei dieses Intervall konfigurierbar bleibt. Beide Verfahren sorgen für eine stark gesteigerte Robustheit und Koexistenz mit anderen Funkdiensten im 2,4 GHz Band. In dieser Schicht wird ein Graph Routing System implementiert, das für verschiedene Klassen von Datenverkehr andere Routen wählen kann. Diese müssen vom System Manager, einem Netzverwaltungsgerät, das im Standard nicht näher spezifiziert ist, definiert werden und werden als sog. *Contracts* an die Graphen vergeben. Die Geräte nutzen die IDs dieser Contracts, um im Netzwerk bekannt zu machen, um welche Art Verkehr es sich bei ihren Daten handelt. Jedes Gerät wählt den nächsten Hop dann basierend auf dieser ID. Der System Manager nutzt zur Bestimmung der Contract Zuweisung verschiedene Eingabedaten, wie die Anforderungen an Latenz und Durchsatz von bestimmten Daten, sowie die Leistungsfähigkeit bestimmter Routen zwischen den Knoten im Netz. [NR12, vgl. S. 13 f.]

Die Netzwerkschicht nutzt 6LoWPAN als Adaptierung, um IPv6 für die Ende-zu-Ende Kommunikation verwenden zu können. Die IPv6 Pakete werden an den Routern, die die

Grenze zum ISA100.11a Netz markieren über 6LoWPAN konvertiert und den Geräten des Sensornetzes zur Verfügung gestellt. Das Routing zum Zielgerät wird dann anschließend über den Mesh Routingmechanismus und die Informationen im Data Link Layer Header der Geräte vorgenommen (mesh-under). Die übergeordnete Transportschicht stellt eine verbindungslose Kommunikation über UDP mit Ende-zu-Ende Verschlüsselung und Integritätsprüfungen zur Verfügung.

ISA100.11a ist wesentlich flexibler einsetzbar als WirelessHART. Es wird sich nicht auf einen beschränkten Funktionsumfang geeinigt, um möglichst einfache Integration verschiedenster Hersteller zu ermöglichen. Eine Ursache hierfür ist die erzwungene Kompatibilität von WirelessHART mit dem HART Anwendungsprotokoll. Diese Einschränkungen muss der ISA Standard nicht machen und kann daher die gesamte Breite der vorhandenen Standards wie IPv6 und 6LoWPAN nutzen. Zusätzlich bietet er ein flexibleres Routing Konzept mit den Backbone Routern, die mehrere Subnetze trennen können. Das Channel Hopping bietet verschiedene, konfigurierbare Algorithmen und die Time Slots sind in ihrer Länge variabel. Außerdem erlaubt der Standard *Over the Air Provisioning* (OTAP), um Geräte verschlüsselt und standardisiert mit Firmware Updates versorgen zu können. ISA100.11.a lässt sich damit sehr viel besser an ein gegebenes Einsatzszenario anpassen als WirelessHART. Ein Nachteil ist der Ressourcenverbrauch eines solchen Netzes. Insbesondere der Network Manager muss komplexe Berechnungen durchführen. Die Initialisierung eines Netzes dauert generell länger als im 802.15.4. [CCS13, S. C-3]

3.2.4 802.15.4e

Die benannten Probleme von 802.15.4 in [3.2.1 auf Seite 32](#) und die Erweiterungen, von denen einige in den vorherigen Abschnitten erläutert wurden und die auf dem Standard aufbauen, mündeten in die Entwicklung der 802.15.4e Erweiterung, die 2012 abgeschlossen wurde. Sie führt insbesondere neue MAC Betriebsmodi ein, die in den folgenden Abschnitten genauer erläutert werden.

Daneben gibt es aber auch einige allgemeine Merkmale, die dem Standard hinzugefügt wurden. So wurden weitere Anstrengungen unternommen, um die Energieeffizienz zu erhöhen. Zwei Verfahren sind hierzu hinzugefügt worden. Das Coordinated Sampled Listening (CSL) ist ein Modus, in dem der Coordinator eine Aufwachsequenz bestimmter Länge sendet, wenn Daten übertragen werden sollen. Der Empfänger ist nur periodisch für eine sehr kurze Zeit aktiviert und hört, ob diese Sequenz auf dem Kanal anliegt, ansonsten wird der Empfänger deaktiviert. Gleiches gilt für den Fall, dass die Zieladresse nicht zum Gerät passt. In diesem Fall würde die Wartezeit bis zum erneuten Aufwachen um die Übertragungszeit der Nutzlast verlängert werden, was weitere Energie spart

[IEE16, S. 135 ff.]. Der zweite Mechanismus sind Receiver Initiated Transmissions (RIT) für PANs, die keinen Beacon verwenden. Das Verfahren ist ähnlich zu CSL mit dem Unterschied, dass aufgrund der fehlenden Slots hier der Empfänger periodisch aktiv seine aktiven Hörperioden aussendet, woraufhin ein Sender diese nutzen kann, um eine Übertragung zu initiieren [IEE16, S. 139].

Die neu eingeführten Information Elements (IE) sind vordefinierte Datencontainer, die verschiedene Aufgaben innerhalb des MAC für die weiteren Betriebsmodi übernehmen. Sie enthalten IDs und eine eigene Payload zu diesen IDs, die in verschiedene Felder eingeteilt ist. Sie können auch in sog. Enhanced ACK Frames enthalten sein und zusätzliche Informationen zum ACK liefern. Bspw. finden IEs in Enhanced ACKs im Time Slotted Channel Hopping Verfahren Verwendung. Hierbei enthalten die IEs im ACK Informationen zur Zeitsynchronisation, die für die Kanalwechsel in diesem Betriebsmodus nötig sind. Auch im DSME Betrieb spielen sie eine wichtige Rolle. Zugehörig sind die neuen *Enhanced Beacons* (EB), die es erlauben die IEs aufzunehmen. Auch die normalen Frames sind mit dem *Multipurpose Frame* so erweitert worden, dass sie IEs enthalten können und so die neuen MAC Modi erlauben.

Zur Erweiterung gehört auch die Einführung einer optionalen Performance Metrik, mit der der MAC Informationen über den Datenverkehr sammeln kann. Hierzu gehören bspw. die Anzahl der erneuten Zustellversuche von Paketen, Anzahl doppelter Frames oder Fehler in der Prüfsumme der FCS. Höhere Layer können diese Daten benutzen, um sich den Übertragungsbedingungen anzupassen. Z.B. können Pakete fragmentiert werden, wenn die Metriken erkennen lassen, dass der Übertragungsweg stark unzuverlässig ist.

Als letztes der allgemeinen Verbesserungsmerkmale ist ein *Fast Association Mechanismus* hinzugefügt worden. Während die normale Assoziation von Knoten im 802.15.4 PAN erfordert, dass zuerst ein Channel Scan durchgeführt werden muss und nach der Anfrage ein zusätzlicher Request gestellt wird, um das Ergebnis zu erhalten, entfällt dies bei Fast Association. Stattdessen wird die Antwort direkt an das anfragende Gerät gesendet und ein Scan entfällt. Dazu muss der Kanal und die PAN ID im Vorfeld bekannt sein. Auch dieser Mechanismus dient der weiteren Einsparung von Energie.

Das Kernstück der Erweiterung sind verschiedene neue MAC Betriebsmodi:

- Time Slotted Channel Hopping (TSCH) – In diesem Modus wird ähnlich wie in WirelessHART oder ISA100.11a ein TDMA Ansatz mit Channel Hopping verfolgt, was zu höherer Robustheit in der Industrieanwendung führen soll.
- Deterministic and Synchronous Multi-channel Extension (DSME) – Soll sowohl niedrige Latenz als auch Zuverlässigkeit bieten. Es kann sowohl in TDMA, wie auch

im einfacher zu realisierenden CSMA Betrieb genutzt werden und bietet ebenfalls Channel Hopping. Außerdem ist es auf den Betrieb als Mesh Netzwerk ausgelegt.

- Low Latency Deterministic Network (LLDN) – Diese Betriebsart ist auf eine Stern Topologie mit nur einem Hop ausgelegt. Ziel sind niedrige Latenzen, wie sie in der Regelungstechnik benötigt werden.
- Asynchronous Multi-Channel Adaptation (AMCA) – Hier werden Netze aus sehr vielen Knoten gebildet, wie sie in Smart Utility Networks verwendet werden. Aufgrund der sehr variablen Ausbreitungsbedingungen, die Knoten im Netz erfahren können, wählt jeder Knoten die für ihn beste Frequenz zum Horchen und wechselt asynchron zu dieser, sobald Daten übertragen werden müssen
- Radio Frequency Identification Blink (BLINK) – Ein einfacher Modus, bei dem ein Knoten seine ID an andere übertragen kann, für Anwendungen wie sie z.B. in der Logistik benötigt werden. In einem solchen Netz gibt es keine Assoziierung mit einem PAN Controller.

Die letzten beiden werden im Standard nur kurz erwähnt und finden sich aktuell in der weiteren Literatur noch nicht wieder. Die weitaus häufiger untersuchten Erweiterungen sind TSCH, DSME und LLDN, die auch im Folgenden genauer betrachtet werden. Darüber hinaus stellen sie für die Verwendung mit UWB die vielversprechendsten Kandidaten dar.

3.2.4.1 Time Slotted Channel Hopping (TSCH)

TSCH nutzt ein TDMA Verfahren, bei dem die verfügbare Zeit auf dem Medium in Zeitschlitze aufgeteilt wird. Das Verwenden von Zeitschlitzen minimiert potentielle Kollisionen, wie sie im CSMA/CA Betrieb vorkommen. Dadurch wird der Durchsatz gesteigert und das Problem der quasi unbegrenzten Latenz im klassischen Betriebsmodus eliminiert. Kombiniert wird dies mit einem Mehrkanalbetrieb. Mehrere Knoten können dabei gleichzeitig den gleichen Zeitschlitz nutzen, wenn sie auf unterschiedlichen Kanälen senden. Zusätzlich wird nach einem bestimmten Muster über die Kanäle gesprungen (Channel Hopping). Ist ein Kanal durch eine andere schmalbandige Übertragung gestört, kann durch das regelmäßige Wechseln der Kanäle die Auswirkung dieser Störung minimiert werden. [DBA16]

Ein PAN in diesem Betriebsmodus wird durch die bereits erwähnten Enhanced Beacons konfiguriert. Die Beacons enthalten Informationen (IEs) über das Channel Hopping und die Timeslots. Das Vorhandensein dieser IEs markiert, dass das PAN im TSCH Modus operiert. Auch normale Knoten, die mit dem Netz assoziiert sind können Enhanced

Beacons periodisch oder nach einem *Enhanced Beacon Request* aussenden. Zusätzliche Filterkriterien in einer IE sorgen dafür, dass z.B. die minimale RSSI für das Senden festgelegt wird. So wird sichergestellt, dass nicht alle Knoten, die den Request empfangen einen Beacon aussenden, sondern nur Knoten, die sich in der Nähe des anfragenden Knotens befinden [IEE16, vgl. S. 191]. Die möglichen Zeitschlitz sind in einem *Slotframe* zusammengefasst (siehe Abb. 3.7). Sobald ein Gerät über ein EB die Konfiguration des Slotframes erhalten hat, kann es am Netzwerkbetrieb teilnehmen. Das bedeutet auch, dass dann keine Beacons mehr benötigt werden, um die Knoten zu synchronisieren und um die klassische CAP und CFP einzuleiten. Stattdessen senden die Knoten in den Zeitschlitz auf den verschiedenen Kanälen. Auch hier können Kollisionen auftreten, die dann per CSMA/CA behandelt werden, aber durch die Architektur weitaus seltener vorkommen, als dies in einer normalen CAP der Fall wäre.

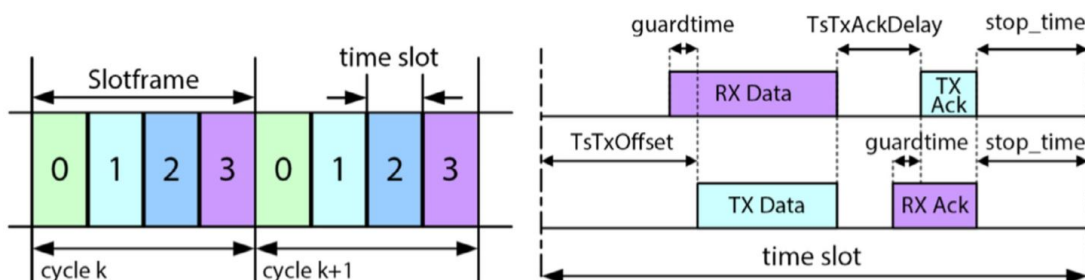


ABBILDUNG 3.7: TSCH Slotframe Struktur²

Die Kanäle (üblicherweise 16) werden durch das Channel Hopping für jeden Timeslot gewechselt. Um zu verhindern, dass ein bestimmter Timeslot ständig den gleichen Kanal bekommt wird die *Absolute Slot Number* (ASN) verwendet, die die Slots seit Formung des Netzes zählt und modulo der Kanalanzahl verrechnet. Die Kombination aus einem Kanal und einem Timeslot innerhalb des Slotframes heißt *Link*. Jeder Verbindung zwischen zwei Knoten ist ein solcher Link zugewiesen. Die Verteilung eines Link Plans, bei dem den Knoten die entsprechenden Links zugewiesen werden ist jedoch nicht Teil des Standards. Sie muss auf höheren Schichten erfolgen und kann z.B. von zentraler Stelle durch den PAN Coordinator durchgeführt werden. Ziel muss dabei sein, die verfügbaren Links möglichst gleichmäßig zu verteilen. Wenn es sich nicht vermeiden lässt, kann ein Link auch geteilt werden. Dabei können wieder Kollisionen auftreten, auch wenn sie weniger häufig vorkommen, weshalb generell weiterhin CSMA/CA mit entsprechenden Retransmits verwendet werden muss.

Innerhalb eines Timeslots können Daten zwischen zwei Parteien gesendet und empfangen werden. Nach dem $TsTxOffset$ (siehe Abb. 3.7) wird die Übertragung gestartet.

²[DBA16, vgl. S. 5]

Das Empfangsfenster nutzt eine zusätzliche Pufferzeit (*guardtime*), um leichten Desynchronisierungen entgegenzuwirken. Diese können auftreten, da die Uhren der Geräte durch verschiedene Einflüsse wie Temperatur, Schwankungen der Spannung, etc. über die Zeit driften. Um trotzdem einen sicheren Empfang zu gewährleisten, wird diese Schutzzeit verwendet. Um zu verhindern, dass nach langer Laufzeit die Uhren von kommunizierenden Knoten so weit auseinander gedriftet sind, dass sie nicht mehr in die Pufferzeit passen, muss ein Gerät seine Uhr regelmäßig synchronisieren. Dazu wird eine *Time Correction IE* verwendet. Diese wird mit den verwendeten *Enhanced ACKs*, die in diesem Betriebsmodus versendet werden mit ausgeliefert oder über den Start des Frames ermittelt. Da der *TsTXOffset* bekannt ist, kann so sehr einfach die aktuelle Abweichung bestimmt werden. Der Knoten, der die Zeit zur Verfügung stellt wird *time source neighbor* genannt. Idealerweise müssen alle diese Zeitquellen z.B. in einer Baumstruktur verbunden sein, um ein abdriften einzelner Netzsegmente aus der globalen Netzzeit zu verhindern. Auch hier macht der Standard keine Vorgaben zu der Art, wie ein solcher zeitgebender Nachbar bestimmt werden soll.

Die größten Nachteile von TSCH liegen im Bereich des Scheduling von Links und dem Routing in Mesh Netzwerken. Es wird vorausgesetzt, dass dies von höheren Layern übernommen wird. Gerade die Verteilung von Links in verteilten Mesh Netzen ist nicht trivial und bedarf einer Lösung auf niedrigeren Layern [DBA16, vgl. S.12]. Auch der Aufbau des Netzes ist ineffizient, da Geräte ihre Empfänger während des Beitritts ständig aktiviert haben müssen. Bei mobilen Geräten, die sich oft neu synchronisieren müssen, verbraucht dies viel Energie.

3.2.4.2 Deterministic and Synchronous Multi-channel Extension (DSME)

Der DSME Betriebsmodus ist auf größtmögliche Flexibilität ausgelegt. Auch bei variablen Verkehrsbedingungen soll er zuverlässig sein und ein deterministisches Latenzverhalten zeigen. Im Gegensatz zu TSCH werden hier wieder die klassischen Zugriffsperioden CAP und CFP verwendet. Bei DSME wird die CFP, in der die Knoten garantierte Zeitslots bekommen können, ausgedehnt. Zusätzlich werden auch hier wieder mehrere Kanäle gleichzeitig verwendet. Der Superframe, der CAP und CFP beinhaltet wird auf einen *Multi-superframe* ausgedehnt, in dem die beiden Perioden flexibel verteilt werden können. So kann es mehr CFPs als CAPs geben, wenn viel periodischer Datenverkehr anfällt (*CAP Reduction*).

DSME erlaubt es, dass Knoten in Punkt-zu-Punkt Manier miteinander kommunizieren können. Es ist außerdem darauf optimiert, Mesh Netzwerke zu bilden. Dazu können Knoten feste Links miteinander eingehen. So ist der Weg über mehrere Knoten festgelegt

und die Latenz deterministisch. Es gibt in einem solchen Netz keinen PAN Coordinator, stattdessen wird das Aussenden der Beacons und die Zuweisung der Slots für die GTS verteilt über alle Knoten erledigt. Weil jedes Knotenpaar mit einer Verbindung die Slots selbst verhandeln kann, passt sich das Netz äußerst schnell an wechselnde Anforderungen des Datendurchsatzes oder die geänderte Verteilung von Knoten im Netz an. Im Vergleich dazu müsste bei solchen Änderungen in einem TSCH Netz jedes Mal das Scheduling neu verhandelt werden, was z.B. bei zentraler Verwaltung in einem Mesh Netz nicht immer möglich sein kann und außerdem bei jeder Änderung sehr viel Datenverkehr als Overhead zur Verwaltung der Slots erzeugt. DSME bietet dagegen zusätzlich noch *Group Acknowledgements*, mit denen mehrere Datenpakete mithilfe eines einzelnen ACKs bestätigt werden können, was den Overhead weiter reduziert.

Der Multi-superframe wird durch Enhanced Beacons eingeleitet. Innerhalb des Beacon Intervalls gibt es dann mehrere Superframes aus CAP und CFP. Anders als in 802.15.4 gibt es keine Phasen, in denen das Netz nicht genutzt werden kann. Es folgt Multi-superframe auf Multi-superframe ohne Pause. Die CAP wird üblicherweise für Kontrollnachrichten und unvorhergesehene Daten verwendet. Da alle Knoten miteinander kommunizieren können müssen, müssen sie während der CAP alle aktiv sein. Die in DSME größere CFP wird genutzt, um periodische Daten mit deterministischer Latenz zu versenden. Um den höheren Energieverbrauch der CAP im Gesamtnetz zu reduzieren wurde ein *CAP Reduction* Modus eingeführt, in dem nur der erste Superframe des Multi-superframes eine CAP enthält. Diese Option, sowie die Anzahl und Dauer der Superframes sind statisch festgelegt im PAN Descriptor.

Bei der Synchronisation neuer Knoten im Netz hören diese auf den verschiedenen Kanälen auf die Enhanced Beacons, die alle Informationen im PAN Descriptor enthalten, um am Netz teilzunehmen. Hört ein Knoten ein solches Beacon bestimmt er den aussendenden Coordinator zu seinem Elternknoten für die Zeitsynchronisation. Für die Aussendung der EBs ist der erste Slot in der CAP Periode jedes Superframes bestimmt. Die Coordinator teilen sich die Aufgabe der Aussendung der Beacons, um Überschneidungen zu vermeiden, indem sie Slots unter sich aufteilen. Dazu wertet ein sich neu anmeldender Coordinator die Information zur Belegung der EB Slots aus den EBs aus, die er empfangen kann und wählt anschließend einen freien Slot für seine EB Aussendung.

Die Verteilung der GTS Zeitschlitz erfolgt in DSME dezentral. Ein Sender fragt dabei eine Reservierung für einen GTS an. Diese werden wie andere Netzwerk Management Kommandos auch in der CAP versendet. Der Empfänger kann den GTS daraufhin reservieren. Dazu hält jede Node eine Tabelle mit Informationen über belegte GTS der Nachbarknoten vor, anhand der eine Reservierung eines freien Slots durchgeführt werden kann. Während des Austauschs dieser Nachrichten überprüfen Nachbarknoten in

Reichweite, ob der zu reservierende GTS mit bereits bestehenden kollidiert und melden dies entsprechend oder aktualisieren im Gutfall ihre eigene Tabelle der Belegungen. Wird der Slot nicht mehr verwendet, kann er wieder abgemeldet werden.

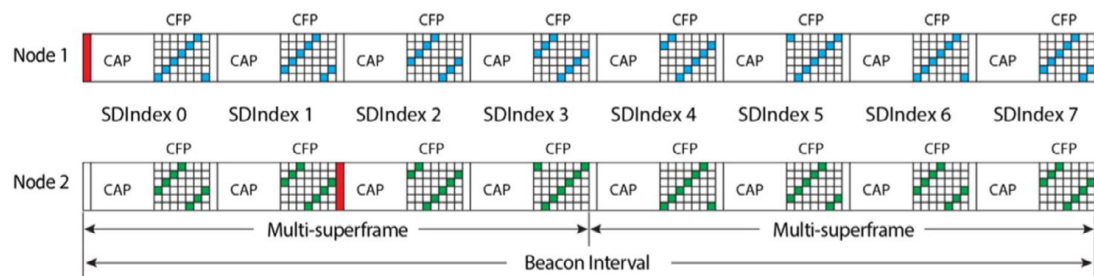


ABBILDUNG 3.8: Channel Hopping im DSME PAN³

Mehrere Kanäle können in DSME wie bei TSCH im Channel Hopping Betrieb oder in einer *Channel Adaptation* genutzt werden. Das Channel Hopping unterscheidet sich von TSCH insofern, als das bei DSME jeder Knoten einen fixen Channel Offset verwendet, wohingegen dieser in TSCH pro Link neu ausgehandelt wird. Ein Sender in DSME muss sich also dem Offset seines gewünschten Empfängers anpassen. Die Sequenz der Kanalwechsel ist aber für alle Knoten im Netz die gleiche. Bei der Channel Adaptation einigen sich zwei Knoten auf einen Kanal, über den sie kommunizieren wollen. Dieser wird auch während der GTS Anmeldung ausgewählt mit der Ausnahme, dass hier nicht nur der Zeitschlitz, sondern auch der Kanal gewählt und in der Allokationstabelle abgelegt wird.

DSME gilt sowohl als die komplexeste Erweiterung im aktualisierten Standard, aber auch als die flexibelste. Sie ist geeignet schwankenden Datenverkehr auch in großer Menge zu transportieren bei gleichzeitiger Garantie von deterministischer Latenz. Wohl aufgrund der Komplexität gibt es bis heute jedoch keine vollständige Implementierung dieses MAC Modus. Hinzu kommt, dass der Standard an vielen Stellen oft nicht klar ist oder Aufgaben wie die Details der GTS Allokation nicht spezifiziert sind [DBA16, vgl. S. 19].

3.2.4.3 Low Latency Deterministic Network (LLDN)

Der letzte Modus, der hier vorgestellt werden soll ist LLDN. Wie die anderen Modi auch wurde es für Anwendungen geschaffen, die ein robusteres Netz benötigen, als es 802.15.4 bieten konnte. Eines der Designziele bei LLDN war die Maßgabe, Daten von 20 verschiedenen Sensoren alle 10 ms versenden zu können. Um dies zu erreichen sieht der LLDN Modus keinen Betrieb als Mesh vor. Die Topologie ist strikt sternförmig mit einem PAN Coordinator als zentraler Anlaufstelle und Knoten, die ausschließlich mit ihm kommunizieren. Wie im klassischen 802.15.4 wird auch hier wieder nur ein Kanal verwendet,

³[DBA16, vgl. S. 15]

der zuvor konfiguriert wurde. Der PAN Coordinator gibt eine Konfiguration aus, die den Knoten feste Zeitschlitze für ihren Uplink zuweist.

Diese Vereinfachungen sind unter anderem Gründe für die geringe mögliche Latenz in einem LLDN Netz. Im Vergleich zu TSCH gibt es hier die zur Ermöglichung der Zeitsynchronisation für das Channel Hopping nötige Warteperiode zu Beginn eines Netzbeitritts nicht. Außerdem wird ein MAC Header mit nur einem Byte Größe verwendet, der für geringe Verarbeitungszeit auf den Netzknoten und damit wiederum zu geringerer Latenz führt. Möglich ist dies, da die Kommunikation nur mit dem PAN Coordinator stattfindet und dieser die Slots den einzelnen Knoten zugewiesen hat. Es ist daher keine Adressierung im Uplink mehr nötig.

Die Initialisierung eines LLDN verläuft in drei Phasen [IEE12a, vgl. S. 32-34]:

1. Discovery – Geräte, die dem Netz beitreten wollen, suchen auf den verschiedenen möglichen Kanälen bis sie einen Beacon eines Coordinators empfangen, der den Discovery Modus anzeigt. Das Gerät sendet daraufhin eine *Discovery Response*, um sich am Netz anzumelden. Nach einem Timeout ohne Empfang weiterer Discovery Responses wechselt der Coordinator in die Konfigurationsphase.
2. Configuration – Empfängt ein Gerät nun ein Beacon, in dem der Konfigurationszustand angezeigt wird, sendet es solange einen *Configuration Status Frame*, in dem es seine aktuelle Konfiguration bekannt gibt, bis es vom PAN Coordinator einen *Configuration Request* mit einer neuen Konfiguration bekommt, oder die Phase vom Coordinator beendet wird. In den Konfigurationsdaten befinden sich auch die Zeitschlitze, die das Gerät bei Kommunikation verwenden soll.
3. Online – Zum Schluss geht das Netz in den Online-Zustand über. Nun wird die im Folgenden erläuterte Superframe Struktur verwendet, um Daten auszutauschen.

Der Frame im Online-Zustand ist in mehrere Timeslots aufgeteilt (siehe Abb. 3.9 auf der nächsten Seite). Wie im klassischen PAN auch beginnt der Frame mit einem Beacon, der von den Knoten zur Synchronisation verwendet wird und in dem das Format des Superframes definiert ist.

Es folgen zwei Slots zur Übermittlung von Management Daten. Diese sind im Online Zustand optional und werden in Discovery und Configuration dazu genutzt, die Informationen zur Formung des Netzes auszutauschen. Die eigentliche Datenübertragung findet in den folgenden Uplink und Bidirectional Timeslots statt. Die Uplink Slots werden dabei ausschließlich zur Kommunikation von Knoten mit dem Coordinator genutzt. Da jeder

⁴[DBA16, vgl. S. 21]

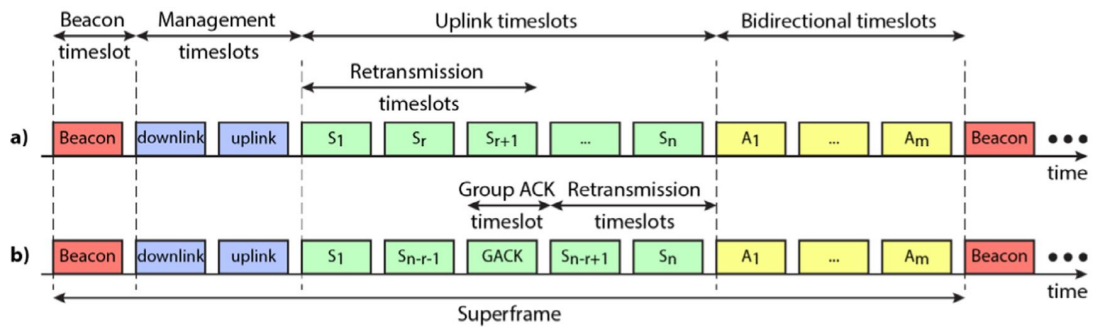


ABBILDUNG 3.9: Aufbau eines Superframes in einem LLDN mit a) und ohne b) Group ACK Slot⁴

Knoten einen spezifischen Slot zugewiesen bekommen hat, gibt es hier keine Kollisionen von Übertragungen. Anschließend ist eine bidirektionale Kommunikation von Knoten und Coordinator möglich. Dabei gibt ein Bit im Beacon vor dem Superframe an, welche Richtung für diesen Frame verwendet wird.

