Bluetooth

7

Kabelverbindungen eignen sich vor allem für den stationären Einsatz, sind aber im mobilen Umfeld nicht sehr flexibel. Die Bluetooth Funktechnologie bietet hier eine ideale Lösung für viele Anwendungsfälle. Um aufzuzeigen, welche Möglichkeiten Bluetooth bietet, gibt Kap. 6 zunächst einen Überblick über die physikalischen Eigenschaften des Systems, sowie den Aufbau und die Funktionsweise des Protokollstacks. Im weiteren Verlauf führt das Kapitel dann in das Konzept der Bluetooth Profile ein und demonstriert deren praktische Funktionsweise und große Anwendungsvielfalt.

7.1 Überblick und Anwendungen

Durch die fortschreitende Miniaturisierung finden heute zunehmend kleine elektronische Geräte Einzug in das tägliche Leben. Mit Bluetooth können diese Geräte drahtlos und ohne direkte Sichtverbindung miteinander kommunizieren. Während im letzten Jahrzehnt Bluetooth für eine Vielzahl unterschiedlicher Anwendungsfälle eingesetzt wurde, ist aktuell jedoch zu beobachten, dass sich der Einsatz dieser Technologie hauptsächlich auf die folgenden Anwendungsbereiche konzentriert:

- Kabellose Verbindung zwischen einem Smartphone und Audioendgeräten wie z. B. Kopfhörer, Freisprecheinrichtungen, Lautsprechern und Headsets für Sprachtelefonie und Musikwiedergabe.
- Austausch von Dateien (z. B. Bildern) zwischen verschiedenen Endgeräten.
- Anbindung von kabellosen Tastaturen und anderen Eingabegeräten an Notebooks, PCs und Smartphones.

Andere Anwendungen die z. B. die Weitergabe einer Internetverbindung zwischen einem Smartphone und einem Notebook, Kalendersynchronisation, Multiplayer-Spiele zwischen

M. Sauter, *Grundkurs Mobile Kommunikationssysteme*,
https://doi.org/10.1007/978-3-658-21647-4_7

mobilen Endgeräten, etc., wurden mittlerweile durch andere Technologien wie z. B. Wi-Fi Tethering und Cloud Dienste abgelöst.

Da heute eine Vielzahl von unterschiedlichen Herstellern Bluetooth Geräte entwickeln, ist eine einwandfreie Interoperabilität grundlegende Voraussetzung für eine reibungslose Kommunikation. Dies wird durch den Bluetooth Standard und Interoperabilitätstests sichergestellt. Nachfolgende Tabelle zeigt die bisher erschienenen Protokollversionen. Grundlegend gilt, dass jede neue Version zur alten Version abwärtskompatibel ist. Das bedeutet, dass ein Bluetooth 2.1 Gerät auch mit einem Bluetooth 3.0 Gerät einwandfrei zusammenarbeitet. Funktionalitäten, die mit einer neuen Version eingeführt wurden, können jedoch nicht zusammen mit älteren Geräten genutzt werden.

Version	Erschienen	Kommentar
1.0B	Dez. 1999	Erste Bluetooth Version, die aber nur von den Geräten der ersten Generation verwendet wurde
1.1	Feb. 2001	Diese Version korrigiert eine Reihe von Fehlern und Zweideutigkeiten der vorhergehenden Version des Standards (Errata List). Auf diese Weise wurde die Interoperabilität zwischen Geräten weiter verbessert
1.2	Nov. 2003	Diese Version führt einige neue Funktionalitäten ein. Die wichtigsten sind: Schnelleres Auffinden von Bluetooth Geräten im Empfangsbereich. Gefundene Geräte können jetzt auch nach der Empfangsqualität sortiert werden, siehe Abschn. 7.4.2. Schnellere Verbindungsaufnahme, siehe Abschn. 7.4.2. Adaptive Frequency Hopping (AFH), siehe Abschn. 7.3. Verbesserte Sprachübertragung z. B. für Headsets (eSCO), siehe Abschn. 7.4.1 und 7.6.4. Verbesserte Fehlererkennung und Flusskontrolle im L2CAP Protokoll. Neue Sicherheitsfunktionalität: Anonyme Verbindungsaufnahme
2.0	2004	Enhanced Data Rate (EDR): Erweitert die Bluetooth 1.2 Spezifikation um schnellere Datenraten. Siehe Abschn. 7.2 und 7.4.1
2.1	2007	Sicherheits- und Detailverbesserungen. Die wichtigsten sind: Secure Simple Pairing: Verbesserung der Sicherheit und Vereinfachung des Pairing Prozesses. Siehe Abschn. 7.5.2 Sniff-Subrating: Weitere Energiesparoption für aktive Verbindungen mit geringem Datenaufkommen. Siehe Abschn. 7.4.2 Erroneous Data Reporting für eSCO Pakete. Siehe Abschn. 7.4.1
3.0+ HS	2009	Verbesserungen bei der Sendeleistungssteuerung und Einführung des High Speed (HS) Modus, der für die Verbindungsaufnahme Bluetooth verwendet, für die Datenübertragung jedoch einen Wi-Fi Kanal. Zwar sind heute die meisten Produkte Bluetooth 3.0 kompatibel, der optionale HS Modus für Wi-Fi Datenübertragung findet bisher jedoch kaum Verbreitung
4.0	2010	Aufnahme von WiBree in den Bluetooth Standard als Low Energy Option und unter der Bezeichnung „Bluetooth Smart“ vermarket.
4.1	2013	Führt folgende Verbesserungen ein: LTE Koexistenz in benachbarten Bändern. Automatisches wiederaufnehmen der Verbindung bei kurzen Signalunterbrechungen. Ein Endgerät kann gleichzeitig als Low Engery Hub und Low Energy Endgerät dienen.

7.2 Physikalische Eigenschaften

Bevor die nächsten Abschnitte näher auf die Funktionsweise von Bluetooth eingehen, folgt hier nun zunächst ein Überblick über die wichtigsten technischen Daten:

Die maximale Datenrate eines Bluetooth Kanals wurde in den ersten Versionen des Standards zunächst auf 780 kbit/s festgelegt. Alle Endgeräte, die direkt miteinander kommunizieren, müssen sich diese Datenrate teilen. Die maximale Datenrate für einen einzelnen Teilnehmer ist deshalb von folgenden Faktoren abhängig:

- Anzahl der Endgeräte, die untereinander gleichzeitig Daten austauschen
- Aktivität anderer Endgeräte

Die höchste Geschwindigkeit aus Sicht eines einzelnen Endgerätes kann erreicht werden, wenn nur zwei Geräte miteinander kommunizieren und nur eines der beiden Geräte eine große Datenmenge zu übertragen hat. In diesem Fall beträgt die maximal mögliche Datenrate 723 kbit/s. Nach Abzug des Overheads ergibt dies eine Datenrate von etwa 650 kbit/s. Dem anderen Endgerät bleibt dann jedoch nur eine Datenrate von etwa 57 kbit/s. Diese Situation gibt es in der Praxis z. B. bei der Dateiübertragung recht oft. Bei dieser Anwendung hat eines der beiden Endgeräte sehr viele Daten zu übertragen, während das andere nur Anfragen und Empfangsbestätigungen schickt. Abb. 7.1 zeigt die möglichen Geschwindigkeiten für dieses Szenario im linken Teil der Grafik.

Möchten beide Endgeräte möglichst schnell senden, liegt die maximal mögliche Geschwindigkeit für beide bei jeweils etwa 390 kbit/s. Abb. 7.1 zeigt diese Situation in der Mitte der Grafik.

Kommunizieren mehr als zwei Endgeräte untereinander, sinkt die maximale Datenrate pro Endgerät weiter, falls alle Endgeräte gleichzeitig mit der maximalen Geschwindigkeit senden wollen. Abb. 7.1 zeigt dies auf der rechten Seite.

Im Jahre 2004 wurde Bluetooth um das Enhanced Data Rate (EDR) Modulationsverfahren erweitert, das Datenraten von bis zu 2178 Mbit/s ermöglicht. Mehr hierzu in Abschn. 7.4.1.

Um diese Übertragungsraten zu erreichen, verwendet Bluetooth einen Kanal im 2,4 GHz ISM (Industrial Scientific and Medical) Band mit einer Bandbreite von 1 MHz. Als Modulationsverfahren wird für normale Pakete das Gaussian Frequency Shift Keying (GFSK) Verfahren verwendet, sowie DQPSK und 8PSK für Enhanced Data Rate Pakete. Die benötigte Bandbreite für eine Bluetooth Übertragung ist verglichen mit Wireless LAN, das für einen Kanal mindestens 22 MHz belegt, sehr gering.

Um eine bidirektionale Übertragung zu ermöglichen, wird ein Übertragungskanal in Zeitschlitze (Slots) mit einer Länge von je 625 µs unterteilt. Alle Endgeräte, die untereinander Daten austauschen, verwenden diesen Kanal abwechselnd. Dies ist der Grund für die variablen Übertragungsgeschwindigkeiten in Abb. 7.1. Hat ein Endgerät mehr zu senden, kann es bis zu 5 aufeinander folgende Zeitschlitze belegen, bevor das Senderecht an ein anderes Endgerät übergeht. Hat dieses nur wenig zu senden, belegt es den

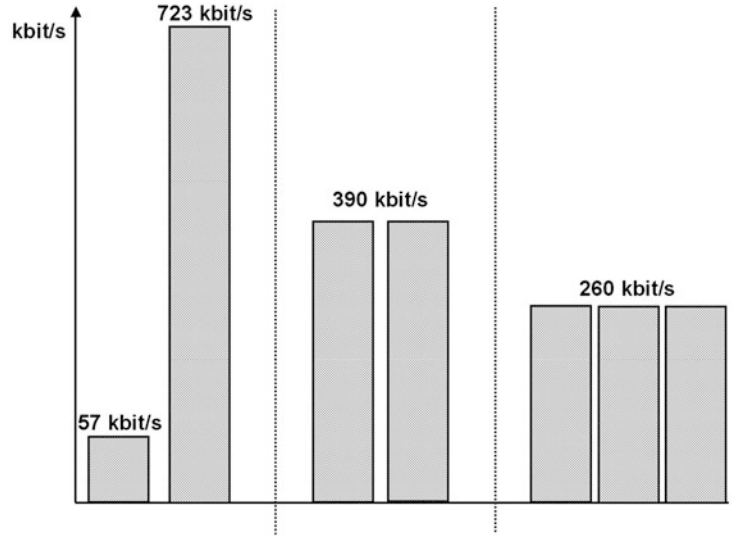

Abb. 7.1 Drei Beispiele für die maximale Geschwindigkeit in Abhängigkeit der Anzahl der Endgeräte und Teilnehmeraktivität

Übertragungskanal nur für einen Zeitschlitz. Auf diese Weise ist es möglich, die Datenrate in beiden Richtungen dynamisch dem Datenaufkommen anzupassen.

Da sich Bluetooth das 2,4 GHz ISM Frequenzband mit anderen Funktechnologien wie z. B. Wireless LAN teilt, sendet Bluetooth nicht auf einer festen Frequenz, sondern wechselt nach jedem Paket die Frequenz. Ein Paket kann dabei eine Länge von einem, drei oder fünf Slots haben. Dieses Verfahren wird Frequency Hopping Spread Spectrum (FHSS) genannt. In den meisten Fällen können somit gegenseitige Störungen vermieden werden. Sollte die Übertragung in einem Zeitschlitz trotz allem einmal gestört sein, werden die Daten automatisch erneut übertragen. Bei Paketen mit einer Länge von einem Slot (625 µs) ist somit die Hopping-Frequenz 1600 Hz, werden 5 Slot Pakete verwendet, beträgt die Hopping-Frequenz 320 Hz.

Damit mehrere Bluetooth Verbindungen, die auch Piconetze genannt werden, an einem Ort gleichzeitig betrieben werden können, verwendet jedes Piconetz eine eigene Hopping-Sequenz. Für das Frequency Hopping stehen Bluetooth im ISM-Band. 79 Kanäle zur Verfügung. Diese Anzahl genügt, um an einem Ort Wireless LAN Netzwerke und viele Bluetooth Netzwerke gleichzeitig und ohne wesentliche gegenseitige Beeinflussung zu betreiben.

Die gegenseitige Beeinflussung von WLAN und Bluetooth in Form einer überlagerten Übertragung auf der gleichen Frequenz bleibt gering, solange Wireless LAN und Bluetooth nur wenig ausgelastet sind. Wie in Kap. 4 gezeigt wurde, werden in einem Wireless LAN bei geringer Aktivität außer kurzen Beacon Frames fast keine Datenpakete gesendet. Ist jedoch ein Wireless LAN stark ausgelastet, wird auch ständig gesendet und eine Bandbreite von 25 MHz, also fast ein Drittel der Bluetooth Kanäle, dauerhaft belegt. In einem solchen Fall ist die Anzahl der gegenseitig zerstörten Pakete recht hoch. Aus diesem Grund wurde mit Bluetooth 1.2 das Adaptive Frequency Hopping (AFH) eingeführt. Sind alle Geräte die in

einem Piconetz kommunizieren zu Bluetooth 1.2 kompatibel, führt das Piconetz Master-Gerät (vgl. Abschn. 7.3) für alle Kanäle eine Kanalabschätzung (Channel Assessment) durch. Der Link Manager (vgl. Abschn. 7.4.3) legt dazu eine Liste aller Kanäle an (Channel Bitmap), die für das Frequency Hopping nicht verwendet werden sollen. Diese wird dann an alle Endgeräte des Piconetzes weitergegeben. Wie die Kanalabschätzung gemacht werden soll, wird vom Standard nicht vorgeschrieben. Mögliche Verfahren sind z. B. das Received Signal Strength Indication (RSSI) Verfahren oder der Ausschluss eines Kanals aufgrund einer hohen Packet Error Rate (PER). Bei Endgeräten, die Bluetooth und WLAN Funk eingebaut haben, bietet der Bluetooth 1.2 Standard auch die Möglichkeit, dass das Endgerät dem Bluetooth Stack Informationen übergibt, welche Kanäle gemieden werden sollen. Dies ist möglich, da das Endgerät weiß, welcher WLAN Kanal aktuell konfiguriert ist und welche Frequenzen somit vom eingebauten Bluetooth Modul vermieden werden sollten.

Da Bluetooth speziell für kleine, mobile und batteriebetriebene Geräte konzipiert wurde, sind im Standard drei verschiedene Sendeleistungen spezifiziert. Endgeräte wie z. B. Mobiltelefone gehören meist zur Leistungsklasse (Power Class) 3 und senden mit einer Leistung von bis zu einem Milliwatt. Endgeräte der Klasse 2 senden mit bis zu 2,5 Milliwatt und Endgeräte der Klasse 1 mit bis zu 100 Milliwatt. Nur Endgeräte wie z. B. manche USB Sticks für Notebooks und PCs haben einen Sender der Leistungsklasse 1. Deren Energieverbrauch ist jedoch im Vergleich zu Leistungsklasse 3 sehr hoch und sollte deshalb nur von Geräten verwendet werden, bei denen der Energieverbrauch keine entscheidende Rolle spielt. Die Reichweiten der einzelnen Leistungsklassen sind natürlich auch dementsprechend unterschiedlich. Während Klasse 3 Endgeräte eine maximale Distanz von 10 m überbrücken und maximal durch eine Wand senden können, schaffen Klasse 1 Endgeräte bis zu 100 m und können auch mehrere Wände durchdringen. Alle Endgeräte, gleich welcher Leistungsklasse, können miteinander kommunizieren. Da jede Kommunikationsverbindung bidirektional ist, bestimmt jedoch das Endgerät mit der geringeren Leistungsklasse die maximal mögliche Reichweite.

Sicherheitsmechanismen spielen bei Bluetooth eine wichtige Rolle. So wurden in den Standard starke Mechanismen für die Authentifizierung aufgenommen. Diese stellen sicher, dass nur vom Benutzer zugelassene Geräte untereinander kommunizieren können. Auch die Verschlüsselung ist Pflichtbestandteil des Standards und muss in jedem Endgerät integriert sein. Die Verschlüsselungssequenzen sind bei Bluetooth bis zu 128 Bit lang und bilden einen wirksamen Schutz gegen fremdes Abhören.

7.3 Piconetze und das Master Slave Konzept

Bei Bluetooth werden alle Geräte die momentan miteinander kommunizieren in einem so genannten Piconetz zusammengefasst. Die in Abb. 7.2 beschriebene Frequency Hopping Sequenz des Piconetzes wird durch die Hardwareadresse des Endgerätes berechnet, das als erstes Kontakt zu einem anderen Endgerät aufnimmt und somit das Piconetz aufbaut. Auf diese Weise ist es möglich, viele Piconetze am gleichen Ort ohne gegenseitige Beeinflussung zu betrieben.

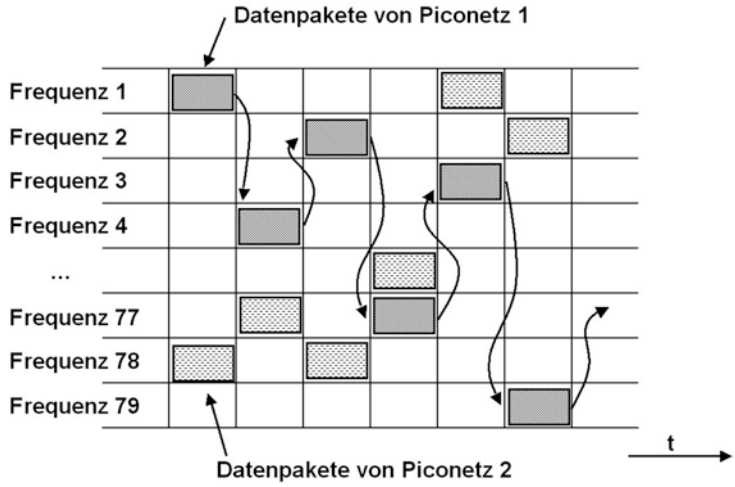

Abb. 7.2 Durch unterschiedliche Hop-Sequenzen können viele Piconetze am gleichen Ort betrieben werden

Ein Piconetz kann ein Master- und bis zu sieben Slave Endgeräte umfassen. Dies scheint auf den ersten Blick sehr wenig zu sein. Da die meisten Bluetooth Anwendungen, wie in Abschn. 7.1 gezeigt, nur Punkt zu Punkt Verbindungen sind, ist diese Zahl aber vollkommen ausreichend. Jedes Endgerät kann Master oder Slave eines Piconetzes sein. Per Definition ist immer jenes Endgerät der Master eines Piconetzes, welches dies ursprünglich aufgebaut hat. Folgendes Beispiel verdeutlicht dieses Konzept:

Ein Anwender hat ein Bluetooth fähiges Mobiltelefon und ein Headset. Nachdem diese zwei Geräte anfangs einmal miteinander gekoppelt wurden (Pairing, siehe Abschn. 7.5.1), können diese Geräte jederzeit miteinander Verbindung aufnehmen und somit für die Dauer eines Telefonats ein Piconetz bilden. Nach dem Ende des Telefonats wird die Bluetooth Verbindung zwischen Mobiltelefon und Headset wieder beendet und das Piconetz dadurch wieder abgebaut. Bei einem ankommenden Telefongespräch nimmt das Mobiltelefon mit dem Headset Kontakt auf und ist somit der Master der Verbindung. Möchte im umgekehrten Fall der Anwender ein abgehendes Telefonat führen, betätigt er eine Taste am Headset. Das Headset nimmt daraufhin Verbindung mit dem Mobiltelefon auf. In diesem Fall ist nicht das Mobiltelefon, sondern das Headset der Master des neu aufgebauten Piconetzes. Befindet sich ein anderer Anwender in unmittelbarer Nähe, der auch gerade per Bluetooth Headset telefoniert, führt dies nicht zu Problemen, da die Frequency Hopping Sequenzen der beiden Piconetze unterschiedlich sind. Durch die ursprüngliche Kopplung von Headset und Mobiltelefon ist auch sichergestellt, dass jedes Headset sein eigenes Mobiltelefon findet und auch nur mit diesem kommunizieren darf.

