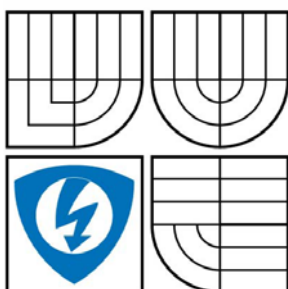


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

UNIVERZÁLNÍ MOBILNÍ KOMUNIKAČNÍ PLATFORMA PRACUJÍCÍ S TECHNOLOGIÍ BLUETOOTH

UNIVERSAL MOBILE COMMUNICATION PLATFORM USING BLUETOOTH TECHNOLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

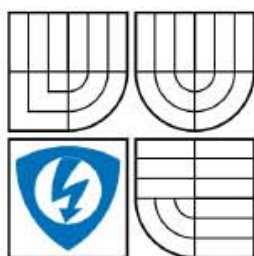
AUTOR PRÁCE
AUTHOR

Bc. RICHARD SOPKO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VÍT DANĚČEK

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Richard Sopko

ID: 83243

Ročník: 2

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Univerzální mobilní komunikační platforma pracující s technologií bluetooth

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku komunikačních technologií v mobilních zařízeních v osobních bezdrátových sítích (WPAN). Zaměřte se především na technologii Bluetooth a její možnou komunikaci s osobním počítačem a ostatními mobilními zařízeními. Dále nastudujte možnosti využití programovacího jazyku JAVA v implementaci s technologií Bluetooth.

Vytvořte návrh koncepce komunikační platformy. Daná platforma bude podporovat komunikaci dvou a více mobilních zařízení mezi sebou. Komunikační platforma bude umožňovat přenos textových zpráv a datových souborů mezi jednotlivými zařízeními.

Vytvořte základní zdrojový kód aplikace v programovacím jazyce JAVA, jenž bude realizovat vámi navrženou koncepci. Pro samotné spojení použijte technologii Bluetooth.

DOPORUČENÁ LITERATURA:

[1] White, J., Hemphill D. Java 2 Micro Edition, Manning Publications Co. Greenwich 2002. ISBN: 1-930110-33-2.

[2] Tkáč, J. Jak na Bluetooth v rekordním čase. Grada Praha 2005. ISBN: 80-247-1081-1.

[3] Eubanks, B. Java na maximum - 101 tipů, knihoven a praktických nápadů pro vaše projekty. Grada Praha 2006. ISBN: 80-251-1111-3.

Termín zadání: 9.2.2009

Termín odevzdání: 26.5.2009

Vedoucí práce: Ing. Vít Daněček

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Abstrakt

Diplomová práce je zaměřená na problematiku komunikačních technologií v mobilních zařízeních v osobních bezdrátových sítích typu WPAN. Práce se skládá ze tří základních částí. První část poskytuje přehled o osobních bezdrátových sítích WPAN s ohledem především na technologii Bluetooth a jejích možnostech komunikace mezi mobilními zařízeními. Druhá část poskytuje možnosti použití programovacího jazyka Java 2 Micro Edition při práci s technologií Bluetooth. Klíčovým bodem této práce je třetí část, která obsahuje návrh koncepce komunikační platformy. Tento návrh slouží jako podklad pro vytvoření aplikace, která je určena pro mobilní telefony. Vytvořená aplikace umožňuje komunikaci výměnou souborů, ale také písemnou konverzaci dvou i více lidí v reálném čase prostřednictvím Bluetooth spojení.

Klíčová slova

Bluetoot, J2ME, MIDlet, WPAN, CLCD, RMS, KVM,

Abstract

This master's thesis is focused on field of communication technologies in mobile devices in personal WPAN type wireless networks. Work consists of three basic parts. First part provides overview of personal WPAN wireless networks and is specialized on Bluetooth technologies and its opportunities of communication between mobile devices. Second part deals with an opportunity of using programming language Java 2 Micro Edition in work with Bluetooth technology. Key point of this work is third part which includes scheme of conception of a communication platform and creating of application designed for mobile phones. Created application enables communication by means of changing files and written conversation of two or more people in real time by Bluetooth connection.

Key words

Bluetoot, J2ME, MIDlet, WPAN, CLCD, RMS, KVM,

SOPKO, R. *Univerzální mobilní komunikační platforma pracující s technologií bluetooth*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 82 s. Vedoucí diplomové práce Ing. Vít Daněček.

Prohlášení o původnosti práce

Prohlašuji, že svojí diplomovou práci na téma „Univerzální mobilní komunikační platforma pracující s technologií bluetooth“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č.140/1961 Sb.

V Brně dne 21. 5. 2009

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Vítovi Daněčkovi, za velmi užitečnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 21.5.2009

.....

Obsah

| | |
|--|-----------|
| ÚVOD | 11 |
| 1. WPAN..... | 13 |
| 1.1 IEEE 802.15 | 13 |
| 2. BLUETOOTH..... | 15 |
| 2.1 Základný popis | 15 |
| 2.1.1 Prehľad štandardov | 16 |
| 2.2 Vrstvový model protokolov systému Bluetooth..... | 16 |
| 2.2.1 Rádiová vrstva (Bluetooth Radio)..... | 17 |
| 2.2.2 Špecifikácia systému Bluetooth v základnom pásme..... | 18 |
| 2.2.3 Ostatné protokoly a rozhranie | 20 |
| 2.3 Topológia Bluetooth..... | 22 |
| 2.4 Bluetooth Profily | 23 |
| 3. OSTATNÉ TECHNOLOGIE WPAN | 25 |
| 3.1 Ultra-WideBand | 25 |
| 3.1.1 MBOA | 26 |
| 3.1.2 DS-UWB | 30 |
| 3.2 ZigBee | 35 |
| 3.2.1 802.15.4 Fyzická vrstva a vrstva prístupu k médiu..... | 35 |
| 3.2.2 Rozšírenie 802.15.4 - ZigBee..... | 37 |
| 4. POROVNANIE TECHNOLOGIÍ WPAN | 40 |
| 5. JAVA 2 MICRO EDITION..... | 42 |
| 5.1 Architektúra J2ME | 42 |
| 5.1.1 Konfigurácia..... | 43 |
| 5.1.2 Virtuálne stroje | 43 |
| 5.1.3 Profily..... | 44 |
| 6. NÁVRH SYSTÉMU | 48 |
| 6.1 Návrh funkčnosti | 48 |
| 7. REALIZÁCIA UNIVERZÁLNEJ KOMUNIKAČNEJ PLATFORMY | 51 |
| 7.1 Aspekt tvorby aplikácie..... | 51 |
| 7.2 Funkčnosť platformy | 52 |

| | | |
|-----------|---|-----------|
| 7.3 | Spustenie aplikácie a úvodné menu..... | 54 |
| 7.3.1 | Komunikácia výmenou textových správ | 56 |
| 7.3.2 | Komunikácia prostredníctvom výmeny súborov..... | 63 |
| 8. | EXPERIMENTÁLNE TESTOVANIE..... | 72 |
| 9. | ZÁVER | 74 |
| | ZOZNAM POUŽITEJ LITERATÚRY | 75 |
| | ZOZNAM POUŽITÝCH SKRATIEK..... | 78 |
| | ZOZNAM PRÍLOH | 81 |
| A | OBSAH PRILOŽENÉHO DVD..... | 82 |

Zoznam obrázkov

| | |
|---|----|
| Obr. 2.1: Vrstvový model protokolov systému Bluetooth | 17 |
| Obr. 2.2: Výmena informácie medzi riadiacou a podriadenou jednotkou..... | 19 |
| Obr. 2.3: Topológia siete Bluetooth | 22 |
| Obr. 2.4: Hierarchia vybraných Bluetooth profilov | 24 |
| Obr. 3.1: Rozdelenie pásem (WiMedia Alliance, Júl 2007 [27]) | 26 |
| Obr. 3.2: Vysielanie signálu použitím troch pásiem | 27 |
| Obr. 3.3: Štruktúra superrámca kontroly prístupu k médiu MAC..... | 29 |
| Obr. 3.4: WiMedia spoločná UWB platforma..... | 30 |
| Obr. 3.5: Rozdelenie pásiem DS-UWB..... | 31 |
| Obr. 3.6: Superrámec DS-UWB..... | 33 |
| Obr. 3.7: Vrstvový model DS-UWB | 34 |
| Obr. 3.8: Topológia siete DS-UWB | 34 |
| Obr. 3.9: Modulácia a rozprestieracia činnosť v pásme 2,45 MHz..... | 36 |
| Obr. 3.10: Modulácia a rozprestieracia činnosť v pásme 868/915 MHz..... | 36 |
| Obr. 3.11: Architektúra 802.15.4..... | 37 |
| Obr. 3.12: Štruktúra superrámca s GTS | 37 |
| Obr. 3.13: Architektúra ZigBee..... | 38 |
| Obr. 3.14: Topológia siete ZigBee | 39 |
| Obr. 4.1: Graf závislosti prenosovej rýchlosti od vzdialenosti technológií WPAN..... | 40 |
| Obr. 5.1: Architektúra J2ME | 42 |
| Obr. 5.2: Architektúra MIDP..... | 46 |
| Obr. 5.3: Prechody medzi stavmi MIDletu..... | 46 |
| Obr. 6.1: Schéma zostavenia spojenia..... | 48 |
| Obr. 6.2: Spôsob vytvorenia klienta alebo serveru..... | 49 |
| Obr. 7.1: Spôsob realizácia aplikácie, kde serverová časť aplikácie poskytne službu klientskej, ktorá mu umožní pripojiť sa..... | 52 |
| Obr. 7.2: UML diagram tried..... | 54 |
| Obr. 7.3: GUI aplikácie: a - úvodné menu, b – nastavenie, c – hľadanie servera v okoly.. | 56 |
| Obr. 7.4: Diagram sekvencií pri vyhľadávaní služby na vzdialených zariadeniach | 59 |
| Obr. 7.5: Navrhnutý komunikačný paket | 60 |
| Obr. 7.6: GUI aplikácie: a – nájdený server v okolí, b – textová komunikácia, | 61 |
| Obr. 7.7: Diagram sekvencií pre zistenie súborového systému zariadenia | 64 |
| Obr. 7.8: GUI aplikácie: a – upozornenie o zvolení súboru, b – priebeh odosielania súboru, c – stav odoslaného súboru. | 66 |
| Obr. 7.9: GUI aplikácie: a – prichádzajúci súbor, b – priebeh sťahovania súboru, c – stav uloženia súboru..... | 70 |
| Obr. 8.1: Testovacie mobilné zariadenia zľava Sony Ericsson K550i, K608i a K530i | 72 |
| Obr. 8.2: Test prenosu textových správ medzi zľava SonyEricsson K608i, K530i a K550i. | 72 |
| Obr. 8.3: Test prenosu dátového súboru medzi (hore) K530i a (dole) K550i..... | 73 |

Zoznam tabuliek

| | |
|--|----|
| Tab. 2.1: Prehľad najdôležitejších verzií štandardov Bluetooth..... | 16 |
| Tab. 2.2: Protokoly a vrstvy vo vrstvovom modeli systému Bluetooth | 17 |
| Tab. 2.3: Výkonové triedy štandardu Bluetooth | 18 |
| Tab. 3.1: Podporované prenosové rýchlosti MultiBand OFDM | 28 |
| Tab. 3.2: Prenosové rýchlosti definované v dolnom pásme DS-UWB | 31 |
| Tab. 3.3: Čísla kanálov pikosietí, čipová rýchlosť a použitie rozprestieraného kódu..... | 32 |
| Tab. 4.1: Prehľad WPAN technológií | 41 |
| Tab. 7.1: Prehľad tried vytvorenej aplikácie. | 52 |

Úvod

Moderné komunikačné prostriedky sú súčasťou modernej spoločnosti. Zdokonaľujú medzilidské vzťahy a navzájom ich zlepšujú natoľko, že sa uzatvárajú do akéhosi kruhu, v ktorom majú ľudia potrebu medzi sebou komunikovať. Prístup k dátam a potreba výmeny informácií sa stali nevyhnutnou súčasťou každodenného života.

Všade prítomná potreba prístupu k dátam viedla k vzniku bezdrôtových technológií ako 802.11b a Wi-Max. Tieto technológie poskytujú pripojenia typu osobný počítač - osobný počítač, alebo osobný počítač - poskytovateľ internetu ISP (Internet Service Provider), a to na vzdialenosť od malého územia až po metropolitné územia. Množstvo aplikácií má oveľa menšie prenosové či komunikačné požiadavky na vzdialenosť, ako sú napríklad pripojenie periférií k mobilnému zariadeniu alebo súčasti osobného počítača. Osobné bezdrôtové siete WPAN (Wireless Personal Area Network) bezpečne vyhovujú týmto typom aplikácií. Ponúkajú dosah signálu v okolí jednotky až stovky metrov a širokú škálu prenosových rýchlostí. Dôležitým faktorom pri akceptovaní nových technológií sa stala spotreba energie, pretože stále viac aktuálne je vybavenie osobných bezdrôtových sietí WPAN rôznymi druhmi zariadení, ktoré sú ľahké a mobilné.

Medzi najpopulárnejších a najrozšírenejších technológií, v rámci komunikácie na krátku vzdialenosť, patrí štandard od skupiny zaoberajúcej sa štandardizáciou IEEE 802.15.1 a tou je Bluetooth. Veľkou výhodou štandardu Bluetooth je, že sa s ním stretne skoro v každom mobilnom zariadení.

Problematiku komunikácie, v mobilných zariadeniach v sieťach typu WPAN, uľahčuje podpora programovacieho jazyka Java. Presnejšie sa jedna o jeho upravenú verziu Java 2 Micro Edition (J2ME), ktorá je určená pre malé zariadenia, akým je i mobilný telefón. Ten musí pre chod Java aplikácii podporovať profil MIDP (Mobile Information Device Profile). Interakciou programovacieho jazyka Java a bezdrôtových štandardov u mobilných telefónov, ako sú Bluetooth, IrDA, WiFi, GPRS, HSCSD a EDGE, môžeme z mobilného telefónu spraviť podľa potrieb multifunkčné univerzálne zariadenie.

Cieľom tejto diplomovej práce je naštudovať problematiku komunikačných technológií u mobilných zariadeniach v osobných bezdrôtových sieťach (WPAN). Zamerať sa predovšetkým na technológiu Bluetooth a jej možnosť komunikácie medzi dvoma a viacerými mobilnými zariadeniami. Priblížiť možnosti využitia programovacieho jazyka Java v implementácii s technológiou Bluetooth.

Dôležitou súčasťou tejto práce je vytvoriť návrh koncepcie komunikačnej platformy. Navrhnutá komunikačná platforma má podporovať komunikáciu dvoch a viacerých mobilných zariadení medzi sebou a taktiež umožňovať prenos textových správ a dátových súborov medzi jednotlivými zariadeniami.

Ďalej je úlohou taktiež vytvoriť základný zdrojový kód aplikácie v programovacom jazyku Java, ktorý bude realizovať navrhnutú koncepciu. Pre samotné spojenie použiť technológiu Bluetooth.

1. WPAN

Osobná sieť za použitia bezdrôtového spojenie sa nazýva WPAN (Wireless Personal Area Network). Používa sa na komunikáciu a vzájomnú spoluprácu medzi zariadeniami ako sú mobilné telefóny, počítače, PDA, pagery, rôzne periférie a spotrebná elektronika. Dosah v WPAN je zvyčajne do 10m až 100m. WPAN majú typický menší dosah než WLAN (Wireless Local Area Network), ale majú tiež výrazne menšie nároky na napájanie, čo vyplýva z ich určenia. Štandardizované technológie spadajúce do kategórie WPAN zahŕňujú Bluetooth, ZigBee, Ultra-wide-band (UWB), IrDA atd.

Najznámejším štandardom kategórie WPAN je bez sporu štandard Bluetooth. Organizácia IEEE zahŕňa siete WPAN do kategórie IEEE 802.15, pričom Bluetooth je vedené pod označením IEEE 802.15.1. Avšak Bluetooth nie je zďaleka jediným štandardom tejto kategórie. Existujú aplikácie, pre ktoré nie je Bluetooth vhodný, a tak sa vyvíjajú ďalšie štandardy pre bezdrôtové siete aby sa navzájom dopĺňali a vylepšovali ako popisuje nasledujúca kapitola.

1.1 IEEE 802.15

IEEE 802.15 je pracovná skupina TG (Task Group) zaoberajúci sa štandardizáciou bezdrôtových osobných sietí WPAN. Do tejto kategórie patrí niekoľko dielčích skupín [15]:

- IEEE 802.15.1 - TG 1: WPAN/Bluetooth
Bluetooth prevzalo IEEE ako jednu z normalizačných variant WPAN s cieľom umožniť bezdrôtovú komunikáciu bez veľkých nárokov na konfiguráciu a spotrebu energie u malých príručných zariadení, ktorý vychádza zo špecifikácie Bluetooth verzie 1.1. Publikované 14. júna.2002.
- IEEE 802.15.2 – TG 2: Koexistencia
Skupina, ktorá sa zaoberala koexistenciou všetkých bezdrôtových technológií v bez licenčnom pásme vytvorených v rámci IEEE 802. V súčasnosti je práca pozastavená.
- IEEE 802.15.3 – TG 3: WPAN High Rate
Táto skupina vytvorila špecifické požiadavky na vrstve prístupu k médiu (MAC) s fyzickou vrstvou (PHY) pre rýchle bezdrôtové osobné siete WPAN, určené pre digitálnu techniku a multimediálne aplikácie.
- IEEE 802.15.3a – TG 3a: UWB
Skupina pracuje na ešte rýchlejšej verzii ako je 802.15.3 a to na báze technológii UWB (Ultra Wide band). IEEE definitívne rozpustila skupinu pracujúcej na príslušnej norme celé dva roky.
- IEEE 802.15.3b – TG 3b

Skupina, ktorá ma za úlohu vylepšiť IEEE 802.15.3 implementáciu a pridať univerzálnosť na vrstve prístupu k médiu (MAC) pri zachovaní spätnej kompatibility.

- IEEE 802.15.3c – TG 3c:– Millimeter Wave Alternative PHY
Skupina riešiaci rádiový prenos najmenej 1 Gbit/s v pásme 57 - 64GHz (pásmo milimetrových vln).
- IEEE 802.15.4 – TG 4: WPAN Low Rate
Skupina pre nízko rýchlostné osobné siete, ktoré vďaka minimálnej spotrebe energie sú používané mesiace a až roky na jednu batériu. Pre názov technológie sa používa ZigBee.
- IEEE 802.15.4a – TG 4a: Low Rate Alternative PHY
Skupina sa zaoberala možnosťami použitia rôznych fyzických vrstiev pre 802.15.4. Vytvorila štandard, ktorý podporuje na fyzickej vrstve Impulzné systémy UWB taktiež tzv. Chirp rozprostrenie spektra.
- IEEE 802.15.4b – TG 4b
Táto skupina má za úlohu opraviť niektoré chyby, zjednodušiť a zefektívniť štandard 802.15.4.
- IEEE 802.15.5 – TG 5: Mash networking
Skupina, ktorá sa snaží nájsť funkčnosť, ktorou je nutné dodať do fyzickej vrstvy prístup k médiu aby bolo možné vytvoriť slučkové (Mash) siete.
- IEEE 802.15.6 – TG 6: Body Area Network
Skupina rozvíja komunikačný štandard optimalizovaný pre zariadenia s nízkou spotrebou v okolí ľudského tela (ale nie je to limitované len na ľudí) aby slúžila rade aplikácií v lekárstve, spotrebnej elektronike, osobnej zábave atd.

2. Bluetooth

Bluetooth je špecifikácia bezdrôtovej komunikácie určená predovšetkým ako náhrada káblu pre komunikáciu na malú vzdialenosť medzi elektronickými zariadeniami. Medzi jej kľúčové vlastnosti patrí miniatúrnosť, nízka cena, robustné spojenie, krátky dosah, malý príkon, automatická konfigurácia, komunikácia bez priamej viditeľnosti a rozšírenosť.

Prvopočiatky technológie Bluetooth siahajú do roku 1994, keď dvaja švédski inžinieri, Sven Mattisson a Jaap Haartsen pracujúci pre Ericsson Mobile Platforms vyvinuli prvú špecifikáciu Bluetooth. Táto špecifikácia bola oficiálne uvedená neziskovou priemyselnou skupinou Bluetooth SIG (Special Interest Group), vo verzii 0.7, ktorá bola založená v roku 1998 Nokiou, Ericssonom, IBM, Intelom a Toshihou. V decembri 1999 bola vytvorená kompletná definícia štandardu 1.0. O rok neskôr bola doplnená radou zmien a boli odstránene nepresnosti. Vznikla tak špecifikácia Bluetooth 1.1 a táto špecifikácia je tiež zahrnutá v štandarde IEEE 802.15.1-2002 [7].

Technológia Bluetooth má za cieľ zjednotiť osobné komunikačné a výpočtové zariadenia. Meno technológie, pôvodne zvolené iba dočasne, nakoniec zostalo, a tak sa dostalo i do povedomia užívateľov. Medzi najvýznamnejších presadzovateľov Bluetooth dnes patria zakladatelia SIG spoločnosti 3Com, Agere, Microsoft a Motorola. Na trhu sa začal komerčne uplatňovať až v roku 2000 a dnes je najpoprednejším predstaviteľom WPAN sieti.

2.1 Základný popis

Systém Bluetooth pracuje v nelicencovanom frekvenčnom pásme ISM (Industrial, Scientific, Medical) na frekvencii 2,4 až 2,4835GHz. Pre nosné frekvencie platí vzťah 2.1.

$$f = 2402 + k [MHz], k = 0, 1, 2, \dots, 78 \quad (2.1)$$

V tomto frekvenčnom pásme je teda k dispozícii 79 rádiových kanálov. Šírka rádiového kanálu je 1 MHz. Aby boli dodržané limity mimo toto pásmo, sú definované tzv. ochranné pásma. Dolné ochranné pásmo má šírku 2 MHz, horné ochranné pásmo 3,5 MHz. Je definovaných niekoľko výkonových úrovní (Tab. 2.3), s ktorými je možné komunikovať do vzdialenosti 10 – 100m. Na maximálny dosah 100 m (záleží na výkonovej triede) môže komunikovať maximálne až 7 zariadení a nemusí byť medzi zariadeniami priama viditeľnosť. Jednotlivé zariadenia sú identifikované pomocou svojej adresy BT_ADDR (Bluetooth Device Address), podobné ako je MAC adresa u Ethernetu. Bluetooth umožňuje vytvoriť dátový spoj symetrický ale aj asymetrický, kde prenosová rýchlosť prijímaných informácií (downlink) je vyššia ako pri odosielaní (uplink) [3].

2.1.1 Prehľad štandardov

Nasledujúca Tab. 2.1 obsahuje kľúčové verzie Bluetooth špecifikácií. Každá novšia špecifikácia je spätne kompatibilná s predchádzajúcimi verziami [3][9].

Tab. 2.1: Prehľad najdôležitejších verzií štandardov Bluetooth

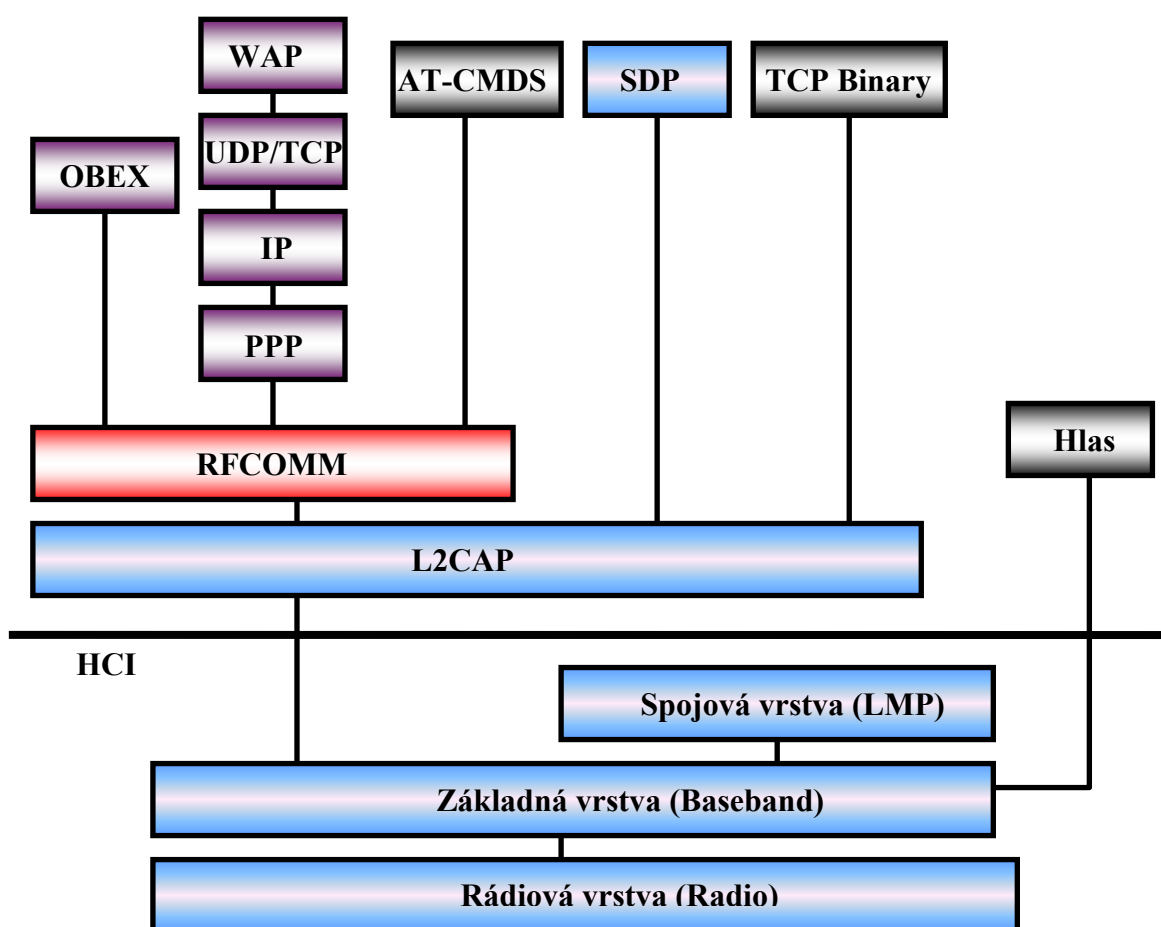
| verzia | vznik | popis | parametre |
|-----------|------------|--|---|
| 0.7 | 19.10.1998 | Prvá verzia, ktorá obsahovala definíciu radiča základného pásma (Baseband) a protokolu spojovej vrstvy LMP (Link Manager Protocol). | FH GFSK – 1Mbit/s |
| 1.0a | 26.7.1999 | Prvá verejne dostupná verzia štandardu s interoperabilitou medzi výrobcami. | FH GFSK – 1Mbit/s |
| 1.0b | 1.12.1999 | Rozšírenie štandardu riešiace problémy s interoperabilitou medzi rôznymi výrobcami. | FH GFSK – 1Mbit/s |
| 1.1 | 14.6.2002 | Ratifikovaný ako IEEE štandard 802.15.1-2002, opravy chýb z verzie 1.0b, pridanie indikátora sily prijímaného signálu RSSI (Received Signal Strength Indicator), podpora nešifrovaných kanálov, riadenie výkonu. | FH GFSK – 1Mbit/s |
| 1.2 | 5.11.2003 | Ratifikovaný ako IEEE štandard 802.15.1-2005 (revízia 802.15.1-2002), Rýchlejšie spojenie a hľadanie zariadení, zvýšenie kvality prenosu zvuku v audio kanáloch systémom eSCO (Extended Synchronous Connections), zvýšenie odolnosti proti nežiaducim interferenciám pomocou AFH (Adaptive frequency-hopping spread spectrum). | FH/AFH GFSK – 1Mbit/s |
| 2.0 + EDR | 8.11.2004 | Zvýšenie rýchlosti prenosu dát pri použití EDR, zníženie spotreby redukovaním pracovných cyklov (duty cycle), zlepšenie BER (Bit Error Rate). | FH/AFH GFSK – 1Mbit/s $\frac{\pi}{4}$ -DQPSK – 2 Mbit/s 8DPSK – 3 Mbit/s |
| 2.1 + EDR | 31.7.2007 | Nižšia spotreba, vylepšený prenos párovania, podpora NFC (Near Field Communication). | FH/AFH GFSK – 1Mbit/s $\frac{\pi}{4}$ -DQPSK – 2 Mbit/s 8DPSK – 3 Mbit/s |

2.2 Vrstvový model protokolov systému Bluetooth

Podľa účelu môžeme vrstvový model protokolov systému Bluetooth Obr. 2.1 rozdeliť do štyroch vrstiev [11] vid'. Tab. 2.2.

