UNIVERSITÄT KOBLENZ · LANDAU

Institut für
ComputerVisualistik

Seminarausarbeitung
Universität Koblenz-Landau WS 2003 / 2004

Von:

Marco Thum
mthum@uni-koblenz.de

Seminar Wireless Networks

29.01.2004

Geleitet von:

Prof. Dr. Ch. Steigner
Dipl.-Inf. H. Dickel

Inhaltsverzeichnis

1. Motivation

Im Jahr 1994 begann die Firma Ericsson mit einer Studie zur Untersuchung von Möglichkeiten zur kostengünstigen und ressourcenschonenden Übertragung von Daten zwischen Mobiltelefonen und ihren Accessoires innerhalb kurzer Distanzen. Ziel war es, zum einen Funkwellen als Übertragungsmedium zu verwenden, um die Nachteile der Sichtkontakt benötigenden Infrarotschnittstelle aufzuheben und sich lästigen Kabelsalat zu ersparen, und zum anderen eine Schnittstelle zu schaffen, die universell für zahlreiche Services eingesetzt werden kann, ohne weitreichende Kenntnisse von Netzwerktechnik zu besitzen. Nach Abschluss der Studie versuchte Ericcson 1997 mit Hilfe anderer namhafter Hersteller von tragbaren elektronischen Geräten einen Standard zu entwickeln. Ein Jahr später schlossen sich nun IBM, Intel, Ericsson, Nokia und Toshiba zur *Bluetooth Special Interest Group (SIG)* zusammen, um an der Spezifikation des Standards *Bluetooth* zu arbeiten: drahtlose Übermittlung von Daten und Sprache per Funk. Inzwischen zählen sich mehr als 2000 Unternehmen zu dieser Interessensgemeinschaft.

2. Systemübersicht

Bluetooth versendet und empfängt seine Daten im Frequenzbereich des weltweit verfügbaren und lizenzfreien 2,4-GHz-ISM-Bandes. Insgesamt stehen 79 MHz Bandbreite im Frequenzbereich von 2402 – 2480 MHz zur Verfügung, welche in 79 Kanäle zu je 1 MHz Bandbreite aufgeteilt ist. Aufgrund eines schmaleren Bandes arbeitet Bluetooth in Frankreich, Spanien und Japan allerdings nur auf 23 Kanälen. Als Spread-Spectrum-Verfahren kommt *Frequency-Hopping* zum Einsatz. Alle Teilnehmer eines Netzwerks führen 1600 Hopps/s durch und können maximal 1 Mbit/s übertragen und empfangen. Zur Modulation wird das einfache *Gaussian Frequency Shift Keying (GFSK)* verwendet. Für ihre Sendeleistung werden die Geräte in drei Klassen unterteilt:

- Klasse I: 100 mW Sendeleistung 100 m Reichweite
- Klasse II: 2,5 mW Sendeleistung 50m Reichweite
- Klasse III: 1 mW Sendeleistung 10m Reichweite

Entgegengesetzt zum 802.11-Standard schließen sich die Geräte *immer* zu einem Ad-Hoc-Netzwerk zusammen. Sobald zwei oder mehr Einheiten miteinander kommunizieren handelt es sich um ein sog. *Piconetz*. Eine Einheit übernimmt dabei die Rolle des *Master*, der Kontakt zu den übrigen Einheiten, den *Slaves*, aufnimmt, zum Austausch von Daten auffordert und das Wechseln der Frequenzen im Frequency-Hopping reguliert. Maximal besteht ein Piconetz aus einem Master und sieben Slaves. Allerdings können mehrere Piconetze in einem Überlappungsbereich durch die pseudozufällige Auswahl der Frequenzen und den häufigen Frequenzwechsel koexistieren, ohne dass ihre Daten kollidieren. Desweiteren können sich mehrere Piconetze zu einem *Scatternetz* zusammenschließen. Eine Einheit bildet dann die Schnittstelle zwischen den Netzen. Eine genauere Beschreibung der Funktionsweise erfolgt im folgenden Kapitel.

Zur Vollduplex-Übertragung von Daten in jede Richtung wird ein Zeitmultplex-Verfahren eingesetzt. Die Arten der Verbindungen unterscheiden sich dabei in *Synchronous Connection Oriented (SCO)* und *Asynchronous Connection Less (ACL)* für synchrone und asynchrone Datenübermittlung.

3. Netzwerktypen

3.1 Piconetz

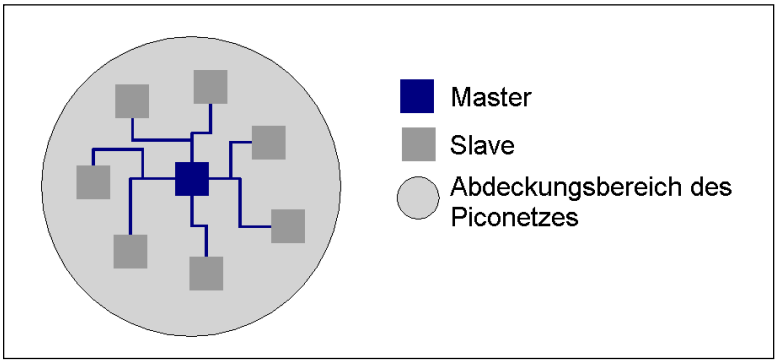


Abb. 3.1: Aufbau eines Piconetzes

Als Piconetz bezeichnet man eine Netzwerkstruktur aus maximal 8 Einheiten, bestehend aus einem Master und bis zu sieben *aktiven* Slaves. Darüber hinaus können sich bis zu 255 *geparkte* Slaves in das Netzwerk integrieren, ohne allerdings Daten mit den aktiven Einheiten austauschen zu können. Die Hauptzustände, die eine Einheit annehmen kann, teilen sich auf in:

- **Connection**
  Zwei Einheiten sind miteinander gekoppelt und können untereinander Daten austauschen. Nur in diesem Zustand kann eine Einheit in einen der später erwähnten *Stromsparzustände* wechseln.
- **Standby**
  Wenn sich eine Einheit nicht im Connection-Zustand befindet, steht sie auf Standby und kann nur durch den Master des Piconetzes durch *Inquiry*- oder *Page-Scans* wieder aufgeweckt werden.

