helion.pl

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

Bluetooth



Autorzy: Brent A. Miller, Chatschik Bisdikian, Ph. D.
Tłumaczenie: Marek Pałczyński
ISBN: 83-7197-968-1
Tytuł oryginału: Bluetooth Revealed, 2E
Format: B5, stron: 319

W największym skrócie Bluetooth można określić jako sposób na pozbycie się kabli i przewodów. Standard Bluetooth pozwala na bezprzewodową łączność i wymianę danych między urządzeniami mobilnymi (telefony komórkowe, notebooki, palmtopy, itp.) oraz stacjonarnymi (PC-ty, drukarki). Aktualizacja kalendarza czy spisu telefonów w palmtopie, łączenie cyfrowego dyktafonu z PC-tem, transmisja plików z aparatu cyfrowego, które następnie prześlesz „komórką" do znajomego, to tylko część zastosowań tej nowatorskiej technologii.

Książka prezentuje dogłębny i szczegółowy opis standardu Bluetooth. Jej szczególną zaletą jest fakt, że autorami są osoby, które bezpośrednio przyczyniły się do opracowania nowej (oznaczonej numerem 1.1) wersji tego standardu. Możliwości Bluetooth 1.1 w porównaniu z wersją poprzednią zostały znacznie rozszerzone, zwłaszcza w zakresie współpracy z innymi systemami. Dzięki tym rozszerzeniom, obszar jego zastosowań znacząco się powiększył.

"*Przeczytawszy po raz pierwszy książkę Brenta i Chatschika byłem pod wrażeniem faktu, że udało im się przedstawić ten skomplikowany temat w tak przejrzysty, zwięzły i dowcipny sposób. Przekształcenie tysięcy stron specyfikacji w kilkaset stron zrozumiałego tekstu nie jest wcale prostym zadaniem. Im jednak udało się to znakomicie.*"

Tom Siep
General Manager,
Bluetooth SIG, Inc.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

O Autorach **9**

Przedmowa do wydania pierwszego **11**

Wstęp **13**

Część I Wprowadzenie do komunikacji bezprzewodowej Bluetooth **15**

Rozdział 1. Czym jest Bluetooth? **17**
Bluetooth Special Interest Group 18
Historia i źródłosłów terminu Bluetooth 22
Przewodnik po książce 23

Rozdział 2. Podstawy technologii **27**
Komunikacja przewodowa i bezprzewodowa 27
Komunikacja bezprzewodowa na częstotliwościach radiowych 28
Bezprzewodowa komunikacja w podczerwieni 31
Rozwiązania radiowe komunikacji Bluetooth 32

Rozdział 3. Modele zastosowań Bluetooth **39**
Komputer bez połączeń kablowych 40
Doskonały zestaw słuchawkowy 41
Telefon — trzy w jednym 42
Interaktywne konferencje (transfer plików) 43
Most internetowy 44
Laptop głośnomówiący 46
Automatyczna synchronizacja 47
Błyskawiczne widokówki 48
Sieci ad hoc 49
Przetwarzanie niejawne 50
Inne scenariusze wykorzystania technologii 51

Rozdział 4. Wprowadzenie do specyfikacji Bluetooth **53**
Przeznaczenie specyfikacji 54
Zakres 55
Struktura specyfikacji 56
Zależności 58
Przewodnik po specyfikacji 59

Część II Specyfikacja Bluetooth 63

Rozdział 5. Stos protokołów Bluetooth 65
Elementy stosu protokołów 65
Grupa protokołów transportowych 67
Grupa protokołów pośredniczących 71
Grupa aplikacji 76

Rozdział 6. Niższe warstwy grupy protokołów transportowych 79
Warstwy protokołów i terminologia 81
Interfejs radiowy Bluetooth 82
Kontroler połączenia i pasmo podstawowe 85
Menedżer połączenia i protokół menedżera połączenia 110

Rozdział 7. Wyższe warstwy grupy protokołów transportowych 123
Warstwa L2CAP 124
Interfejs kontrolera hosta (HCI) 136

Rozdział 8. Protokoły pośredniczące — SDP i RFCOMM 147
Protokół RFCOMM 148
Protokół wyszukiwania usług (SDP) 154

Rozdział 9. Protokoły współpracy z IrDA 165
Protokoły współpracy z IrDA 167

Rozdział 10. Sterowanie połączeniami audio i telefonią 173
Procedury sterowania połączeniami audio i telefonią 174

Część III Profile zastosowań Bluetooth 183

Rozdział 11. Profile Bluetooth 185
Profile wersji 1.x 185

Rozdział 12. Profile ogólne 189
Zależności 189
Profil dostępu podstawowego (GAP) 190
Profil aplikacji wyszukiwania usług (SDAP) 196

Rozdział 13. Profil telefonii 201
Zależności 201
Profil telefonii bezprzewodowej 203
Profil interkomu 208
Profil zestawu słuchawkowego 210

Rozdział 14. Profil szeregowy i profil wymiany obiektów 213
Zależności 214
Profil portu szeregowego 214
Profil ogólnej wymiany obiektów 216
Profil wypychania obiektów 218
Profil transferu plików 222
Profil synchronizacji danych 225

Rozdział 15. Profile pracy sieciowej 231
Zależności 232
Profil pracy sieciowej z połączeniem komutowanym 232
Profil dostępu do LAN 236
Porównanie LAP i DUNP 241
Profil faksu 243

Część IV Całościowe spojrzenie na technologię Bluetooth 247

Rozdział 16. Porównanie z podobnymi technologiami 249
Rodzaje sieci 249
Inne technologie bezprzewodowe 251
Porównanie IrDA i bezprzewodowej komunikacji Bluetooth 252
Porównanie HomeRF i bezprzewodowej komunikacji Bluetooth 254
Podsumowanie technologii WPAN 256
Bezprzewodowe sieci LAN 256
IEEE 802.15 a bezprzewodowa komunikacja Bluetooth 258

Rozdział 17. Program kwalifikacyjny 263
Czym jest program kwalifikacyjny? 263
Czemu służy program kwalifikacyjny? 264
Program kwalifikacyjny Bluetooth 266

Część V Przyszłość technologii Bluetooth 273

Rozdział 18. Po opublikowaniu specyfikacji 1.x 275
Zmiany w SIG 276
Przyszłe aplikacje 284
Produkty Bluetooth 288

Rozdział 19. Wnioski 293
Rynek technologii Bluetooth 293
Szanse 294

Dodatki 297

Bibliografia — cytowane źródła 299

Lista znaków towarowych 303

Skorowidz 305

Rozdział 5.

Stos protokołów Bluetooth

Zasadnicza część specyfikacji Bluetooth koncentruje się na stosie protokołów. Stos ten pozwala urządzeniom na wzajemne lokalizowanie i łączenie się, a także umożliwia wymianę danych z każdym z nich oraz uruchamianie wspólnych, interaktywnych aplikacji. W niniejszym rozdziale zostaną zaprezentowane główne elementy stosu protokołów Bluetooth, ze szczególnym uwzględnieniem zależności występujących pomiędzy różnymi warstwami. Poszczególne protokoły szczegółowo przedstawiono w następnych rozdziałach.