Tab. 2.2: Protokoly a vrstvy vo vrstvom modeli systému Bluetooth

| Protokolové skupiny | Protokoly vo vrstvom modeli |
|-----------------------------|--|
| Hlavné protokoly Bluetooth | Prenos v základnom pásme (Baseband), Protokol spojovej vrstvy (LMP), L2CAP, SDP |
| Protokol na náhradu vodičov | RFCOMM |
| Telefóny kontrolný protokol | TCS Binary |
| Adoptované protokoly | PPP, UDP/TCP/IP, OBEX, WAP |



Obr. 2.1: Vrstvový model protokolov systému Bluetooth

2.2.1 Rádiová vrstva (Bluetooth Radio)

Rádiová vrstva systému Bluetooth definuje použité frekvenčné pásmo, šírku rádiového kanálu a jej organizáciu. Ďalej určuje špecifikáciu rádiového rozhrania určuje typ použitej modulácie, modulačnú rýchlosť a charakteristiky prijímačov a vysielačov rádiových signálov [3]. S rušením a únikovými kanálmi sa vyrovnávajú pomocou

technológie frekvenčného skákania FHSS (Frequency Hopping Spread Spectrum), kde sa kanály menia 1600 krát za sekundu na 79 kanáloch, čo je dostačujúce pre prípadnú kolíziu, napríklad s DECT paketom, ktorý je obsluhovaný v rovnakom pásme.

Proti štandardným systémom (DECT), kde sa frekvencia mení jeden krát za sekundu má rýchlym skákaním výhodu vo väčšej odolnosti proti rušeniu a väčší bezpečnostný prenos [10]. Podľa vysielaného výkonu sú zariadenia Bluetooth rozdelené do troch výkonových tried vid'. Tab. 2.3. Vysielané výkony sa vzťahujú k zariadeniam s pevnou anténou a ziskom 0 dBi.

Tab. 2.3: Výkonové triedy štandardu Bluetooth

| Trieda | Dosah [m] | Max. výstupný výkon [dBm] |
|------------------|-----------|---------------------------|
| Class I | 100 | 20 |
| Class II | 10 | 4 |
| Class III | 1 | 0 |

Zariadenie používajúce výkonovú triedu 1 (Class I) musí mať implementované riadenie výkonu (power control) pre výkony vyššie než 4 dBm. Pre nižšie výkony než 4 dBm je riadenie výkonu voliteľné a je to používané hlavne k zníženiu spotreby Bluetooth zariadení. Výkon môže zariadenie meniť s minimálnym krokom 2 dB a maximálnym krokom 4 dB. K riadeniu výkonu sa používa tzv. LMP (Link Manager Protokol) príkazy. Keď jedno z komunikujúcich zariadení nepodporuje riadenie výkonu (triedy 1 alebo 2), musí i druhé komunikujúce zariadenie dodržiavať maximálny výkon 4 dBm [3].

Celková prenosová kapacita prostredníctvom Bluetooth je 1Mbit/s. Vzhľadom k tomu, že sa jedna o paketovú technológiu, po odpočítaní dát fyzického protokolu a hlavičky sa maximálna rýchlosť zníži na 723 kbit/s.

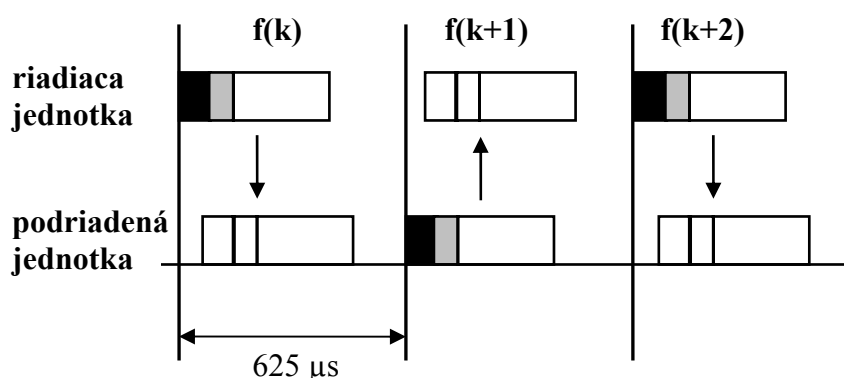
Štandard definuje dva módy. Povinný mód nazývaný tiež „Basic Rate“ používa frekvenčnú moduláciu GFSK (kľúčovanie) a dosahuje teoretické celkové prenosové rýchlosti 1 Mbit/s. Voliteľné rozlíšenie štandardu EDR používa fázovú moduláciu PSK (kľúčovanie) v dvoch variantách: pi/4-DQPSK (2 Mbit/s) a 8DPSK (3 Mbit/s). Oba módy umožňujú plne duplexný prenos a komunikácia je multiplexovaná pomocou techniky TDD (Time Division Duplex), kde sa časovo striedajú smery prenosu (riadiace zariadenie (master) vyzýva riadené zariadenie (slave) k činnosti)[3][10].

2.2.2 Špecifikácia systému Bluetooth v základnom pásme

Úlohou základnej vrstvy (Baseband) je administrácia rádiovkej vrstvy a to algoritmu skákania, opravy chýb, prípravy dát, dátové prenosy, zaistenie hlasovej a audio komunikácie, dátová bezpečnosť, identifikácia a šifrovanie [3][10].

- Fyzický kanál – tvorí pseudonáhodná postupnosť skokov cez 79 rádiových kanálov. Dva alebo viac Bluetooth zariadení používajú rovnaký kanál z pikosieti

(piconet). V pikosieti je jedna riadiaca stanica (master) a jedna alebo viac riadených (slave) staníc. Sekvencie skákania je unikátna pre každú pikosieť a vypočítava sa pomocou adresy riadiacej stanice (BD_ADDR) a vnútorných hodín riadiacej stanice (master). Podriadené stanice (slave) sú časovo a skokovo synchronizované na fyzický kanál, ktorý je rozdelený do časových úsekov (timeslotov). Časové úseky sú číslované podľa vnútorných hodín riadiacej stanice (master) 0 až $2^{27} - 1$. Každý úsek je vysielaný na jednej nosnej frekvencii a dĺžka jedného úseku môže byť 1, 3 alebo 5 timeslotov. Riadiaca (master) a riadená (slave) stanica Bluetooth vysielajú striedavo tak, že riadiaca stanica vysielá iba na začiatku párnych timeslotov, podriadená stanica na začiatku nepárnych timeslotov vid' Obr. 2.2.



Obr. 2.2: Výmena informácie medzi riadiacou a podriadenou jednotkou

- Prevádzkový kanál – medzi riadiacou a podriadenou stanicou môžu vzniknúť dva druhy prevádzkových kanálov:
 - Synchronný spojovo orientovaný SCO (Synchronous Connection-Oriented). Symetrický kanál typu bod-bod medzi riadiacou a podriadenou stanicou v pikosieti. Pakety sú vysielané v rezervovaných intervaloch (rezervovaná šírka pásma) rýchlosťou prenosu 64 kb/s na 1 timeslot v intervale a presne danou periódou, preto sú vhodné pre prenos časovo kritických informácií (audio-informácie). Každá Bluetooth stanica dokáže obsluhovať naraz až 3 SCO kanály. Riadiaca stanica môže obsluhovať 3 SCO spoje s jednou alebo až s 3 riadenými stanicami.
 - Asynchronný bezspojovo orientovaný ACL (Asynchronous Connection-Less). Jedná sa o spoje typu bod-multybod medzi riadiacou stanicou a všetkými riadenými stanicami podieľajúcich sa v pikosieti. Prevádzkový kanál ACL je vysielaný v časových úsekoch, ktoré nie sú pre prenos SCO kanálu. Poskytuje paketovo orientované asynchrónne spojenie a je vhodné pre prenos všetkých dátových tokov, ktoré nie sú citlivé na časové oneskorenie (best effort). Prenos súboru (OBEX), FTP atď.

Dáta sú prenášané prostredníctvom paketov v timeslotoch. Pakety môžu mať iba presne definovanú dĺžku, ktorá môže byť 1 timeslot (625 μ s), 3 timesloty (1,875 ms) alebo 5 timeslotov (3,125 ms). Každý paket nesie informácie o odosielateľovi, prijímateľovi a spôsobom kompresie. Na začiatku každého paketu je prístupový kód (72 bitov), a ním nasleduje záhlavie (54 bitov) a nakoniec sú prenášané užívateľské dáta (0-2745 bitov). Prístupový kód je určený riadiacou stanicou, má pseudonáhodné vlastnosti, je unikátny pre každú pikosieť a slúži k synchronizácii a autorizovanému prístupu do siete. Záhlavie obsahuje adresu riadenia prístupu na médium, informácie o typu paketu, riadiace bity, bity pre opravu chýb (ARQ) a bity pre doprednú kontrolu chýb v záhlaví v podobe 18 informačných bitov, ktoré sú zabezpečené kanálovým kódovaním FEC na výsledný počet 54 bitov. Celková dĺžka paketu môže byť v rozmedzí 126 až 2871 bitov.

2.2.3 Ostatné protokoly a rozhranie

Link Manager Protocol (LMP)

Konfigurovanie linky zaisťuje Link Manager a to synchronizáciu zariadení pri vytváraní spojenia, autentifikáciu zariadení na základe privátneho kľúča s využitím metódy výzva/odzva, šifrovanie spojenia, detekcia a korekcia chýb a mnoho ďalších funkcií. Vyhľadáva v okolí iné zariadenia a komunikuje s ich Link Managerom prostredníctvom Link Manager Protokolu (LMP) a vytvára spojenie. Na vykonávaní im poskytovaných služieb Link Manager používa služby vrstvy Link Controlleru [3][10].

Host Controller Interface (HCI)

Host Controller Interface je rozhranie, ktoré poskytuje uniformný prístup softwarovej časti protokolového zásobníka k fyzickej časti zariadenia Bluetooth (hardware a firmware). Skladá sa z troch častí [12]:

- HCI firmware – Nachádza sa v časti Bluetooth hardware. Implementuje HCI príkazy prostredníctvom príkazov Baseband a Link Manager vrstvy, prístup k hardwarovým, stavovým a riadiacim registrom a k registrom udalostí.
- HCI driver – Nachádza sa v časti Host (software). Zabezpečuje analýzu prijatých udalostí, na základe ktorých poskytuje informácie vyšším vrstvám.
- Host Controller Transport Layer (Physical Bus Driver) – definuje niekoľko rôznych spôsobov komunikácie: USB, UART a RS-232. Tieto spôsoby pripojujú hardwarové časti Bluetooth zariadení k hlavnému zariadeniu (Host).

Logical Link Control and Adaption Protocol (L2CAP)

Jedná sa o vrstvu dátových spojení, ale v softwarovej časti umiestnená nad základnou vrstvou Baseband. Poskytuje ovládanie pre aplikácie, lebo tie nemajú k hardwaru priamy prístup. Zapuzdruje pakety do formátov vhodných pre nižšie vrstvy

a multiplexuje spojenie tak, aby mohlo byť využívané viacerými aplikáciami. Maximálna veľkosť L2CAP paketov je 64 KB a poskytuje vyšším vrstvám spojovo a nespojovo orientované dátové služby. Tiež monitoruje kvalitu služieb pre zaistenie parametrov dohodnutých pri vytváraní spojenia. Prenáša dáta medzi rôznymi protokolmi na vyšších vrstvách. Podporuje dva typy spojení pre základnú vrstvu Baseband a to SCO a ACL [12].

Protokol RFCOMM

RFCOMM (Radio Frequency Communication) je to jednoduchý protokol poskytujúci emuláciu sériového portu RS-232 pomocou L2CAP. Tento protokol RFCOMM podporuje až 60 simulovaných sériových liniek medzi dvoma zariadeniami. Počet súčasne spustených liniek je špecifikovaný pre každú implementáciu rôzne [12].

Service Discovery Protocol (SDP)

Protokol poskytuje aplikáciám informácie o dostupných službách na iných Bluetooth zariadeniach v danej pikosieti. Vyhľadáva a popisuje ich. Je nenáročný na kapacitu fyzického kanálu. Využíva požiadavka/odpoveď model komunikácie, kde každá komunikácia sa skladá z jedného požiadavku a jednej odpovedi. Protokol obsahuje obmedzenie, že môže existovať iba jedna požiadavka bez odpovedi na jedno pripojené zariadenie. Inými slovami klientovi musí prísť odpoveď na každú požiadavku predtým než vyšle ďalšiu požiadavku v rámci toho istého L2CAP spojenia [12].

Telephony control – binary (TCP Binary)

Protokol je bitovo orientovaný a definuje vytvorenie prenosovej linky, prenos hlasu, dát a riadenie medzi jednotlivými Bluetooth zariadeniami.

Hlas

Štandard Bluetooth definuje služby pre prenos zvuku medzi zariadeniami Bluetooth. Prenos zvuku je po otvorení a zostavení prenosovej linky medzi dvoma zariadeniami Bluetooth zaisťovaný priamo, nevyužíva sa služba vrstvy L2CAP.

Telephony control – AT commands (AT-CMDS)

Štandard Bluetooth podporuje skupinu servisných príkazov AT, využívaných pre riadenie a konfiguráciu sériových telefónnych modemov. Táto služba využíva vrstvu RFCOMM, emulujúcu služby sériového portu a umožňuje komunikáciu s jednotkou Bluetooth pomocou textových AT príkazov s plne definovanou syntaxou.

Point-to-point (PPP)

Protokol point-to-point (PPP) je paketovo orientovaný protokol. Je súčasťou TCP/IP, kde sa využíva k prenosu paketov IP cez sériové rozhranie RS-232. Protokol využíva k zaisteniu spojenia služby vrstvy RFCOMM.

Protokoly UDP - TCP - IP

Skladba protokolov UDP - TCP - IP slúžia k prepojeniu zariadenia Bluetooth so zariadením na internetu. Aplikácia využívajúca k prenosu dát protokol IP ich prenáša cez protokol PPP, ktorý ich ďalej predáva vrstve RFCOMM.

WAP

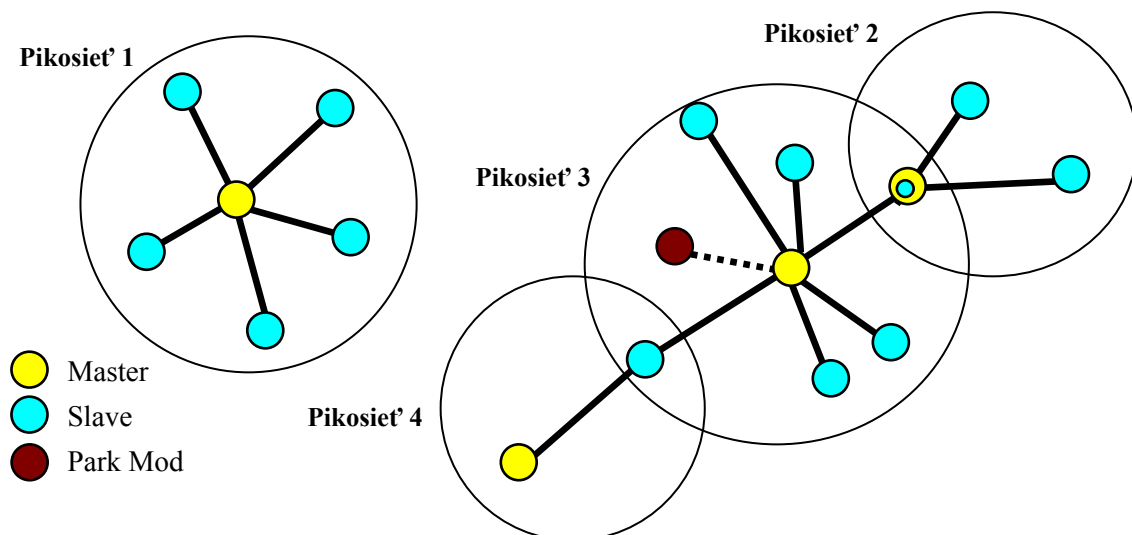
Wireless application protocol (WAP) je protokol pre bezdrôtovú komunikáciu, určená k sprístupneniu internetových služieb v rámci rôznych bezdrôtových komunikačných sieťových riešení. Keďže protokol WAP je primárne vytvorený pre mobilné zariadenia, Bluetooth môže byť v tomto prípade základnou bezdrôtovou sieťou pre prenos informácie od riadiacej jednotky WAP ku klientovi WAP.

Protokol OBEX

Protokol OBEX (OBject EXchange) je voliteľný protokol aplikačnej vrstvy navrhnutý pre výmenu dát a riadiacich informácií. Protokol využíva architektúru klient-server a je nezávislý na transportnom mechanizme a prenosovom programovom rozhraní. OBEX používa RFCOMM ako hlavný transportný protokol.

2.3 Topológia Bluetooth

Dve alebo viac (maximálne 8) aktívnych zariadení používajúcich rovnaký fyzický kanál vytvárajú pikosieť (ad-hoc sieť). Bluetooth podporuje jak dvojbodovú, tak mnohobodovú komunikáciu.



Obr. 2.3: Topológia siete Bluetooth

V pikosieti je jedno zo zariadení riadiaca stanica (master) a ostatné sú podriadené stanice (slave). Protokol umožňuje pripojiť i viac podriadených staníc k riadiacemu s využitím tzv. parkovacieho módu (Park Mod). Jedná sa o neaktívne pripojenie k riadiacej stanici, ale zachováva sa synchronizácia s pikosieťou a teda aj s riadiacou stanicou. Špecifikácia

dovoľuje simultánne použiť až 10 pokosiet na ploche o dosahu 10 metrov. Pikosieť možno združovať do tzv. rozprestretých sietí (scatternet) vid' Obr. 2.3. Všetky dáta sú prenášané prostredníctvom paketov [3].

2.4 Bluetooth Profily

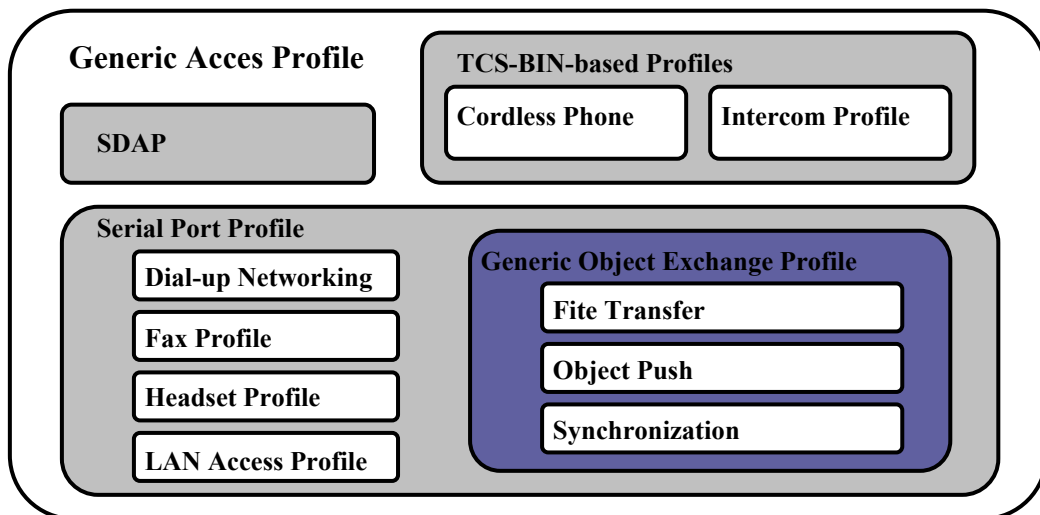
Bluetooth sa skladá z niekoľko protokolov, ktoré sú rozdelené do vrstiev a tomu sa hovorí vrstvový model. Bluetooth profil určujú, ktoré protokoly budú použité skrz vrstvový protokolový model vid' Obr. 2.4. Profily teda netvorí nové funkcie, ale určujú, ktoré protokoly sa z celej špecifikácie Bluetooth využijú. Pre Bluetooth špecifikáciu 1.1 existuje 13 základných profilov [11][12]:

- GAP (Generic Access Profile) je základný profil pre všetky ostatné, aby sa tým zaistila všeobecná kompatibilita. Zaisťuje napr. hľadanie zariadení v okolí a naviazanie spojenia.
- SDAP (Service Discovery Application Profile) je určený k vyhľadávaniu služieb poskytovaných ostatnými dostupnými zariadeniami Bluetooth.
- CTP (Cordless Telephony Profile) je profil bezdrôtového telefónu. Je využívaný bezdrôtovými telefónmi pre zaistenie bezdrôtového spojenia so základovou stanicou.
- IP (Intercom Profile) umožňuje naviazať spojenie medzi dvoma bezdrôtovými telefónmi bez toho aby bol hovor smerovaný cez telefónu sieť.
- SPP (Serial Port Profile) používa RFCOMM protokol. Emuluje sériovú linku a tak poskytuje jednoduchú bezdrôtovú implementáciu náhrady za existujúci RS-232 až do rýchlosti 128Kb/s.
- HP (Headset Profile) umožňuje bezdrôtovo pripojiť slúchadlo s mikrofónom k počítaču, mobilnému zariadeniu (hands-free) alebo inému zariadeniu podporujúce tento profil.
- DNP (Dial-up Network Profile) je profil, ktorý poskytuje štandardný prístup k Internetu a inými vytáčaným službám cez Bluetooth.
- FP (Fax Profile) umožňuje bezdrôtovo naviazať faxové spojenie prostredníctvom verejnej telefónnej siete a odosielať a prijímať faxy prostredníctvom jednotky s rozhraním Bluetooth.
- LAP (LAN Access Profile) definuje dátové spojenie medzi jednotkami Bluetooth a lokálnou sieťou LAN. Profil umožňuje sa bezdrôtovo pripojiť k miestnej sieti LAN alebo k internetu protokolom PPP.
- GOEP (Generic Object Exchange Profile) definuje protokoly a procedúry určené k prenosu objektom medzi zariadeniami. Využíva sa pri prenosu súboroch a synchronizácii medzi zariadeniami.

- OPP (Object Push Profile) je základný profil pre posielanie „objektov“ ako sú obrázky, elektronické vizitky a ďalšie. OPP používa API z profilu OBEX. Prenos je vždy inicializovaný odosielateľom (klient) a nie príjemcom (server).
- FPP (File Transfer Profile) zaisťuje prenos súborov medzi zariadeniami Bluetooth. Profil stavia na profile GOEP.
- SP (Synchronization Profile) slúži k synchronizácii elektronických kalendárov, diárov a iných aplikácii bežiacich na jednotlivých zariadeniach Bluetooth. Profil stavia na profile GOEP.

Existujú i ďalšie ako napr. PAN, HID. Jednotlivé profily sú nahrane už pri výrobe alebo sú nahrane pomocou ovládačov alebo firmwarov.

- PAN (Personal Area Networking Profile) popisuje, ako môžu dve a viac zariadení s podporou Bluetooth vytvoriť ad-hoc sieť a ako môže s pomocou rovnakého mechanizmu pristupovať k vzdialenej sieti cez prístupový bod.
- HID (Human Interface Device Profile) poskytuje podporu pre zariadenia ako sú klávesnice, myši a ďalšie. Je navrhnutý na poskytovanie spojenia s nízkym oneskorením a s nízkymi energetickými požiadavkami. Bluetooth HID vychádza z HID protokolu definovaného pre USB, čo zvyšuje kompatibilitu v rôznych operačných systémoch.