Darüber hinaus sind sechs weitere Zustände definiert, die dem Piconetz die Aktivität der Geräte mitteilen und zusätzlich für deren Energiehaushaltung verantwortlich sind:

- **Active**
  Ein Slave wartet in diesem Zustand auf Pakete vom Master, um danach seinerseits Daten versenden zu können.
- **Sniff**
  Eine Einheit in diesem Zustand ist immer noch mit dem Master verbunden, wartet allerdings nicht mehr auf dessen Pakete und kann über seine Adresse angesprochen werden. Die Stromaufnahme des Gerätes ist wesentlich geringer.
- **Hold**
  Die Übertragungstätigkeit einer Einheit wird in diesem Zustand für eine vom Master vordefinierte Zeit eingestellt. Der Slave kann diesen Zustand vom Master erfragen, wenn er beispielsweise an einem anderen Piconetz teilnehmen möchte.
- **Park**
  In diesem Zustand ist eine Einheit nicht mehr aktiv an der Kommunikation und somit am Piconetz beteiligt. Zwar hält ein Slave weiterhin die Synchronisation der Frequenzen mit dem

Master aufrecht, tauscht aber keine Daten mit diesem aus. Um aus diesem Zustand zurückzukehren, ist eine hohe Reaktionszeit erforderlich.

- **Inquiry**
  Eine Einheit sendet bei unbekannter Adresse von Gegenstellen eine *Inquiry*-Nachrichten, um festzustellen, mit welchen Einheiten eine Verbindung aufgenommen werden kann.
- **Page**
  Sind die Adressen der Geräte innerhalb der Reichweite eines Gerätes bereits bekannt, so kann ein Gerät im Zustand Page Verbindungen zu diesen aufbauen. Ein Master sendet hierzu eine Page-Nachricht auf 16 reservierten Frequenzen. Maximal vergehen nach Spezifikation 2,56 Sekunden, bis ein Master ein Slave erreicht, im Durchschnitt sind es allerdings 0,64 Sekunden.

Festzustellen ist an dieser Stelle, dass ein Slave nur Daten mit dem Master austauschen kann, nicht aber mit anderen Slaves. Um seinerseits Daten zu den übrigen Einheiten des Piconetzes transferieren zu können, muss er mit dem Master die Rollen tauschen. Desweiteren befindet sich ein Slave standardmäßig im Standby-Modus, in welchem es alle 1,28 Sekunden 32 reservierte Frequenzen nach eingehenden Nachrichten aus dem eigenen Piconetz abtastet..

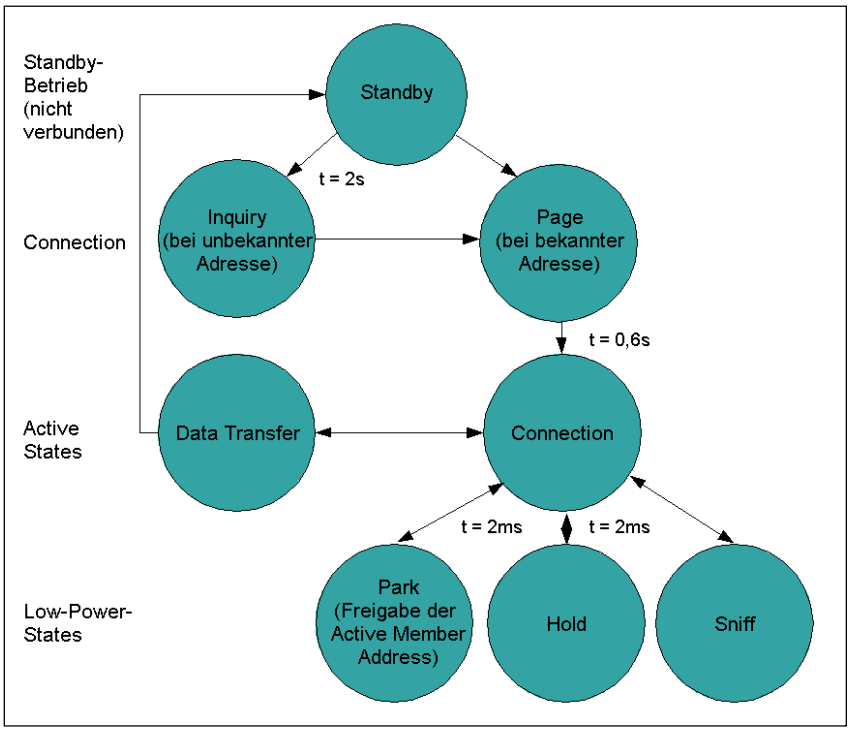


Abb. 3.2: Zustände eines Bluetooth-Gerätes

Im Sinne des Stromsparens sind die Zustände Active, Sniff, Hold und Park hauptsächlich für mobile Endgeräte interessant, denn dort sind die Ressourcen meistens knapp.

3.2 Scatternetz

Mehrere Piconetze können zu einem Scatternetz zusammengeschlossen werden. Die Netze werden über eine gemeinsame Einheit miteinander gekoppelt. Sie kann sowohl ein Master als auch ein Slave

sein, aber nicht in mehreren Piconetzen als Master fungieren. Befinden sich mehrere Piconetze im selben Abdeckungsbereich ist der Datendurchsatz höher.