Elementy stosu protokołów

Najogólniejsze kryterium podziału stosu protokołów Bluetooth ilustruje rysunek 5.1. Poszczególne elementy stosu (protokoły, warstwy, aplikacje, itp.) podzielono na trzy grupy logiczne:

- grupę *protokołów transportowych* (*transport protocol*),
- grupę *protokołów pośredniczących* (*middleware protocol*),
- grupę *aplikacji* (*application*).



Rysunek 5.1.
Ogólny podział stosu protokołów Bluetooth

Grupa protokołów transportowych — Grupa ta składa się z protokołów zaprojektowanych z myślą o umożliwieniu urządzeniom Bluetooth wzajemnej lokalizacji oraz tworzenia, konfigurowania i zarządzania zarówno fizycznymi, jak i logicznymi połączeniami, dzięki którym wyższe warstwy protokołów i aplikacje mogą przy użyciu protokołów transportowych przesyłać dane. W skład protokołów grupy wchodzą protokoły radiowe, pasma podstawowego, menedżera połączenia, połączenia logicznego i adaptacji oraz interfejsu kontrolera hosta (HCI).[1]

Grupa protokołów pośredniczących — Działanie aktualnych i przyszłych aplikacji opartych na połączeniach Bluetooth wymaga dodatkowych protokołów transportowych wchodzących w skład tej właśnie grupy. Grupa protokołów pośredniczących zawiera zarówno protokoły autorstwa firm zewnętrznych oraz standardy przemysłowe, jak i protokoły opracowane przez SIG specjalnie z myślą o bezprzewodowej komunikacji Bluetooth. Do tych pierwszych należy zaliczyć protokoły związane z Internetem (PPP, IP, TCP, itp.), protokoły aplikacji bezprzewodowych, protokoły przesyłania obiektów adaptowane ze standardu IrDA, itp. Do drugiego wspomnianego zestawu należą trzy protokoły zaprojektowane specjalnie dla komunikacji Bluetooth i pozwalające wielu rodzajom innych aplikacji na korzystanie z połączeń Bluetooth. Protokół emulacji portu szeregowego, zwany RFCOMM, umożliwia aplikacjom, które normalnie korzystałyby z portu szeregowego, na bezproblemowe komunikowanie się poprzez połączenia Bluetooth. Protokół sygnalizacyjny telefonicznych połączeń pakietowych umożliwia sterowanie zaawansowanymi funkcjami telefonicznymi, takimi jak zarządzanie grupami czy wsparcie dla bezprzewodowych słuchawek telefonicznych i stacji bazowych. W końcu, protokół wyszukiwania usług (SDP) pozwala urządzeniom rozpoznawać pełnione funkcje i pozyskiwać informacje na temat sposobów korzystania z nich.

Grupa aplikacji — Grupa składa się z rzeczywistych aplikacji, które korzystają z połączeń Bluetooth. Mogą to być klasyczne aplikacje, które nie były budowane pod kątem transmisji Bluetooth, takie jak aplikacja zestawiania połączenia modemowego czy przeglądarki internetowej. Z drugiej jednak strony mogą być również aplikacjami przeznaczonymi do komunikowania się w technologii Bluetooth — na przykład aplikacja zarządzająca urządzeniami telefonicznymi przy wykorzystaniu protokołu obsługi telefonii.

Protokoły grupy transportowej szczegółowo przedstawiono w rozdziałach 6. i 7. Rozdziały od 8. do 10. omawiają protokoły warstwy pośredniczącej, które zostały opracowane lub zaadaptowane przez SIG. Trzecia część książki została poświęcona różnorodnym aplikacjom determinowanym profilami zastosowań.

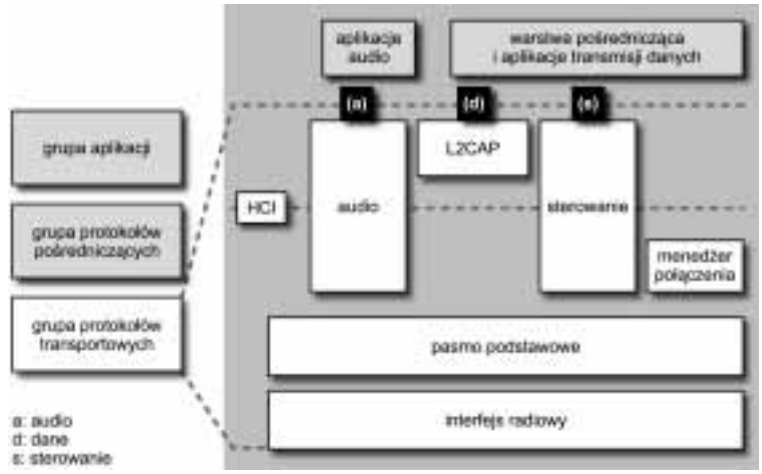
Zanim przejdziemy do szczegółowego omówienia protokołów i aplikacji, przybliżymy kluczowe protokoły grupy transportowej i pośredniczącej oraz zależności pomiędzy nimi.

[1] HCI nie jest w gruncie rzeczy protokołem. Niemniej specyfikacja HCI definiuje formaty pakietów przechodzących przez interfejs hosta oraz związki pomiędzy tymi pakietami. Jeżeli na przykład nadajemy pakiet A, spodziewamy się odbioru pakietu B. Formaty i zależności pomiędzy pakietami są zasadniczymi elementami specyfikacji protokołu. Z tego względu HCI zalicza się do protokołów transportowych.

Grupa protokołów transportowych

Organizacja protokołów w ramach grupy transportowej została przedstawiona na rysunku 5.2. Znajdują się tam protokoły opracowane przez SIG, których zdanie polega na przenoszeniu danych i sygnałów audio pomiędzy urządzeniami. Niniejszy rozdział prezentuje protokoły w odwrotnym porządku, czyli z punktu widzenia urządzenia nadawczego, które przekazuje ruch z wyższych warstw transportowych do niższych. Oznacza to, że w urządzeniach odbiorczych dane propagują w przeciwną stronę. Kolejne rozdziały będą prezentować protokoły z punktu widzenia urządzenia odbiorczego, co jest układem zgodnym z proponowanym przez specyfikację.

Rysunek 5.2.
Stos protokołów grupy transportowej



Protokoły transportowe obsługują zarówno asynchroniczne transmisje wymiany danych, jak i synchroniczne (okresowe) transmisje komunikacji głosowej o jakości telefonicznej (64 Kbps). W celu utrzymania wysokiego poziomu obsługi, wymaganego w aplikacjach audio, ruch audio traktowany jest priorytetowo. Dane audio omijają pośrednie warstwy protokołów i są przekazywane bezpośrednio z aplikacji audio do warstwy pasma podstawowego, gdzie w postaci małych pakietów transmitowane są poprzez interfejs radiowy Bluetooth.

