6LoWPAN

Einführung in die Technologie und Implementierung mit Contiki

Julian Rebholz,Tino Weber, Alex Jäger

12. Juni 2019

Inhaltsverzeichnis

1 Einführung

Das Jahr 1969 gilt heute als Geburtsstunde des Internets. Mithilfe des Telnet-Protokolls kommunizierten vier Rechner in Kalifornien und Utah miteinander und legten somit den Grundstein für eine Technologie, die die Welt im Laufe der Zeit veränderte wie kaum eine vor ihr. [4] Nur fünf Jahre später entwickeln Robert Kahn und Vinton Cerf gemeinsam das Transmission Control Protocol (TCP) und machen damit erste Schritte hin zu einem modernen Netzwerk wie wir es heute kennen und nutzen.
Als das World Wide Web (WWW) 1990 am CERN in Genf entwickelt wird, handelt es sich noch um eine Kommunikationsplattform für Wissenschaftler. Zu diesem Zeitpunkt ist kaum zu erahnen, welche Trends sich im Jahr 2019 - nur ein halbes Jahrhundert nach dem Beginn des Internets - entwickeln. Die grundlegende Idee besteht heutzutage daraus, möglichst viele Geräte verschwindend geringer Größe an das Internet anzubinden, sodass sogar einzelne Sensoren selbstständig Daten an zentrale Server schicken können. Dieser technologische Wandel stellt Ingenieure und Informatiker vor ganz neue Herausforderungen, die es zu bewältigen gilt.
Auf den ersten Blick ist nicht ganz ersichtlich, warum kleine Geräte schwieriger ins Internet zu integrieren sind als große. Die Problematik, dass das Internet Protocol Version 4 (IPv4) durch die 32-bit Internet Protocol (IP)-Adresse nur $2^32 \approx 4.200.000.000$ verschiedene Teilnehmer adressieren kann ist durch das Internet Protocol Version 6 (IPv6) und die Adressgröße von 128bit gelöst. Theoretisch böten die $2^128 \approx 340$ Sextillionen verschiedenen Adressen ausreichend Möglichkeiten, jedem denkbaren Gerät eine eigene IP-Adresse zuzuordnen.
Viel mehr sind es Probleme des lokalen Teilnehmers als des Netzwerks, die neue Technologien unabdingbar machen. Sowohl software- als auch hardwareseitig sind kleine Sensoren nämlich herkömmlich nicht dazu ausgelegt, internetfähig zu kommunizieren. Klassische IP-Stacks überforderten in der Vergangenheit die Rechenleistung der - nicht selten 8-bit - Mikrocontroller, leistungshungrige Wi-Fi-Module arbeiten zu ineffizient für die meist durch kleine Knopfzellen angetriebenen Geräte. Kabelbasierte Kommunikation fällt aufgrund der flexiblen Anbringung von Sensoren als Lösung ohnehin meist aus. Diese Gründe sorgten dafür, dass die Teilnahme kleiner, leistungsschwacher, batteriebetriebener Geräte in der Vergangenheit unmöglich erschien. Heute gilt es als erreichbares Ziel und wird mit dem Schlagbegriff Internet der Dinge oder englisch Internet of Things (IoT) gekennzeichnet.
Ein erster Schritt hin zu Realisierung dieses Ziels war der uIP-Stack. Als Adam Dunkels vom Swedish Institute of Computer Science (SICS) ihn entwickelte, war er der erste IP-Stack, der kompiliert so klein war, dass er die geringe Speichermöglichkeit von Mikrocontrollern nicht überlastete. Abhängig vom eingesetzten C Compiler kann die Größe des kompilierten Stacks zwischen mehreren hundert Bytes bis hin zu wenigen Kilobytes betragen. Um den Speicherverbrauch des Stacks (Hinweis: Hier 'Stack' im Sinne von Speicher) möglichst gering zu halten, setzt uIP auf globale Variablen, um Werte zwischen Funktionen zu kommunizieren. [1, S.5] Das 2003 ebenfalls von Adam Dunkels entwickelte Betriebssystem Contiki, das in späteren Kapiteln besprochen wird, setzt auf diesem Stack auf.