Der Master eines Piconetzes hat die Kontrolle, wer zu welchem Zeitpunkt Daten auf dem Kanal übertragen darf. Um einem Slave Endgerät das Senderecht zu erteilen, schickt ihm der Master ein Datenpaket. Das Slave Endgerät wird über eine 3-Bit Adresse im Header des Datenpakets identifiziert, die ihm bei der ersten Kontaktaufnahme zugewiesen

wurde. Das Datenpaket des Masters kann je nach Datenaufkommen 1–5 Slots lang sein. Hat der Master keine Daten für den Slave, sendet er ein leeres Paket. Unabhängig, ob das Paket Nutzdaten enthält oder nicht, übergibt der Master dem Slave auf diese Weise implizit das Senderecht. Der Slave kann dann in den nächsten 1–5 Slots ein Antwortpaket zurückschicken. Bei Bluetooth 1.1 antwortet der Slave auf der nächsten Frequenz in der Frequency Hopping Abfolge. Bei Bluetooth 1.2 wurde dieses Konzept leicht geändert, der Slave antwortet hier auf der zuvor vom Master verwendeten Frequenz. Hat der Slave keine Daten für den Master, antwortet er trotzdem mit einem leeren Paket als Empfangsbestätigung für das zuvor vom Master eingegangene Paket. Nach spätestens 5 Slots geht das Senderecht wieder automatisch an den Master über, auch wenn der Slave noch weitere Daten in seinem Sendepuffer hat. Danach kann der Master entscheiden, ob er wieder diesem, oder einem anderen Slave das Senderecht erteilt. Empfing der Master in den letzten Datenpaketen keine Nutzdaten und ist auch sein Sendepuffer leer, kann er eine Sendepause von bis zu 800 Slots einlegen, um damit Strom zu sparen. Da ein Slot eine Dauer von 625 µs hat, entsprechen 800 Slots einer Sendepause von 0,5 s (Abb. 7.3).

Da ein Slave nicht vorhersehen kann, zu welchem Zeitpunkt Datenpakete des Masters eingehen, kann er keine Verbindungen zu weiteren Geräten aufnehmen. In manchen Fällen ist es deshalb notwendig, dass Master und Slave ihre Rollen tauschen können. Diese Funktion ist z. B. notwendig, wenn ein Smartphone mit einem PC Kontakt aufgenommen hat, um mit ihm Daten zu synchronisieren. Da das Smartphone die Verbindung zum PC aufgebaut hat, ist er der Master des Piconetzes. Während die Verbindung besteht, möchte der Nutzer des PCs jedoch ein Bild von einem weiteren Smartphone zu sich übertragen und muss deshalb zusätzlich eine Verbindung zu Smartphone 2 herstellen. Dies ist aber nur möglich, wenn Smartphone 1 (Master) und PC (Slave) die Rollen im Piconetz tauschen. Diese Prozedur wird auch Master-Slave Role Switch genannt. Nach dem Rollentausch ist der PC der Master des Piconetzes zwischen ihm und Smartphone 1. So ist es ihm möglich, zusätzlich Kontakt zu Smartphone 2 aufzunehmen, während die Datenübertragung mit Smartphone 1 noch läuft. Durch die Kontaktaufnahme mit Smartphone 2 und der Übertragung des Bildes verringert sich jedoch die Datenrate zwischen PC und Smartphone 1.

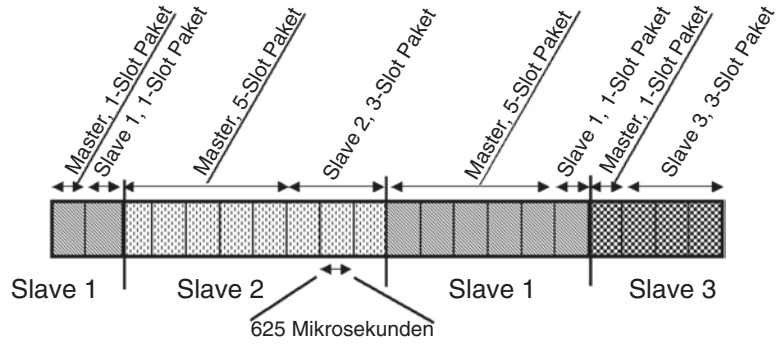

Abb. 7.3 Kommunikation zwischen einem Master und drei Slave Endgeräten

7.4 Der Bluetooth Protokoll Stack

Abb. 7.4 zeigt die unterschiedlichen Schichten des Bluetooth Protokoll Stacks und dient den nachfolgenden Unterkapiteln als Referenz. Die einzelnen Bluetooth Protokollschichten halten sich nur lose an das 7 Schichten OSI Modell, da manche Bluetooth Layer Aufgaben aus unterschiedlichen OSI Schichten übernehmen.

7.4.1 Der Baseband Layer

Die Eigenschaften der physikalischen Schicht, also der Radioübertragung, wurden im vorhergehenden Abschnitt schon beschrieben. Auf den Eigenschaften des physikalischen Kanals setzt dann der Baseband Layer auf, der typische Aufgaben eines Layer 2 Protokolls wie z. B. das Framing von Datenpaketen übernimmt. Für die Datenübertragung bietet der Baseband Layer drei unterschiedliche Frametypen:

Für die Paketdatenübertragung werden bei Bluetooth Asynchronous Connection-Less (ACL) Pakete verwendet. Wie in Abb. 7.5 gezeigt, besteht ein ACL Paket aus einem 68–72 Bit langen Access Code, einem 18 Bit Header und einem 0-2744 Bit langen Feld für die eigentlichen Nutzdaten (Payload).

Vor der Übertragung werden die 18 Header-Bits noch durch einen Forward Error Correction Algorithmus in 54 Bits kodiert (1/3 FEC). Dies stellt sicher, dass Übertragungsfehler in den meisten Fällen korrigiert werden können. Je nach Größe des Nutzdatenfeldes benötigt ein ACL Paket 1, 3 oder 5 Slots zu je 625 µs Dauer.

Der Access Code am Anfang des Pakets dient in erster Linie zur Identifikation des Piconetzes, zu dem das aktuelle Paket gehört. Erzeugt wird der Access Code deshalb aus der Geräteadresse des Piconet Masters. Der eigentliche Header des ACL Pakets besteht aus einer Reihe von Bits, die folgende Funktionen haben: Die ersten drei Bits des Headers ist die Logical Transfer Address (LT_ADDR) eines Slaves, die der Master bei der Verbindungsaufnahme zuweißt. Über die 3 Bit lassen sich insgesamt bis zu 7 Slaves adressieren.

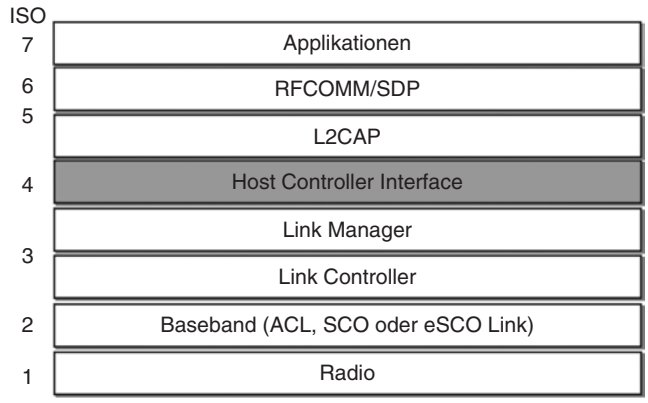

Abb. 7.4 Der Bluetooth Protokoll Stack

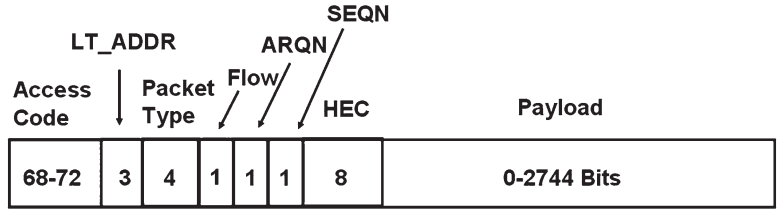

Abb. 7.5 ACL Paket

Daran anschließend folgt der Pakettyp mit 4 Bits, der den Aufbau des restlichen Pakets näher beschreibt. Nachfolgende Tabelle zeigt die unterschiedlichen Möglichkeiten für ACL Pakete. Neben der Anzahl der Slots eines Paketes, unterscheiden sich die Pakettypen auch in der Anwendung einer Forward Error Correction (FEC) für den Nutzdatenteil. Diese ermöglicht es auf der Empfängerseite, Übertragungsfehler zu korrigieren. Nachteil ist jedoch, dass die Anzahl der Nutzdatenbits pro Paket reduziert wird. Mit einer 2/3 FEC wird für zwei Nutzdatenbits ein Bit für die Fehlerkorrektur hinzugefügt. Statt zwei Bits werden dann drei Bits übertragen (2/3). Außerdem wird bei ACL Paketen grundsätzlich eine CRC Checksumme berechnet, um Fehler erkennen zu können.

Paket- typ	Anzahl Slots	Linktyp	Payload (Bytes)	FEC	CRC
0100	1	DH1	0–27	Nein	Ja
1010	3	DM3	0–121	2/3	Ja
1011	3	DH3	0–183	Nein	Ja
1110	5	DM5	0–224	2/3	Ja
1111	5	DH5	0–339	Nein	Ja

Um einen Empfangspufferüberlauf zu vermeiden, kann ein Gerät über das Flow Bit seiner Gegenstelle signalisieren, für den Moment keine weiteren Daten zu senden.

Über das ARQN Bit teilt ein Endgerät seiner Gegenstelle mit, ob das zuvor gesendete Paket korrekt empfangen wurde. Ist dieses Bit nicht gesetzt, sendet die Gegenstelle das zuvor übertragene Paket erneut.

Um auch den kompletten Verlust eines Pakets erkennen zu können, folgt als nächstes Feld im ACL Header das Sequence (SEQN) Bit. Dieses wird bei jeder Übertragung eines neuen Pakets auf den jeweils anderen Bitwert gesetzt. Werden zwei aufeinander folgende Pakete mit identischem SEQN Bit empfangen, bedeutet dies für Endgerät-2, dass sein letztes Paket Endgerät-1 nicht erreicht hat und Endgerät-1 daraufhin sein Paket wiederholt hat. Endgerät-2 wiederholt daraufhin sein Paket mit Empfangsbestätigung zu Endgerät-1 und ignoriert alle Pakete, bis wieder ein Paket mit korrektem SEQN Bit von Endgerät-1 empfangen wird. Auf diese Weise wird sichergestellt, dass auch bei mehrfachem Paketverlust die Empfangsbestätigung trotzdem zugestellt werden kann.

Als letztes Header-Feld folgt der Header Error Check (HEC). Dieses Feld stellt sicher, dass bei falsch empfangenem Header das Paket beim Empfänger ignoriert wird.

Auf den ACL Header folgt das Payload-Feld. Dieses enthält am Anfang den Payload Header, der folgende Aufgaben erfüllt: Das erste Feld wird L_CH (Logical Channel) genannt. Es gibt an, ob das Payload Feld Nutzdaten (L2CAP Pakete, vgl. Abschn. 7.4.6) oder Signalisierungsdaten in Form einer LMP Nachricht enthält (vgl. Abschn. 7.4.3) (Abb. 7.6).

Mit dem Flow Bit kann ein voller Empfangspuffer auf der L2CAP Nutzdatenschicht gemeldet werden. Schließlich enthält der Payload Header noch ein Längenfeld. Abgeschlossen wird ein ACL Paket immer durch eine 16 Bit Checksumme.

Da bei der Übertragung von ACL Paketen keine Bandbreite garantiert werden kann, eignen sich diese nicht für die Übertragung von Echtzeitdaten wie z. B. Sprache. Für diese Anwendung gibt es auf dem Baseband Layer zusätzlich den Synchronous Connection Oriented (SCO) Pakettyp. Im Unterschied zu ACL Paketen werden SCO Pakete zwischen Master und Slave in fest vorgegebenen Intervallen übertragen. Das Intervall wurde dabei so gewählt, dass die resultierende Bandbreite genau 64 kbit/s beträgt.

Bei SCO Verbindungen ist das Slave Endgerät autonom, es sendet sein SCO Datenpaket auch dann, wenn es zuvor kein Paket vom Master erhalten hat. Dies ist bei einer SCO Verbindung problemlos möglich, da Pakete zu vordefinierten Intervallen gesendet und empfangen werden. Der Slave ist somit also nicht auf eine Sendeerlaubnis des Masters angewiesen und es ist implizit sichergestellt, dass zu dieser Zeit nur er Daten überträgt. Auf diese Weise wird erreicht, dass trotz eines nicht erhaltenen Pakets in Empfangsrichtung das eigene Sprachpaket trotzdem übertragen wird.

Der Header eines SCO Paketes entspricht dem eines ACL Paketes, die Flow, ARQN und SEQN Felder werden bei SCO Paketen jedoch nicht verwendet. Die Länge des Nutzdatenfeldes beträgt immer genau 30 Bytes. Je nach verwendetem Fehlerkorrekturverfahren entspricht dies 10, 20 oder 30 Nutzdatenbytes. Nachfolgende Tabelle gibt einen Überblick über die möglichen SCO Pakettypen.

Paket-typ	Anzahl Slots	Linktyp	Payload (Bytes)	FEC	CRC
0101	1	HV1	10	1/3	Nein
0110	1	HV2	20	2/3	Nein
0111	1	HV3	30	keine	Nein
1000	1	DV	10 (+0–9)	2/3	Ja

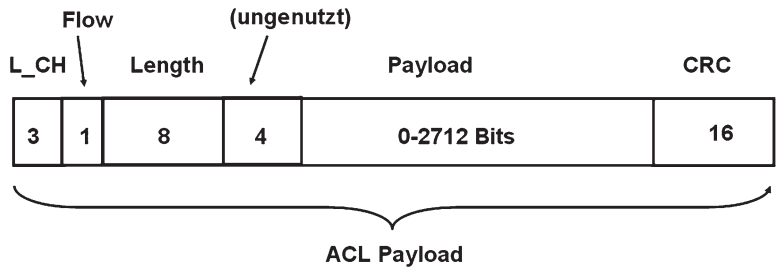

Abb. 7.6 Das ACL Payload-Feld mit Header

Die letzte Zeile der Tabelle zeigt einen Spezialpakettyp, der gleichzeitig SCO und ACL Daten enthält. Dieser Pakettyp wird verwendet, wenn neben den reinen Sprachdaten auch Steuerdaten zu übertragen sind. Wie später in Abschn. 7.6.4 im Zusammenhang mit dem Headset Profil gezeigt wird, werden zwischen einem Headset und einem Mobiltelefon nicht nur Sprachdaten, sondern auch in manchen Fällen Signalisierungsdaten (z. B. Lautstärkeregelung) übertragen. Die SCO Sprachdaten werden in einem solchen „DV“ Datenpaket dann in den ersten 10 Bytes übertragen, auf die 0–9 Bytes für den ACL Kanal folgen. Die in der Tabelle eingetragene Forward Error Correction und Checksumme wird nur für den ACL Teil verwendet. Der Standard schreibt die Verwendung eines DV Pakets nicht zwingend vor, falls Sprache und Daten gleichzeitig zwischen zwei Geräten zu übertragen sind. Eine weitere Möglichkeit ist, eigenständige ACL Pakete in den von der SCO Verbindung nicht verwendeten Slots zu senden. Dritte Möglichkeit ist, die Sprachdaten eines Slots zu verwerfen und statt des SCO Pakets ein ACL Paket zu schicken.

Da bei SCO Paketen nicht festgestellt werden kann, ob die Nutzdaten des Pakets korrekt übertragen wurden, werden bei schlechten Übertragungsbedingungen fehlerhafte Pakete an höhere Protokollschichten weitergegeben. Diese erzeugen bei der Wiedergabe der Sprache hörbare Knackgeräusche. Außerdem limitiert die maximale Geschwindigkeit eines SCO Kanals von 64 kbit/s die Anwendungsmöglichkeiten eines SCO Kanals, da z. B. Musikdaten beim Audiostreaming meist höhere Datenraten benötigen. Um diese Nachteile zu beseitigen, wurde mit Bluetooth Version 1.2 der Enhanced-SCO (eSCO) Pakettyp eingeführt. Dieser bietet folgende Vorteile:

Die Datenrate eines eSCO Kanals kann beim Aufbau der Verbindung festgelegt werden. Auf diese Weise sind konstante Datenraten bis zu 288 kbit/s in beide Richtungen möglich.

eSCO Pakete besitzen für den Nutzdatenteil eine Checksumme. Beim Auftreten eines Übertragungsfehlers kann das Paket erneut übertragen werden, falls noch genügend Zeit vor der Übertragung des nächsten regulären Pakets bleibt. Abb. 7.7 zeigt diese Situation. Bluetooth macht sich für dieses Verfahren den Umstand zunutze, dass z. B. bei einer 64 kbit/s eSCO Verbindung nur ein Bruchteil der gesamten Bandbreite des Kanals genutzt

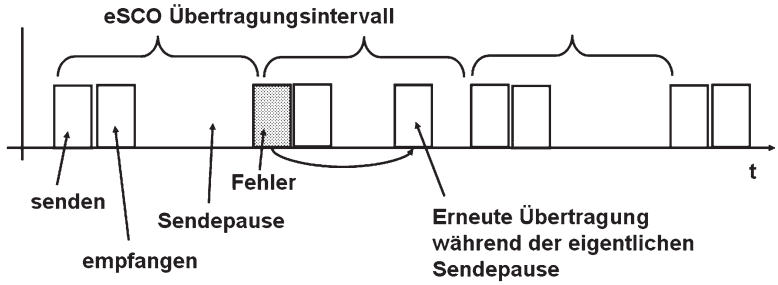

Abb. 7.7 Erneute Übertragung eines eSCO Pakets nach einem Übertragungsfehler

wird und somit genug Zeit für eine erneute Übertragung bleibt. Trotz der mehrfachen Übertragung eines Pakets bleibt dadurch die Datenrate konstant. Um der Gegenseite ein verlorenes oder fehlerhaftes Paket zu signalisieren, wird das von ACL Paketen bekannte Acknowledge Verfahren verwendet. Ist es bis zur Übertragung des nächsten regulären Paketes nicht möglich ein Paket korrekt auszuliefern, wird es verworfen. Somit ist gewährleistet, dass der Datenstrom nicht ins Stocken gerät. Ab Bluetooth V2.1 kann auch ein nicht korrekt empfangenes Paket an höhere Schichten zusammen mit einer Fehlerindikation weitergegeben werden (Erroneous Data Reporting). Dies macht Sinn, wenn ein Codec kleine Übertragungsfehler selber ausgleichen kann.

Um die Übertragungsgeschwindigkeit von Bluetooth zu erhöhen, erschien 2004 die Bluetooth Version 2.0+ Enhanced Data Rate (EDR). Kern von EDR ist die Verwendung von neuen Modulationsverfahren für den Nutzdatenteil eines ACL oder eSCO Paketes. Während Header und Nutzdatenteil der zuvor beschriebenen Pakete per GFSK moduliert werden, wird der Nutzdatenteil von EDR ACL oder eSCO Paketen per DQPSK oder 8DPSK moduliert. Diese Verfahren erlauben pro Übertragungsschritt die Übertragung von mehr als einem Bit. Auf diese Weise kann unter Beibehaltung der Kanalbandbreite von 1 MHz und der Slotzeit von 625 µs die Übertragungsgeschwindigkeit gesteigert werden. Um rückwärtskompatibel zu sein, wird der Header jedes Paketes weiterhin über GFSK moduliert. Somit kann der Header auch von einem Bluetooth Endgerät ohne EDR Funktionalität korrekt empfangen werden. Auch bei Wireless LAN wird dieses Verfahren verwendet, um die Kompatibilität zwischen 802.11b und den schnelleren 802.11 g und 11n Varianten zu gewährleisten. Die Beibehaltung der bisherigen Headermodulation sorgt außerdem dafür, dass auch nicht-EDR Geräte bei der Übertragung von Multislotpaketen zwischen dem Master und einem anderen Gerät weiterhin ihren Empfänger abschalten und somit Strom sparen können.

Die nachfolgende Tabelle gibt einen Überblick über alle möglichen ACL Pakettypen und die maximale Datenrate im asymmetrischen Betrieb. Asymmetrisch bedeutet, dass 5 Slot Pakete in Vorwärtsrichtung verwendet werden und 1 Slot Pakete in der Gegenrichtung. Im ersten Teil der Tabelle sind alle ACL Pakettypen aufgelistet, die von allen Bluetooth Endgeräten beherrscht werden. Im zweiten und dritten Teil der Tabelle sind dann die EDR ACL Pakettypen aufgelistet. 2-DH1, 3 und 5 werden mit DQPSK moduliert, 3-DH1, 3, 5 mit 8DPSK. Die Zahl 1, 3 oder 5 am Ende des Namens gibt die Anzahl der Slots an, die das Paket belegt.