Zanim przejdziemy do kolejnych zagadnień związanych z grupą protokołów transportowych, musimy zauważyć, że protokoły grupy transportowej nie należą do *warstwy transportowej* (czwartej warstwy) siedmiowarstwowego modelu OSI (Open System Interconnection)! W odniesieniu do tego modelu protokoły grupy transportowej znalazłyby się w *warstwie łącza danych* (warstwa druga) i *warstwie fizycznej* (warstwa pierwsza). Podsumowując — zbiór protokołów grupy transportowej tworzy wirtualny kanał, który jest używany do przesyłania danych z jednego urządzenia do drugiego za pomocą interfejsu radiowego Bluetooth. Protokoły te w gruncie rzeczy definiują mechanizmy transportu danych pomiędzy komunikującymi się urządzeniami. Do tego też nawiązuje nazwa grupy protokołów. Używa się także terminu „grupa protokołów transportowych Bluetooth". Celem jest odróżnienie protokołów transportowych opracowanych przez Bluetooth SIG od protokołu warstwy transportowej modelu OSI (warstwa czwarta). Należy zauważyć, że komunikacja pomiędzy urządzeniami Bluetooth

zawsze wymaga użycia wszystkich protokołów tej grupy, co nie ma miejsca w przypadku innych protokołów (spoza tej grupy), nawet tych, które zostały opracowane przez SIG (np. RFCOMM).

Warstwa L2CAP

Cały ruch aplikacji wymiany danych przekazywany jest w pierwszej kolejności poprzez warstwę *protokołu kontroli połączenia logicznego i adaptacji* (L2CAP — *logical link control and adaptation protocol*). Warstwa L2CAP uniezależnia protokoły wyższych warstw oraz aplikacje od szczegółowych rozwiązań protokołów transportowych niższej warstwy. Tym samym nie ma potrzeby informowania wyższych warstw o wykonywanych przeskokach częstotliwościowych interfejsu radiowego i pasma podstawowego czy konkretnych formatach pakietów stosowanych w transmisji poprzez interfejs radiowy Bluetooth. L2CAP pozwala na multipleksację protokołów, dzięki czemu interfejs radiowy może być współdzielony przez wiele protokołów i aplikacji. Prowadzi również segmentację pakietów — dzieli duże pakiety, które pochodzą z wyższych warstw, na ich mniejsze odpowiedniki, stosowane w transmisji w paśmie podstawowym. Analogiczny proces odwrotny — reasemblacja pakietów — zachodzi w urządzeniu odbiorczym. Co więcej, w trakcie połączenia dwóch równorzędnych urządzeń, warstwa L2CAP umożliwia utrzymanie odpowiedniej jakości usług poprzez negocjowanie i akceptowanie poziomu obsługi. W zależności od żądanego poziomu obsługi, warstwa L2CAP może sterować nowym ruchem (napływem danych nowych połączeń) i koordynować pracę niższych warstw tak, aby utrzymać wymagany poziom obsługi. Warstwa L2CAP i jej protokoły radiowe opisano szczegółowo w rozdziale 7.

Warstwa menedżera połączenia

Zadaniem znajdującego się w każdym urządzeniu menedżera połączenia jest negocjowanie, przy pomocy *protokołu menedżera połączenia* (LMP — *link manager protocol*), parametrów łącza radiowego pomiędzy urządzeniami Bluetooth. Parametry te określają szerokość pasma, którą należy zarezerwować, żeby zapewnić odpowiedni poziom obsługi dla wymiany danych (L2CAP) oraz okresową rezerwację pasma dla połączeń audio. Menedżer połączenia Bluetooth w urządzeniu komunikacyjnym stosuje uwierzytelnianie na zasadzie „wezwanie-odpowiedź". Nadzoruje *łączenie w pary* (tworzenie wiarygodnego połączenia pomiędzy urządzeniami poprzez generowanie i przechowywanie kluczy uwierzytelniających, które później wykorzystywane są do uwierzytelniania urządzeń) i szyfrowanie, w razie potrzeby, danych transmitowanych przez interfejs radiowy. W sytuacji, gdy uwierzytelnianie nie powiedzie się, menedżer połączenia może zerwać połączenie z danym urządzeniem i nie dopuścić do jakiejkolwiek dalszej komunikacji z nim. Menedżer połączenia umożliwia również sterowanie mocą poprzez negocjowanie przejścia do pracy w trybie małej aktywności pasma podstawowego (zagadnienie wprowadzone w rozdziale 2. i szczegółowo omówione w rozdziale 6.). Wymienia w tym celu stosowne informacje, jak na przykład czas pozostawania pasma podstawowego w trybie małej aktywności. Co więcej, menedżer połączenia może zażądać dostosowania poziomu mocy nadawczej w celu dalszego ograniczenia poboru mocy, zgodnie z tym, co zostało opisane w rozdziałach 2. i 6. Warstwa menedżera połączenia oraz jego protokół radiowy zostały szczegółowo scharakteryzowane w rozdziale 6.

Warstwa pasma podstawowego i warstwa radiowa.

Warstwa pasma podstawowego definiuje i urzeczywistnia interfejs radiowy Bluetooth. Stosując analogię do kabla połączeniowego, można by powiedzieć, że pasmo podstawowe determinuje „kształt" i „rozkład wyprowadzeń" przyłącza. Pasmo podstawowe definiuje proces wzajemnego wyszukiwania urządzeń oraz sposób zestawiania połączeń pomiędzy nimi. Narzuca urządzeniom funkcje master bądź slave (opisane w rozdziale 2.) — urządzenie, które inicjuje połączenie staje się urządzeniem typu master, natomiast drugie urządzenie staje się jednostką typu slave. Do zadań pasma podstawowego należy także definiowanie zasad doboru sekwencji przeskoków częstotliwościowych w komunikujących się urządzeniach. Określa reguły współdzielenia interfejsu radiowego przez kilka urządzeń. Bazuje pod tym względem na technice pakietowego połączenia *dupleksowego z podziałem czasowym* (TDD — *time division duplex*) oraz na schemacie przepytywania. Szczegółowo określa zasady współdzielenia interfejsu radiowego przez połączenia synchroniczne i asynchroniczne. W transmisji synchronicznej na przykład, urządzenie master okresowo transmituje dane lub przepytuje urządzenie slave. W zależności od tego, czy mamy do czynienia z połączeniami synchronicznymi, czy asynchronicznymi, pasmo podstawowe definiuje różne rodzaje pakietów oraz różne procedury ich przetwarzania, w tym detekcję i korekcję błędów, *wybielanie*[2] sygnału, szyfrowanie, transmisję i retransmisję pakietów. Warstwa pasma podstawowego i jego protokół radiowy zostaną szczegółowo omówione w rozdziale 6.

Należy zwrócić uwagę na fakt, że model master-slave nie obowiązuje w warstwach znajdujących się powyżej warstwy menedżera połączenia. W warstwie L2CAP i wyższych komunikacja odbywa się na zasadach określonych przez model peer-to-peer (równy-z-równym). Nie ma zatem rozróżnienia na pracę w trybie master czy slave.