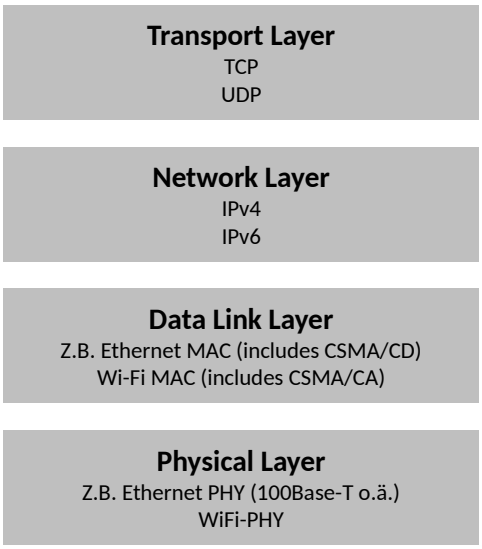

Abbildung 1: Beispielhafter Aufbau der ersten vier Schichten des OSI-Modells eines internetfähigen Geräts

Dieses Dokument präsentiert ein Projekt, das auf Basis von Contiki ein IPv6 over Low power Wireless Personal Area Network (6LoWPAN)-Netzwerk aufbaut. Dabei wird zunächst die Notwendigkeit des Protokolls 6LoWPAN und der darunter fallenden Norm IEEE 802.15.4 (15.4) - auch in Hinsicht ihres Beitrags zum Internet der Dinge - besprochen. Darauf folgend werden sukzessive die technischen Hintergründe von 15.4, 6LoWPAN und IPv6/UDP im Hinblick auf die Besonderheiten bzgl. 6LoWPAN erläutert. Dies soll eine Einführung in diese Technologie von Physical Layer bis hin zum Transport Layer des Open System Interconnection (OSI)-Modells darstellen. Daraufhin wird, um eine Realisierung eines Netzwerks auf Basis dieser Technik zu ermöglichen, eine Einführung in Contiki gegeben und zum Abschluss das konkrete Projekt erklärt.

1.1 Motivation

Der Wunsch, kleine Geräte internetfähig zu machen, wurde bereits in der Einleitung erläutert. Um dieses Ziel zu erreichen, ist die Implementierung einer Reihe von Protokollen nötig.

- Layer 1 und 2 des OSI-Modells werden für gewöhnlich durch Techniken nach IEEE 802.11 (Wi-Fi) oder IEEE 802.3 (Ethernet) realisiert. Diese Normen sind nicht zwingend notwendig, um als Teilnehmer im Internet aufzutreten, sind aber gängige Lösungen.
- Zentraler Bestandteil des Internets ist IP auf Layer 3. Erst durch dessen Implementierung kann ein Gerät am Internet teilnehmen.

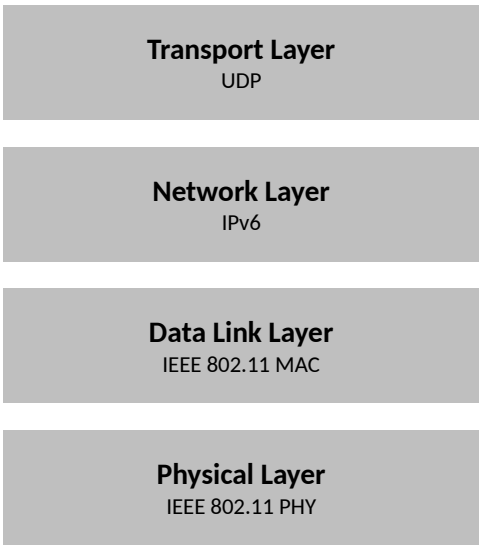

Abbildung 2: Theoretisch möglicher Aufbau der ersten vier Schichten des OSI-Modells für einen Sensor

- Auf Layer 4 sind die Protokolle User Datagram Protocol (UDP) und TCP gängig, um verbindungslos (UDP) oder verbindungsorientiert (TCP) Daten zu verschicken.

Somit ergibt sich ein Aufbau der ersten vier Schichten des OSI-Modells wie in Grafik 1.

Will man nun - wie es im IoT häufig der Fall ist - von einem Sensor Daten zur Verwaltung an einen zentralen Server schicken, so ist der Einsatz von UDP sinnvoll. Es handelt sich ohnehin meist um kleine Datensätze, die die Größe eines einzelnen IP-Pakets nicht selten nicht überschreiten. Ein komplexer Verbindungsaufbau wie bei TCP ist hierzu nicht nötig. Ein unterwegs verloren gegangenes Paket richtet durch die ständige Aktualisierung von Sensordaten kaum großen Schaden an.
Auf Layer 3 ist der Einsatz von IP obligatorisch. Weil es sich bei IPv4 um ein veraltetes Protokoll handelt, das immer mehr von IPv6 verdrängt wird und moderne, zukunftsfähige Sensornetzwerke kaum noch darauf setzen, wird in dieser Ausarbeitung nur IPv6 betrachtet.
Bei den beiden unteren Layern ist der Einsatz kabelgebundener Lösungen kaum realistisch. Sensoren sollen die Möglichkeit bieten, ohne großen Montageaufwand flexibel eingesetzt werden zu können. Durch den geringen Stromverbrauch reichen schon einfache Knopfzellenbatterien als Energielieferant, weshalb eine Verlegung von Kabeln nur zu Kommunikationszwecken nicht sinnvoll wäre. Lösungen wie Ethernet fallen somit weg und auf den ersten Blick erscheint der Einsatz von Wi-Fi sinnvoll. Somit ergäbe sich ein tatsächlicher Aufbau der ersten vier Schichten des OSI-Modells wie in Grafik 2 gezeigt. Herkömmliche Funkprotokolle - IEEE 802.11 alias Wi-Fi sei hier nur exemplarisch - bedienen die Anforderungen moderner Applikationen. Diese Anforderungen