Typ	Payload (Bytes)	Datenrate uplink (kbit/s)	Datenrate downlink (kbit/s)
DM1	0–17	108,8	108,8
DH1	0–27	172,8	172,8
DM3	0–121	387,2	54,4
DH3	0–183	585,6	86,4
DM5	0–224	477,8	36,3
DH5	0–339	723,2	57,6
2-DH1	0–54	345,6	345,6
2-DH3	0–367	1174,4	172,8

Typ	Payload (Bytes)	Datenrate uplink (kbit/s)	Datenrate downlink (kbit/s)
2-DH5	0–679	1448,5	115,2
3-DH1	0–83	531,2	531,2
3-DH3	0–552	1766,4	265,6
3-DH5	0–1021	2178,1	177,1

Durch die neuen Pakettypen ist es nicht mehr möglich, alle Pakettypen eindeutig über das 4 Bit lange Paket Type Feld zu identifizieren (vgl. Abb. 7.5). Die Bluetooth Spezifikation behilft sich deswegen mit folgendem Umweg: Im Grundzustand ist EDR deaktiviert. Erkennen zwei Bluetooth Endgeräte beim Einrichten einer Verbindung, dass sie beide EDR beherrschen, können die Link Manager der beiden Geräte (vgl. Abschn. 7.4.3) diese Funktionalität aktivieren und die Bitkombinationen des Paket Type Felds werden den 2-DHx und 3-DHx Typen zugeordnet.

Während EDR die DQPSK Modulation als verbindlich vorschreibt, bleibt die 8DPSK Modulation für die 3-DHx Pakete optional. Ob ein Endgerät also eine maximale Datenrate von 1448,5 oder 2178,1 Mbit/s unterstützt kann nicht von seiner EDR Fähigkeit abgeleitet werden.

Neben ACL, SCO und eSCO Paketen für die eigentliche Datenübertragung gibt es noch eine Anzahl weiterer Pakettypen, die nur für den Aufbau oder den Erhalt einer Verbindung verwendet werden:

ID Pakete werden vor dem Verbindungsaufbau von einem Gerät gesendet, um andere Geräte ausfindig zu machen. Da das Timing und die Hopping Sequenz der Gegenstelle zu diesem Zeitpunkt nicht bekannt sind, enthält ein solches Paket nur den Access Code.

Ein Frequency Hop Synchronization (FHS) Paket wird während eines Verbindungsaufbaus zwischen zwei Endgeräten in der Inquiry und Paging Phase gesendet. Inquiry und Paging werden im nächsten Unterkapitel genauer vorgestellt. Es enthält neben der 48 Bit Device Adresse des sendenden Geräts auch Timing Informationen, um die weitere Verbindungsaufnahme zu erleichtern.

NULL Pakete dienen der Empfangsbestätigung eines zuvor eingegangenen Pakets, enthalten aber keine Nutzdaten. NULL Pakete müssen nicht bestätigt werden. Somit bieten sie die Möglichkeit, den gegenseitigen Bestätigungskreislauf zu unterbrechen, wenn keine Daten mehr im Sendepuffer anstehen.

Ein weiteres Spezialpaket ist das POLL Paket. Mit diesem kann überprüft werden, ob Slaves bei längerer Übertragungspause noch im Piconetz angesprochen werden können. Wie das NULL Paket enthält es keine Nutzdaten.

7.4.2 Der Link Controller

Auf dem Baseband Layer baut die Link Controller Schicht auf. Wie der Name schon andeutet, ist der Link Controller für den Aufbau, den Erhalt und den korrekten Abbau von

Verbindungen zuständig. Für die Verwaltung der Verbindungen wird auf dieser Schicht ein Zustandsmodell verwendet. Für ein Gerät, das eine Verbindung zu einem anderen Gerät aufbauen möchte, gibt es folgende Zustände:

Möchte ein Endgerät bisher noch unbekannte Geräte in seiner Umgebung finden, wird der Link Controller von den höheren Protokollschichten angewiesen, in den Inquiry Zustand zu wechseln. In diesem Zustand sendet das Gerät in jedem Slot auf zwei unterschiedlichen Frequenzen ein ID Paket aus.

Alle Endgeräte, die eine Verbindungsaufnahme von unbekannten Geräten zulassen, müssen von Zeit zu Zeit in den Inquiry Scan Zustand wechseln und dort auf abwechselnden Frequenzen nach ID Paketen Ausschau halten. Die Empfangsfrequenz wird hier jedoch nur alle 1,28 s geändert. Um Strom zu sparen, oder die Verbindung mit anderen Endgeräten aufrecht zu erhalten, sucht ein Endgerät aber nicht im gesamten Intervall nach ID Paketen. Der Bluetooth Standard schlägt eine Scanzeit von 11,25 ms pro 1,28 s Intervall vor. Durch die Kombination aus schnellem Frequenzwechsel des suchenden Endgerätes und langsamem Frequenzwechsel des Ausschau haltenden Endgeräts, ergibt sich eine 90 % Wahrscheinlichkeit, dass sich die Geräte innerhalb von 10 s finden.

Um die Geschwindigkeit der Suche zu beschleunigen, wurde mit Bluetooth 1.2 der so genannte Interlaced Inquiry Scan eingeführt. Mit dieser Methode wird statt auf einer Frequenz pro Periode auf zwei Frequenzen pro Periode nach ID Paketen gesucht. Außerdem ist es seit dieser Bluetooth Version möglich, eine Empfangsstärkemessung (RSSI, Received Signal Strength Indication) für gefundene Geräte an höhere Schichten weiterzugeben. Somit ist es möglich, die Liste der gefundenen Geräte nach der Empfangsstärke zu sortieren. Dies ist vor allem dann sinnvoll, wenn z. B. während einer Messe sehr viele Bluetooth Endgeräte in der Nähe sind und ein Nutzer seine elektronische Visitenkarte an ein Endgerät senden möchte, das sich in unmittelbarer Nähe befindet. Da dieses Gerät besser als weiter entfernte Geräte empfangen wird, erscheint es auf diese Weise ganz oben in der Liste.

Empfängt ein Endgerät ein ID Paket, sendet es ein Frequency Hop Synchronization (FHS) Paket zurück, das neben seiner Device-Adresse auch Frequency Hopping und Synchronisationsinformationen enthält.

Das suchende Endgerät hat nach Empfang des FHS Paketes die Möglichkeit, die Inquiry Prozedur fortzusetzen, um weitere Endgeräte zu finden. Alternativ kann die Inquriy Prozedur auch beendet werden, um sofort über die nachfolgend beschriebene Paging Prozedur eine ACL Verbindung zu dem neu gefundenen Endgerät herzustellen.

Auch Master Endgeräte, die sich schon in einer aktiven Verbindung befinden, können von Zeit zu Zeit in den Inquiry Scan Zustand wechseln. Somit sind sie auch während einer bestehenden Verbindung weiterhin für unbekannte Endgeräte sichtbar. Manche Endgeräte wie z. B. Mobiltelefone unterstützen diese optionale Funktionalität jedoch nicht.

Möchte ein Anwender gar keinen Kontakt von unbekannten Geräten zulassen, kann die Inquiry Scan Funktion abgeschaltet werden. Somit können nur noch Geräte mit der nachfolgend beschriebenen Paging Prozedur Kontakt aufnehmen, denen die Device Adresse des Endgeräts bekannt ist. Diese Einstellung ist sinnvoll, nachdem der Anwender seine

Bluetooth Geräte untereinander bekannt gemacht hat (Pairing, siehe Abschn. 7.5.1) und fortan nur noch mit diesen kommunizieren will.

Um eine ACL Verbindung aufzubauen, müssen Endgeräte, denen die Device Adresse eines anderen Endgerätes schon bekannt ist, oder diese zuvor mit einer Inquiry Prozedur gefunden haben, eine Paging Prozedur durchführen. Das Paging funktioniert ähnlich dem Inquiry, ID Pakete werden in schneller Reihenfolge auf unterschiedlichen Frequenzen gesendet. Statt einer allgemeinen Adresse enthält das Paket jedoch die Geräteidentifikation der Gegenstelle, die zuvor über das FHS Paket ermittelt wurde, oder noch von der letzten Verbindung bekannt ist. Die Gegenstelle antwortet darauf ebenfalls mit einem ID Paket und gibt somit dem anfragenden Gerät die Möglichkeit, ein FHS Paket zurückzusenden, das seine Hopping Sequenz etc. enthält. Abb. 7.8 zeigt den Ablauf der Paging Prozedur und Übergang in den Connected Zustand.

Führt ein Endgerät Inquiry und Page Scans durch, und bestehen keine aktiven Verbindungen zu anderen Geräten, ist der Stromverbrauch eines Bluetooth Chips sehr niedrig. Typisch ist dann ein Energieverbrauch von weit unter einem Milliwatt. Bei Akkukapazitäten von Mobiltelefonen im Bereich von 4000–5000 Milliwattstunden ist somit gewährleistet, dass die Bluetooth Funktionalität nur einen geringen Einfluss auf die Standby-Zeit des Geräts hat.

Nach erfolgreichem Paging befinden sich beide Endgeräte im Connection Active Zustand und der Datenaustausch über die neue ACL Verbindung kann beginnen.

Bei der Verbindungsaufnahme kann es vorkommen, dass der Slave der neuen Verbindung auch gleichzeitig Master einer anderen Verbindung ist, die schon vorher bestanden

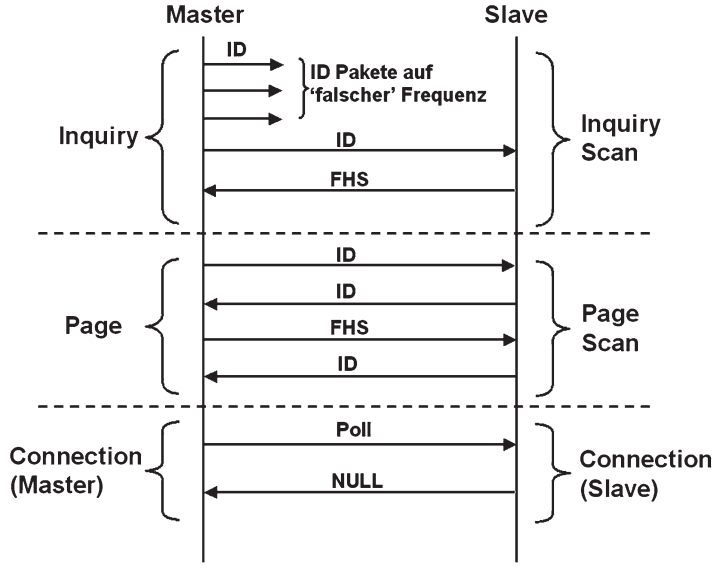

Abb. 7.8 Verbindungsaufbau zwischen zwei Bluetooth Geräten

hat. In solchen Fällen wird von den oberen Bluetooth Protokollschichten schon beim eingehenden Paging die Verbindung nur mit der Bedingung zugelassen, sofort nach der Verbindungsaufnahme automatisch einen Master-Slave Rollentausch durchzuführen. Nur so ist es möglich, dass das Endgerät gleichzeitig mit zwei anderen Endgeräten Daten austauschen kann.

Der Stromverbrauch während einer aktiven Verbindung hängt im Wesentlichen von der Leistungsklasse des Endgeräts ab (vgl. Abschn. 7.2). Während einer aktiven Verbindung kann es jedoch auch vorkommen, dass für einige Zeit keine Daten zu übertragen sind. Gerade für Endgeräte wie Smartphones ist es in dieser Zeit sehr wichtig, möglichst wenig Strom zu verbrauchten und somit die Laufzeit des Gerätes zu erhöhen. Für solche Fälle definiert der Bluetooth Standard für den Connected Zustand drei Unterzustände:

Der erste Unterzustand ist der Connection-Hold Zustand. Um in diesen Zustand zu wechseln, einigen sich Master und Slave über die Dauer des Hold Zustandes. Danach können Sender und Empfänger für diese Zeitdauer komplett abgeschaltet werden. Nach Ende der Hold Periode wechseln Master und Slave wieder automatisch in den Connection-Active Zustand.

Wesentlich flexibler ist der Connection-Sniff Zustand. Dieser Stromsparmodus ist ideal für Verbindungen mit wenig oder zeitweise keiner Aktivität geeignet. Master und Slave einigen sich beim Aktivieren des Sniff-Modus darauf, in welchen Intervallen und für wie lange pro Intervall ein Slave den Übertragungskanal abhören soll. In der Praxis ist zu beobachten, dass der Connection-Sniff Mode für folgende Anwendungen genutzt wird:

- Bei allen Profilen bei längerer Inaktivität (z. B. 15 s): Üblich sind dann Sniff Intervalle von z. B. 2 s. Bei erneuter Aktivität wird der Sniff-Modus wieder abgeschaltet, um eine möglichst hohe Übertragungsgeschwindigkeit zu erreichen.
- Bei Human Interface Device (HID) Profilen für Tastaturen und Mäuse: Da hier die benötigte Bandbreite gering ist, können sich Verbindungen für diese Profile ständig im Sniff-Modus befinden.

Im Sniff-Modus reduziert sich der Stromverbrauch des kompletten Bluetooth Chips auf weit unter 1 mW.

Ab Bluetooth Version 2.1 gibt es zusätzlich den Sniff-Subrating Mode, um den Energieverbrauch vor allem für HID Geräte weiter zu verringern. Endgeräte im Sniff-Mode können mit diesem Mechanismus eine weitere Reduzierung des Sniff Intervalls nach einem gewünschten Timeout aushandeln. Nach Ablauf des Timers fällt die Verbindung automatisch in den Sniff-Subrating Modus. Wird dann ein Paket empfangen, fällt die Verbindung in den normalen Sniff-Modus zurück und der Timer startet von neuem.

Um die Leistungsaufnahme noch weiter zu reduzieren, gibt es den Connection-Park Zustand. In diesem Zustand gibt der Slave seine Piconetadresse (LT_ADDR) auf und überprüft nur noch sehr selten, ob der Master die Verbindung reaktivieren möchte.

7.4.3 Der Link Manager

Die nächste Schicht des Protokoll Stacks (vgl. Abb. 7.4) ist die Link Manager Schicht. Während die zuvor besprochene Link Controller Schicht Datenpakete je nach Verbindungszustand sendet und empfängt, ist die Aufgabe des Link Managers die Einrichtung und Aufrechterhaltung von Verbindungen. Dies beinhaltet folgende Operationen:

- Aufbau einer ACL Verbindung zu einem Slave und Vergabe einer Linkadresse (LT_ADDR).
- Abbau von Verbindungen.
- Konfiguration einer Verbindung wie z. B. das Aushandeln der maximalen Anzahl von Slots von ACL oder eSCO Paketen.
- Einschalten der Enhanced Data Rate (EDR) Übertragung, falls beide Geräte diese Erweiterung unterstützen.
- Durchführung eines Master-Slave Rollentausches.
- Durchführen des in Abschn. 7.5.1 beschriebenen Pairings.
- Aktivierung und Kontrolle der Authentifizierung und Verschlüsselung, falls dies für die Verbindung von höheren Schichten gefordert wird.
- Kontrolle des mit Bluetooth 1.2 eingeführten Adaptive Frequency Hoppings (AFH).
- Management (Aktivierung/Deaktivierung) der Stromsparmodi Hold, Sniff und Park.
- Aufbau einer SCO oder eSCO Verbindung und Aushandeln der verwendeten Parameter wie z. B. die zu verwendenden Fehlerkorrekturmechanismen, Datenübertragungsraten (nur eSCO), etc.

Der Link Manager führt diese Operation entweder auf Befehl von höheren Schichten aus (vgl. nächstes Kapitel), oder aufgrund von Anfragen des Link Managers der Gegenstelle. Link Manager zweier Bluetooth Endgeräte kommunizieren, wie in Abb. 7.9 gezeigt, über ACL Verbindungen mit dem Link Manager Protocol (LMP). Ob es sich bei einem eingehenden ACL Paket um Nutzdaten oder um eine LMP Nachricht handelt, erkennt der Link Manager, wie in Abb. 7.6 gezeigt, über das Logical Channel (L_CH) Feld des ACL Nutzdatenheaders.

Damit eine Verbindung zu höheren Schichten nach erfolgreichem Aufbau einer ACL Verbindung hergestellt werden kann, muss zunächst der Link Manager des Geräts, das die ACL Verbindung veranlasst hat (Master), mit dem Link Manager der Gegenseite Kontakt aufnehmen. Dies geschieht mit einer LMP_Host_Connection_Request Nachricht. Danach können optionale Konfigurationsnachrichten ausgetauscht werden. Beendet wird die LMP Verbindungsphase durch gegenseitiges Senden einer LMP_Setup_Complete Nachricht. Nach diesem Schritt ist es dann möglich, Nutzdatenpakete transparent zwischen den zwei Endgeräten auszutauschen. Es können jedoch auch jederzeit innerhalb des Nutzdatenstroms weitere LMP Nachrichten eingeschoben werden, die für die am Anfang des Abschnitts beschriebenen Operationen notwendig sind.

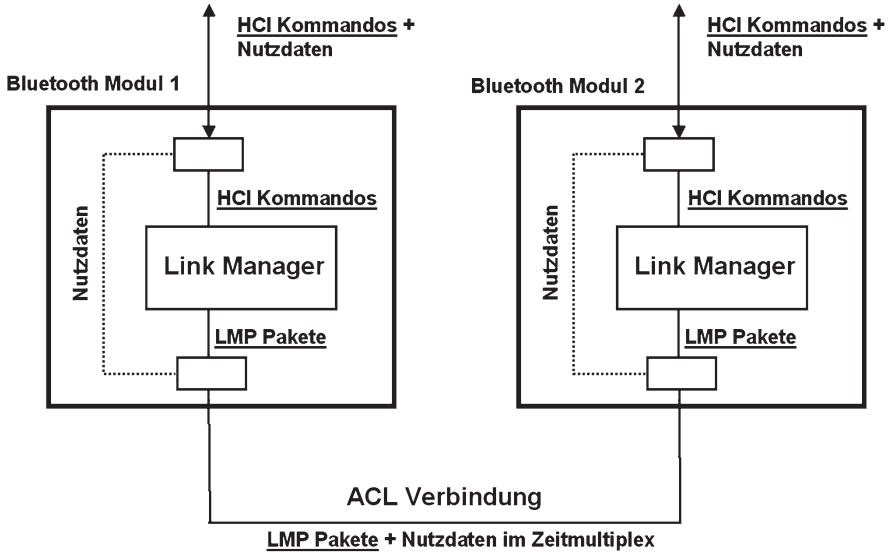

Abb. 7.9 Kommunikation zwischen zwei Link Managern per LMP

7.4.4 Das HCI Interface

Die nächste Ebene im Bluetooth Protokollstack ist das Host Controller Interface (HCI). Bei den meisten Bluetooth Implementierungen wird dieses Interface verwendet, um das Endgerät und den Bluetooth Chip physikalisch voneinander zu trennen. Ausnahmen sind z. B. Headsets, die aufgrund ihrer physikalischen Größe und der Limitation auf Sprachübertragung alle Bluetooth Protokollschichten in einem Chip integrieren.

Über die HCI Schnittstelle können zwischen Endgerät (Host) und Bluetooth Chip (Controller) Daten und Kommandos für den Link Manager in definierten Kommandos und Nachrichtenpaketen übertragen werden. Der Bluetooth Standard sieht für das HCI Interface zwei Schnittstellentypen vor:

Für Endgeräte wie z. B. Notebooks eignet sich die USB (Universal Serial Bus) Schnittstelle am besten. Der Bluetooth Standard definiert für dieses Hardware Interface, wie HCI Kommandos und Datenpakete über USB zu übertragen sind.

Für kompakte Endgeräte, wie z. B. Smartphones kann auch ein serielles Interface verwendet werden, das UART (Universal Asynchronous Receiver and Transmitter) genannt wird. Von den verwendeten Spannungspegeln abgesehen, ist dieses Interface mit der von PCs bekannten seriellen RS-232 Schnittstelle kompatibel. Während die RS-232 Schnittstelle jedoch auf eine Geschwindigkeit von 115 kbit/s beschränkt ist, können Daten über die UART Schnittstelle bei manchen Bluetooth Chips mit bis zu 1,5 Mbit/s übertragen werden. Dies ist auch notwendig, da die maximale Bluetooth Datenrate die Datenrate einer gewöhnlichen RS-232 Schnittstelle bei weitem übersteigt. Welche Geschwindigkeit

auf der UART Schnittstelle verwendet wird, bleibt den Entwicklern des Host Endgerätes überlassen.

Auf der HCI Schnittstelle können eine Reihe unterschiedlicher Pakettypen übertragen werden. Dies sind:

- Kommandopakete (Commands), die vom Host an den Link Manager im Bluetooth Chip übertragen werden.
- Antwortpakete auf Kommandos, die der Bluetooth Controller an den Host zurückschickt. Diese Pakete werden Events genannt. Events können auch ohne vorheriges Kommando an den Host geschickt werden, wenn z. B. ein anderes Bluetooth Gerät Kontakt aufnehmen möchte.
- Nutzdatenpakete von und zum Bluetooth Chip.