Rozważania na temat radiowych transmisji pakietowych są bez bezprzedmiotowe bez zastosowania współdziałających ze sobą nadajników i odbiorników radiowych czy transceiverów[3]. Projekt interfejsu radiowego Bluetooth zawiera kilka parametrów, które czynią go optymalnym w pracy ze stosem protokołów Bluetooth w ramach bezprzewodowej komunikacji o małym zasięgu. Część radiowa rozdziału 6. omawia szczegóły radia Bluetooth.

Warstwa HCI

Interfejs radiowy, pasmo podstawowe i menedżer połączenia mogą być umieszczane razem w module Bluetooth. Moduł taki dołączany jest następnie do urządzenia macierzystego (*host*), dzięki czemu urządzenie to zyskuje możliwość bezprzewodowej komunikacji w standardzie Bluetooth. W przypadku takiej konfiguracji, host składa się z warstwy L2CAP i dowolnych, adekwatnych elementów wyższych warstw stosu protokołów. Moduł przyłączany jest do urządzenia macierzystego przy pomocy pewnego interfejsu fizycznego, zwanego *elementem transportowym hosta* (*host transport*). Przykładami mogą tu być: magistrala USB (*Universal Serial Bus*), port RS232 lub UART.

[2] Termin *wybielania* związany jest z kodowaniem kanałowym sygnału i zostanie szczegółowo opisany w rozdziale 6.

[3] Transceiver jest urządzeniem nadawczo-odbiorczym — *przypis tłumacza*.

Moduły Bluetooth opracowywane są przez różnych producentów. Jest to możliwe dzięki temu, że specyfikacja definiuje wspólny interfejs dostępu do znajdujących się w module niższych warstw stosu protokołów. Interfejs ten nie jest zależny od konkretnego fizycznego interfejsu, który łączy hosta z modułem. *Interfejs kontrolera hosta* (HCI — *host controller interface*) jest pojedynczym, standardowym interfejsem, który pozwala wyższym warstwom stosu, w tym aplikacji, na dostęp do pasma podstawowego, menedżera połączenia i innych rejestrów sprzętowych. Komendy HCI służą wprowadzaniu modułu w określony tryb pracy, w którym realizowane mogą być na przykład zadania uwierzytelniania czy przywoływania urządzeń. Zdarzenia HCI informują wyższe warstwy stosu o wynikach operacji wyszukiwania, o ustawieniach umieszczonego w paśmie podstawowym kodeka audio, o sile obieranego sygnału itp. Oczywiście przez warstwę HCI przechodzą zarówno połączenia synchroniczne. jak i asynchroniczne, zarówno w trakcie nadawania, jak i odbioru danych przez hosta.

Mimo, iż z reguły warstwa HCI znajduje się pod warstwą L2CAP, nie jest elementem wymaganym przez specyfikację. Jedynym powodem, dla którego została opracowana, jest chęć umożliwienia współdziałania różnych urządzeń macierzystych z modułami Bluetooth, niezależnie od producentów każdego z nich. Konkretny produkt, który implementuje technologię Bluetooth, nie musi spełniać założeń specyfikacji warstwy HCI by móc w pełni korzystać z interfejsu radiowego Bluetooth. Na przykład w silnie zintegrowanym systemie stanowiącym element większego systemu, HCI może w ogóle nie istnieć lub może znajdować się w innym miejscu w ramach stosu protokołów (np. nad warstwą L2CAP), może również mieć postać inną niż przewiduje to specyfikacja. HCI i różnego rodzaju protokoły transportu danych z hosta przedstawiono szczegółowo w rozdziale 7.

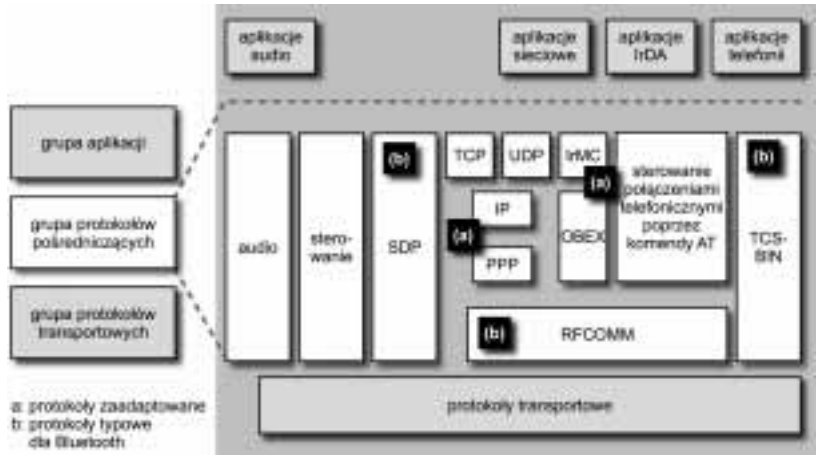
Pokazana na rysunku 5.2 *ścieżka sterowania* (*control path*) służy przekazywaniu informacji sterujących pomiędzy poszczególnymi warstwami. Przykładowo warstwa L2CAP powiadamia menedżera połączenia o jej wymaganiach dotyczących jakości usługi. Inny przypadek to sytuacja, kiedy aplikacja przekazuje żądanie użytkownika końcowego dotyczące obniżenia poboru mocy. Zazwyczaj, choć nie zawsze, możliwości sterowania udostępnione wyższym warstwom (w tym użytkownikowi końcowemu) służą wyborowi trybu pracy urządzenia. W trybie tym urządzenie pozostaje do czasu ponownej zmiany trybu pracy, której dokonano w wyższych warstwach stosu. Możliwe jest na przykład osobiste włączenie lub wyłączenie uwierzytelniania lub szyfrowania danych w połączeniu z konkretnym urządzeniem. Jednostka wyższej warstwy (np. aplikacja czy jej użytkownik) może przełączyć urządzenia na tryb pracy o obniżonym poborze mocy. Operacja taka zostanie odwzorowana na stosowny sygnał sterujący, zrozumiały dla menedżera połączenia, który z kolei będzie w stanie odpowiednio zareagować. Podobnie, możliwe jest przestawienie danego urządzenia na tryb pracy umożliwiający jego wyszukanie (w trybie tym urządzenie odpowiadałoby na zapytania przesyłane przez inne urządzenia) lub przełączenie na tryb pracy komunikacji tylko z zaufanymi urządzeniami (urządzenie odpowiada na żądania ustanowienia połączenia pochodzące tylko od określonych, znanych urządzeń, które i tak podlegają uwierzytelnianiu).

Ścieżka sterowania nie jest jednoznacznie opisana w specyfikacji, choć wzmianki na jej temat pojawiają się przy okazji różnych protokołów stosu. Tak czy inaczej, specyfikacja HCI opisuje znaczną część informacji, które są przez nią przenoszone. Przyjmujemy zatem taką samą zasadę prezentacji sygnałów sterujących przenoszonych za pośrednictwem HCI przez ścieżkę sterowania oraz taką samą zasadę wpływu tych sygnałów na różnego rodzaju protokoły i tryby pracy.