lassen sich zum größten Teil in einem Wort zusammenfassen: Datendurchsatz. Heutige Applikationen für Endanwender benötigen nicht selten Datenraten im Bereich vieler Mbit/s. Diese Geschwindigkeiten werden durch eine hohe Leistungsaufnahme entsprechender Module erkauft. So kann die Leistungsaufnahme eines einfachen Wi-Fi-fähigen System on a Chip (SoC) wie des ESP8266EX der Firma Espressif Systems durchaus 500milli (m)Watt (W) erreichen [2]. Das ist unter Betracht der Tatsache, dass Knopfzellenbatterien für gewöhnlich nicht mehr als 3000mWh Energie speichern ein viel zu hoher Wert.
Sensor-Netzwerken benötigen im Gegensatz dazu nur geringe Datenraten, da nur wenige Daten - durchaus im Bereich einiger Bytes - verschickt werden. Andererseits ist hier ganz besonders auf die Leistungsaufnahme zu achten. Um diesen außergewöhnlichen Anforderungen gerecht zu werden, wurde eine Funktechnologie entwickelt und in IEEE 802.15.4 genormt. Weil man bei dieser Technologie auf einen einheitlichen Namen verzichtete - sie wird meist einfach als IEEE 802.15.4 bezeichnet - wird sie in dieser Ausarbeitung 15.4 genannt.
Der Standard 15.4 definiert Physical (PHY) und Medium Access Control (MAC) Layer eines Übertragungsprotokolls für Wireless Personal Area Network (WPAN). Durch lange Ruhephasen wird hier die Leistungsaufnahme einzelner Netzteilnehmer so weit reduziert, dass die gängigen Akkulaufzeiten sich für gewöhnlich im Bereich vieler Monate bewegen. Der Preis dafür ist eine geringe Datenrate von maximal 250 kilo (k)Bit/s bzw. im Sub-1-GHz-Netz sogar nur 40 kBit/s. Was bei anderen Anwendungen inakzeptabel wäre, reicht aber in diesem Falls absolut aus, um einfach Sensordaten ausreichend schnell zu übermitteln.
Eine Eigenschaft von 15.4 ist die maximale Paketgröße von 127 Bytes. IPv6 fordert jedoch eine Maximum Transmission Unit (MTU) von 1280 Byte. Das bedeutet, dass MTU vom auf Layer 2 eingesetzten Protokoll fordert, ein Paket mit maximal 1280 Bytes zu übertragen. Damit ist 15.4 als Übertragungsprotokoll eine IP-Pakets zunächst gar nicht möglich. Darüber hinaus hat der Header des IPv6-Protokolls eine Größe von 40 Bytes. Gemeinsam mit dem 8 Byte großen UDP-Header und dem 25 Byte großen MAC-Header von 15.4 blieben in einem Paket gerade mal 54 Byte für die Payload, wodurch sich die Datenrate von Nutzdaten auf ca. 17 kBit/s im Sub-1-GHz-Netz reduziert. [5]
Diese beiden Hindernisse verlangen von einer Implementierung von IPv6 auf Basis von 15.4 zwei Eigenschaften:

- Fragmentierungs- und Defragmentierungsebene für IP-Pakete einführen, um die MTU nicht zu verletzen.
- Header von IP und UDP möglichst weit komprimieren, um den Anteil an Nutzdaten in einem Datenpaket zu erhöhen.