Ein Knoten kann lediglich einen Frame pro Zeitslot versenden. Die Länge der Slots richtet sich deshalb nach der Länge des längsten zu erwartenden Frames. So können die Slots eng an die zu erwartenden Frames angepasst werden, was eine minimale Latenz garantiert. Die ACKs zur Bestätigung sendet der Coordinator als *Group ACK* im Beacon des nächsten Superframes aus, das durch eine Bitmap im Beacon dargestellt wird. Dabei können die ersten Slots der Uplink Phase für die Retransmissions reserviert werden. Für die bidirektionale Phase wird dies ebenso verwendet. Ist die Richtung aber auf *Downlink* gesetzt, so bestätigen die Knoten den Empfang im nächsten Superframe im gleichen Slot, wozu dieser vom Coordinator auf Uplink gestellt werden muss.

Jeder LLDN Timeslot, der geteilt verwendet wird (*Shared Group Timeslot*), wie z.B. ein Management Timeslot, ist zusätzlich noch unterteilt, wie in (siehe Abb. 3.10 auf der nächsten Seite) zu sehen ist. Der Zeitraum zwischen t_0 und t_1 ist dabei exklusiv für einen Knoten reserviert, falls dieser ihn in der Konfigurationsphase vom Coordinator zugewiesen bekommen hat. Zwischen t_1 und t_2 gibt es eine Phase in der andere Knoten den Slot nutzen können, falls er nicht von der reservierenden Node verwendet wird. Dazu senden sie Anfragen an den Coordinator, der den Kanal für einen Knoten freigeben kann. Weil mehrere Knoten in dieser Zeit senden können, wird hier eine einfache Variante von CSMA/CA mit nur einer Backoff Phase verwendet. Der letzte Bereich zwischen t_2 und t_3 ist für den Coordinator reserviert. Die Zeiten sind konfigurierbar. So ist es auch möglich den Slot nur einem einzigen Knoten zuzuweisen, indem das Intervall t_0 - t_1 den gesamten Slot umfasst.

⁴[DBA16, vgl. S. 21]

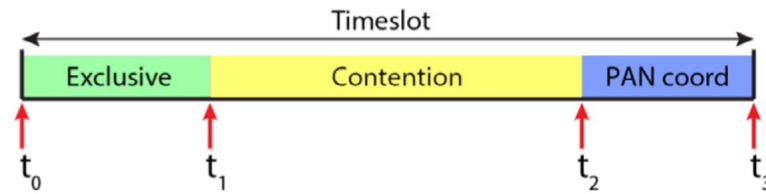


ABBILDUNG 3.10: Aufteilung eines LLDN Timeslots⁵

LLDN zeichnet sich durch seine einfachen, aber sehr zuverlässigen Superframes aus, die verzögerungsarme Verbindungen erlauben. Die single-hop Topologie auf einem einzigen Kanal sorgt dabei für deterministische Latenzen im Netzwerk. Gleichzeitig ist dieser robuste Aufbau für andere Anwendungen nachteilig. Bspw. bestimmt die Zahl der Timeslots die Anzahl der Knoten im Netz. Kommt ein neuer Knoten hinzu, muss das gesamte Netz in die Konfigurationsphase zurückspringen, was bei Anwendungen, die auf unterbrechungsfreie Kommunikation angewiesen sind, problematisch sein kann. Dadurch, dass nur ein einzelner Kanal mit hauptsächlich fest zugewiesenen Slots verwendet wird, kann ein solches Netz nicht gut auf sich ändernde Durchsatzanforderungen reagieren. Für Sensorik- und Regelungsaufgaben stellt die Verwendung von LLDN aber in dieser Hinsicht kein Problem dar.

3.3 Ultra Wideband

Der in dieser Arbeit eingesetzte Physical Layer ist ein impulsbasiertes Ultrabreitbandverfahren (UWB). Seine Vorteile liegen in der sehr geringen verwendeten Energie pro MHz und der relativen Störuneempfindlichkeit, da das verwendete Spektrum extrem breit ist. Die geringe Leistungsdichte führt auch dazu, dass eine UWB Übertragung typischerweise im Spektrum nicht nachweisbar ist und im Normalfall nur den Rauschpegel geringfügig erhöht.

Klassische Funkdienste setzen für die Übertragung von Informationen auf die Modulation einer Trägerwelle (siehe Abb. 3.11 auf der nächsten Seite). Die Bandbreite eines solchen modulierten Signals ist dabei vergleichsweise schmal und reicht von wenigen KHz bis zu einigen MHz. Physikalisch günstige Bereiche im Frequenzspektrum, für die sich einfache Transceiver bauen lassen sind heutzutage stark überfüllt. Daher beschränken Regulierungsbehörden die nutzbaren Bandbreiten zusätzlich, um die Koexistenz von Funkdiensten zu gewährleisten. Übliche Modulationen sind häufig Phasenumtastverfahren oder Quadraturamplitudenmodulation (QAM), die durch kleinere Phasen- und Amplitudensprünge eine höhere Symbolrate erreichen. Der ursprüngliche 802.15.4 Standard beschreibt ein BSPK und ein QPSK Verfahren.

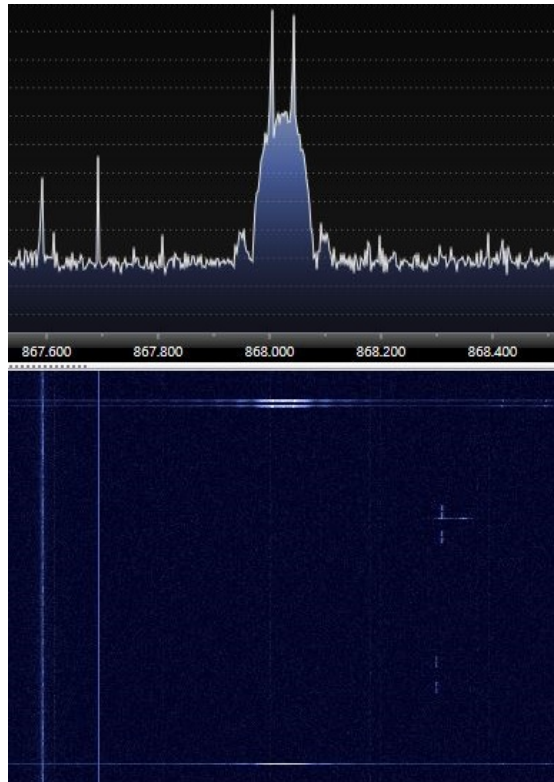


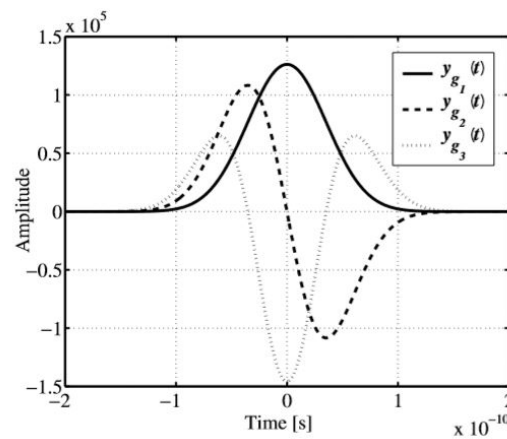
ABBILDUNG 3.11: Schmalbandige FSK Übertragung mit Trägerwelle auf 868 MHz. Der Träger ist nur sichtbar, wenn gesendet wird. In diesem Beispiel handelt es sich um UDP Pakete

Diese Art der Übertragung birgt Probleme. Da die Bandbreite, insbesondere bei den Modulationsverfahren, die in WSN eingesetzt werden, sehr schmal sind sind sie störanfällig. Hinzu kommt, dass die spezifizierten Frequenzbereiche für 802.15.4 Netze stark überlastet sind. Im 2,4 GHz Bereich findet sich bspw. auch Bluetooth und WLAN, die allgegenwärtig sind und die Kanäle eines WSN stören. Ein weiteres Problem, das häufig schmalbandige Übertragungen trifft ist Multipath-fading, bei dem die Funkwellen von der Umgebung reflektiert und zeitversetzt beim Empfänger ankommen, so dass sie das eigentliche Signal überlagern und so abschwächen. Dies beeinträchtigt nahezu alle Leistungsparameter in einem PAN, wie eine Modellierung in [Di 14] zeigt.

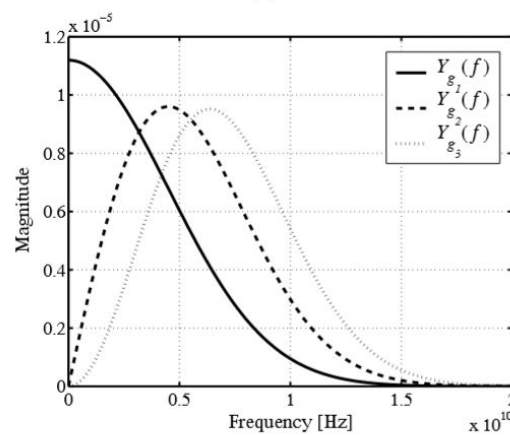
Hier kann ein UWB Physical Layer Abhilfe schaffen. Ein solcher ist bereits im 802.15.4a spezifiziert worden [IEE07]. Das Verfahren arbeitet impulsbasiert, wobei die Abfolge der Pulse die Information kodiert und die Dauer der einzelnen Pulse die Breite im Frequenzspektrum definiert (Leck-Effekt⁶). In einem UWB Transceiver werden zur Pulsformung gaußförmige Signale verwendet, da sie eine sehr gleichmäßige Verteilung über das Frequenzspektrum besitzen [MLR07, S. 31] (siehe Abb. 3.12 auf der nächsten Seite). Als Modulationsart wird BPSK in Kombination mit der *Burst Position Modulation*

⁶Wird ein Signal, das durch eine nicht kontinuierliche Sinusschwingung erzeugt wird Fourier-transformiert, ergibt sich ein Frequenzspektrum mit Anteilen, die nicht im Signal enthalten sind

(BPM)⁷ verwendet. In dieser wird Information durch variable Pausen zwischen den Pulsen übertragen. Insgesamt können so 2 Bit pro Symbol übertragen werden. Reguliert sind drei Frequenzbereiche bzw. Kanäle, in denen UWB betrieben werden darf. Sie liegen unter 1 GHz, zw. 3 – 5 GHz und zw. 6 – 10 GHz. Es sind jedoch mehr Kanäle definiert. Ein Kanal belegt dabei immer mindestens 500 MHz. Zusammen mit der *Pulse Repetition Frequency* (PRF), die die Grundwiederholrate der Pulse bestimmt, ergeben sich so verschiedene, mögliche Übertragungsraten von 110 Kb/s, 850 Kb/s, 6.8 Mb/s und 27 Mb/s. Die Aussendung erfolgt in Bursts zwischen denen ein Schutzintervall eingefügt ist.



(a)



(b)

ABBILDUNG 3.12: Frequenzspektrum von Ultrabreitbandübertragungen im Vergleich. Die Gaußwelle (fein gestrichelt) verteilt sich gleichmäßig über das Spektrum ([MLR07, S. 30])

UWB kann ebenfalls gut für Lokalisierung verwendet werden, denn die große Bandbreite erlaubt eine sehr genaue Laufzeitmessung des Signals, wodurch die Positionsauflösung besonders groß ist [Kar10, S. 50].

Auf der Physical Layer Seite findet die Synchronisierung in der PPDU über eine lange Präambel im Synchronization Header mit bis zu 4096 Symbolen statt. Nur wenn Teile

⁷auch *Pulse Position Modulation* (PPM) genannt

dieser empfangen werden, kann das restliche Paket überhaupt vom Empfänger gesehen werden. Der Header enthält im vgl. zu allen anderen PPDU's zusätzliche Information über die Datenrate, die für die Payload folgt und das Ranging.

Da kein Träger verwendet wird und das Vorhandensein einer UWB Übertragung nur durch feststellen der Präambel überhaupt möglich ist, ist ein Nachteil von UWB, dass kein CSMA/CA verwendet werden kann. Dies benötigt die Möglichkeit, einen Träger auf dem Übertragungskanal identifizieren zu können.

3.4 Kapitelzusammenfassung

Dieses Kapitel lieferte eine kurze Beschreibung des ursprünglichen 802.15.4 Standards, wie er 2003 veröffentlicht wurde und von vielen Produkten mittlerweile eingesetzt wird. Im weiteren Verlauf wurden die Probleme, die die Datenübertragung in einem solchen Netz teilweise massiv beeinträchtigen erörtert. Insbesondere für den Bereich der Industriesteuerungen oder dem für diese Arbeit wichtigen Einsatz in Raumfahrtanwendungen ist der klassische Standard daher ungeeignet. Probleme in der Ausbreitungscharakteristik der Funkwellen lassen sich auf physischer Ebene lösen und sind der Grund, warum im Rahmen der Arbeit die Ultrabreitbandtechnologie eingesetzt wird. Lösungen für die Probleme auf Seite des MAC Layers sind durch verschiedene Standards aus der Industrieautomatisierung hervorgegangen, von denen WirelessHART und ISA100.11a als prominenteste Vertreter gelten. Viele ihrer Erweiterungen des 802.15.4 mündeten in die MAC Erweiterungen des 802.15.4e Standards, dessen verschiedene MAC Betriebsmodi hier beschrieben wurden.

Die Umsetzung im folgenden Kapitel widmet sich der Auswahl und Implementierung eines dieser Verfahren in Kombination mit einem UWB Physical Layer. Dabei wird der zu implementierende Modus auf Basis der Anforderungen in Kapitel 2 auf Seite 14 ausgewählt.

Umsetzung

In diesem Kapitel wird die Umsetzung einer der beschriebenen MAC Layer Erweiterungen erläutert. Dazu wird im Abschnitt 4.1 zuerst die Auswahl von Technologien für die Implementierung vorgenommen und die dadurch beeinflusste Systemarchitektur entwickelt. Es folgt in Abschnitt 4.2 auf Seite 59 eine Beschreibung der verwendeten Hardware und ihrer Besonderheiten, auf der das zu implementierende System letztendlich ausgeführt werden soll. Zuletzt wird eine Beschreibung der Implementierung mit ihrer Einbettung in das ausgewählte Betriebssystem gegeben (siehe 4.3 auf Seite 62). Dabei wird auch das Softwaredesign des MAC Layers mit Bezug auf den Standard dargestellt.

4.1 Entwurf

Der hier beschriebene Entwurf besteht aus zwei Teilbereichen. Bevor mit der Entwicklung der Architektur in 4.1.3 auf Seite 56 begonnen werden kann erfolgt zunächst eine Technologieauswahl auf Basis der in Kapitel 2 auf Seite 14 ermittelten Anforderungen im nächsten Abschnitt.

4.1.1 Auswahl des MAC Layers

Für die Implementierung des MAC Layers muss eine geeignete 802.15.4 Erweiterung ausgewählt werden, die erwarten lässt, dass sie die gestellten Anforderungen an das System erfüllt. Aus den Einbettungsanforderungen (siehe 2.1.5 auf Seite 19) ergibt sich zusätzlich die Notwendigkeit zur Auswahl eines geeigneten embedded Betriebssystems, das idealerweise bereits eine Implementierung für einen OSI Stack mitbringt.

Die Anforderung REQ-14 (MAC Standard) besagt, dass eine MAC Erweiterung ausgewählt werden soll, die auf einem etablierten Standard basiert, um die Interoperabilität mit anderen Systemen zukünftig gewährleisten zu können. Hierzu bieten sich die in 3.2.4 auf Seite 37 beschriebenen Erweiterungen des 802.15.4e an. Auch das Green Book der CCSDS zum Einsatz drahtloser Sensornetzwerke auf Raumfahrzeugen empfiehlt die Verwendung dieses Standards [Spa17].

Einen der wichtigsten Faktoren stellt die Latenz dar. Insbesondere für Regelungsprozesse ist eine niedrige Latenz besonders wichtig, da entsprechende Aktoren je nach Anwendung schnell nachgeführt werden müssen, damit das System nicht außer Kontrolle gerät.

Auf Raumfahrzeugen ist dies z.B. in der kritischen Lageregelung der Fall. Ein weiterer wichtiger Aspekt ist eine deterministische Latenz, also eine Verzögerung, die bei gleichbleibenden Rahmenbedingungen des Netzes eine vorhersagbare Latenz garantiert, sodass Anwendungsprozesse diese in ihren Berechnungen kompensieren können. TSCH bietet diesen Aspekt z.B. nicht, da unter anderem durch Kollisionen im Link Plan generell CSMA/CA verwendet wird und bei Kollisionen so die Latenz nicht vorhersehbar sein kann.

Der Datendurchsatz ist bei Sensorwerten zur Regelung im Vergleich üblicherweise nicht besonders hoch und spielt daher eine nicht allzu große Rolle. Hinzu kommt die Tatsache, dass die verwendeten UWB Transceiver eine Bruttodatentransferrate von 6,5 Mbit/s im vgl. zu den für 802.15.4 PHYs üblichen 250 kbit/s bieten. Trotzdem hat der MAC Layer einen Einfluss auf den Overhead und damit auf die Ausnutzung dieser möglichen Übertragungsrate. Hier bieten LLDNs einen Vorteil durch ihre besonders sparsamen MAC Header. Es muss jedoch auch das Zusammenspiel in Netzen mit mehreren Knoten betrachtet werden. TSCH bietet durch das Channel Hopping weniger Möglichkeiten für Kollisionen und Störungen durch schlechte Ausbreitungsbedingungen und kann damit eine höhere effektive Übertragungsrate durch weniger nötige Retransmits erreichen. DSME bietet darüber hinaus die Nutzung von mehreren Kanälen gleichzeitig, sodass mehrere Knotenpaare zur selben Zeit Daten austauschen können ohne einen Linkplan berücksichtigen zu müssen. Dies bietet aus Sicht des gesamten Netzes die höchstmögliche Übertragungsrate.

Der MAC hat auch einen Einfluss auf den Energieverbrauch der Knoten des Netzes. Ein höherer Overhead durch zusätzliche IEs, wie sie zur Verwaltung der Kanalzuordnung in TSCH und DSME benutzt werden müssen erfordern zusätzliche Übertragungskapazität, wodurch der Transceiver länger aktiviert sein muss. Hier sind LLDNs aufgrund des Betriebs auf nur einem Kanal und der Sterntopologie im Vorteil.

Ein weiterer wichtiger Faktor, der berücksichtigt werden muss, ist die Komplexität einer Implementierung des MAC Layers. Die Verwendung möglichst energiesparender eingebetteter Systeme führt dazu, dass kleine Mikrocontroller mit limitierter Speicherausstattung genutzt werden. Um Strahlungseffekten in nicht gesondert gehärteter Elektronik besser begegnen zu können, kann außerdem auf größere Silizium Strukturbreiten gesetzt werden. Dies erhöht die mögliche Total Ionizing Dose (TID), die ein Bauteil vor seinem Ausfall aufnehmen kann [OM03, S. 494], verringert aber unter anderem auch die Menge an Speicher, die auf der gleichen Fläche untergebracht werden kann. Unter diesem Gesichtspunkt ist ein weniger komplexer MAC Layer wie LLDN zu bevorzugen. Auf die

größere Flexibilität, wie sie z.B. von DSME geboten wird kann insbesondere in Anbetracht der relativ starren Netzwerkkonfiguration an Bord des Raumfahrzeugs verzichtet werden.

Die Netzrekonfiguration und Rekonfigurationsdauer (nach REQ-08, REQ-09) sind Parameter, die auch für die relativ starre Konfiguration von Sensornetzwerken auf Raumfahrzeugen wichtig sind. Im Normalfall melden sich in einem solchen Netz keine weiteren Knoten an. Es ist jedoch denkbar, mehrere Sensorknoten zur Erhöhung der Ausfallsicherheit einzusetzen. Im Fehlerfall muss das Netz dann den neu aktivierten Knoten integrieren können. DSME und TSCH verhalten sich hier ähnlich wie ein klassisches 802.15.4 Netz und integrieren neue Knoten im laufenden Betrieb. LLDNs benötigen hierzu dagegen eine Rekonfigurationsphase, in der der normale Netzbetrieb unterbrochen wird. Für den hier vorgestellten Anwendungsfall spielt dies jedoch keine große Rolle und fließt daher mit einer entsprechend geringen Gewichtung in die Bewertung ein.

Nicht nur die funktionalen Eigenschaften des MAC Layers sind entscheidend. Einer der wichtigsten Faktoren ist die generelle Eignung des MAC für die Verwendung mit Ultrabreitband Transceivern. Eines der Hauptmerkmale von TSCH und DSME ist ihre Fähigkeit Channel Hopping durchzuführen, um Störungen vorzubeugen. Genau diese Funktionen bietet UWB aber bereits auf physischer Ebene durch die breitbandige Übertragung. Ein zusätzlicher Kanalwechsel ist daher wenig sinnvoll und kann aufgrund des elektrischen Aufbaus des Transceivers u.U. zu Problemen führen. Denkbar wäre ein Betrieb von TSCH und DSME in einem Einkanal Modus. Zumindest bei DSME ist jedoch die parallele Übertragung über mehrere Kanäle eine Grundfunktion, die der Erhöhung der Übertragungsrates im Gesamtnetz dienen soll.

Berücksichtigt man all diese Faktoren, lässt sich die Entscheidungstabelle [4.1 auf der nächsten Seite](#) aufstellen, die die Ergebnisse nochmal zusammenfasst. Die Gewichtungen erfolgen dabei zwischen eins und drei und entsprechen der Bedeutung des Faktors für die Erfüllung der Anforderungen, wobei der Wert drei einen sehr wichtigen Faktor beschreibt und der Wert eins einen weniger wichtigen. Entsprechend erfolgt die Bewertung der einzelnen Kriterien auf der gleichen Skala.

Im Ergebnis liegen die Low Latency Deterministic Networks vorne. Damit ist die Wahrscheinlichkeit, dass eine MAC Implementierung auf Basis dieser Erweiterung die gesteckten Ziele erfüllt am größten.

Nicht alle Funktionen des LLDN sind für den Einsatz in einer Satellitenmission nötig. Da es sich um fest installierte Netze mit nicht variabler Knotenzahl handelt, die nicht dynamisch konfiguriert werden müssen, können viele Funktionen des LLDN in einer ersten Implementierung vernachlässigt werden. Da in der WICome Nutzlast lediglich

Kriterium	Gewichtung	TSCH	DSME	LLDN
Latenz	3	1	2	3
Datendurchsatz	2	2	3	2
Energieeffizienz	1	2	1	3
Komplexität	1	2	1	3
Netzrekonfiguration	1	3	3	1
Eignung für UWB	3	1	1	3
Gewichtete Summe		17	20	29

TABELLE 4.1: Entscheidungsmatrix zur Auswahl des zu implementierenden Standards

die Leistungsfähigkeit dieser MAC Anpassung in Kombination mit UWB untersucht werden soll, können weitere Einschränkungen gemacht werden. Tabelle 4.2 liefert hier einen Überblick über die Grundfunktionen, die LLDN bietet und wo diese implementiert werden sollten. Die Spalte *Satellitenmission* gibt die Verwendbarkeit einer Funktion für Satellitenmissionen im Allgemeinen an. Für das WICome Experiment gibt es Funktionen, die zur Einschätzung des MACs nötig sind und im Rahmen dieser Arbeit implementiert werden. Die Spalte *optional* gibt Funktionen an, die je nach Ausgestaltung der anzusteuernenden Nutzlast und der zu übertragenden Daten notwendig sein können und in weiterer Arbeit implementiert werden müssten.

Funktionalität	Einsetzbar in		
	Satellitenmission	WICome optional	WICome
Discovery Phase			
Configuration Phase	■		
Online Phase	■	■	■
Bidirectional Timeslots	■	■	
Group ACKs	■	■	■
Separate Group ACKs	■		
Shared Group Timeslots	■		
Beacon Frames	■	■	■
Command Frames	■		
ACK Frames	■	■	
Data Frames	■	■	■

TABELLE 4.2: Umzusetzende Features der LLDN Spezifikation

Für WICome sind die Phase Discovery und Configuration allgemein nicht relevant. Es handelt sich um eine feste Anzahl von Netzwerkknoten mit einer definierten Aufgabe, die keiner Neukonfiguration der Netzwerkparameter bedarf. Anders verhält es sich bei Satellitenmissionen allgemein. Hier kann eine Configuration Phase sinnvoll sein, um

Parameter des Netzes, wie die Länge der Slots oder die Verteilung von Uplink und bidirektionalen Slots zu verändern. Bedient ein Knoten hier bspw. mehrere Datenquellen, kann es nötig sein, das Netzwerk statt auf geringere Latenz auf einen höheren Durchsatz pro Slot zu optimieren. Trotzdem ist auch hier eine Discovery Phase wenig sinnvoll, da die Anzahl der Knoten bekannt ist. Bei Entwicklung eines generischen Produktes für Industriesteuerungen, für die LLDN ursprünglich vorgesehen ist, ist die Discovery Phase hingegen sinnvoll, da so die Ermittlung der Netzwerkkonfiguration bei zuvor unbekannter Knotenanzahl auf Knopfdruck möglich ist.

Bidirektionale Timeslots sind hingegen je nach Ausrichtung der Nutzlast auch für WI-Come relevant, da hier in der Online Phase zusätzlich Informationen vom Coordinator an die Knoten übertragen werden kann, z.B. um Kommandos an die angeschlossenen Subsysteme zu geben. Zur Evaluation der Leistung eines LLDN ist dies generell aber nicht notwendig. Damit sind auch die separaten ACK Frames nicht nötig, da sie nur für die Bestätigung des Empfangs von Nachrichten des Coordinators in einem Downlink Slot der bidirektionalen Slots Verwendung finden.

Da Command Frames nur für die Discovery und Configuration Phase Verwendung finden, müssen sie auch nur bei Verwendung dieser implementiert werden. Die dazugehörigen Management Timeslots, die während der beiden Phasen genutzt werden, sind nach Standard Shared Group Timeslots, erlauben also, dass mehrere Knoten während eines solchen Slots konkurrierend auf das Übertragungsmedium zugreifen. Wie in [4.3.4.2 auf Seite 74](#) dargestellt, müsste bei Verwendung von UWB hierzu statt CSMA/CA ein ALOHA Zugriffsverfahren implementiert werden.

Beacon und Data Frames müssen natürlicherweise in allen Fällen implementiert werden. Beacons enthalten unter anderem das Group ACK Bitfeld, das die erfolgreichen Übertragungen aus dem letzten Superframe markiert. Damit ist die Anzahl der möglichen Neuübertragungen auf eine begrenzt, wenn keine erneuten Übertragungen in den jeweils nächsten Superframes erfolgen sollen. Für Satellitenmissionen allgemein könnte das Separate GACK Feature interessant sein, da hier mit dem zusätzlichen GACK Feld aus dem Beacon zwei Retransmit Versuche im folgenden Superframe möglich sind. So lässt sich bei Verlust von kritischen Daten mit geringerer Latenz und höherer Zuverlässigkeit eine Neuzustellung erreichen.

4.1.2 Auswahl eines RTOS

Eine weitere grundsätzliche Entscheidung für die Umsetzung stellt die Wahl eines geeigneten Betriebssystems dar, das die Integration des MAC Layers in einen bestehenden OSI

Stack ermöglichen soll (REQ-13). Aktuell gibt es eine Vielzahl von embedded Betriebssystemen, die für unterschiedlichste Anwendungen optimiert sind. Nach Anforderung REQ-15 muss ein Echtzeitbetriebssystem verwendet werden, da z.B. Regelungsanwendungen hier eine Prozessverwaltung verlangen, die besonders kurze Kontextwechselzeiten ermöglicht und deterministisches Zeitverhalten garantiert.