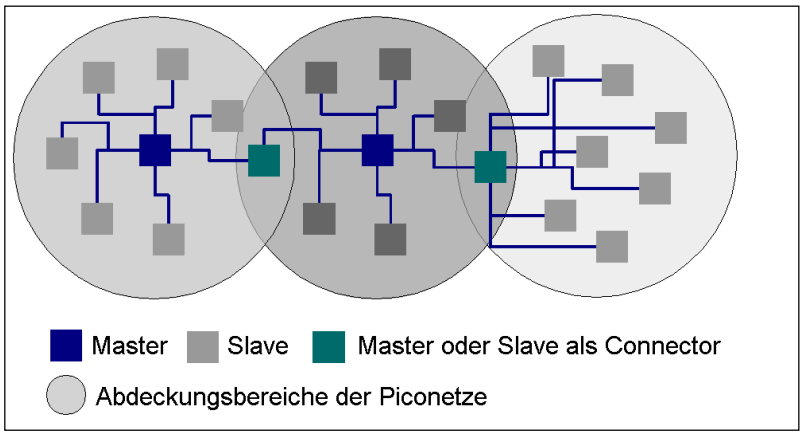


Abb. 3.3: Aufbau eines Scatternetzes

4. Bluetooth-Protokoll-Stack

Der Protokoll-Stack von Bluetooth regelt die Kommunikation zwischen den Geräten und dient hauptsächlich dem Auffinden anderer Geräte und derer angebotenen Services. Bei der Spezifikation wurde darauf geachtet, dass soviele bereits bestehende Protokolle wie möglich integriert werden. Der Stack teilt sich in vier Gruppen von Protokollen auf:

- **Bluetooth Core Protocols**
  Baseband, LMP, L2CAP und SDP
- **Cable Replacement Protocols**
  RFCOMM
- **Telephony Control Protocols**
  TCS-Binary, AT-Commands
- **Adopted Protocols**
  PPP, UDP, TCP, IP, WAP, OBEX

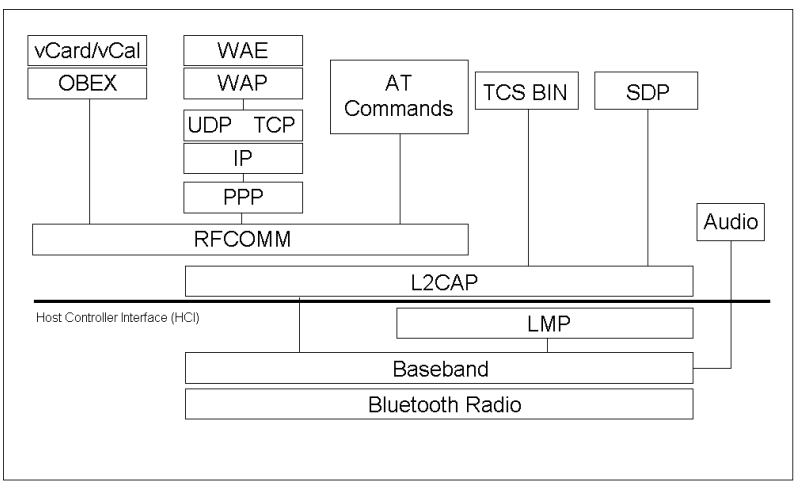


Abb. 4.1: Protokollschichten der Bluetooth-Schnittstelle

5. Bluetooth Radio

Unter Bluetooth Radio versteht man eine Art „Luftschnittstelle“, die die Art der Verbindungsaufnahme über die Funkwellen beschreibt. Mittels des Frequency-Hopping-Spread-Spektrums, das mit einem Zeitmultiplexer arbeitet, werden die Daten übertragen. Bluetooth verwendet hierbei Slow-Hopping. Jedes Paket wird zur Übertragung in einen einzelnen *Time Slot* gesteckt *(Single Slot)*. Aufgrund des häufigen Frequenzwechsels von 1600 Hopps / sec, den alle Slaves synchron mit dem Master ausführen, sind die Pakete daher sehr klein. Umgerechnet finden etwa 0,00016 Hopps / bit statt. Zusätzlich macht dies Bluetooth robuster gegen Störsignale. Eine Überlappung der Frequenzen ist statistisch gesehen sehr unwahrscheinlich.

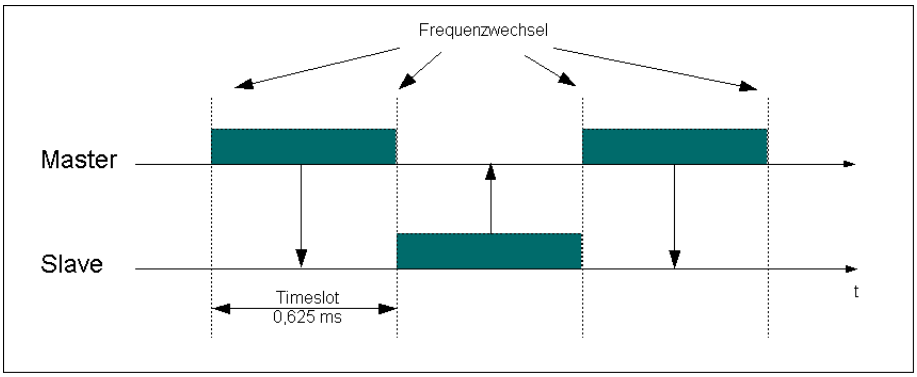


Abb. 5.1: Frequenzwechsel und Aufteilung der Time Slots beim Senden und Empfangen

5.1 Verbindungstypen

Da Bluetooth mehreren Anwendungszwecken gerecht werden muss und lediglich 1 Mbit/sec an Übertragungsvolumen zur Verfügung steht, gibt es zwei Arten von Verbindungen: synchrone und asynchrone. Ebenso wird unterschieden, ob der Master mit einem oder mehreren Slaves kommuniziert.

Synchronous Connection Oriented (SCO-Link)

Der SCO-Link wird für sog. *Point-to-Point*-Verbindungen verwendet, wenn ein Master mit einem einzelnen Slave Daten austauscht. Besonders für zeitkritische Übertragungen wie etwa Sprache muss gewährleistet sein, dass die Pakete synchron gesendet und empfangen werden. Kommt es auf dem Übertragungsweg zu einem Datenverlust, kann ein Paket nämlich nicht wiederholt gesendet werden. Aufgrund dieser Priorität reserviert der Master Time Slots und somit Bandbreite für die Übertragung. Maximal kann ein Slave drei SCO-Links eines Masters verwalten und maximal zwei zu verschiedenen Slaves von einem Master.