Um beiden Anforderungen gerecht zu werden, wurde die 6LoWPAN Zwischenschicht eingeführt. Diese gliedert sich zwischen Layer 2 (15.4 MAC) und Layer 3 (IPv6) als Adaptionslayer ein und ermöglicht so die Nutzung von IP und darauf aufbauend UDP auf Basis von 15.4. Darüber hinaus muss zur Kommunikation nach außen mindestens ein Netzwerkteilnehmer in der Lage sein, Datenpakete, die mit 6LoWPAN verschickt wurden in

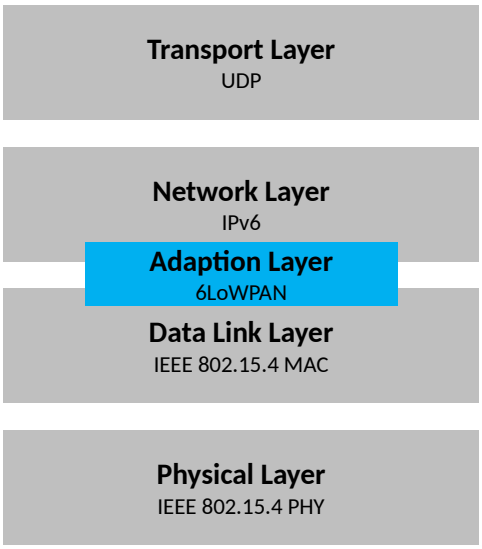

Abbildung 3: Aufbau der ersten vier Schichten des OSI-Modells unter Verwendung von 6LoWPAN

ganz normale UDP/IP-Pakete zu wandeln und über eine herkömmliche Übertragungstechnologie weiterzuleiten. Erst dadurch wird ein einzelner Sensor fähig, am Internet teilzunehmen. Die ersten vier Schichten des OSI-Modells eines solchen Sensors ergeben sich dann also wie in Graphik 3 gezeigt.

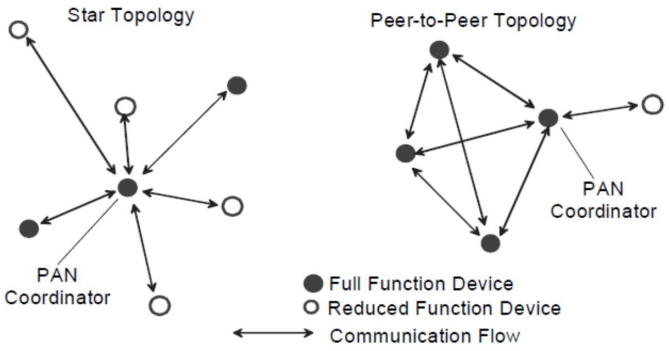

Abbildung 4: Topologien von 15.4-Netzwerken nach [3, S.46]

2 Der IEEE 802.15.4 Funkstandard

IEEE 802.15.4 oder kurz 15.4 beschreibt einen Funkstandard, der bei geringer Datenrate die Leistungsaufnahme einzelner Teilnehmer möglichst weit zu reduzieren versucht. Das ist nötig, um den Anforderungen von Sensornetzwerken gerecht zu werden.
Durch den Standard werden der PHY- und MAC-Layer eines WPAN definiert.

2.1 Teilnehmer und Topologien

Alle Teilnehmer eines 15.4 Personal Area Network (PAN) werden in zwei Kategorien eingeteilt:

- Ein Full-Function Device (FFD) ist ein Teilnehmer mit vollständiger Implementierung von 15.4. Er kann alle Rollen annehmen.
- Ein Reduced-Function Device (RFD) hat reduzierte Fähigkeiten. Es kann nur mit einem FFD kommunizieren und somit nicht als Coordinator (siehe unten) auftreten.

15.4 ist gemäß IEEE für zwei verschiedene Netzwerk-Topologien ausgelegt:

- Sterntopology
- Peer-to-Peer-Topology

In beiden Topologien existiert ein sogenannte PAN-Coordinator, der jedoch verschiedene Aufgaben übernimmt.
In einer Stern-Topologie ist der PAN-Coordinator der zentrale Kommunikationspunkt des Netzwerks. Alle Geräte kommunizieren nur mit ihm, was bedeutet, dass der Coordinator einen vergleichsweise hohen Energiebedarf hat. Die Teilnehmer hingegen müssen nur für ihre eigene Kommunikation aufwachen und können die restliche Zeit im Sleeping-Modus verbringen. Diese Topologie bietet sich an, wenn man einen netzbetriebenen