Grupa protokołów pośredniczących

Grupę protokołów pośredniczących przedstawiono na rysunku 5.3. Protokoły pośredniczące korzystają ze znajdujących się niżej protokołów transportowych oraz udostępniają warstwom aplikacji standardowy interfejs, który służy komunikowaniu się poprzez elementy transportowe. Każda z warstw pośredniczących stosuje standardowy protokół, który z punktu widzenia aplikacji umieszcza komunikację na wyższym poziomie abstrakcji niż miałoby to miejsce w przypadku bezpośredniego komunikowania się z niższą warstwą — warstwą protokołów transportowych. W skład protokołów pośredniczących wchodzą:

- *RFCOMM* — wirtualny port szeregowy,
- *protokół wyszukiwania usług* (SDP — *Service Discovery Protocol*) — wykorzystywany do opisu udostępnianych usług oraz lokalizowania poszukiwanych usług,
- zestaw protokołów współpracy z IrDA (*IrDA interoperability*) zaadaptowanych ze standardu IrDA i umożliwiających współdziałanie z aplikacjami standardu IrDA,
- *specyfikacja obsługi telefonii* (TCS — *Telephony Control Specification*) — wykorzystywana do obsługi połączeń telefonicznych, którymi mogą być zarówno połączenia audio jak i połączenia transmisji danych.



Rysunek 5.3. *Grupa protokołów pośredniczących*

Każdy protokół wraz z komunikacją audio Bluetooth (która nie jest protokołem jako takim, choć jest rozważana jako część stosu protokołów) został w tym rozdziale krótko scharakteryzowany. Szczegółowe informacje na temat protokołów zawierają dalsze rozdziały książki.

Warstwa RFCOMM

Najpowszechniejszymi obecnie interfejsami komunikacyjnymi, stosowanymi przez komputery i urządzenia telekomunikacyjne, są porty szeregowe. Większość technologii komunikacji szeregowej zakłada wykorzystanie kabli do przesyłania danych z portów szeregowych. Ponieważ celem bezprzewodowej komunikacji Bluetooth jest zastąpienie połączeń kablowych, funkcje obsługi komunikacji szeregowej i związanych z nią aplikacji stały się istotnym elementem pierwotnych modeli zastosowań, których celem było eliminowanie tego typu połączeń. W skład aplikacji, które najczęściej stosują komunikację szeregową (wraz z towarzyszącymi jej kablami), wchodzą: aplikacje transferu plików i obiektów w sieciach peer-to-peer, aplikacje synchronizacji danych czy aplikacje pracy sieciowej z połączeniami komutowanymi.

W celu umożliwienia prowadzenia komunikacji szeregowej w ramach połączeń Bluetooth, stos protokołów został wyposażony w warstwę symulującą działanie portu szeregowego, zwaną *RFCOMM*. Z punktu widzenia aplikacji, RFCOMM stanowi wirtualny port szeregowy. Pozwala zatem aplikacjom zaprojektowanym z myślą o kablowej transmisji szeregowej na łatwe przejście do sfery szeregowej komunikacji bezprzewodowej. Aplikacja korzysta z RFCOMM niemalże na takich samych zasadach, jak ze standardowego portu szeregowego, który przyłączono za pomocą kabla. Realizacja zadań takich jak synchronizacja, praca sieciowa z połączeniami komutowanymi itp. nie wymaga wprowadzania istotnych (jeżeli w ogóle wymaga jakichkolwiek) zmian w samej aplikacji.

Z założenia protokół RFCOMM pozwala starszym aplikacjom, które używają portu szeregowego, na korzystanie z technologii transportu Bluetooth.

RFCOMM jest standardem ETSI (European Telecommunications Standard Institute) TS 07.10. Definiuje on multipleksowaną transmisję szeregową w pojedynczym łączu szeregowym. Specyfikacja Bluetooth zaadaptowała część standardu ETSI TS 07.10, wprowadzając pewne modyfikacje, opracowane specjalnie dla komunikacji Bluetooth.

Ponieważ komunikacja szeregowa jest tak powszechna w urządzeniach cyfrowych, realizowana przez RFCOMM funkcja portu szeregowego staje się istotnym składnikiem stosu protokołów. Szczególne znaczenie ma fakt, że wersja 1.x[4] specyfikacji udostępnia tę własność starszym aplikacjom. Szczegóły dotyczące tematu warstwy RFCOMM można znaleźć w rozdziale 8.

Warstwa SDP

Tworzenie modelu zastosowań Bluetooth może być pod pewnymi względami rozpatrywane jako udostępnianie pewnego zbioru usług. Podstawowym celem budowania sieci urządzeń jest umożliwienie im wzajemnego komunikowania się, a tym samym wzajemnego udostępniania własnych usług. W tradycyjnych sieciach (jak na przykład sieć lokalna Ethernet) usługi (jak udostępnianie plików, udostępnianie drukarek, odwzorowywanie nazw, łączenie sieci czy rutowanie połączeń) udostępniane są przez pewien

[4] W trakcie pisania książki ostatnią dostępną wersją specyfikacji była wersja 1.1.

zbiór urządzeń, zwanych zazwyczaj „serwerami", innym urządzeniom sieciowym, zwanym zazwyczaj „klientami". W wielu przypadkach klient potrafi zlokalizować serwer dzięki statycznej konfiguracji, która zwykle jest tworzona i zarządzana przez administratora systemu, konfigurującego komputery klienckie (lub przekazującego niezbędne informacje konfiguracyjne użytkownikowi, który z kolei sam konfiguruje urządzenie).

W przypadku sieci dynamicznych, tworzonych ad hoc (tak jak sieci udostępniane przez technologię Bluetooth), taka statyczna konfiguracja jest niewystarczająca. Urządzenia Bluetooth (dwa lub większa ich ilość) mogą rozpoczynać komunikację pod wpływem chwilowego impulsu. Skoro więc zdolne są do korzystania z własnych usług, wymagają istnienia bardziej dynamicznych środków ich wyszukiwania. Pierwszą czynnością urządzenia komunikacyjnego po zestawieniu kanału komunikacyjnego mogłoby być zebranie informacji na temat usług dostępnych dla danego urządzenia. Do tego celu służy właśnie *protokół wyszukiwania usług* (SDP — *Service Discovery Protocol*). SDP definiuje standardowe metody, dzięki którym urządzenia Bluetooth rozpoznają i zapamiętują usługi, oferowane przez inne urządzenia. Analogicznie, SDP definiuje także zasady, na jakich urządzenia opisują swoje usługi podczas oferowania ich innym urządzeniom.

Z punktu widzenia użytkownika końcowego, wartość sieci dynamicznej zależy w dużej mierze od możliwości wyszukiwania usług. SDP został zaprojektowany i zoptymalizowany specjalnie pod kątem efektywnego realizowania tego zadania w ramach środowiska Bluetooth. SDP został szczegółowo opisany w rozdziale 8.