Asynchronous Connection Less (ACL-Link)

Nimmt ein Master zu mehreren Slaves Kontakt auf, spricht man von einer *Point-to-Multipoint*-Verbindung. Hierzu werden die übrigen nicht von den SCO-Links reservierten Time Slots verwendet. Für jede Master-Slave-Verbindung steht nur ein ACL-Link zur Verfügung. Einen Ausgleich zu SCO schafft aber die Möglichkeit, dass sich bei Paketverlust Daten wiederholt senden lassen.

6. Baseband

Das Baseband setzt auf dem Radio Layer im Link Controller auf und besitzt vielseitige Aufgaben:

- Steuerung der physikalischen Funkverbindung (SCO und ACL)
- Encoding und Decoding der Datenpakete
- Festlegung der Hopp-Sequenz
- Fehlerkorrektur
- Datentransfer
- Verwalten der logischen Verbindungen
- Adressierung
- Sprach- und Audio-Kommunikation
- Authentisierung, Autorisation und Verschlüsselung

6.1 Standardpakete

Ein versendetes Paket besitzt nach der Bluetooth-Spezifikation den in Abb. 6.1 gezeigten Aufbau und besteht aus einem 72 Bit *Access Code*, einem 54 Bit *Header* und einer *Payload* zwischen 0 und 2745 Bits. Innerhalb eines Piconetzes ist der Access-Code für alle Pakete gleich. Bei einem ungültigen Access-Code wird das Paket abgelehnt. Die Typen eines Access-Codes sind wie in Tab. 6.1 aufgelistet zu unterscheiden.

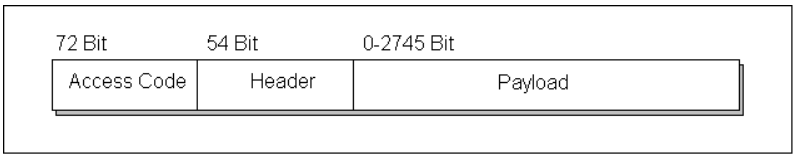


Abb. 6.1: Aufbau des Standardpakets

Typ	Bedeutung
Channel Access Code (CAC)	Identifizierung des zugehörigen Piconetzes
Device Access Code (DAC)	Spezielle Signalisierungen
Inquiriy Access Code (IAC)	Erkennung von in Reichweite liegenden Geräten (Unterteilung in G(eneral)IAC & D(evice)IAC)

Tab. 6.1: Access-Code-Typen

Der Header liefert Steuerinformationen und ist regulär 18 Bit lang. Er wird allerdings aus Sicherheitsgründen 3fach gesendet, was die resultierenden 54 Bit erklärt. Der Header teilt sich nun auf in:

Typ	Bedeutung
AM_ADDR	Unterscheidung aktiver Slaves (Active Member Address)
TYPE	Paketart (ACL & SCO)
FLOW	Flusssteuerungsbit für Pakete (Schutz vor BufferOverflow; nur für ACL gültig)
ARQN	Empfangsbestätigung
SEQN	Sequence-Number zur Ordnung der Daten im Datenstrom und zur Erkennung verloren gegangener

	Pakete
HEC	8-Bit CRC-Code für einen fehlerfreien Empfang des Headers (Header Error Check)

Tab. 6.2: Aufteilung des Headers

6.2 Paketarten

Die Arten von Paketen sind nach der Art der Verbindung zu unterscheiden. ACL und SCO besitzen jeweils sowohl eigene und als auch gemeinsame Paketarten.

Geimsame Kontrollpakete für ACL- und SCO-Verbindungen

ID-Paket
Das Paket ist ein 68 Bit großer Access-Code, der entweder ein DAC oder IAC sein kann, und dient als Antwortpaket für Response-, Paging- und Inquiring-Anfragen.

NULL-Paket
Das NULL-Paket ist 126 Bit groß und aufgeteilt in einen 72 Bit Access-Code, der einen CAC beinhaltet, und einen 54 Bit Header. Das Paket übermittelt Informationen zum Status der Verbindung, wie etwa eine Empfangsbestätigung (ARQN) oder den Status des Empfängerpuffers (FLOW). Dieses Paket wird vom Empfänger nicht bestätigt.

POLL-Paket
Dieses Paket ist wie das NULL-Paket aufgebaut, erfordert allerdings eine Empfangsbestätigung. Der Master spricht hiermit die Slaves im Piconetz an. Das Slave muss daraufhin immer antworten, auch wenn es keine Daten zu versenden hat. Steuerinformationen wie ARQN oder FLOW werden allerdings nicht ausgewertet.

FHS-Paket
Das FHS-Paket ist ein Kontrollpaket bestehend aus einem 144 Bit großen Informationsfeld (*Payload)* und einem 16 Bit CRC. Um es vor Übertragungsfehlern zu schützen, wird es zusätzlich mit einer *Forward Error Correction (FEC)* codiert, wodurch sich seine Größe auf 240 Bit ausdehnt. Es dient etwa der Beantwortung von Paging- oder Inquiry-Paketen oder der Ausführung des Master-Slave-Rollentausches, bei welchem die Memberadressen ausgetauscht werden. Zusätzlich reguliert es die Synchronisation der Frequenzsprünge vor dem Aufbau eines Piconetzes und enthält alle für die Verbindung relevanten Daten.

DM1-Paket
Dieses Paket enthält lediglich Steuerinformationen.

SCO-Pakete

Typischerweise sind SCO-Pakete Single-Slot-Pakete (Belegung eines Time-Slots) und enthalten keinen CRC, da eine Überprüfung auf Korrektheit bei Audiodaten, für welche diese Verbindung gedacht ist, hinfällig ist. Ein SCO-Paket wird nur einmalig versendet, ohne dass eine Empfangsbestätigung gesendet wird. Hauptsächlich werden bei einer SCO-Verbindung sog. *High-*

Quality-Voice-Pakete für Sprachübertragung verwendet. Die Größe eines SCO-Pakets beträgt fixe 240 Bit, trägt den Bezeichner *Voice Field* und wurde vor der Übertragung mit einem einer FEC geschützt.