Obr. 2.4: Hierarchia vybraných Bluetooth profilov

3. Ostatné technológie WPAN

3.1 Ultra-WideBand

Technológia UWB sa vybrala dosť odlišnou cestou ako Bluetooth alebo ZigBee. UWB signál sa vysiela vo forme mnoho tisícou krátkych pulzov (0,2 ns až 1,5 ns) rozprestretých v celej šírke, ktorá odpovedá až niekoľkým GHz. Ako je známe zvýšením vysielacieho výkonu zvýšime dosah a zvýšením šírky pásma zvýšime dátovú rýchlosť, práve preto UWB ponúka veľmi vysokú dátovú rýchlosť s relatívne nízkou spotrebou pre aplikácie na krátku vzdialenosť. Dobré sa znáša s inými rádiovými technológiami, nie je náchylná na rušenie a je veľmi bezpečná proti odposluchu. Signál UWB má menší výkon než iné signály na rovnakom kmitočtu (vrátane šumu), takže teoreticky nedochádza k žiadnemu rušeniu. Systémy vykazujú rovnomerné rozprestretie výkonu cez veľmi veľký rozsah kmitočtu. Preto ju je možno hlavne pre úzkopásmové prijímače dosť ťažko odlišiť od šumu. Ďalšia výhoda súvisiaca s pulznou moduláciou použitou u UWB je minimalizácia skreslenia odrazom, teda príjem signálu z viacerých ciest (multipath distortion) a tým menší vplyv má medzi-symbolová intenzita ISI (Inter-Symbol Interference) [13].

UWB je technológia budúcnosti a jej hlavnou úlohou je nahradiť všetky vodiče v domácnosti medzi TV, DVD prehrávačom, domácim kinom a podobne. Dátová rýchlosť na malé vzdialenosti dosťahuje HDTV prenosom videa a viackanalového zvuku. Kompresný pomer môže eliminovať klesajúcu dátovú rýchlosť s rastúcou vzdialenosťou.

Pracovná skupina IEEE 802.15.3a, ktorá mala za úlohu vytvoriť jeden štandard pre UWB, v roku 2003 vyhlásila súťaž o riešenie fyzickej vrstvy pre UWB. Zadaním bolo ponúknuť rýchlosť 110 Mbit/s na vzdialenosť minimálne 10 m (a 480 Mbit/s do 1 m), s využitím pásma o šírke 500 MHz z kmitočtového intervalu 3,1 GHz až 10,6 GHz. Rozhodovalo sa medzi dvoma návrhmi: MBOA (Intel) z WiMedia Alliance a DS-UWB (FreeScale a Motorola) z UWB Forum.

Na podporu vyriešenia situácie s výberom konečného riešenia pre fyzickú vrstvu 802.15.3a vznikol kompromis, ktorý riešil rozdielnosť medzi oboma návrhmi prostredníctvom protokolu CSP (Common Signaling Protocol). Ten umožňoval viacerým typom fyzickej vrstvy UWB spoluprácu v rovnakom dosahu, bez toho aby došlo k vzájomnému rušeniu. Návrh prišiel od spoločnosti Pulse-Linc Inc. Využíva pásmo spoločného MBOA a DS-UWB (500 MHz okolo stredu 3,9 GHz) a používa buď BPSK (Binary Phase-Shift Keying), alebo DPSK (Differential PSK). Pomocou TDMA (Time-Division Multiple Access) sa kanál prideliuje kolokovaným UWB zariadeniam. Návrh sa nestretol s veľkou pozornosťou.

Dlhé mesiace neschopnosti dosiahnuť zhody pri tvorbe návrhu technického riešenia bola pracovná skupina 802.15.3a dňa 19.1.2006 na Havai rozpustená. Dopomohol k tomu

i fakt, že Intel a nim vedená WiMedia Alliance nečakala na rozhodnutie štandardizačnej skupiny IEEE a svoje riešenie UWB prijali ako normu v organizácii ECMA (European Computer Manufacturers Association) [14][15].

3.1.1 MBOA

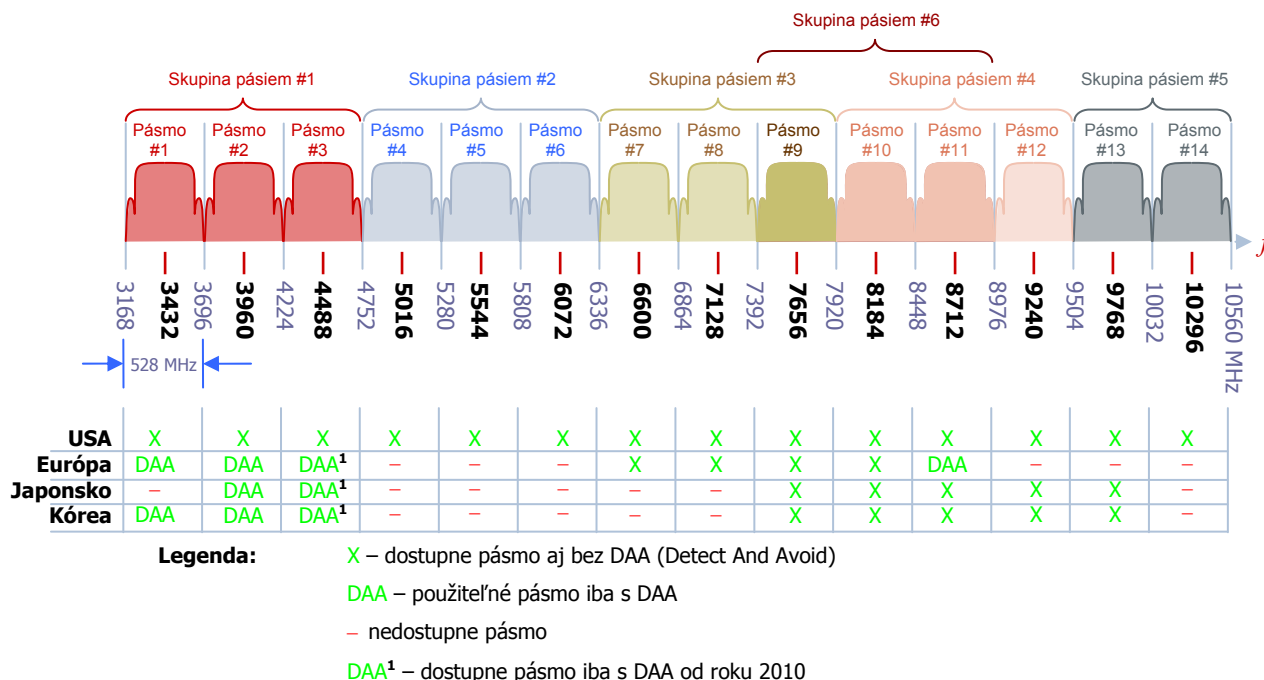
Návrh skupiny pod názvom MultiBand OFDM Alliance (MBOA) predstavilo riešenie MB-OFDM (MultiBand Orthogonal Frequency Division Multiplexing). Princíp je podrobne popísaný v štandarde ECMA-368 [5] a ECMA-369 [6] pre spoločnú rádio platformu WiMedia UWB. Táto technológia bola tiež štandardizovaná v medzinárodnej ISO špecifikácii ISO/IEC 26907 a ISO/IEC 26908.

Spoločná rádiová platforma

Spoločná rádiová platforma zahŕňa Fyzickú vrstvu a vrstvu prístupu k médiu vid' Obr. 3.4. Technológia podporuje prenosové rýchlosti 53,3; 80; 106,7; 160; 200; 320; 400 a 480 Mbit/s. Spočíva v rozdelení UWB pásma 3,1 - 10,6 GHz na menšie pásma po 528 MHz, kde sa ich vojde 14. Pre pásma platí vzťah 3.1.

$$f_c(n_b) = 2904 + 528 \times n_b \text{ (MHz)}, n_b = 1, \dots, 14. \quad (3.1)$$

Kanály sú rozdelené do 4 skupín po troch pásmach, 5. skupiny obsahuje dva pásma a 6. skupina sa prekrýva s 3. a 4. vid' Obr. 3.1. U technológii sa vyžaduje v niektorých pásmach DAA (Detect And Avoid), aby bola schopná detekovať vysielania prioritnejších zariadení (Radar, WiMax) a vyhnúť sa ich vysielaniam [27].



Obr. 3.1: Rozdelenie pásem (WiMedia Alliance, Júl 2007 [27])

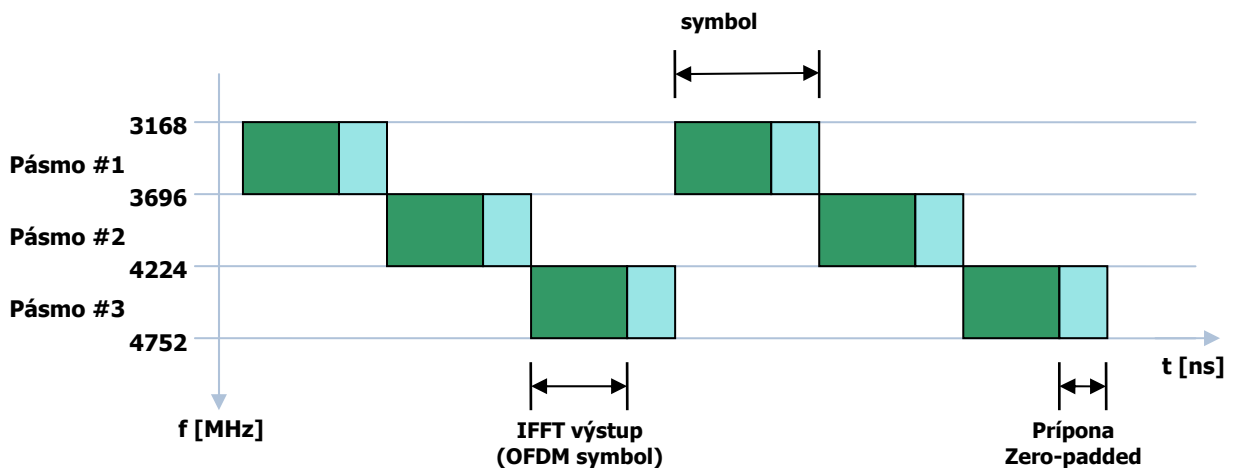
Zatiaľ sa počíta s výrobou iba zariadení s podporou prvej skupiny pásiem, ale v marci 2008 vyšlo oznámenie WiMedia Aliance o registrácii dvoj-pásmových čipsetov podporujúcich prvú a tretiu skupinu kanálov [25]. Ďalšie skupiny kanálov sa pridajú v budúcnosti.

OFDM modulátor vytvára ortogonálny frekvenčne delený multiplex nosných. Procesom inverznej rýchlej furierovej transformácie (IFFT) sa vytvoria jednotlivé nosné (128), ktoré sú potom modulované namapovanými symbolmi s pridaním prípony s nulami tzv. zero-padded suffix (Obr. 3.2) na konci korekčnej doby [28].

Signál vytvorený OFDM moduláciou má 128 sub-nosných, kde sa používa QPSK (Quadrature Phase Shift Keying) alebo 16QAM (16 Quadrature Amplitude Modulation) na každej nosnej. Z toho 100 je dátových, 12 pilotných signálov slúžia pre synchronizáciu frekvenčného offsetu a fázy, 10 sub-nosných je bezpečnostných a posledných 6 pilotných signálov vrátane jednosmernej zložky sú nulové. Vysielací výkon nižší ako -41dBm/MHz [28].

Prípona z núl slúži k zníženiu medzi-symbolovej interferencie ISI (Inter-Symbol Interference) a medzi-kanálovej interferencie ICI (Inter-Channel Interderence), spôsobenej viaccestným šírením signálu. Ak je doba trvania nulovej prípony dlhšia ako oneskorenie viaccestného šírenia, potom je ISI eliminovaná. Keďže nulová prípona nenesie žiadnu informáciu, tým znižuje skutočnú informačnú priepustnosť, nemala by byť príliš dlhá.

Táto technológia je určená pre použitie v prenosných zariadeniach závislých na batérii preto je spotreba energie jednou z najdôležitejších požiadaviek pre WPAN. WiMedia Aliance uvádza spotrebu 1mW/Mbps.



Obr. 3.2: Vysielanie signálu použitím troch pásiem

Navrhnutý UWB systém ďalej rozprestiera kódovane dáta technikou frekvenčného skákania použitím časovo-frekvenčných kódov TFC (Time-Frequency Codes) vid'. Obr. 3.2. Štandard ECMA špecifikuje tri typy TFC.

- kódované informácie sú rozprestreté cez tri pásma, nazývané ako časovo-frekvenčne prekladanie TFI (Time-Frequency Interleaving).
- kódované informácie sú rozprestreté cez dva pásma, nazývane dvoj-pásmove TFI alebo TFI2.
- kódované informácie sú vysielané v jednom pásme nazývane tiež fixovane frekvenčné prekladanie FFI (Fixed Frequency Interleaving).

Podpora TFI, TFI2 a FFI je povinná. V rámci prvých štyroch a šiestej skupiny sa používajú štyri TFC typu TFI, tri TFI2 a tri FFI, čo umožňuje až desať kanálov v každej z týchto skupín. V piatej skupine sa používajú dva FFI a jeden TFI2. V šiestej skupine sa FFI kanály a jeden TFI2 kanál plne prekrýva s kanálmi v tretej a štvrtej skupine, celkovo teda 53 použiteľných kanálov.

Pre doprednú korekciu chýb sa využíva puncturové konvolučné kódovanie s kódovým pomerom 1/3, 1/2, 5/8 a 3/4 vid'. Tab. 3.1. Čím je kódový pomer nižší, tým sa pridáva k dátam viac redundancie a tým je oprava chýb v prijímači účinnejšia. Technológia ponúka rýchlosť 480 Mbit/s na vzdialenosť 2 - 4 m a 110 Mbit/s na vzdialenosť 10 – 15 m so zaistením QoS. Pre zlepšenie výkonnosti a maximalizáciu frekvenčnej diverzity v prítomnosti iných zariadení sa používa rozprestretie vo frekvenčnej a časovej oblasti. Rozprestretie vo frekvenčnej oblasti spočíva vo vysielaní tej istej informácie na dvoch samostatných sub-nosných s rámci rovnakého OFDM symbolu. Rozprestretie v časovej oblasti spočíva vo vysielaní rovnakej informácie v dvoch po sebe idúcich OFDM symboloch [5].

Tab. 3.1: Podporované prenosové rýchlosti MultiBand OFDM

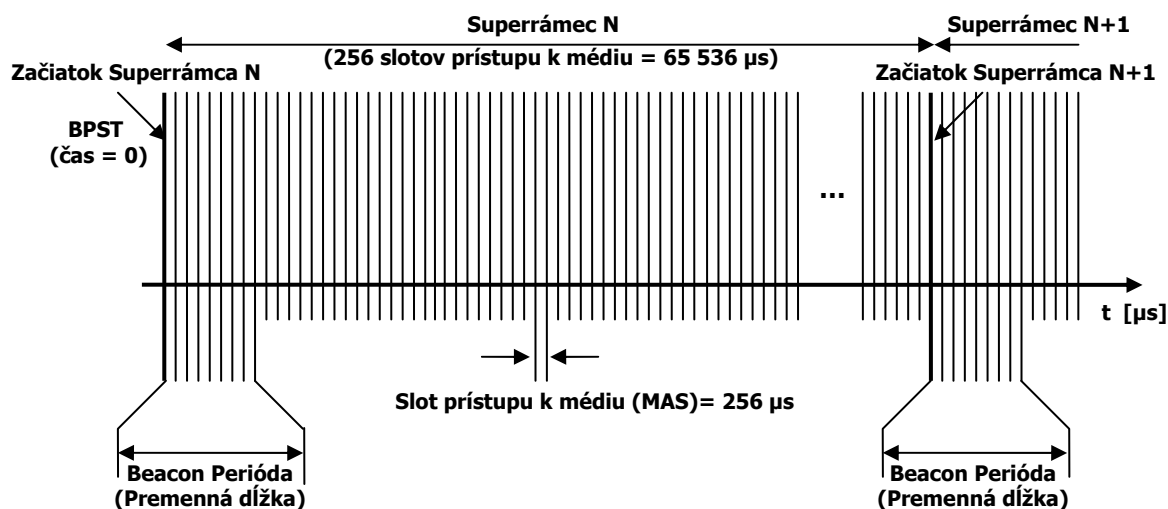
| Dátová rýchlosť [Mbit/s] | Modulácia | Kódový pomer | Rozprestretie vo frekvenčnej oblasti | Rozprestretie vo časovej oblasti |
|--------------------------|-----------|--------------|--------------------------------------|----------------------------------|
| 53,3 | QPSK | 1/3 | Áno | Áno |
| 80 | QPSK | 1/2 | Áno | Áno |
| 106,7 | QPSK | 1/3 | - | Áno |
| 160 | QPSK | 1/2 | - | Áno |
| 200 | QPSK | 5/8 | - | Áno |
| 320 | 16QAM | 1/2 | - | - |
| 400 | 16QAM | 5/8 | - | - |
| 480 | 16QAM | 3/4 | - | - |

Kanál prístupu k médiu je rozdelený do superrámecov vid'. Obr. 3.3. Dĺžka jedného superrámca je 65 536 μ s a je rozdelená do 256 slotov MAS (Medium Access Slots) s dĺžkou trvania jedného slotu je 256 μ s. Superrámec umožňuje všetkým zariadenia vo WPAN aby boli synchronizované. To umožňuje koordinovať vysielanie a prijímanie informácií a tým optimalizovať prístup k médiu a spotrebu energie. Superrámec pozostáva

s beacon periódy BP (Beacon Period) a periódy určenej na prenos dát DTP (Data Transfer Period). BP je premennej dĺžky a obsahuje pevne dĺžky beacon slotov, ktoré využívajú susedné zariadenia v okolí každého zariadenia na odoslanie vlastných beacon nesúce informáciách o zariadení a riadení siete. Pomocou beacon sa tiež vysielajú informácie napr., že zariadenie prechádza do stavu šetrenia energie (hibernation) alebo prechádza na iný kanál a podobne. Každé zariadenie si definuje vlastné BP. Aby nedochádzalo k vzniku viacerých BP a bol poskytnutý riadny prístup k médiu všetkým, každé zariadenie skenuje médium pred zahájením svojej činnosti. Ak pri skenovaní média detekuje od iného zariadenia BP, pripojí sa k nemu. Ak nedetekuje žiadnu BP vytvorí nový BP posielaním vlastného beaconu a tým zaistí koexistencie medzi zariadeniami. V zbytku superrámca, perióde prenosu dát DTP, zariadenia prístupujú k médiu a to:

- použitím dopredu určeného MAS, ktorý si zariadenia rezervovali pomocou protokolu DRP (Distributed Reservation Protocol)
- konkurenčným prístupovým mechanizmom PCA (Prioritized-based Channel Access).

DRP protokol umožňuje rôznorodým zariadeniam rezervovať MAS. Všetky zariadenia musia rešpektovať rezervované sloty, vhodná pre isochrónnu komunikáciu. V prípade prístupu PCA je potrebné pred vysielaním sledovať médium, s tým je spojená malá réžia a je to vhodnejšie pre asynchrónnu komunikáciu [5].



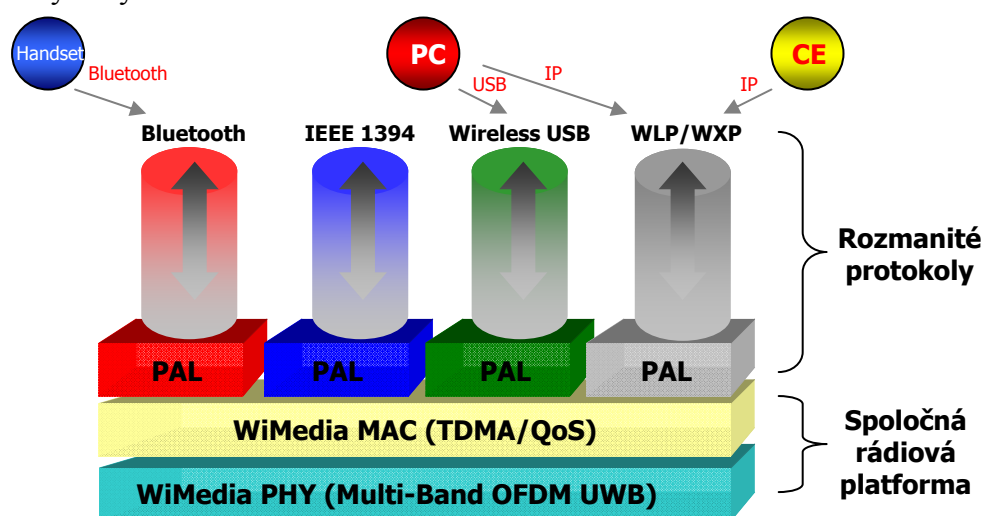
Obr. 3.3: Štruktúra superrámca kontroly prístupu k médiu MAC

Protokolovo adaptačná vrstva MBOA

Technológia od WiMedia Alliance zahrnuje protokolovo adaptačnú vrstvu PAL (Protocol Adaptation Layer), ktorá zaisťuje konvergenciu medzi množstvom protokolov ako Wireless USB, Bluetooth, IEEE 1394, WLP/WXP (WiMedia Link Layer Control protocol/WiMedia eXtended Platform). PAL pracujúcej nad MAC a PHY vrstvou UWB a v podstate predstavuje PHY a MAC pre samotný napr. Bluetooth vid' Obr. 3.4. Vďaka

tomu, že technológie Wireless USB a Bluetooth atď. používajú rovnako definované UWB, môžu spoločne koexistovať. Prostredníctvom WLP a WXP umožňuje tejto univerzálnej technológii podporu i TCP/IP. WLP je protokol adaptačnej vrstvy PAL a slúži ako rozhranie medzi vyššími vrstvami sieťových protokolov a WiMedia MAC vrstvou. WXP je rozširujúca platforma pre poskytnutie schopnosti plnej spolupráce protokolov a teda aplikácii založených na IP. WiMedia UWB je optimalizovaná pre osobné počítače (PC), spotrebnú elektroniku (CE), mobilné zariadenia a automobilový segment trhu.

Na fyzickej vrstve je možné dosiahnuť dátovú rýchlosť až 480 Mbit/s čo je dosť pre podporu mnoho ďalších požadovaných protokolov ako je aj Ethernet, DVI a HDMI. WiMedia plánuje podporovať bezdrôtové multimediálne aplikácie pracujúce s prenosovými rýchlosťami až niekoľko Gbit/s.



Obr. 3.4: WiMedia spoločná UWB platforma

Pre predstavu výkonnosti tejto technológie WiMedia UWB uviedla príklady v [26]. Typické prenosné zariadenie s batériou 1800mAh (nabitá na 85%) pri prenosovej rýchlosti UWB = 480 Mbit/s (účinných 70%) dokáže preniesť za 24 sekúnd 1GByte na vzdialenosť 2 m až 4 m pri ktorej sa spotrebuje 0,05% batérie (UWB = 1mW/Mbps). Na vybitie celej batérie by sa prenieslo 1,8 TByte za dobu 13,9 hodiny.

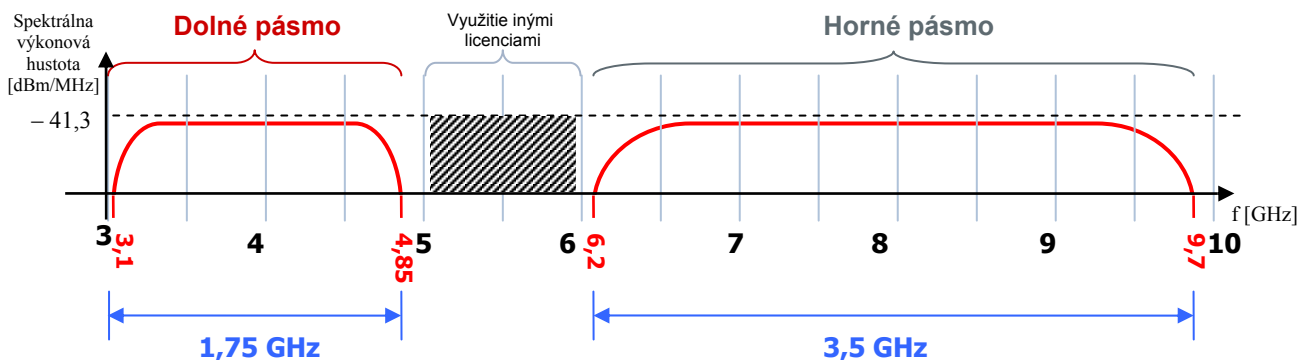
3.1.2 DS-UWB

Radikálnejší návrh predložila skupina, ktorá pod názvom DS-UWB prezentovala metódu DS-CDMA (Direct-Sequence Code-Division Multiple Access).

Spoločná rádiová platforma

Metóda DS-CDMA spočívala v priamej postupnosti viacnásobného prístupu s kódovým delením a s použitím binárnej fázovej modulácie BPSK. Pracuje v dvoch nezávislých pásmach. Nižšie pásmo leží medzi 3,1 – 4,85 GHz, vyššie pásmo medzi 6,2 – 9,7 GHz vid' Obr. 3.5. Medzera medzi pásmami je spôsobená využitím pásma inými

licenciami. V každom pásme sa vyskytuje 6 jedinečných pikosieťových kanálov. Štandard požaduje po výrobcach funkčnosť 1. až 4. (nižšie pásmo) kanálu. Implementácia vyšších kanálov 5.až 12. (nižšie a vyššie pásmo) je voliteľná vid' Tab. 3.3. Dátové toky v jednotlivých kanáloch sa od seba líšia inými sekvenciami rozprestieracieho kódu cez viac impulzov použitím PRN (Pseudo Random Code) [16][18].



Obr. 3.5: Rozdelenie pásiem DS-UWB

UWB poskytuje vo vzdialenosti niekoľko metrov vysoké dátové rýchlosti vid' Tab. 3.2. Každá prenosová rýchlosť dát v Tab. 3.2 určuje „nominálnu“ hodnotu dátovej rýchlosti.