Generell bieten eingebettete Echtzeitbetriebssysteme keinen besonders großen Funktionsumfang im vgl. zu typischen Kernen von gewöhnlichen Betriebssystemen. Das Hauptaugenmerk liegt auf dem Scheduler, der den Prozessen ihre Rechenzeit zuweist und ggf. einer einfachen Speicherverwaltung, die Multithreading-fähig ist. Gerätetreiber, wie sie für andere Betriebssystemkernel üblich sind, sind oft nicht Teil von eingebetteten Betriebssystemen. Im Normalfall wird der Applikationscode zusammen mit dem Betriebssystemkernel statisch gelinkt. Auf Basis dieser Grundfunktionalität gibt es einige Systeme, die sich drahtlosen Sensornetzwerken verschrieben haben und entsprechende Protokollstacks für gängige IoT Protokolle mitbringen. Für die Auswahl werden im Folgenden die momentan am weitesten verbreiteten Betriebssysteme dieser Art, Contiki, openWSN und RiOT verglichen.

Einer der wichtigsten Faktoren ist die Unterstützung möglichst vieler Hardwareplattformen. So lässt sich sicherstellen, dass die entwickelte Lösung auch in Zukunft für andere Mikrocontrollerplattformen zur Verfügung steht. Contiki setzt hauptsächlich auf die Texas Instruments TI MSP430 Serie und bietet darüber hinaus wenig Unterstützung für andere Plattformen. RiOT unterstützt neben den MSP430 auch ARM Cortex Kerne von ST Microelectronics. Da openWSN auf freeRTOS basiert, ist es auf nahezu allen gängigen Mikrocontrollerplattformen lauffähig.

Alle der untersuchten Betriebssysteme sind auf den Einsatz im Internet-of-Things optimiert und bieten daher auch einen passenden Protokollstack. Neben typischen Implementierungen von z.B. TCP und UDP wird auf Applikationsebene auch CoAP als wichtiges Protokoll für IoT Anwendungen angeboten. Auf IP Ebene finden sich neben IPv4 und IPv6 auch das auf drahtlose Sensornetze optimierte Routingprotokoll RPL. In einem Adaption Layer wird zusätzlich 6LoWPAN eingesetzt, das redundante Information in Headern entfernt. Alle untersuchten Systeme unterstützen die genannten Protokolle, unterscheiden sich jedoch teilweise in der Implementierung der MAC Layer. Eine Ausnahme bildet Contiki, dass zusätzlich bereits 6TiSCH implementiert, eine Adaptierung, die speziell auf TSCH zugeschnitten ist. Durch den Einsatz von LLDN hat dies für die Bewertung jedoch keine Relevanz.

Die Reife des eingesetzten Betriebssystems spielt gerade für den Einsatz in kritischen Umgebungen eine wichtige Rolle. Hier liegt FreeRTOS durch die lange Entwicklungszeit und viele Audits des Systems klar vorne. Mit SafeRTOS existiert bspw. eine Version

von FreeRTOS, die nach EN 61508 Safety Integrity Level 3 (SIL3) zertifiziert ist. Am schlechtesten schneidet hier Contiki ab, da eine Integration von verschiedenen Projekten in einen gemeinsamen Stand aktuell schon lange nicht mehr vorgenommen wurde und zusätzlich viele Fehlerberichte teils kritischer Probleme unbearbeitet sind.

Zuletzt muss auch der Speicherverbrauch des Systems betrachtet werden. Alle der untersuchten Systeme sind auf möglichst geringen Speicherverbrauch optimiert, damit sie auf den genannten Mikrocontrollern eingesetzt werden können. Einzig openWSN sticht durch die Basis von freeRTOS hier etwas heraus, da dieses durch seine geringe Komplexität und Optionsschaltern zur Compilezeit besonders kleine Firmwares erzeugen kann.

Kriterium	Gewichtung	Contiki	openWSN	RiOT
Platformunterstützung	3	1	3	1
IoT Protokolle	1	2	2	2
Reife	3	1	3	2
Speicherverbrauch	2	2	3	2
Gewichtete Summe		12	26	15

TABELLE 4.3: Entscheidungsmatrix zur Auswahl des embedded Betriebssystems

In Tabelle 4.3 sind diese Betrachtungen wie auch bei der Auswahl der 802.15.4 Erweiterung zusammengefasst und gewichtet worden. Damit ist openWSN für die geplante Implementierung am Besten geeignet und wird mit LLDN eine MAC Implementierung für UWB erhalten.

4.1.3 Architektur

Der LLDN MAC Layer muss in den bestehenden Netzwerk Stack des openWSN Betriebssystems eingebettet werden. Das Ziel von openWSN ist es, ein vollständiges IoT Betriebssystem zur Verfügung zu stellen, das auf der TSCH Erweiterung des 802.15.4e basiert und die entsprechenden übergeordneten Schichten zur Verwaltung der TSCH Parameter mitbringt. Aktuell ist hier eine vorläufige Version von 6top¹ implementiert, die als Teil von 6TiSCH² als Schicht über dem eigentlichen MAC für die verteilte Organisation des Scheduling und des Sprungmusters über die verfügbaren Kanäle verantwortlich ist. Wie in 3.2.4.1 auf Seite 39 beschrieben ist dieses Link Scheduling nicht Teil des IEEE Standards und daher auf höherer Ebene zu realisieren. Die IETF hat mit 6top in 6TiSCH ein solches Protokoll geschaffen.

Das OSI Modell gibt eine Schichtenarchitektur vor, in der ein zu versendendes Paket mit Headern und Trailern für die einzelnen in den Schichten realisierten Protokolle versehen

¹6TiSCH Operation Sublayer

²6TiSCH – IPv6 over the TSCH mode of IEEE 802.15.4e

wird. Standards wie auch der IEEE 802.15.4 definieren Schnittstellen, die die Schichten zur Verfügung stellen müssen, um interoperabel eingesetzt werden zu können.

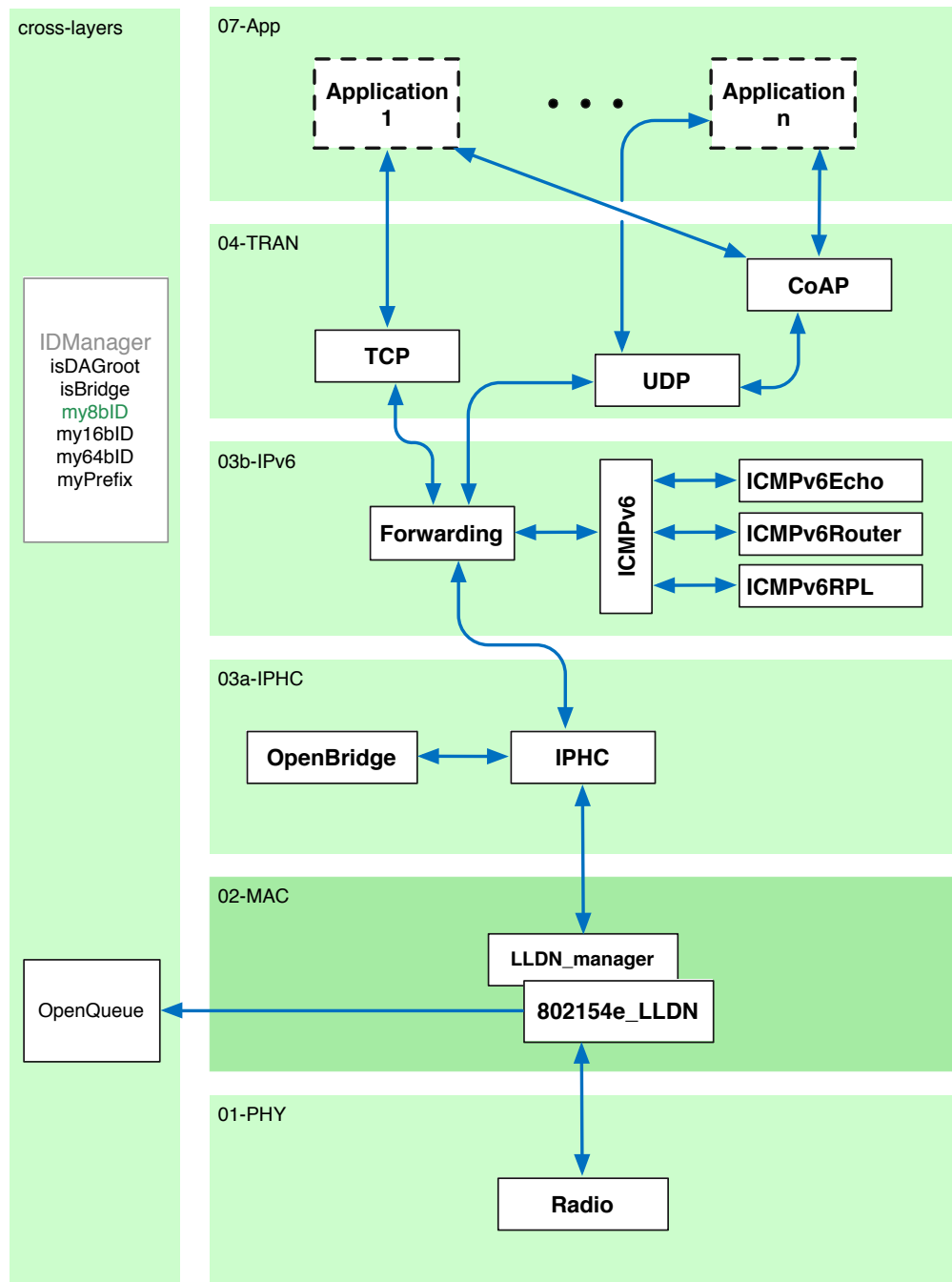


ABBILDUNG 4.1: Architektur des Netzwerkstacks³

Der openWSN Stack stellt den Applikationen eine Transportschicht zur Verfügung, in der die gängigen Transportprotokolle TCP und UDP, sowie das auf Sensornetze optimierte Constrained Application Protocol (CoAP), das auf UDP basiert, angeboten

³Adaptiert nach: https://openwsn.atlassian.net/wiki/download/attachments/1081394/stack_organization.png

werden. Darunter liegt die OSI Schicht 3, die durch die Adressierung der Knoten mit IPv6 das Routing über RPL ermöglicht. Dazu steht ein ICMPv6 Stack mit entsprechendem Forwarder zur Verfügung (siehe Abb. 4.1 auf der vorherigen Seite). Die nächst tiefere Schicht ist ein Adaptierungslayer nach 6LoWPAN. In Netzwerkstacks, wie sie in IoT Systemen Verwendung finden, sorgt das Protokoll für eine Komprimierung der IP Header, unterstützt das Routing in Mesh Netzwerken, wie sie im 802.15.4 häufig anzutreffen sind und nimmt eine Fragmentierung der Pakete vor, da der 802.15.4 Standard eine maximale Paketlänge von 127 Byte erlaubt und die größeren IP Pakete auf diese heruntergebrochen werden müssen. Auf dieser Ebene ist in openWSN auch die *OpenBridge* implementiert. Sie erlaubt das Weiterleiten von Paketen über eine serielle Verbindung an andere Systeme, sodass das Sensornetzwerk z.B. an das Internet angebunden werden kann.

Die 6LoWPAN Schicht reicht die Pakete schließlich an den MAC weiter. In der TSCH Version von openWSN ist diese Schicht zweigeteilt und besteht aus dem eigentlichen TSCH MAC auf einer niedrigeren Ebene und der bereits erwähnten 6top Verwaltungsschicht darüber. Der für diese Arbeit zu implementierende LLDN MAC ersetzt nun diese geteilte Schicht mit einer einzigen LLDN MAC Schicht (siehe Abb. 4.1 auf der vorherigen Seite). LLDN benötigt im Gegensatz zu TSCH keine übergeordnete Verwaltungsschicht, wie sie zur Verteilung der einzelnen Links hier nötig ist. Alle Knoten synchronisieren sich hier auf den Beacon des Coordinators. Die Entscheidung, ob ein Knoten Coordinator oder einfacher Knoten im Netz ist, muss zusammen mit den üblichen Parametern, wie der PAN ID auch hier auf höherer Ebene erfolgen. Innerhalb von openWSN wird diese Konfiguration auf Applikationsebene vorgenommen. Auch das Verhalten des Coordinators, wie z.B. der Wechsel zurück in die Konfigurationsphase wird auf höherer Ebene vorgenommen und ist nicht selbstverwaltet innerhalb der MAC Schicht.

Das *LLDN_manager* Modul nimmt nun das Paket vom IPHC (IP Header Compression) Modul der übergeordneten Schicht entgegen, fügt die entsprechenden MAC Parameter im MAC Header hinzu und entscheidet anhand des aktuellen Zustands darüber, wie die Nachricht versendet wird und reicht umgekehrt die empfangenen Nachrichten an den höheren Layer zur Dekodierung weiter. Handelt es sich um einen normalen Knoten im Netz, wird dieser zu den in der Konfigurationsphase zugewiesenen Uplink Slots einen Sendevorgang des Pakets über den PHY Layer anstoßen, um ein Paket zum Coordinator zu übertragen. Zuvor hat dieser in der Initialisierung des Moduls die Discovery- und die Konfigurationsphase, die durch den Coordinator diktiert werden, durchlaufen. Die Verwaltung dieses Zustandsautomaten ist ebenfalls Aufgabe des Moduls. Alle Knoten können außerdem in den für die bidirektionale Kommunikation des Frames vorgesehenen Zeiten senden und empfangen. Das *802154e_LLDN* Modul stellt dabei eine Schnittstelle für die Operationen auf Slot Ebene zur Verfügung, sodass die Knoten zur richtigen Zeit ihren zugewiesenen Timeslot wahrnehmen. Das Modul wird ebenfalls für den Ausgleich der

zeitlichen Drift zwischen den Knoten verwendet, indem es die Sendezeiten der Coordinator Beacons als Anhaltspunkt für die Driftkorrektur verwendet.

Ist der richtige Zeitpunkt für den MAC gekommen, um zu empfangen oder zu senden, greift er über eine Schnittstelle auf den PHY Layer zu. Dieser sorgt für die Kommunikation mit dem Funkmodul und muss im Rahmen dieser Arbeit die Anbindung an einen SPI (Serial Peripheral Interface) Treiber schaffen, der dafür sorgt, dass die gewünschten Daten von oder zum Funkmodul übertragen werden.

Abschließend werden in einem *cross-layers* Bereich alle weiteren Funktionen zusammengefasst, die sich keiner bestimmten Schicht zuordnen lassen und von allen gleichermaßen verwendet werden. Dazu gehören u.A. die Verwaltung der verschiedenen möglichen MAC Adressen, sowie das OpenQueueElement. Dieses stellt die Basisdatenstruktur für alle Verarbeitungen in den Schichten dar und enthält alle nötigen Informationen aus diesen, um einfach durch den Protokollstapel gereicht werden zu können.

Die bestehende Architektur wird dabei erweitert und soll mit dem eigentlichen openWSN Stack koexistieren können. Dadurch können die anderen Schichten ohne Änderungen übernommen werden und zukünftige Erweiterungen können einfach integriert werden. Dennoch müssen z.B. in den cross-layer Modulen Erweiterungen vorgenommen werden, um diese sinnvoll kompatibel zum LLDN MAC zu halten.

4.2 Verwendete Hardware

Für drahtlose Sensornetzwerke ist neben der Transceiverhardware, die die physische Übertragung durchführt auch eine Mikrocontrollerplattform nötig, die für die Ausführung der höheren Netzwerkschichten und der eigentlichen Applikationen nötig ist. Dabei bieten moderne 32-Bit Architekturen die nötige Leistung, um Transceiver und komplexe Netzwerkstacks betreiben zu können.

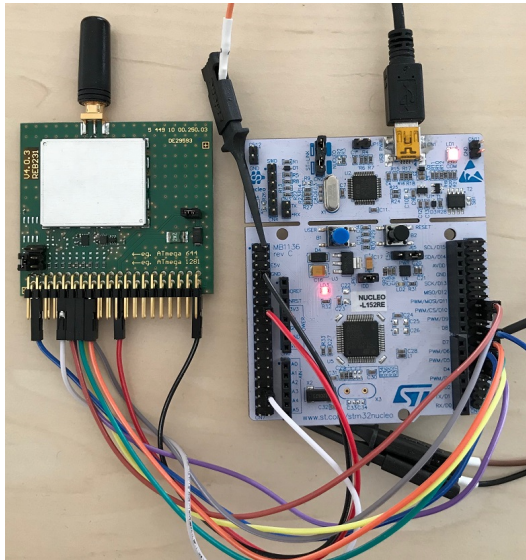
Im Gegensatz zu Systemen, die Mikroprozessoren mit dediziertem Speicher und Peripherie einsetzen, vereint ein Mikrocontroller all diese Funktionen in einem platzsparenden Chip bei sehr geringem Energieverbrauch. Neben der CPU ist meist RAM und häufig Flash ROM bereits integriert, sodass ohne weitere Bauteile bereits Software ausgeführt werden kann. Zur grundsätzlich verbauten Peripherie zählen dabei auch Hardware Timer und Interruptcontroller, die interne und externe Unterbrechungsanforderungen an die CPU verarbeiten. Je nach Hersteller und Modell des Mikrocontrollers werden verschiedene Schnittstellen zur Kommunikation mit anderen Komponenten ebenfalls bereitgestellt. Dazu gehören z.B. I²C oder SPI, die die Anbindung an serielle Bussysteme ermöglichen,

oder auch die klassische UART Kommunikation. Im Fall von IoT Geräten werden Schnittstellen der Peripherie genutzt, um mit der Transceiverhardware zu kommunizieren. Bspw. wird der Transceiver über eine der genannten seriellen Verbindungen zum Senden oder Empfangen von Daten aufgefordert. Zusätzlich werden auch die genauen Hardware Zeitgeber verwendet, um die im MAC Layer festgelegten Sende- und Empfangszeiten einzuhalten. Nur so ist eine stabile Kommunikation in komplexen MAC Implementierungen wie TSCH, die das Einhalten sehr präziser Zeitvorgaben fordern, erst möglich.

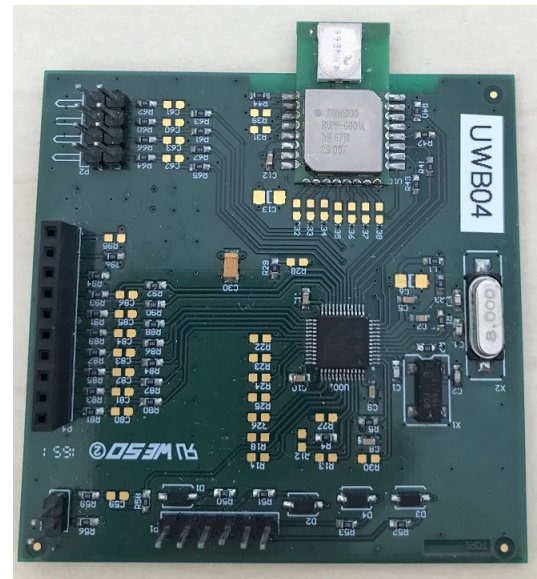
Heutzutage am häufigsten eingesetzt werden immer noch 8-Bit Mikrocontroller. 32-Bit fähige Chips sind jedoch mittlerweile fast ebenso günstig in der Beschaffung und häufig einfacher einzusetzen. So haben diese durch ihre 32-Bit Adressen die Möglichkeit, direkt einen großen Adressbereich für Speicher und Peripherie zu verwalten. Im Vergleich zu 8-Bit Controllern sind sie relativ zu ihrer Leistung zudem energieeffizienter. ARM Ltd. gehört dabei zu den Firmen, deren Prozessorarchitektur am weitesten verbreitet ist. Dabei stellt diese nur das Referenzdesign des Controller Kerns zur Verfügung und fertigt selbst keine Chips. Mit ARM Cortex-M wurde eine Architektur für 32-Bit Mikrocontroller geschaffen, die für eingebettete Systeme besonders interessant ist, da sie u.A. über einen integrierten Hochgeschwindigkeitsbus verfügt, um Peripheriecontroller anzubinden. Zusätzlich bietet der integrierte Nested Vectored Interrupt Controller (NVIC) eine echtzeitfähige Verwaltung von Interrupts in verschiedenen Prioritätsstufen. Durch eine initial vergebene Vektortabelle ist die Zuordnung zur Interrupt Anfrage mit geringer Latenz möglich (12 Taktzyklen). Mit der Cortex-M3 Architektur konnten im Jahr 2006 die ersten 32-Bit Mikrocontroller mit einem Stückpreis unterhalb der 1 \$ Grenze gefertigt werden. Die eigentliche Fertigung der Controller wird von Lizenznehmern wie STMicroelectronics übernommen. Diese ergänzen das lizenzierte Referenzdesign um die Peripherie, wie Art und Anzahl der Schnittstellen, Ausprägung der Hardwaretimer und natürlich um die Ausstattung mit Speicher.

Die Cortex-M Architektur teilt sich in die Klassen M0, M1, M3, M4 und M7 auf, wobei diese aufsteigend komplexer werden und einen größeren Funktionsumfang zur Verfügung stellen. Controller der M0 und M1 Klasse sind zwar sehr günstig und energieeffizient, bieten aber nicht den kompletten Befehlssatz und damit auch nicht die Leistungsfähigkeit der komplexeren Modelle. Chips auf Cortex-M3 Basis unterstützen dagegen den vollen Befehlssatz und beinhalten z.B. bereits einen SysTick Timer und können eine Memory Protection Unit (MPU) enthalten. Beides sind Funktionalitäten, die für den Betrieb von Betriebssystemen auf den Mikrocontrollern wichtig sind. Vorhergehende Arbeiten am DLR, wie WirelessCompose [DSP17] setzen zum Betrieb der Ultrabreitbandtransceiver auf STM1151 Mikrocontroller mit einem Cortex-M3 Kern. Dieser wurde ausgewählt, da er die höchstmögliche Energieeffizienz und trotzdem genügend Leistung bietet. Die l1xx Serie von STMicroelectronics bietet die energieeffizientesten M3 Kerne mit einer geringen

Taktrate von bis zu 32Mhz. Das eingesetzte Modell hat dabei eine Speicherausstattung von 256 kB an Flash mit 80 kB SRAM. Trotzdem sind gängige serielle Schnittstellen und z.B. ein SDIO Interface zum Betrieb von SD-Karten integriert. Wie bei großen Mikrocontrollerherstellern üblich, bietet auch STM eine Peripheriebibliothek, mit der für die Softwareentwicklung der Zugriff auf die Hardware erleichtert wird.



(a) Verwendetes Evaluationsboard mit integriertem Programmieradapter auf Basis eines STM1152RE Controllers mit 2,4 GHz Modul auf einem Breakout Board



(b) Die eingesetzte Zielhardware. Integriert auf dem Board sind neben dem Mikrocontroller und dem DW1000 auch Stromversorgung Adapter für die Nutzung von SD Karten

ABBILDUNG 4.2: Verwendete Hardware – Gegenüberstellung der Hardware, die zur Entwicklung genutzt wurde und der letztendlichen Zielhardware

Im Gegensatz zu Mikrocontrollern gibt es bei Ultrabreitbandtransceivern noch keine unüberschaubare Anzahl an Modellen auf dem Markt. Für die vorangegangenen Arbeiten wurde hier das Decawave DW1000 Modul verwendet. Vorteilhaft bei diesem Modell ist die Unterstützung durch den Hersteller. Decawave stellt Treiber und ausführliche Dokumentation zur Verfügung. Außerdem sind umfangreiche Tests mit der Hardware durchgeführt und veröffentlicht worden. So gibt es verschiedene Untersuchungen z.B. zur erreichbaren Reichweite oder der erreichbaren Bitfehlerrate. Das Modul selbst stellt 6 Breitbandkanäle zwischen 3,5 und 6,5 GHz zur Verfügung. Dabei können drei verschiedene physische Übertragungsraten von 110 kb/s, 850 kb/s oder 6,8 Mb/s gewählt werden. Auf das Modul wird per SPI Verbindung zugegriffen. Durch das Schreiben in verschiedene Register wird die gewünschte Aktion ausgelöst. So kann bspw. ein Sendepuffer befüllt werden und durch ein weiteres Signal das Modul zum Senden des Pufferinhaltes bewegt werden. Empfangene Daten werden ebenso über einen Empfangspuffer abgeholt. Um verschiedene Zustandsänderungen zu signalisieren verwendet der DW1000 ein Interrupt

Signal, das mit dem steuernden Mikrocontroller verbunden ist und diesem so z.B. signalisieren kann, wenn Daten empfangen wurden oder der Sendevorgang abgeschlossen ist.

Die Kombination dieser Hardwarekomponenten ist energieeffizient und sehr günstig herzustellen. Da sie sich in vorhergehenden Experimenten bewährt hat, wird sie auch im Rahmen dieser Arbeit für die Weiterentwicklung verwendet. [Abbildung 4.2 auf der vorherigen Seite](#) zeigt die verwendete Hardware während der Entwicklung und die finale Zielhardware, auf der die Evaluation durchgeführt wurde. Gut zu sehen ist hierbei, dass in der Entwicklung ein Evaluationsboard mit den vielen herausgeführten Pins von Vorteil sein kann. So können hier sehr einfach Messungen durchgeführt oder Logic Analyzer angeschlossen werden.

4.3 Implementierung

Der zu implementierende LLDN MAC Layer muss in den Netzwerkstack des openWSN Betriebssystems integriert werden. Die Herausforderung hierbei liegt in der Wiederverwendung der bestehenden Schichten, sodass die eigentliche TSCH Implementierung von openWSN weiter verwendet werden kann. Zusätzlich muss das gesamte System auf die Zielplattform des STM151 portiert werden, da dieser spezielle Controller von openWSN nicht unterstützt wird. Für die DW1000 Ultrabreitbandtransceiver ist außerdem die Integration eines Treibers in die Abstraktionsschicht für Funkhardware von openWSN erforderlich. Schlussendlich kann dann die Implementierung des eigentlichen MAC Layers erfolgen. Dessen Entwicklung wird durch eine zusätzlich implementierte Simulationsumgebung unterstützt, die es erlaubt, das System auf PCs mit gängigen Desktop Betriebssystemen auszuführen und so eine wesentlich bessere Testbarkeit und schnelleres Debugging ermöglicht.

Die Firmware von openWSN basiert auf einem von zwei möglichen Betriebssystemkernen, die ausgewählt werden können. Den Normalfall stellt FreeRTOS dar, das einen Scheduler für präemptives Multitasking zur Verfügung stellt. Alternativ und zu Debuggingzwecken existiert eine Eigenentwicklung des openWSN Teams, die nicht präemptiv arbeitet und eine wesentlich geringere Komplexität aufweist. Im laufenden System kümmern sich drei FreeRTOS Tasks um die Ausführung des Netzwerkstacks und der Applikationen. Einer dieser Tasks behandelt die Applikationen mit niedriger Priorität, wohingegen die restlichen beiden für Callbacks ausgelöst durch Interrupts (bspw. bei Empfang eines Paketes) und die Behandlung von Timern verwendet werden. Daneben existiert der

Netzwerkstack, in den der LLDN MAC integriert werden muss und der in [4.1.3 auf Seite 56](#) genauer beschrieben ist. Schließlich stellen die Board Support Packages (BSP) und Treiber die Schnittstelle zur entsprechenden Hardware her.

In den folgenden Abschnitten wird nun die Implementierung der einzelnen Komponenten, die für die Anpassung von openWSN nötig sind, im Detail erläutert.

4.3.1 Board Support Package

Ein Board Support Package bindet eine spezifische Hardwarekomponente an die zu entwickelnde Firmware an. Das sind zum einen die Mikrocontroller, auf denen die Firmware ausgeführt werden soll, aber auch die unterschiedlichen Transceiver, die unterstützt werden. In openWSN existiert hierzu eine Abstraktionsschicht aus Headern, die grundlegende Funktionen für den Zugriff auf die Hardware deklarieren. Die eigentliche Implementierung ist von Board zu Board unterschiedlich. Bspw. kann der Zugriff auf das SPI Interface, das für die Kommunikation mit vielen Transceivern verwendet wird, die Verwendung anderer Hardwareregister nötig machen.