Auf der UART Schnittstelle werden die unterschiedlichen Pakettypen durch einen Header unterschieden. Das erste Byte eines Pakets gibt dabei an, um welchen Pakettyp es sich handelt. Wird USB als Übertragungsschnittstelle für das HCI Interface verwendet, werden die unterschiedlichen Pakettypen über unterschiedliche USB Endpunkte identifiziert. Eine USB Pollrate von einer Millisekunde sorgt dafür, dass Event Pakete und Nutzdatenpakete, die vom Bluetooth Chip an den Host zu übertragen sind, mit sehr kurzer Verzögerung erkannt und abgeholt werden.

Linux Distributionen für den PC unterstützen heute üblicherweise Bluetooth und bieten eine interessante Möglichkeit, über Shell Kommandos das HCI Interface zu tracen. Mit dem „hcitool con“ Kommando können beispielsweise alle Bluetooth Geräte angezeigt werden, die aktuell mit dem PC verbunden sind. Mit „hcitool info <Geräteadresse>“ können weitere Details über ein Gerät ausgelesen werden. Das „hciconfig“ Kommando mit diversen Parametern bietet die Möglichkeit, die Konfiguration und weitere Informationen über die unterstützten Bluetoothfähigkeiten des PCs auszugeben. Das wohl interessanteste Kommando ist jedoch „hcidump -X“, mit dem der komplette Datenaustausch auf dem HCI Interface zwischen Betriebssystem und Bluetooth Chip visualisiert werden kann. Mit „hcidump -w dumpfilename“ können auch alle Pakete abgespeichert werden und dann z. B. mit Wireshark im Detail analysiert werden.

Abb. 7.10 zeigt, wie ein Bluetooth Modul über das HCI Interface veranlasst wird, eine Verbindung zu einem anderen Bluetooth Endgerät aufzubauen. Über das HCI_Create_Connection Kommando werden dem Bluetooth Controller alle benötigten Informationen für den Verbindungsaufbau übergeben. Der wichtigste Parameter ist die Device-Adresse des anderen Bluetooth Gerätes. Nach Erhalt des Kommandos quittiert der Controller dieses mit einer HCI_Command_Status Event Nachricht und startet als nächstes die Suche nach dem anderen Endgerät. Der Ablauf dieser Suche ist in Abb. 7.8 zu sehen, wobei für diesen Fall jedoch die dort gezeigte Inquiry Phase entfällt, da die Bluetooth Device Adresse des anderen Gerätes schon bekannt ist. Konnte die Verbindung erfolgreich aufgebaut werden, sendet der Bluetooth Controller ein HCI_Connection_Complete Event zurück. Wichtigster Parameter ist ein Connection Handle, um Pakete von und zu unterschiedlichen Endgeräten

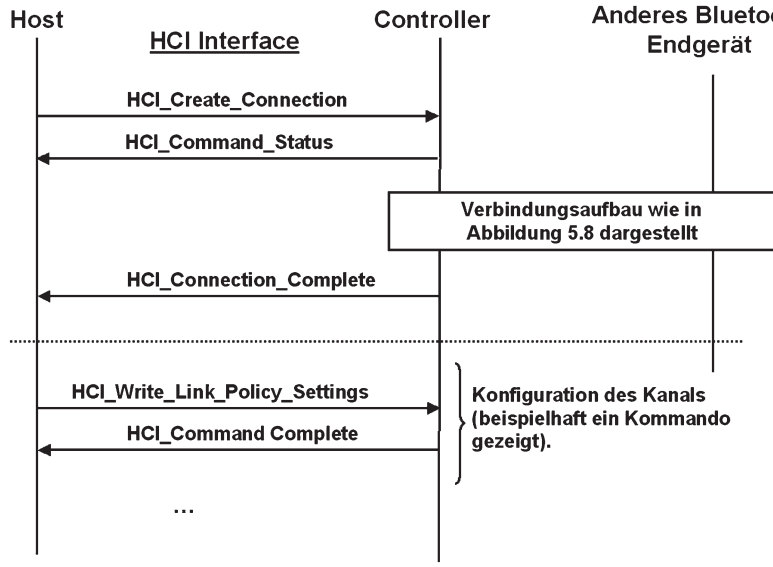

Abb. 7.10 Aufbau einer Verbindung per HCI Kommando

unterscheiden zu können. Das Connection Handle steht über diese Zuweißung in direkter Beziehung zum L_CH Parameter eines ACL bzw. SCO Paketes.

Für die Kontrolle einer Verbindung und die Konfiguration des Bluetooth Controllers gibt es eine Vielzahl weiterer HCI Kommandos und Events. Nachfolgende Tabelle zeigt eine kleine Auswahl der Kommandos:

Kommando	Aufgabe
Setup_Synchronous_ Connection	Für die Sprachübertragung (z. B. mit einem Headset) baut dieses Kommando einen SCO oder eSCO Sprachkanal auf.
Accept_Connection_Request	Bei einer ankommenden Bluetooth Verbindung signalisiert der lokale Link Manager dies den höheren Schichten über ein Connection_Request Event. Möchte der Host die Verbindung zulassen, antwortet er dem Link Manager im Controller Chip mit diesem Kommando.
Write_Link_Policy_Settings	Über dieses Kommando kann der Host die möglichen Verbindungszustände wie Hold, Park und Sniff erlauben oder sperren.
Read_Remote_Supported_ Features	Mit diesem Kommando kann ein Host den Bluetooth Controller anweisen, bei einer Gegenstelle eine Liste aller verfügbaren Bluetooth Funktionalitäten anzufordern. So kann der Host z. B. ermitteln, welche Multislot Pakettypen das andere Endgerät unterstützt, welche Stromsparmechanismen möglich sind, ob Adaptive Frequency Hopping verwendet werden kann, usw.
Disconnect	Beenden einer Verbindung.

Kommando	Aufgabe
Write_Scan_Enable	Mit diesem Kommando kann der Host kontrollieren, ob das Bluetooth Modul periodische Inquiry- und oder Page Scans durchführen soll. Wird beides abgeschaltet, können nur abgehende Verbindungen aufgebaut werden, das Gerät ist für andere Bluetooth Geräte unsichtbar.
Write_Inquiry_Scan_Activity	Übergibt dem Bluetooth Controller Werte für die Konfiguration des Inquiry Scans wie z. B. die Größe des Inquiry Scan Zeitfensters.
Write_Local_Name	Über dieses Kommando übergibt der Host einen „lesbaren" Gerätenamen an das Bluetooth Modul. Dieser kann dann automatisch anderen Geräten übergeben werden, die nach Bluetooth Geräten suchen. So ist es möglich, dem Benutzer eine Liste mit Gerätenamen statt Bluetooth Device Adressen anzuzeigen.

7.4.5 Der L2CAP Layer

Im nächsten Schritt der Verbindungsaufnahme wird über eine bestehende ACL Verbindung eine L2CAP (Logical Link Control and Adaptation Protocol) Verbindung aufgebaut. Diese Protokollschicht befindet sich über dem HCI Layer und kann mehrere logische Verbindungen zu einem Gerät über eine physikalische ACL Verbindung multiplexen. Somit kann z. B. während dem bestehen einer Bluetooth Dial-Up Verbindung zwischen einem PC und einem Mobiltelefon noch eine weitere zusätzliche logische Verbindung für die Übertragung eines Adressbucheintrages zwischen den Geräten aufgebaut werden. Bestehen zu einem Zeitpunkt noch weitere ACL Verbindungen zu anderen Geräten, kann die L2CAP Schicht auch Daten von und zu unterschiedlichen Geräte multiplexen. Ein solches Szenario ist in Abb. 7.11 dargestellt. Während einer Internet Dial-Up Verbindung über Slave 1 wird gleichzeitig noch eine Datei aus dem Speicher des Mobiltelefons zum Master übertragen, sowie ein MP-3 Datenstrom zwischen Master und Slave 2 übertragen.

Der Aufbau einer L2CAP Verbindung erfolgt über eine L2CAP_Connection_Request Nachricht. Wichtigster Parameter ist der Protocol Service Multiplexer (PSM). Dieser gibt an, an welche höhere Schicht Pakete nach erfolgreichem L2CAP Verbindungsaufbau weitergereicht werden sollen. Für die meisten Bluetooth Anwendungen wird der PSM 0×0003 verwendet, mit dem eine Verbindung zur RFCOMM Schicht hergestellt wird. Die RFCOMM Schicht stellt für Anwendungen eine virtuelle serielle Verbindung zu einem entfernten Bluetooth Endgerät her und wird in Abschn. 7.4.8 genauer beschrieben. Außerdem enthält die L2CAP_-Connection_Request Nachricht eine Connection ID (CID), über die fortan alle L2CAP Pakete der Verbindung identifiziert werden. Die CID ist notwendig, da die RFCOMM Schicht von mehreren Diensten gleichzeitig verwendet werden kann und somit der PSM nur beim Verbindungsaufbau eindeutig ist. Nimmt die Gegenstelle die L2CAP Verbindung an, sendet sie ein L2CAP_Connection_Response zurück und teilt ihrerseits eine Connection ID

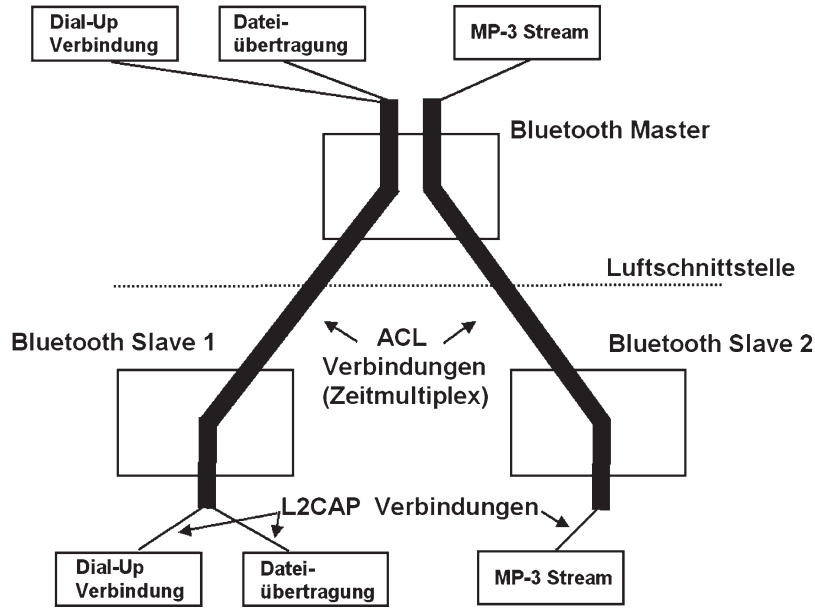

Abb. 7.11 Multiplexing verschiedener Datenströme

zu, über die L2CAP Pakete in der Gegenrichtung identifiziert werden. Danach ist die Verbindung eingerichtet und kann verwendet werden. Optional gibt es jetzt die Möglichkeit, weitere Parameter für die Verbindung über das L2CAP_Configuration_Request Kommando zu übertragen. Dazu zählen, z. B. die Anzahl der erneuten Sendeversuche bei Paketverlust und die maximale Paketlänge, die von einem Gerät unterstützt wird.

Eine weitere wichtige Aufgabe der L2CAP Schicht ist die Segmentierung von Datenpaketen aus höheren Schichten. Dies ist notwendig, wenn Pakete aus höheren Schichten größer als ein ACL Paket sind. Ein 5 Slot ACL Paket hat beispielsweise eine maximale Größe von 339 Bytes. Werden von der Anwendungsschicht größere Pakete angeliefert, werden diese in kleinere Stücke aufgeteilt und in mehreren ACL Paketen versandt. Im Header jedes ACL Paketes wird außerdem vermerkt, ob es den Anfang eines L2CAP Paketes darstellt, oder ein nachfolgendes Teilstück ist. Auf der Gegenseite kann dann die L2CAP Schicht mit dieser Information aus mehreren ACL Paketen wieder ein einziges Paket zusammensetzen, das an die Anwenderschicht weitergereicht wird.

7.4.6 Das Service Discovery Protocol

Theoretisch könnte nach dem Aufbau einer ACL und L2CAP Verbindung der Datentransfer zwischen zwei Endgeräten sofort aufgenommen werden. Bluetooth eignet sich jedoch für eine Vielzahl unterschiedlicher Dienste, und die meisten Endgeräte bieten mehrere Dienste gleichzeitig an. Ein Mobiltelefon beherrscht beispielsweise Dienste wie Internet Verbindung (Dial-Up Network), Dateitransfer, den Austausch von Adressen und Terminen

und vieles mehr. Damit ein Bluetooth Gerät in Erfahrung bringen kann, welche Dienste andere Bluetooth Endgeräte bieten und wie diese angesprochen werden können, muss vor dem Verbindungsaufbau zum eigentlichen Dienst eine Service Datenbank befragt werden. Die Service Datenbank wird über L2CAP PSM 0×0001 angesprochen und das Protokoll zur Kommunikation wird Service Discovery Protocol (SDP) genannt. Dieser Schritt kann entfallen, wenn das Endgerät genau weiß, wie der Dienst angesprochen werden kann. Bluetooth ist jedoch sehr flexibel und erlaubt Diensten, ihre Verbindungsparameter zur Laufzeit zu ändern. Einer dieser Verbindungsparameter ist z. B. die zu verwendende RFCOMM-Kanalnummer. Mehr hierzu in Abschn. 7.4.8.

Auf Anwenderebene werden Dienste auch Profile genannt. Der Headset Dienst/das Headset Profil stellt sicher, dass ein Headset mit allen gängigen Bluetooth Telefonen zusammenarbeitet, die ebenfalls das Headset Profil unterstützen. Mehr zu Bluetooth Profilen in Abschn. 7.5.

Jeder Bluetooth Dienst hat seine eigene universelle Identifikationsnummer (Universally Unique ID, UUID), über die er in der SDP Datenbank gefunden werden kann. Der Dial-Up Server Dienst hat z. B. die UUID 0×1103. Damit sich der Bluetooth Stack eines PCs mit diesem Dienst z. B. auf einem Mobiltelefon verbinden kann, wird nach der ersten Verbindungsaufnahme zuerst die SDP Datenbank des Mobiltelefons nach den nötigen Einstellungen für diesen Dienst befragt. Dies geschieht über eine SDP_Service_Search_Attribute_Req Nachricht. Wichtigster Parameter, den der Client der SDP Datenbank des anderen Gerätes übergibt, ist die UUID des Dienstes. Die Datenbank liefert dann in einer SDP_Service_Search_Attribute_Response Nachricht die benötigten Parameter in Form von Records zurück. Im Falle des Dial-Up Server Dienstes liefert die Datenbank die Information zurück, dass für diesen Dienst die L2CAP Schicht, sowie die im nächsten Unterkapitel vorgestellte RFCOMM Schicht zu verwenden sind (Abb. 7.12).

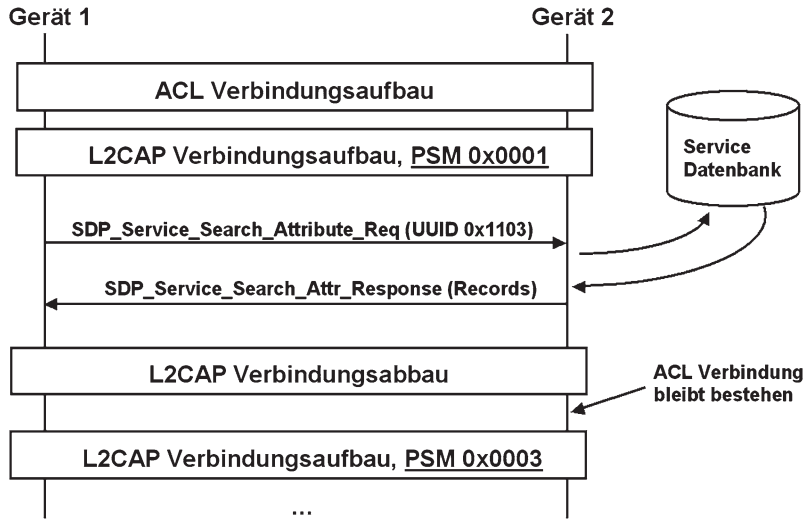

Abb. 7.12 Verbindungsaufbau zu einem Dienst mit vorheriger Datenbankabfrage

Die Service Datenbank eines Bluetooth Geräts bietet außerdem eine allgemeine Suchmöglichkeit. Diese wird von einem Endgerät verwendet, wenn es ein neues Bluetooth Gerät gefunden hat und der Benutzer wissen möchte, welche Dienste dieses Gerät anbietet. Die Nachricht für eine allgemeine Suche in der Datenbank lautet SDP_Service_Search_Request. Statt einer spezifischen UUID wie im Beispiel oben, wird die UUID der Public Browse Group (0×1002) übergeben. Die Datenbank liefert dann die UUIDs aller Dienste die es anbietet an das andere Endgerät. Die weiteren Parameter der einzelnen Dienste können nun mit SDP_Service_Search_Attribute_Request Anfragen an die Datenbank ausgelesen werden. Bei einer Anfrage liefert die Datenbank auch einen frei wählbaren Namen des angeforderten Dienstes im Klartext zurück. Auf diese Weise ist eine flexible länder- und sprachspezifische Anzeige eines Dienstnamens für den Anwender möglich. Der Name dient jedoch nur zur Benutzerinformation, der Bluetooth Stack selber identifiziert einen Dienst immer über die UUID und niemals über den Namen.

Oft werden die Informationen auch lokal auf der Anwenderschicht gespeichert, damit dem Anwender bei erneuter Nutzung eines Geräts die Liste der verfügbaren Dienste eines entfernten Geräts schneller angezeigt werden kann.

Um die Datenbankabfrage zu beenden, löst das abfragende Gerät die L2CAP Verbindung durch Senden einer L2CAP_Disconnection_Request Nachricht auf. Möchte das Gerät anschließend sofort eine Verbindung zu einem Dienst herstellen, bleibt die ACL Verbindung bestehen, und es wird sofort wieder ein L2CAP_Connection_Request Nachricht geschickt. Diese Nachricht enthält jedoch nicht die PSM ID 0×0001 für die Service Datenbank, sondern die PSM ID für die nächst höhere Schicht, die der gewünschte Dienst verwendet. Abgesehen von Sprachdiensten verwenden die meisten anderen Dienste den RFCOMM Layer, der eine virtuelle serielle Schnittstelle bietet. Dieser wird über den PSM 0×0003 angesprochen.

7.4.7 Der RFCOMM Layer

Wie in Abschn. 7.4.6 gezeigt, wird der L2CAP Layer verwendet, um mehrere Datenströme über eine physikalische Verbindung zu multiplexen. Die Service Datenbank ist z. B. eine Anwendung, die über den L2CAP Protocol Service Multiplexer (PSM) 0×0001 angesprochen wird. Andere Dienste könnten auf gleiche Weise über andere PSM angesprochen werden. In der Praxis verwenden jedoch einige Dienste noch einen weiteren gemeinsamen Layer, der RFCOMM genannt wird und über PSM 0×0003 angesprochen wird. RFCOMM stellt den Diensten virtuelle serielle Schnittstellen zur Verfügung und vereinfacht diesen dadurch die Datenübertragung.

Wie diese seriellen Schnittstellen verwendet werden, hängt von den übergeordneten Diensten ab. Mit dem „Serial Port“ Dienst beispielsweise wird über den RFCOMM Layer eine virtuelle serielle Schnittstelle für beliebige „nicht“ Bluetooth Anwendungen bereitgestellt. Diese unterscheidet sich aus Sicht einer Anwendung nicht von anderen seriellen Schnittstellen. Meist bekommen virtuelle serielle Bluetooth Schnittstellen vom Betriebs-

system die COM-Port Nummern 3, 4, 5, 6, 7 usw. zugeteilt. Welche genau, entscheidet sich bei der Installation des Bluetooth Protokoll Stacks auf einem PC. Diese seriellen Schnittstellen wurden z. B. vor dem Aufkommen der Smartphone Wi-Fi Hotspot Funktionalität bei der Einrichtung eines neuen Modemtreibers für das DFÜ-Netzwerk verwendet. Sobald das DFÜ-Netzwerk für den Aufbau einer Internet Verbindung diesen COM-Port öffnet, wird automatisch eine Bluetooth Verbindung zur Gegenseite hergestellt. Damit diese automatische Verbindungsaufnahme funktioniert, muss zuvor über die Bluetoothsoftware diese COM-Port Nummer einmalig mit der gewünschten Gegenstelle verbunden werden.