Tab. 3.2: Prenosové rýchlosti definované v dolnom pásme DS-UWB

| Nominálna Prenosu rýchlosť dát [Mbit/s] | Kódový pomer | BPSK dĺžka kódu | 4-BOK dĺžka kódu | Symbolová rýchlosť |
|--|-----------------|--------------------|------------------------|-----------------------|
| 28 | 1/2 | L=24 | - | F_{chip}/L |
| 55 | 1/2 | L=12 | - | F_{chip}/L |
| 82.5 | 3/4 | L=12 | - | F_{chip}/L |
| 110 | 1/2 | L=6 | L=12 | F_{chip}/L |
| 110 | 1 | L=12 | L=24 | F_{chip}/L |
| 165 | 3/4 | L=12 | L=24 | F_{chip}/L |
| 220 | 1/2 | L=3 | L=6 | F_{chip}/L |
| 220 | 1 | L=6 | L=12 | F_{chip}/L |
| 330 | 1/2 | L=2 | L=4 | F_{chip}/L |
| 440 | 1 | L=3 | L=6 | F_{chip}/L |
| 500 | 3/4 | L=2 | L=4 | F_{chip}/L |
| 660 | 1 | L=2 | L=4 | F_{chip}/L |
| 660 | 1/2 | L=1 | L=2 | F_{chip}/L |
| 1000 | 3/4 | L=1 | L=2 | F_{chip}/L |
| 1320 | 1 | L=1 | | L=2 |

V praxi, každá pikosieť pracuje s trochu rozdielnymi čipovými rýchlosťami (F_{chip}) a z toho dôvodu presná prenosová rýchlosť v určitom móde je pre každú pikosieť trochu odlišná. Pre príklad: v móde 660 Mbit/s ak máme štyri pikosiete, kde každá má inú čipovú rýchlosť tak aj výsledne prenosové rýchlosti v rozsahu 656 až 676 Mbit/s.

Pre jednoduchú synchronizáciu v prijímačoch a udržanie dobrého oddelenia medzi rôznymi kanálmi pikosietí sa používa pridelovacia (acquisition) sekvencia modulovaná pridelovacím kódovým slovom PAC (Piconet Acquisition Codeword). Každá pikosieť používa teda jeden z PAC v kombinácii s určitou čipovou rýchlosťou [16].

Tab. 3.3: Čísla kanálov pikosietí, čipová rýchlosť a použitie rozprestieraného kódu

| Kanál pikosiete | Čipová rýchlosť [MHz] | Stredná frekvencia [MHz] | Rozprestieraci kód (PAC) |
|-----------------|-----------------------|--------------------------|--------------------------|
| 1 | 1313 | 3939 | 1 |
| 2 | 1326 | 3978 | 2 |
| 3 | 1339 | 4017 | 3 |
| 4 | 1352 | 4056 | 4 |
| 5 | 1300 | 3900 | 5 |
| 6 | 1365 | 4094 | 6 |
| 7 | 2626 | 7878 | 1 |
| 8 | 2652 | 7956 | 2 |
| 9 | 2678 | 8034 | 3 |
| 10 | 2704 | 8112 | 4 |
| 11 | 2600 | 7800 | 5 |
| 12 | 2730 | 8190 | 6 |

Návrhový štandard používa moduláciu BPSK alebo 4-BOK (quaternary bi-orthogonal keying), ktorá je nepovinná. BPSK a 4-BOK modulácie sa využívajú na vytvorenie dátových symbolov. Symboly vysielané za sebou zostavujú sekvenciu UWB pulzov. Rôzne podporované prenosové rýchlosti Tab. 2.1 závisia [16]:

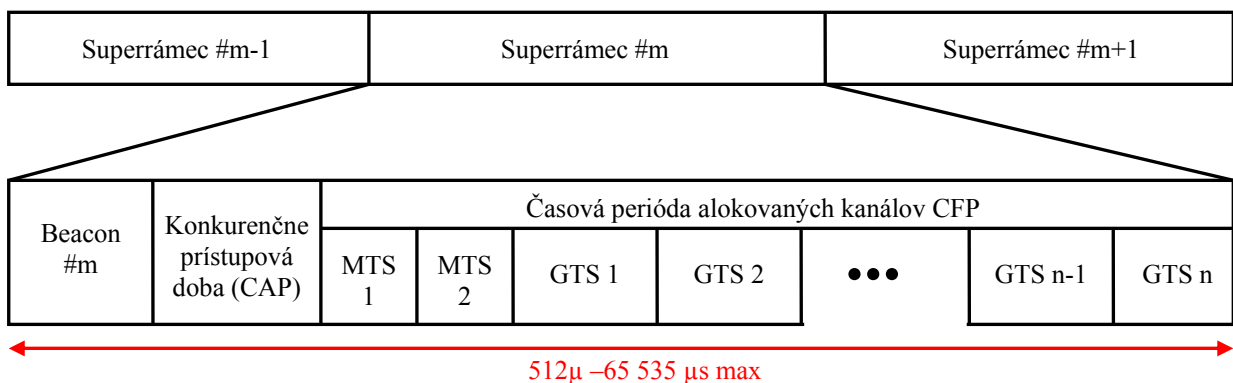
- na použití jednej z rozprestieracích kódových sekvencií, kde každá sekvencia môže mať 1 až 24 pulzov (chips).
- na rôznej dĺžke jednej z dvoch špecifikovaných konvolučných kódov použitého pre doprednú korekciu chýb.

Vysoká symbolová rýchlosť zvyšuje riziko medzi-symbolovej interferencie a preto je nutné používať ekvalizáciu na vstupe. Vo vnútri miestnosti sa signál šíri niekoľkými

cestami a doporučuje sa používať RAKE prijímače, ktorý podobne ako u CDMA potlačuje viaccestné šírenie.

Kanál prístupu k médiu je rozdelený do superrámecov vid'. Obr. 3.6. Dĺžka jedného superrámca môže byť maximálne 65 535 μ s s dĺžkou trvania jedného slotu je 256 μ s. Superrámec sa pozostáva z troch častí [16][18]:

- beacon, používa sa na prenos kontrolných informácií celou pikosieťou, na alokáciu GTS (Guaranteed Time Slots) pomocou vysielacieho indexu pre aktuálny superrámec a poskytuje časovú synchronizáciu. Beacon posiela iba PNC (Piconet Coordinator) za pomoci TDMA
- konkurenčná prístupová doba CAP (Contention Access Period) počas, ktorej zariadenia prístupujú k médiu prostredníctvom CSMA/CA. Táto doba je využívaná pre autentifikáciu/spojenie požiadavky/odpovede, vysielanie parametrov pre dohodu vysielania a iné príkazy. Doba CAP je nepovinná. PNC môže nahradiť CAP s časovými slotmi riadenia MTS (Management Time Slots) použitím prístupu prostredníctvom slotted Aloha.
- časová perióda alokovaných kanálov CFP (Contention Free period), používa sa na jednosmerne garantované časové sloty GTS (Guaranteed Time Slots) pridelených PNC pre izochrónne a asynchrónne vysielanie dát, pre prenos radiacích časových slotov MTS (Management Time Slots). Ak je GTS použitý tak obsahuje MAC rámec, ktorý obsahuje zdrojovú a cieľovú adresu, identifikátor zariadenia, typ atd. Po MAC rámci nasledujú dáta. GFP používa metódu TDMA so slotmi pridelenými koordinátorom

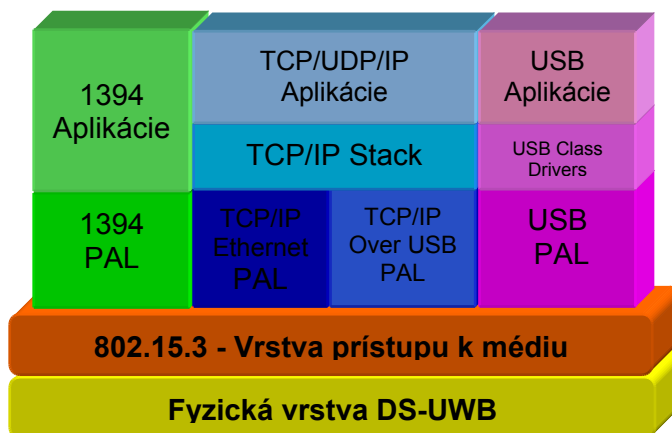


Obr. 3.6: Superrámec DS-UWB

Protokolovo adaptačná vrstva DS-UWB

Technológia od UWB fóra zahrňuje protokolovo adaptačnú vrstvu PAL (Protocol Adaptation Layer), ktorá zaisťuje konvergenciu medzi množstvom protokolov ako Cable Free USB podobne ako u technológii od WiMedia Alliances vid'. Obr. 3.7 PAL pracujúcej nad MAC a PHY vrstvou UWB a v podstate predstavuje PHY a MAC pre samotný napr.

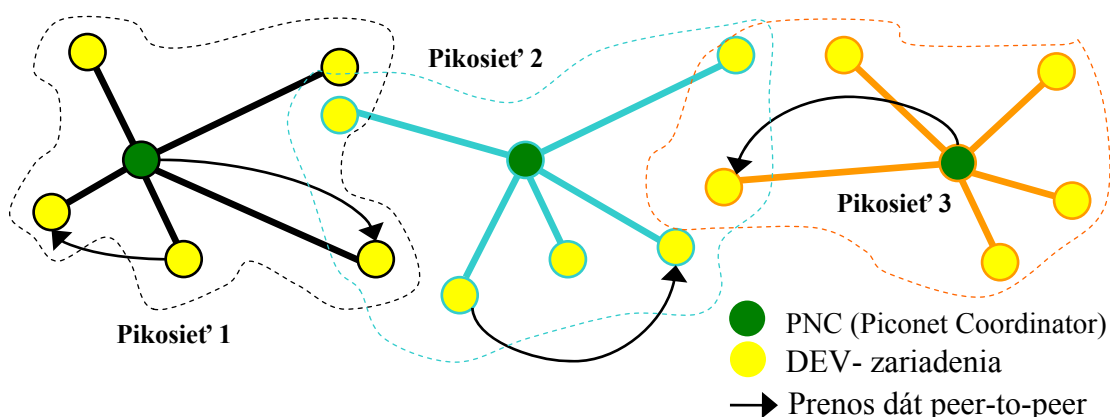
USB atď.. Vďaka tomu, že technológie nad PAL používajú rovnako definované UWB, môžu spoločne koexistovať [18].



Obr. 3.7: Vrstvový model DS-UWB

Topológia siete

Každý zariadenie v sieti sa môže stať koordinátorom pikosiete PNC (Piconet Coordinator), ale v rámci jednej pikosiete môže byť maximálne jeden koordinátor PNC vid'. Obr. 3.8. Koordinátor koordinuje sieť a umožňuje prístup prostredníctvom TDMA. V sieti môžu komunikovať zariadenia medzi sebou komunikáciou peer-to-peer v rámci TDMA, kde pakety nemusia byť smerované skrz PNC. Vedľa seba môže existovať viac susedných pikosietí (max. 6 v rámci jedného pásma), ktoré medzi sebou zdieľajú spoločný frekvenčný kanál. Každá pikosieť pracuje nezávisle na iných pikosieťach pretože každá pikosieť používa odlišné spojenie PAC s určitou číповou rýchlosťou vid'. Tab. 3.3 [18].



Obr. 3.8: Topológia siete DS-UWB

3.2 ZigBee

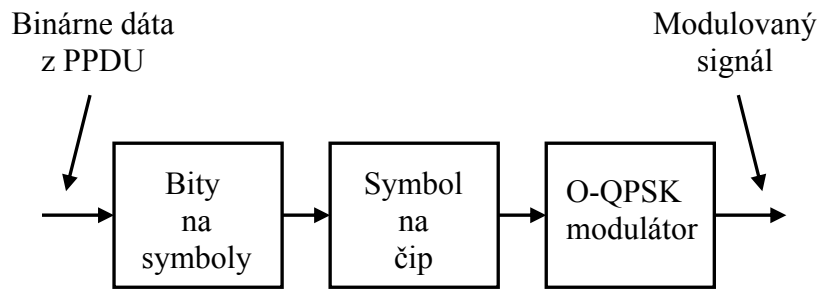
Hoci Bluetooth spotrebovávajú o dosť menej energie ako napríklad 802.11b, je to stále dosť energie, aby boli tieto zariadenia nabíjané každých pár dní. Lepšie je na tom štandard IEEE 802.15.4, ktorý definuje fyzickú (PHY) a vrstvu prístupu na médium (MAC) pre veľmi nízke výkony, pre nízko-výkonové sieťové pripojenia. Tento štandard je určený pre rozmiestnené systémy s dlhou životnosťou a nízkymi požiadavkami na prenosovú rýchlosť, kde zariadenia musia byť schopné pracovať (operovať) autonómne mesiace alebo roky bez výmeny batérie. Túto problematiku zahŕňa štandard 802.15.4, tak aj komplementárny ZigBee štandard [18].

3.2.1 802.15.4 Fyzická vrstva a vrstva prístupu k médiu

802.15.4 ponúka 27 kanálov rozložených okolo troch rôznych častí bezlicenčných pásiem. Jeden kanál je dostupný v pásme od 868,0 MHz do 868,6 MHz a je určený pre Európu. Ďalších 10 kanálov je v pásme od 902 MHz do 928 MHz s odstupom 2 MHz medzi jednotlivými kanálmi. Tieto kanály sú určené pre USA. Posledných 16 kanálov je v pásme od 2,4 GHz do 2,4835 GHz s odstupom 5 MHz medzi jednotlivými kanálmi. Toto pásmo je dostupné celosvetovo. Na rozdiel od Bluetooth, zariadenia nevyužívajú metódu frekvenčného skákania počas komunikácie. 802.15.4 vysielateľ vyžaduje na vysielanie minimálne 1mW a BER menej ako 1%. V závislosti na vysielacom výkone 802.15.4 ponúka rozsah približne od 1m do 100m, čo je porovnateľné s Bluetooth.

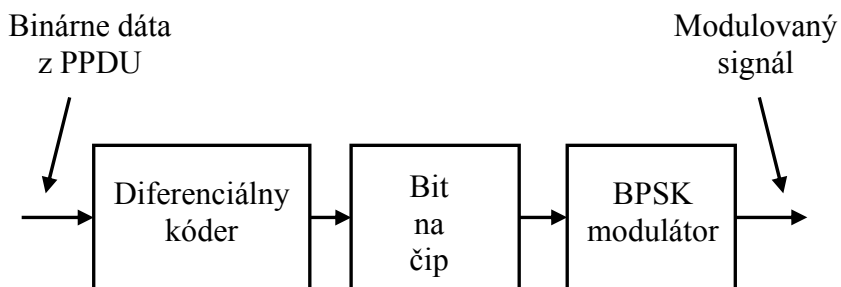
Využíva sa technika priameho rozprestretia spektra DSSS (Direct-sequence spread spectrum) pre minimalizáciu straty dát v dôsledku rušenia a interferencií, aj keď presnosť prenosu informácií sa líši pásmo od pásma [19].

V pásme 2,45 GHz majú kanály prenosovú rýchlosť 250 kbit/s čo odpovedá 62,5 kBaud (4bit/symbol). Ako dátová modulačná technika sa využíva 16-ary quasi-ortogonal založená na DSSS. Každá 4-bitová sekvencia z PPDU (PHY protocol data unit) je namapovaná na jeden zo 16 možných symbolov. Každý symbol je priamo namapovaný na jednu 32-taktovú (čipovú) ortogonálnu postupnosť (nulová vzájomná korelácia). Pre prenos sa každá taktová (čipová) sekvencia moduluje na nosnú použitím O-QPSK (Offset – Quadrature Phase Shift Keying). Pretože každý dátový symbol je reprezentovaný 32-bitovou postupnosťou, potom tzv. čipová rýchlosť je 2 Mchips/s (32 násobok symbolovej rýchlosti) vid'. Obr. 3.9 [19].



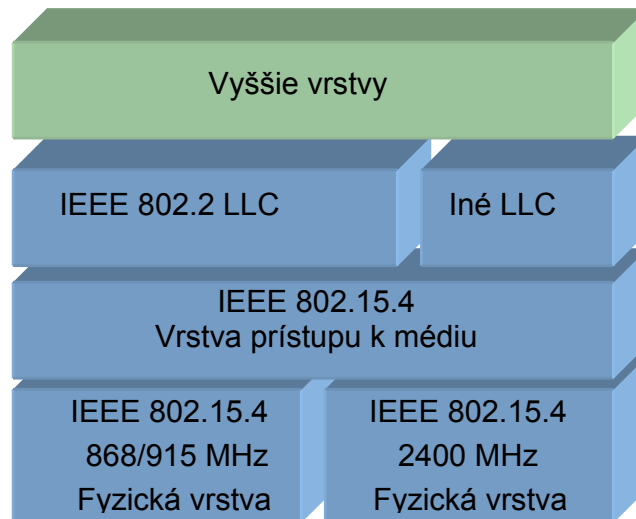
Obr. 3.9: Modulácia a rozprestieracia činnosť v pásme 2,45 MHz

V pásme 868 MHz má kanál prenosovú rýchlosť 20 kbit/s a v pásme 915 MHz má kbit/s. Obe pásma používajú rovnakú dátovú moduláciu BPSK s diferenciálnym kódovaním (DSSS), kde každý bit z PPDU je reprezentovaný 15-taktovým (čipovým) pseudonáhodným kódom (postupnosť maximálnej dĺžky: m-sequence). Každá taktová (čipová) sekvencia je modulovaná na nosnú použitím BPSK. Čipová rýchlosť je 300 kchip/s pre pásmo 868 MHz a 600 kchip/s v pásme 915 MHz vid' Obr. 3.10 [19].



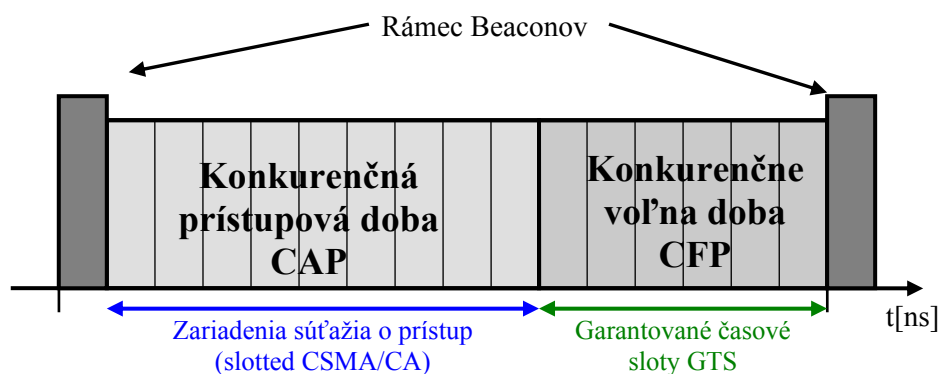
Obr. 3.10: Modulácia a rozprestieracia činnosť v pásme 868/915 MHz

Zariadenia 802.15.4 môžeme rozdeliť do dvoch kategórií, v závislosti na použitej topológii a spôsobe prístupu na médium použitej v sieti. Plne funkčné zariadenia FFD (Full Function devices) môžu komunikovať priamo s hocijakým iným zariadením. Na rozdiel od FFD zariadenia s redukovanými funkciami RFD (Reduced Function devices) môžu komunikovať jedine s FFD. Štandard 802.15.4 dovoľuje vytvárať siete s jedným preskokom (one-hop) ako je topológia hviezda (star), s viacerými preskokmi (multi-hop) peer-to-peer topológiu alebo ich kombináciu. Siete prvého typu sú vhodnejšie tam kde je málo FFD zariadení, naopak sieť druhého typu sa používa tam kde je veľa FFD zariadení, kde sa vyžaduje vyššia odolnosť proti zlyhaniu uzla. Hoci 802.15.4 definuje povolené topológie, ale nedefinuje vrstvy ktoré ich v skutočnosti podporujú. Za smerovanie v rámci týchto topológií sú zodpovedné vyššie vrstvy nad IEEE.



Obr. 3.11: Architektúra 802.15.4

Plne funkčné zariadenie (FFD) môže byť voliteľné aj ako uzol tzv. koordinátor, ktorý riadi prístup k médiu. Tento uzol periodicky vysiela beacon rámce aby určil synchronizáciu, identifikáciu PAN a popis štruktúry sepperrámca Obr. 3.12. Interval medzi beacon rámcami je konštantný, ale užívateľský voliteľný. Časť od začiatku becaon rámce do začiatku nasledujúceho beacon rámce sa nazýva superrámec, ktorý je rozdelený do 16 rovnako veľkých timeslotu. Beacon rámce môžeme oddialiť násobkom $15,38 \text{ ms} * 2^n$ ($0 \leq n \leq 14$), až na 252 s. Timesloty sú rozdelené na časť CAP (Contention Access Period) a CFP (Contention Free Period). V dobe trvania CAP prebieha komunikácia metódou CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). Člen PAN môže požiadať koordinátorsky uzol o vyhradený timesloty GTS (Guaranteed Time Slot). Vyhradené timesloty sa prenášajú v dobe CFP a sú určené predovšetkým pre pomalé a prioritné zariadenia [18][19].

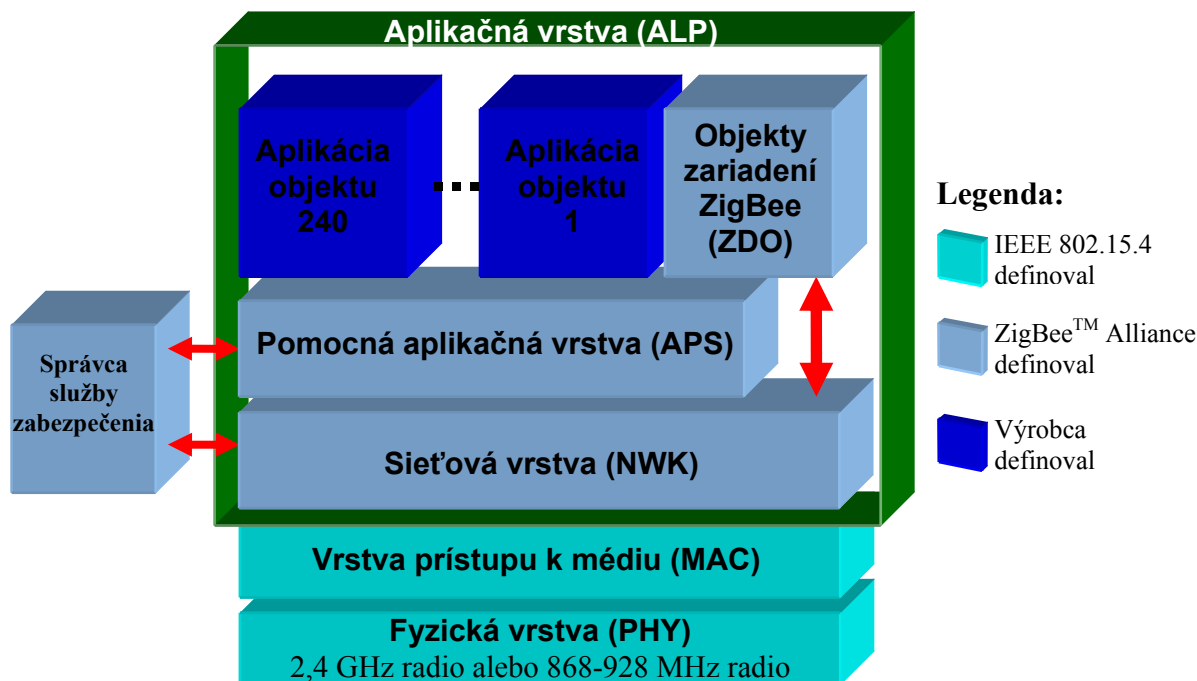


Obr. 3.12: Štruktúra superrámca s GTS

3.2.2 Rozšírenie 802.15.4 - ZigBee

ZigBee štandard rozširuje schopnosti 802.15.4 štandardu do budúcnosti. ZigBee architektúra je zobrazená na Obr. 3.13. Je postavená na štandarde 802.15.4-2003 a to na fyzickej vrstve (PHY) a vrstve prístupu na médium (MAC). Nad týmto základom vytvorila

ZigBee Alliance sieťovú (NWK) vrstvu a kostru pre aplikačnú vrstvu (APL). Kostra aplikačnej vrstvy obsahuje pomocnú aplikačnú podvrstvu (APS) a objekty zariadení ZigBee (ZDO) a s aplikačných objektov definovaných výrobcami.



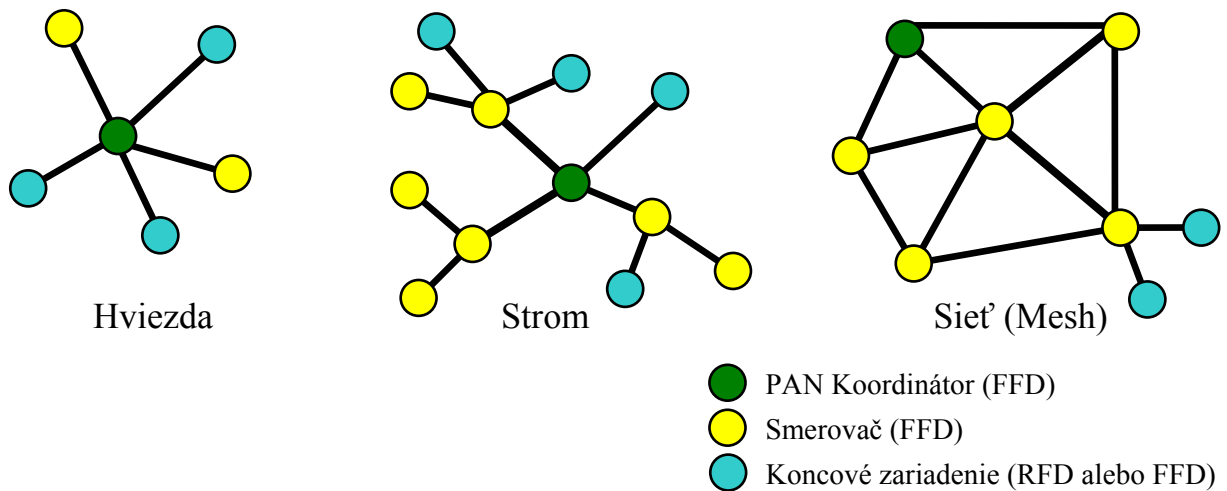
Obr. 3.13: Architektúra ZigBee

Pomocná aplikačná vrstva (APS) je určená k udržiavaniu väzobnej tabuľky, ktorá umožňuje prepojenie dvoch zariadení na základe ich potrieb a služieb a tiež preposiela správy medzi vzájomne poprepájanými zariadeniami. Objekt ZigBee (ZDO) určuje o aké zariadenie v sieti sa bude jednať (koordinátor, router alebo koncové zariadenie). Jeho úlohou je hľadať zariadenia v sieti, zaistiť poskytované služby, vytvárať alebo odpovedať na žiadosti spojení, vytvárať zabezpečené spojenia medzi zariadeniami. Úlohou sieťovej vrstvy (NWK) je zabezpečiť rámce a smerovať ich k cieľovým uzlom, hľadať priame jednoskokové (one-hop) susedné uzly a ukladať si informácie o nich. Ak sa jedna o koordinátor NWK mu umožňuje komunikáciu a pridelovanie adries novým zariadeniam [20].