Protokoły współpracy z IrDA

Bezprzewodową komunikację w podczerwieni oraz jej podobieństwo do pewnych aspektów bezprzewodowej komunikacji Bluetooth przedstawiono w rozdziale 2. Protokoły wymiany i synchronizacji danych w środowisku bezprzewodowym zostały już wcześniej zdefiniowane przez Infrared Data Association (IrDA). SIG zdecydowała się więc zaadaptować niektóre z protokołów oraz modeli danych IrDA. Przyczyna tkwiła w tym, że obie technologie komunikacji bezprzewodowej — IrDA i Bluetooth — charakteryzują się tymi samymi zasadniczymi własnościami, modelami zastosowań i aplikacjami.

Najważniejszą sprawą, konieczną przy wymianie danych pomiędzy urządzeniami, jest zdefiniowanie ich formatu, zarówno pod względem semantycznym jak i syntaktycznym. Protokół *wymiany obiektów w podczerwieni* (*infrared object exchange* — IrOBEX lub OBEX) został opracowany przez IrDA jako sesyjny protokół komunikacji typu peer-to-peer. Jednym z zastosowań protokołu OBEX mogłaby być wymiana ściśle zdefiniowanych obiektów. OBEX pozwala na wymianę takich obiektów jak: elektroniczne wizytówki (format *vCard*), listy elektroniczne i komunikaty innego typu (format *vMessage*), wpisy terminarza (format *vCal*) itp. OBEX stanowi podstawowy element, na którym budowane są modele transferu plików (wymiany obiektów) i jednokierunkowego przesyłania obiektów (wypychania obiektów). Poza tym, innym protokołem autorstwa IrDA jest protokół *komunikacji ruchomej w podczerwieni* (IrMC — *infrared mobile communication*) — protokół pozwalający na synchronizowanie wyżej wymienionych obiektów.

Przeznaczeniem warstw protokołów współpracy z IrDA jest umożliwienie współdziałania technologii na poziomie aplikacji. Warstwy sieciowe zostały szerzej opisane w rozdziale 9.

Warstwy sieciowe

Bezprzewodowa komunikacja Bluetooth stosuje topologię sieci peer-to-peer (urządzeń równorzędnych), zamiast topologii z gatunku sieci LAN. Niemniej jednak technologia ta pozwala na pracę sieciową poprzez przyłączanie się do większych sieci za pomocą połączenia komutowanego lub sieciowego punktu dostępowego. Specyfikacja porusza także zagadnienia współpracy z *protokołem aplikacji bezprzewodowych* (WAP — *Wireless Application Protocol*), który stanowi specyfikację pracy w sieciach bezprzewodowych, znajdującą zastosowanie między innymi w telefonach komórkowych.

Zestawianie połączenia w sieciach komutowanych opiera się na wykorzystaniu warstwy komend AT, która wchodzi w skład protokołów warstw pośredniczących. W większości przypadków siecią docelową jest sieć używająca protokołu internetowego — IP (Internet Protocol). Po zestawieniu połączenia komutowanego z siecią IP, urządzenie (np. komputer przenośny czy palmtop), które zainicjowało połączenie sieciowe, może korzystać ze standardowych protokołów internetowych (np. TCP, UDP, HTTP).

Dane urządzenie może także połączyć się z siecią IP poprzez sieciowy punkt dostępowy, zgodnie z tym, co opisuje profil „dostępu do LAN przy użyciu protokołu PPP". W takim przypadku połączenie urządzenia z punktem dostępowym sieci realizowane jest za pomocą łącza Bluetooth. Sieciowy punkt dostępowy przyłącza je dalej do większej sieci (najczęściej, choć nie zawsze, przewodowej). Przyłączenie do punktu dostępowego poprzez łącze Bluetooth realizowane jest za pomocą internetowego protokołu PPP (*Point-to-Point*). Tak jak w przypadku pracy sieciowej z połączeniem komutowanym, po ustanowieniu połączenia PPP interakcja urządzenia z siecią odbywa się w oparciu o standardowe protokoły internetowe. Podobnie wygląda dostęp do sieci WAP poprzez bramę WAP. Ponownie ustanawiane jest połączenie PPP z punktem dostępowym sieci IP, który pozwala na dalszą interakcję z siecią przy wykorzystaniu standardowego protokołu WAP.

Brak w wersji 1.1 specyfikacji profili czy stosu protokołów obsługującego protokoły internetowe (jak TCP/IP) bezpośrednio w łączu Bluetooth nie ma żadnego znaczenia. Oznacza jedynie, że dostęp do sieci w wersji 1.1 specyfikacji wymaga użycia protokołu PPP, zgodnie z tym, co przed chwilą opisano. Mimo iż nie ma wątpliwości, że powinna istnieć możliwość bezpośredniego operowania stosem protokołów IP w ramach mechanizmów transportu wykorzystującego bezprzewodową komunikację Bluetooth, SIG nie zdefiniowała stosownych zasad postępowania (czyli profilu).[5] Przyszłe wersje specyfikacji będą jednak najprawdopodobniej uwzględniać problem bezpośredniego stosowania protokołów internetowych w ramach bezprzewodowej komunikacji Bluetooth.

[5] Niemniej jednak prace SIG obejmują problem stworzenia odpowiednich rozwiązań dla komunikacji IP w sieciach peer-to-peer. Odpowiada za to grupa robocza sieci osobistych. Zostało to szczegółowo opisane w części V książki.

Warstwa TCS i zagadnienia audio

Jak już wcześniej wspominano, jedną z kluczowych zalet bezprzewodowej komunikacji Bluetooth jest jej zdolność do przenoszenia zarówno danych, jak i informacji głosowych. Protokoły prezentowane do to tej pory znajdują zastosowanie głownie w transmisji danych. Tymczasem realizacja funkcji telefonicznych (w tym sterowanie połączeniem i zarządzanie grupami) należy do obowiązków warstwy *specyfikacji obsługi telefonii* (TCS — *Telephony Control Specification*). Operacje te są często związane z połączeniami głosowymi, w których zadanie TCS polega na ustalaniu parametrów połączenia. Po jego zestawieniu, powstały kanał audio jest w stanie przenieść informacje głosowe związane z prowadzoną rozmową. TCS wykorzystuje się także do ustanawiania połączeń transmisji danych, tak jak w profilu pracy sieciowej z połączeniem komutowanym. W tym przypadku treść połączenia przenoszona jest w postaci standardowych pakietów danych poprzez warstwę L2CAP.

Protokoły TCS są zgodne ze specyfikacją Międzynarodowej Unii Telekomunikacyjnej (ITU-T — International Telecommunications Union – Telecommunication) Q.931. Z uwagi na fakt, że używają kodowania binarnego, są określane przez specyfikację jako *TCS-BIN*. Podczas opracowywania specyfikacji, SIG zastanawiała się nad jeszcze jednym protokołem, zwanym *TCS-AT*. TCS-AT definiował protokół sterowania pracą modemu (nazywany komendami AT), który był przekazywany przez warstwę RFCOMM. Ostatecznie specyfikacja nie definiuje oddzielnego protokołu TCS-AT, mimo iż przesyłane przez RFCOMM komendy AT są faktycznie wykorzystywane przez niektóre aplikacje. TCS-BIN znajduje zastosowanie w niektórych profilach telefonii wersji 1.x. Jednakże nic nie stoi na przeszkodzie, aby aplikacje stosowały komendy AT przesyłane przez interfejs szeregowy RFCOMM (jak pokazano na rysunku 5.3), choć specyfikacja nie wymienia komend AT jako oddzielnego protokołu. Niektóre profile wersji 1.x stosują, zamiast protokołu TCS-BIN, właśnie komendy AT przesyłane przez RFCOMM. Są to między innymi profile: zestawu słuchawkowego, faksu, sieci z łączem komutowanym.