Für die Portierung auf den STM115x war zuerst die Verwendung einer passenden Peripheriebibliothek vonnöten. Im System ist bereits ein BSP für die STM F3 Klasse von Mikrocontrollern vorhanden, aus denen Teile wiederverwendet werden konnten. Die beiden Peripheriebibliotheken weichen jedoch an vielen Stellen voneinander ab. Insbesondere besitzt der L1 Chip eine etwas neuere Hardwarearchitektur und nutzt den neueren AHB (Advanced High Performance Bus) im Vergleich zum langsameren APB (Advanced Peripheral Bus), was Änderungen im BSP zur Folge hat. Die Gestaltung der Hardwaretimer für die Echtzeituhr ist grundsätzlich verschieden, da sie eine völlig andere Hardwarearchitektur benutzen.

Grundsätzlich muss für jedes neue BSP die Zuordnung aller Schnittstellen, wie SPI, I²C oder UART, überprüft werden. Diese sind in der Peripheriebibliothek größtenteils gleich zur vorhandenen F3 Bibliothek mit der Ausnahme der Zuordnungen zu den Interrupts. Diese werden über den NVIC initialisiert und weisen einige Hardwareunterschiede auf, sodass sie anders initialisiert werden müssen. Ein Modul zur Verwaltung der Timer für die Funkhardware musste ebenfalls neu geschrieben werden, da diese anders arbeiten und je nach Hardwartertimer auch eine höhere 32-Bit Auflösung bieten.

Im Gegensatz zur Softwareentwicklung unter Desktopbetriebssystemen wird bei der Entwicklung von eingebetteten Systemen eine statische ausführbare Datei erstellt, die sowohl Betriebssystem und Applikationscode enthält. Insbesondere enthält sie auch den Startup Code, der die Hardware initialisiert und für jeden Controller angepasst werden

muss. Dieser enthält die Vektortabelle der Interrupts für den NVIC und die weitere Initialisierung, wie das Resetverhalten und das Setzen der verschiedenen Taktfrequenzen und Skalierungsfaktoren für den Systemtakt und den Peripheriebus. Abschließend kopiert dieser initialisierte Daten in den RAM, alloziert Platz für den Stack, der von in C geschriebener Software erwartet wird, initialisiert ihn und setzt den Stackpointer des Prozessors, woraufhin `main` aufgerufen wird. Zuletzt wird ein Linker Skript benötigt, das den Zusammenbau der einzelnen Codesegmente steuert. Insbesondere muss dieses sicherstellen, dass der Startupcode am Beginn der Datei liegt, um zuerst ausgeführt zu werden.

4.3.2 Simulationsumgebung

Die Implementierung eines MAC Layers auf der Hardware allein ist besonders mühsam. Um die korrekte Funktionsweise eines Teilaspekts zu prüfen, muss der Code für die Hardware übersetzt, anschließend auf alle Boards geflasht werden und mittels eines Hardwaredebuggers oder Logicanalyzers die korrekte Funktion überprüft werden. Je nach verwendetem Hardwaredebugger kann allein die Übertragung des neuen Codes auf das Board sehr zeitaufwändig sein. Schnelle Iterationen sind so nicht möglich.

Daher wurde im Rahmen dieser Arbeit das openWSN System um die Möglichkeit erweitert, die Software für PCs übersetzen zu können. Dadurch können mehrere virtuelle Boards auf einem Rechner gestartet werden und miteinander kommunizieren. Dabei kann einfach auf die laufende Kommunikation und die Betriebsparameter der einzelnen Boards zugegriffen und diese überprüft oder getestet werden.

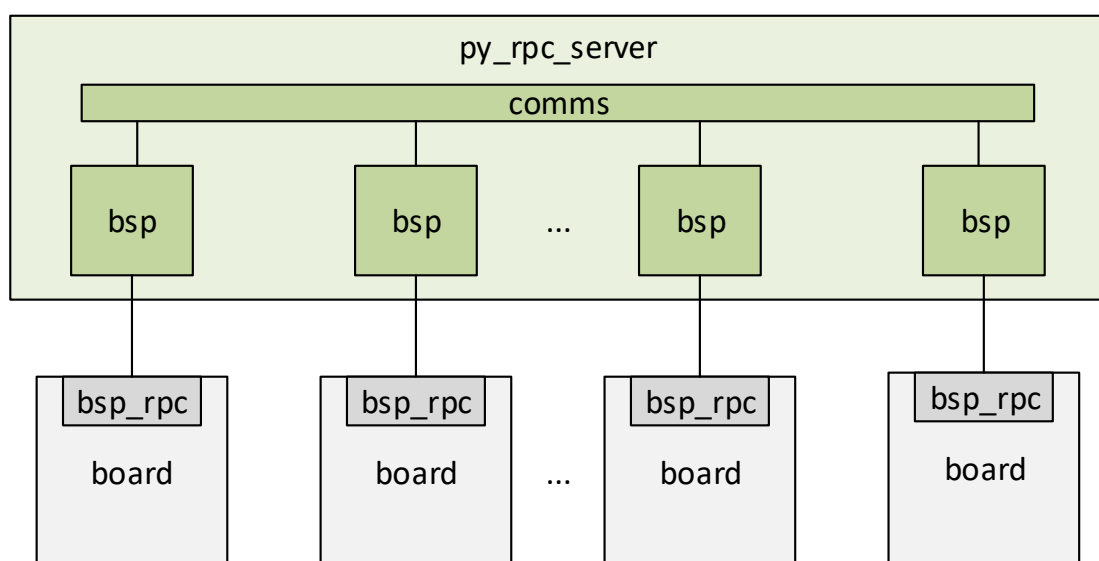


ABBILDUNG 4.3: Architektur der Simulationsumgebung

Der Großteil des Codes ist in reinem C geschrieben und lässt sich dementsprechend einfach auf gängigen Desktopbetriebssystemen übersetzen. Die Ausnahme stellt hier das BSP dar. Es greift entweder direkt auf Register der Zielhardware zu oder delegiert dies über die entsprechenden Peripheriebibliotheken. Für PCs musste hier also ein BSP geschaffen werden, das die abstrahierte, vorgegebene Hardware API von openWSN nachbildet. Realisiert wurde dies durch einen Remote Procedure Call (RPC) Ansatz, der über normale Socket Kommunikation eine Verbindung zu einem in Python implementierten Backend aufbaut und hier Kommandos absetzt und Daten abrufen kann. Jede der entsprechenden Funktionen im BSP ist dabei einem entsprechenden Kommando zugeordnet. Gleichzeitig findet ein regelmäßiges Polling statt, um Ereignisse vom Server abzufragen. [Abbildung 4.3 auf der vorherigen Seite](#) zeigt die Architektur dieses Systems und wie hier verschiedene virtuelle Boards miteinander kommunizieren.

Die Hardwarefunktionen, die das BSP im Normalfall zur Verfügung stellt werden nun durch den zentralen RPC Server zur Verfügung gestellt. Initial meldet sich ein Board mit einem Kommando und einer ID an, woraufhin der Server ein virtuelles BSP initialisiert, in dem der Zustand dieses spezifischen Boards abgelegt wird. Dazu gehören die beiden von openWSN verwendeten Hardwaretimer zur Verwaltung des Betriebssystemscheduler und der Radiotimer, der die Aktivitätsphasen des Funkmoduls steuert. Eine solche, simulierte Schnittstelle zu einem Funkmodul ist ebenfalls Teil des BSPs. Während ein reguläres BSP an dieser Stelle eine SPI Kommunikation mit der Hardware aufbaut, werden im simulierten BSP die Daten an den RPC Server übertragen. Dieser stellt den MAC Frame den anderen verbundenen Boards daraufhin in deren virtuellen Empfangspuffern zur Verfügung.

Neben der schnelleren Entwicklung bietet dieser Ansatz auch die Möglichkeit, einfacher Tests des Systems durchzuführen. Es können z.B. die virtuellen Timer der einzelnen Boards so beeinflusst werden, dass sie schneller auseinanderdriften und so der Synchronisierungsalgorithmus (siehe [4.3.4 auf Seite 68](#)) leichter geprüft werden kann. Da auch voller Zugriff auf die MAC Frames besteht, können bei diesen Fehler eingepflanzt werden, um die Fehlerkorrektur auf der Empfangsseite zu testen.

Das System ist in dieser Form sehr flexibel und kann bei Bedarf auf vielen Rechnern über das Netzwerk verteilt ausgeführt werden, um so sehr viele Knoten gleichzeitig zu simulieren.

4.3.3 DW1000 Treiber

Die für die LLDN Erweiterung eingesetzten Ultrabreitbandtransceiver benötigen einen Treiber und müssen in die Abstraktionsschicht des openWSN Betriebssystems eingebunden werden. Da das openWSN System ausschließlich für die TSCH Erweiterung des 802.15.4e entwickelt worden ist, zielt es auf günstige und einfach aufgebaute 2,4 GHz Transceiver ab. Das für TSCH charakteristische Channel Hopping dient u.A. dazu, eine höhere Übertragungssicherheit im überlasteten 2,4 Ghz Band zu erreichen, weshalb sich diese Technologie besonders für die hier eingesetzte Funkhardware eignet.

Die überwiegende Mehrheit der für den Einsatz mit Mikrocontrollern konzipierten Funkmodule für den relativ niedrigwertigen 802.15.4 Betrieb setzen auf die Kommunikation über einen Inter-IC Bus wie I²C oder SPI. Dabei werden von einem Treiber im embedded System entsprechende Register des Transceiver Chips über diesen Bus gesetzt. Bspw. kann so die Konfiguration des Moduls vorgenommen werden und z.B. der Kanal für die Übertragung gewählt werden. Auch die Betriebsmodi werden so gewählt und versetzen den Transceiver in den Sende- oder Empfangsmodus. Bevor das Senden signalisiert wird, muss zuvor ein Sendepuffer mit den gewünschten Nutzdaten befüllt werden.

Zusätzlich werden von den Transceivern Interrupts verwendet, um bestimmte Ereignisse zu signalisieren und z.B. einen steuernden Mikrocontroller, der sich im Schlafmodus befindet, aufzuwecken. Beim Senden von Paketen kann z.B. der Beginn und das Ende der Übertragung signalisiert werden. Ist der Sendevorgang abgeschlossen, kann der Controller z.B. entscheiden, ob er das Funkmodul wieder in den Empfangsmodus versetzt, damit dieses die Antworten auf die gesendeten Pakete erhalten kann. Noch wichtiger sind Interrupts für den Empfang von Paketen. Befindet sich ein Transceiver im Empfangsmodus, kann dieser bspw. signalisieren, dass eine Präambel und damit der Beginn eines Paketes empfangen wurde. Ist der Empfang abgeschlossen erfolgt ein weiterer Interrupt, woraufhin der Mikrocontroller den Empfangspuffer sicher auslesen kann und einen kompletten Frame erhält.

Der hier verwendete DW1000 UWB Chip setzt diese Funktionalität grundsätzlich ähnlich um wie die bisher unterstützten 2,4 Ghz Module. Diese werden wie der DW1000 auch über SPI angebunden, bieten aber nur den oben genannten Basisumfang ohne weitere Funktionen. Das DW1000 Modul bietet zusätzliche Unterstützungsfunktionen wie automatische ACKs, ein eigenes hochgenaues Timing System um den Transceiver zu bestimmten Zeiten aus dem eigenen Schlafmodus zu wecken und in den Empfangsmodus zu versetzen, sowie verschiedene Unterstützungsfunktionen zum Betrieb eines MACs. Grundsätzlich bietet es sich an, diese in der Implementierung auch zu nutzen, da sie

im Allgemeinen stromsparender sind, als dies auf Seite des steuernden Mikrocontrollers umzusetzen. Da im LLDN aber bspw. auch Group ACKs unterstützt werden müssen, konnte z.B. das automatische ACK Feature des Chips nicht verwendet werden. Auch die weiteren MAC Unterstützungsfunktionen sind nicht kompatibel mit dem Einsatz in LLDNs. Verwendung finden dagegen die automatische Berechnung der CRC Prüfsumme der übertragenen Pakete sowie der integrierte Timer.

Die Herstellerfirma Decawave stellt hierfür bereits einen Treiber zur Verfügung, der Abstraktionsfunktionen für die Kontrolle des Chips über die SPI Register zur Verfügung stellt. Zur Verwendung mussten lediglich die Interrupt und SPI Definition im bereits entwickelten BSP für den STM15x angepasst werden. Um die unterschiedlichen Transceiver aus dem MAC heraus über eine gemeinsame Schnittstelle ansprechen zu können, existiert im Betriebssystem eine `radio` Abstraktionsschicht, die für jedes unterstützte Funkmodul eine eigene Implementierung, typischerweise mit den eigens nötigen SPI Kommandos, enthält. Auch für den DW1000 wurde diese Schicht implementiert, wobei diese nicht direkt die Befehle per SPI absetzt, sondern hierfür den genannten Decawave Treiber verwendet. Die implementierte Radioabstraktion nutzt weitgehend Standardparameter, wie sie für den PHY nach 802.15.4e vorgesehen sind. Für zukünftige Anpassungen bietet der DW1000 darüber hinaus aber die Möglichkeit längere Frames, als die im Standard vorgesehenen 127 Byte zu versenden. Durch die nötige lange Präambel in der UWB Übertragung ist dies besonders beim schnellen 6,8 Mb/s Betriebsmodus von Vorteil und kann Energie sparen. Da das Modul insbesondere im Empfangsmodus mit ca. 160 mA Stromaufnahme besonders viel Leistung aufnimmt, ist das korrekte Wechseln der verschiedenen Betriebsmodi in der Radioabstraktion und der Nutzung durch den MAC besonders wichtig. Für Erweiterungen existiert neben Idle und Tiefschlafmodus auch ein *SNIFF* genannter Modus, bei dem bei bekannten Konfigurationsparametern der anderen Teilnehmer das Abhören der Präambel zeitweise unterbrochen werden kann und so weitere Energie gespart wird.

Trotz der Bemühungen, die einzelnen Schichten des Netzwerkstacks strikt voneinander zu trennen, müssen in der Realität einige Annahmen über die Hardware gemacht werden. Der TSCH MAC in openWSN geht z.B. implizit davon aus, dass bestimmte Interrupts auf jeden Fall vorhanden sind. Für die schnellen Kanalwechsel in TSCH ist eine besonders genaue Synchronisation der Zeiten der einzelnen Controller erforderlich. Ist die Abweichung zu groß, können Knoten im Netz ihren Einsatz verpassen und hören nicht im richtigen Slot zu oder schlimmer, senden in einem anderen Slot als ihnen zugewiesen ist. Für die Synchronisation verwendet der TSCH MAC die Ankunftszeit eines Frames und dessen Abweichung zum internen Timer. Die einfachen 2,4 GHz Funkmodule signalisieren dies über einen RX Interrupt nach Empfang des SFD. Die DW1000 UWB Transceiver bieten diese Möglichkeit jedoch nicht. Sie haben einen eigenen Controller an Bord, der

viele komplexere Funktionen ermöglicht. Da auch eine sehr genaue Ranging Funktion vorhanden ist, die sehr hoch aufgelöste Zeitstempel benötigt wird nur der Abschluss des Empfangs signalisiert. Der Zeitstempel des SFDs kann dann gesondert per SPI abgerufen werden.

Die Implementierung des LLDN MAC kann auf diesen Umstand hin angepasst werden. Da der TSCH MAC jedoch für die Evaluation als Vergleich weitere Verwendung finden soll, musste ein Workaround genutzt werden. Im Normalfall werden zwei Callback Funktionen aufgerufen, die bei Beginn des Empfangs und zum Ende Aktionen im MAC auslösen können und ihren Zeitstempel mitliefern. Diese können mit dem DW1000 Modul nur beide direkt hintereinander mit Ende des Empfangs aufgerufen werden, was einen falschen Empfangszeitstempel zur Folge hat, wenn der interne Timer des Mikrocontrollers genutzt wird. Dieser kennt lediglich den Zeitpunkt des Interrupts und sorgt durch die falsche Ankunftszeit für eine schnelle Desynchronisierung der Knoten. Das `radiotimer` Modul wurde nun so angepasst, dass eine Synchronisation des Mikrocontroller Timers mit dem internen Timer des DW1000 Moduls stattfindet. Bei einem Empfang werden weiterhin beide Callbacks zu Start und Ende direkt aufgerufen, wobei die übergebene Zeit des Startcallbacks nun die umgerechnete Zeit aus dem Empfangszeitstempel des DW1000 Moduls enthält. Damit bleibt die Kombination aus DW1000 und l15x kompatibel zum TSCH MAC und ist für die Evaluation einsetzbar.

4.3.4 LLDN MAC Layer

Für die Implementierung eines LLDN basierten MAC Layers ist neben der in den vorherigen Abschnitten beschriebenen Anbindung an den physical Layer und das Board Support Package auch die Anpassung der über dem MAC liegenden Schicht nötig. Im verwendeten openWSN System ist dies die 6LoWPAN Schicht, die über das IPHC (IP Header Compression) Modul die darunter liegende SixTOP Implementierung steuert, die den TSCH MAC kontrolliert. Für die LLDN Implementierung musste diese enge Verbindung der Module aufgelöst werden und eine direkte Anbindung des IPHC Moduls an das LLDN Modul erfolgen.

Wie in [4.1.1 auf Seite 50](#) dargestellt wird eine angepasste Implementierung des Standards vorgenommen, die diesen nicht komplett abdeckt. Neben dieser organisatorischen Einschränkung gibt es auch technische Gründe, die die Umsetzung von Basisfunktionalität beeinflusst. So bietet die verwendete UWB Technologie prinzipbedingt nicht die Möglichkeit, ein CSMA/CA Verfahren durchzuführen da kein Träger hierzu vorhanden ist. Grundsätzlich ist auch keine Form von CCA möglich, wenn die physische Präambel

der UWB Übertragung nicht empfangen wurde. Damit können die Shared Group Timeslots, wie sie in Retransmits und Management Timeslots Verwendung finden nicht standardkonform genutzt werden. Um diese dennoch nutzen zu können, müsste ein ALOHA Mechanismus implementiert werden, wie er vom Hersteller des UWB Transceivers empfohlen wird [Dec15, vgl. S. 196-197].

Der entwickelte MAC verwendet zwei Zustandsautomaten, um den Kommunikationsablauf nach Standard zu gewährleisten. Der untere der beiden organisiert den Ablauf innerhalb eines Slots und sorgt für die Einhaltung der verschiedenen Warteperioden, die für das Laden der Daten oder auch für die durch Drift ausgelöste Abweichung der Zeitsynchronisation eingehalten werden müssen. Darüber sorgt ein zweiter Automat für den korrekten Ablauf des Superframes und steuert so z.B. die Aufwachzeiten des Coordinators zu Beginn eines jeden Superframes, damit dieser den Beacon aussenden kann. Ist ein solcher Zeitpunkt erreicht, konfiguriert das Modul den darunterliegenden Slot-Zustandsautomaten, sodass dieser korrekt für das Senden oder Empfangen der verschiedenen Frame Typen eingerichtet ist.

Der LLDN MAC ist dabei voll in den openWSN Stack integriert und kann von Applikationen entweder allein genutzt werden, um den Overhead möglichst gering zu halten, oder aber Protokolle der höheren Schichten wie TCP und UDP verwenden. Ist in Zukunft die Portierung auf andere Hardware vonnöten, so muss lediglich das BSP angepasst werden.

4.3.4.1 Zustandsautomaten und Chronogramm

Netzwerkprotokolle im Allgemeinen und MAC Layer im Besonderen werden üblicherweise als Zustandsautomaten realisiert, da der Ablauf der Kommunikation mit seinen vielen möglichen Fehlerzuständen mit anderen Ansätzen schwer umzusetzen ist. Hinzu kommt, dass sich Zustandsautomaten sowohl formal analysieren, aber auch in der Praxis einfach testen lassen.

Wie in 4.1 auf Seite 57 bereits dargestellt, besteht die LLDN Schicht aus zwei Modulen, dem *LLDN* selbst und einem *LLDN Manager*. Das *LLDN Manager* Modul kümmert sich dabei um die übergeordnete Ablauforganisation der Kommunikation in den Superframes. Das Modul ist als Zustandsautomat realisiert, dessen Zustandsübergänge durch das Voranschreiten der Slots der darunter liegenden Schicht ausgelöst werden. Alle Zeitvorgaben ergeben sich damit aus Vielfachen der Slotlänge, womit keine zusätzlichen Timer nötig sind. Die für die einzelnen Phasen nötigen Zeitkonstanten sind den Vorgaben des Standards entnommen und bestimmen z.B. die Aussendung des Beacons, falls der Knoten der Coordinator ist, oder die Zeitpunkte der Upload- und bidirektionalen Timeslots. Außerdem muss das Modul die Verwaltung der ACKs und Retransmits

übernehmen. Im Gegensatz zu anderen MAC Layern erfolgt das Versenden von ACKs und die etwaige Neuaussendung von Paketen nicht direkt im Anschluss an die ursprüngliche Übertragung, sondern wird in Group ACK Feldern oder in ACK Frames und Retransmit Slots im nächsten Superframe vorgenommen.

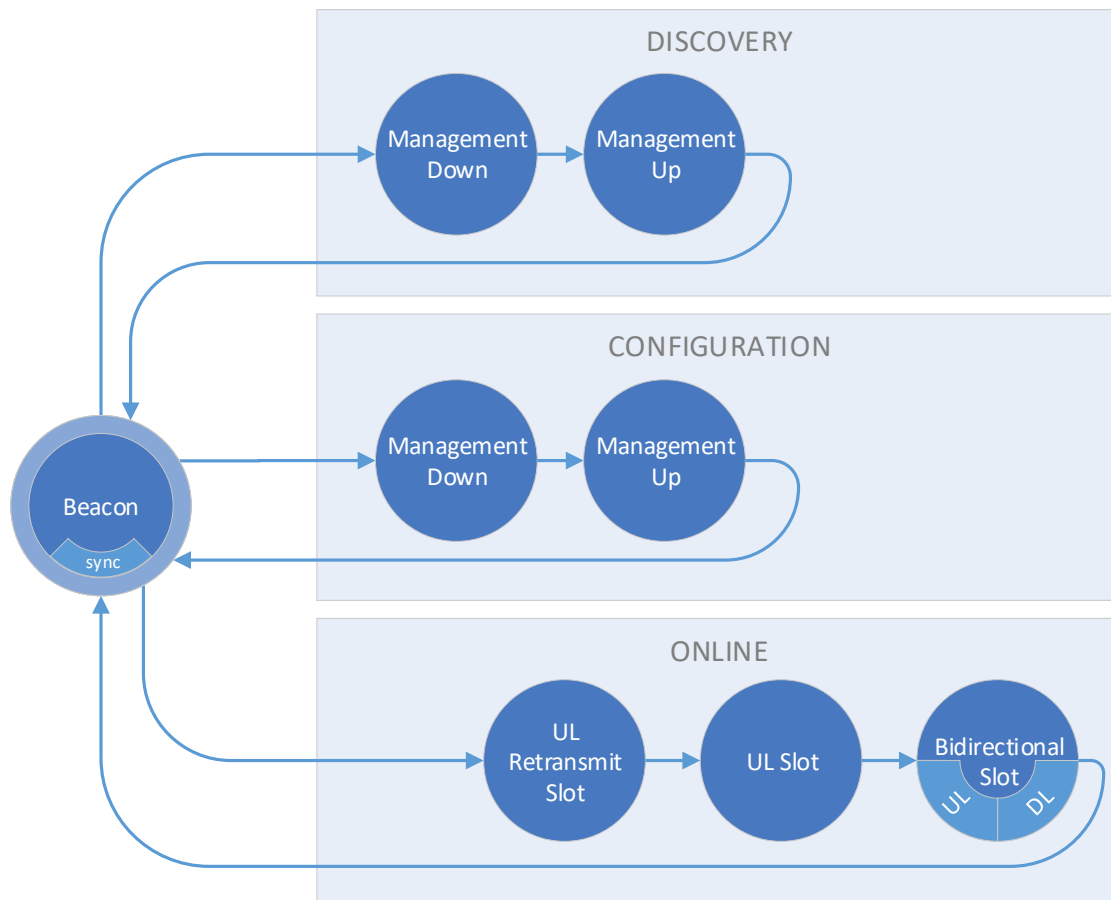


ABBILDUNG 4.4: Zustandsautomat des LLDN Managers

Abbildung 4.4 zeigt die möglichen Zustände im LLDN Manager. Der *Beacon* Zustand markiert den Start und ist je nach Gerätetyp entweder für die Vorbereitung eines Beacon Frames zur Aussendung oder zur Synchronisation vorgesehen. Im Fall des PAN Coordinators wird der Beacon mit der aktuellen Superframe Konfiguration und der Group ACK Bitmap für die Empfangsbestätigung aus dem letzten Superframe befüllt und in die Sendewarteschlange eingereiht. Ein teilnehmendes Gerät wird in diesem Zustand lediglich für einen Empfangsslot vorbereitet. Die anschließend weiter ausgeführte Slot Statemaschine auf der unteren Ebene sendet den vorbereiteten Beacon oder schaltet den Transceiver auf Empfang, erkennt im empfangenen Paket den Beacon und synchronisiert sich mit diesem.

Obwohl die Discovery und Configuration Phasen für diese Arbeit nicht implementiert

sind, sind die Grundzustände für eine etwaige Erweiterung bereits vorgesehen. Den entsprechenden Management Timeslots sind wiederum eigene Zustände im LLDN Manager zugewiesen. Im *Management Downlink* Zustand können dann Command Frames für die Konfiguration der Knoten durch den Coordinator vorbereitet oder durch normale Knoten empfangen werden. Umgekehrt verhält es sich im darauf folgenden *Management Uplink* Zustand.

In der Online Phase folgen zuerst Slots für die wiederholte Übertragung verlorengegangener Pakete. Im *UL Retransmit Slot* Zustand greift daher der Manager eines Netzwerkknoten auf die Group ACK Information des zuvor empfangenen Beacons zurück, um zu bestimmen, ob das im letzten Uplink Slot versendete Paket erneut gesendet werden muss. Der Coordinator erwartet in diesem Zustand die erneuten Übertragungen und ist im Gegensatz zu den normalen Knoten in allen Time Slots aktiv. Ähnlich verhält es sich im *UL Slot* Zustand. Der Coordinator erwartet hier die Übertragungen der anderen Knoten im Netz, leitet diese an die höheren Schichten im Stack weiter und befüllt seine Group ACK Bitmap für den nächsten Superframe. Ein sendender Knoten wird in den meisten dieser Slots wieder schlafen gelegt bis der für ihn konfigurierte Uplink Slot erreicht ist. Der Manager signalisiert daraufhin an die Slot State Machine, dass es sich um einen Sendeslot handelt und Daten aus der Warteschlange übertragen werden können.

Auch der *Bidirectional Slot* Zustand ist bereits vorgesehen, obgleich er im Entwurf für das konkrete Projekt ausgeschlossen wurde. Da das Transmission Direction Bit vom Coordinator in jedem Superframe bereits gewechselt und mit dem Beacon ausgesendet wird, lässt sich die Funktionalität in einer Erweiterung einfacher hinzufügen. Dabei besteht lediglich ein Unterschied zwischen Coordinator und normalen Knoten bei der Wahl zwischen Sende- und Empfangslot in der Slot State Machine. Zusätzlich muss hier der Sonderfall des ACK Frames implementiert werden, der nur von Knoten in bidirektionalen Slots gesendet wird, die Nachrichten vom Coordinator empfangen haben und diese bestätigen wollen.