Topológia siete

Sieťová vrstva podporuje sieťové topológie typu hviezda (star), strom (tree) a sieť (mesh) vid' Obr. 3.14. ZigBee bližšie špecifikovala zariadenia štandardu 802.15.4 podľa úlohy zariadení v hierarchii z dvoch na tri [20][22].

- Koordinátor (FFD) – nim je sieť riadená. Koncové zariadenia v topológii hviezda priamo komunikujú s koordinátorom. V topológii typu strom alebo sieť koordinátor spúšťa komunikáciu a určuje parametre siete.
- Smerovač (FFD) – podieľa sa na smerovaní protokolov ZigBee. Používa sa na rozšírenie siete.
- Koncové zariadenia (RFD) – tieto zariadenia môžu medzi sebou komunikovať jedine prostredníctvom koordinátora alebo smerovača.



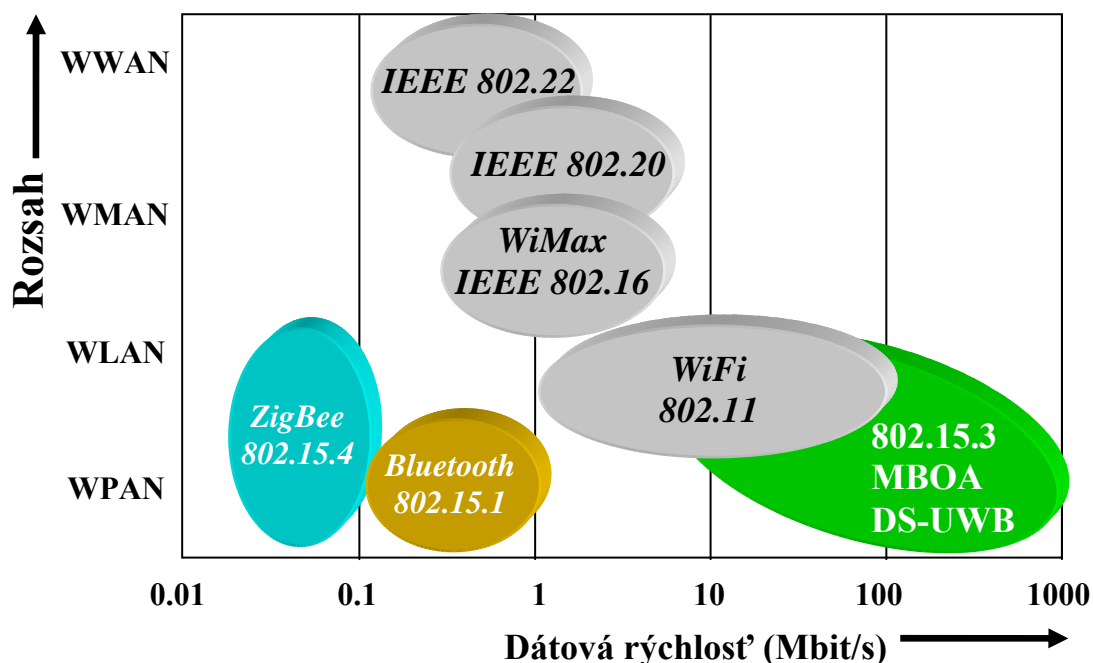
Obr. 3.14: Topológia siete ZigBee

4. Porovnanie technológií WPAN

V Tab. 4.1 a na Obr. 4.1 je vidieť rôzne technológie, s ktorými sa môžeme stretnúť pri voľbe riešenia problematiky komunikácie v osobných sieťach WPAN.

Bluetooth je špecifikácia bezdrôtovej komunikácie určená predovšetkým ako náhrada káblu pre komunikáciu na malú vzdialenosť medzi elektronickými zariadeniami v dosahu 1m až 100 m s maximálnou prenosovou rýchlosťou 3 Mbit/s. Medzi jej kľúčové vlastnosti patrí miniatúrnosť, nízka cena, robustné spojenie, krátky dosah, malý príkon, automatická konfigurácia, komunikácia bez priamej viditeľnosti a rozšírenosť.

802.15.4/ZigBee je jednoduchý štandard, ktorý ponúka menšiu prenosovú rýchlosť ako Bluetooth (max. 250 kbit/s) na porovnateľnú vzdialenosť 1m až 100 m medzi mnoho zariadeniami. Za primárnu úlohu má poskytnúť bezdrôtové spojenie so zameraním na veľmi nízku spotrebu a to tak nízku, aby zariadenie vydržalo s bežnými tužkovými batériami i niekoľko mesiacov (a dokonca i roky) v prevádzke. Táto technológia sa pre svoju nízku cenu, nízkym nárokom na hardware a nízku spotrebu najviac uplatní v oblasti riadenia budov, spotrebnej elektronike a priemysle, napríklad v podobe batériovo napájaných bezdrôtových senzorov [20].



Obr. 4.1: Graf závislosti prenosovej rýchlosti od vzdialenosti technológií WPAN

UWB je technológia pre bezdrôtové osobné siete, ktorá ponúka unikátnu kombináciu nízkej spotreby ($\sim 1\text{mW/Mbit/s}$) a vysokej dátovej priepustnosti na krátku vzdialenosť.

MBOA a DS-UWB predložili univerzálne riešenia, ktoré by využívali rôzne aplikácie na spoločnej rádiovkej platforme. Boj o podporu vyhrala WiMedia (MBOA)

vedená Intelom, keď si ich zvolilo USB-IF (USB-Implementers Forum) za "certifikované" bezdrôtové USB (Wireless USB), ale aj keď v máji 2005 Bluetooth SIG oznámil spoluprácu s WiMedia integrovať jeho UWB technológiu do budúcej verzie Bluetooth. Na to firmy Freescale Semiconductor a Motorola opustili UWB Forum. Freescale sústredilo svoje UWB úsilie na vytvorenie vlastnej verzie bezdrôtového USB s názvom Cable-Free USB [24].

Spolupráca na vývoji nového štandardu medzi Bluetooth SIG a WiMedia súčasnosti stále pokračuje. Spojením ľahkého vývoja aplikácií, rozšírenia, robustnosti, spätnej kompatibility a vysokej dátovej priepustnosti s nízkou spotrebou sa stane z technológie Bluetooth na UWB podľa môjho názoru líder pre multimédia v rámci WPAN.

Tab. 4.1: Prehľad WPAN technológií

| | Bluetooth | 802.15.4 | | UWB | |
|--------------------------------|--|---------------------------------|---|--|-----------------------------|
| | | štandard | ZigBee | MBOA | DS-UWB |
| Rádiové Spektrum | 2,4 GHz | 868 MHz, 915 MHz, 2,4 GHz | | Skup-1: 3,168-4,752 GHz Skup-2: 4,572-6,336 GHz Skup-3: 6,336-7,920 GHz Skup-4: 7,920-9,504 GHz Skup-5: 9,504-10,560 GHz | 3,1-4,85 GHz 6,2-9,7 GHz |
| Max. prenosová rýchlosť | 3 Mbit/s | 250 kbit/s | | 480Mbit/s | 1,33Gbit/s |
| Vysielací výkon | < 100 mW | > 1 mW | | 1mW/Mbps | - |
| Max. dosah | 1 m – 100 m | 1m – 100 m | | 5 m-20 m | 2 m- 4 m |
| Sieťová topológia | hviezda, scatternet | Hviezda, peer-to-peer | Hviezda, skupinový strom, sieť (Mesh) | sieť (Mesh), peer-to-peer | peer-to-peer |
| Modulácia | GFSK, 8DPSK $\frac{\pi}{4}$ -DQPSK | BPSK (868/915 MHz), O-QPSK | | QPSK, 16QAM | BPSK, 4-BOK |
| Využitie pásma | FHSS | DSSS | | OFDM, FH | DSSS |
| Využitie | náhrada káblov | Senzory, diaľkové ovládače | | Náhrada vysokorýchlostných káblov na krátku vzdialenosť (náhrada USB, FireWire) | |

5. Java 2 Micro Edition

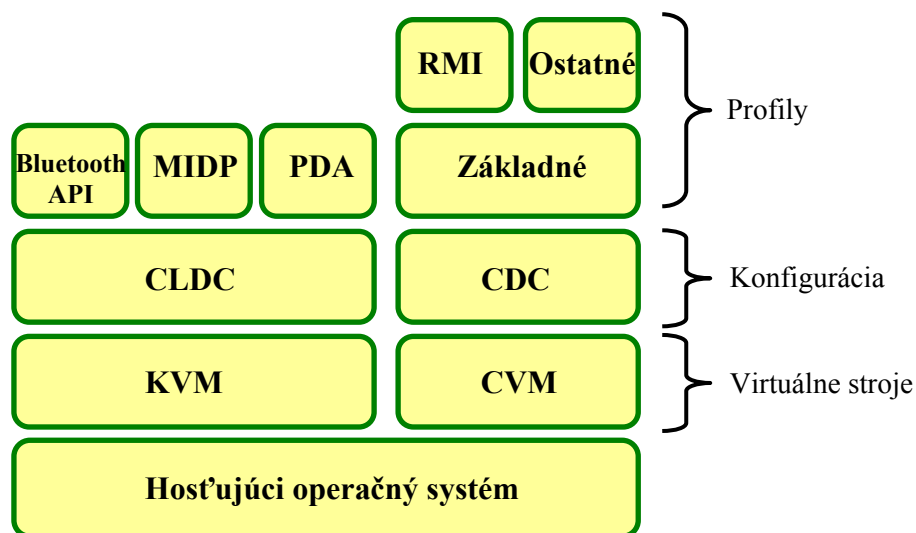
Java 2 Micro Edition (J2ME) je verzia produktu Java spoločnosti Sun Microsystems určená pre zariadenia, akými sú napríklad mobilné telefóny, pagery, osobný digitálny asistenti (PDA), prídavné zariadenia a ďalšie malé zariadenia. Vývoj J2ME zaisťuje projekt JCP (Java Community Process), ktorý umožňuje komukoľvek podieľať sa na vývoji [1].

Výhodou Javy je prenositeľnosť kódu medzi zariadeniami a objektovo orientované programovanie. J2ME musela byť prispôbená pre malé zariadenia. Obsahuje len na určité balíčky zo štandardnej Javy a boli pridané nové balíčky, triedy a metódy riešiacie ich limity ako sú obmedzené výpočtové vlastnosti, malá operačná pamäť, nízke rozlíšenie displeja a ďalšie.

5.1 Architektúra J2ME

Architektúra definuje tieto komponenty vid'. Obr. 5.1:

- Java virtuálne stroje pre rôzne zariadenia s odlišnými nárokmi
- skupina knižníc a API, spustiteľné na každom virtuálnom stroji, nazývaných konfigurácie a profily
- rôzne nástroje pre vývoj a nastavenie zariadení



Obr. 5.1: Architektúra J2ME

Veľmi dôležité je rozlišovať medzi konfiguráciou a profilom, nasleduje vysvetlenie pojmov.

5.1.1 Konfigurácia

Zariadenia ako mobilné telefóny, pagery, organizéry sa líšia svojimi funkciami a vlastnosťami, ale cez toto však používajú podobné procesy a majú podobné množstvo pamäti. Preto vznikli od autorov J2ME konfigurácie. Konfigurácia definuje horizontálne členenie produktu založené na dostupnom množstve pamäti a výkonu procesoru zariadenia. Z týchto informácií konfigurácia zisti podporované rysy jazyka Java, virtuálneho stroja a podporované Java knižnice a API. Existujú dve základné konfigurácie

- CLDC (Connected Limited Device Configuration)
- CDC (Connected Device Configuration)

CDC

Konfigurácia CDC (štandard JSR-36, JSR-218) je určená pre výkonnejšie zariadenia s možnosťami pripojenia k sieti. Patria tu zariadenia ako navigačné systémy pre vozidlá, domáce spotrebiče, prídavné zariadenia a ďalšie. CDC používa plnú verziu virtuálneho stroja Javy podobného tomu, ktorý sa používa u J2SE (Java 2 Standard Edition). Rozdiel je v pamäti a zobrazovacích schopnosti príslušného zariadenia. Požiadavky na CDC zariadení podľa oficiálnej špecifikácie J2ME sú [1]:

- Zariadenie riadené 32-bitovým procesorom,
- minimálne 2MB pamäti (RAM, ROM/flash) pre Javu,
- vyžadovaný plne funkčný virtuálny stroj Java 2 „Blue Book“,
- pripojenie k nejakému typu siete,
- užívateľské rozhranie s rôznym stupňom prepracovanosti, ale nie je to povinné.

CLDC

Obvyklejšou konfiguráciou pre J2ME je CLDC, ktorá kladie na zariadenia omnoho menšie požiadavky ako CDC. Je určená pre zariadenia ako sú PDA, osobné organizátory, mobilné telefóny a ďalšie. Požiadavky pre J2ME CLDC podľa oficiálnej špecifikácie J2ME sú [1]:

- 160 – 512 KB celkovej pamäti pre prostredie Java vrátane RAM a pamäti ROM/flash,
- obmedzené zdroje energie, často batériové napájanie,
- pripojenie k niektorému typu siete, často s obmedzenou šírkou pásma (často 9600 b/s a menej),
- rôzne prepracované užívateľské rozhranie, ale nie je to povinné.

5.1.2 Virtuálne stroje

Ako už bolo zmienené, každá konfigurácia stanovuje svoju vlastnú skupinu podporovaných rysou Java virtuálneho stroja. To znamená, že pre každú konfiguráciu je

potrebný iný Java virtuálny stroj. Virtuálny stroj pre CDLC je KVM (Kilo Virtual Machine) a virtuálny stroj pre CDC sa nazýva CVM (Compact Virtual Machine).

KVM

KVM je kompletne javovské pracovné prostredie špeciálne vytvorené pre malé zariadenia s obmedzenými zdrojmi a s niekoľko stovkami kilobajtov celkovej pamäti. Cez to sa jedná o skutočný virtuálny stroj podľa špecifikácie pre virtuálne stroje. Obsahuje špecifické odchýlky, nutné pre k správne fungovaniu na malých zariadeniach. Správa o J2ME, ktorú môžete nájsť na [30], popisuje KVM takto [1]:

- Vytvorené pre 16-bitové a 32-bitové CISC a RISC procesory a procesory s minimálnou rýchlosťou 25 MHz,
- nároky na statickú pamäť od 50 do 80 KB,
- vysoko mobilné s ustálenými rozmermi a prispôsobivé,
- za daných podmienok čo možno najrýchlejší a kompletní bez nutnosti obetovať iné, vyššie spomenuté vlastnosti dizajnu

CVM

CVM je virtuálny stroj vytvorený pre väčšie zariadenia, napríklad tie, ktoré používajú konfiguráciu CDC. Podporuje všetky rysy JVM (Java Virtual machine).

5.1.3 Profily

Profil je v podstate API (sadá programových rozhraní) tvoriace nastavbu konfigurácie. Poskytuje program prístup k vlastnostiam špecifických pre dané zariadenie. Nasleduje prehľad niekoľkých profilov pre obe konfigurácie (CDC, CDLC). Profilov je o dosť viac, ale uvádzam len tie základné.

Základný profil

Základný profil (Foundation Profile) rozširuje programové rozhranie, ktoré poskytuje CDC a slúži ako základ pre ďalšie profily. Základný profil nedodáva žiadne API pre užívateľské rozhranie, pretože zariadenia pre ktoré je CDC určené (set-top boxy, TV apod.) nie sú na GUI tak fixované.

Osobný profil

Osobný profil (Personal Profile) je rozšírenie Základného profilu o grafické užívateľské rozhranie. Jedná sa o kompletne prostredie s plnou podporou AWT (Abstract Window Toolkit). Tento profil je závislý na základnom profile.

RMI profil

RMI profil (Remote Method Invocation) je ďalšie rozšírenie Základného profilu. Poskytuje prostredie pre vzdialené volanie metód Javy kompatibilných s RMI API zo štandardnej edície J2SE [1].

PDA profil

PDA profil je založený na CLDC a definuje užívateľské rozhranie (podskupina AWT) a API pre ukladanie dát v zariadeniach.

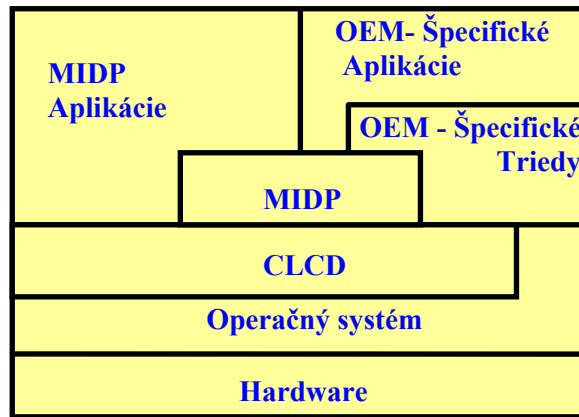
MIDP

MIDP (Mobile Information Device Profile) je profil poskytujúci API v zariadeniach ako je mobilný telefón, teda zariadeniam s konfiguráciou CLCD. Obsahuje triedy pre užívateľské rozhranie, trvalé ukladanie dát a prácu v sieti. Aplikáciám bežiacich pod touto kategóriou sa hovorí MIDlet podľa základnej triedy profilu MIDP, ktorá existuje v niekoľkých verziách od MIDP 1.0 cez MIDP 2.0 až po najnovší MIDP 2.1. Zariadenie s MIDP musí spĺňať nasledujúce parametre :

- Veľkosť displeja minimálne 96 x 54 pixlov, dve farby,
- klávesnica poprípade dotyková obrazovka,
- 32 KB pamäť pre prácu Javy, 128 KB stálej pamäti pre komponenty MIDP a 8 KB stálej pamäti pre dlhodobé ukladanie dát aplikáciou,
- možnosť obojsmerného sieťového spojenia.

Profil MIDP je vrcholom CDLC vid'. Obr. 5.2 a zameriava sa na nasledujúce v CDLC neriešené oblasti [29]:

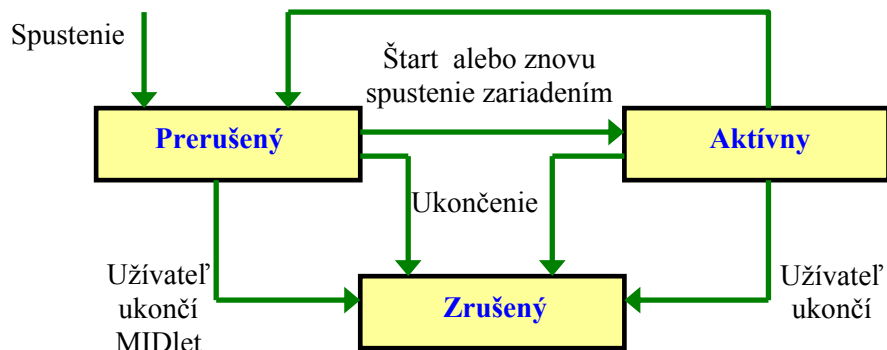
- Správa priebehu aplikácii – balíček *javax.microedition.midlet*,
- užívateľské rozhranie a udalosti – balíček *javax.microedition.lcdu* určený k tvorbe komponent grafického užívateľského rozhrania v aplikácii,
- pripojiteľnosť k sieti – rozširuje rozhranie *ContentConnection* obecného pripojovacieho systému (Generic Connection Framework) tím, že poskytuje rozhranie *httpConnection*,
- ukladanie dát v zariadení – balíček *javax.microedition.rms*, zavádza databázový záznamový systém.



Obr. 5.2: Architektúra MIDP

V architektúre MIDP nad CLCD a MIDP sa nachádzajú takzvané OEM – Špecifické triedy a aplikácie, ktoré sú definované výrobcom zariadení. Sú určené pre využitie špeciálnych vlastností zariadenia. Pri použití týchto tried dôjde k obmedzeniu prenositeľnosti aplikácie.

Java aplikácie pre MIDP sú MIDlety, ktoré sa tvoria ako jedna alebo viac tried Javy, ich objekty sú komprimované do JAR súboru. Obdobne ako je to u javovských apletov tak aj MIDP aplikácie majú svoj priebeh, pracujúci na mobilných zariadeniach. MIDlet sa môže nachádzať v troch stavoch vid'. Obr. 5.3.



Obr. 5.3: Prechody medzi stavmi MIDletu

Keď je MIDlet prvý krát spustený, nachádza sa v stave prerušený. Keď je pripravený, kontrolný program ho privedie do stavu aktívny. V tejto chvíli MIDlet beží a užívateľ s ním môže pracovať. Aplikácia môže byť znova prerušená buď systémom MIDP, alebo sám program. Navyše je možné MIDP presunúť do stavu vymazania, a to z oboch stavov. Do stavu vymazania ho presunie buď systém MIDP, alebo programátor. V stave zrušený by mal MIDlet uvoľniť systému MIDP všetky zdroje, ktoré v tu chvíľu používa [29].

Špecifikácia MIDP 2.0 pri, ktorej vývoji sa podieľalo i niekoľko výrobcov mobilných telefónov je predovšetkým určená práve pre mobilné telefóny. Špecifikácia

rozširuje knižnice o multimedialne API (prehrávanie tónov, audio súborov, videa), sieťové rozšírenie (sockety), SMS podpora a mnoho ďalších. Obsahuje teda skoro dvakrát viac tried ako MIDP 1.0 a z toho dôvodu sa zvýšili aj nároky na mobilné zariadenia:

- 256 kB RAM navyše nad požiadavky CLDC,
- 128 kB pre beh JRE (Java Runtime Environment).

Java API pre Bluetooth

Bluetooth je rozširujúcim prostriedkom pre bezdrôtové prepojenie malých zariadení ako mobilné telefóny, pagery, atď.

Aplikačné rozhranie jazyka Java je definované a spravované skupinou JCP (Java Community Process) . Každé novo vzniknuté rozhranie sa nazýva JSR (Java Specification Request). Triedy Javy, ktoré podporujú technológiu Bluetooth sa nazývajú JABWT (Java APIs for Bluetooth Wireless Technology) a teda aplikačné rozhranie Javy pre Bluetooth. JSR pre toto aplikačné rozhranie má označenie JSR-82 a je určené pre konfiguráciu CLDC a obsahuje podporu základných protokolov. Najhlavnejší profil je základný profil GAP (Generic Access Profile), ktorý zastrešuje viď. Obr. 2.4:

- Profil emulácie sériovej linky SPP (Serial Port Profile) – využíva protokol RFCOMM poskytujúci emuláciu sériového portu RS-232 pomocou L2CAP,
- profil pre vyhľadávanie služieb servera SDAP (Service Discovery Application Profile) – využíva protokol SDP (Service Discovery Protocol)
- profil na prenos súborov GOEP (Generic Object Exchange Profile) – využíva protokol OBEX (Object Exchange Protocol).

Nároky na zariadenie s týmito API sú:

- Minimálna 512 KB celkovej pamäti bez pamäťových nárokov samotnej aplikácie,
- hardware Bluetooth pre sieťové pripojenie,
- vyhovujúcu implementáciu CLDC alebo CDC.

JSR-82 aktuálne špecifikuje dva nezávislé voliteľné balíčky a to:

- Java API pre Bluetooth (*javax.bluetooth*)
- Java API pre OBEX (*javax.obex*)

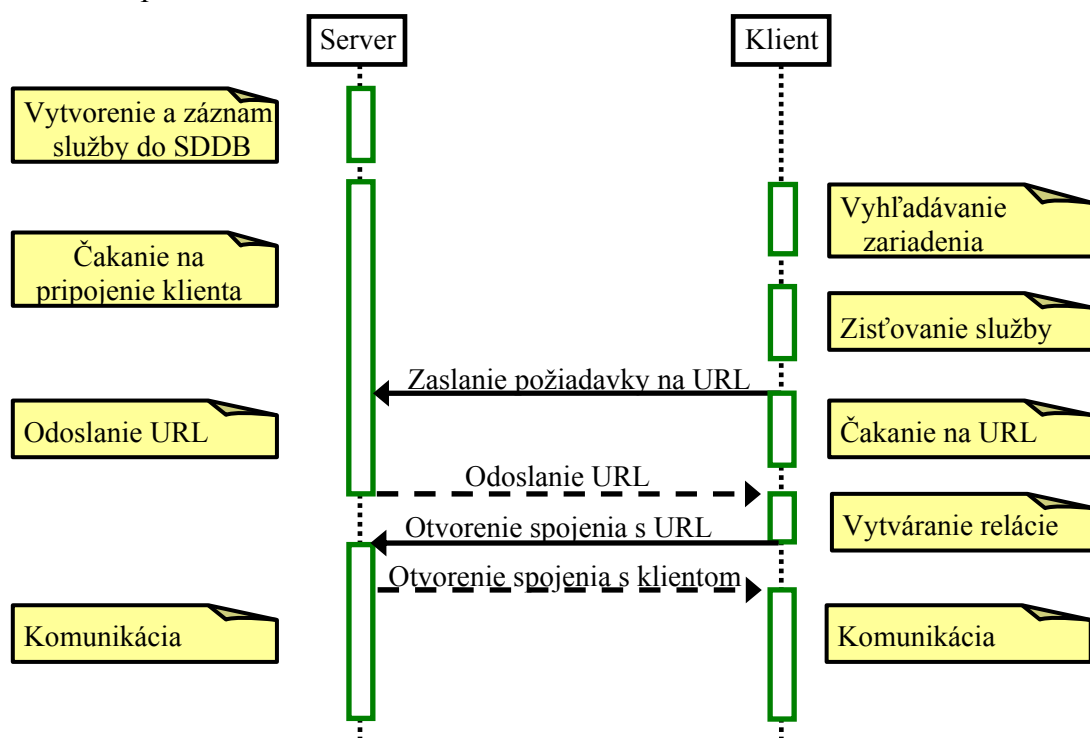
6. Návrh systému

Požiadavky na univerzálnu komunikačnú platformu boli kladené na možnosť komunikácie a výmenu súborov medzi zariadeniami ako sú mobilne telefóny a to bezdrôtovo s využitím technológie Bluetooth. Systém by mal umožňovať komunikáciu medzi viacerými zariadeniami.