Technologia Bluetooth w sposób wyjątkowy traktuje sygnały audio, a szczególnie głos. Z uwagi na fakt, że połączenia audio mają charakter *izochroniczny*, co znaczy, że są z nimi związane pewne ograniczenia czasowe, ruch audio kierowany jest zazwyczaj bezpośrednio z warstwy pasma podstawowego i do niej. Nie przechodzi przez warstwy wyższe, jak L2CAP.[6] Typowy ruch audio przenoszony jest w formie pakietów o specjalnej strukturze, zwanych *pakietami połączeń synchronicznych* (SCO — *synchronous connection-oriented*). Technologia komunikacji Bluetooth pozwala na jednoczesne korzystanie z maksymalnie trzech kanałów audio (przy czym pozostaje pewna szerokość pasma na transmisję danych).

Komunikacja audio w standardzie Bluetooth zachodzi z prędkością 64 Kbps i stosuje dwa schematy kodowania: 8-bitową logarytmiczną *modulację impulsowo-kodową* (PCM — *pulse code modulation*) oraz *przyrostową modulację o zmiennym nachyleniu* (CVSD — *continous variable slope delta*). Kodowanie audio wykorzystuje kompresję według charakterystyki A lub μ.

[6] Pakietowe, cyfrowe sygnały audio mogą być przenoszone jako standardowe pakiety danych przez warstwę L2CAP. Taki przypadek jest jednak traktowany jak transmisja danych. Zakładamy, że wszystkie znajdujące się w książce odniesienia do głosu lub sygnałów akustycznych dotyczą ruchu audio przenoszonego bezpośrednio do łącza w paśmie podstawowym i formowanego do postaci pakietów SCO.

Z uwagi na fakt, że połączenia głosowe stanowią podstawową aplikację komunikacji audio (szczególnie w urządzeniach takich jak inteligentne telefony wyposażone w bezprzewodową komunikację Bluetooth), sygnał audio utożsamia się z głosem, choć oczywiście głos nie jest jedynym rodzajem sygnałów akustycznych, które mogą być przenoszone przez pasmo podstawowe Bluetooth. O ile kodowanie strumienia audio z przepływnością 64 Kbps jest wystarczające, strumień ten może być nadawany i odbierany za pośrednictwem łącz Bluetooth. Tym samym kanały audio Bluetooth oprócz głosu mogą przenosić sygnały akustyczne innego rodzaju, np. muzykę.[7]

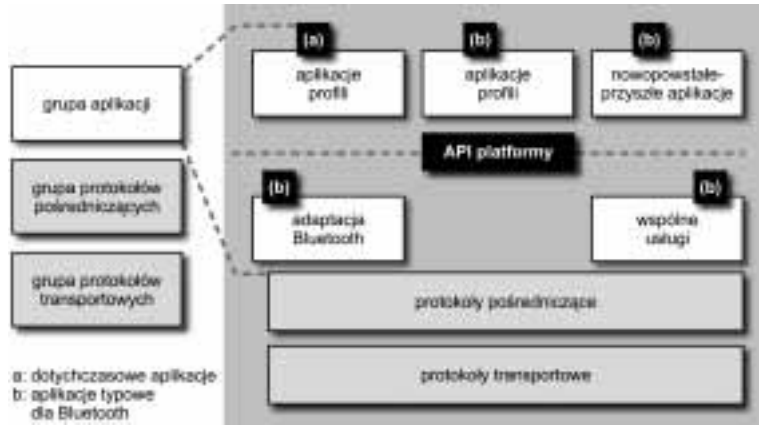
Jeśli pamiętamy o tym, że połączenia głosowe są zasadniczą częścią i kluczową zaletą bezprzewodowej komunikacji Bluetooth, możemy być trochę zdziwieni, że zagadnieniom audio poświęcono zaledwie kilka stron specyfikacji. Nie wynika to z faktu potraktowania problemu jako nieistotnego, ale z tego, że informacje audio są w większości przesyłane w specjalnych pakietach bezpośrednio do pasma podstawowego. Nie istnieje w związku z tym potrzeba zamieszczania szerokich opisów połączeń audio — format danych audio określono przy pomocy istniejących standardów. Z drugiej jednak strony zrozumienie struktury oraz zasad użycia pakietów SCO pasma podstawowego jest dość istotne, pozwala bowiem na pełniejsze zapoznanie się z komunikacją audio w ramach technologii Bluetooth. Zagadnienia audio szczegółowo porusza rozdział 10., choć pośrednio informacje na ten temat można znaleźć przy okazji omawiania pasma podstawowego w rozdziale 6.

Grupa aplikacji

Zaliczenie pewnych protokołów do grupy protokołów pośredniczących (szczególnie protokołów współpracy z IrDA, jak IrOBEX czy IrMC) mogłoby budzić sprzeciw osób chcących zaliczyć je do protokołów warstwy aplikacji. Nie to jednak mieliśmy na myśli, pisząc o grupie aplikacji. Grupa aplikacji, w znaczeniu przyjętym przez SIG, odnosi się do oprogramowania, które znajduje się powyżej stosu protokołów. Dotyczy to oprogramowania dostarczanego przez producentów urządzeń, niezależnych dostawców oprogramowania czy innych osób korzystających ze stosu protokołów w celu opracowania zestawu funkcji, które będą służyć użytkownikowi urządzenia Bluetooth. Niniejszą definicję oprogramowania aplikacyjnego ilustruje rysunek 5.4, w którym uwzględniono kilka możliwych konfiguracji oprogramowania warstwy aplikacji.

Największym zainteresowaniem w ramach grupy aplikacji cieszą się te aplikacje, które potrafią wykorzystać możliwości, dla jakich został stworzony Bluetooth. Wynika to z faktu, że dla stosu protokołów zawartego w urządzeniu Bluetooth, ktoś i tak musi napisać oprogramowanie, które zrealizowałoby funkcje takie jak zestawienia połączenia sieciowego, transfer plików, komunikacja z użyciem zestawu słuchawkowego itp. SIG poprzestała na zdefiniowaniu protokołów pośredniczących i transportowych. Nie zajmuje się definiowaniem protokołów aplikacyjnych jako takich, tak samo jak nie zajmuje się

[7] Technologia Bluetooth została zoptymalizowana pod kątem przenoszenia informacji głosowych. Nie projektowano jej z myślą o muzyce wysokiej jakości (np. jakości CD). Transmisja muzyki wymaga z reguły zastosowania pewnych technik kompresji.