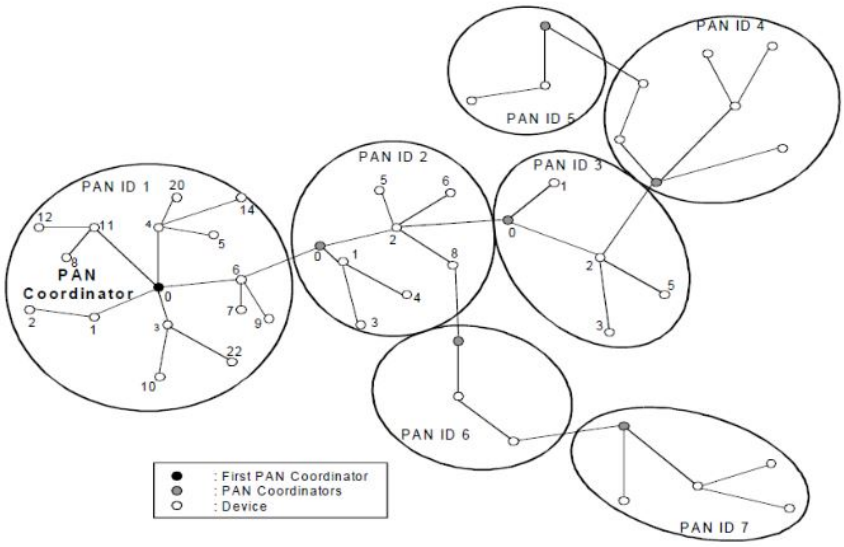

Abbildung 5: Ein Multicluster Tree Netzwerk nach [3, S.48]. Die Verbindungen zeigen nicht den Kommunikationsfluss, sondern Parent-Child-Beziehungen

Coordinator und mehrere batteriebetriebene Teilnehmer hat wie es z.B. in der Gebäudeautomatisierung oder bei Personal Computer (PC) Peripheriegeräten der Fall ist.
Eine Peer-to-Peer-Topologie bietet sich offensichtlich für effizientes Routing an. Hierbei kommunizieren alle FFD miteinander und RFD als sogenannte Leafs mit einem FFD. Anders als bei der Stern-Topologie ist somit der Coordinator nicht zentrale Kommunikationsstelle, sondern bietet Synchronisierungsservices o.ä. Diese Topologie bietet sich durch die Möglichkeit, Nachrichten per Multi-Hop zu routen im Bereich der Sensor-Netzwerke besonders an.

Eine besondere Form der Peer-to-Peer-Technologie ist der Cluster Tree. Ein FFD beginnt einen neuen Cluster Tree, indem es eine freie PAN ID für das Netzwerk sucht und beginnt, sogenannte Beacons auszusenden. Das Device wird damit automatisch der Coordinator dieses Netzwerks. Ein Nachbar, der das Beacon empfängt, kann nun anfragen, am Netzwerk teilzunehmen. Erlaubt der Coordinator das, merkt er sich den Nachbarn als Child und dieser wiederum merkt sich den Coordinator als Parent. Der Nachbar beginnt nun seinerseits Beacons auszusenden, die weitere Geräte (auch solche, die nicht in direkter Reichweite des Coordinators sind) nutzen können, um dem Netzwerk teilzunehmen. Dadurch können große Netzwerke mit komplexen Routing-Tabellen entstehen. Darüber hinaus kann der PAN Coordinator einen Teilnehmer seines Netzwerks anweisen, ein eigenes Cluster zu konstruieren, bei dem er als PAN Coordinator auftritt. Dieser wiederum hält weiterhin Kontakt zu einem der Devices des originalen Netzwerks, wodurch große Multicluster-Netzwerke entstehen können. Grafik 5 zeigt ein solches Netzwerk. Durch diese Technik wird ein großer Netzwerkradius erreicht, wobei die Latenz einer Nachricht mit der Größe des Netzwerks zunimmt.

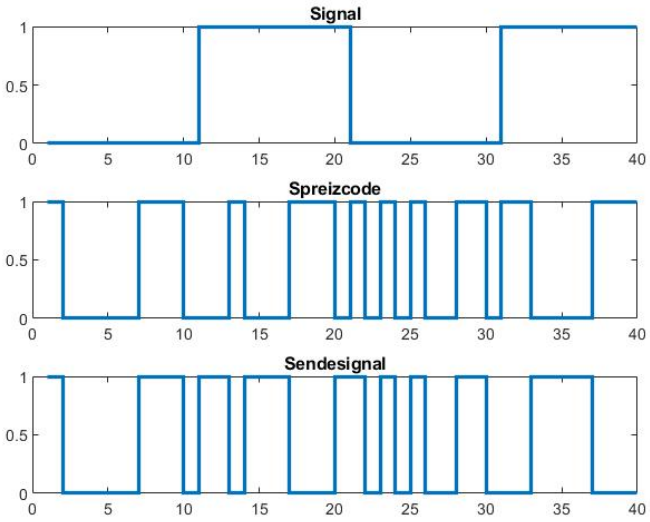

Abbildung 6: Ein per DSSS gespreiztes Signal

2.2 Physical Layer

2.2.1 Genutzte Frequenzbänder

15.4 nutzt die folgenden Frequenzbänder:

- 2,4-2,4835 GHz ISM Band (16 Kanäle) weltweit
- 902-928 MHz ISM Band (10 Kanäle) in den USA
- 868-868,6 MHz Band (1 Kanal) in Europa

Dabei wird im 902MHz-Band ein Kanalabstand von 2 MHz und im 2,4GHz-Band ein Kanalabstand von 5 MHz eingesetzt. Die Kanäle werden mit Channel 0 (868 MHz), Channel 1-10 (902 MHz) und Channel 11-26 (2,4 GHz) bezeichnet.

2.2.2 Genutzte Spreiz- und Modulationsverfahren

Zum Verständnis der Modulation des Physical Layer durch 15.4 sind Kenntnisse über verschiedene Modulationsverfahren nötig. Diese sind im Folgenden kurz beschrieben.

Direct Sequence Spread Spectrum (DSSS)
Das Direct Sequence Spread Spectrum (DSSS) - ein sogenanntes Spreizverfahren - dient dazu, den Bitstrom auf ein breiteres Frequenzband zu spreizen. Dabei wird die ursprüngliche Bitfolge auf einen pseudo-zufällige Subbit-Strom moduliert. Die einzelnen Subbits

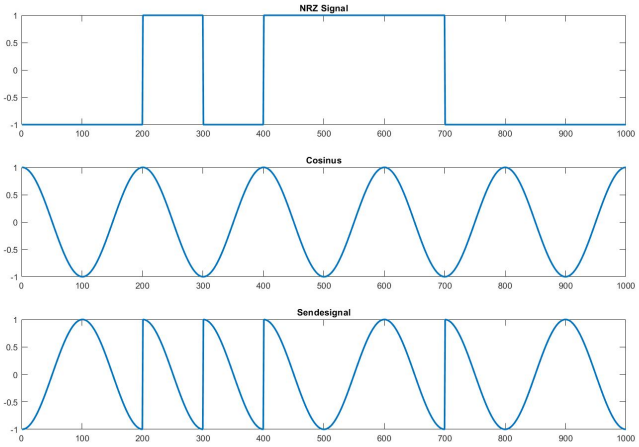

Abbildung 7: Ein BPSK-moduliertes Signal mit einer Cosinus-Schwingung als Träger

dieses Stroms werden Chips genannt, der Strom Chipfolge oder Spreizcode.
In Abbildung 6 wird das Signal mit einem Spreizcode unter Verwendung der XOR-Funktion gespreizt. Auf jedes Bit des Signals kommen dabei 10 Chips, wodurch eine Chiprate erreicht wird, die zehn Mal so hoch wie die Bitrate ist. Dieses Verhältnis wird auch Spreizfaktor genannt.
Sinn der DSSS ist es, ein Signal unempfindlicher schmalbandiger Störungen zu machen. Offensichtlich muss dem Empfänger zu Demodulation entweder der Spreizcode oder - da er pseudo-zufällig ist - der Generator zur Erstellung des Spreizcodes bekannt sein.

Binary Phase Shift Keying (BPSK)
Die Binary Phase Shift Keying (BPSK) ist die einfachste Form des Phase-Shift-Keyings. Die Bitfolge wird dabei als bipolares Non-return to zero (NRZ) dargestellt und daraufhin mit dem Trägersignal multipliziert. Grafik 7 zeigt das Prinzip.

Offset Quadrature Phase Shift Keying (O-QPSK)
Das Offset Quadrature Phase Shift Keying (O-QPSK) ist eine komplexere Form des Phase-Shift-Keyings, bei der mit jedem Symbol ein Informationsgehalt von zwei Bits übertragen wird. Zunächst wird normales QPSK betrachtet. Hierbei wird der Bitstrom aufgeteilt in einen Real- und einen Imaginärbitstrom, wobei jedes zweite Bit in den Imaginärbitstrom fließt. Diese Ströme gehen daraufhin als Eingänge in einen **QAM!** (**QAM!**) Modulator, welcher sie mit einem Cosinus (Real) bzw. Sinus (Imaginär) multipliziert und die Resultate subtrahiert, um das Sendesignal zu bilden.
Eine sinnvollere Aufteilung wird dadurch erreicht, dass der Bitstrom in der komplexen Zahlenebene so gemappt wird, dass der komplexe Zahlenstrom, der durch das Mapping entsteht gleichmäßig um den Ursprung verteilt ist. Das kann beispielsweise dadurch realisiert werden, dass die Zahlenfolge 00 als 0+0j dargestellt wird, die Zahlenfolge 01 als