Um Anwendungen eine komplette serielle Schnittstelle zu bieten, simuliert der RFCOMM Layer nicht nur die Sende- und Empfangsleitungen, sondern auch die Statusleitungen Request to Send (RTS), Clear to Send (CTS), Data Terminal Ready (DTR), Data Set Ready (DSR), Data Carrier Detect (CD), sowie die Ring Indicator (RI) Leitung. Bei einer physikalisch vorhandenen seriellen Schnittstelle werden diese Leitungen über einen UART (Universal Asynchronous Receiver and Transmitter) Baustein angesprochen. Aus diesem Grund simuliert die Bluetoothsoftware für den „Serial Port“ Dienst einen kompletten UART Baustein. Während ein UART Baustein die Befehle der Anwendungsschicht auf physikalische Leitungen umsetzt, sendet der virtuelle Bluetooth UART Baustein die erhaltenen Steuerkommandos und Daten in RFCOMM Paketen verpackt an den L2CAP Layer weiter.

Auch andere Dienste, wie z. B. der auch heute noch gebräuchliche Dateitransferdienst (OBEX), setzen die RFCOMM Schicht ein. Über unterschiedliche RFCOMM Kanalnummern ist es möglich, beim Verbindungsaufbau auszuwählen, welcher Dienst angesprochen werden soll. Die Kanalnummer ist Teil der Dienstbeschreibung in der Servicedatenbank. Fragt also ein anderes Gerät die Servicedatenbank eines Bluetooth Geräts nach dem OBEX Dienst, so erfährt es über die Antwort, dass dieser Dienst über die L2CAP Schicht zu erreichen ist und als nächst höhere Schicht RFCOMM benutzt. Hieraus kann das Endgerät zunächst schließen, dass der L2CAP PSM 0×0003 zu verwenden ist, um die Verbindung zum RFCOMM Layer herzustellen (L2CAP nach RFCOMM). Außerdem entnimmt das Endgerät der OBEX Dienstbeschreibung, mit welcher RFCOMM-Kanalnummer dieser angesprochen werden kann (RFCOMM zu Anwendung). Da die RFCOMM-Kanalnummer dynamisch einem Dienst zugeordnet werden kann, ist vor der Verbindungsaufnahme deswegen immer die Service Datenbank zu befragen, um die korrekte Kanalnummer zu erhalten.

Abb. 7.13 zeigt, wie unterschiedliche Kanalschichten Datenströme multiplexen. Während der HCI Layer die Verbindung zu mehreren Geräten multiplext (Connection Handles), können über den L2CAP Layer unterschiedliche Dienste pro Gerät adressiert werden (PSM und CID). Dies wird in der Praxis verwendet, um zwischen der Service Datenbank (PSM 0×0001) und der RFCOMM-Schicht (PSM 0×0003) zu unterscheiden. Von der Service Datenbank abgesehen, verwenden die meisten Bluetooth Dienste die RFCOMM-Schicht und müssen deshalb noch zusätzlich durch unterschiedliche RFCOMM-Kanalnummern voneinander unterschieden werden.

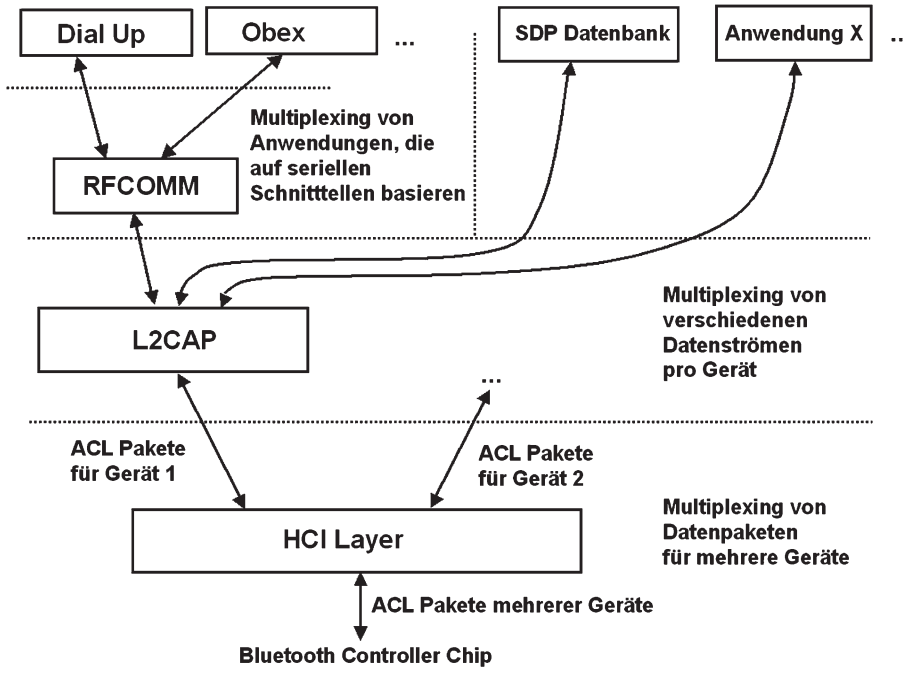

Abb. 7.13 Multiplexing auf den einzelnen Protokollschichten

Die RFCOMM Kanalnummer ermöglicht es außerdem, bis zu 30 RFCOMM Dienste zwischen zwei Geräten gleichzeitig zu verwenden. Somit ist es möglich, während einer Dial-Up Verbindung auch gleichzeitig Dateien mit dem Object Exchange Dienst (OBEX) zu übertragen. Da beide Dienste unterschiedliche RFCOMM Kanalnummern verwenden, können die RFCOMM Datenpakete der beiden Dienste im Zeitmultiplex übertragen werden und am Empfänger wieder dem richtigen Dienst zugestellt werden.

7.4.8 Aufbau einer Verbindung im Überblick

Abb. 7.14 zeigt den Aufbau einer Bluetooth Verbindung durch die unterschiedlichen Schichten noch einmal im Überblick. Um Kontakt zu einer Anwendung auf einem entfernten Bluetooth Gerät aufzunehmen, baut ein Endgerät zunächst eine ACL Verbindung auf. Nach der Konfiguration des ACL Übertragungskanals wird dann über den Protocol Service Multiplexer (PSM) eine L2CAP Verbindung zur Bluetooth Service Datenbank aufgebaut, um den Service Record der Anwendung anzufordern. Dieser enthält alle Informationen für den weiteren Verbindungsaufbau, wie beispielsweise, welche Protokolle auf höheren Schichten zu verwenden sind und wie diese konfiguriert werden. Nach erfolgreicher Übertragung des Service Records wird die L2CAP Verbindung wieder abgebaut, die ACL Verbindung bleibt jedoch zwischen den zwei Geräten bestehen.

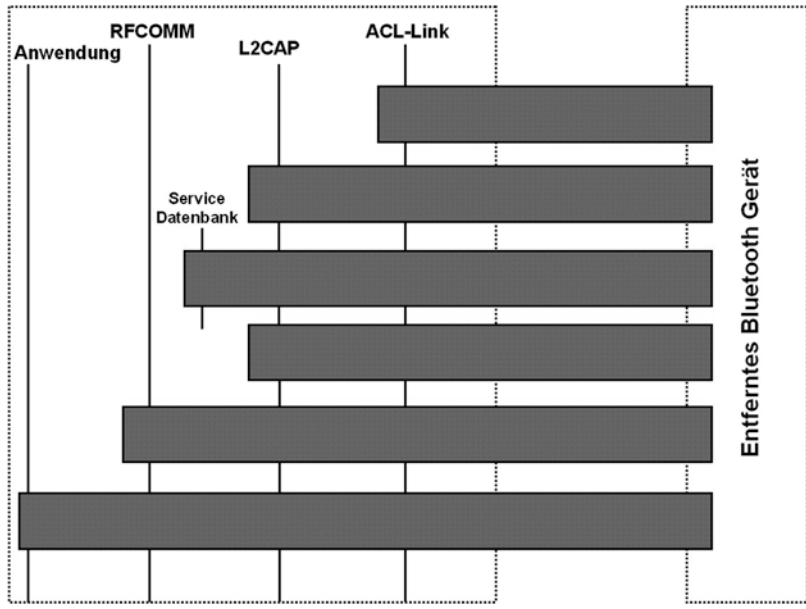

Abb. 7.14 Einzelne Stufen eines Bluetooth Verbindungsaufbaus

Über die ACL Verbindung wird jetzt Kontakt zur eigentlichen Anwendung aufgenommen. Dies geschieht im ersten Schritt durch Aufbau einer L2CAP Verbindung. Viele Anwendungen verwenden außerdem die RFCOMM Schicht, die serielle Schnittstellen bereitstellt. Aufgrund der beim RFCOMM Verbindungsaufbau übergebenen Kanalnummer kann der Bluetooth Stack schließlich die Verbindung zwischen dem RFCOMM Layer und der eigentlichen Anwendung, wie z. B. dem OBEX Dienst, herstellen. Wie die Anwendungsschichten der zwei Bluetooth Geräte miteinander kommunizieren, ist Sache der jeweiligen Anwendung und für alle bisher beschriebenen Schichten inklusive des RFCOMM Layers transparent. Um die Interoperabilität auch auf der Anwendungsschicht zu gewährleisten, definiert Bluetooth so genannte Profile, die in Abschn. 7.6 beschrieben werden.

7.5 Bluetooth Sicherheit

Da Bluetooth Funkwellen nicht an der Wohnungstür halt machen, spezifiziert der Bluetooth Standard eine Reihe von Sicherheitsfunktionen. Alle Verfahren sind optional und müssen beim Verbindungsaufbau oder während einer laufenden Verbindung nicht unbedingt verwendet werden. Diese Entscheidung wurde bewusst getroffen, da manche Dienste keine Sicherheitsfunktionen benötigen. Welche Dienste dies sind, liegt im Ermessen des Herstellers und des Anwenders. So kann sich der Hersteller eines Mobiltelefons z. B. entscheiden, einen eingehenden Dateitransfer ohne Authentifizierung der Gegenstelle zuzulassen. Die eingehende Datei wird dann in einem Zwischenspeicher gehalten und der Benutzer kann

dann auswählen, ob er die Datei speichern oder verwerfen möchte. Bei anderen Diensten, wie z. B. beim früher verwendeten Modemdienst, ist es hingegen gerade umgekehrt. Hier sollte immer eine Authentifizierung beim Verbindungsaufbau erfolgen, da sonst ein fremdes Gerät z. B. eine Internetverbindung ohne Wissen des Gerätebesitzers aufbauen könnte.

Die bei Bluetooth verwendeten SAFER+ (Secure And Fast Encryption Routine) Verschlüsselungsmechanismen wurden an der ETH Zürich entwickelt und sind öffentlich verfügbar. Bis heute wurden keine Methoden bekannt, diese zu kompromittieren. In der Praxis wurden jedoch zwischenzeitlich Schwachstellen beim einmaligen Aushandeln der Schlüssel gefunden. Diese erlauben es Angreifern, beim Abhören des gleich nachfolgend beschriebenen Pairing, die Schlüssel zu berechnen und Verbindungen dann zukünftig abzuhören. Aus diesem Grund wurden mit Bluetooth 2.1 neue Pairing Mechanismen eingeführt, die in Abschn. 7.5.2 beschrieben werden.

7.5.1 Pairing bis Bluetooth 2.0

Erster Schritt der Sicherheitsvorkehrungen, der einmalig durchgeführt werden muss, ist das so genannte Pairing zweier Endgeräte. Aus Sicht des Anwenders bedeutet ein Pairing von zwei Endgeräten, dass auf beiden Endgeräten eine identische PIN Nummer eingegeben werden muss. Diese wird im Anschluss verwendet, um auf beiden Seiten einen Link Key zu generieren. Der Link Key wird in beiden Endgeräten gespeichert und kann in Zukunft für die Authentifizierung und Verschlüsselung verwendet werden. Das Pairing der zwei Endgeräte läuft, wie in Abb. 7.15 gezeigt, in folgenden Schritten ab:

Um das Pairing zu starten, sendet das auslösende Endgerät eine LMP_IN_RAND Nachricht über eine neue aufgebaute ACL Verbindung an das andere Endgerät. Der Inhalt der Nachricht ist eine Zufallszahl. Mit dieser wird zusammen mit der PIN und der Geräteadresse ein Initialisierungskey generiert, der K_{init} genannt wird. Da die PIN nicht zwischen den Geräten ausgetauscht wird, kann K_{init} nicht von einem dritten Gerät berechnet werden.

Mit Hilfe von K_{init}, der auf beiden Seiten identisch ist, wird jetzt auf jeder Seite ein Teil eines Combination Keys erstellt. Dieser basiert auf K_{init}, der Geräteadresse eines der beiden Geräte und einer weiteren Zufallszahl, die aber nicht zwischen den zwei Geräten ausgetauscht wird. Im Anschluss werden die jeweils halben Combination Keys mit K_{init} noch XOR verknüpft und danach untereinander über LMP_COMB_KEY Nachrichten ausgetauscht. Die XOR Verknüpfung ist notwendig, um die zwei Combination Key Hälften nicht im Klartext über die Luftschnittstelle übertragen zu müssen.

Da K_{init} auf beiden Seiten bekannt ist, kann die XOR Verknüpfung wieder rückgängig gemacht werden und beide Seiten erhalten dann durch die Kombination der beiden Combination Key Hälften den endgültigen Link Key. Dieser ist zukünftig die Grundlage für die Authentifizierung und Verschlüsselung zwischen den zwei Geräten.

Da der mit dieser Methode generierte Link Key in beiden Endgeräten gespeichert wird, braucht das Pairing nur beim Aufbau der ersten Kommunikationsverbindung durchgeführt werden. Über die Endgeräteadresse der Gegenstelle kann bei der nächsten Verbindungs-

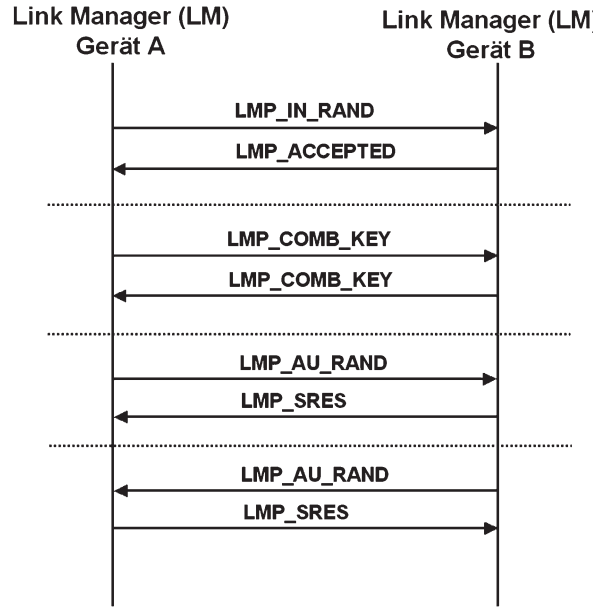

Abb. 7.15 Pairing zwischen zwei Bluetooth Geräten

aufnahme der Link Key dann auf beiden Seiten aus der Link Key Datenbank entnommen werden. Die Authentifizierung erfolgt dann ohne zutun des Anwenders.

Um zu überprüfen, ob der Link Key auf beiden Seiten richtig erzeugt wurde, findet im Anschluss an das Pairing eine gegenseitige Authentifizierung statt. Wie diese Abläuft, wird im nächsten Unterkapitel beschrieben. Wie in Abb. 7.15 ebenfalls zu sehen ist, wird das komplette Pairing von der Link Manager Schicht in den Bluetooth Chips der beiden Endgeräte durchgeführt. Über das HCI Interface muss für die Pairing Prozedur lediglich die PIN Nummer übergeben werden.

7.5.2 Pairing ab Bluetooth 2.1 (Secure Simple Pairing)

In 2005 entdeckten Yaniv Shaked und Avishai Wool einige Schwachstellen die es ermöglichen, nach dem Abhören der Pairing Prozedur die PIN und die Link Keys zu berechnen. Dies war wohl ein wichtiger Grund, warum mit Bluetooth 2.1 der Pairing Mechanismus komplett geändert wurde. Der neue Mechanismus trägt den Namen Secure Simple Pairing und umfasst eine Reihe unterschiedlicher Pairing Protokolle für unterschiedliche Sicherheitsanforderungen:

Das Numeric Comparison Protocol: Der wichtigste Unterschied dieses Pairing Verfahrens zum bisherigen Verfahren ist, dass statt einer PIN ein Public/Private Key Verfahren zusammen mit dem Elliptic Curve Diffie-Hellmann Kryptoalgorithmus verwendet wird. Jedes Gerät hat dazu einen privaten und öffentlichen (public) Schlüssel. Beim Pairing schickten beide Endgeräte jeweils ihre öffentlichen Schlüssel zur Gegenstelle, die damit eine Zufallszahl verschlüsselt und zurückschickt. Nach Empfang der verschlüsselten

Zufallszahl entschlüsseln die Endgeräte diese mit ihrem privaten Schlüssel und verwenden dann die Zufallszahlen um die Link Keys zu erzeugen. Die Ver- und Entschlüsselung funktioniert nur in eine Richtung, d. h. eine Nachricht, die mit dem öffentlichen Schlüssel chiffriert wurde, kann nur mit dem privaten Schlüssel wieder dechiffriert werden. Da die privaten Schlüssel niemals übertragen werden, kann somit kein anderes Gerät, welches das Pairing belauscht, die Nachrichten dekodieren und somit keine korrekten Link Keys erzeugen. Eine ähnliche Art der Authentifizierung findet sich auch bei Wireless LAN mit EAP-TLS im Enterprise Mode (vgl. Abschn. 4.3.7) sowie beim ersten Zugriff auf eine verschlüsselte Website mit Secure http (HTTPS, SSL/TLS).

Da sich die zwei Endgeräte bisher nicht kannten, könnte bei dieser Art des Pairing ein Angreifer ein Gerät zwischen A und B schalten und sich gegenüber A als B ausgeben und gegenüber B als A. Dies wird oft als Man in the Middle Attack (MITM) bezeichnet. Um diese Möglichkeit auszuschließen, geht das Numeric Comparison Protocol nach der Generierung der Link Keys noch einen Schritt weiter und beide Endgeräte errechnen eine 6-stellige Zahl, die dann dem Anwender gezeigt wird. Das Pairing ist erst dann abgeschlossen, wenn der Anwender auf beiden Endgeräten die Zahl bestätigt. Die Berechnungsvorschrift für die 6-stellige Zahl ist so gestaltet, dass bei einer MITM Attacke das zwischengeschaltete Endgerät diese Zahl nicht für beide Geräte berechnen kann. Die Bluetooth SIG gibt an, dass auf diese Weise die Chance eines erfolgreichen MITM Angriffs bei 1:1.000.000 liegt.

Das Just Works Protocol: Dieses Protokoll ist identisch zum Numeric Comparison Protokoll, es wird jedoch am Ende der Pairing Prozedur keine 6-stellige Zahl berechnet, die der Anwender auf beiden Endgeräten bestätigen muss. Dies bietet zwar keinen Schutz vor einem MITM Angriff, manche Endgeräte wie z. B. Headsets haben jedoch kein Display, um die 6-stellige Zahl darzustellen. Aus diesem Grund sollte ein Pairing für solche Geräte nur durchgeführt werden, wenn hinreichend sicher ist, dass kein Angreifer die Pairing Prozedur abhören und verändern kann. Da diese Schwachstelle nur den Pairing Prozess betrifft, sind alle später aufgebauten und verschlüsselten Verbindungen trotzdem sicher, und das Just Works Protocol bietet somit für die meisten Anwendungen ausreichend Sicherheit beim Pairing. Sollte während des Pairings eine MITM Attacke erfolgreich gewesen sein, muss der Angreifer jedoch bei jeder zukünftigen Kommunikation dabei sein, da sonst der Verbindungsaufbau fehlschlägt.

Das Passkey Protokoll: Bei diesem Protokoll wird ein Passkey (PIN) für die Authentifizierung verwendet. Für den Anwender ist diese Art des Pairing identisch zum bisherigen Verfahren. Die PIN wird jedoch während des Pairings nicht wie in Abschn. 7.5.1 gezeigt verwendet, sondern es kommt wiederum zu einem Public/Private Key Austausch in Verbindung mit jeweils unabhängigen Zufallszahlen auf beiden Seiten. Für jedes einzelne Bit wird eine verschlüsselte Bestätigung, die Commitment genannt wird, auf beiden Seiten generiert. Eingangsparameter für den dazu verwendeten Algorithmus sind auf beiden Seiten beide öffentlichen Schlüssel, eine auf beiden Seiten unterschiedliche Zufallszahl und das aktuelle Bit der PIN. Im ersten Schritt tauschen beide Endgeräte das Commitment für ein Bit aus. Danach schickt Endgerät A die verwendete Zufallszahl, damit Endgerät B das

Commitment über den Umkehralgorithmus überprüfen kann. War die Nachricht korrekt, schickt Endgerät B seine eigene Zufallszahl zurück, damit auch Gerät A überprüfen kann, ob das zuvor gesendete Commitment authentisch ist. Für das nächste Bit wird der Prozess in umgekehrter Richtung durchgeführt, d. h. Gerät B sendet als erstes sein Commitment. Ein Gerät in der Mitte kann bei diesem Prozess somit die Commitments nicht fälschen, da das PIN Bit erst aus dem Commitment zurückberechnet werden kann, nachdem im zweiten Schritt die Zufallszahlen ausgetauscht wurden. Da die Commitments alternierend sind, kann ein Angreifer also nur von jeder Seite ein Bit bekommen, bevor er selber zuerst ein Commitment schicken muss. Dies kann er jedoch nicht, da er nicht über das PIN Bit verfügt.