HV1-Paket
Dieses Paket trägt 10 Byte an Informationen und beinhaltet 1,25 ms Audio bei 64 kBit/s. Es wird in jedem zweiten Time-Slot übertragen.

HV2-Paket
Dieses Paket trägt 20 Byte an Informationen und beinhaltet somit 2,5 ms Audio bei 64 kBit/s. Es wird in jedem vierten Time-Slot übertragen.

HV3-Paket
30 Byte an Informationen und somit 3,75 ms Audio bei 64 kBit/s überträgt dieses Paket in jedem sechsten Time-Slot. Es ist nicht durch einen FEC geschützt.

DV-Paket
Ein DV-Paket überträgt sowohl Audio also auch Daten. Seine Payload besteht daher nicht nur aus dem Voice Field, das auf 80 Bit verkleinert ist, sondern zusätzlich aus einem bis zu 150 Bit großen *Data Field*. Dieses Datenfeld setzt sich wiederum aus 10 Byte Informationen, ein Byte Payload und einem 16 Bit CRC zusammen. Beide Felder werden getrennt voneinander behandelt. Das Datenfeld kann im Gegensatz zum Voice Field bei fehlerhafter Übertragung erneut gesendet werden.

ACL-Pakete

Für asynchrone Verbindungen stehen sieben verschiedenen Arten von Paketen zur Verfügungen. Alle Pakete, bis auf das *AUX1*, besitzen einen CRC und werden bei Fehlern erneut übertragen. Die Payload setzt sich aus einem Payload Header, einem Payload Body und einem CRC zusammen.

DM1-Paket
DM ist die Abkürzung für *Data-Medium-Rate* und steht für ein Paket, dass ausschließlich Daten überträgt. Es ist aufgeteilt in einen 18 Byte Payload einschließlich 1 Byte Header und einen 16 Bit CRC. Es ist durch eine FEC geschützt.

DH1-Paket
Die Abkürzung *DH* steht für *Data-High-Rate* und signalisiert so bereits die höhere Payload von 28 Byte. Das Paket ist nicht durch FEC geschützt.

DM3-Paket
Die Payload dieses Pakets beträgt 123 Byte und inkludiert einen 2 Byte Header. Eine FEC bietet Schutz vor Übertragungsfehlern.

DH3-Paket
Für dieses Paket beträgt die Payload 185 Byte und wie *DM3* besitzt es einen 2 Byte Header. Eine FEC fehlt.

DM5-Paket
Die Payload beträgt 226 Byte inklusive einem 2 Byte Header. FEC schützt.

DH5-Paket
341 Byte groß ist hier die Payload mit einem 2 Byte Header. Das Paket ist nicht mittels FEC geschützt.

AUX1-Paket
Das Paket ist wie *DH1* aufgebaut, besitzt allerdings keinen CRC. Die Playload vergrößert sich somit auf 30 Byte.

Transferraten

Je nach Transfer- und Paketart einer ACL-Verbindung entstehen unterschiedliche Transferraten. Einen Überblick über theoretisch mögliche Datenraten schafft Tab. 6.3:

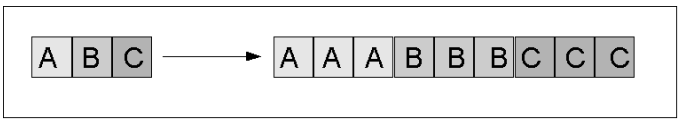
Pakettyp	Symmetrisch	Asymmetrisch	
DM1	108,8 kbit/s	108,8 kbit/s	108,8 kbit/s
DH1	172,8 kbit/s	172,8 kbit/s	172,8 kbit/s
DM3	256 kbit/s	387,2 kbit/s	54,4 kbit/s
DH3	384 kbit/s	585,6 kbit/s	86,4 kbit/s
DM5	286,7 kbit/s	477,8 kbit/s	36,3 kbit/s
DH5	432, 6 kbit/s	723,2 kbit/s	57,6 kbit/s

Tab. 6.3: Transferraten bei ACL-Verbindungen

6.3 Fehlerkorrekturen

Aufgrund der nicht geringen Fehleranfälligkeit bei Funkübertragungen sieht die Spezifikation Maßnahmen zur Fehlerkorrektur bzw. –prävention für die Pakete vor *(Forward Error Correction)*, doch nicht jedes Gerät muss zwangsweise all seine versendeten Pakete schützen. Ein Paket kann immer mit oder ohne *FEC* versendet werden. Vorgeschrieben ist lediglich der Schutz der Header, da diese Informationen über das komplette Paket enthalten. Zu unterscheiden sind:

- **1/3 FEC**
  Jedes Bit einer Information wird dreimal wiederholt.



- **2/3 FEC**
  Die Information wird mittels eines gekürzten Hamming-Codes (Fire-Code) geschützt.[1]
- **Automatic Request Scheme (ARQ)**
  Ein mit einem CRC versehenen Paket wird bis zu seiner Bestätigung wiederholt gesendet (nicht bei Headern und *Voice Payload*).

[1] Weitere Informationen unter http://de.wikipedia.org/wiki/Hamming-Abstand

6.4 Management des Funkkanals

In einem Piconetz wird der Funkverkehr hauptsächlich vom Master bestimmt. Zum Master wird diejenige Einheit ernannt, die als erstes versucht, Kontakt zu den anderen Einheiten des Netzes aufzunehmen. Der Master legt den Kanal fest, auf welchem die Kommunikation stattfindet. Zusätzlich bestimmt die sog. *Bluetooth-Device-Address* (BD_ADDR) die Frequency-Hopping-Sequenz und den Channel Access Code.