6.1 Návrh funkčnosti

Aplikácie využívajúce pre komunikáciu Bluetooth môžu byť buď server (poskytovateľ služieb) alebo klient (užívateľ služieb). Medzi zariadeniami môže fungovať aj skutočná peer-to-peer komunikácia ak by aplikácia umožňovala koncovému bodu sa stať buď server alebo klient podľa potreby.

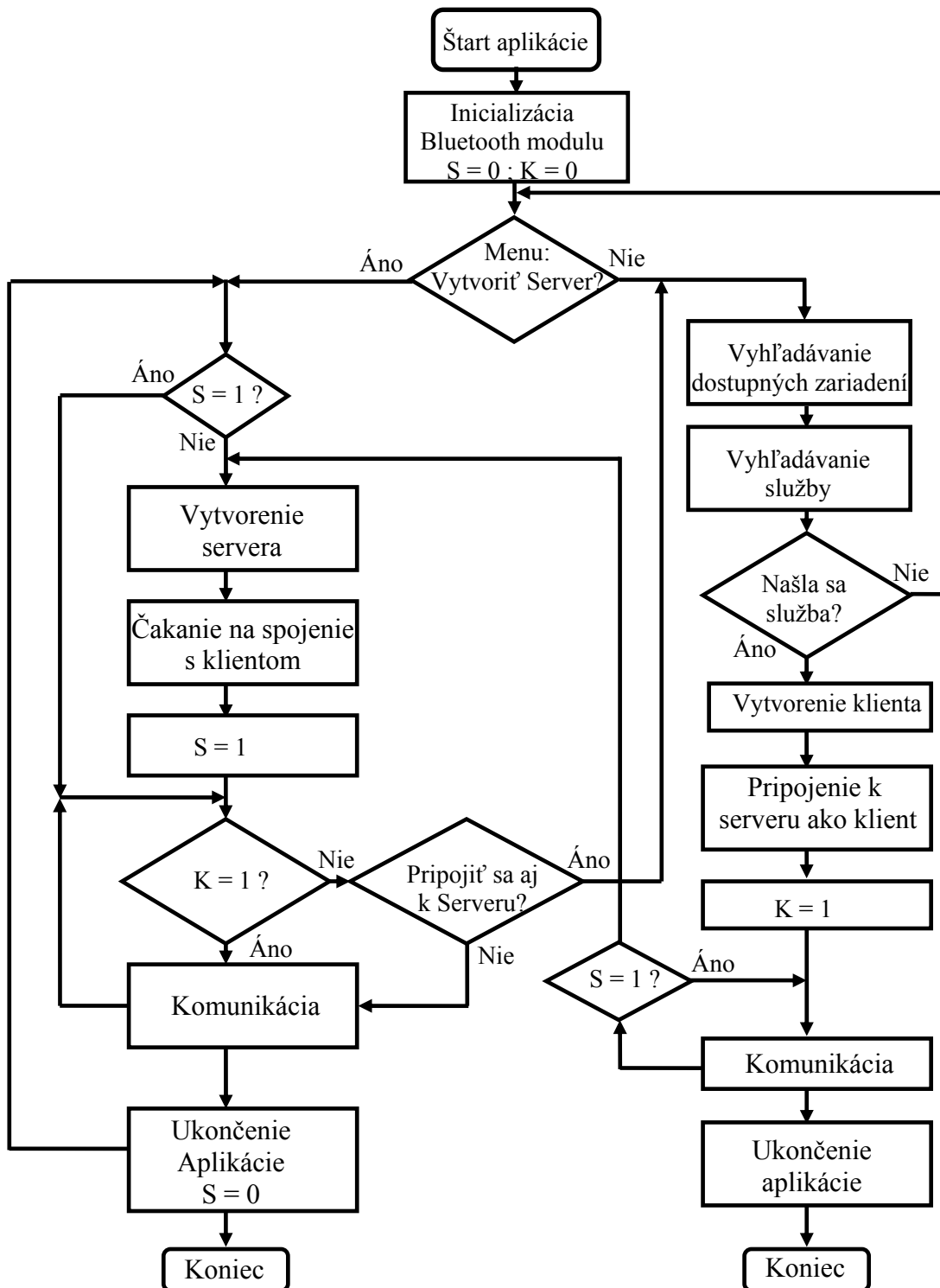
Mnou navrhnutá aplikácia je založená na peer-to-peer komunikácii. Po spustení aplikácie sa zobrazí užívateľské menu v ktorom si užívateľ môže zvoliť či sa chce pokúsiť pripojiť k serveru alebo chce rovno vytvoriť server. Pri zvolení možnosti pripojiť sa k serveru sa zaháji vyhľadávanie dostupných zariadení v okolí so zapnutým modulom Bluetooth. Po vyhľadanej zariadení sa na týchto zariadeniach začne vyhľadávať požadovaná spustená služba.



Obr. 6.1: Schéma zostavenia spojenia

Pred vyhľadávaním je potrebná inicializácia Bluetooth modulu. Ak sa v okolí nájde zariadenie s požadovanou službou a teda server, užívateľovi sa ponúkne možnosť pripojiť sa k nemu ako klient. Po pripojení k serveru ako klient vid' Obr. 6.1, aplikácia vytvorí na

pozadí serverovú časť aby umožnila pripojenie do komunikácie ďalšiemu zariadeniu ako klient. Pripojený klient môže komunikovať prostredníctvom svojho servera s jeho serverom. Prostredník plní funkciu klienta ako aj funkciu servera.



Obr. 6.2: Spôsob vytvorenia klienta alebo serveru

Ak v okolí nenájde žiadané zariadenie s požadovanou službou tak sa užívateľovi ponúkne možnosť vytvorí server (master) a teda vytvorí a zaznamená službu do databázy ponúkaných služieb SDDB (Service Discovery Database) a tým ich sprístupni okolitým

klientom. Po zverejnení bude čakať na prichádzajúce spojenia od klientov, za účelom vytvorenia spojenia. Vývojový diagram navrhutej aplikácie je na Obr. 6.2.

Akonáhle sa klientske zariadenie pripojí k serveru, dôjde k možnosti komunikácie medzi zariadeniami. Počas komunikácie s klientom má aplikácia stále možnosť vytvoriť spojenie so vzdialeným serverom ako klient a teda komunikovať ako klient a ako server súčasne. Pri tejto komunikácii bude zariadenie s aplikáciou súčasne riadiacim (master) a klientskym (slave) zariadením. Pri komunikácii viacerých zariadení vznikne rozprestrená sieť (scatternet) vid'. Obr. 2.3.

Po ukončení komunikácie zo serverovou časťou aplikácie sa relácia s daným klientom ukončí ale server stále beží a umožňuje naviazanie spojenia s iným klientom pre vytvorenie ďalšej komunikácie. Ukončenie komunikácie s klientskou časťou aplikácie ukončí komunikáciu s daným serverom, ale na pozadí bude stále aktívna serverová časť aplikácie ale aj možnosť znova sa pripojiť k ďalšiemu serveru. Aplikácia umožňuje komunikáciu medzi viacerými zariadeniami prostredníctvom vlastne navrhnutého protokolu.

Najdôležitejšie funkcie aplikácie sú vo vytvorení spojenia, udržaní spojenia s možnosťou výmeny komunikácie a ukončenie spojenia. Aplikácia vykonáva činnosť v poradí:

- Spustenie aplikácie,
- inicializácia Bluetooth modulu,
- naviazanie spojenia s druhým zariadením,
- komunikácia,
- možnosť pripojenia ďalšieho zariadenia,
- uzatvorenie spojenia,
- ukončenie aplikácie.

7. Realizácia univerzálnej komunikačnej platformy

V rámci tejto diplomovej práce bola vytvorená aplikácia určená pre komunikáciu medzi mobilnými telefónmi ako výmeny textových správ ako aj výmeny súborov. Táto kapitola bude popisovať jednotlivé hľadiská a princípy tvorby tejto aplikácie. Jedna sa teda o implementačný popis a teda ako boli vlastnosti aplikácie realizované.

7.1 Aspekt tvorby aplikácie

Pri tvorbe takejto aplikácie je nutné zvoliť pre akú cieľovú skupinu užívateľoch bude aplikácia určená, respektíve pre aké typy mobilných telefónov. Obecná Java sa snaží vo svojich edíciách mať jednotnú špecifikáciu, ale z dôvodov technických odlišností u mobilných zariadení edícia J2ME určuje špecifikáciu konfiguráciou a profilom.

Keďže diplomová práca je zameraná pre mobilné telefóny, ktoré obsahujú technológiu Bluetooth, je voľba konfigurácie jednoduchá a teda CLCD. Na úrovni CLCD existuje technické rozlíšenie a to na CLCD verzii 1.0 a verzii 1.1. Základným aspektom pri voľbe verzii je rozdiel medzi verziami, kde verzia 1.1 na rozdiel od 1.0 pracuje už plávajúcou rádovou čiarkou. Vytvorená aplikácia nevyžaduje prácu s plávajúcou čiarkou a teda konfigurácia je v súlade CLCD 1.0.

Dôležitejšie rozhodnutie je na úrovni profilu. Jedným z princípov MIDP je predstava, že môžete napísať jednu aplikáciu, ktorá funguje na mnohých rôznych typoch zariadení. Mobilne telefóny využívajú profily MIDP 1.0, MIDP 2.0 a MIDP 2.1. Keďže vytvorená aplikácia nevyžaduje použitie vlastnosti, ktoré ponúka MIDP 2.0 respektíve 2.1 je voľba profilu MIDP 1.0 vhodnejšia. Tým sa zaručí väčšia prenositeľnosť medzi rôznymi mobilnými telefónmi.

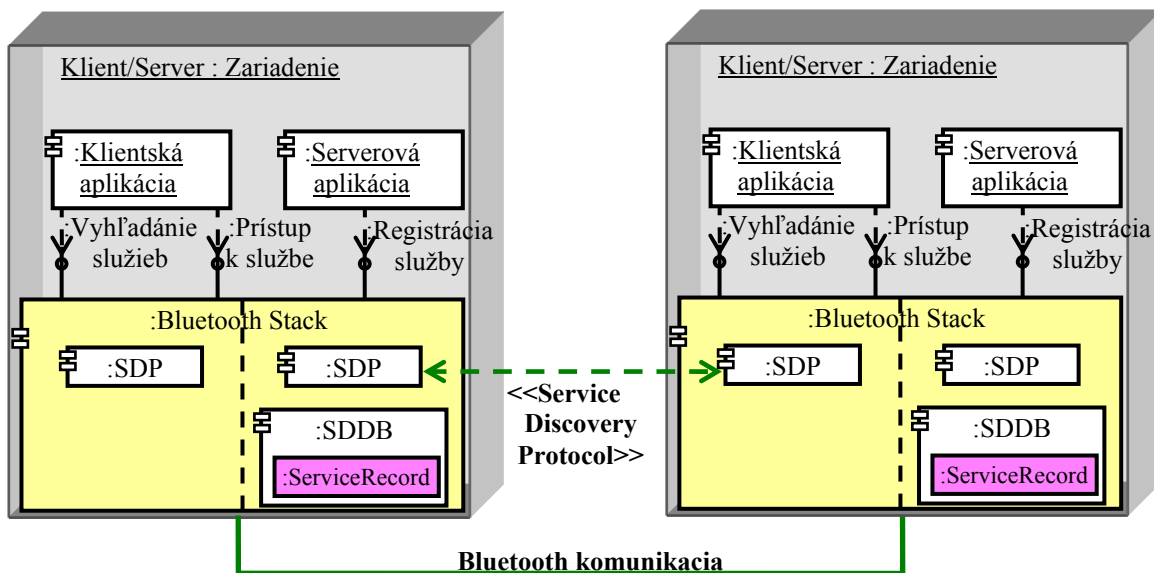
Pokiaľ je mobilná aplikácia určená pre CLCD 1.0 a MIDP 1.0 a s nešpecifikovaným rozlíšením displeja, jedná sa o takzvanú generickú aplikáciu (z pohľadu konfigurácie, profilu a zobrazenia je aplikácia prenositeľná na ktorýkoľvek mobilný telefón). Tieto obmedzenia sú najobecnejšieho charakteru.

Aplikácia teda bola vytvorená vo voľne šíriteľnom open-source vývojovom prostredí NetBeans 6.5.1 [31] na platforme J2ME s použitím CLCD 1.0, MIDP 1.0 a Java API pre Bluetooth (súčasť J2ME). Aplikácia zaisťuje komunikáciu dvoch a viac mobilných zariadení medzi sebou prostredníctvom technológie Bluetooth. Komunikácia spočíva vo výmene textových správ a dátových súborov medzi jednotlivými zariadeniami.

Pre vývoj bol použitý emulátor mobilného telefónu obsiahnutý v Java Wireless Toolkit 2.5.2, ktorý je súčasťou vývojového prostredia. Emulátor predstavuje obecný telefón, ktorý sa riadi iba podľa nadefinovaných profilov a použitých API.

7.2 Funkčnosť platformy

Aplikácia je naprogramovaná tak, že obsahuje klientsku aj serverovú časť vid'. Obr. 7.1. Zariadenie, ktoré má spustenú túto aplikáciu môže byť súčasne server a súčasne klient a komunikovať s oboma stranami súčasne tak ako to bolo popísané v návrhu (vid'. kapitola 6.1. *Návrh funkčnosti*).



Obr. 7.1: Spôsob realizácia aplikácie, kde serverová časť aplikácie poskytne službu klientskej, ktorá mu umožní pripojiť sa.

Aplikácia umožňuje komunikáciu prostredníctvom výmeny súborov ako aj písomnú konverzáciu dvoch alebo viacerých ľudí v reálnom čase prostredníctvom Bluetooth spojenia. Aplikácia využíva triedy Java API pre Bluetooth (vid'. kapitola 5.1.3 *Java API pre Bluetooth*) pre nastavenie protokolov potrebných pre komunikáciu medzi zariadeniami. Pre vyhľadávanie bol využitý protokol SDP (Service Discovery Protocol), pre výmenu textových správ protokol RFCOMM a pre výmenu súborov bol využitý protokol OBEX (Object EXchange).

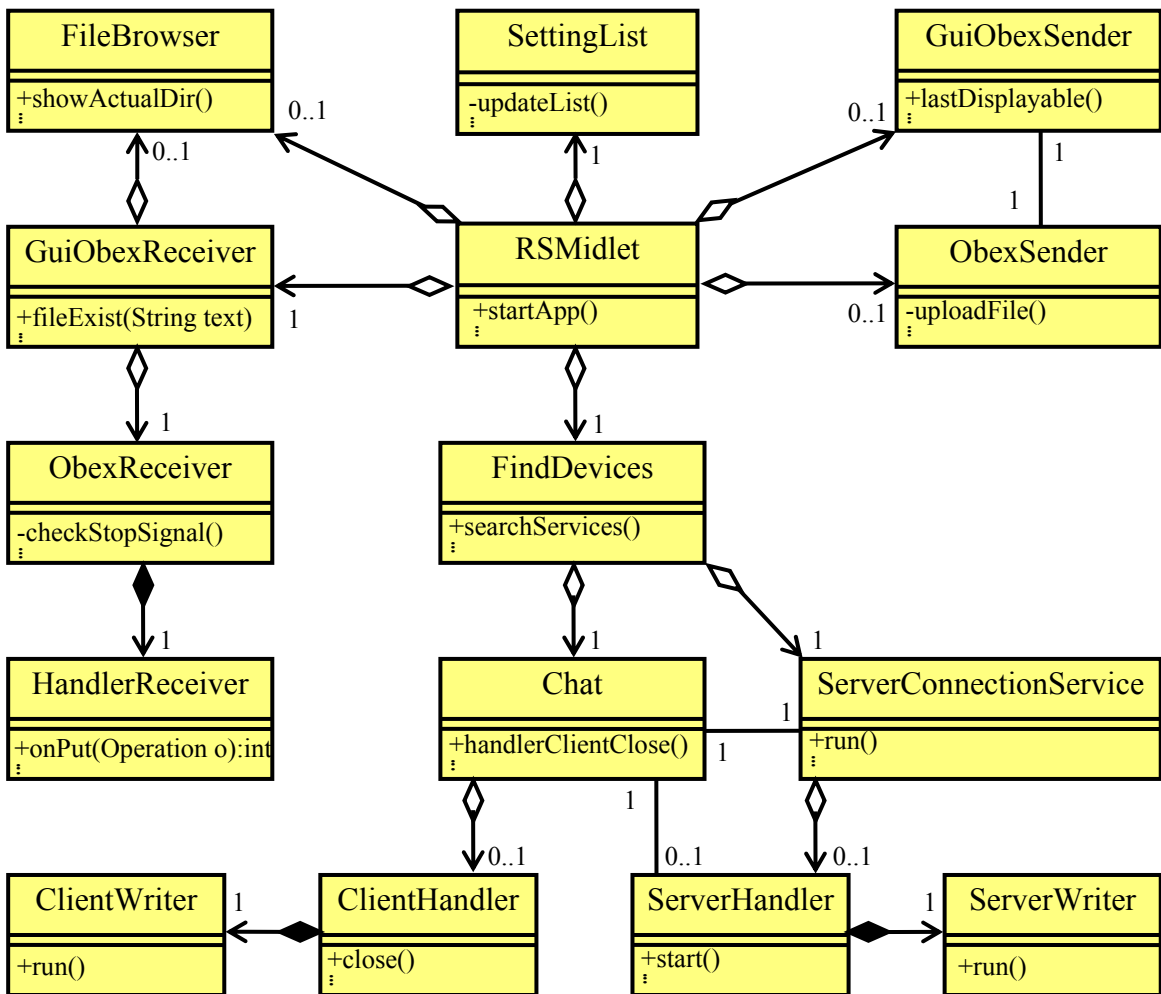
Vytvorená aplikácia pozostáva z niekoľkých tried zobrazených v UML diagrame tried Obr. 7.2. Po spustení aplikácie s názvom „RSBluetooth“ je zavolaná hlavná spúšťacia trieda RSMidlet. Nasleduje výpis použitých a mnou naprogramovaných tried so základným popisom funkčnosti Tab. 7.1.

Tab. 7.1: Prehľad tried vytvorenej aplikácie.

| Názov triedy | Základný popis funkčnosti |
|--------------|--|
| RSMidlet | Dedí abstraktnú triedu MIDlet a teda stará sa o životný cyklus aplikácie. Keďže vytvára väčšinu instancií obsahuje dôležité atribúty ako sú adresy pre pripojenie klienta alebo vytvorenie servera (vid'. kapitola 7.3 <i>Trieda RSMidlet</i>). |

| | |
|-------------------------|--|
| SettingList | Ukladá poprípade vyčítava údaj o mene používanom pri čete poprípade odoslaniu súboru a umožňuje nastavenie použitia bezpečnosti komunikácie a to autentifikáciu, šifrovanie a autorizáciu (vid'. kapitola 7.3.1 <i>Trieda SettingList</i>). |
| FindDevices | Umožňuje vytvoriť server alebo vyhľadať server v okolí a následne sa k nemu pripojiť ako klient (vid'. kapitola 7.3.1 <i>Trieda FindDevices</i>). |
| Chat | Spracováva prijaté správy, vytvára správy na odoslanie, taktiež zobrazuje spracované správy a informuje užívateľa o udalostiach (vid'. kapitola 7.3.1 <i>Trieda Chat</i>). |
| ServerConnectionService | Úlohou je vytvoriť a zverejniť služby, sprístupniť ich vzdialenými zariadeniami a čakať na prichádzajúce klientske spojenie. (vid'. kapitola 7.3.1 <i>Trieda ServerConnectionService</i>). |
| ServerHandler | Hlavnou úlohou je prijímanie textových správ s pripojeným klientom. (vid'. kapitola 7.3.1 <i>Triedy ServerHandler a ClientHandler</i>). |
| ServerWriter | Vnorená trieda v triede ServerHandler, ktorá sa stará o odosielanie správ z fronty pre pripojené klientske zariadenie (vid'. kapitola 7.3.1 <i>Triedy ServerHandler a ClientHandler</i>). |
| ClientHandler | Hlavnou úlohou je prijímanie textových správ s pripojeným serverom. (vid'. kapitola 7.3.1 <i>Triedy ServerHandler a ClientHandler</i>). |
| ClientWriter | Vnorená trieda v triede ClientHandler, ktorá sa stará o odosielanie správ z fronty pre pripojené serverové zariadenie (vid'. kapitola 7.3.1 <i>Triedy ServerHandler a ClientHandler</i>). |
| FileBrowser | Úlohou je sprístupnenie a zobrazenie súborového systému (vid'. kapitola 7.3.2 <i>Trieda FileBrowser</i>). |
| GuiObexSender | Zaisťuje grafické užívateľské rozhranie pri odosielaní súboru serveru (vid'. kapitola 7.3.2 <i>Triedy GuiObexSender a ObexSender</i>). |
| ObexSender | Úlohou triedy je odosielanie súboru na server (kapitola 7.3.2 <i>Triedy GuiObexSender a ObexSender</i>). |
| GuiObexReceiver | Zaisťuje grafické užívateľské rozhranie pri prijímaní súboru od klienta. (vid'. kapitola 7.3.2 <i>Triedy GuiObexReceiver a ObexReceiver</i>). |

| | |
|-----------------|--|
| ObexReceiver | Úlohou triedy je prijímanie súboru od klienta (viď. kapitola 7.3.2 <i>Triedy GuiObexReceiver a ObexReceiver</i>). |
| HandlerReceiver | Je to vnorená trieda v triede ObexReceiver, ktorá slúži na oznamovanie prichádzajúceho spojenia pre prijatie súboru zo strany klienta (viď. kapitola 7.3.2 <i>Triedy GuiObexReceiver a ObexReceiver</i>). |



Obr. 7.2: UML diagram tried

7.3 Spustenie aplikácie a úvodné menu

Užívateľské prostredie aplikácie je jednoduché a je vytvorené pomocou niekoľkých formulárov a listov. Grafické rozhranie je tvorené komponentmi balíčka *javax.microedition.lcdui*. Jedná sa napríklad o abstraktnú triedu *Screen* a jej podtriedy *Form*, *List*, *TextBox*, *Alert*. Ďalej sú tu odvodené prvky od triedy *Item* ako je *TextField*, *Gauge* a ovládacie prvky odvodené od triedy *Command*. Trieda *Screen* poskytuje vysokoúrovňové zobrazovacie komponenty, ktoré sú napríklad nezávislé na typu a veľkosti

displeja, a tým je zaistená prenositeľnosť medzi rôznymi mobilnými telefónmi a okrem toho aj ľahšiu modifikáciu pri možnostiach budúceho rozšírenia. Použité komponenty sú:

- **Form** – vytvára komunikačné okno so správami a okno pre zadanie mena pre textovú komunikáciu (Obr. 7.6.b),
- **List** – slúži k vytvoreniu úvodného menu ako aj menu nastavení, zoznamu vyhľadaných zariadení v okolí a k vytvoreniu prehľadávača súborov. V liste je využitá možnosť vkladania ikoniek k jednotlivým položkám o veľkosti 16 x 16 pixlov (Obr. 7.3.a),
- **Alert** – používa sa pre informovanie o výsledku uskutočnenej akcie (správnosť voľby súboru/adresára), upozornenie na prichádzajúce súbory a stav odoslania/príjmu súboru (Obr. 7.8.a).

Trieda RSMidlet

Po spustení aplikácie RSBluetooth je zavolaná hlavná spúšťacia trieda RSMidlet, ktorá dedí abstraktnú triedu MIDlet. To je povinnosťou každej mobilnej aplikácie (viď. kapitola 5.1.3 *MIDP*). Tato trieda implementuje povinné metódy `startApp()`, `pauseApp()` a `destroyApp()`, a teda stará sa o životný cyklus midletu. Hlavná trieda RSMidlet vytvára väčšiu časť instancií a preto obsahuje vo svojich atribútoch dôležité informácie ako sú pripojovacie URL, ktoré sú v tvare: `btsp://hostname:[CN | UUID];parametre`

- `btsp` - je schéma pre RFCOMM `StreamConnection`,
- `hostname` alebo `localhost` - pre nastavenie serverového spojenia, alebo Bluetooth adresa zariadenia pre pripojenie klientskeho zariadenia. Použitím `localhost` ako `hostname` definujeme, že sa jedná o serverové spojenie,
- `CN` (Channel Numer) – hodnota, ktorú používa klient pre pripojenie na server, je to podobný koncept ako TCP/IP port,
- `UUID` (Universally Unique Identifier) - sa používa pri nastavovaní služby na servery. Každé UUID je unikátne.
- `parametre` - zahrnujú meno (`name`) pre názov/popis služby, a taktiež bezpečnostné parametre ako je autentifikácia (`authenticate`), autorizácia (`authorize`) a šifrovanie (`encrypt`).

```
//pre klienta
private String uuidChat = "112233445566778899";
private String uuidFile = "998877665544332211";
//pre server
private String urlChat = "btsp://localhost:" + uuidChat + ";name=RsChat";
private String urlFile = "btgoep://localhost:" + uuidFile + ";name=RsFile";
```

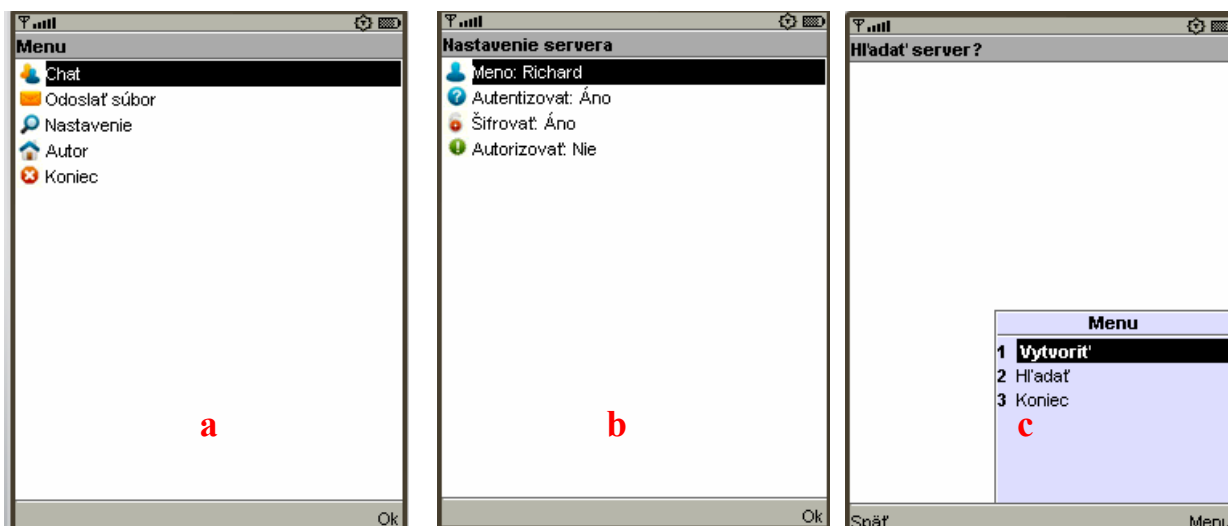
Vykonaním metódy `startApp()` dôjde k inicializácii displeja a zobrazeniu úvodného menu Obr. 7.3.a. Vytvorí sa taktiež instanciu triedy `SettingList`, ktorá musí byť inicializovaná pred spustením četu poprípadne položky “Nastavenie” (popis implementácie v nasledujúcich kapitolách). Menu obsahuje nasledujúce položky:

- **Chat** (komunikácia výmenou textových správ)
- **Odoslať súbor** (odoslanie súboru)
- **Nastavenie** (nastavenie vlastnosti komunikácie)
- **Autor** (informácie o autorovi aplikácie)
- **Koniec** (ukončenie aplikácie)

Ak užívateľ vyberie nieakú položku z menu (zoznamu), zavolá aplikačný manažér poslucháča udalostí stlačenia tlačítka `CommandListener`, kde sa zistí, ktoré položka bola vybraná.