Das Out of Band Protokoll: Schließlich wurde mit Bluetooth 2.1 auch noch ein Verfahren spezifiziert, um die Authentifizierung nicht über den Bluetooth Funkkanal, sondern teilweise oder ganz über andere Übertragungswege durchzuführen. Beispielsweise kann diese Variante zusammen mit Near Field Communication (NFC) verwendet werden. Hierfür müssen sich die Geräte während des Pairings in unmittelbarer Nähe zueinander befinden, der Anwender hält die Geräte also in der Praxis zusammen. Dies schließt eine MITM Attacke aus, da ein eventueller Angreifer zwar potentiell den Nachrichtenaustausch abhören könnte, jedoch selber keine Möglichkeit hat, sich zwischen die zwei Teilnehmer zu schalten und Nachrichten zu fälschen. Der Bluetooth Standard unterstützt sowohl aktive NFC Chips, die senden und empfangen können, sowie passive NFC Chips, die nur senden können, wenn ihnen über die Antenne eine Spannung induziert wird. Dies ist notwendig, da manche Endgeräte wie z. B. Headsets keinen Platz für eine zusätzliche NFC Antenne haben. In solchen Fällen wird ein passiver NFC Chip z. B. auf dem Benutzerhandbuch oder der Verpackung angebracht. Während des Pairing Prozesses wird dann ein Bluetooth Endgerät mit aktivem NFC Chip, der sowohl senden als auch empfangen kann, an den passiven NFC Chip gehalten. Der passive NFC Chip überträgt dann alle notwendigen Informationen um ein Pairing ohne weitere Benutzerinteraktion durchzuführen.

NFC eignet sich neben dem Pairing auch für Anwendungen, in denen bei Berührung von zwei Geräten eine Aktion durchgeführt werden soll. Ein praktisches Beispiel ist der automatische Ausdruck eines Fotos auf einem Fotodrucker, da mit einem Mobiltelefon oder einem anderen Gerät aufgenommen wurde. Der Nutzer wählt das Bild auf seinem Telefon aus und hält das Telefon dann an den Fotodrucker. Beide Geräte erkennen sich dann über ihre NFC Schnittstelle und beginnen automatisch mit der Übertragung des Bildes.

7.5.3 Authentifizierung

War das Pairing zweier Geräte erfolgreich, können sich diese fortan beim Verbindungsaufbau über den Link Key authentifizieren. Dieser Vorgang funktioniert nach dem allgemeinen Challenge/Response Verfahren, dass z. B. auch bei GSM, GRPS und UMTS verwendet wird. Für die Authentifizierung werden bei Bluetooth drei Parameter benötigt:

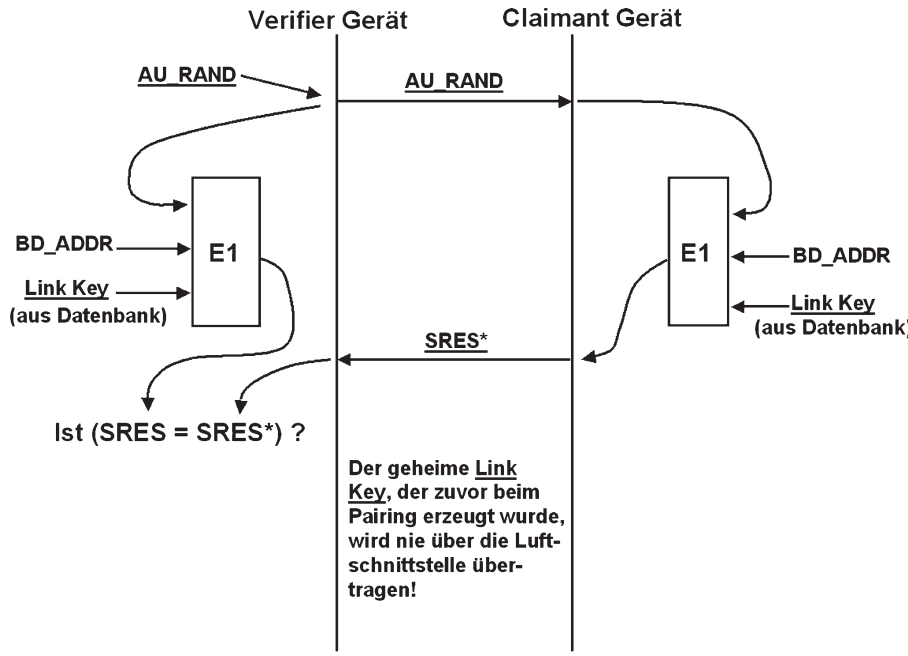

Abb. 7.16 Authentifizierung eines Endgeräts

- Eine Zufallszahl
- Die Bluetooth Adresse des Geräts, das die Authentifizierung auslöst (BD_ADDR).
- Der 128 Bit Link Key, der beim Pairing der Geräte erzeugt wurde.

Wie in Abb. 7.16 gezeigt, schickt das auslösende Endgerät (Verifier) für die Authentifizierung die Zufallszahl an die Gegenstelle (Claimant). Der Link Manager des Claimant Endgeräts verwendet daraufhin die BD_ADDR des Verifier Endgeräts, um den Link Key für diese Verbindung über das HCI Interface vom Host anzufordern.

Mit der Zufallszahl, der BD_ADDR, sowie dem Link Key, berechnet der Link Manager des Claimant nun eine Antwort, die Signed Response* (SRES*) genannt wird. Die so berechnete SRES* schickt der Link Manager danach an das Verifier Endgerät zurück. Dieses hat die gleiche Operation ausgeführt und seine eigene SRES errechnet. Die beiden Ergebnisse können nur identisch sein, wenn der Link Key auf beiden Seiten identisch war. Da der Link Key niemals über die Luftschnittstelle übertragen wird, kann sich kein Gerät erfolgreich authentifizieren, mit dem zuvor kein Pairing durchgeführt wurde.

7.5.4 Verschlüsselung

Nach erfolgreicher Authentifizierung können beide Endgeräte jederzeit die Verschlüsselung aktivieren oder deaktivieren. Als Schlüssel wird jedoch nicht der beim Pairing erzeugte Link Key verwendet. Stattdessen wird ein auf beiden Seiten der Verbindung

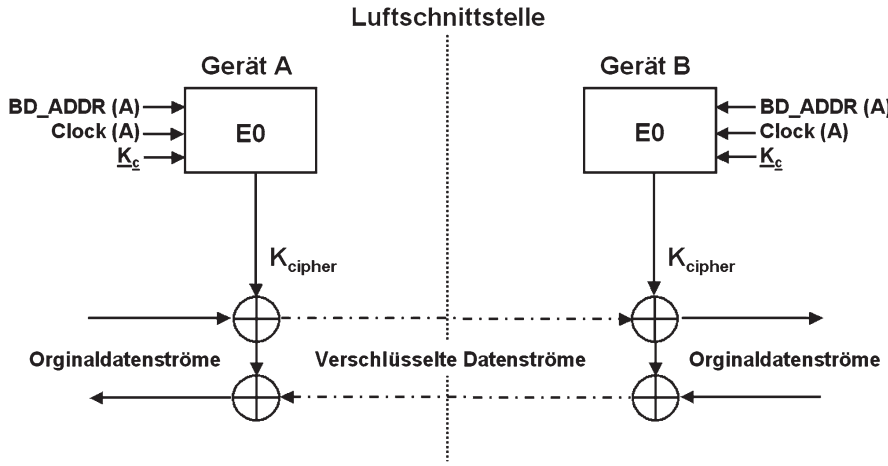

Abb. 7.17 Bluetooth Verschlüsselung mit einer Ciphersequenz

eigens bei der Aktivierung der Verschlüsselung generierter Ciphering Key benutzt. Wichtigster Parameter für die Erzeugung des Ciphering Keys ist neben dem Link Key der Verbindung eine Zufallszahl, die beim Start der Verschlüsselung zwischen den Link Managern ausgetauscht wird. Auf diese Weise ist gewährleistet, dass bei jeder Aktivierung der Verschlüsselung ein neuer Ciphering Key verwendet wird (Abb. 7.17).

Der Ciphering Key hat üblicherweise eine Länge von 128 Bit. Es können jedoch auch kürzere Ciphering Keys verwendet werden, wenn Bluetooth Chips für ein Land hergestellt werden, für das es Exportrestriktionen für starke Verschlüsselungskeys gibt.

Zusammen mit der Geräteadresse des Masters und den 26 untersten Bits der Master Echtzeituhr (Master Real Time Clock) dient der Ciphering Key als Eingangswert für den SAFER+ Algorithmus E0, der einen kontinuierlichen Bitstrom erzeugt. Da der aktuelle Wert der Master Real Time Clock auch dem Slave bekannt ist, kann auf beiden Seiten der Verbindung der gleiche Bitstrom generiert werden. Der Bitstrom wird dann über bitweise Modulo-2 Operationen mit dem zu verschlüsselnden Datenstrom kombiniert. Verschlüsselt wird der komplette Teil des ACL Nutzdatenpaketes inklusive der CRC Checksume vor dem optionalen Hinzufügen einer Forward Error Correction (FEC).

7.5.5 Autorisierung

Ein weiteres wichtiges Konzept der Bluetooth Sicherheit ist die Autorisierung des Nutzers für einen Dienst. Dieser weitere Schritt ist nötig, um manche Dienste nicht allen, sondern nur bestimmten Endgeräten zugänglich zu machen. So könnte man auf einem PC einem Nutzer eines anderen Bluetooth Geräts das Recht einräumen, auf ein freigegebenes Verzeichnis Dateien abzulegen oder von dort abzuholen. Der Dateitransfer Dienst (OBEX) ist also für diesen Nutzer aktiviert.

Über die Autorisierung kann für jeden Dienst einzeln festgelegt werden, welche bekannten Bluetooth Endgeräte auf diesen zugreifen dürfen. Es bleibt dabei dem Hersteller eines Bluetooth Gerätes überlassen, wie diese Funktionalität genutzt wird. Manche Mobiltelefonhersteller beispielsweise erlauben jedem entfernten Endgerät, mit dem ein Pairing erfolgreich durchgeführt wurde, die Benutzung des Dial-Up Dienstes. Andere Mobiltelefonhersteller bauen jedoch noch eine zusätzliche Sicherung ein und fordern vom Nutzer des Mobiltelefons eine explizite Autorisierung des Verbindungswunsches. Dies geschieht über eine Nachricht auf dem Display des Mobiltelefons, die der Besitzer des Mobiltelefons bestätigen muss.

Bluetooth Stacks auf PCs bieten meist eine sehr flexible Autorisierungsfunktionalität an. Dienste können dort sehr flexibel konfiguriert werden:

- Dienst ohne Authentifizierung und Autorisierung nutzbar.
- Dienst darf von allen authentifizierten Geräten ohne weitere Autorisierung verwendet werden. Dies setzt ein einmaliges Pairing voraus.
- Dienst darf nach Authentifizierung und Autorisierung einmalig oder für eine bestimmte Zeitdauer verwendet werden.
- Dienst darf von einem bestimmten Endgerät nach Authentifizierung und einmaliger Autorisierung immer verwendet werden, eine nochmalige Autorisierung ist nicht erforderlich.

Zusätzlich bieten manche Bluetooth Stacks auf dem PC an, immer eine Information auf dem Bildschirm anzuzeigen, wenn ein Dienst von einem entfernten Gerät aufgerufen wird. Dies dient nur zur Information des Nutzers des PCs, der Zugriff wird automatisch gewährt.

7.5.6 Sicherheitsmodi

Zu welchen Zeitpunkten beim Verbindungsaufbau eine Authentifizierung, Verschlüsselung und Autorisierung durchgeführt werden, ist abhängig von der Implementation des Bluetooth Stacks und der Konfiguration durch den Anwender. Der Bluetooth Standard gibt dazu drei mögliche Konfigurationen vor:

Im Sicherheitsmodus 1 (Security Mode 1) findet keine Authentifizierung statt und die Verbindung wird nicht verschlüsselt. Dieser Sicherheitsmodus eignet sich z. B. für die Adress- oder Terminübertragung zwischen zwei Endgeräten. Oft kennen sich die Teilnehmer nicht und es wäre zu umständlich, mit den Geräten vor dem Austausch einer elektronischen Visitenkarte ein Pairing durchzuführen. Die elektronische Visitenkarte wird dann meist von den Geräten in ein extra Verzeichnis kopiert und erst in den Adresskalender aufgenommen, wenn der Benutzer dies bestätigt.

Im Sicherheitsmodus 2 bestimmt der Anwender, ob für eine Verbindung eine Authentifizierung, Verschlüsselung und Autorisierung nötig ist. Viele Bluetooth PC Benutzerober-

flächen erlauben diese Konfiguration individuell für jeden einzelnen Dienst. Sicherheitsmodus 1 entspricht Sicherheitsmodus 2 eines Dienstes, der weder Authentifizierung noch Verschlüsselung aktiviert hat.

Im Sicherheitsmodus 3 wird beim Aufbau jeder Verbindung automatisch eine Authentifizierung und Verschlüsselung vom Bluetooth Chip hergestellt. Dies geschieht schon während der ersten Link Manager Kommunikation, also noch vor dem Aufbau einer L2CAP Verbindung. Bei einer eingehenden Kommunikation fordert deshalb der Bluetooth Controller über die HCI Schnittstelle den Link Key für eine neue Verbindung an. Wurde mit dem entfernten Gerät bisher kein Pairing durchgeführt, kann der Bluetooth Host dem Controller keinen Link Key zurückgeben. In diesem Fall schlägt der Verbindungsaufbau fehl. Sicherheitsmodus 3 ist also vor allem für Geräte gedacht, die nur mit Geräten kommunizieren, mit denen zuvor ein Pairing durchgeführt wurde. Für Mobiltelefone, die auch nicht authentifizierte Verbindungen z. B. für die Übertragung von Adressdaten erlauben, ist dieser Modus nicht geeignet.

Sicherheitsmodus 4 ist dem Service Level Enforced Security Mode 2 sehr ähnlich, wurde jedoch für die neuen Pairingmechanismen für Bluetooth 2.1 spezifiziert (vgl. Abschn. 7.5.2). In diesem Modus wählt ein Dienst aus, welche Security Kategorie er für das Pairing verlangt:

- Es wird ein gesicherter Link Key verlangt (Numeric Comparsion, Out of Band oder Passkey Protokoll sind notwendig)
- Es wird nur ein nicht gesicherter Link Key benötigt (Just Works Protokoll)
- Der Dienst benötigt keine Sicherheit

7.6 Bluetooth Profile

Wie in der Einleitung dieses Kapitels gezeigt, ist Bluetooth für eine Vielzahl sehr unterschiedlicher Anwendungen geeignet. Diese Anwendungen haben immer eine Server- und eine Client Seite. Ein Client nimmt durch Aufbau einer Bluetooth Verbindung Kontakt zum Master auf und die Datenübertragung beginnt. Bei den meisten Bluetooth Anwendungen sind die Aufgaben der Masterseite und der Clientseite unterschiedlich. Bei der Übertragung eines Adressbucheintrags beispielsweise, nimmt der Client Kontakt mit dem Server auf. Der Client überträgt einen Termin, ist also eine Sendekomponente, der Server empfängt ihn, ist also eine Empfangskomponente. Um zu gewährleisten, dass der Client auch mit einem Server kommuniziert, der von einem anderen Hersteller programmiert wurde, spezifiziert der Bluetooth Standard so genannte Bluetooth Profile. Für jede Anwendung (Headset, Termin- und Dateiübertragung, Audiostreaming, etc.) gibt es ein Bluetooth Profil, das genau beschreibt, wie die Serverseite und die Clientseite miteinander kommunizieren. Unterstützen zwei Endgeräte das gleiche Bluetooth Profil, ist die Interoperabilität gewährleistet.

Anmerkung: Das Client/Server Prinzip der Bluetooth Profile darf nicht mit dem Master/Salve Konzept der unteren Bluetooth Protokollschichten verwechselt werden. Beim Master Slave Konzept geht es um die Kontrolle des Piconetzes, also wer zu welcher Zeit senden darf, während das Client/Server Prinzip einen Dienst und einen Nutzer des Dienstes beschreibt. Ob nun das Bluetooth Endgerät, auf dem der Server eines Dienstes läuft, der Master oder der Slave im Piconetz ist, spielt keine Rolle.

Nachfolgende Tabelle gibt einen Überblick über zahlreiche Bluetooth Profile für die verschiedensten Anwendungen. In der Praxis ist heute jedoch zu beobachten, dass sich die Bluetooth Nutzung auf nur noch wenige Profile beschränkt. Diese werden in den nachfolgenden Unterkapiteln genauer beschrieben.

Profilname	Anwendungsgebiet
Headset Profile	Kabellose Headsets für Mobiltelefone.
Hands-Free Profile	Verbindung zwischen Freisprecheinrichtung und Mobiltelefon.
SIM-Access Profile	Zugriff einer Freisprecheinrichtung auf die SIM-Karte eines Mobiltelefons.
Human Interface Device (HID) Profile	Anbindung von Mäusen, Tastaturen und Joysticks an Endgeräte wie PCs, Notebooks und Smartphones.
File Transfer Profile	Übertragung von Dateien zwischen Bluetooth Geräten.
Object Push Profile	Einfache Übertragung von Dateien zwischen Bluetooth Geräten für Ad-Hoc Datenaustausch.
Advanced Audio Distribution Profile	Profil für die Übertragung von Audio Streaming Dateien (z. B. MP-3).
Dial Up Networking (DUN) Profile	Bluetooth Verbindung zwischen einem Mobiltelefon und einem externen Gerät wie PC oder Notebook.
FAX Profile	Profil für FAX Übertragung.
LAN Access Profile	IP Verbindung zwischen Smartphone, PC oder Notebook zu einem Local Area Network (LAN) und dem Internet.
Personal Area Network (PAN) Profile	Wie LAN Access Profile, es wird jedoch eine Ethernet Netzwerkkarte auf dem PAN Gerät simuliert.
Synchroization Profile	Synchronisation von Personal Information Manager (PIM) Anwendungen für Adressen, Termine, Notizen, etc.
Basic Imaging Profile	Übertragung von Bildern, für den Einsatz mit Digitalkameras gedacht.
Hard Copy Cable Replacement Profile	Kabelersatz zwischen Drucker und einem Endgerät (z. B. PC).
Basic Printing Profile	Drucken ohne Druckertreiber von mobilen Geräten wie Smartphones an beliebigen Druckern.
Unrestricted Digital Information Profile	Übertragung von breitbandigen leitungsvermittelten Verbindungen zwischen einem Endgerät und einem 3G Mobiltelefon.

7.6.1 Grundlegende Profile: GAP, SDP und Serial Profile

Bluetooth spezifiziert zwei Profile, die keine eigentlichen Anwendungen aus Sicht des Benutzers darstellen. Das Generic Access Profile (GAP) legt fest, wie zwei Geräte in unterschiedlichen Situationen Kontakt miteinander aufnehmen und wie sie sich dabei verhalten sollen. Das Profil beschreibt unter anderem:

- Die Präsentation von Bluetooth spezifischen Parametern wie der Geräteadresse (BD_ADDR) oder der PIN für den Anwender.
- Sicherheitsaspekte (Security Mode 1–3)
- Verhalten im Idle Mode (z. B. Inquiry, Device Discovery)
- Verbindungsaufbau

Durch das GAP Profil kann somit sichergestellt werden, dass sich die Benutzeroberflächen für die Konfiguration des Bluetooth Stacks von verschiedenen Endgeräten in den wichtigsten Punkten sehr ähnlich sind. Außerdem wird durch das GAP Profil erreicht, dass bei der Verbindungsaufnahme genau spezifiziert ist, welche Aktionen und Nachrichten in welcher Reihenfolge durchgeführt werden.

Wie in Abschn. 7.4.7 gezeigt, besitzt ein Bluetooth Endgerät eine Service Datenbank, in der jeder Server-Dienst alle wichtigen Informationen für die Verbindungsaufnahme hinterlegen kann. Über das Service Discovery Profil (SDP) wird festgelegt, wie auf diese Datenbank zugegriffen werden kann, und wie und in welcher Struktur die nachfolgend vorgestellten Profile ihre Informationen in der Service Datenbank hinterlegen.