Das Timing der Kommunikation wird zudem durch die interne Systemzeit des Masters bestimmt. Diese Zeit wird wird den Slaves mitgeteilt, die ihrerseits einen Offset zu ihrer eigenen Systemzeit addieren, um die Synchronisation mit dem Master zu erreichen. In regelmäßigen Abständen schickt der Master ein Poll zu den Slaves, um die Synchronisation aufrechtzuerhalten.

Bluetooth-Adressierung

Jedes Gerät in einem Piconetz besitzt eine eindeutige, von seinem internen Zustand abhängige Geräteadresse, über die es von den anderen Einheiten angesprochen werden kann.

Bluetooth Device Address (BD_ADDR)
Diese 48 Bit große Adresse ist eine nach dem IEEE 802-Standard festgelegte Adresse, die fest im ROM eines jeden Bluetooth-Gerätes vom Hersteller integriert worden ist. Sie ist nachträglich nicht zu ändern und kann als MAC-Adresse bezeichnet werden.

Active Member Address (AM_ADDR)
Diese Adresse wird jedem Slave eines Piconetzes dynamisch von seinem Master bei Einleitung einer Verbindung zugeteilt. Sie ist 3 Bit groß und wird für Broadcasts mit Nullen gefüllt. Ein Slave akzeptiert nur dann Pakete, wenn sie entweder seine eigene passende AM_ADDR oder die Broadcast-Adresse beinhalten. Der Master selbst verfügt nicht über solch eine Adresse.

Parked Member Address (PM_ADDR)
Wenn ein Master einen Slave in den PARK-Zustand versetzt, verfällt dessen Active Member Address und er bekommt stattdessen eine 8 Bit große PM_ADDR zugewiesen. Sobald die Einheit wieder aufgeweckt wird, erhält sie wieder eine AM_ADDR. Besteht die Adresse nur aus Nullen, so kann die Einheit nur durch die BD_ADDR in das aktive Netz zurückgeholt werden.

Access Request Address (AR_ADDR)
Um die Synchronisation mit dem Master aufrecht zu erhalten und Access-Request-Nachrichten im passenden Time-Slot versenden zu können, wird einem geparkten Slave zusätzlich eine AR_ADDR mitgegeben. Diese Adresse ist nicht eindeutig und kann von mehreren Slaves verwendet werden.

6.5 Verbindungsaufbau

Für das im Sequenzdiagramm in Abb. 6.2 vorgestellte Beispiel für den Verbindungsaufbau zwischen zwei Einheiten sind die Aktionen *Inquiry Scan* und *Paging Scan* noch vorzustellen.

Inquiry Scan
In regelmäßigen Abständen (alle 2 Sekunden) führt ein Gerät einen sog. *Inquiry Scan* durch, währenddessen es nach den Anfragen anderer Geräte zum Verbindungsaufbau sucht. Es werden ID-Pakete mit passendem Inquiry Access Code erwartet, auf dessen Empfang das Gerät in den Zustand *Inquiry Response* wechselt und seinerseits mit einem FHS-Paket antwortet.

Page Scan
Ähnlich wie beim Inquiry Scan wartet auch hier ein Gerät auf eine Anfrage. Diesmal allerdings sollte ein empfangenes Paket den eigenen Device Access Code beinhalten, woraufhin die Einheit in den Zustand *Slave Response* wechselt und ein *Acknowledge (ACK)* versendet.

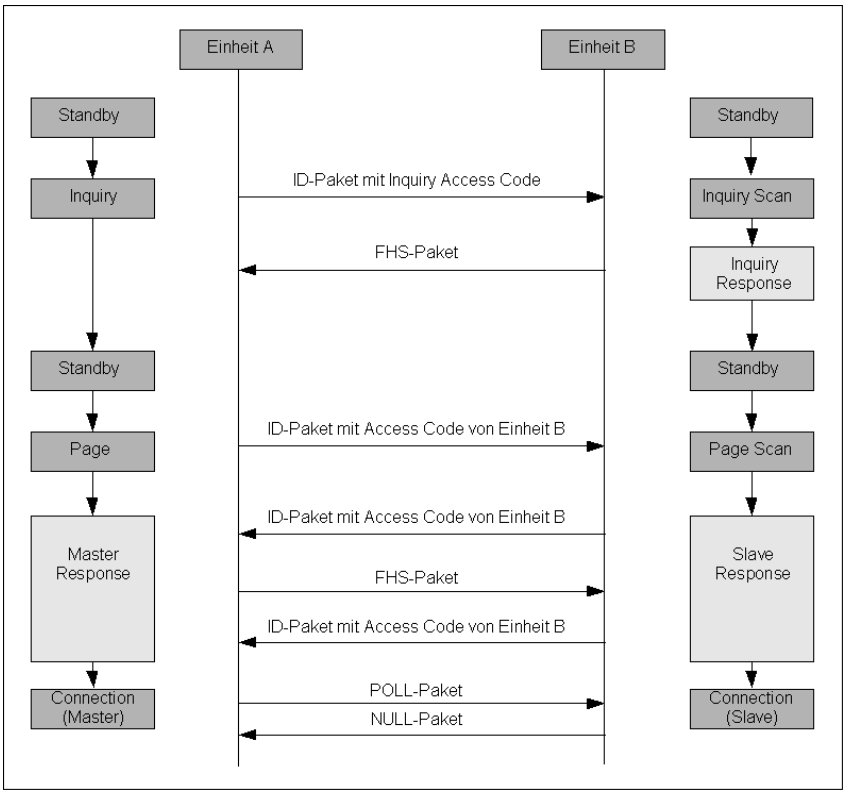


Abb. 6.3: Sequenzdiagramm eines Verbindungsaufbaus zwischen zwei BT-Geräten

6.6 Bluetooth Audio

Als schwierig erweist sich immer die Übertragung von Sprache. Zum einen muss dies synchron erfolgen, da niemand nach einem Zeitmuster spricht, und zum anderen dürfen Daten nicht verloren gehen, da sie nicht erneut gesendet werden können. Daher werden die Daten über eine synchrone SCO-Verbindung übertragen. Als Kompressionsverfahren stehen hierzu *Pulse Code Modulation (PCM)* und *Continuous Variable Slope Delta Modulation (CVSD)* mit einer Datenrate von jeweils 64 kBit/s zur Verfügung. Ist ein Piconetz stark von Interferenzen beeinträchtigt, ist *CVSD* besser geeignet, da selbst bei hohen Bitfehlerraten die Empfangsqualität nur durch Hintergrundrauschen gestört wird.