7.3.1 Komunikácia výmenou textových správ

Ak si užívateľ vyberie možnosť komunikovať prostredníctvom textových správ v hlavnej triede `RSMidlet` program zistí volaním metódy `getChatNick()`, či už bolo nastavené meno požadované pre následný čet (výmenu súborov). Metóda `getChatNick()` zisťuje meno z triedy inicializovanej po spustení midletu a to `SettingList`. Ak táto trieda nemá uložené meno z predchádzajúceho spustenia (spúšťame midlet prvýkrát), aplikácia vyžiada od užívateľa zobrazením list komponenty triedy `SettingList` zadanie mena Obr. 7.3.b. Táto komponenta sa zobrazí aj po zvolení položky z hlavného menu „Nastavenie“.



Obr. 7.3: GUI aplikácie: a - úvodné menu, b – nastavenie, c – hľadanie servera v okolí.

Trieda SettingList

Hlavnou úlohou triedy `SettingList` je uložiť popřípade vyčítať údaj o mene používanom pri následnom čete, aby so zisťovaním neodťažoval užívateľa pri každom spustení znova. Štandardná J2ME nemá k ukladaniu dát súborový systém, ale používa veľmi jednoduché ukladanie polí bajtov, takzvaný systém správy záznamu RMS (Record Management System). Úložisko záznamu alebo ináč databáza (Record store) sa skladá zo zoznamu záznamu. Tieto záznamy tvoria polia bajtov a sú indexované od „1“. Jednotlivé databázy musia mať v rámci jednej sady midletov unikátne názvy. Za zachovanie dát pri vypnutí telefónu či výmene batérie odpovedá platforma. Pri odstránení sady midletov z telefónu sú automaticky odstránené i všetky jej databázy. Triedy a rozhrania špecifické pre prácu s databázami sa nachádzajú v balíku `javax.microedition.rms`, kde základnou triedou je `RecordStore`. Pre ukladanie a načítanie dát sú použité vstupné a výstupné toky z balíku `java.io`. Pre názornosť je uvedený príklad uloženia mena do pamäte.

```
public boolean addToStore(ByteOutputStream data) {
    // store = RecordStore.openRecordStore(storeName, true);
    if (store != null) {
        try {
            byte[] bytes = data.toByteArray();
            store.addRecord(bytes, 0, bytes.length);
            return true;
        } catch (Exception e) {
            return false;
        }
    }
    return false;
}
```

V triede `SettingList` sa taktiež nastavujú požiadavky serverovej časti nutné pre vytvorenie bezpečného spojenia s klientskou časťou vzdialeného zariadenia. Nastavenie bezpečnosti umožňuje vyžadovať od klientskeho zariadenia autentizáciu, autorizáciu a šifrovanie dát. Preddefinované je v aplikácii vyžadované autentizácia a šifrovanie vid'. Obr. 7.3.b.

Autentizácia spočíva v overení identity vzdialeného zariadenia. Autentizáciou sa vyžiada od každého zariadenia zadanie PIN kódu z ktorého sa šifrovacím algoritmom E_{22} odvodí potrebný 128-bitový zdieľaný kľúč linky. Ak sa zadané PIN kódy na jednotlivých stranách zhodujú, autentizácia je úspešná.

Vyžadovaním šifrovania docielime, že všetka komunikácia bude šifrovaná prúdovou šifrou E_0 v oboch smeroch. Pretože Bluetooth šifrovanie vyžaduje zdieľaný kľúč linky, šifrovanie vyžaduje autentizáciu. To znamená, že šifrovanie bez autentizácie nieje povolené.

Autorizácia je proces, kde poskytovateľ služby a teda serverové zariadenia povoľuje prístup iba na určité služby určitým zariadeniam. Implementácia autorizácie môže vyvolať aplikácia otázku pre serverové zariadenie, či klientskemu zariadenie môže byť povolený prístup k jeho službe. Tak ako šifrovanie tak aj autentizácia vyžaduje identifikáciu klientskeho zariadenia a preto autorizácia bez autentizácie nieje povolená. Aplikácia rieši tento problém automatickým nastavením `encrypt=false` a `authorize=false` ak je nastavené `authenticate=false`. Nastavovaním týchto bezpečnostných parametrov si užívateľ môže voliť v aplikácii úroveň bezpečnosti komunikácie so vzdialeným zariadením.

Trieda FindDevices

Ak si užívateľ vyberie možnosť komunikovať prostredníctvom textových správ a meno pre následný čít sa už nachádza v pamäti, hlavná trieda `RSMidlet` vytvorí instanciu triedy `FindDevices` a zobrazí sa komponenta list tejto triedy Obr. 7.3.c. Pri volaní konštruktora tejto triedy je potrebná inicializácia Bluetooth zariadenie.

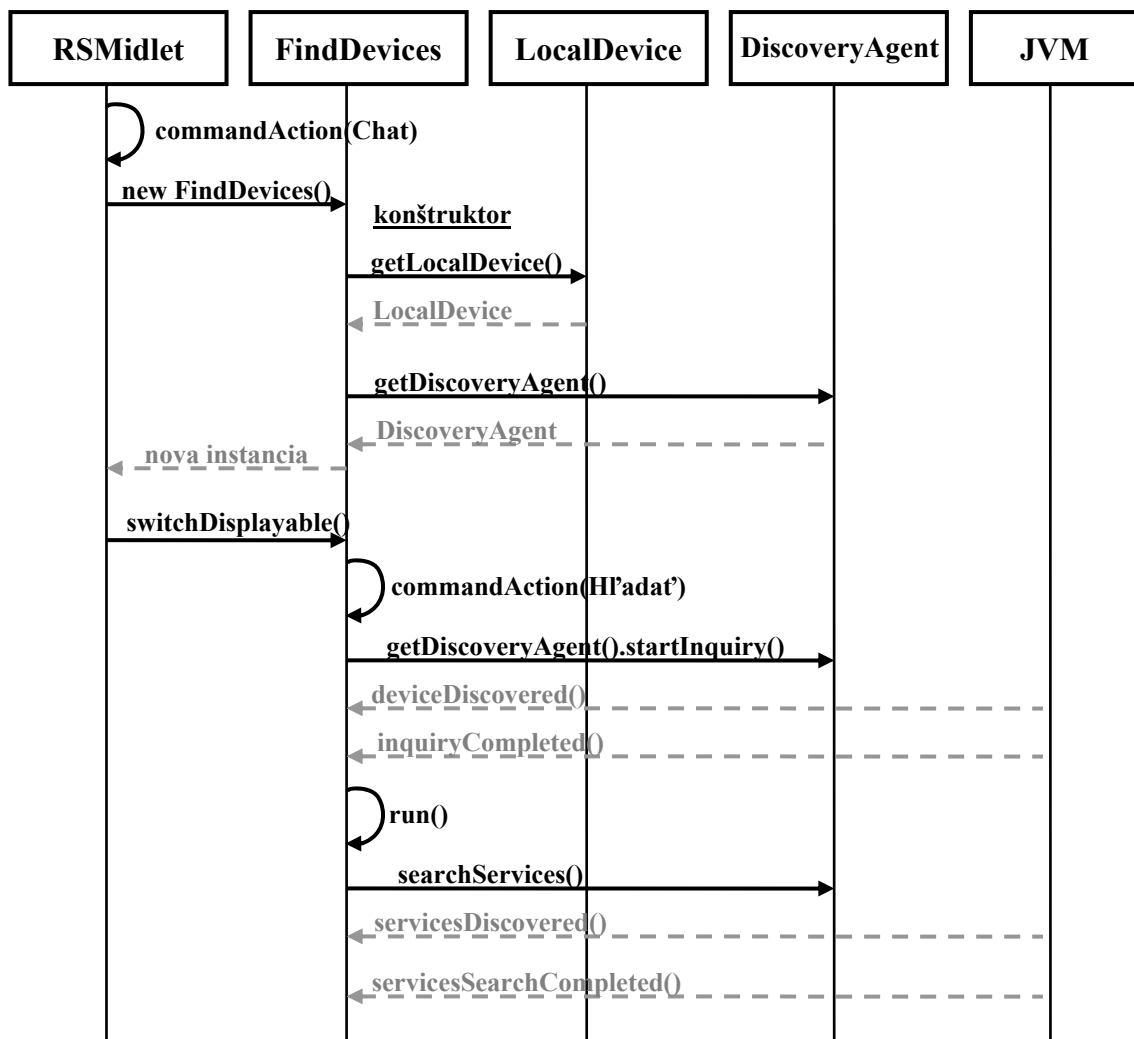
```
try {
    discoveryAgent = LocalDevice.getLocalDevice().getDiscoveryAgent();
    localDevice = LocalDevice.getLocalDevice();
} catch (BluetoothStateException ex) {
}
```

Zobrazená komponenta triedy `FindDevices`, nám umožňuje vytvoriť server alebo vyhľadať server v okolí a následne sa k nemu pripojiť ako klient.

Ako je ukázane na Obr. 7.4, pri výbere možnosti „Chat“ sa vytvorí instanciu triedy `FindDevice`, ktorá má prístup k triedam `LocalDevice` a `DiscoveryAgent`. Trieda `LocalDevice` obsahuje metódy pre správu lokálneho zariadenie Bluetooth napr. dokáže zistiť jeho MAC adresu, alebo prepnúť ho do módu viditeľného pre ostatných. Trieda `DiscoveryAgent` obsahuje metódy pre vyhľadávanie služieb na vzdialených zariadeniach.

Pri volaní konštruktora dôjde k inicializácii Bluetooth zariadenia a následne sa zobrazí komponenta list triedy `FindDevice`. Po zvolení možnosti „Hľadať“ (hľadať zariadenie v okolí) sa volaním metódy `startInquiry()` zaháji vyhľadávanie zapnutých zariadení Bluetooth v okolí. Keď sa v okolí nájde zariadenie so zapnutým Bluetooth javovský virtuálny stroj JVM volá zainplementovanú metódu `deviceDiscovered()` v triede `FindDevices` s jeho údajmi. Po vyhľadaní všetkých dostupných zariadení týmto spôsobom nakoniec JVM zavolá zainplementovanú metódu `inquiryCompleted()` a tým umožni začať vyhľadávanie dostupných služieb na vyhľadaných zariadeniach volaním metódy `searchServices()` triedy `DiscoveryAgent`. JVM volá

zainplementovanu metódu triedy `FindDevices` pri nájdení každej dostupnej služby a po vyhládaní všetkých služieb v okolí na všetkých dostupných zariadeniach, JVM volá zainplementovanú metódu triedy `FindDevices` a to `servicesSearchCompleted()`. Týmto spôsobom aplikácia dokáže vyhládať v okolí server so spustenou, požadovanou službou a zobraziť ho na displej mobilného telefónu, aby tým bola užívateľovi poskytnutá možnosť pripojiť sa ako klient Obr. 7.6.a.



Obr. 7.4: Diagram sekvencií pri vyhládaní služby na vzdialených zariadeniach

Navrhnutý komunikačný paket

Pre textovú komunikáciu bol navrhnutý komunikačný paket, kde správa je vkladaná za informačne pole o dĺžke 55 bajtov Obr. 7.5. Prvých 5 bajtov nesie informáciu o dĺžke prenášaného packetu. Ďalšie pole o dĺžke 25 bajtov nesie informáciu o mene odosielateľa potrebného pre odlišenie autorov správy v čete. Posledná časť v informačnom poly o dĺžke 25 bajtov nesie informáciu o MAC adrese Bluetooth zariadenia odosielateľa, ktorá je pre každé zariadenie jedinečná.

| | | | | | | | | | | | | | | |
|--------------------|---|---|---|---|---|---|--------------------------|---|---|---|---|--------------------------------|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 2 | 3 | 3 | 3 | ... | 5 | 5 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | | 9 | 0 | 1 | 2 | | 4 | 5 |
| Dĺžka paketu 5B | | | | | | | Meno odosielateľa 25B | | | | | MAC adresa odosielateľa 25B | | |
| Prenášaná správa | | | | | | | | | | | | | | |

Obr. 7.5: Navrhnutý komunikačný paket

Trieda Chat

Pri zvolení možnosti „Vytvoriť“ (vytvoriť server), trieda `FindDevices` vytvorí instanciu triedy `ServerConnectionService` (viď. kapitola 7.3.1 *Trieda `ServerConnectionService`*) a instanciu triedy `Chat`.

```

if (chat == null) {
    chat = new Chat(this, rsMidlet, localDevice);
    connectionService = new ServerConnectionService(url, chat);
}
rsMidlet.switchDisplayable(chat, true);

```

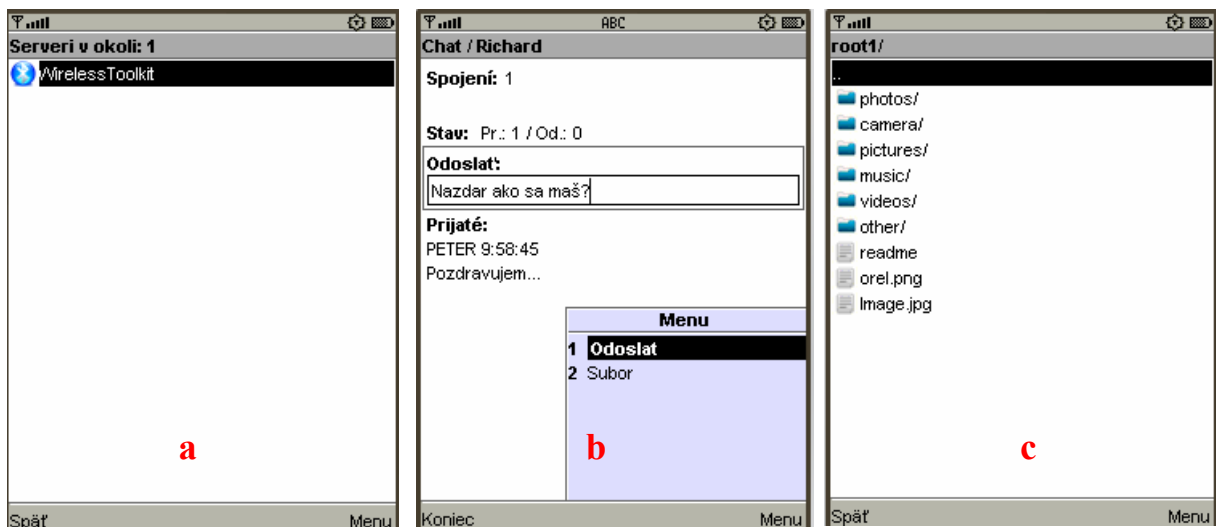
Trieda `Chat` je hlavnou triedou, ktorá spracováva prijaté správy zo serverovej časti (`ServerHandler`) a z klientskej časti (`ClientHandler`), vytvára správy na odoslanie, taktiež zobrazuje spracované správy a informuje užívateľa o udalostiach prostredníctvom svojej komponenty typu formulár Obr. 7.6.b.

Proces spracovania prijatej správy spočíva v zistení mena, MAC adresy a samotnej správy odosielateľa z navrhnutého komunikačného paketu. Meno sa kontroluje pre prípad ak by komunikovali viaceré strany s rovnakým menom a nedošlo tak k zámene pôvodcu správy. Ak by k takému prípadu došlo, aplikácia doplní duplikované meno o textový reťazec „[2]“. Nutnosť spracovať MAC adresu spočíva v zabránení vzniku komunikačných slučiek, ktoré môžu vzniknúť z dôvodu realizácie riešenia aplikácie. Ak príde zariadeniu paket, ktorý obsahuje v informačnom poli jeho vlastnú MAC adresu, nepošle tento paket ďalej, ale upozorní užívateľa o slučke a umožní mu ukončiť spojenie zo serverovej strany aplikácie. Prenášanie MAC adresy odosielateľa je teda potrebné pre jednoznačné určenie pôvodcu správy.

Vytváranie správy sa zaháji pri požiadavku užívateľa o odoslanie správy. Vytvorí sa komunikačný paket obsahujúci meno, MAC adresu miestneho zariadenia Bluetooth a samotný text správy. Tento paket sa predá klientskej (`ClientHandler`) alebo serverovej (`ServerHandler`) časti v závislosti na tom, ktorá časť má inicializované spojenie (môžu nastať aj oba prípady naraz).

Užívateľ je informovaný o udalostiach na displeji viď. Obr. 7.6.b. Položka „Spojení:“ informuje o počtu spojení. Môže nadobúdať hodnoty „0“ (čaká na pripojenie

klienta), „1“ (je naviazane jedno spojenie so vzdialeným zariadením), „2“ (komunikuje s klientom a aj so serverom). Položka „Stav:“ informuje o naviazaní klientskeho/serverového spojenia, úspešnosť odoslania alebo prijatia správy, informáciu o nutnosti odpojenia klientskej časti z dôvodu vzniku slučka a informácie o chybách. Položka „Odoslať:“ umožňuje užívateľovi vložiť text na odoslanie komunikujúcim zariadeniam. Položka „Prijaté:“ zobrazuje prijaté a odoslané správy. Tvar zobrazovania je definovaný tak, že v prvom riadku sa zobrazí meno odosielateľa v tvare veľkých písmen, za meno sa pridá čas odoslania správa a na ďalšom riadku je samotná správa.



Obr. 7.6: GUI aplikácie: a – nájdený server v okolí, b – textová komunikácia, c – prehľadávanie súborového systému zariadenia.

Trieda `ServerConnectionService`

Hlavnou úlohou triedy `ServerConnectionService` je vytvoriť záznam služby (service record), túto službu pridať do databázy objavovaných služieb SDDB (Service Discovery Database) vzdialenými zariadeniami, registrovať túto službu a čakať na prichádzajúce klientske spojenie.

Implementácia automaticky vytvorí záznam služby, keď v aplikácii vytvoríme oznamovač spojenia (connection notifier) `StreamConnectionNotifier` volaním `Connector.open()` s URL pre vytvorenie servera .

```
try {
    streamConnectionNotifier = (StreamConnectionNotifier) Connector.open(url);
} catch (IOException ex) {
    String errorMessage = "Chyba pri vytváraní služby serverom: " + ex.getMessage();
    chat.handleError(errorMessage);
    aborting = true;
}
```

Ak je vytvorený oznamovač spojenia a záznam služby, server je tak pripravený registrovať službu a čakať na pripojenie klienta. Volaním metódy `acceptAndOpen()` oznamovača sa vloží záznam služby pre asociované spojenie do SDDB (Service Discovery Database) čo spraví službu viditeľnú pre ostatných klientov. Metóda tu čaká, kým nepríde prichádzajúce spojenie. Dĺžku po ktorú bude služba viditeľná pre ostatných klientov sa dá nastaviť volaním `LocalDevice.getLocalDevice()`, kde špecifikujeme IAC (Inquiry Access code).

- `DiscoveryAgent.LIAC` (Limited Inquiry Access Code) – zariadenie bude viditeľné pre ostatné zariadenia po určitú krátku dobu, zvyčajne jednu minútu.
- `DiscoveryAgent.GIAC` (General Inquiry Access Code) – zariadenie bude stále viditeľné pre ostatné zariadenia, čo je pre aplikáciu vhodnejšia voľba.

```
LocalDevice.getLocalDevice().setDiscoverable(DiscoveryAgent.GIAC);
StreamConnection streamConnection =
    (StreamConnection) streamConnectionNotifier.acceptAndOpen();

if (!aborting) {
    ServerHandler serverHandler = new ServerHandler(this, streamConnection, chat);
    chat.handlerServerAcceptAndOpen(this, serverHandler);
}
```

Triedy `ServerHandler` a `ClientHandler`

Triedy majú rovnakú funkciu a teda aj metódy, ale odlišujú sa v spôsobe naviazovania komunikácie. Trieda `ClientHandler` po vyhľadani zariadení v okolí s požadovanou službou získa url adresu pre pripojenie a zahájí spojenie ako klient. Instancia tejto triedy sa vytvára v triede `Chat`.

```
//získať adresy zvoleného servera
url = serviceRecord.getConnectionURL(
    ServiceRecord.NOAUTHENTICATE_NOENCRYPT, false);
//pripojiť sa k zvolenému serveru
connection = (StreamConnection) Connector.open(url);
```

Po pripojení klienta na zverejnenú službu triedou `ServerConnectionService` sa vytvorí v tejto triede instancia triedy `ServerHandler` s odkazom na triedu `Chat`, potrebnú pre zobrazenie a spracovanie informácií.

Nasledujúci popis je rovnaký pre obe triedy, ktorých hlavnou úlohou je výmena textových správ s pripojeným klientom/serverom.

So zahájením spojenia sa spustí vlákno v ktorom sa čaká na príjem správy a taktiež sa vytvorí vnorená trieda `ServerWriter/ClientWriter`, ktorá sa stará o odoslanie

správ pre pripojené zariadenie z fronty správ vkladanych z triedy Chat. Odosielanie správy prebieha tak, že sa vyberie správa z fronty, zistí sa dĺžka tejto správy, následne sa doplní komunikačný paket o informáciu dĺžky správy a odošle paket so správou vzdialenému zariadeniu.

Pri prijímu správa sa najprv zoberie prvých 5 bajtov, ktoré nesú informáciu o dĺžke prichádzajúcej správy. Následne sa alokuje potrebná pamäť pre prijatie a dôjde k samotnému stiahnutiu celej správa. Prijatá správa sa predá pre ďalšie spracovanie do triedy Chat.

```
//prijem sprav
while (!aborting) {
    int lengthPacket = 0;
    try {
        //alokuje pamäť pre prijem infa o dĺžke prichádzajúceho paketu 5B
        byte[] lengthInfoBuf = new byte[lengthInfo];
        //prijme prvých 5B z prichádzajúceho paketu
        readStream(inputStream, lengthInfoBuf);
        //zisti veľkosť paketu z 5B
        lengthPacket = readLength(lengthInfoBuf);
        //alokuje pamäť pre zbytok paketu
        byte[] packet = new byte[lengthPacket];
        //prijme celý zbytok paketu
        readStream(inputStream, packet);
        //informuje a zobrazí prijatú správu
        chat.handlerServerReceived(this, packet);
    } catch (IOException ex) {
        close();
        if (lengthPacket == 0) { // došlo k ukončeniu spojenia s klientom
            chat.handlerServerClose(this);
        } else { // chyba pri prijímu správy
            chat.handlerServerErrorClose(this, ex.getMessage());
        }
    }
}
```

7.3.2 Komunikácia prostredníctvom výmeny súborov

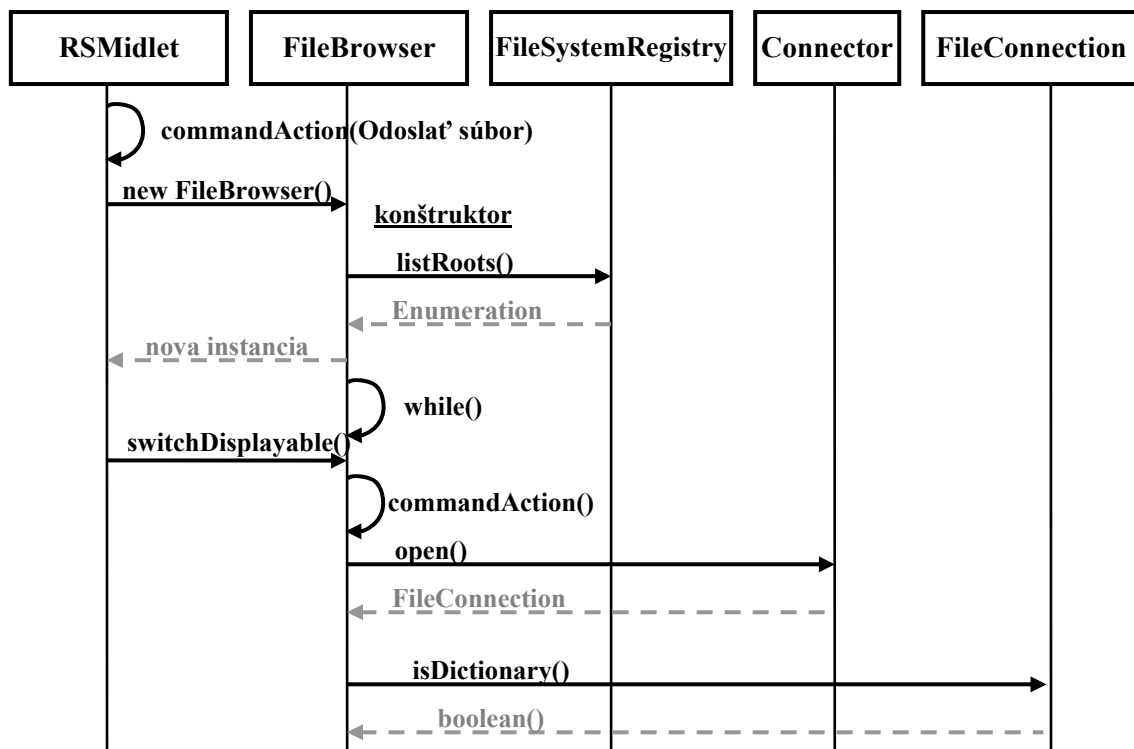
Ak si užívateľ zvolí komunikáciu prostredníctvom výmeny súborov voľbou „Odoslať súbor“ z hlavného menu Obr. 7.3.a, skontroluje sa obdobným spôsobom ako v kapitole 7.3.1 *Komunikácia výmenou textových správ*, či už bolo zadané potrebné meno pre výmenu súborov (čet). Ak toto meno už bolo zadané vytvorí sa v hlavnej triede RSMidlet instancie tried FileBrowser (viď. kapitola 7.3.2 *Trieda FileBrowser*)

a `GuiObexSender` (viď. kapitola 7.3.2 *Triedy `GuiObexSender` a `ObexSender`*). Keďže sa jedná o odosielanie súboru aplikácia očakáva zvolenie súboru na odoslanie v triede `FileBrowser`.