Das Serial Port Profile (SPP) ist ein grundlegendes Profil, auf dem zahlreiche nachfolgend vorgestellte Profile aufbauen. Wie der Name schon andeutet, stellt dieses Profil eine serielle Schnittstelle für beliebige Anwendungen zur Verfügung. Es verwendet dazu die in Abschn. 7.4.8 vorgestellte RFCOMM Schicht. Über das Serial Port Profile können beliebige Anwendungen, die Daten über eine serielle Schnittstelle übertragen, kommunizieren. Anpassungen der Anwendungen an Bluetooth sind nicht notwendig, da aus Ihrer Sicht auf eine ganz normale serielle Schnittstelle zugegriffen wird. Abb. 7.18 zeigt den Protokollstack, den das Serial Port Profile verwendet.

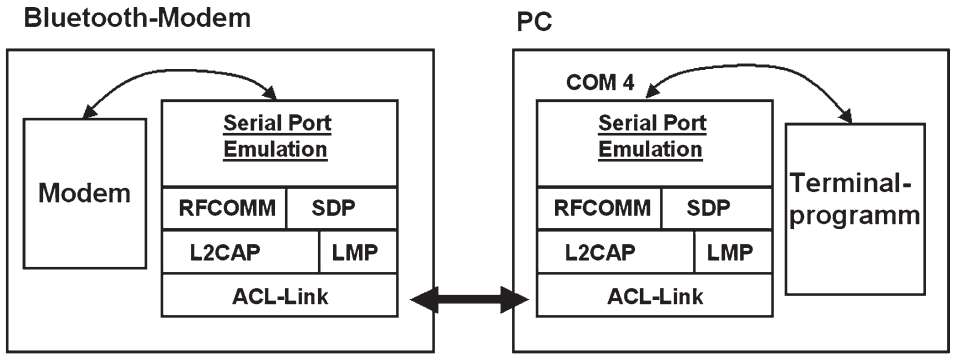

Abb. 7.18 Das SPP stellt eine serielle Schnittstelle zur Verfügung

7.6.2 Object Exchange Profile: FTP, Object Push und Synchronize

Um strukturierte Objekte wie Dateien, Visitenkarten, Termine, Adressbucheinträge oder generell Objekte zu übertragen, wird das Object (OBEX) Exchange Profile verwendet (Abb. 7.19).

Die Verbindung zwischen zwei Geräten besteht bei diesen Profilen nur während der Übertragung eines oder mehrerer unmittelbar aufeinander folgender Objekte und wird danach sofort wieder abgebaut. Zu diesem Zweck definiert der Bluetooth Standard als Grundlage für weitere Profile das General Object Exchange (OBEX) Profile (GOEP), das auf den L2CAP und RFCOMM Schichten aufsetzt. Drei weitere Object Exchange (OBEX) Profile verwenden dann dieses Profil für spezifische Dienste.

Für die Übertragung einer oder mehrerer Dateien, oder sogar eines ganzen Verzeichnisbaumes, wurde das File Transfer Profile (FTP) entwickelt. Dieses sollte nicht mit dem File Transfer Protocol aus der TCP/IP Welt verwechselt werden, das ebenfalls mit FTP abgekürzt wird.

Eingesetzt wird das OBEX FTP Protokoll hauptsächlich, um zwischen PCs und Smartphones Dateien auszutauschen. Diese können sich an einem beliebigen Ort innerhalb eines Dateisystems befinden. Zu diesem Zweck definiert das allgemeine OBEX Profil (GOEP) die Kommandos CONNECT, DISCONNECT, PUT, GET, SETPATH und ABORT, die binär kodiert über eine aufgebaute RFCOMM Verbindung zur Gegenstelle übertragen werden. Manche PC Bluetooth Stacks klinken das Dateisystem einer Bluetooth Gegenstelle, ähnlich einer normalen Netzwerkverbindung, in den Verzeichnisbaum des lokalen Dateimanagers ein. Klickt der Benutzer das Bluetooth Gerät an, wird über das allgemeine OBEX GET Kommando das Root-Directory des entfernten Bluetooth Gerätes angefordert und dann im Dateimanager dargestellt. Der Anwender hat dann die Möglichkeit, eine oder mehrere Dateien auszuwählen und auf den lokalen PC zu übertragen. Auch diese Aktion wird in ein GOEP GET Kommando umgesetzt. Der Anwender kann auch eine Datei in ein Verzeichnis eines anderen Bluetooth Gerätes kopieren. Zu diesem Zweck wird das allgemeine OBEX PUT Kommando verwendet.

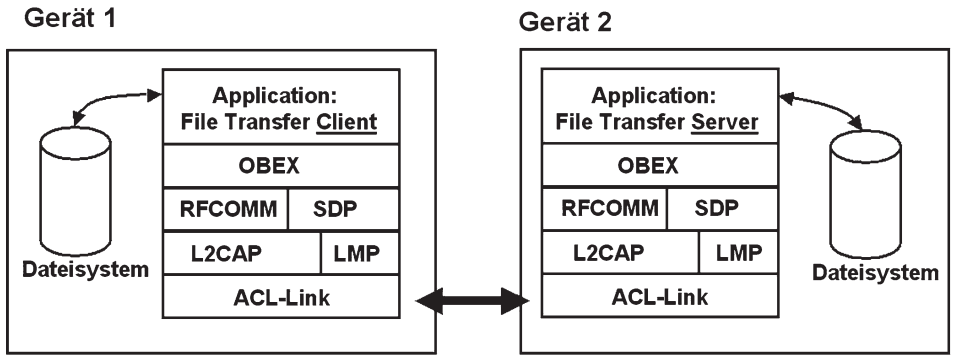

Abb. 7.19 OBEX mit File Transfer Profile als Anwendung

Wechselt der Anwender in ein Unterverzeichnis, wird in dieses über das OBEX SETPATH Kommando verzweigt und dessen Inhalt anschließend über das allgemeine OBEX GET Kommando angefordert. Wie das nachfolgende Beispiel in der Textbox zeigt, wird der Inhalt eines Verzeichnisses in lesbarer Form als XML Beschreibung übertragen.

Im OBEX Protokoll Layer werden CONNECT, DISCONNECT, PUT, GET, SETPATH und ABORT Kommandos und die entsprechenden Antworten darauf als Pakete behandelt. Der Wert des ersten Byte des Pakets beschreibt die Art des Kommandos. Nach einem zwei Byte Längenfeld folgen dann die Parameter des Kommandos. Ein Parameter kann z. B. ein Verzeichnisname, eine Verzeichnisauflistung oder eine angeforderte Datei sein. Diese Parameter werden im Standard etwas verwirrend als Header bezeichnet. Um die Art der Parameter auseinander halten zu können, hat jeder Parameter im ersten Byte eine Typinformation. Der Typ eines Parameters kann z. B. „Dateiname“ oder „Body“ (also die eigentliche Datei) sein.

```
<xml version="1.0">
<!DOCTYPE folder-listing SYSTEM „obex-folder listing.dtd">
<folder-listing-version="1.0">
   <folder name="Camera" modified="2004117T100840"
    user perm="RWD" group perm"W" />
   <folder name="other pics" modified="2004117T13321"
    user perm="RWD" group perm"W" />
</folder-listing>
```

Die maximale Paketgröße beträgt 64 kByte. Um größere Dateien (also Header vom Typ „Body“) zu übertragen, wird die Datei automatisch vom OBEX Layer in mehrere Pakete aufgeteilt.

Während FTP in der Praxis heute an Bedeutung verloren hat, findet die etwas einfachere Anwendung des General Object Exchange Profils, das Object Push Profile, in der Praxis durchaus noch Anwendung. Dieses wird z. B. verwendet, wenn der Benutzer eines Mobiltelefons einen Kalendereintrag, einen Adressbucheintrag oder eine Datei über Bluetooth zu einem anderen Gerät übertragen möchte. Die Funktionsweise dieses Profils ist identisch zum File Transfer Profil, es verwendet ebenfalls die allgemeinen OBEX Kommandos wie PUT und GET. Das Object Push Profile unterstützt jedoch keine Verzeichnisoperationen und Löschen von Dateien. Auf diese Weise wird erreicht, dass der Benutzer beim Senden der Informationen möglichst wenige Entscheidungen treffen muss und der Vorgang somit schnell durchgeführt werden kann (Abb. 7.20).

Viele Endgeräte erlauben einen eingehenden Object Push Transfer ohne vorherige Authentifizierung und Verschlüsselung. Das empfangene Objekt wird dann nach Erhalt zunächst in einen Zwischenpuffer gelegt und erst nach Bestätigung des Benutzers in den Terminkalender, in das Adressbuch, oder, im Falle einer Datei, in ein Verzeichnis kopiert.

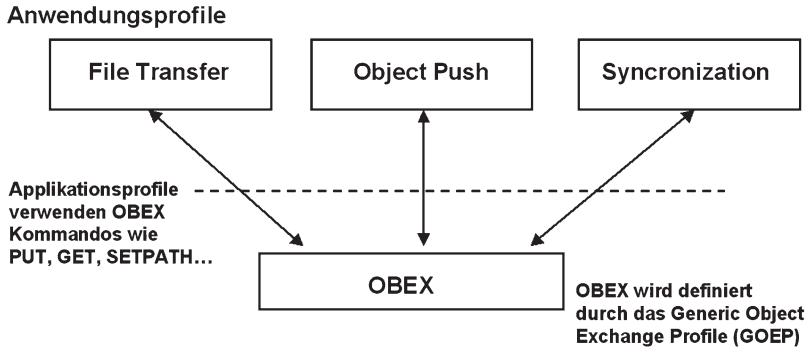

Abb. 7.20 Zusammenhang zwischen OBEX, GOEP, FTP, Object Push und Synchronization Profile

Für die Übertragung von Kalender- und Adressbucheinträgen schreibt das Object Push Profile das vCalendar, bzw. das vCard Format vor (www.imc.org). Dies ist Voraussetzung, um Adressbuch- und Kalendereinträge zwischen beliebigen Programmen und Endgeräten austauschen zu können. Bei anderen Objekten, wie z. B. Bildern, kann anhand der Endung des Dateinamens erkannt werden, um welche Art Datei es sich handelt.

Obwohl das Profil „Object Push" heißt, spezifiziert es auch optional eine Business Card Pull Funktion. Mit dieser Funktion kann man eine zuvor hinterlegte Standardvisitenkarte von einem Gerät anfordern. Die Business Card Exchange Funktion ergänzt diese Funktion, in dem nicht nur eine Visitenkarte angefordert wird, sondern auch die bei sich hinterlegte Visitenkarte automatisch dem anderen Gerät geschickt wird.

Das dritte Profil, das auf GOEP aufsetzt, ist das Synchronization Profile. Wie das File Transfer Profil hat sich auch dieses Profil in der Praxis nicht weit verbreitet, soll aber vollständigkeitshalber trotzdem erwähnt werden. Das Synchronization Profil ermöglicht den automatischen Abgleich von Objekten wie Terminkalender- und Adressbucheinträgen, sowie Notizen zwischen zwei Geräten. Auch dafür werden wieder die allgemeinen OBEX Kommandos wie GET und PUT verwendet. Gegenüber dem Object Push Profil, über das vom Anwender nur ausgewählte Objekte, wie z. B. ein Adressbucheintrag, zu einem anderen Gerät übertragen werden können, spezifiziert das Synchronization Profile, wie der komplette Datenbestand einer Datenbank synchronisiert werden kann. Bei der ersten Synchronisation wird einmalig der komplette Datenbestand in beide Richtungen übertragen, bei allen folgenden Synchronisationen werden dann nur noch die geänderten Objekte übertragen. Zu diesem Zweck führen beide Geräte eine Protokolldatei über alle Änderungen. Damit Anwendungen unterschiedlicher Hersteller ihre Datenbankeinträge austauschen können, werden wie auch im Object Push Profil standardisierte Formate wie vCard oder vCalendar verwendet.

Der Bluetooth Standard definiert den Ablauf der Synchronisation nicht selbst, sondert verwendet dazu das Synchronisationssystem, das im IrMC Standard der Infrared Data Association (www.irda.org) definiert wurde.

7.6.3 Headset, Hands-Free und SIM-Access Profile

Drahtlose Headsets für Mobiltelefone waren die ersten Geräte, die mit Bluetooth Funktionalität auf den Markt kamen. Für die Sprachverbindung zwischen Mobiltelefon und Headset wird das Headset Profil verwendet. Dieses Profil ist eine Besonderheit, denn es verwendet als eines der wenigen Profile auch SCO oder eSCO Pakete (vgl. Abschn. 7.4.1). Mit diesen wird zwischen Mobiltelefon und Headset ein Sprachkanal mit 64 kbit/s aufgebaut. Sind Mobiltelefon und Headset kompatibel zu Bluetooth 1.2, werden automatisch eSCO Pakete verwendet, die Verbindung profitiert dann von automatischer Fehlerkorrektur und Adaptive Frequency Hopping (AFH). Diese in Bluetooth 1.2 eingeführten Funktionalitäten steigern die Sprachqualität vor allem dann wesentlich, wenn die Bluetooth Verbindung aufgrund eines großen Abstands, einer geringen Sendeleistung, oder durch Hindernisse nicht optimal ist.

Um ein Headset mit einem Mobiltelefon verwenden zu können, müssen die zwei Geräte einmalig miteinander ein Pairing durchführen. Danach versucht das Mobiltelefon bei jedem eingehenden Anruf automatisch, eine Verbindung zum Headset herzustellen. Für die Signalisierung zwischen Headset und Mobiltelefon, das im Headset Profil als Audio Gateway (AG) bezeichnet wird, wird eine ACL Verbindung verwendet. Wie in Abb. 7.21 zu sehen ist, wird für die Signalisierungsverbindung auf höheren Schichten L2CAP und RFCOMM verwendet.

Um Kommandos und die dazugehörigen Antworten zwischen Audio Gateway und Headset auszutauschen, wird das von Modems bekannte AT-Kommandoset verwendet. Das Headset Profil beschränkt sich jedoch auf nur wenige Kommandos. Wie in Abb. 7.22 zu sehen ist, baut das Audio Gateway bei einem eingehenden Anruf zuerst eine Signalisierungsverbindung auf (ACL) und sendet über den Signalisierungskanal den String „RING“. Das Headset benachrichtigt daraufhin den Anwender über den eingehenden Anruf, in dem z. B. eine Melodie gespielt wird. Der Nutzer kann dann den Anruf durch Betätigen einer

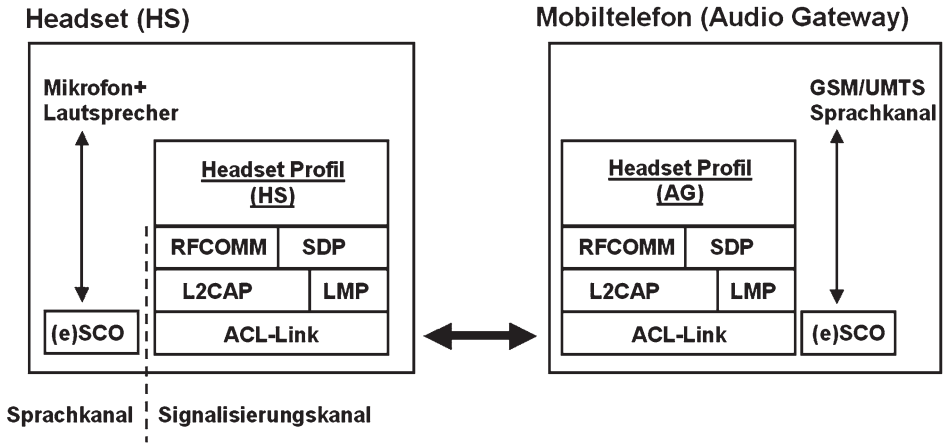

Abb. 7.21 Headset Protokollstack

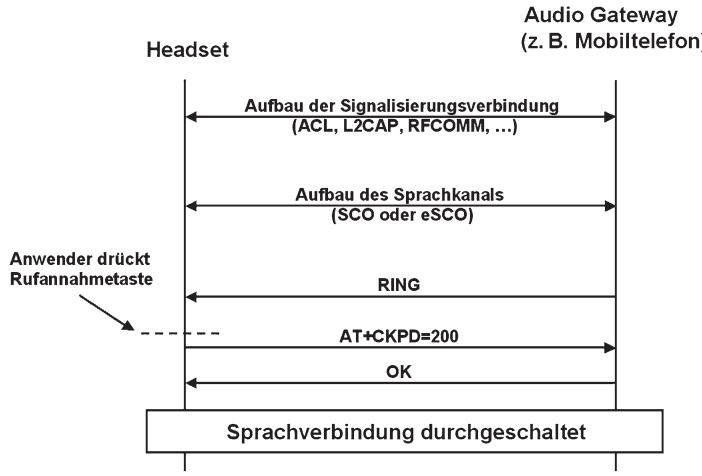

Abb. 7.22 Aufbau von Signalisierungs- und Sprachverbindung

Taste am Headset annehmen. Das Betätigen der Taste bewirkt, dass das Headset das AT-Kommando at+ckpd = 200 an das Audio Gateway zurückschickt. Dieses nimmt daraufhin das Gespräch an und stellt es zum Headset durch.

Um ein abgehendes Gespräch zu führen, kann umgekehrt auch das Headset eine Verbindung zum Audio Gateway herstellen. Zusammen mit einer im Audio Gateway (also im Mobiltelefon) vorhandenen Sprachwahlfunktion lassen sich somit abgehende Gespräche über das Headset starten, ohne das Mobiltelefon in die Hand zu nehmen.

Da die Bedienmöglichkeiten durch die Größe des Headsets begrenzt sind, bietet das Headset Profil außer der Gesprächsfunktionalität nur noch die Steuerung der Lautstärke. Dies geschieht über die Befehle +vgm für die Lautstärke des Mikrofons und mit +vgs für die Lautstärke des Lautsprechers. Mit diesen Befehlen kann also vom Mobiltelefon aus die Lautstärke im Headset geändert werden.

Ein Headset kann auch mit einem PC gekoppelt werden, falls der Bluetooth Stack des PCs das Headset Profil unterstützt und die Rolle des Audio Gateways übernehmen kann. Auf diese Weise kann das Headset z. B. zusammen mit einer Voice over IP Software verwendet werden. Außerdem ist es durch die Umleitung der Soundkarten Ein- und Ausgänge auf das Headset theoretisch auch möglich, Musik, MP3 Streams, etc. über das Headset abzuspielen. Dies macht jedoch wenig Sinn, da der SCO Kanal auf 64 kbit/s begrenzt ist und nur für Sprachtelefonie ausgelegt ist. In der Praxis bedeutet dies, dass das Audiosignal nur mono übertragen wird und das Frequenzband auf 300-3400 Hz begrenzt ist.

Stark verwandt mit dem Headset Profil ist das Hands-Free Profil. Bei der Entwicklung dieses Profils standen jedoch nicht Headsets im Vordergrund, sondern KFZ-Freisprecheinrichtungen. Wichtigste Aufgabe des Hands-Free Profil ist das Ersetzen der Kabelverbindung zwischen Freisprecheinrichtung und Mobiltelefon. Auf diese Weise muss das Mobiltelefon bei Fahrtantritt nicht in einer Halterung festgemacht werden und kann sich während der Fahrt an einer beliebigen Stelle im Auto befinden. Diese Aufgabe könnte auch mit dem Headset Profil bewerkstelligt werden. Da Freisprecheinrichtungen aber

heute weit mehr Funktionen bieten, als nur an- und abgehende Gespräche zu führen, wurde das Hands-Free Profil definiert.

Die grundsätzliche Funktionsweise des Hands-Free Profil ist mit dem Headset Profil identisch. Kommandos und entsprechende Antworten werden zwischen Freisprecheinrichtung (Hands-Free Unit) und dem Mobiltelefon (Audio Gateway) ebenfalls über AT-Kommandos ausgetauscht. Außerdem wird ebenso wie beim Headset Profil der Sprachkanal über eine SCO oder eSCO Verbindung geleitet. Zusätzlich zu den Funktionen des Headset Profils bietet das Hands-Free Profil auch folgende Möglichkeiten:

- Die Übertragung der Rufnummer des Anrufers an die Freisprecheinrichtung (CLIP Funktion).
- Abweisen von ankommenden Gesprächen von der Freisprecheinrichtung aus.
- Wählen einer Telefonnummer von der Freisprecheinrichtung.
- Gespräch halten sowie Dreierkonferenzsteuerung.
- Übertragung von Statusinformationen wie verbleibende Batteriekapazität und GSM/UMTS Empfangsstärke des Mobiltelefons.
- Roaminganzeige.
- Deaktivieren der optionalen Echounterdrückung im Endgerät, falls dies vom Endgerät unterstützt wird. Dies ist sinnvoll, wenn die Freisprecheinrichtung eine eigene Echounterdrückung besitzt.