6.7 Sicherheit

Die Spezifikation unterscheidet drei verschiedene Sicherheitsstufen:

Sicherheitsstufe	Maßnahmen
Security Mode 1 (non-secure)	Keine
Security Mode 2 (service level enforced)	Nach Verbindungsaufbau; abhängig von verwendetem Service und Kanal
Security Mode 3 (link level enforced)	Vor Verbindungsaufbau; z.B. Ablehnung von Verbindungen

Tab. 6.4: Sicherheitsstufen

Jede Sicherheitsmaßnahme, die bei der Kommunikation zweier Einheiten eingeleitet wird, basiert auf einem 128 Bit großen Verbindungsschlüssel *(Link Key)*, der im Grunde nicht mehr ist als eine Zufallszahl. Mit diesem Schlüssel wird der Sitzungsschlüssel *(Encryption Key)* generiert und die Einheiten bei ihrer Gegenstelle authentifiziert. Der Link Key ist in vier Arten zu unterteilen:

Schlüsselart	Merkmale	Verwendung
Kombinationsschlüssel (Combination Key)	Kombination aus den Schlüsseln zweier Geräte	Generierung des Sitzungsschlüssels
Geräteschlüssel (Unit Key)	Intern erzeugter, für ein einziges Gerät festgelegter Schlüssel	
Temporärschlüssel oder Masterschlüssel (Temporary Key)	Nur gültig für aktuelle Sitzung und für Master	Aufbau zweier Verbindungen zu zwei Slaves gleichzeitig
Initialisierungsschlüssel (Initialisation Key)	Erzeugt aus Zufallszahl, PIN-Code und Geräteadresse (BD_ADDR) des Senders	Schutz der Verbindungsparameter bei der Übertragung

Tab. 6.5: Schlüsselarten

Der Sitzungsschlüssel dient der Verschlüsselung der Daten und wird aus Verbindungsschlüssel, einer früher generierten 96 Bit *Ciphering Offset Number (COF)* und einer weiteren 128 Bit Zufallszahl generiert. Die entstehenden Schlüssel werden niemals übertragen, sondern immer nur intern aus den übermittelten Daten während der Initialisierungsphase generiert. Dies bedeutet, dass die Einheiten in einem Piconetz sich immer gegenseitig bekannt sein und vertrauen müssen.

Lower Level Security (Link Level Security Mode 3)

Bevor eine Verbindung zwischen zwei Einheiten zustande kommen kann, muss sich die anfragende Einheit *(Claimant)* bei der Gegenstelle *(Verifier)* authentifizieren. Hierzu wird ein *Challenge Response Verfahren* verwendet. Der Verifier übergibt dem Claimant eine Zufallszahl. Mittels Link Key und Bluetooth Device Address wird hieraus ein Code erzeugt, der an den Verifier zurückgeschickt und dort mit dem eigenen erzeugten Code abgeglichen wird. Der verwendete Link Key muss zuvor beim sog. *Pairing* oder *Bonding*, dem erstmaligen Koppeln beider Geräte, ausgetauscht und gespeichert worden sein.

Mittels eines *Stream Ciphers*, der für jedes Paket aus Bluetooth Device Address des Masters, dessen interner Systemzeit und dem Sitzungsschlüssel erzeugt wird, werden anschließend die Payloads der Pakete verschlüsselt. Access Code und Header bleiben unverschlüsselt.

Higher Level Security (Service Level Enforced Mode 2)

Neben der Verschlüsselung von Daten an sich möchte man den Zugriff auf die Dienste einer Einheit kontrollieren können.

Trusted Units
Dies sind Einheiten, denen vertraut wird und somit Zugriff auf alle angebotenen Dienste eines Gerätes erteilt wird.

Untrusted Units
Einheiten, denen man nicht vertraut, sind nur beschränkte Zugriffe erlaubt.

Die Dienste können laut Spezifikation zudem ohne Zugriffsschutz, mit Schutz durch Authentifikation und mit Schutz durch Autorisation und Authentifikation ausgestattet werden.

6.8 Profiles

Die *SIG* spezifiziert Protokolle *(Profiles)* für das Kommunikationsverhalten zwischen den Geräten und ihren Anwendungen, deren Liste ständig erweitert wird[2]. Wenn zwei Einheiten miteinander kommunizieren und Daten austauschen möchten, müssen sie beide die Protokolle nutzen und die Spezifikation erfüllen. Eine kurze Übersicht über die wichtigsten Profile bietet Abb. 6.4.

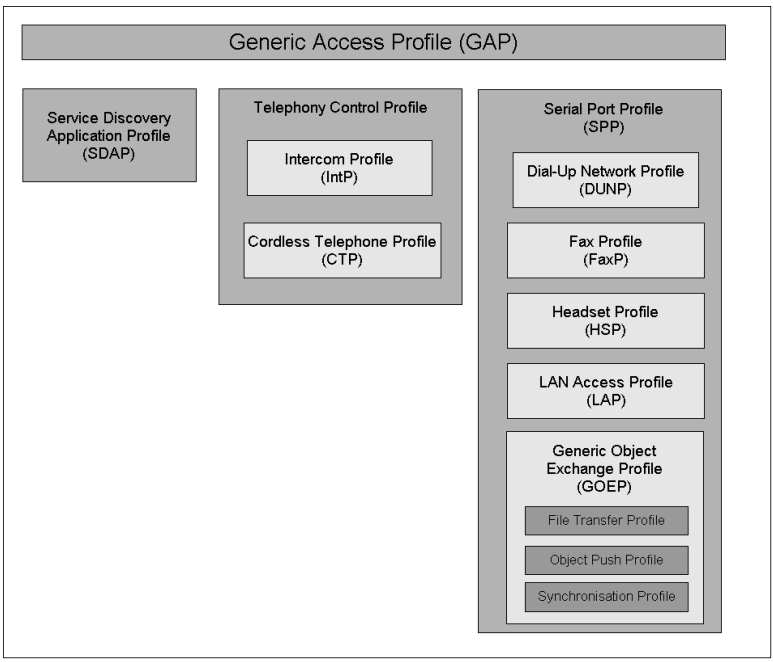


Abb. 6.4: Auswahl der wichtigsten Service Profiles[3]