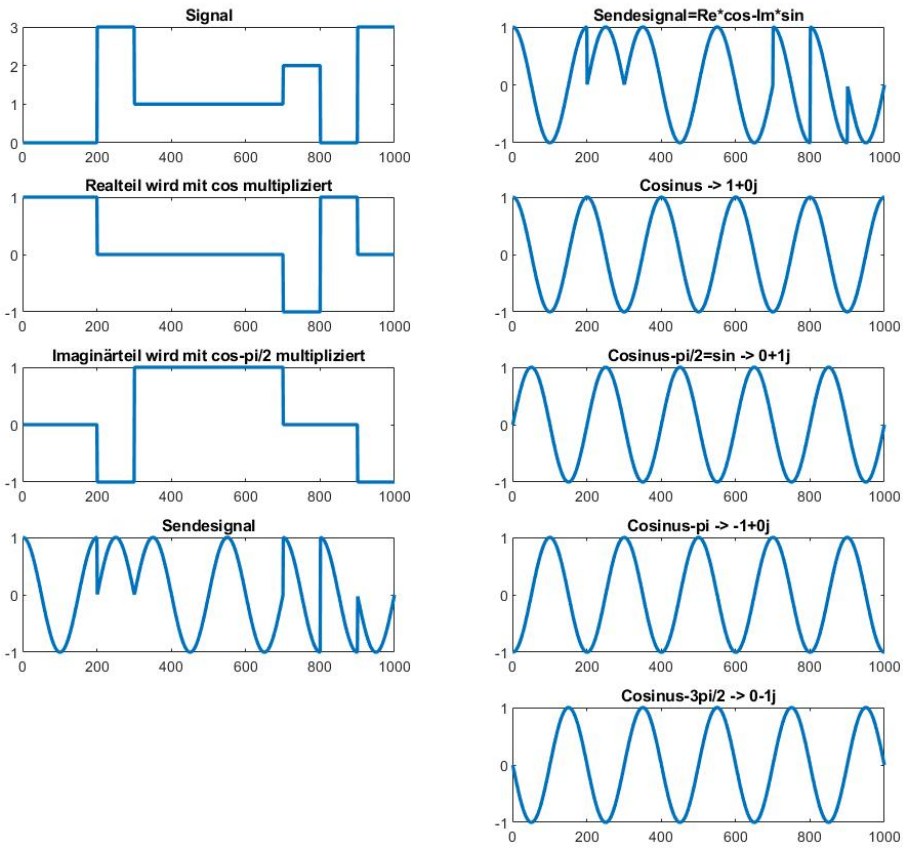

Abbildung 8: Beispielhafte Erzeugung eines Sendesignals mit QPSK (links) und das Sendesignal im Vergleich mit phasenverschobenen Cosinus-Schwingungen (rechts).

Abbildung 9: Ablauf der Modulation bei 15.4 aus [3, S.412]

0+1j usw. Häufig werden auch die resultierenden Werte um $\pi/4$ verschoben, also 00 auch 1+1j gemappt, 01 auf -1+1j usw.
Statt eines binären Zahlenstroms wird häufig ein Zahlenstrom mit Werten im Bereich [0,3] eingesetzt. Dieser hat die halbe Symbolrate des binären Stroms, aber statt einem zwei Bit/Symbol, wodurch wiederum die gleiche Bitrate erreicht wird. Der Vorteil besteht darin, dass er die gleiche Symbolrate wie der resultierende komplexe Zahlenstrom hat.
Die Erzeugung eines Signals mit QPSK ist in Grafik 8 anschaulich dargestellt.
Im Sendesignal ist klar erkennbar, dass bei Nulldurchgängen, also zum Beispiel von 1+0j zu -1+0j große Phasensprünge resultieren. Weil man dies vermeiden will, verschiebt man Real- und Imaginärteil des komplexen Signals um $\pi/2$ und bezeichnet das Resultat als Offset-QPSK oder kurz O-QPSK.