Unterhalb des LLDN Managers (siehe [4.1 auf Seite 57](#)) sorgt das LLDN Modul für die Verwaltung der einzelnen Slots. Ist für einen Slot Sende- oder Empfangsaktivität nötig, wird die LLDN Schicht angestoßen und durchläuft das in [4.5 auf der nächsten Seite](#) dargestellte Diagramm. Da es sich hier um eine strikte Abfolge handelt, für die die Zeitkonstanten sehr wichtig sind, bietet sich die Darstellung in einem Chronogramm an. Es ist auch ein wichtiges Hilfsmittel für das Debugging, da sich Zustandsänderungen mit den strikten Zeitvorgaben oft nur sinnvoll über das Schalten von GPIO Pins beobachten lassen. Wie auch der LLDN Manager ist es als Zustandsautomat realisiert. Die Zustandsübergänge werden durch zwei Arten von Interrupts ausgelöst. Zum einen durch Hardware Timer, die zu bestimmten Phasen im Slot auslösen und durch Interrupts des

Transceivers, wenn dieser entweder ein Paket empfangen hat oder der Sendevorgang eines Paketes abgeschlossen ist.

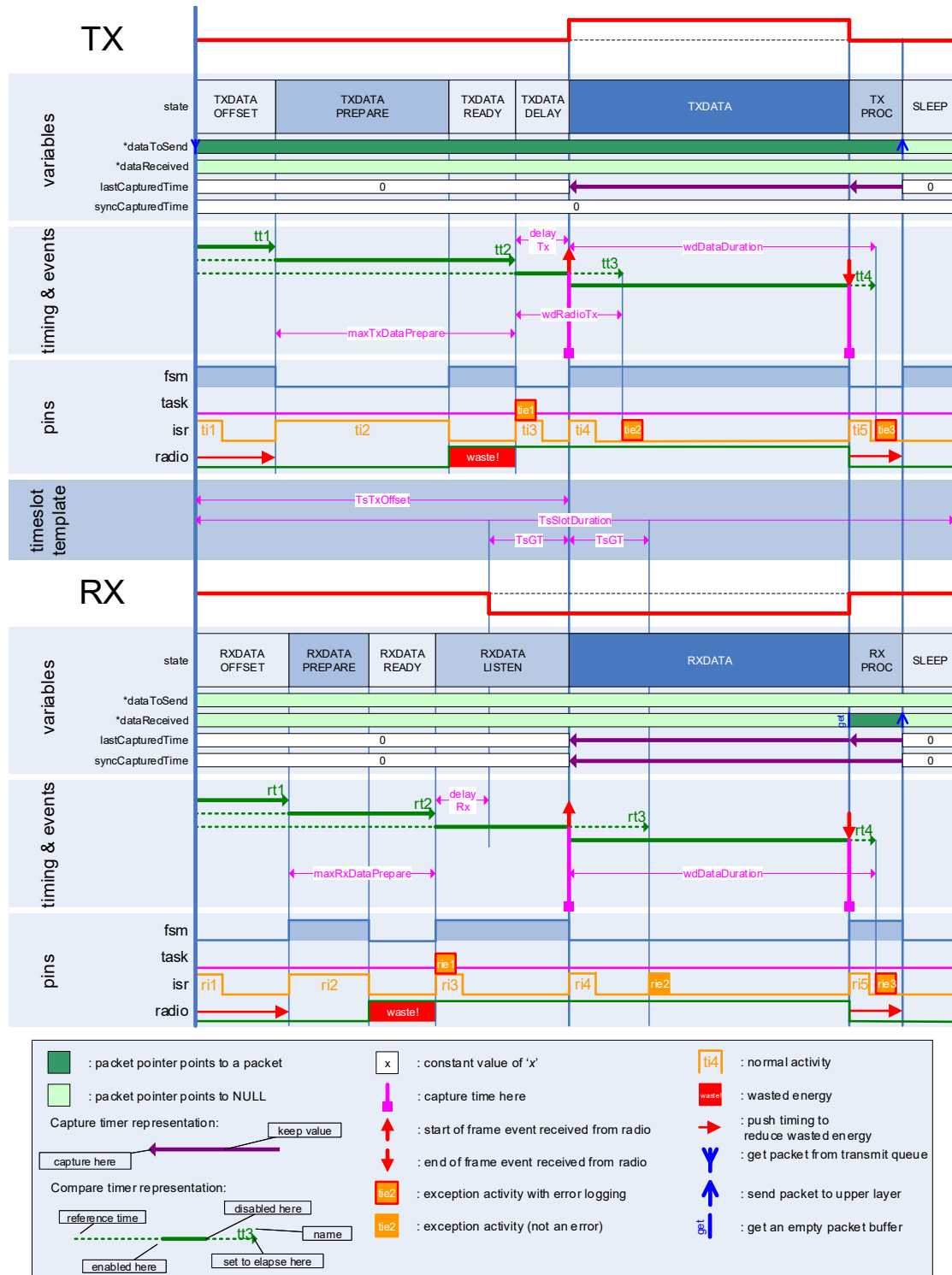


ABBILDUNG 4.5: Chronogramm des LLDN Timeslot Zustandsautomaten (adaptiert nach [Tho12a])

In 4.5 auf der vorherigen Seite sind die einzelnen Phasen innerhalb eines Slots als Chronogramm dargestellt. Eine solche Unterteilung der Betriebsphasen ist nötig, um die strengen Zeitvorgaben einzuhalten. Die mit *TXDATA* und *RXDATA* markierten Phasen stellen dabei die eigentliche Übertragung von Daten dar und müssen frei gehalten werden von anderen Aktivitäten, die diese Startzeitpunkte verzögern können. In der Mitte des Diagramms ist mit *timeslot template* die Gesamtzeit des Slots ($T_{\text{slotDuration}}$) mit den wichtigsten Zeiten aufgetragen. Die Zeit bis zum eigentlichen Aussenden eines Datenpaketes (T_{TxOffset}) wird dabei von den Phasen *TXDATA OFFSET*, *PREPARE*, *READY* und *DELAY* eingenommen. In *TXDATA OFFSET* wird entschieden, welches Paket in diesem Slot gesendet werden muss. Anschließend wird der Timer $tt1$ aufgezo-gen. Er dient dazu, die folgenden Phasen weiter zu verzögern, denn in der *READY* Phase wird der Transceiver aktiviert und verbraucht sehr viel mehr Energie als im vorhergehenden Schlafmodus, auch wenn er noch nicht aktiv sendet. Läuft der Timer aus, wird in Interrupt $ti2$ die Phase *TXDATA PREPARE* ausgeführt und das vorbereitete Paket aus der Warteschlange per SPI in den Transceiver Puffer übertragen. Zu Beginn wird der Timer $tt2$ gestartet. Läuft er aus, wird das Kommando zum Senden des Paketes gegeben. Der Fehlerfall $tie1$ kann im Interrupt auftreten, wenn das Vorbereiten des Paketes noch nicht abgeschlossen ist. Andernfalls wird in $ti3$ mit der Phase *TXDATA DELAY* das Sendekommando gegeben. Es wird um den festen Wert delayTX verzögert, der die Zeit angibt, die zwischen Absetzen des Sendekommandos und dem Verlassen der SFD an der Antenne des Transceivers liegt, damit der eigentliche Sendezeitpunkt möglichst optimal getroffen wird. wdRadioTx ist dabei die Zeit, in der der Transceiver mit dem Senden begonnen haben muss. Läuft der zu ihr gehörende Timer $tt3$ ab wird der Fehlerinterrupt $tie2$ ausgelöst und der Slot beendet. Bei normaler Funktion wird im $ti4$ Interrupt dieser Timer abgebrochen und der Rest des Paketes kann in *TXDATA* übertragen werden. Zusätzlich wird der letzte Timer $tt4$ gestartet. Läuft dieser aus, ohne in $ti5$ abgebrochen worden zu sein, hat die Übertragung zu lange gedauert und der Slot wird beendet. Wird $ti5$ jedoch im Gutfall durch den End-of-Frame Interrupt des Transceivers ausgelöst, ist die Übertragung abgeschlossen und kann in *TXPROC* an die höheren Schichten gemeldet werden. Anschließend kann das System für den Rest des Slots in den Schlafmodus versetzt werden. In diesem befindet es sich, bis der übergeordnete Slot Timer zum nächsten Slot auslöst, in dem dieser Knoten aktiv sein muss. Im Sonderfall des Coordinators ist dies direkt der folgende Slot.

Die Empfangsseite läuft ähnlich wie die Sendeseite ab. So wird in *RXDATA OFFSET* die Konfiguration vorgenommen und bis zur aktiven Phase abgewartet. In der *PREPARE* und *READY* Phase muss lediglich der Transceiver auf Empfang gestellt sein. Der Fehlerzustand $rie1$ wird hier nur für den Fall ausgelöst, dass dies nicht erfolgreich war. In *RXDATA LISTEN* wird nun der Empfang nach der Wartezeit delayRx aktiviert, die

analog zu `delayTx` die Verzögerung zwischen Absetzen des Empfangskommandos und der eigentlichen Empfangsbereitschaft angibt. Wie in *timeslot template* zu sehen, beginnt der Empfang zum `tsGT` Fenster und damit früher als die Aussendung auf der Sendeseite. Denn obwohl sich die Knoten mit jedem Paket, das sie vom Coordinator empfangen synchronisieren, driften die internen Uhren wenige $\mu\text{s/s}$ auseinander. Um Pakete dennoch sicher empfangen zu können, wird der Empfänger also um `tsGT` früher aktiviert. Ist nach zweimaliger `tsGT` kein Paket empfangen worden, wird im Fehlerzustand der Slot beendet. Es muss sich nicht um einen Fehler handeln, denn es könnte schlicht kein Paket gesendet worden sein. Ist jedoch ein Paket empfangen worden wird dieses in *RX PROC* verarbeitet. Handelt es sich um einen Data Frame, wird dieser an die höheren Schichten gemeldet. Verwaltungsdaten wie Beacons oder ACKs werden hingegen nur innerhalb des MACs verarbeitet.

Das Chronogramm stellt den Ablauf eines Slots im synchronisierten Zustand dar. Ist ein Knoten nicht synchronisiert, läuft ein einfacherer Prozess ab. Dazu wird er in die Phase *SYNC LISTEN* versetzt und aktiviert den Empfänger durchgehend bis ein Beacon empfangen wurde. Anschließend wird dieser in *SYNC RX* zur Berechnung der neuen Slotzeiten verwendet. Der Knoten geht anschließend gemäß seiner Konfiguration wieder in den Schlafmodus bis zum nächsten aktiven Slot über.

4.3.4.2 Implementierungsdetails

Im folgenden sollen einige Details der Implementierung näher betrachtet werden, die neben der im vorigen Abschnitt beschriebenen Funktion der Zustandsautomaten eine wichtige Rolle bei der Umsetzung gespielt haben.

Eine wichtige Grundlage für den korrekten Ablauf der Zustandsautomaten auf Knoten und Coordinator ist die Synchronisation im Netz. Alle Knoten müssen sich auf den gleichen Startzeitpunkt eines Superframes einigen, damit sichergestellt ist, dass jeweils der richtige Knoten seine Funkaktivität im ihm zugewiesenen Slot durchführt. Wird dies nicht korrekt verwaltet, können sich Teilnehmer gegenseitig stören. Da in LLDNs auch die Identität eines sendenden Knoten durch den Slot bestimmt wird, in dem er sendet, wäre im Fehlerfall auch diese Information verloren. Haben sich die Knoten einmal auf die Zeit des Superframes synchronisiert, muss diese Synchronisation aufrecht erhalten werden. Neben kleinen zeitlichen Abweichungen, wie sie durch unterschiedliche Interrupts auf den Mikrocontrollern vorkommen können, spielt die Zeitdrift der taktgebenden Quarzoszillatoren eine wichtige Rolle. Typische Quarze, die zur Ableitung der Systemfrequenz eingesetzt werden, haben eine spezifizierte Abweichung von 10 ppm^4 , also können sie

⁴parts per million

mit $10 \mu\text{s/s}$ driften. Die maximale Abweichung zwischen zwei Knoten kann damit $20 \mu\text{s}$ betragen. Gibt es einen größeren Temperaturunterschied zwischen beiden Quarzen kann dieser Wert jedoch auch ein Vielfaches davon betragen.

Da es sich beim LLDN um eine einfache Stern Topologie handelt, gibt hier immer der Coordinator durch Aussendung der Beacons die zentrale Zeit vor. Im Gegensatz zu anderen WSN Implementierungen müssen in LLDNs alle Knoten die Beacons empfangen. Zum einen, da diese den aktuellen Übertragungsstatus (Discovery, Configuration, Online) enthalten, zum Anderen, da im Beacon das Group ACK Feld enthalten ist, das die Knoten benötigen, um den Empfang ihrer Übertragungen aus dem vorherigen Superframe zu bestätigen. In der Implementierung wird so sichergestellt, dass alle Teilnehmer regelmäßig gezwungen sind, ein Paket zu empfangen und so eine Korrektur ihrer Synchronisation durchzuführen.

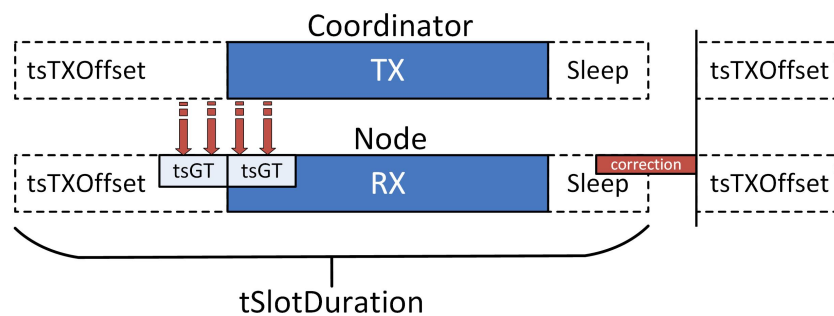


ABBILDUNG 4.6: Zeitsynchronisierung bei Empfang eines Paketes

Abbildung 4.6 zeigt, wie die Synchronisierung bei Empfang eines Paketes abläuft. Jedes Paket, das ein Knoten vom Coordinator empfängt, wird dabei zur Synchronisation verwendet, wobei der Empfang des Beacons dafür sorgt, dass sich ein Knoten auf jeden Fall einmal pro Superframe synchronisieren kann. Bei Start eines neuen Slots wird direkt ein Timer mit $t_{\text{SlotDuration}}$ gestellt, der einen Interrupt zu Beginn des nächsten Slots zur Folge hat. Wird nun von einem Knoten entweder der Beacon oder ein beliebiger anderer, gültiger LL-Frame empfangen, wird diese Zeit angepasst. Sowohl sendender als auch empfangender Knoten warten zunächst den $t_{\text{sTXOffset}}$ ab, der wie in 4.5 auf Seite 72 dargestellt, zur Vorbereitung genutzt wird. Der empfangende Knoten nutzt eine *Guard Time* (t_{sGT}), die sowohl vor als auch nach dem berechneten idealen Startzeitpunkt der Übertragung liegt. Die Funkhardware wird also früher aktiviert und es wird eine Pufferzeit gewartet, in der die Übertragung starten kann. Die Differenz zwischen erwartetem und tatsächlichem Übertragungszeitpunkt wird bei Verarbeitung des empfangenen Paketes genutzt, um den Auslösezeitpunkt des zu Beginn aktivierten Timers für den nächsten Slot um den Korrekturwert anzupassen.

Die zentrale Datenstruktur, in der die zu sendenden und empfangenen Pakete vorgehalten werden ist die *OpenQueue*. Die enthält eine fest vorgegebene Anzahl an *OpenQueueElement* genannten Einträgen, die genau ein Paket darstellen. Ein solches Element besteht aus einem bis zu 127 Byte großen Payload Feld und Pointern, die Beginn des Feldes, des Headers, usw. anzeigen. Mit verschiedenen Paketfunktionen kann Platz in der Nutzlast reserviert werden und die Pointer und Längfelder werden entsprechend angepasst. Neben der Verwaltung der Nutzlast gibt es viele Felder für die verschiedenen Schichten, sodass wichtige Daten aus den Headern, wie z.B. Adressen, erhalten bleiben, wenn das Paket den Stack hinauf gereicht wird und die Felder aus der Nutzlast gelesen werden. Neu hinzugefügt wurde hier die Möglichkeit einer 8 Bit Short ID, wie sie im LLDN Anwendung findet. Sie wird aus der MAC Adresse des Transceivers abgeleitet. Außerdem implementiert ein Element ein Eigentümer und Ersteller System. Die Komponente im System, die ein Element z.B. zum Versenden reserviert wird der Ersteller des Paketes und kann so immer verfolgt werden. Ist eine Komponente mit der Bearbeitung des Paketes fertig, setzt sie das Eigentümer Feld auf die jeweils nächste Komponente im Stack, die für die weitere Verarbeitung zuständig ist. Die LLDN Implementierung ist z.B. in eine übergeordnete LLDN Manager Schicht und eine untergeordnete LLDN Schicht zur Verwaltung eines Slots unterteilt. Hat der LLDN Manager bspw. die Befüllung eines zu sendenden Beacons abgeschlossen, setzt dieser das Eigentümer Feld auf die untergeordnete Schicht. Beginnt die Zustandsmaschine auf der Slotebene einen neuen Slot in Sendekonfiguration, wird sie das nächste Paket aus der *OpenQueue* auswählen, dessen Eigentümer sie nun ist. Wird ein Paket nicht mehr benötigt, weil es z.B. durch ein ACK bestätigt wurde und nicht mehr für den Retransmit vorgehalten werden muss, werden diese Felder zurückgesetzt und es kann von einer anderen Komponente neu reserviert werden.

Das Eigentümer System bettet sich in die Verwendung des RTOS Schedulers ein. Um Ressourcen möglichst schnell wieder abzugeben, werden die jeweils nächsten Schichten im Stack nicht einfach aufgerufen. Stattdessen wird ein neuer Task im System mit einer entsprechenden Benachrichtigungsfunktion angelegt. Stehen wichtigere Aufgaben an, können diese vom Scheduler vorgezogen werden. Wird eine solche Funktion einer bestimmten Schicht durch den Scheduler aufgerufen behandelt diese dann die ihr zugewiesenen Pakete aus der *OpenQueue*. Zudem gibt es eine Anzahl an reservierten Elementen in der Warteschlange, die nur von den niedrigsten Schichten reserviert werden können, falls der Rest der Elemente belegt ist. So wird verhindert, dass Netzwerkverwaltungsaufgaben nicht mehr wahrgenommen werden können, wenn das Paketaufkommen höher als der Durchsatz ist. Bspw. muss für den Empfang des Beacons immer ein Feld reservierbar sein. Kann für diesen kein Platz gefunden werden, wird das Paket verworfen und die Behandlung der ACKs findet nicht statt. Dadurch kann der Empfang anderer Pakete nicht

bestätigt werden und diese verbleiben länger in der Warteschlange für Retransmits, die nicht nötig wären.

Da eine Configuration Phase nicht implementiert ist, findet keine dynamische Verwaltung von Knoten auf dem PAN Coordinator statt. Stattdessen ist die Konfiguration fest kodiert und basiert auf der MAC Adresse der einzelnen Knoten. Die nötigen Parameter werden dabei in einer Konfigurationsdatenstruktur im LLDN Manager vorgehalten. So können die Knoten bestimmen, in welchen Slots sie aktiv sein müssen und welche Slotlänge einzuhalten ist. Der Coordinator hält dabei eine Zuordnung der 8 Bit short ID zum zugewiesenen Slot vor. Empfängt er Daten in einem Slot kann er so direkt die entsprechende Absenderadresse im `OpenQueueElement` ausfüllen. Damit können die höheren Schichten dann das Paket identifizieren und zuordnen.

Werden zukünftig die bidirectional Timeslots implementiert, muss ein Knoten zwischen zwei zu versendenden Paketen auswählen und bestimmen, ob diese im Uplink oder bidirectional Slot gesendet werden sollen. Hierzu ist bereits ein Feld im `OpenQueueElement` vorgesehen, das für diese Unterscheidung genutzt werden kann. Da das Verhalten zu dieser Unterscheidung im Standard nicht geregelt ist, muss eine höhere Schicht entscheiden, wie die Daten gesendet werden sollen. Dies kann z.B. auf Basis des Paketaufkommens entschieden werden.

Um den LLDN MAC mit der vorherigen TSCH Implementierung vergleichen zu können, wird im Großteil der Evaluation eine Slotlänge von 15 ms beibehalten. Um die maximale Leistung bei Verwendung der wesentlich schnelleren UWB Transceiver zu erreichen, kann die Mindestlänge eines Slots nach [AX14, S. 5570] berechnet werden:

$$t_{TS} = \frac{(p \cdot sp + pr + (m + n) \cdot sm + IFS)}{v}$$

wobei p = Anzahl Oktette im PHY Header,

pr = Anzahl Symbole in PHY Präambel und SFD,

sp = Anzahl Symbole pro Oktett im PHY Header,

m = Oktette an MAC Overhead, Header und Footer,

n = Maximal erwartete Nutzlast in Oktetten,

sm = Anzahl von Symbolen pro Oktett in der PSDU,

$$IFS = \begin{cases} \text{macSIFSPeriod} & \text{wenn } m + n \leq a\text{MaxSIFSFrameSize} \text{ oder} \\ \text{macLIFSPeriod} & \text{wenn } m + n > a\text{MaxSIFSFrameSize}, \end{cases}$$

v = Symbolrate

Dabei ist zusätzlich der Parameter p_r eingefügt worden, um die Präambel des UWB Transceivers mit aufzunehmen. Diese, inklusive des SFD wird in Symbolen berechnet. Für die Einbeziehung des Inter Frame Spacings gibt es zwei mögliche Fälle. Nach [IEE16, S. 61] kann es hier eine kurze oder lange Periode abhängig von der Konstanten $aMaxSifsFrameSize$ geben. Der Standard gibt hierfür 18 Oktette einer MPDU an. Beim geplanten Einsatz liegen Frames häufig über dieser Grenze, weshalb im Folgenden von einem Long Inter Frame Spacing ausgegangen wird. Für UWB liegt diese bei 40 Symbolen. Setzt man die weiteren Werte ein, so erhält man für die Slotlänge:

$$t_{TS} = \frac{(2,6250 \cdot 8S + 136S + (40 + 1230) \cdot 9,14S + 40S)}{7,78MSym/s}$$

$$= 1358S$$

Die Anzahl der Symbole pro Oktett wird beeinflusst z.B. durch die Reed Solomon Kodierung im PHY. Insgesamt liegt die Symbolrate mit 6,81 MBit/s Übertragungsrate bei 7,78 MSym/s, da alle 330 Bit 48 Bit an Reed Solomon Paritätsbits eingefügt werden. Hinzu kommen in dieser Konfiguration die Präambel mit 128 Symbolen und 8 Symbolen SFD. Bei insgesamt 1358 Symbolen folgt mit einer Symbolrate von 7,78 MSym/s eine minimale Timeslot Zeit von 175 μ s [Dec15, S. 195]. Dies gilt für einen voll ausgenutzten MAC Frame und für diese spezielle Konfiguration. Der DW1000 UWB Transceiver bietet noch weitere Modi mit niedrigeren Übertragungsraten, um die Reichweite zu erhöhen. In diesen Fällen verändert sich die Länge des Slots entsprechend. Der Parameter n , der auch im Beacon versendet wird, kann dabei verwendet werden, um die Slotlänge auf das notwendige Minimum für eine spezifische Anwendung einzuschränken.

Werden zukünftig Shared Group Timeslots eingesetzt, muss ALOHA statt CSMA/CA verwendet werden. Bei einer Auslastung der Luftschnittstelle von unter 18 % erlaubt das Verfahren eine Packet Delivery Ratio von 97 % [Dec15, vgl. S. 196]. Durch die hohe Geschwindigkeit von bis zu 6,8 Mb/s kann ein voller 127 Byte Standard 802.15.4 Frame in ca. 250 μ s übertragen werden. Bei einer Slotlänge von 15ms lassen sich so etwa 60 Knoten pro Slot unterbringen. Bei 18 % Auslastung können so 10 Knoten in einem Slot über das ALOHA Verfahren zugreifen und erreichen trotzdem eine Zustellungsrate von 97 % der Pakete.

4.4 Kapitelzusammenfassung

In diesem Kapitel wurde die Umsetzung der in Kapitel 2 auf Seite 14 ermittelten Anforderungen beschrieben. Zuerst wurde dazu eine Technologieauswahl durchgeführt, die die vorgestellten Erweiterungen des 802.15.4e im Hinblick auf ihre Einsetzbarkeit in Kombination mit UWB und unter dem gegebenen Szenario bewertet. Es hat sich gezeigt, dass die *Low Latency Deterministic Network* Erweiterung hier am Besten für eine Umsetzung geeignet ist. Zudem wurde *openWSN* als Echtzeitbetriebssystem ausgewählt, das bereits einen Netzwerkstack für die Integration liefert. Die bestehende Schichtenarchitektur ließ sich entsprechend einfach anpassen, um den zu entwickelnden LLDN MAC aufzunehmen. In der Implementierungsphase wurden neben dem MAC Layer selbst auch die Module zur Unterstützung der neuen Hardware entwickelt, sowie eine Simulationsumgebung, die unterstützend in der Implementierung Verwendung fand.

Mit Abschluss dieses Kapitels können nun weitere der eingangs gestellten Forschungsfragen beantwortet werden:

Welche vorhandenen MAC Erweiterungen des 802.15.4 sind mit IR-UWB sinnvoll nutzbar? Die Wahl fiel hierbei auf die LLDN Erweiterung der Standarderweiterung. Grundsätzlich bietet sich die einfache Stern Topologie eines LLDN Netzes für die Verwendung auf Raumfahrzeugen an, da hier keine Mesh Netzwerke benötigt werden. Kernfunktionalität der Alternativen TSCH und DSME ist ihre Fähigkeit des Channel Hoppings. Durch die Verwendung von UWB wird das Ziel dieser Funktion nahezu überflüssig. Abschließend lässt LLDN durch die zu erwartende geringe Latenz im Vergleich zu den Alternativen eine bessere Einsetzbarkeit bei kritischen Regelungsaufgaben erwarten.

Wie lässt sich die Lösung in bestehende Protokollstacks integrieren? OpenWSN wurde als Betriebssystem ausgewählt, da es eine hohe Robustheit bei gleichzeitig modernem Protokollstack verspricht. In die vorhandene Schichtenarchitektur ließ sich die neue MAC Schicht durch die Verwendung der vom System zur Verfügung gestellten abstrahierten Schnittstellen integrieren. So wurde bspw. die Abstraktionsschicht für die neue Funkhardware implementiert, die weiterhin eine Nutzung mit bereits vorhandenen Transceivern erlaubt.

Welche Probleme treten bei IR-UWB auf? Wie können diese ggf. durch MAC Schicht Erweiterungen gelöst werden? Eines der Hauptprobleme des verwendeten UWB Transceivers ist der im Vergleich hohe Stromverbrauch. Der Empfang der UWB Signale auf Seite des Analogfrontends kann nicht so energiesparend wie bei üblichen trägerwellenbasierten Verfahren durchgeführt werden, weshalb das Modul im Empfang sogar einen

höheren Verbrauch aufweist als im Sendebetrieb. Der LLDN MAC schafft hier Abhilfe, indem durch die besonders kurzen Header sowohl die Sende- als auch die Empfangszeiten kurz gehalten werden. Durch das Group ACK Feature kann die Aktivitätszeit der Transceiver weiter verringert werden. Ein weiteres Problem stellt das nicht vorhandene Carrier Sensing dar, das in der nächsten Frage gesondert behandelt wird.

Wie lässt sich das in den MAC Layern verwendete CSMA/CA Verfahren mit UWB nutzen? Das UWB Übertragungsverfahren besitzt prinzipbedingt keine Möglichkeit, ein CSMA/CA Verfahren durchzuführen, da es hier keinen Träger gibt, dessen Präsenz gemessen werden kann, um festzustellen, ob weitere Sender aktuell den Kanal belegen. Mit den DW1000 Transceivern ist an dieser Stelle also für zukünftig zu implementierende Shared Group Timeslots nur das ALOHA Verfahren sinnvoll einsetzbar. Durch die hohe Übertragungsrate von bis zu 6,8 Mb/s lassen sich auch in sehr kurzen Timeslots mehrere Geräte unterbringen. Dabei sollte für diese spezielle Hardware eine Auslastung des Mediums von 18 % nicht überschritten werden.