Rysunek 5.4. *Ogólny wygląd grupy aplikacji z jej możliwymi konfiguracjami oraz warstwą adaptacji i warstwą typowych usług*

tworzeniem interfejsu programistycznego aplikacji (API — Application Programming Interface), choć urzeczywistnienie scenariuszy zastosowań przewidzianych w specyfikacji bezprzewodowej komunikacji Bluetooth wymaga zbudowania aplikacji. Wdrożenie wartościowego dla użytkownika końcowego profilu Bluetooth polega na dodaniu kodu aplikacji, który korzysta ze znajdującego się poniżej stosu protokołów. Chociaż poszczególne profile zastosowań określają zasady współpracy aplikacji w danym modelu, ich szczegóły implementacyjne nie są definiowane w specyfikacji. Programistom aplikacji pozostawiono dostatecznie dużo swobody, by ich produkty mogły konkurować ze sobą dzięki dodatkowym funkcjom lub interfejsom użytkownika bez naruszania zasad współdziałania, które określono w profilu.

Przykładem może być dowolne istniejące („starej daty") oprogramowanie, które zaprojektowano z myślą o korzystaniu z mechanizmów transportowych innych niż bezprzewodowa komunikacja Bluetooth i które może być nadal stosowane przy użyciu łącz Bluetooth. Z uwagi na fakt, że SIG zdefiniowała warstwy protokołów, których celem jest obsługa dotychczasowego oprogramowania (takie jak warstwy współpracy z IrDA czy RFCOMM), przed istniejącymi aplikacjami pojawiła się możliwość korzystania z łącz Bluetooth bez konieczności ich przebudowywania lub po dokonaniu niewielkich korekt. Do przykładów można zaliczyć istniejące aplikację wymiany i synchronizacji obiektów IrDA czy aplikację pracy sieciowej z połączeniem komutowanym. Obie korzystają z istniejących protokołów, które są obsługiwane przez środowiska Bluetooth, mimo iż zostały zaprojektowane do pracy przez port szeregowy. Dzięki temu, że stos protokołów emuluje port szeregowy (warstwa RFCOMM), w wielu istniejących aplikacjach powinno być możliwe dość łatwe odwzorowanie ich środowiska z IrDA czy komunikacji szeregowej na środowisko Bluetooth. Operacja nie powinna wymagać dużych (o ile w ogóle jakichkolwiek) zmian w aplikacji. Jednym ze sposobów urzeczywistnienia tej idei na różnych platformach może być opracowanie „cienkiej" warstwy *programowej adaptacji Bluetooth*, która odwzorowałaby istniejącą warstwę szeregową oraz inne rodzaje komunikacji na stos protokołów komunikacyjnych Bluetooth, zgodnie z tym, co zilustrowano na rysunku 5.4.

Inną możliwością jest zbudowanie nowych aplikacji, od początku zaprojektowanych z myślą o działaniu w środowisku Bluetooth. Podejście to jest uzasadnione w sytuacjach, kiedy nie istnieją aplikacje mogące urzeczywistnić dany model zastosowań Bluetooth lub gdy istnieje potrzeba wyposażenia aplikacji w funkcje, które bazują na unikatowych

właściwościach stosu protokołów. W przypadku, gdy przeznaczeniem aplikacji jest uzupełnienie danej platformy o bezprzewodową komunikację Bluetooth, dobrym rozwiązaniem może okazać się zbudowanie dla nich *wspólnych usług* (*common services*). W skład wspólnych usług wchodziłyby: usługi systemu zabezpieczeń, usługi zarządzania połączeniami, usługi SDP itp. Wspólne usługi aplikacyjne mogłyby być tworzone za pomocą kodu warstwy aplikacji, np. aplikacja zarządzania zabezpieczeniami (zagadnienie poruszane w [Muller99]), konsola sterowania Bluetooth (być może wraz z interfejsem użytkownika pozwalającym mu na wybranie urządzenia i usługi w pikosieci, z którymi chciałby się połączyć), wspólna aplikacja klienta i serwera SDP (także z interfejsem użytkownika do wyszukiwania i przeglądania usług). Reprezentacja wspólnych usług aplikacyjnych również znajduje się na rysunku 5.4.

Skoro specyfikacja nie definiuje API, jak budować standardowe aplikacje modeli zastosowań Bluetooth? Odpowiedzią są profile. Przypomnijmy sobie, że zadaniem profili jest określenie, w sposób ogólny, punktu wyjścia dla wykorzystania stosu protokołów w celu urzeczywistnienia danego modelu zastosowań. Ponieważ bezprzewodowa komunikacja Bluetooth będzie wykorzystywana najprawdopodobniej w niezliczonej ilości urządzeń i na różnych platformach, trudno byłoby (o ile jest to w ogóle możliwe) opracować pojedynczy standard API, który sprawdzałby się we wszystkich urządzeniach i na każdej z platform. Jednak dzięki temu, że SIG wybrała profile jako środek ustanawiania zasad współpracy pomiędzy różnymi urządzeniami, zapoznanie się z nimi dostarcza wskazówek odnośnie sposobów budowania aplikacji dla różnego rodzaju platform, istotnych dla technologii Bluetooth. Jak już wcześniej wspomniano, profile Bluetooth mogą być realizowane w oparciu zarówno o dotychczasowe aplikacje, jak i o te zaprojektowane specjalnie z myślą o komunikacji Bluetooth.

W sytuacji, gdy po wdrożeniu technologii na danej platformie zachodzi potrzeba opracowania nowego API, znawcy platformy okazują się lepszymi ekspertami niż znawcy technologii. Z tego względu SIG postanowiła nie opracowywać Linux API, Windows API, Symbian API, itd. Zamiast tego za pomocą profili zdefiniowano wszystkie funkcje potrzebne znawcom platformy do opracowania API, które współpracowałoby z aplikacjami Bluetooth na konkretnej platformie. Tym samym, mimo iż specyfikacja nie zawiera API, dostarcza programistom aplikacji niezbędnych wskazówek. Niektóre profile robią to w sposób bardziej, inne mniej bezpośredni. Na przykład profil aplikacji wyszukiwania usług opisuje potencjalne modele programowania aplikacji i definiuje prymitywy wyszukiwania usług, co może w prosty sposób prowadzić do przekształcenia go w API konkretnej platformy.

Tworzenie aplikacji nie jest ograniczone do oprogramowania, które urzeczywistniłoby dany profil. Podobnie jak w przypadku urządzeń, obszar zastosowań jest ogromny, być może nieograniczony. Wersja 1.x profili przedstawia jedynie podstawowy zbiór współdziałających ze sobą aplikacji. Prace nad nowymi profilami z pewnością będą kontynuowane, choć zakłada się, że nie będą to jedyne przypadki zastosowań, dla których powstawać będą nowe aplikacje. Postępowi inżynierowie bez wątpienia znajdą wiele nowych możliwości wykorzystania technologii Bluetooth i budowania nowych scenariuszy zastosowań. Z czasem, gdy oprogramowanie wdrażające istniejące profile na różnych platformach będzie się powiększać, zdobyte doświadczenie oraz powstałe API staną się punktem wyjścia do prac nad nowymi aplikacjami, a przed pracami tego typu otwierają się szerokie perspektywy.