[2] https://www.bluetooth.org/spec/

[3] Glossar zu den gängigsten Profilen: Heise Bluetooth DB: http://www.heise.de/mobil/bluetooth/db/default.phtml?a=glossar

Die Profile sind ineinander geschachelt und teilweise voneinander abhängig. Neben dem übergeordneten *Generic Access Profile (GAP)* ist das *Service Discovery Application Profile (SDAP)* als wichtigstes Profil zur gegenseitigen Erkennung und Authentifizierung spezifiziert. Darüber hinaus wacht es über die Dienste, die ein Gerät und dessen Kommunikationspartner zur Verfügung stellen.

Aussicht

Ebenso wie 802.11 wird auch Bluetooth ständig weiterentwickelt. Seit November 2003 liegt der Standard in Version 1.2 vor. Dort sind Verbesserungen hinsichtlich Verbindungsaufbau, Sprachübertragung und Quality of Service vorgesehen. Desweiteren soll ein *Adaptive Frequency Hopping* die Störung von 802.11-Netzen verringern. Dank seiner effektiven Modulation ist Bluetooth zwar selbst robust gegen andere Funksignale auf der 2,4 GHz-Frequenz im ISM-Band, beeinflusst aber den Funkverkehr anderer Netze.

Die SIG berücksichtigt fortlaufend neue Produkt- und Anwendungsideen mit der Spezifikation neuer Profile. Die Erweiterbarkeit ist daher gewährleistet, ohne dass eine neue generelle Spezifikation verabschiedet werden muss. Alle neu entwickelten Geräte müssen vor ihrer Marktreife von einer unabhängigen Abteilung für Qualitätssicherung der SIG begutachtet und beglaubigt werden. Dennoch ist damit nicht garantiert, dass zwei Geräte, die gleiche Anwendungsprofile besitzen und dahingehend zueinander kompatibel sind, auch miteinander kommunizieren können.

Bisher sind weltweit ca. 1200 Geräte registriert, die Bluetooth integrieren. Diese Zahl wächst stetig an. Dennoch scheint der Standard bei Endanwendern nur langsam beliebt zu werden. Der „Markenname“ Bluetooth ist zwar in aller Munde, doch nur die wenigsten nutzen ihn im Alltag.

Fazit

Sicherlich ist aufgefallen, dass die Protokolle und Mechanismen von Bluetooth nur kurz angerissen sind. Es wurde deshalb darauf verzichtet, auf zu viele Details einzugehen, um den Vergleich zu möglichst vielen im Seminar vorgestellten Themen zu 802.11 zu ermöglichen und vor allem die unterschiedlichen Funktionsweisen hinsichtlich Aufbau von Netzwerken, Paketen und Sicherheitsmechanismen zu verdeutlichen. Die Diskussion hierzu kann sicherlich besser im Seminar stattfinden. Es sei nur soviel gesagt: Bluetooth macht (fast) alles anders als 802.11.

802.11 und Bluetooth sind sicherlich keine konkurrienden Verfahren zum Aufbau drahtloser Netzwerke. Bluetooth wurde aus einer ganz anderen Motivation heraus entwickelt, nämlich um eine einzige Infrastruktur zwischen vielen verschiedenen Peripheriegeräten im kleinen Rahmen zu schaffen, die sich gegenseitig spezifische Dienste zur Verfügung stellen. 802.11 ist dagegen auf den Aufbau größerer Netzwerke und den Anschluss drahtloser Stations an ein bestehendes Ethernet ausgerichtet. Bluetooth wird 802.11 daher nicht ersetzen, sondern nur das Spektrum der Möglichkeiten mit Funktechnik erweitern können, schon allein deshalb, weil die Übertragungskapazität auf 1 Mbit/s beschränkt ist.

Literatur & Internetquellen

Bluetooth – The Official Website
http://www.bluetooth.com

Bluetooth – The Official Bluetooth Membership Website
http://www.bluetooth.org

Bluetooth Core Specification v1.1
https://www.bluetooth.org/spec/

Bluetooth Core Specification v1.2
https://www.bluetooth.org/spec/

Gmür, Chrigi
Bluetooth, ELKO- das Elektronik-Kompendium, http://www.elektronik-kompendium.de/public/chrigi/bluetooth.htm , zuletzt gesichtet am 18. Januar 2004

Hascher, Wolfgang
Bluetooth – der Kabel-Killer, TecChannel, http://www.tecchannel.de/hardware/477/ , 2000

Merkle, A. & Terzis, A.
Digitale Funkkommunikation mit Bluetooth, Franzis' Verlag, Poing, 2002

Sappok, Sören & Zivadinovic, Dusan
Profile in Blau - Die Protokolle des Kurzstrecken-Funks Bluetooth, c't Ausgabe 18/02, Heise Verlag, 2002

Zivadinovic, Dusan
Privat-Funk - Bluetooth als Netzwerk- und Schnittstellenmodul, Heise mobil, http://www.heise.de/mobil/artikel/2003/02/26/privatfunk/ , 2003

Zivadinovic,Dusan & Özkilic, Murat
Mini-Netze - LAN und Internet: PDAs und Notebooks per Funk ankoppeln, c't Ausgabe 20/02, Heise Verlag, 2002