Das folgende Kapitel widmet sich der Evaluation der hier implementierten Lösung und liefert Antworten auf die restlichen Forschungsfragen, die sich mit der Leistungsfähigkeit des Systems befassen.

Evaluation

Die hier durchgeführten Evaluationsszenarien vergleichen einige Key Performance Indikatoren, wie sie für drahtlose Netzwerke oft erhoben werden. Da mit der gewählten openWSN Infrastruktur bereits eine weitere Implementierung des ebenfalls aus 802.15.4e stammenden TSCH MAC Layers vorhanden ist, bietet sich ein Vergleich mit diesem an. Sowohl LLDN, als auch TSCH verfolgen das Ziel, einen robusten MAC Layer auch für die Anwendung in rauen Umgebungen zu schaffen.

Neben den in [A auf Seite 98](#) durchgeführten Unittests stellen die hier vorgenommenen Leistungsmessungen einen Systemtest dar. Hier wird die komplette Kette aller Komponenten sowohl unter Normalbedingungen, als auch unter Vollast in Kombination genutzt, wodurch u.U. weitere Fehler aufgedeckt werden können.

Für die Tests wird grundsätzlich ein Netz aus drei Knoten verwendet. Ein Knoten stellt dabei den Wurzelknoten für die Zeitsynchronisierung und absolute Slotnummer im TSCH Netz dar, bzw. nimmt die Aufgabe des PAN Coordinators im LLDN Netz wahr. Die weiteren Knoten liefern Frames mit maximal möglicher Payload an diesen verwaltenden Knoten. Die Längen der Superframes/Slotframes setzen sich aus atomaren Zeitkonstanten zusammen, die prinzipbedingt für beide Verfahren abweichen. Da in einem TSCH Netz jede Übertragung bestätigt wird, setzt sich die Länge jeder Zelle zusätzlich aus einer Wartezeit nach dem Senden des Paketes und der Zeit für das Empfangen des ACKs zusammen.

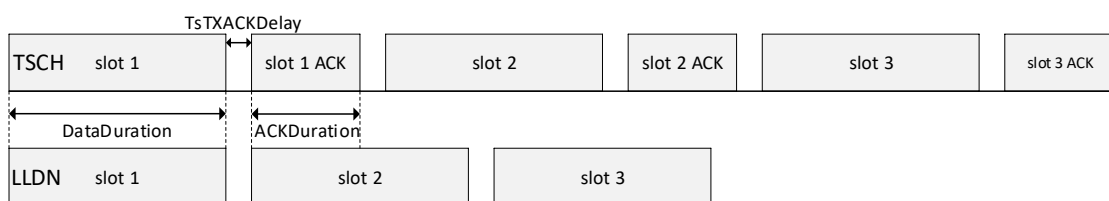


ABBILDUNG 5.1: Gegenüberstellung der Frame Zeiten mit (TSCH) und ohne (LLDN) nötiges ACK

Die Gesamtlänge eines solchen Slotframes ergibt sich mit n Geräten bei einer TX Dauer von T_d mit ACK Delay T_{dly} und Dauer des ACK von T_{ack} zu 5.1. Im worst-case Fall eines LLDN Superframes, der die gleiche Anzahl n an Uplink und bidirektionalen Timeslots aufweist und durch r Retransmit Slots begleitet wird, ergibt sich die Gesamtlänge des LLDN Superframes wie in 5.2. Trotz fehlen der ACKs ist ein LLDN Superframe in diesem Fall also immer länger als sein TSCH Pendant. LLDN bietet damit aber auch eine

theoretisch 50 % höhere Übertragungsrate durch die abwechselnd in up- und downlink Konfiguration genutzten bidirektionalen Slots.

$$\sum_{i=0}^n (T_d + T_{dly} + T_{ack}) \quad (5.1)$$

$$r \cdot T_d + \sum_{i=0}^n (2(T_d + T_{dly})) \quad (5.2)$$

Da bidirektionale Slots für diese Arbeit nicht implementiert wurden und um die Vergleichbarkeit durch eine gleiche Übertragungskapazität zu wahren, werden sie in der Konfiguration nicht vorgesehen, womit der LLDN Superframe kürzer ist, als der TSCH Slotframe.

Um andere Engpässe im Timing, bspw. durch die Ausführung von Managementaufgaben im 6top Layer des TSCH Netzes auszuschließen wurde die Basisslotlänge auf 10 ms festgelegt. So ist sichergestellt, dass die gemessenen Leistungsdaten auf die MAC Layer selbst zurückzuführen sind.

5.1 Packet Delivery Ratio

Eine wichtige Kenngröße für drahtlose Sensornetzwerke ist die Packet Delivery Ratio. Sie gibt das Verhältnis von gesendeten zu erfolgreich empfangenen Datenpaketen an. Interessant sind diese Werte insbesondere bei LLDN Netzen, da sie im vgl. zu TSCH oder einem klassischen 802.15.4 Netz keine variable Anzahl an Retransmits haben. In der Standardkonfiguration gibt es sogar nur einen erneuten Übertragungsversuch, der mit zusätzlichem Group ACK Frame in der Mitte des Superframes auf ggf. zwei erhöht werden kann. Durch die Übertragung der ACKs im Beacon ist auch die genaue Synchronisierung auf den Superframe wichtig. Wacht ein Knoten durch falsches Timing oder Drift der Uhren zum falschen Zeitpunkt auf, wird der Beacon u.U. nicht empfangen, womit auch die Bestätigung für das gesendete Paket im vorherigen Frame verloren ist. Die PDR kann sich bei nur einem Neuübertragungsversuch unter diesem Aspekt erheblich verschlechtern. Die Robustheit der physischen Übertragungsebene durch UWB sollte hier jedoch zumindest bei fehlerhaften Übertragungen durch Störungen ausgleichend wirken.

Getestet wurde die PDR sowohl in einem TSCH wie auch einem LLDN Netz. Dabei wurden Pakete maximaler Größe von 127 Byte mit einer Frequenz von 10 Hz in unterschiedlichen Entfernungen gesendet. Um eine ausreichende Datenbasis zu haben, sind 1000 Pakete pro Test versendet worden. Damit erstreckt sich ein Test auf über 1,5

min, wodurch eine ausreichend lange Messzeit gegeben ist, um auch Langzeiteffekte, z.B. durch thermische Drift der Quarze zu berücksichtigen. Sowohl auf dem Coordinator, als auch auf den sendenden Knoten sind dazu Daten über die versendeten Pakete, empfangene ACKs, sowie empfangene Pakete auf dem Coordinator aufgezeichnet worden. Zusätzlich wird die Anzahl der Desynchronisierungen ermittelt für den Fall, dass ein Knoten aufgrund dieses Zustandes die Nachrichten nicht empfangen konnte.

Test	Pakete empfangen	RTs empfangen	PDR %	davon RTs %
LLDN 1 m	1000	0	100,00 %	0,00 %
LLDN 10 m	930	54	98,40 %	5,40 %
LLDN 25 m	656	124	78,00 %	12,40 %
TSCH 1 m	1000	0	100,00 %	0,00 %
TSCH 10 m	898	61	95,90 %	6,10 %
TSCH 25 m	715	135	85,00 %	13,50 %

TABELLE 5.1: Packet Delivery Ratio – TSCH und LLDN im Vergleich bei verschiedenen Entfernungen

Tabelle 5.1 zeigt die Ergebnisse der Tests bei unterschiedlichen Entfernungen der Knoten voneinander. Es zeigt sich generell kein signifikanter Unterschied zwischen den beiden MAC Layern in Bezug auf die PDR. Entgegen der Vermutung, dass die höhere Anzahl von erneuten Übertragungsversuchen bei Verwendung von TSCH insbesondere bei größerer Entfernung zu einer besseren PDR führt, hat sich nicht bestätigt. Allgemein kann gesagt werden, dass die PDR damit zum großen Teil durch die Wahl von UWB im PHY Layer bestimmt ist und nicht oder nur minimal auf den MAC zurückgeführt werden kann.

Zu beachten ist, dass TSCH bei Nutzung von UWB kein Channel Hopping durchführt. Dies ließe sich erreichen, indem die unterschiedlichen Kanäle in Kombination mit den verschiedenen parallel betreibbaren Pulse Repetition Frequencies genutzt würden. Durch das extrem große abgedeckte Spektrum all dieser Kanäle und die PRF sind sehr unterschiedliche Ausbreitungsbedingungen möglich. Ein Kanal mit niedrigerer Mittenfrequenz neigt dabei dazu, eine höhere Reichweite zu ermöglichen [Dec15, S. 195], was die Anzahl verlorener Pakete durch ein zu schwaches Signal reduzieren kann.

5.2 Latenzverhalten

Eine der Haupteigenschaften, die ein LLDN ausmacht, ist die theoretisch niedrige Latenz, die zum einen durch das Ersetzen von typischen ACKs durch zusammengefasste Group ACKs erreicht wird und zum anderen Resultat der stark eingeschränkten Header

der Datenframes ist, die im günstigsten Fall nur ein Byte lang sind. Durch die Sterntopologie und die starre Konfiguration des Superframes sind diese Latenzen zusätzlich deterministisch und können damit bei Berechnungen berücksichtigt werden.

Wichtig in diesem Zusammenhang ist die Definition des Begriffes der Latenz. Während in typischen Computernetzwerken die Round Trip Latenz eines Paketes gemessen wird, also die Zeit, die ein Paket vom Absender zum Empfänger und wieder zurück benötigt, bietet es sich bei Sensornetzwerken wie LLDN an, die einfache Latenz von einem Sensor zum Coordinator zu messen. Für Regelungsaufgaben, die bspw. auf an den Coordinator angeschlossenen Systemen durchgeführt werden, ist diese Latenz wichtiger als ein Round Trip, auch da Sensor und evtl. zu steuernder Aktor oft nicht auf demselben Knoten liegen.

Die tatsächlich erreichbare Latenz hängt zudem davon ab, wann ein Paket an den MAC Layer übergeben wurde. Sowohl in einem TSCH Netz als auch im LLDN wird eine Folge von Slots durchlaufen und ein Slot bestimmt, in dem der eigene Knoten senden darf. Im worst-case Fall kann die Latenz also fast die Länge des gesamten Superframes betragen, wenn ein Paket in die Warteschlange kurz nach der letzten Sendemöglichkeit eingereicht wird.

Die Tests, die für diesen Abschnitt durchgeführt wurden, messen die einfache Latenz von einem Knoten zum Coordinator sowohl für ein TSCH Netz, als auch für ein LLDN. Beide Netze agieren grundsätzlich unterschiedlich in der Verteilung der Slots, bspw. sind im LLDN nach Beginn des Superframes zuerst die Retransmit Slots vorgesehen. Würde ein Test die Pakete daher zu einem fixen Zeitpunkt relativ zum Start des Superframes senden, wären die Ergebnisse nicht aussagekräftig. Stattdessen wacht die implementierte Testprozedur in zufälligen Zeitintervallen innerhalb eines Superframes einmal auf und reiht ein Paket in die Sendewarteschlange ein. Beide Netze sind dabei so konfiguriert, dass der Superframe trotz der unterschiedlichen Funktionsweise jeweils immer 60 ms lang ist. Durch die bei TSCH nötigen ACKs bedeutet das, dass der Superframe lediglich 4 Slots lang ist, wohingegen bei LLDN 6 Slots zur Verfügung stehen. Gemessen werden die Zeiten ab dem Zeitpunkt, zu dem sie auf der Sendeseite eingereicht wurden bis zu dem Zeitpunkt, an dem sie im Empfänger die MAC Schicht verlassen haben. Dazu wird die Offset Zeit zum Beginn des Superframes sowohl beim Senden, als auch beim Empfangen des Paketes auf den Knoten ausgegeben. Anschließend können die Ausgaben kombiniert und die Latenzen berechnet werden. Die Abweichung der Quarze zwischen den beiden Knoten liegt bei wenigen $\mu\text{s/s}$ und ist daher für diese Betrachtung vernachlässigbar. Verwendet wurden für den Test 1000 Pakete pro Netz.

Kapitel 5.2 auf der nächsten Seite zeigt, dass die durchschnittliche Latenz in dieser Konfiguration bei LLDN um ca. 5,478 ms unter der des TSCH Netzes liegt. Im TSCH Netz muss für jedes Paket ein ACK gesendet werden und die kumulierten Warte-, Empfangs-

und Verarbeitungszeiten tragen daher zu dieser Abweichung bei. Beheben ließe sich diese in der TSCH Implementierung, wenn die empfangenen Daten bereits vor Senden des ACKs an die nächsthöhere Schicht zur Verarbeitung weitergereicht würden. Dann würden, zumindest bei zufälliger Verteilung der Paketeinreihungen, LLDN und TSCH gleichauf liegen.

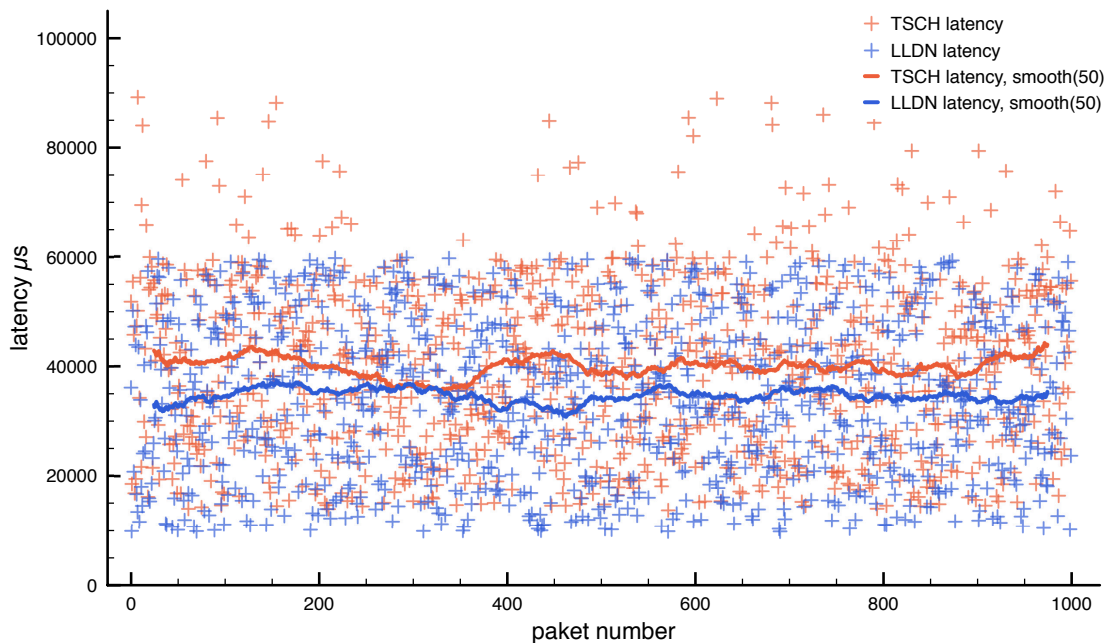


ABBILDUNG 5.2: Latenz pro Paket bei zufälligem Zeitpunkt der Einreihung in die Sendewarteschlange für TSCH und LLDN

Durch einige Ausreißer in den TSCH Ergebnissen lässt sich außerdem erkennen, dass das System keine deterministische Latenz garantieren kann. Von anderen Knoten, die nicht an der Kommunikation beteiligt sind, werden in gewissen Abständen keep-alive Nachrichten gesendet, um die Synchronisation mit dem Netz nicht zu verlieren. In wenigen Fällen hat dies offensichtlich zur Folge, dass vom Wurzelknoten die vorgegebene Zellenreihenfolge geändert wird und sich die Latenz über die Länge des Superframes erhöht. Die Standardabweichung der Latenz liegt dadurch bei TSCH um $1380 \mu\text{s}$ über der von LLDN.

In einem weiteren Test wurde die minimal mögliche Latenz in einem LLDN ermittelt. Durch die fest vorgegebene Struktur des Superframes erlaubt LLDN es schließlich, dass mit dem Ermitteln der Sensordaten bis kurz vor Beginn des eigenen Uplink Slots gewartet werden kann, um so maximal frische Daten versenden zu können. Zusätzlich wurden die Slot Parameter angepasst, da hierfür keine Vergleichbarkeit mit TSCH nötig ist. Die UWB Transceiver erlauben bei 127 B Nutzlast bei maximaler Geschwindigkeit eine minimale Übertragungszeit von $\approx 225 \mu\text{s}$ laut Datenblatt [Dec15, S. 197]. Neben der reinen Übertragungszeit von $\approx 150 \mu\text{s}$ kommen noch die Präambel und PHY Header dazu, die

im Fall des DW1000 UWB Transceivers eine abweichende Symbolrate besitzen. Zu der reinen Sendezeit kommt noch die interne Verarbeitung im Transceiver hinzu. Hier wurde eine Zeit von $507 \mu\text{s}$ insgesamt gemessen. Zusätzlich muss aber auch die Verarbeitungszeit auf dem Mikrocontroller betrachtet werden. Der verwendete STM32L151 benötigt mit seinen 32 MHz einige Zeit zur Vorbereitung der Pakete beim Senden und beim Verarbeiten auf Empfangsseite. Insbesondere, wenn die Sicherheitsfeatures des MAC Layers Verwendung finden und die Daten verschlüsselt werden, wird signifikant Rechenzeit verbraucht. Wie in 4.3.4.1 auf Seite 69 bereits beschrieben, sind hierzu bestimmte Zustände im Zustandsautomaten der Slotverwaltung vorgesehen, die über Timer begrenzt sind. So braucht die Vorbereitung eines Paketes (TXDATAPREPARE) bei maximaler Länge eines Paketes etwa $839 \mu\text{s}$ und die Verarbeitung bei Empfang maximal $792 \mu\text{s}$. Enthalten sind hier sowohl die Verarbeitungszeiten der Pakete, als auch die SPI Kommunikation mit dem Transceiver. Um sicherzustellen, dass die Vorbereitungsphase auf jeden Fall abgeschlossen wird und z.B. kleinere Interrupts nicht zum Versenden eines unvollständigen Paketes führen, wurde die Vorbereitungszeit auf 1 ms festgelegt. Nach dieser Zeit läuft der entsprechende Timer aus und das Kommando zum Start der Übertragung wird an den Transceiver gesendet. Die Produktion der Testdaten ist per Timer auf 1 ms vor der eigentlichen Slotzeit festgesetzt worden, sodass zeitnah ein Paket zur Verfügung steht. Mit entsprechender Pufferzeit ergibt sich eine Slotlänge von 4 ms. Der Empfang kann im Idealfall bei vollen 127 Byte Frames etwa 3,3 ms nach Beginn des Slots erfolgen.

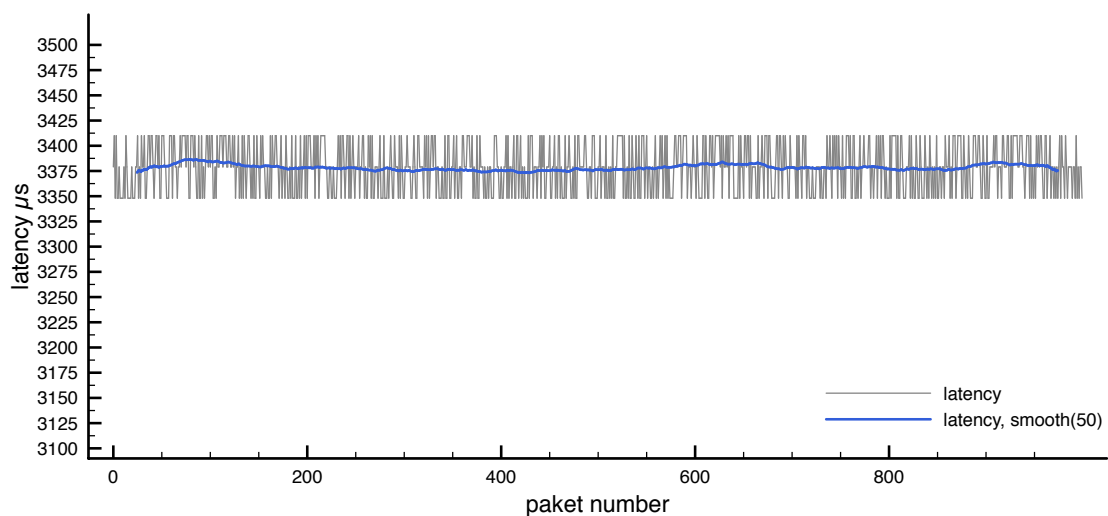


ABBILDUNG 5.3: Latenz in einem LLDN bei Optimierung aller Parameter und Einreihung der Pakete in Synchronisation mit den Uplink Slots

Abbildung 5.3 zeigt die so erreichbare Latenz bei Verwendung von 1000 Paketen ab dem Zeitpunkt der Einreihung in die Warteschlange bis zur Verarbeitung auf dem Coordinator. Die Werte zeigen Sprünge zwischen drei Werten, da die Auflösung der Timer lediglich ca. $31 \mu\text{s}$ beträgt. Die durchschnittliche Latenz liegt so bei $3,378 \text{ ms}$. Bei geringerer

Datenmenge, wie sie bei reinen Sensoren häufiger vorkommt, ließe sich die Latenz hier noch um ca. 150 μ s verkürzen, wenn z.B. nur 4 Byte Nutzdaten verwendet würden. Den Hauptanteil an der Latenz hat aber weiterhin die Verarbeitungszeit auf dem Controller. Die geforderte Latenz von unter 8 ms nach REQ-04 lässt sich aber trotzdem leicht erfüllen. Nur durch die Wahl von schnelleren Mikrocontrollern ließe sich an dieser Stelle eine noch geringere Latenz erreichen.

5.3 Durchsatz

Mehr noch als die Latenz ist besonders der Durchsatz für beide MACs von der Konfiguration des Superframes abhängig. Um hier TSCH und LLDN ansatzweise vergleichen zu können, wird die Konfiguration aus [5.2 auf Seite 83](#) verwendet, sodass wieder gleichlange Superframes von 60 ms zur Verfügung stehen. Gemessen wurde der Durchsatz von reinen Nutzdaten zwischen zwei Knoten durch Anwendungscode, der dafür sorgt, dass die Sendewarteschlange konstant gefüllt ist. Sowohl TSCH, als auch LLDN nutzen hierbei jeweils einen Slot pro Superframe zur Übertragung der Daten. Limitierende Faktoren sollten in diesem Test also hauptsächlich durch den Unterschied im Overhead zwischen beiden Verfahren bestehen.

In einem zweiten Test wurde der mögliche Maximaldurchsatz von LLDN mit optimierten Parametern getestet. Obwohl hier nun die minimale Sendezeit für ein Paket von 225 μ s genutzt wird, kann durch die festen Vorbereitungs- und Verarbeitungszeiten nur eine minimale Slotlänge von 3325 μ s erreicht werden. Geht man von der vorgesehenen Netzwerkkonfiguration von WICome aus, werden vier Knoten benötigt. Sieht man nun neben dem Beacon zwei Slots für Retransmits, vier für den Uplink und nochmals vier für die bidirektionale Kommunikation vor, so ergibt sich eine realistische Superframelänge von 36,575 ms.

Für die Tests wurde zusätzlich sichergestellt, dass keine Retransmits stattgefunden haben, damit das Ergebnis des reinen Durchsatzvermögens nicht verfälscht wird.

Tabelle [5.2 auf der nächsten Seite](#) zeigt die Ergebnisse der Messungen. Grundsätzlich lässt sich feststellen, dass die verwendeten TDMA Verfahren die Bandbreite erheblich schmälern. Der verwendete DW1000 Transceiver ist in der Lage eine Übertragungsrate von 850 kB/s ohne die Verwendung dieser MAC Layer zu erreichen. Ein sehr leichter Vorteil in der Vergleichskonfiguration lässt sich wie erwartet bei LLDN erkennen. Ein LL-Data Frame nutzt im Vergleich zu seinem TSCH Pendant nur 1 Byte an Header Daten. Bei beiden kommt noch die CRC Prüfsumme im Footer von 2 Byte hinzu.

Netzart	Superframelänge	Nutzdaten	Durchsatz
LLDN	60 ms	124 B	2,018 kB/s
TSCH	60 ms	120 B	1,953 kB/s
LLDN optimiert	36,575 ms	124 B	3,311 kB/s

TABELLE 5.2: Vergleich des Durchsatzes verschiedener Netzkonfigurationen

TSCH und LLDN sind in dieser Hinsicht schlecht vergleichbar. Der Superframe hat in der verglichenen Konfiguration weniger Slots und besäße bei mehreren aktiven Knoten einen geringeren Gesamtdurchsatz. TSCH ist im Vergleich zu LLDN aber flexibler. Nicht nur können Knoten in ihren Links auf verschiedenen Kanälen miteinander reden, sodass der kumulierte Gesamtdurchsatz im Netz wesentlich höher ist, durch die Konfiguration mehrerer Slotframes kann bestimmten Geräten mehr Sendezeit zugewiesen werden als anderen.

Trotz der relativ hohen Datenrate der UWB Transceiver kann festgehalten werden, dass der Einsatz von TDMA Verfahren in Kombination mit Low Power Mikrocontrollern zwar für den Einsatz in Sensornetzwerken geeignet ist und auch die eingangs gestellte Anforderung REQ-03 erfüllt, aber insbesondere Nutzlasten, die eine sehr hohe Datenrate benötigen, weiterhin andere Übertragungsverfahren nutzen müssen.

5.4 Energieeffizienz

Drahtlose Sensorknoten erreichen ihren niedrigen Energieverbrauch hauptsächlich durch Verwendung sehr kurzer Arbeitsphasen. Ist keine Aktivität erforderlich schalten sowohl Transceiver, als auch Mikrocontroller in einen Tiefschlaf- bzw. Stop Modus, der oft mit wenigen μA auskommt. Sind bspw. nur Daten im Minutentakt gefordert, kann das System für wenige Millisekunden aufwachen, die Daten aufbereiten, versenden und für den Großteil der Minute wieder in den Tiefschlaf übergehen.

Für sporadische Aussendungen in einem ALOHA Betriebsmodus ist dies ggf. ausreichend. Bei Verwendung eines MAC Layers, der eine Synchronisierung der Knoten voraussetzt kann der Tiefschlaf nur bei längerer Inaktivität Verwendung finden. Nach dem Aufwachen aus einem solchen Zustand müssen oft Taktgeber neu initialisiert werden, die wiederum auf das Einschwingen von Quarzoszillatoren warten müssen. Dieser Vorgang kann mehrere Millisekunden in Anspruch nehmen, wodurch der Einsatz von Tiefschlafzuständen bei den eng getakteten Superframes im LLDN nicht möglich ist. Der verwendete DW1000 Transceiver unterscheidet zwischen einem solchen Deepsleep Modus und einem Idle Modus, in dem er sich befindet, wenn keine Sende- oder Empfangsaufgaben anstehen. Im

Vergleich zum Deepsleep mit 50 nA verbraucht der Transceiver im Idle Modus laut Datenblatt 19 mA. Für kritische Anwendungen, in denen der Netzwerkknoten ständig mit dem Netz synchronisiert bleiben muss, ist dies also die Grundlast, die in Kauf genommen werden muss.

Ist für längere Zeit keine Datenübertragung nötig, kann der Knoten auch bei Einsatz von LLDN in den Tiefschlaf übergehen. Dieser muss sich nach dem Aufwachen aber erneut mit dem Netz synchronisieren, was im worst-case Fall die Dauer eines ganzen Superframes erfordern kann. Möglich ist dies, da LLDN keine Verbindung der Knoten überwacht und folglich vom Coordinator z.B. auch keine keep-alive Nachrichten eingefordert werden.