2.2.3 Konkretes Modulationsverfahren

Tatsächlich benutzt 15.4 eine Kombination aus DSSS und O-QPSK im 2,4GHz-Band und DSSS kombiniert mit BPSK in den restlichen Bändern. Der Ablauf der Modulationsverfahren ist in Abbildung 9 dargestellt. Zunächst wird der Datenstrom in Oktette aufgeteilt. Jedes Oktett wird wiederum in zwei Symbole aufgeteilt, wobei die LSB (b0,b1,b2,b3) ein Symbol bilden und die MSB (b4,b5,b6,b7) ein anderes. Es ist darauf zu achten, dass das Symbol der LSB zuerst versendet wird.
Weil ein Symbol aus 4 bit besteht, kann es 16 verschiedene Werte annehmen. Entsprechend des Werts des Symbols wird einer von 16 verschiedenen, zueinander orthogonalen Spreizcodes verwendet, die im 2,4GHz-Band eine Länge von 32 Chips und in den restlichen Bändern eine Länge von 16 Chips haben und in der IEEE-Norm für das 2,4GHz-Band wie folgt festgelegt sind.

Wert des Symbols	32-bit Spreizcode
0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
2	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
3	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
4	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
5	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
6	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
7	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1
8	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
9	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1
10	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1
11	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
12	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
13	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
14	1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
15	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

Für die restlichen Bänder werden die Symbole auf folgende Spreizcodes gemappt. [3, S.414]

Wert des Symbols	16-bit Spreizcode
0	0 0 1 1 1 1 1 0 0 0 1 0 0 1 0 1
1	0 1 0 0 1 1 1 1 1 0 0 0 1 0 0 1
2	0 1 0 1 0 0 1 1 1 1 1 0 0 0 1 0
3	1 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0
4	0 0 1 0 0 1 0 1 0 0 1 1 1 1 1 0
5	1 0 0 0 1 0 0 1 0 1 0 0 1 1 1 1
6	1 1 1 0 0 0 1 0 0 1 0 1 0 0 1 1
7	1 1 1 1 1 0 0 0 1 0 0 1 0 1 0 0
8	0 1 1 0 1 0 1 1 0 1 1 1 0 0 0 0
9	0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 0
10	0 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1
11	1 1 0 0 0 0 0 1 1 0 1 0 1 1 0 1
12	0 1 1 1 0 0 0 0 0 1 1 0 1 0 1 1
13	1 1 0 1 1 1 0 0 0 0 0 1 1 0 1 0
14	1 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0
15	1 0 1 0 1 1 0 1 1 1 0 0 0 0 0 1

3 6LoWPAN Adaption Layer und IPv6

4 Contiki als OS für verteilte Systeme

5 Beispielhafte Implementierung eines 6LoWPAN-Netzwerks auf Basis von Contiki

Abkürzungsverzeichnis

15.4	IEEE 802.15.4
6LoWPAN	IPv6 over Low power Wireless Personal Area Network
BPSK	Binary Phase Shift Keying
DSSS	Direct Sequence Spread Spectrum
FFD	Full-Function Device
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
k	kilo
m	milli
MAC	Medium Access Control
MTU	Maximum Transmission Unit
NRZ	Non-return to zero
O-QPSK	Offset Quadrature Phase Shift Keying
OSI	Open System Interconnection
PAN	Personal Area Network
PC	Personal Computer
PHY	Physical
QAM	Quadraturamplitudenmodulation
RFD	Reduced-Function Device
SICS	Swedish Institute of Computer Science
SoC	System on a Chip
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

W	Watt
WPAN	Wireless Personal Area Network
WWW	World Wide Web

Literatur

[1] Dunkels, Adam: uIP- A FreeSmallTCP/IPStack / Swedish Institute of Computer Science. 2001. – Forschungsbericht. – Online erhältlich unter 'http://www.dunkels.com/adam/download/uip-doc-0.5.pdf'. Abgerufen am 12. Juni 2019

[2] Espressif Inc (Hrsg.): *ESP8266EX Datasheet*. https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf: Espressif Inc, 2018. (ESP8266EX) . – Version 6.0

[3] IEEE: *IIEEE SStandardfforLLow–RRate WWireless Networks*. IEEE, New York, 2016

[4] Konitzer, Andreas: *Geschichte des Internets*. Website, . – Online erhältlich unter 'https://www.lmz-bw.de/medien-und-bildung/medienwissen/informatik-robotik/historisches/geschichte-des-internets/'. Abgerufen am 12. Juni 2019.

[5] Lehmann, Enrico: Grundlagen6LoWPAN - Einführung in das „Internet der Dinge“ / dresden elektronik ingenieurtechnik gmbh. 2012. – Forschungsbericht. – Online erhältlich unter 'https://www.dresden-elektronik.de/fileadmin/Downloads/Dokumente/white_paper/fundamentals_6lowpan-WP-de.pdf'. Abgerufen am 12. Juni 2019