Seit Version 1.6 des Hands-Free Profile ist optional auch die Verwendung eines mono Wideband Sprachcodecs (mSBC) möglich und erlaubt somit eine bessere Sprachqualität, falls das Netzwerk AMR-Wideband unterstützt (vgl. Kap. 3). Bei Headsets wird diese Erweiterung manchmal auch als „HD Voice“ kompatibel bezeichnet. Eine weitere Lösungsmöglichkeit für die gleichzeitige Verwendung eines Headsets und einer KFZ-Freisprecheinrichtung bietet das SIM-Access Profil. Im Unterschied zum Headset und Hands-Free Profil dient das Mobiltelefon beim SIM-Access Profil nicht als Audio Gateway, und somit als Brücke zum Mobilfunknetzwerk, sondern stellt nur die SIM Karte einem externen Gerät zur Verfügung. Abb. 7.22 zeigt dieses Szenario. Das externe Gerät, in den meisten Fällen also eine KFZ-Freisprecheinrichtung, enthält ein eigenes GSM/UMTS Mobiltelefon, jedoch ohne SIM Karte. Wird die Freisprecheinrichtung bei Fahrtantritt aktiviert, wird per Bluetooth Kontakt zum gekoppelten Mobiltelefon hergestellt. Durch die Aktivierung des SIM-Access Servers im Mobiltelefon wird automatisch der Mobilfunkteil deaktiviert. Dies ist notwendig, da die Mobiltelefoneinheit in der Freisprecheinrichtung fortan die Kommunikation mit dem Mobilfunknetzwerk übernimmt. Ein großer Vorteil dieser Methode ist weiterhin, dass die Freisprecheinrichtung auch an die KFZ-Spannungsversorgung und an eine Außenantenne angeschlossen ist. Dies können Headset und Hands-Free Profil nicht bieten.

Abb. 7.22 zeigt außerdem den für das SIM-Access Profil verwendeten Protokollstack. Auf der L2CAP Verbindung wird der RFCOMM Layer für eine serielle Übertragung zwischen Freisprecheinrichtung (SIM-Access Client) und Mobiltelefon (SIM-Access Server) verwendet. Neben SIM-Access Profil Kommandos für die Aktivierung,

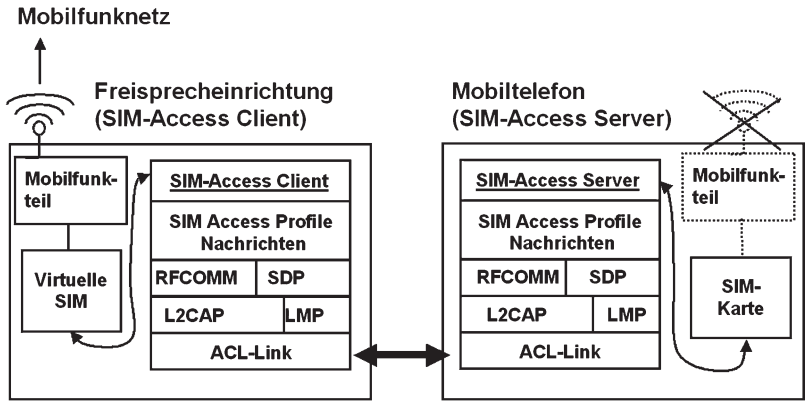

Abb. 7.23 Funktionsweise des SIM-Access Profils

Deaktivierung und den Reset der SIM-Karte werden über den Bluetooth Kanal auch SIM-Karten Kommandos und Antwortnachrichten ausgetauscht. Kommandos und Antwortnachrichten werden als Application Protocol Data Units (APDUs) übertragen. Diese wurden bereits in Abschn. 1.10 beschrieben und in den Abb. 1.49 und 1.50 dargestellt. Statt die APDUs also zwischen Mobilfunkteil und SIM Karte des Mobiltelefons über das elektrische Interface auszutauschen, werden mit dem SIM-Access Profil die APDUs über die Bluetooth Schnittstelle ausgetauscht. Für die Software der Freisprecheinrichtung, die auf dem SIM-Access Profile aufsetzt, ist es also völlig transparent, dass die SIM Karte nicht fest eingebaut ist, sondern über Bluetooth angesprochen wird (Abb. 7.23).

Durch die Verwendung von APDUs können nicht nur die Dateien auf der SIM Karte gelesen und geschrieben werden, sondern es kann auch der Authentifizierungsalgorithmus der SIM-Karte angesprochen werden, der zu einer Zufallszahl (RAND) eine Signed Response (SRES) erzeugt (vgl. Abschn. 1.6.4). Außerdem kann auch das SIM Application Toolkit Protokoll über die Bluetooth Verbindung genutzt werden. Auch diese Nachrichten werden, wie ebenfalls in Abschn. 1.10 gezeigt, in APDUs verpackt.

7.6.4 High Quality Audio Streaming

Sowohl das Handset- als auch das Handsfree Profil wurden ursprünglich entwickelt, um Sprache in Telefonqualität und in mono zu übertragen. Für Hifi Audiostreaming reicht diese Qualität jedoch bei weitem nicht aus. Für diese Anwendung wurde deshalb das Advanced Audio Distribution Profil (A2DP) entwickelt, das Audiodaten mit Bandbreiten von 127–345 kbit/s überträgt. Da solche Datenraten nicht über SCO Verbindungen transportiert werden können, verwendet dieses Profil ACL Links zum Datentransport. Erste Versionen des Profils gibt es schon seit 2003, es dauerte jedoch einige Jahre, bis erste Geräte etwa 2006/07 auf den Markt kamen. Zu Geräten, die das A2DP Profil unterstützen sind Mobiltelefone mit eingebautem MP-3 Player und Kopfhörer. Kopfhörer unterstützen üblicherweise sowohl das A2DP und die Handsfree und Headset Profile. Mit einem eingebauten Mikrofon können diese dann sowohl für Musik als auch zum Telefonieren verwendet werden.

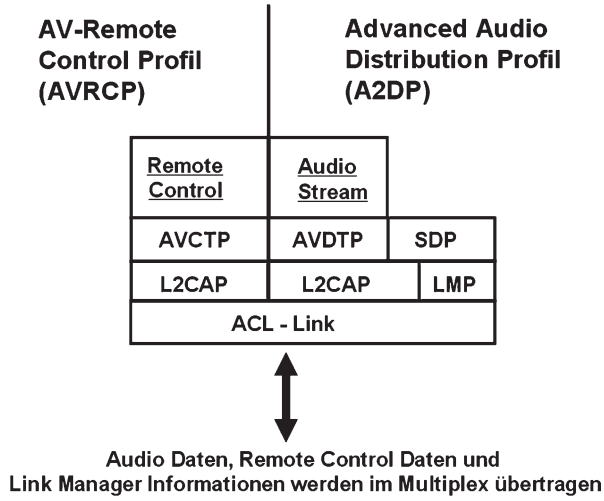

Abb. 7.24 Der A2DP Protokoll Stack inklusive Remote Control

Abb. 7.24 zeigt den A2DP Protokollstack. Das Profil basiert auf GAP und erlaubt somit anderen Endgeräten, die unterstützten Funktionalitäten in der SDP Datenbank abzufragen. Oberhalb des L2CAP Layer wurde das Audio Video Distribution Transfer Protocol (AVDTP) für die Datenübertragung spezifiziert. Wie der Name des Protokolls schon andeutet, kann es sowohl für die Übertragung von Audiodaten, als auch für die Übertragung von Videostreams verwendet werden. Das A2DP Profil verwendet das Protokoll jedoch lediglich für die Übertragung von Audiodaten. Neben der Übertragung des reinen Audiostreams werden auch Kontrollinformationen wie z. B. Codec Vereinbarungen und Austausch von Parametern wie der benötigten Bandbreite über das AVDTP Protokoll abgewickelt. Höhere Kontrollfunktionen wie z. B. das Springen zum nächsten Musikstück oder das Pausieren der Übertragung sind nicht Teil von AVDTP und werden über das Audio/Video Control Transport Protocol (AVCTP) übertragen, das nachfolgend beschrieben wird.

Der Bluetooth Standard erlaubt einem Endgerät, mehrere Verbindungen zu mehreren Geräten gleichzeitig geöffnet zu haben. Unterstützt ein Gerät dies, kann z. B. eine A2DP Verbindung zwischen einem Notebook und einem Kopfhörer aufgebaut sein, während gleichzeitig das Notebook Dateien, wie z. B. Bilder, mit einem weiteren Endgerät austauscht. Eine A2DP Übertragung benötigt jedoch für einen Audiostream in guter Qualität schon einen großen Teil der über Bluetooth möglichen Bandbreite, so dass die Dateiübertragung entsprechend langsam erscheint. Unterstützten alle Endgeräte im Piconet den Bluetooth 2.0+ EDR Standard, wird dies sicher weniger auffallen, da die Bandbreite dann etwa 2 Mbit/s beträgt. Dies ist deutlich mehr als bei Version 1.2 mit einem Limit von 723 kbit/s, von denen dann mit dem besten Audio Codec 345 kbit/s für die Audioübertragung verwendet werden.

Das A2DP Profil spezifiziert zwei Rollen für eine Verbindung. Die Audio Source Rolle wird von Geräten wie MP-3 Playern, Mobiltelefonen oder einem Mikrofon übernommen. Die andere Seite der Verbindung ist die Audio Senke (Audio Sink) Rolle, die üblicherweise von einem Headset oder einem Bluetooth Lautsprecherset übernommen wird.

Um mindestens einen gemeinsamen Audio Codec für eine A2DP Übertragung zwischen zwei Geräten zu gewährleisten, enthält die A2DP Spezifikation ein proprietäres Audioformat, das Sub-band Codec (SBC) genannt wird. Dieses muss von allen A2DP kompatiblen Endgeräten unterstützt werden und wird nachfolgend kurz beschrieben. Außerdem definiert der Standard die Übertragung anderer Codecs wie MPEG 1-2 Audio, MPEG-2,4, AAC und ATRAC über das Audio/Video Distribution Protocol (AVDTP). Diese Codecs sind optional. Der Standard erlaubt auch die Übertragung von weiteren Codecs über AVDTP. Um eine Interoperabilität zwischen Geräten unterschiedlicher Hersteller zu gewährleisten, muss ein Gerät jedoch immer in der Lage sein, einen Audiostream in SBC zu konvertieren, wenn ein anderes Gerät kein anderes optionales Format unterstützt.

Grundsätzlich ist der SBC Codec wie folgt aufgebaut: Als Eingangssignal erwartet der SBC Codec ein PCM kodiertes Audiosignal mit einer Abtastfrequenz von 44,1 oder 48 kHz. Der Codec teilt im ersten Schritt dann das Frequenzband des Eingangssignals in mehrere Teilbereiche auf, die auch als Unterbänder (Sub-Bands) bezeichnet werden. Der Standard rät, das Signal entweder in vier oder in acht Unterbänder aufzuteilen. Danach wird ein Skalierungsfaktor für jeden Unterkanal berechnet, der die Lautstärke des Signals in diesem Unterband beschreibt. Die Skalierungsfaktoren werden dann miteinander verglichen, um dem Unterband mit der meisten Signalinformation auch die meisten Bits für die Kodierung zuzuordnen. Der Standard schlägt vor, mindestens 19 Bit für eine mittlere Audioqualität und einen Monokanal zu verwenden und bis zu 55 Bits für Stereokanäle mit hoher Qualität. Nach der Kodierung der Audioinformation der Unterkanäle werden die Datenströme komprimiert. Der Kompressionsfaktor ist variabel, und es besteht somit hier nochmals die Möglichkeit, eine Abwägung zwischen Datenrate und Audioqualität zu treffen. Wird für die Kompression der niedrigste Faktor verwendet und Stereokanäle mit höchster Qualität kodiert, erzeugt dies einen Datenstrom mit einer Geschwindigkeit von 345 kbit/s.

Um Benutzeranweisungen vom Audio Sink Device (z. B. einem Kopfhörer) wie Lautstärkeregelung, nächster Track, Pause, etc., zurück zum Audio Source Gerät (z. B. einem MP-3 Player) zu übertragen, wird das Audio/Video Remote Control Profile verwendet. Wie in Abb. 7.25 gezeigt, wird dazu das Audio/Video Control Transport Protocol verwendet. Auch diese Nachrichten sind standardisiert um sicherzustellen, dass Endgeräte verschiedener Hersteller zusammenarbeiten. Das Profil unterscheidet Controller und Target (Ziel) Geräte und gruppiert diese in folgende Kategorien:

- Kategorie 1: Abspiel- und Aufnahmegeräte
- Kategorie 2: Monitor/Verstärker
- Kategorie 3: Audio/Video Empfänger (z. B. Radio)
- Kategorie 4: Menu

Des Weiteren definiert der Standard eine Vielzahl von Kontrollkommandos (Operation IDs) und legt fest, welche Geräte in welchen Gerätekategorien diese Kommandos jeweils unterstützen müssen und welche optional sind. Standardisierte Kontrollkommandos sind z. B.: ‚select', ‚up', ‚right', ‚root menu', ‚setup menu', ‚channel up', ‚channel down', ‚volume up', ‚volume down', ‚play', ‚stop', ‚pause', ‚eject', ‚forward' und ‚backward'.

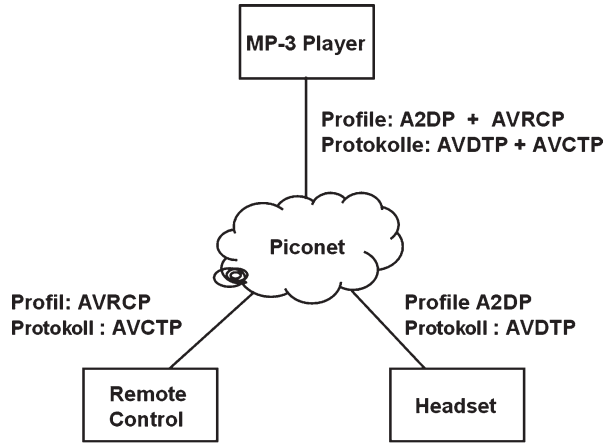

Abb. 7.25 Gleichzeitige Übertragung eines Audio Streams und Kontrollkommandos zwischen verschiedenen Geräten

Endgerätehersteller können auch selber Kommandos definieren, diese können jedoch nicht zwischen Geräten unterschiedlicher Hersteller verwendet werden.

Zwischen der Audio Streaming Session mit dem A2DP Profil und der Kontrollsession mit dem Remote Control Profil gibt es keine direkte Verbindung. Somit ist es möglich, in einem Piconet einen MP-3 Player für die Übertragung von Musik zu einem Kopfhörer zu verwenden, währen Lautstärkekommandos und andere Anweisungen von einem dritten Gerät, wie z. B. einer Fernbedienung, an den MP-3 Player gesendet werden können. Dieses Szenario ist in Abb. 7.25 dargestellt.

7.6.5 Das Human Interface Device (HID) Profile

Eine Anwendung, die in den letzten Jahren an Bedeutung gewonnen hat, ist die Anbindung von Eingabegeräten wie Tastaturen und Mäusen an Notebooks und Tablets. Im Notebookbereich verwenden drahtlose Mäuse oft ein proprietäres Protokoll und einen proprietären USB Empfänger. Dies ist bei Tablets nicht möglich, da kein proprietärer USB Empfänger an ein solches Endgerät angeschlossen werden kann. Da Tablets aber üblicherweise mit Bluetooth ausgestattet sind, können solche Eingabegeräte über das Human Interface Device (HID) Profil verwendet werden.

Das HID-Protokoll verwendet zwei L2CAP Verbindungen. Die erste Verbindung wird als ein Kontrollkanal verwendet und Daten werden synchron übertragen, d. h. auf jede Anfrage folgt immer eine Antwort, bevor die nächste Anfrage gesendet wird. Die zweite L2CAP Verbindung wird für einen HID Interrupt Kanal verwendet, über den asynchrone Informationen ausgetauscht werden, z. B. wenn der Anwender eine Taste drückt oder diese wieder loslässt. Da HID ein generisches Profil ist, werden Informationen in der SDP Datenbank verwendet, um herauszufinden, welche Ein- und Ausgabenachrichten ein Gerät unterstützt. Eingabenachrichten können z. B. Tastaturbenachrichtigungen oder Mausbewegungen sein. Ausgabenachrichten werden in umgekehrter Richtung gesendet, z. B. an Force Feedback Joysticks.

Da HID Geräte üblicherweise batteriebetrieben sind, spielt der Stromverbrauch eine große Rolle. Auf der Bluetooth Seite aktivieren deshalb der Host und das HID Device den Bluetooth Sniff Modus, nachdem die L2CAP Verbindung und der Interrupt Kanal aufgebaut wurden. Eine typische Sniff Rate in der Praxis sind 40 Millisekunden. Zusätzlich kann Snriff Subrating verwendet werden, um den Stromverbrauch in Nutzungspausen der Tastatur und der Maus weiter zu senken. Abb. 7.26 zeigt Ausschnitte einer HID Input Nachricht, die von

```
Frame 176: 19 bytes on wire (152 bits), 19 bytes captured (152 bits)
    Encapsulation type: Bluetooth H4 with linux header (99)
    [...]
    [Protocols in frame: hci_h4:bthci_acl:btl2cap:bthid]
    Point-to-Point Direction: Received (1)
Bluetooth HCI H4
    [Direction: Rcvd (0x01)]
    HCI Packet Type: ACL Data (0x02)
Bluetooth HCI ACL Packet
    .... 0000 0010 0011 = Connection Handle: 0x0023
    ..10 .... .... .... = PB Flag: First Automatically Flushable Packet (2)
    00.. .... .... .... = BC Flag: Point-To-Point (0)
    Data Total Length: 14
Bluetooth L2CAP Protocol
    Length: 10
    CID: Dynamically Allocated Channel (0x0041)
    [PSM: HID-Interrupt (0x0013)]
Bluetooth HID Profile
    1010 .... = Transaction Type: DATA (0x0a)
    .... 00.. = Parameter reserved: 0x00
    .... ..01 = Report Type: Input (0x01)
    Protocol Code: Keyboard (0x01)
    0... .... = Modifier: RIGHT GUI: False
    .0.. .... = Modifier: RIGHT ALT: False
    ..0. .... = Modifier: RIGHT SHIFT: False
    [...]
    Reserved: 0x00
    Keycode 1: a (0x04)
    Keycode 2: <ACTION KEY UP> (0x00)
    [...]
0000  02 23 20 0e 00 0a 00 41 00 a1 01 00 00 04 00 00
0010  00 00 00
```

Abb. 7.26 HID Input Nachricht einer Tastatur

einer Tastatur zu einem Notebook gesendet wurde. Hier ist zu sehen, dass die Nachrichtenlänge nur 19 Bytes beträgt, trotz Nutzung der ACL, L2CAP und HID Protokolle, die ineinander verschachtelt sind. Außerdem kann man in der Nachricht sehen, dass PSM 13 verwendet wurde, um den HID Interruptkanal aufzubauen. Die Nutzdaten in der Nachricht beschränken sich auf nur ein Byte (0x04h), welches das kleine 'a' repräsentiert, auf das der Nutzer auf der Tastatur gedrückt hat.

7.7 Fragen und Aufgaben

1. Welche maximale Geschwindigkeit bietet Bluetooth und von welchen Faktoren hängt diese ab?
2. Was bedeutet der Begriff Frequency Hopping Spread Spectrum (FHSS) und welche erweiterten Möglichkeiten bietet der Bluetooth 1.2 Standard?
3. Was ist der Unterschied zwischen Inquiry und Paging?
4. Welche Stromsparmodi gibt es bei Bluetooth?
5. Welche Aufgaben hat der Link Manager?
6. Wie können über das L2CAP Protokoll unterschiedliche Datenströme für unterschiedliche Anwendungen im Zeitmultiplex übertragen werden?
7. Welche Aufgaben hat die Service Discovery Datenbank?
8. Wie können mehrere Dienste gleichzeitig die RFCOMM Schicht verwenden?
9. Was ist der Unterschied zwischen der Bluetooth Authentifizierung und Autorisierung?
10. Warum gibt es eine Vielzahl unterschiedlicher Bluetooth Profile?
11. Welche Profile gibt es für die einfache und schnelle Übertragung von Dateien und Objekten zwischen zwei Bluetooth Endgeräten?
12. Wie unterscheidet sich das Hands-Free Profil vom SIM-Access Profil?

Lösungen sind auf der Website zum Buch unter http://www.cm-networks.de zu finden.