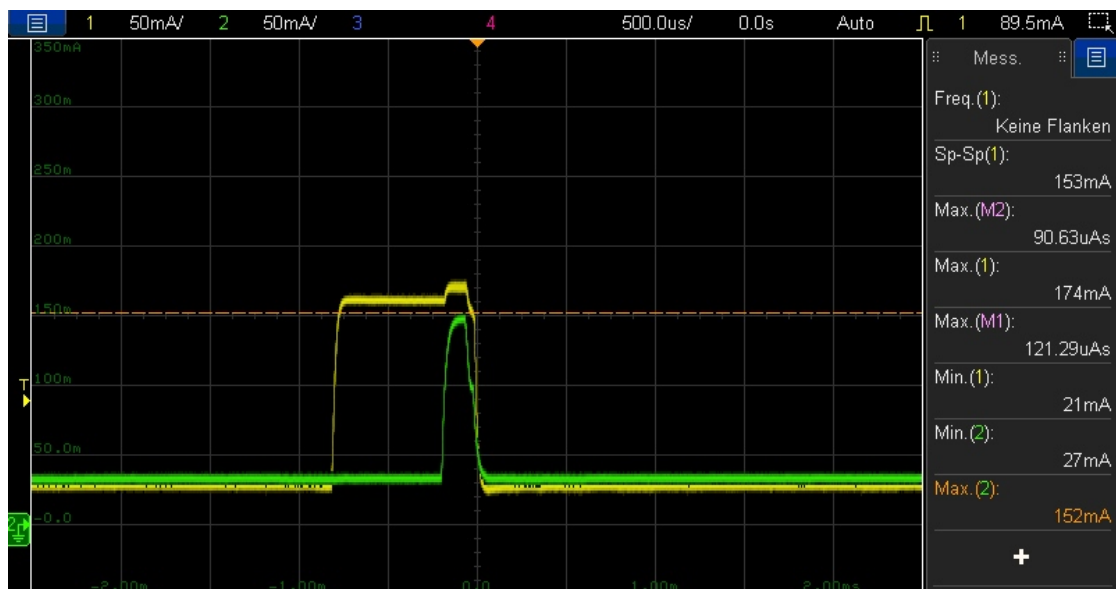


ABBILDUNG 5.4: Energieverbrauch der Transceiver eines Knotens (gelb) und des Coordinators (grün) während eines Beacon Slots

Abbildung 5.4 zeigt den Stromverbrauch in einem typischen Slot. Zu sehen ist deutlich, dass das UWB Modul während des Empfangs sogar einen höheren Verbrauch aufweist, als das sendende Modul. Auch zu sehen ist die Guard Time, in der der Empfänger bereits beginnt zu horchen, bevor die eigentliche Übertragung beginnt.

Abbildung 5.5 auf der nächsten Seite zeigt den Verbrauch eines Knotens über den Verlauf eines Superframes mit einer Transmission im Uplink Slot. Zusätzlich wird im zugeordneten bidirektionalen Slot auf eine Aussendung des Coordinators gewartet. Im integrierten Verbrauch ist zu erkennen, dass die eigentlichen Sende und Empfangszeiten nur vergleichsweise geringen Einfluss auf den Verbrauch haben. Der Grundverbrauch im Idle Modus wiegt hier über den gesamten Superframe gesehen schwerer.

Der Verbrauch pro gesendetem Paket liegt damit maximal im Bereich von 1,1 mAs (3,63 J) für einen Standardknoten. Im Coordinator ist der Verbrauch entsprechend höher, da

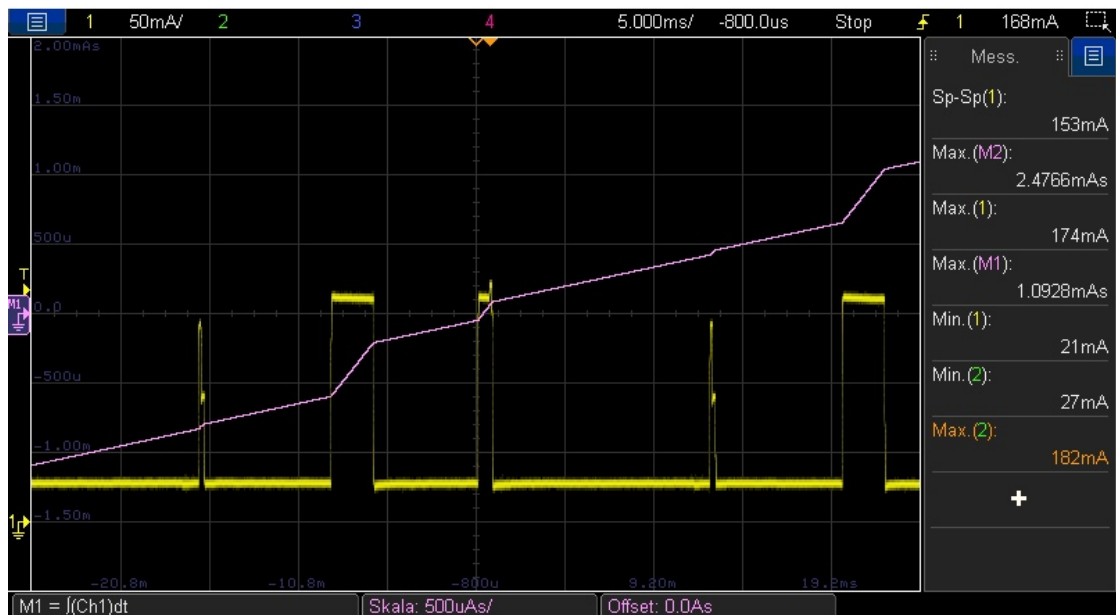


ABBILDUNG 5.5: Energieverbrauch des Transceivers eines Knotens während eines gesamten Superframes mit aktivem Uplink Slot (gelb). In lila der integrierte Verbrauch über den Verlauf des Superframes.

er in allen Slots auf potentielle Nachrichten warten und entsprechend lange im Empfangsmodus ist. Deshalb liegt bei diesem der Verbrauch bei etwa 2,5 mAs (8,25 J) pro Superframe. Finden generell mehr Übertragungen statt, muss nicht die gesamte Guard Time gewartet werden und die Übertragung ist früher beendet. In einem voll ausgelasteten Netz würde der Coordinator also weniger Energie verbrauchen, als in einem mit geringerer Auslastung.

5.5 Kapitelzusammenfassung

Der entwickelte LLDN MAC Layer wurde unter Berücksichtigung der unterschiedlichen Superframelängen in seinen Key-Performance Indikatoren mit der vorhandenen TSCH Implementierung von openWSN verglichen, sowie hinsichtlich der maximal möglichen Leistung untersucht.

Sowohl in der Packet Delivery Ratio als auch im Durchsatz zeigt sich keine signifikante Abweichung der beiden verglichenen MAC Layer voneinander, wenn die Betriebsparameter so gewählt werden, dass beide im gleichen Zeitrahmen operieren. Die Latenz im System hingegen lässt sich durch Einsatz von LLDN mit seinen Group ACKs und geringen Headergrößen stark verbessern. Außerdem ist sie im Vergleich zum TSCH Netz deterministisch. Der Energieverbrauch, insbesondere der Knoten erlaubt einen sehr energieeffizienten Betrieb bei lediglich 1,1 mAs pro Paket.

Mit den ermittelten Ergebnissen können die restlichen Forschungsfragen nun beantwortet werden:

Wie wirkt sich die Verwendung eines alternativen MAC Betriebsmodus auf den Energieverbrauch des Systems aus? Da mit Verwendung eines 802.15.4 MAC Layers immer auch eine Synchronisierung der Knoten einhergeht, muss ein Großteil der Energie des Systems dazu eingesetzt werden, diese Synchronisierung aufrecht zu erhalten. Im Fall des hier verwendeten Transceivers können z.B. die Schlafmodi deshalb nicht benutzt werden. Dadurch, dass die Transceiver sich für den Großteil des Superframes aber im Leerlauf befinden und die Kommunikation durch die Synchronisierung ansonsten kollisionsfrei ablaufen kann, erreichen sie einen sehr niedrigen Verbrauch pro Paket.

Welche Verbesserung der Robustheit der Übertragung lässt sich erreichen? Die Untersuchungen haben gezeigt, dass die Robustheit der Datenübertragung in erster Linie vom verwendeten UWB Physical Layer abhängt. Wird hier, anders als für den Test eine niedrigere Übertragungsrates bei gleichzeitig längerer Präambeldauer verwendet, lässt sich die Reichweite und PDR für andere Einsatzszenarien noch weiter erhöhen. LLDN hat prinzipbedingt einen Vorteil gegenüber z.B. dem TSCH Layer, da es keine aufwändigen Verwaltungsaufgaben zur Aufrechterhaltung des Netzes durchführen muss. So werden unnötige Neusynchronisierungen vermieden und eine deterministische Latenz geboten.

Welchen Einfluss hat die MAC Implementierung auf die Latenz? Im Vergleich mit TSCH als Vertreter typischer 802.15.4 MAC Layer zeigt LLDN eine deutlich niedrigere Latenz, insbesondere durch das Fehlen des typischen ACK Zyklus und die Auslagerung der Bestätigungen in die Group ACKs des Beacons. Durch die Verkleinerung des MAC Headers

auf 1 Byte im Idealfall sinkt die Latenz weiter. Es müssen nicht wie in z.B. in TSCH IEs im Header transportiert werden, um die Netzverwaltung zu sichern. Festgehalten werden muss weiterhin, dass ein Großteil der Latenz nicht durch die Transceiver oder die Funktionsweise des MACs verursacht wird, sondern durch die langen Verarbeitungszeiten auf den verwendeten Low-Power Mikrocontrollern.

Welchen Einfluss hat die MAC Implementierung auf den Datendurchsatz? Der Datendurchsatz im LLDN, ähnlich wie auch im vergleichbar konfigurierten TSCH Netz ist grundsätzlich sehr gering und hängt stark von der Anzahl der Knoten im Netz ab, ist aber für Sensornetze dennoch ausreichend. Der reine Durchsatz im LLDN ist durch die kleineren Header der Datenframes geringfügig größer. Doch selbst bei maximaler Optimierung der Parameter ist auch in einem Netz mit wenigen Knoten nur eine Übertragungsrate von wenigen kB/s möglich.

Fazit

In dieser Arbeit wurde eine MAC Schicht für Ultrabreitband Funktechnologie entwickelt und auf ihre Anwendbarkeit zur Intra-Satellitenkommunikation hin untersucht. Dabei wurde eine Basisimplementierung mit geringer und deterministischer Latenz geschaffen, die auch den besonderen Anforderungen von kritischen Regelungssystemen genügt.

In den folgenden Abschnitten wird die Arbeit nun noch einmal zusammengefasst und es wird eine ausführliche Betrachtung möglicher Verbesserungen und Weiterentwicklungen gegeben.

6.1 Zusammenfassung

Der Einsatz von drahtlosen Sensor- und Aktuatornetzwerken stellt mit höherer Leistung der entsprechenden Transceiver und der fortschreitenden Miniaturisierung der Komponenten eine alternative zur gängigen kabelgebundenen Kommunikation zwischen Subsystemen auf Raumfahrzeugen dar. Insbesondere moderne, leistungsfähige Ultrabreitband Funkmodule bieten hier auf physischer Ebene sehr stabile Verbindungseigenschaften durch ihre Resistenz gegen Multipath-fading, das durch die vielen Metalloberflächen innerhalb eines Raumfahrzeuges ein besonderes Problem für herkömmliche Übertragungsarten darstellt.

Bei der Kommunikation von vielen Netzwerkknoten bietet sich der Einsatz einer Sicherungsschicht zur Koordination des Übertragungsverhaltens an. In dieser Arbeit wurde anhand diverser Forschungsfragen und einer Anforderungsanalyse für eine spezifische Satellitenmission eine MAC Schicht aus dem 802.15.4 Standard für die Kombination mit der physischen Schicht der Ultrabreitbandübertragung ausgewählt. Der Standard ist explizit für diese Art von von niedriggradigen WPANs geschaffen worden (LRWPAN), hat aber insbesondere in seiner ursprünglichen Definition starke Probleme in der zuverlässigen und damit zeitnahen Zustellung von Paketen. Aus diesem Grund wurden mit der der 802.15.4e Erweiterung des Standards eine Reihe von alternativen MAC Layern eingeführt, die diese Nachteile versuchen auszugleichen. Für diese Arbeit fiel die Wahl dabei auf die Low Latency Deterministic Network Spezifikation aus der Menge der verfügbaren Erweiterungen. LLDNs zeichnen sich durch ihre einfache Topologie mit sehr geringen Latenzen und einer deterministischen Zustellung der Pakete aus. Für die Ultrabreitbandtechnologie existierte bis jetzt keine Implementierung dieser Erweiterung.

Umgesetzt wurde die Implementierung dieses MAC Layers auf Mikrocontrollern der STM1151 Reihe, die sich durch ihren niedrigen Energieverbrauch bei geringer Taktrate auszeichnen. Die UWB Konnektivität wurde durch Anbindung an ein Decawave DW1000 Modul hergestellt. Um der Anforderung nach Integration in einen vollständigen Netzwerkstack Rechnung zu tragen ist der LLDN Layer in das gängige echtzeitfähige openWSN Betriebssystem integriert worden. Eine besondere Herausforderung hierbei war die Einhaltung der strikten Zeitvorgaben, die in einem solchen TDMA Verfahren beachtet werden müssen. Insbesondere die Synchronisierung der einzelnen Knoten auf einen Superframe muss sehr genau durchgeführt werden, damit das Netz lauffähig ist.

Das entwickelte System wurde bezüglich der wichtigsten Leistungsindikatoren evaluiert und teilweise mit dem Betrieb im TSCH Modus verglichen. Dabei hat sich gezeigt, dass die Packet Delivery Ratio in erster Linie vom verwendeten UWB PHY Layer bestimmt wird. Bei der Kommunikation innerhalb eines Raumfahrzeuges ist mit entsprechender Parametrisierung so eine minimale Anzahl an Paketverlusten zu erwarten. Der Durchsatz eines solchen Netzes liegt wie für TDMA Verfahren üblich weit unter der möglichen Rate, die die verwendeten Transceiver zur Verfügung stellen können, hängt aber bei allen Verfahren des 802.15.4 von der Anzahl der Knoten im Netz ab. Auch hier ist die Verwendung der UWB Technologie vorteilhaft, denn sie liefert mit 6,8 MBit/s eine deutliche höhere Übertragungsrate als die in diesem Bereich üblichen 250 kBit/s anderer Transceiver in den ISM Bändern. Durch Einsatz von LLDN lässt sich die Latenz im Vergleich signifikant reduzieren. Es muss aber darauf geachtet werden, dass die Produktion von Daten mit dem Lauf des Superframes synchronisiert wird, um die minimal mögliche Latenz zu erreichen. Generell trägt die Verarbeitungszeit auf den verwendeten Mikrocontrollern mit ihrer geringen Taktrate einen großen Anteil an der Gesamtlatenz. Im Energieverbrauch liegt ein Knoten mit etwa 3,6 Joule pro Paket im Superframe weit unterhalb der geforderten Grenzen für den Gesamtenergieverbrauch des Experimentes nach den Anforderungen. Der Verbrauch impliziert, dass der Knoten ständig mit dem Netz synchronisiert ist. Je nach Anwendung muss dies aber nicht nötig sein und hier könnte weitere Energie eingespart werden.

6.2 Ausblick

Das hier entwickelte System kann als sehr gute Basis für weitere Entwicklungen verwendet werden. Mit der Integration in die stark modularisierten Komponenten des openWSN Systems sind leicht Erweiterungen und Modifikationen am bestehenden LLDN MAC Layer möglich. Im Verlauf der Arbeit sind Verbesserungsmöglichkeiten im Bereich des

entwickelten Systems allgemein und in Bezug auf den verwendeten Standard und seine Anwendbarkeit in Raumfahrtprojekten identifiziert worden, die in den folgenden Abschnitten zusammenfasst sind.

6.2.1 System

Im entwickelten System gibt es einige Verbesserungsmöglichkeiten, die teilweise für den letztendlichen Einsatz auf einem Satelliten noch betrachtet werden müssten.

Einer dieser Faktoren betrifft mögliche Tiefschlafphasen eines Sensorknotens. Aktuell wacht ein Knoten zu jedem Beacon eines Superframes auf, um mit dem Netz synchronisiert zu bleiben. Bei Geräten, die keine hohe Updaterate benötigen kann jedoch die Verwendung des Tiefschlafmodus (Stop Mode) des Controllers sinnvoll sein. Hierbei müsste die Möglichkeit geschaffen werden, den MAC extern zu deaktivieren, sodass die entsprechenden Interrupts zurückgesetzt werden und etwaige nötige Daten in den Backupregistern vorgehalten werden. Vorteilhaft ist hier, dass der LLDN Coordinator keinen Verbindungsstatus für seine Knoten vorhält, sodass keine Reinitialisierung nötig ist.

In der Simulationsumgebung wäre für einen besser durchführbaren Unittest eine Push Funktionalität nötig. So könnten zeitrelevante Aspekte evtl. ebenfalls mit dieser Lösung getestet werden, ohne auf echte Hardware zurückzugreifen zu müssen.

Mit der Verwendung von openWSN wurde Wert auf eine kompatible Integration des LLDNs in dieses System gelegt. Dabei wurden auch Datenstrukturen wiederverwendet, die eine Vielzahl von Feldern enthalten, die lediglich für den Betrieb mit TSCH interessant sind, aber beibehalten werden müssen, weil sie in höheren Netzwerkschichten ebenfalls verwendet werden. Es müsste erörtert werden, ob eine Integration in den aktuellen Entwicklungsstand von openWSN vorteilhaft ist, um von zukünftigen Weiterentwicklungen profitieren zu können, oder ob eine eigenständige Lösung und damit einhergehende Verkleinerung der Codebasis hier sinnvoller ist.

Ein eher allgemeiner Punkt lässt sich unter dem Stichwort Language Based Security zusammenfassen. Eingebettete Systeme werden fast ausschließlich in C und C++ entwickelt, was u.A. mit den oft nur in diesen Sprachen verfügbaren Peripheriebibliotheken der Mikrocontrollerhersteller zusammenhängt. Diese Sprachen bieten z.B. mit ihrem ungefilterten Zugriff auf Speicherstrukturen zu viele Möglichkeiten Fehler zu machen und sind daher insbesondere beim Einsatz in sicherheitsrelevanten Systemen kritisch zu betrachten [SN18]. In den letzten Jahren haben Sprachen wie bspw. Rust hier Vorarbeit geleistet und zeigen, dass in Bezug auf Speichermanagement inherent sichere und komfortabel zu nutzende Sprachen keine Leistungseinbußen zur Folge haben müssen.

Beispielhaft sei die Embedded Working Group des Rust Projektes¹ genannt, die u.A. eine Hardwareabstraktionsschicht für eingebettete Systeme entwickelt.

Nicht Teil dieser Arbeit, aber trotzdem eines der wichtigsten Themen für den tatsächlichen Einsatz ist der Schutz vor Strahlungseffekten auf den Netzwerkknoten. Die stark erhöhte Dosisleistung im Weltraum kann nicht nur zum Ausfall von Knoten führen, sondern im ungünstigeren Fall lediglich einzelne Register, z.B. des Transceivers verändern. So kann z.B. der Kanal, auf dem der Coordinator operiert durch ein solches Ereignis gewechselt werden, womit das gesamte Netz offline ist. Hier müssen geeignete Verfahren gefunden werden, die fehlerhaft arbeitende Knoten deaktivieren und ggf. redundante Geräte hinzuschalten.

6.2.2 Standard

Die hier implementierte LLDN Spezifikation bietet eine gute Basis für etwaige Weiterentwicklungen zur weiteren Anpassung an den Einsatz in Satellitenmissionen. Grundsätzlich muss festgehalten werden, dass es vermutlich keine Weiterentwicklungen an LLDN selbst mehr geben wird. Die Spezifikation ist in neueren Versionen des 802.15.4 nicht mehr enthalten, da sich nach den ursprünglichen Entwicklern keine Betreuer hierfür finden konnten. Generell ist mit den Bemühungen der IETF um 6TiSCH eine Konzentration auf den TSCH Betriebsmodus festzustellen.

Auch LLDN hat einige Nachteile, die aufgrund dieser Entwicklung zumindest im 802.15.4 vermutlich nicht mehr behoben werden. Der im Vergleich zu den mehrkanalfähigen Spezifikationen geringe Durchsatz ist ein Kritikpunkt, der in der Literatur zu vielen weiteren Arbeiten geführt hat, wie z.B. der Erweiterung *Multichannel LLDN* [PAB14] oder einer Variante hiervon mit Nachrichtenpriorisierung [PL16]. Hinzu kommt die teilweise unzureichende Spezifikation bestimmter Frame Typen. So sind bspw. die Command Frames, die in der Management Phase im LLDN ausgetauscht werden müssen, um das Netz zu konfigurieren, nicht ausreichend spezifiziert. Die Reihenfolge und Bitpositionen der Felder sind hier nicht beschrieben. So ist ein interoperabler Betrieb zwischen verschiedenen Implementierungen nicht möglich.

Die Mehrkanalfähigkeit ist dabei eine der Erweiterungen, die auch für einen raumfahrtspezifischen MAC Layer interessant sind. In Kombination mit UWB sind z.B. neben den regulären Kanälen die verschiedenen Pulsraten, die parallel betrieben werden können, eine Möglichkeit viele Übertragungskanäle zu nutzen und so die verfügbare Bandbreite zu erhöhen. In einem solchen MAC könnte auch die Möglichkeit vorgesehen werden, bestimmten Sensoren mehr Zeit im Superframe zu geben, um so deren höheren Bedarf

¹<https://github.com/rust-lang-nursery/embedded-wg>

abzudecken. Bspw. könnten so Nutzlastdaten schneller übertragen werden, wohingegen Thermalsensoren, die eine niedrige Ausleserate haben entsprechend weniger Zeit zugewiesen bekommen. Durch genaue Kenntnis der Anwendungsdomäne lässt sich hier die Leerlaufzeit stark verkürzen.

Wenn der Standard so weiterentwickelt wird und damit inkompatibel zum LLDN wird, kann auch an anderer Stelle mit dem ursprünglichen Standard gebrochen werden. So haben die UWB Transceiver aufgrund ihrer hohen Geschwindigkeit im Vergleich zu klassischen 802.15.4 die Möglichkeit, größere Frames mit bis zu 1024 Byte statt der üblichen 127 Byte zu versenden. So kann Verarbeitungszeit auf den Controllern und insbesondere die bei UWB je nach Konfiguration besonders lange Präambelphase insgesamt verkürzt werden.

Anhang A

Verifikation

Die Evaluation in 5 auf Seite 81 kann als Systemtest des implementierten LLDN MAC Layers angesehen werden. Dabei werden die funktionalen und nicht-funktionalen Anteile auf Gesamtsystemebene mit verschiedenen Parametern und in unterschiedlichen Szenarien getestet. Eine solche Überprüfung reicht jedoch nicht aus, um eine zufriedenstellende Testüberdeckung zu erreichen. Insbesondere Fehlerfälle und Ausnahmen im Ablauf des MACs werden so nicht explizit getestet.

Dieser Anhang soll einen Überblick über die unternommenen Schritte zur Verifikation des Systems liefern. Da der MAC Layer eine besonders kritische Komponente im Netzwerkstack ist, ist ein gründlicher Test hier zwingend erforderlich. Ein fehlerhaft arbeitender MAC kann seinen Transceiver im schlimmsten Fall so falsch ansteuern, dass die Kommunikation auf dem genutzten Kanal für alle anderen Knoten in Reichweite unmöglich wird. Der LLDN Layer wurde daher in einem zustandsbasierten Unit Test einer Reihe verschiedener Testprozeduren unterworfen, um die Überdeckung der Zustände im Automaten der Slot Ebene zu prüfen. Zusätzlich wurde eine statische Codeanalyse durchgeführt, um weitere verdeckte Fehler in der Implementierung zu finden.

Zu den Anforderungen, die noch nicht im Evaluationsteil behandelt wurden gehört der Codefootprint. Dieser wurde gemessen und liegt mit der LLDN Implementierung inklusive des Betriebssystems bei 8,21 kB RAM und 35 kB Flash. Damit ist die Anforderung in REQ-16 ohne Weiteres erfüllt.

A.1 Unit Tests

Der Unit Test einer MAC Schicht erweist sich als schwierig, da hier für die korrekte Operation verschiedene Zeitparameter eingehalten werden müssen. Außerdem muss für die Kommunikation immer mindestens eine Gegenstelle vorhanden sein, die entsprechende Daten empfängt oder sendet. Fehlerfälle hingegen sind so äußerst schlecht abzubilden. Es würde eine eigens implementierte Gegenstelle erfordern, um z.B. die Funktion der Retransmits sicher testen zu können.

Das in 4.3.2 auf Seite 64 beschriebene Board Support Package, das in der Entwicklung benutzt wurde, könnte hier zur Lösung Verwendung finden, da hier relativ einfach die entsprechenden Manipulationsfunktionen ergänzt werden könnten. Der Nachteil liegt darin, dass eben nicht das echte BSP Verwendung findet und so z.B. keine genauen Aussagen bei Fehlern bezüglich des Zeitverhaltens gemacht werden können. Da, soweit möglich immer auf der echten Hardware getestet werden sollte ([Grü17, S. 109]), wurde der hier implementierte Unit Test direkt auf der Zielhardware durchgeführt.

Zur Anwendung kommt dabei μ CUnit¹ als Testframework für eingebettete Systeme. Es zeichnet sich durch seinen geringen Overhead und wenig benötigte Rechenleistung aus. Eine Funktion, die besonders bei den zeitlich kritischen Abläufen im LLDN von großer Bedeutung ist. Wichtig bei Testcode und den entsprechenden Frameworks, die auf Mikrocontrollern mit begrenzten Ressourcen operieren müssen, ist neben dem geringen Speicherverbrauch auch der Verzicht auf die dynamische Allokierung von Speicher.

Um alternative Ausführungspfade in der LLDN Schicht nun sicher auszuführen, wird das Prinzip der Fault Injection umgesetzt. Dabei werden über eine spezielle Schnittstelle Fehlermodi in das System eingepflanzt, die dann so zur Ausführung von Code führen, der im Normalfall z.B. nur bei fehlerhafter Kommunikation mit einem anderen Knoten zustande kommen würde und aufwändig auf der echten Hardware herbeigeführt werden müsste. Auch Fehler z.B. der Transceiverhardware können so simuliert werden, ohne eine für diesen Zweck viel zu aufwändige Hardware-in-the-loop Testumgebung aufzubauen, bei der die Transceiverschnittstelle durch eine speziell angepasste Hardwaresimulation ersetzt werden würde.

Im Fall des Zustandsautomaten auf der Slot Ebene ist eine solche Fehlerinjektion relativ einfach umzusetzen. Die Aktivierung für den Test erfolgt über einen Schalter zur Compilezeit, um den entsprechenden Code nicht in die produktive Firmware übernehmen zu müssen. Eine Modifikation von Code innerhalb des Zustandsautomaten ist nicht nötig, da sich alle Fehler extern auslösen lassen. Dies hat den Vorteil, dass der zu testende Code exakt so auch produktiv eingesetzt wird. Die meisten möglichen Fehlerzustände gehen auf den Transceiver zurück. Da dieser ein eigenes Modul zur Ansteuerung besitzt, lassen sich hier mittels Fault Injection z.B. eine Zeitdrift bei Paketempfang simulieren oder auch nötige Callbacks deaktivieren.

Die Herleitung der durchgeführten Testfälle ist im folgenden Abschnitt genauer beschrieben. Zusammen mit den in der Evaluation durchgeführten Tests decken diese nun auch alle durch Tests zu verifizierenden Anforderungen ab, die in der Anforderungsanalyse spezifiziert wurden.

¹<http://www.ucunit.org>

A.2 Testfallherleitung

Um das bestmögliche Verhältnis von eingesetztem Testaufwand zu erreichter Testabdeckung zu erreichen, wurden zustandsbasierte Tests [Grü17, S. 149-150] auf Basis des LLDN Zustandsautomaten auf Slot Ebene durchgeführt. Der Slot-Zustandsautomat steuert den gesamten Ablauf eines LLDN Slots. Sämtliche Fehler, z.B. in der Hardware-schnittstelle, aber auch in der Kommunikation mit anderen Knoten, werden hier erkannt und entsprechend verarbeitet. Eines der größten Risiken hierbei ist, dass der Zustandsautomat in einem Zustand verbleibt und sich nicht mehr aus diesem befreit. Dabei kommt die gesamte Kommunikation zum Erliegen. Handelt es sich bei dem betroffenen Knoten um den Coordinator, ist das gesamte Netz damit offline.

Abbildung A.1 auf der nächsten Seite zeigt den Zustandsautomaten, wie er für die Slotverwaltung Verwendung findet. Alle Kanten in diesem Graphen stellen einen Zustandsübergang dar und müssen daher in einem Testfall abgeprüft werden. Dabei ist der Gutfall relativ einfach zu testen, wenn die gesamte Kette der Zustände durchlaufen wird und der Slot regulär endet. Für Fehlerfälle findet aber die Fault Injection Anwendung, sodass die entsprechenden Pfade ausgeführt werden. Obwohl sämtliche Pfade immer in `endSlot()` landen, um den Slot korrekt zu beenden, lässt sich durch die Auslösung der seriellen Ausgaben mit entsprechendem Fehlercode genau feststellen, welcher Fehlerzustand ausgelöst wurde, sodass keine weiteren Eingriffe im Code des Zustandsautomaten nötig sind, um dies zu verifizieren.

Die erste Sonderrolle nimmt die Synchronisation auf den Superframe ein. Ein nicht synchronisierter Knoten befindet sich so lange im `SYNCLISTEN` Zustand, bis ein entsprechender Beacon empfangen wurde. Durch Fehlerinjektion im Radiomodul lässt sich eine leicht abweichende Empfangszeit vorgeben, die zu einer Drift führt und so einen Rückfall in den Synchronisierungszustand verursachen muss.

Die möglichen Fehlerzustände vom `READY` Zustand aus stellen sowohl für RX als auch TX das gleiche Problem dar: Die Vorbereitungszeit war zu kurz. Die Fault Injection verringert hierzu einfach die entsprechende Zeit, sodass der Fehler ausgelöst wird.

Der SFD Fehlerzustand auf Sendeseite tritt auf, wenn der Transceiver nicht rechtzeitig ein Interrupt auslöst, um zu signalisieren, dass der SFD versendet wurde. Auch hier kann durch Manipulation im Radiomodul der entsprechende Callback deaktiviert werden, sodass im Zustandsautomaten der entsprechende Timer abläuft und der Transceiverfehler angenommen wird. Ähnlich verhält es sich auf der Empfangsseite mit dem Timeout. Hier handelt es sich jedoch weniger um einen Fehler, sondern um den normalen Abbruch des Slots, wenn nichts empfangen wird. Dies ist auch regulär der Fall, z.B. im Coordinator, da

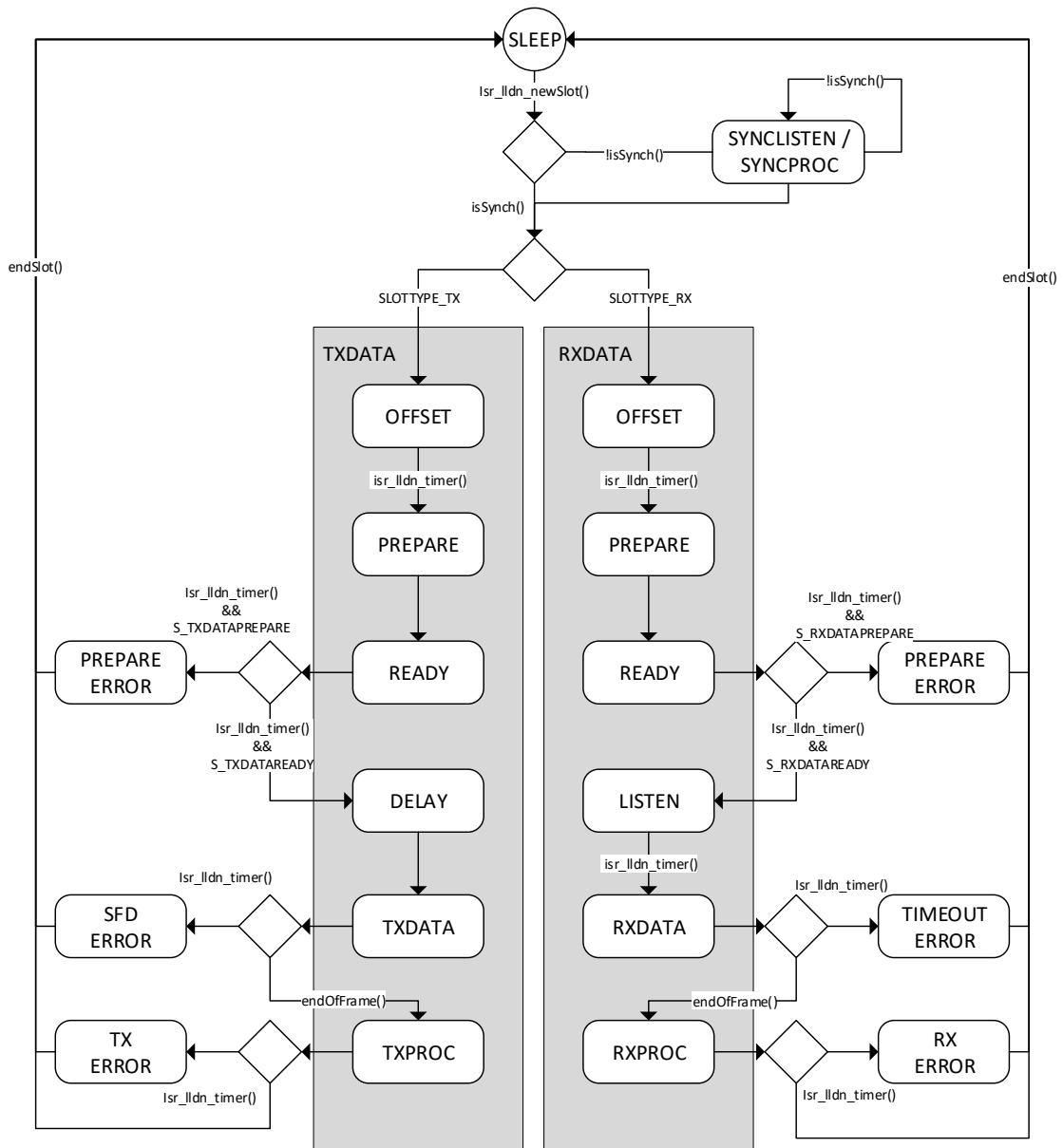


ABBILDUNG A.1: Zustandsautomat der Slotverwaltung für die Testfallherleitung

dieser auf jedem Slot hört. Hat ein Knoten keine Daten zum Senden vorliegen, wird der Slot nicht genutzt und der Empfang wird nach dem Timeout abgebrochen. Entsprechend ist zum Test auch keine Fehlerinjektion nötig.

Der letzte Fehler auf Sendeseite tritt auf, wenn der Interrupt zum Ende des Frames nicht rechtzeitig ausgelöst wird, z.B. weil der Frame zu lang ist oder ein Fehler im Transceiver vorliegt. Dieser Fehler lässt sich wieder durch die Deaktivierung des entsprechenden Callbacks auslösen und prüfen. Die Empfangsseite besitzt einen ähnlichen Fehler, wenn der Transceiver das Ende des momentan empfangenen Frames nicht signalisiert.

Durch diese Tests ist das größte Risiko durch Prüfung des Slot-Zustandsautomaten abgedeckt. Dieses Vorgehen prüft die übergeordnete Managementebene jedoch nicht. Die korrekte Abfolge der Slots im Superframe wird mit den Tests in der Evaluation auf Systemebene geprüft. Dazu gehören mit den PDR Überprüfungen auch die funktionale Prüfung z.B. der Retransmits, die auf dieser Ebene verwaltet werden.

Mit größerem Testumfang könnte zukünftig noch breiter getestet werden. Dazu bietet sich eine Risikoanalyse an, um weitere Aspekte des Gesamtsystems zu identifizieren, die zusätzlich getestet werden sollten, womit eine noch höhere Testüberdeckung erreicht werden kann. Dazu gehören z.B. dedizierte Tests des Zustandsautomaten auf der Management Ebene. In der aktuellen Implementierung ist dies weniger kritisch, kann aber höhere Komplexität mit der Einführung von bidirectional Timeslots und optionaler zweiter GACK Phase mit sich bringen. Eine weitere wichtige Komponente, die nicht direkt Teil des LLDNs ist, aber im openWSN System eine zentrale Rolle spielt ist die Open-Queue. Sie enthält die Elemente mit der Nutzlast, die zu senden ist oder empfangen wurde. Hier sollten zukünftig Grenzwerttests der Warteschlange durchgeführt werden, um sicherzustellen, dass diese korrekt arbeitet und nicht überläuft. Innerhalb eines open-QueueElements wird viel mit Pointerarithmetik gearbeitet. Ein Risikofaktor, der genauer Inspektion bedarf. Die Warteschlange enthält außerdem hochpriorisierte Elemente, wie z.B. Beacon Pakete, deren geänderte Behandlung einer genaueren Prüfung bedarf.

Zuletzt wäre zukünftig noch ein automatischer Regressionstest denkbar. Dabei könnte ein unabhängiger Knoten die gesendeten Pakete von Knoten mit der zu testenden Firmware prüfen. Hierbei kann dann neben dem korrekten Superframemuster auch die Beschaffenheit der versendeten Frames geprüft werden. So kann z.B. ein Beacon auf Konformität mit dem 802.15.4e getestet werden. Auch bei zukünftigen Weiterentwicklungen des Codes könnte so sichergestellt werden, dass keine Regressionen bzgl. der Standardkonformität auftreten.

A.3 Whitebox: Statische Codeanalyse

Moderne Compiler entdecken bereits viele Probleme während des Übersetzungsvorgangs und geben entsprechende Hinweise, ohne dass ein eigenes Tool zur Codeanalyse notwendig wäre. Trotzdem bietet sich der Einsatz eines solchen Werkzeugs an, um erweiterte Ergebnisse zu erhalten. Ein typisches Tool zur statischen Codeanalyse hat eine wesentlich höhere Laufzeit als ein gewöhnlicher Kompilervorgang und prüft wesentlich mehr Regeln ab, als der Compiler. Im Bereich des Speichermanagements ist die Prüfung umfangreicher und insbesondere bei embedded Projekten besonders wichtig. Durch die

kleine Menge an Speicher, die hier zu Verfügung steht kann ein etwaiges Speicherleck sehr schnell zum Ausfall des Gerätes führen.

Für die Prüfung fand der Clang Static Analyzer Verwendung. Im Vergleich zu üblichen kommerziellen Lösungen für C Projekte wie PC-lint oder Splint ist dieser frei verfügbar. Die ermittelten Probleme im LLDN Layer wurden dabei behoben. Dennoch gibt es hier insbesondere, da es sich um ein embedded Projekt handelt, viele falsche Positive. Außerdem gibt es noch diverse angezeigte Probleme im gesamten restlichen openWSN Projekt, die zukünftig analysiert und ggf. behoben werden sollten, um einen sicheren Betrieb zu gewährleisten.

Literatur

- [ACF11] G. Anastasi, M. Conti und M. Di Francesco. »A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks«. In: *IEEE Transactions on Industrial Informatics* 7.1 (Feb. 2011), S. 52–65. ISSN: 1551-3203. DOI: [10.1109/TII.2010.2085440](https://doi.org/10.1109/TII.2010.2085440).
- [AX14] Mashood Anwar und Yuanqing Xia. »IEEE 802.15. 4e LLDN: Superframe configuration for networked control systems«. In: *Control Conference (CCC), 2014 33rd Chinese*. IEEE, 2014, S. 5568–5573.
- [Cal02] Ed Callaway u. a. »Home networking with IEEE 802.15. 4: a developing standard for low-rate wireless personal area networks«. In: *IEEE Communications magazine* 40.8 (2002), S. 70–77.
- [CCS13] CCSDS. *Spacecraft Onboard Interface Systems – Low Data-Rate Wireless Communications For Spacecraft Monitoring and Control*. Bd. Recommended Practice, Issue 1. Magenta Book. Washington, DC, USA, 2013.
- [Cha16] Maximilien Charlier u. a. »Support for IEEE 802.15. 4 ultra wideband communications in the Contiki operating system«. In: *Communications and Vehicular Technologies (SCVT), 2016 Symposium on*. IEEE, 2016, S. 1–6.
- [DBA16] Domenico De Guglielmo, Simone Brienza und Giuseppe Anastasi. »IEEE 802.15.4e: A survey«. en. In: *Computer Communications* 88 (Aug. 2016), S. 1–24. ISSN: 01403664. DOI: [10.1016/j.comcom.2016.05.004](https://doi.org/10.1016/j.comcom.2016.05.004).
- [DEA06] Ilker Demirkol, Cem Ersoy und Fatih Alagoz. »MAC protocols for wireless sensor networks: a survey«. In: *IEEE Communications Magazine* 44.4 (2006), S. 115–121.
- [Dec15] Decawave Ltd. *DW1000 User Manual*. 2015.
- [Di 14] Piergiuseppe Di Marco u. a. »Modeling IEEE 802.15. 4 networks over fading channels«. In: *IEEE Transactions on Wireless Communications* 13.10 (2014), S. 5366–5381.
- [Die06] Dietmar Rudolph. *Spread Spectrum Modulation*. Vorlesungsskript. Technische Fachhochschule Berlin, 2006.
- [DSP17] Martin Drobczyk, Christian Strowik und Claudia Philpot. »A wireless communication and positioning experiment for the ISS based on IR-UWB«. In: *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*. IEEE, 2017, S. 1–6.

- [Dun11] Adam Dunkels. »The contikimac radio duty cycling protocol«. In: (2011).
- [Eur08] European Commission for Space Standardization. *ECSS-E-ST-50-12C – Space Wire – Links, nodes, routers and networks*. Juli 2008.
- [Eur09] European Commission for Space Standardization. *ECSS-M-ST-10C – Space Project Management - Project planning and implementation*. März 2009.
- [Grü17] Stephan Grünfelder. *Software-Test für Embedded Systems: Ein Praxishandbuch für Entwickler, Tester und technische Projektleiter*. de. dpunkt.verlag, Mai 2017. ISBN: 978-3-96088-149-0.
- [Hau16] Hauke Martens. *Implementierung eines drahtlosen Sensornetzwerks zur unterstützenden Sensorauslese am Beispiel des Kompaktsatelliten Eu:CROPIS*. Master-Thesis. Bremen: Hochschule Magdeburg-Stendal, Juli 2016.
- [IEE07] IEEE Computer Society u. a. *IEEE standard for information technology: telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements. Part 15.4, Amendment 1, Part 15.4, Amendment 1*, English. OCLC: 172962953. New York, NY: Institute of Electrical and Electronics Engineers, 2007. ISBN: 978-0-7381-5583-8 978-0-7381-5584-5.
- [IEE12a] IEEE. »IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer«. In: *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)* (Apr. 2012), S. 1–225. DOI: [10.1109/IEEESTD.2012.6185525](https://doi.org/10.1109/IEEESTD.2012.6185525).
- [IEE12b] IEEE Computer Society u. a. *IEEE standard for local and metropolitan area networks. Part 15.4, Amendment 1, Part 15.4, Amendment 1*, English. OCLC: 798579376. New York: Institute of Electrical and Electronics Engineers, 2012. ISBN: 978-0-7381-7225-5. (Besucht am 10. 01. 2018).
- [IEE16] IEEE. »IEEE Standard for Low-Rate Wireless Networks«. In: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (Apr. 2016), S. 1–709. DOI: [10.1109/IEEESTD.2016.7460875](https://doi.org/10.1109/IEEESTD.2016.7460875).
- [Kar10] E. Karapistoli u. a. »An overview of the IEEE 802.15.4a Standard«. In: *IEEE Communications Magazine* 48.1 (Jan. 2010), S. 47–53. ISSN: 0163-6804. DOI: [10.1109/MCOM.2010.5394030](https://doi.org/10.1109/MCOM.2010.5394030).
- [KNS17] D. M. King, B. G. Nickerson und W. Song. »Evaluation of ultra-wideband radio for industrial wireless control«. In: *2017 IEEE 38th Sarnoff Symposium*. Sep. 2017, S. 1–6. DOI: [10.1109/SARNOF.2017.8080385](https://doi.org/10.1109/SARNOF.2017.8080385).

- [KWW11] Klaus Wittmann, Wilfried Ley und Willi Hallmann. *Handbuch der Raumfahrt-technik : mit 135 Tabellen*. 4., aktualisierte und erw. Aufl. München: Hanser, 2011. ISBN: 978-3-446-42406-7.
- [Lec09] Aubin Lecointre u. a. »Performance evaluation of impulse radio ultra wide band wireless sensor networks«. In: *Military Communications Conference, 2009. MILCOM 2009. IEEE*. IEEE, 2009, S. 1–7.
- [LFX09] Shaobin Li, Guochen Fan und Guodong Xu. »Application of Wireless Satellite Bus in micro-satellite design«. In: *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*. IEEE, 2009, S. 2612–2616.
- [LSH08] Tomas Lennvall, Stefan Svensson und Fredrik Hekland. »A comparison of WirelessHART and ZigBee for industrial applications«. In: *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*. IEEE, 2008, S. 85–88.
- [Mic17] Michael Jetzschmann u. a. »Scalability and Modularity as Dimensions of Flexibility of a Microsatellite Platform«. In: *IAC*. Adelaide: International Astronautical Congress, 2017.
- [MLR07] M. Ghavami, L. B. Michael und R. Kohno. *Ultra Wideband Signals and Systems in Communication Engineering*. 2. Aufl. Wiley & Sons, 2007. ISBN: 978-0-470-02763-9.
- [MPT13] Konstantin Mikhaylov, Nikolaos Plevritakis und Jouni Tervonen. »Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI«. en. In: *Journal of Sensor and Actuator Networks* 2.3 (Aug. 2013), S. 589–613. ISSN: 2224-2708. DOI: [10.3390/jsan2030589](https://doi.org/10.3390/jsan2030589).
- [Noh09] Masahiro Nohmi. »Initial experimental result of pico-satellite KUKAI on orbit«. In: *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*. IEEE, 2009, S. 2946–2951.
- [NR12] Mark Nixon und T. X. Round Rock. »A Comparison of WirelessHART and ISA100.11a«. In: *Whitepaper, Emerson Process Management (2012)*, S. 1–36.
- [OM03] T.R. Oldham und F.B. McLean. »Total ionizing dose effects in MOS oxides and devices«. en. In: *IEEE Transactions on Nuclear Science* 50.3 (Juni 2003), S. 483–499. ISSN: 0018-9499. DOI: [10.1109/TNS.2003.812927](https://doi.org/10.1109/TNS.2003.812927).
- [PAB14] Gaetano Patti, Giuliana Alderisi und Lucia Lo Bello. »Introducing multi-level communication in the IEEE 802.15.4e protocol: the MultiChannel-LLDN«. In: *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*. IEEE, 2014, S. 1–8.

- [PHI13] Tuomas Paso, Jussi Haapola und Jari Iinatti. »Feasibility study of IEEE 802.15.4e dsme utilizing ir-uwband s-aloha«. In: *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*. IEEE, 2013, S. 1863–1867.
- [PL16] Gaetano Patti und Lucia Lo Bello. »A Priority-Aware Multichannel Adaptive Framework for the IEEE 802.15.4e-LLDN«. In: *IEEE Transactions on Industrial Electronics* 63.10 (Okt. 2016), S. 6360–6370. ISSN: 0278-0046, 1557-9948. DOI: [10.1109/TIE.2016.2573754](https://doi.org/10.1109/TIE.2016.2573754).
- [Pol08] S. Pollin u. a. »Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Medium Access Layer«. In: *IEEE Transactions on Wireless Communications* 7.9 (Sep. 2008), S. 3359–3371. ISSN: 1536-1276. DOI: [10.1109/TWC.2008.060057](https://doi.org/10.1109/TWC.2008.060057).
- [SN18] Shea Newton und Nathan Aschbacher. *The Challenge of Using C in Safety-Critical Applications*. Whitepaper. Portland: PolySync Technologies, Inc., Apr. 2018, S. 7.
- [Spa17] Consultative Committee for Space Data Systems. *Wireless Network Communications Overview For Space Mission Operations*. Green Book. Washington, DC, USA, 2017.
- [SRK10] N. Salman, I. Rasool und A. H. Kemp. »Overview of the IEEE 802.15.4 standards family for Low Rate Wireless Personal Area Networks«. In: IEEE, Sep. 2010, S. 701–705. ISBN: 978-1-4244-6315-2. DOI: [10.1109/ISWCS.2010.5624516](https://doi.org/10.1109/ISWCS.2010.5624516).
- [Tho12a] Thomas Watteyne. *Implementing IEEE 802.15.4e – State Machine*. Nov. 2012. URL: <https://openwsn.atlassian.net/wiki/spaces/OW/pages/688251/State+Machine>.
- [Tho12b] Thomas Watteyne u. a. »OpenWSN: A Standards-Based Low-Power Wireless Development Environment«. In: *European Transactions on Telecommunications* 23.5 (2012), S. 480–493.
- [Tho17] Thomas Gärtner u. a. »Scalable Avionics for the DLR Micro- and Minisatellite Platforms S2TEP and COMPSAT«. In: *DASIA*. Göteborg: Data Systems in Aerospace, 2017.
- [Xie14] Shuanglong Xie u. a. »Wireless Sensor Network for Satellite Applications: A Survey and Case Study«. In: *Unmanned Systems* 02.03 (Juli 2014), S. 261–277. ISSN: 2301-3850. DOI: [10.1142/S2301385014400056](https://doi.org/10.1142/S2301385014400056).

Begriffe und Abkürzungen

- 6LoWPAN** IPv6 over Low Power Wireless Personal Area Network. Kommunikationsprotokoll, das die IPv6 Datenübertragung in drahtlosen Sensornetzwerken z.B. durch Kompression und Fragmentierung der Pakete optimiert.
- AR** Acknowledge Request – Signalisiert der Gegenstelle, dass eine Bestätigung der empfangenen Nachricht gewünscht ist.
- BPSK** Binary Phase Shift Keying – Ein Phasenmodulationsverfahren bei dem zwischen zwei Phasenlagen unterschieden wird.
- BSP** Board Support Package – Abstraktionsschicht, die die Hardware eines spezifischen Mikrocontrollers an den Rest eines Systems anbindet.
- CAP** Contention Access Period – Zugriffsperiode im 802.15.4 MAC Layer, bei der Knoten um das Medium konkurrieren.
- CCA** CCA – Clear Channel Assessment – Feststellung eines Transceivers, ob der gewählte Kanal frei ist. Im Gegensatz zur Energy Detection werden hier Präambeln von Headern dekodiert, um z.B. die Dauer festzustellen, die der Kanal für eine andere Übertragung blockiert sein wird festzustellen.
- CFP** Contention Free Period – Zugriffsmethode im 802.15.4 MAC Layer, bei der Knoten garantierte Zeitslots vom Coordinator zugesichert bekommen. Dabei muss kein CSMA/CA Verfahren durchgeführt werden.
- CoAP** Constrained Application Protocol – Auf dem REST Architekturmodell basierendes zustandsloses Applikationsprotokoll, das auf den Einsatz auf IoT Geräten optimiert ist.
- COTS** Commercial off the Shelf.
- CRC** Cyclic Redundancy Check – Ein Verfahren zur Berechnung einer Prüfsumme.
- CSMA/CD** Carrier Sense Multiple Access / Collision Avoidance – Verfahren eines Transceivers, um festzustellen, dass auf einem Kanal momentan übertragen wird und um Kollisionen zu vermeiden, indem die Übertragungsfenster analysiert werden.

DSME	Deterministic and Synchronous Multi-channel Extension – Erweiterter MAC Modus des 802.15.4e mit Channel Hopping, garantierten Zeitschlitzten und einer Konzentration auf Mesh Netzwerke.
DSSS	DSSS – Direct Sequence Spread Spectrum. Eine Modulationsart, bei der eine Frequenzspreizung auf Basis eines Codes vorgenommen wird, der zuvor aus den zu übertragenden Daten extrahiert wurde. Durch die Spreizung auf eine größere Bandbreite wird das Signal unempfindlicher gegen schmalbandige Störungen.
ED	Energy Detection – Oft in energiesparenden Transceivern eingesetztes Verfahren, um festzustellen, dass ein Signal auf einer gegebenen Frequenz vorhanden ist.
FCS	Frame Check Sequence – Prüfsumme über Header und Daten, die in einem Trailer des MAC Frames gesendet wird.
GPIO	General Purpose Input/Output – Als Ein- oder Ausgabe konfigurierbare Pins eines Mikrocontrollers, die keine vorbestimmte Funktion erfüllen und frei ansteuerbar sind.
GTS	Guaranteed Time Slot – Zeitslot, der einem Knoten von einem Coordinator in der CFP zugesichert wird.
I²C	Inter-Integrated Circuit – Ein serieller Bus zur Kommunikation zwischen integrierten Schaltkreisen.
IEEE	Institute of Electrical and Electronics Engineers.
IFS	Inter Frame Space – Wartezeit zwischen zwei nacheinander übertragenen Frames.
LLDN	Low Latency Deterministic Network – Erweiterter MAC Modus des 802.15.4e, der niedrige Latenz und hohe Zuverlässigkeit bietet.
LQI	Link Quality Indicator – Eine Metrik, die die Einfachheit der Demodulation eines empfangenen Signals angibt. Verwandt ist die Received Signal Strength Indication (RSSI), die Auskunft über die Stärke des Signals gibt, aber keine Aussage über die Qualität trifft.
LRWPAN	Low Rate Wireless Personal Area Network – WPAN mit geringer Datenrate.

MAC	Medium Access Control – Die Sicherungsschicht nach dem OSI Modell stellt die Schnittstelle zur Übertragungshardware dar und sorgt für eine zuverlässige Übertragung.
OSI	Open Systems Interconnection Model – Schichtenmodell, dass Netzwerkkommunikation in möglichst abgeschlossene und damit portierbare Teile einteilt.
PDI	Packet Delivery Ratio – Prozentuale Rate mit der Datenpakete erfolgreich zugestellt werden im vgl. zu Paketen die ausgesendet wurden.
PHR	Physical Header – Enthält die Länge des folgenden Nutzdatenstroms.
PPDU	Physical Protocol Data Unit – Format des Frames, der über das physische Medium versendet wird.
PRF	Pulse Repetition Frequency – Rate, mit der in einem UWB System die Pulse wiederholt werden.
QAM	Quadrature Amplitude Modulation – Kombination aus Phasen und Amplitudenmodulation um eine höhere Symbolrate zu erreichen.
QPSK	Quadrature Phase Shift Keying – Ein Phasenmodulationsverfahren bei dem zwischen vier Phasenlagen unterschieden wird.
RPL	Routing Protocol for Low Power and Lossy Networks – Von der IETF spezifiziertes Routingprotokoll für Sensornetze.
RTOS	Real Time Operating System – Echtzeitbetriebssystem.
S2TEP	Small Satellite Technology Experiment Platform.
SFD	Start Frame Delimiter – Markiert den Beginn eines physischen Frames.
SHR	Synchronization Header – Feste Bitfolge zur Synchronisation eines Empfängers.
SNR	Signal to Noise Ratio – Signalrauschabstand.
SPI	Serial Peripheral Interface – Ein vollduplexfähiges Bussystem, das häufig Verwendung findet, um Peripherie an Mikrocontroller anzubinden.

TDMA	Time Division Multiple Access – Zugriffsverfahren, bei dem Kollisionsfreiheit durch das Zuteilen von Zeitschlitzten an beteiligte Knoten sichergestellt wird.
TSCH	Time Slotted Channel Hopping – Erweiterter MAC Modus des 802.15.4e, der dynamische Kanalwechsel und ein TDMA Verfahren bietet.
UART	Universal Asynchronous Receiver Transmitter – Schnittstelle zur seriellen Datenübertragung.
WPAN	Wireless Personal Area Network.
WSN	Wireless Sensor Network.