Trieda `FileBrowser`

Trieda `FileBrowser` slúži špeciálne pre prácu so súborovým systémom mobilného zariadenia a s súborovým systémom prídavnej pamäti (pamäťová karta), ak ju zariadenie obsahuje.

Pri volaní konštruktora sa inicializuje súborový systém (Obr. 7.7) a zobrazí sa.



Obr. 7.7: Diagram sekvencií pre zistenie súborového systému zariadenia

Zobrazenie súborového systému je prostredníctvom komponenty `list`, ktorý sa v závislosti na požiadavkách užívateľa mení Obr. 7.6.c. Ak si užívateľ vyberie niejaký adresár alebo súbor zo zoznamu, aplikačný manažér zavolá poslucháča udalosti stlačenia tlačítka `CommandListener`, ktorý zistí ktoré tlačítko bolo stlačené a podľa toho aplikácia určí ďalšiu akciu.

Ak sa jedná o odoslanie súboru aplikácia očakáva zvolenie súboru. Ak sa správne zvolí súbor, užívateľ je tým informovaný upozorňujúcou komponentov `alert` triedy `GuiObexSender`.

Pri prichádzajúcom súbore aplikácia umožňuje zvoliť adresár pre uloženie. Po stiahnutí súboru je volaná trieda `FileBrowser`, ale očakáva zvolenie adresára pre

uloženie prichádzajúceho súboru. O správnosti zvolenia adresára je užívateľ informovaný upozorňujúcou komponentov alert triedy `GuiObexReceiver`.

Najdôležitejšie metódy v triede sú `traverseDirectory()`, ktorá je potrebná pre správny pohyb súborovým systémom a `showActualDir()`, ktorá slúži k prístupu a zobrazeniu súborového systému.

```
void showActualDir() {
    Enumeration e;
    FileConnection actualDir = null;
    try { //jedna sa o koreň súborového systému
        if (MAIN_ROOT.equals(actualDirName)) {
            //získanie koreňového adresára
            e = FileSystemRegistry.listRoots();
            //vyčistiť komponentu List
            while(size() > 0){
                delete(0);
            }
            setTitle(actualDirName); //nastaviť title
        } else { //nejedna sa o koreň súborového systému
            // otvor zoznam súborového systém s požadovanou adresou
            actualDir = (FileConnection) Connector.open("file://" + actualDirName);
            e = actualDir.list();
            while(size() > 0){
                delete(0);
            }
            setTitle(actualDirName); //nastaviť title
            append(UP_DIRECTORY, null); // pridať ".."
        }
    }
}
```

Pri volaní metódy `listRoot()` z triedy `FileSystémRegistry` z JSR-75 (File Connection API) umožňujúca prístup k súborom, získame zoznam koreňového adresára. Prístup ku konkrétnemu adresáru alebo súboru realizujeme volaním metódy `Connector.open(url)`, kde `url` musí byť v správnom tvare ako napr. „file:///c:/photo/“. O správnosť `url` sa stará spomínaná metóda `traverseDirectory()`. Návratová hodnota z prístupu k súborovému systému mobilného zariadenia je zoznam (Enumeration), ktorý zobrazíme do komponenty `list` triedy `FileBrowser`.

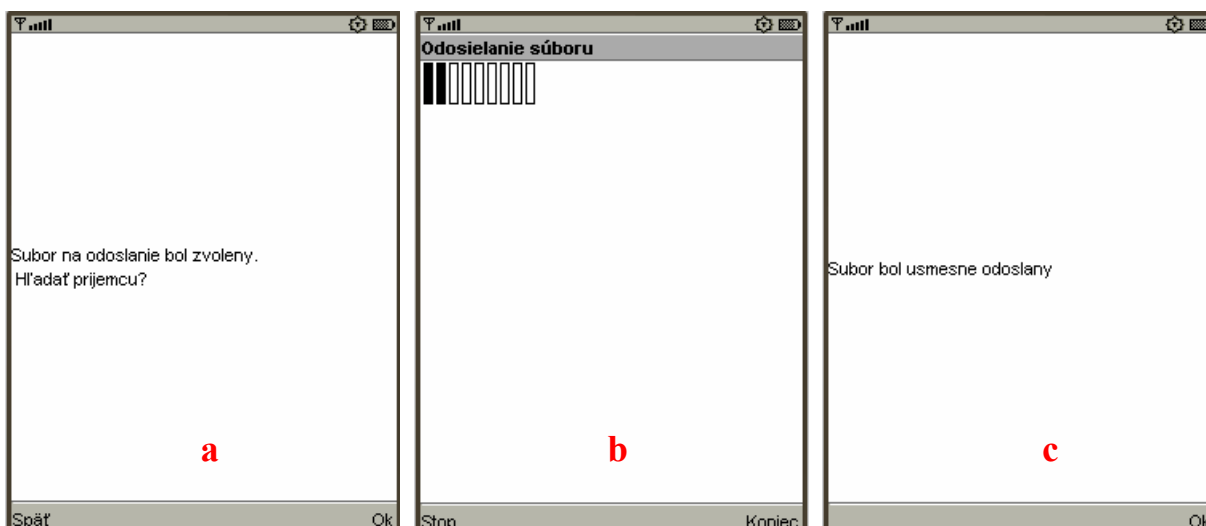
```

while (e.hasMoreElements()) {
    //názov adresára alebo súboru
    String fileName = (String) e.nextElement();
    try { //jedna sa o adresár
        if (fileName.charAt(fileName.length() - 1) == SEP_CHAR) {
            append(fileName, Image.createImage("/rsIcons/Adresar.PNG"));
        } else { //jedna sa o súbor
            append(fileName, Image.createImage("/rsIcons/File.PNG"));
        }
    } catch (IOException ioe) {
    }
}
}

```

Triedy GuiObexSender a ObexSender

Funkciou triedy GuiObexSender je zaistiť grafické užívateľské rozhranie pri odosielaní súboru, ktoré sa odosiela v triede ObexSender. Po úspešnom zvolení súboru v triede FileBrowser dôjde k zobrazeniu upozorňovacej komponenty alert triedy GuiObexSender Obr. 7.8.a. Upozornením oznamuje aplikácia užívateľovi, že súbor na odoslanie bol zvolený a či má pokračovať a to vo vyhľadani príjemcu správy.



Obr. 7.8: GUI aplikácie: a – upozornenie o zvolení súboru, b – priebeh odosielania súboru, c – stav odoslaného súboru.

Potvrdenie upozornenia vytvorí instanciu triedy FindDevices z triedy RSMidlet. Pri volaní konštruktora sa inicializujú parameter url pre hľadanie služby určenej pre príjem súboru (url servera) a zobrazí sa komponenta list vyhľadávania zariadení v okolí z triedy FindDevices Obr. 7.6.a. Po vyhľadaní a zvolení obdobne, ako to je popísané v kapitole 7.3.1 *Trieda FindDevices*, sa vytvorí instancia triedy ObexSender z triedy

RSMidlet s potrebnými parametrami ako je url pre pripojenie k serveru a zvolený súbor na odoslanie. Pri inicializácii vytvorenej triedy ObexSender sa spustí vlákno, v ktorom sa načíta zvolený súbor, vytvorí spojenie zo serverom a odošle samotný súbor.

Vytvorenie klientskeho spojenia pre výmenu súborov (OBEX) aplikácia vytvára volaním metódy `Connector.open(url)`, kde url je adresa serveru.

```
// vytvorenie spojenia
private void createConection() throws IOException {
    // snaha o vytvorenie klientskeho spojenia so serverom
    connection = Connector.open(connectionURL);
    clientSession = (ClientSession) connection;
    // odoslanie zostavovacieho paketu // čaká na odpoveď od servera
    HeaderSet response = clientSession.connect(null);
    // server odpovedal // získaj odpoveď
    int codee = response.getResponseCode();
    // ak súhlasí zo spojením tak pokračuj
    if (codee != ResponseCodes.OBEX_HTTP_OK) {
        throw new IOException();
    }
}
```

Pre dokončenie naviazania klientskej relácie pošle klient spojovací paket serverovej časti, ktorý obsahuje v hlavičke číslo verzie OBEX, príznaky a maximálnu dĺžku paketu a to volaním metódy `connect(null)`. Aby bola klientska relácia vytvorená musí serverová aplikácia súhlasiť s parametrami a potvrdiť požiadavku na vytvorenie relácie zodpovednou `ResponseCodes.OBEX_HTTP_OK`. Klientska aplikácia čaká na odpoveď od serverovej aplikácie. Po príchode odpovede získa informáciu o stavu volaním metódy `getResponseCode()`. Pri chybe spojenia posielajú serverová aplikácia iný kód odpovede ako napríklad: `ResponseCodes.OBEX_INTERNAL_ERROR`, ktorý informuje klientsku aplikáciu o type chyby. Tieto odpovede posielané serverovou aplikáciou sú definované v triede `javax.obex.ResponseCodes`.

Ak serverová aplikácia súhlasí vo vytvorení relácie, klientska aplikácia doplní hlavičku paketu prenášaného súboru volaním metódy `createHeaderSet()` a vyplní ju informáciami ako je odosielateľ, názov súboru a jej veľkosť. Po vytvorení hlavičky otvorí a zahájí odosielanie serverovej aplikácii.

Po úspešnom odoslaní súboru serverovej aplikácii je znova zobrazená komponenta upozornenia alert triedy `GuiObexSender`, ktorá informuje o úspešnosti poprípade chybe s odoslaním súboru.

```

private void uploadFile() throws IOException, Throwable {
    int position = 0;
    // vytvorenie hlavičky
    HeaderSet headerSet = clientSession.createHeaderSet();
    // vkladanie informácií do hlavičky
    headerSet.setHeader(HeaderSet.NAME, selectedNameOfFile); //meno súboru
    headerSet.setHeader(HeaderSet.WHO, nickChat.getBytes()); //odosielateľ
    headerSet.setHeader(HeaderSet.LENGTH, new Long(file.length)); //veľkosť
    // vloženie do hlavičky
    operation = clientSession.put(headerSet);
    // otvorenie vysielacieho toku
    outputStream = operation.openOutputStream();
    //inicializácia progresBaru
    guiObexSender.sendingProgress(file.length);
    //odosielanie súboru //while
    while (position != file.length) {

```

Triedy GuiObexReceiver a ObexReceiver

Funkciou triedy `GuiObexReceiver` je zaistiť užívateľské rozhranie pre príjem súboru o ktorého príjem sa stará trieda `ObexReceiver`. Instancia triedy `GuiObexReceiver` vzniká v triede `RSMidlet`. Vznik instance triedy `GuiObexReceiver`, v ktorej konštruktoře sa volá instancia triedy `ObexReceiver` je podmienený nutnosťou inicializácie Bluetooth zariadenia a preto možnosť prijatia súboru je možná až pri zvolení položky „Chat“ z hlavného menu triedy `RSMidlet` Obr. 7.3.a. Pri vzniku triedy `ObexReceiver` vzniká aj vnorená trieda `HandlerReceiver`. V konštruktoře `ObexReceiver` sa spúšťa vlákno, ktoré sa stará o vytvorenie služby a zviditeľneniu služby klientom. Na prichádzajúce spojenie čaká vnorená trieda `HandlerReceiver`, ktorá plní funkciu poslucháča udalostí, ktorý odpovedá na požiadavky typu OBEX od klienta k serveru vďaka rozširujúcej triede `ServerRequestHandler` definovanej `javax.obex.ServerRequestHandler`.

```

public void run() {
    try {
        synchronized (this) {
            if (session == null) {
                LocalDevice.getLocalDevice().setDiscoverable(DiscoveryAgent.GIAC);
                //vytvorenie záznamu služby s adresou servera
                session = (SessionNotifier) Connector.open(urlFile);
            }
            state = OPENED;
        }
        while (state != CLOSED) {
            // zviditeľnenie služby pre klientov
            // čaká na pripojenie klienta vnorenou triedou HandlerReceiver
            session.acceptAndOpen(handlerReceiver);
            // obmedzenie viacerých prístupov naraz
            preventOverloading();
        }
    }
}

```

Po príchode požiadavku so syntaxou `connect(null)`, ktorý žiada server o odpoveď či môže byť vytvorená relácia zo strany klienta, odpovie vnorená trieda servera kódom odpovede `ResponseCodes.OBEX_HTTP_OK` (súhlasím s vytvorením relácie).

```

class HandlerReceiver extends ServerRequestHandler {

    public HandlerReceiver() {
    }
    //odpoveď na klientske put(headerSet) paketu obsahujúci súbor
    //po úspešnom prijatí bude odpoveď: ResponseCodes.OBEX_HTTP_OK
    public int onPut(Operation operation) {
        return receiveFile(operation);
    }
    //odpoveď na klientsku otázku connect(null): ResponseCodes.OBEX_HTTP_OK
    public int onConnect(HeaderSet request, HeaderSet reply) {
        return ResponseCodes.OBEX_HTTP_OK;
    }
}

```

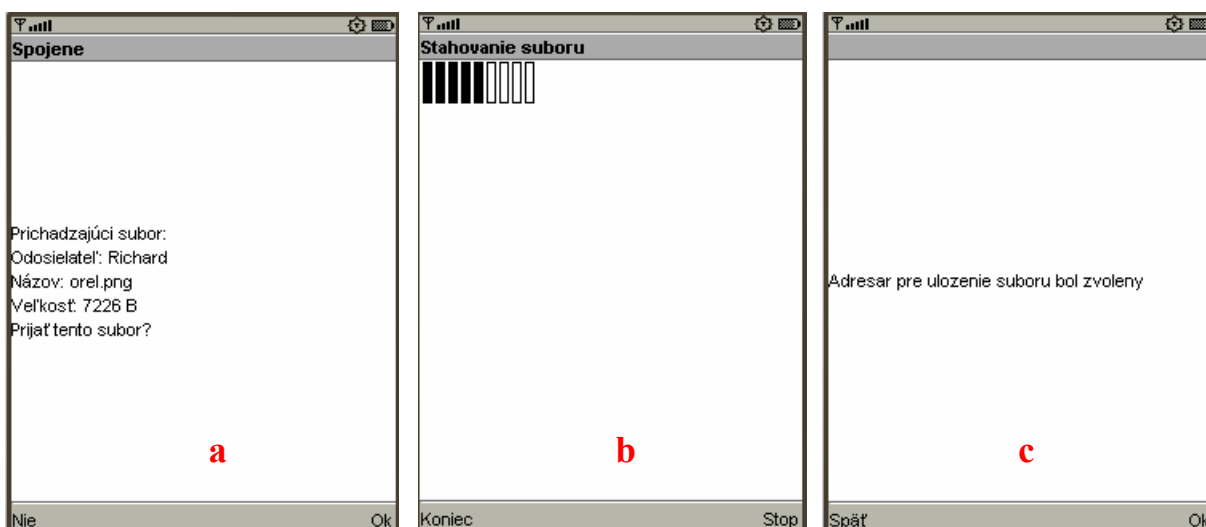
Pri príchode paketu zo strany klienta s nastavenou hlavičkou (metóda `put(headerSet)` vid'. kapitola 7.3.2 *Triedy GuiObexSender a ObexSender*) informácií o odosielateľovi a súbore, serverová aplikácia zahájí prijímanie súboru. Pred samotným príjmom výbere

informácie z hlavičky paketu a spýta sa užívateľa či chce prijať tento súbor. Upozornenie na prichádzajúci súbor sa uskutočňuje prostredníctvom upozorňujúcej komponenty alert Obr. 7.9.a. Ak užívateľ súhlasí s príjmom automaticky sa zaháji samotné sťahovanie súboru.

```
//zistenie z hlavičky paketu informácie
headers = operation.getReceivedHeaders();
nameOfFile = (String) headers.getHeader(HeaderSet.NAME);//meno súboru
whoSendFile = (byte[]) headers.getHeader(HeaderSet.WHO);//odosielateľ a
//veľkosť
lengthOfFile = (int) ((Long) headers.getHeader(HeaderSet.LENGTH)).longValue();
//zmena byte[] na string
String whoString = new String(whoSendFile);

// opýtaj sa užívateľa či chce prijať súbor
if (guiObexReceiver.askReceiveFile(nameOfFile, whoString, lengthOfFile)) {
    //prijímanie súboru, uloženie súboru do zvoleného adresára
    guiObexReceiver.choiceDictionary(downloadImage(lengthOfFile));
    //odpoveď klientovi súbor bol prijatý úspešne
    responseCode = ResponseCodes.OBEX_HTTP_OK;
} else { //odmietnutie príjmu súboru
    responseCode = ResponseCodes.OBEX_HTTP_FORBIDDEN;
}
```

O priebehu sťahovania súboru je užívateľ informovaný komponentov formulár triedy GuiObexReceiver Obr. 7.9.b. Po stiahnutí súboru je odoslaná odpoveď stavu prenosu súboru klientskej aplikácii metódou onPut () z vnorenej triedy HandlerReceiver.



Obr. 7.9: GUI aplikácie: a – prichádzajúci súbor, b – priebeh sťahovania súboru, c – stav uloženia súboru.

Po stiahnutí súboru sa z triedy `GuiObexReceiver` vytvorí instancia triedy `FileBrowser`. Pri volaní konštruktora tejto triedy sa inicializuje súborový systém mobilného zariadenia a zobrazí sa. Funkcia triedy `FileBrowser` je popísaná vid'. kapitola 7.3.2 *Trieda FileBrowser*, ale požiadavka je zvolenie adresára pre uloženie súboru. O správnosti zvolenia adresára je užívateľ informovaný upozorňujúcou komponentov alert. Potvrdením sa zaháji samotne ukladanie prijatého súboru. Otvory sa súborový systém v mieste zvoleného adresára a pridá sa k adrese meno súboru. Pred samotným uložením sa najprv skontroluje či už takýto súbor existuje. Ak zvolený súbor existuje, ukončí sa metóda a upozorní užívateľa o existencii súboru upozorňujúcou komponentov alert z triedy `GuiObexReceiver`. Ak súbor ešte neexistuje vytvorí sa, uloží na zvolenom mieste a informuje užívateľa o stave uloženia Obr. 7.9.c. Po skončení procesu ukladania súboru sa aplikácia vráti do miesta, kde sa nachádzala pred prijatím súboru.

```
//informuj užívateľa o ukladaní súboru
guiObexReceiver.infoStatusForm("Ukladám súbor", false);
try { //otvor súborový systém na zvolenom mieste
    fileConnection = (FileConnection) Connector.open("file://"
                                                    + dictionaryURL + nameOfFile);
    //zisti či už existuje zvolená cesta, ak áno informuj užívateľa a ukončí
    if (fileConnection.exists()) {
        guiObexReceiver.infoStatusForm(dictionaryURL + nameOfFile
                                        + " tento subor už existuje", true);
        return;
    }
    //vytvor požadovanú cestu do súborového systému
    fileConnection.create();
    outputStream = fileConnection.openOutputStream();
    //zápis súbor do súborového systému
    for (int i = 0; i < file.length; i++) {
        outputStream.write(file[i]);
    }
    fileConnection.close();
    outputStream.close();
    //informuj o stave uloženia užívateľa
```

8. Experimentálne testovanie

V rámci testovania vytvoreného zdrojového kódu bola úspešne prevedená experimentálna komunikácia v reálnych podmienkach medzi mobilnými telefónmi. Pre test funkčnosti aplikácie výmeny textových správ a dátových súborov boli využité mobilné telefóny typu Sony Ericsson K550i, K608i a K530i.



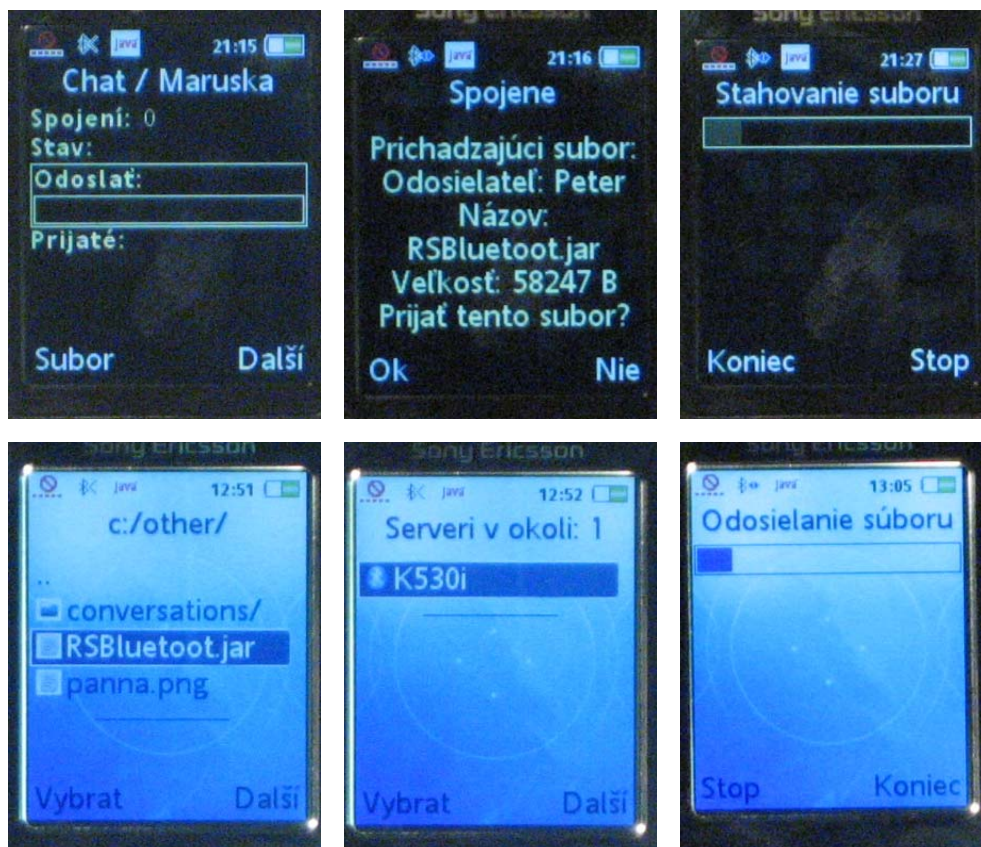
Obr. 8.1: Testovacie mobilné zariadenia zľava Sony Ericsson K550i, K608i a K530i

Na Obr. 8.2 je zaznamenaný prenos textových správ medzi testovacími telefónmi. Mobilný telefón K608i v tomto prípade plnil funkciu servera pre K550i a zároveň bol klientom pre mobilný telefón K530i. Správu „Zdravím vas...“ odoslanú z mobilného telefón K608i obdržali všetky zariadenia zúčastnené komunikácie.



Obr. 8.2: Test prenosu textových správ medzi zľava SonyEricsson K608i, K530i a K550i.

Na Obr. 8.3 je zaznamenaný prenos dátového súboru medzi testovacími telefónmi K530i a K550i. Mobilný telefón K530i plnil funkciu servera ku ktorému sa pripojil mobilný telefón K550i ako klient za účelom odoslania súboru s názvom „RSBluetooth.jar“. Mobilný telefón K550i po vyhľadani zariadení v okolí, zvolí príjemcu súboru. Príjemca súboru je upozornený o prichádzajúcom súbore. Po povolení prijatia nastane samotný prenos súboru. Prenos súboru medzi zariadeniami prebehol úspešne. Prijatý súbor si následne príjemca na K530i môže uložiť do ním zvoleného adresára.



Obr. 8.3: Test prenosu dátového súboru medzi (hore) K530i a (dole) K550i.

9. Záver

Táto diplomová práca sa zaoberala problematikou komunikačných technológií mobilných zariadení v sieťach typu WPAN. Následne bola priblížená problematika programovania J2ME aplikácií a ich využitie pri bezdrôtových prenosoch v zariadeniach, ktoré využívajú štandard technológie Bluetooth.

V prvej časti tejto diplomovej práce boli charakterizované osobné bezdrôtové siete WPAN, určené predovšetkým ako náhrada metalického vedenia pre komunikáciu a vzájomnú spoluprácu na malú vzdialenosť medzi zariadeniami ako sú mobilné telefóny, počítače, rôzne periférie a spotrebná elektronika. V kapitole boli tiež vymenované a krátko charakterizované technológie, ktoré spadajú do tejto kategórie podľa IEEE 802.15.

V ďalších troch kapitolách bol prevedený teoretický popis štandardov WPAN a to UWB, ZigBee a predovšetkým Bluetooth. Vymenované technológie spadajúce do WPAN siete boli následne zhodnotené a porovnané v samostatnej kapitole.

Nasledujúca kapitola bola venovaná problematike programovacieho jazyka Java a to konkrétne J2ME, pomocou ktorého sa dajú konfigurovať zariadenia s obmedzenými funkciami a vlastnosťami. Takéto zariadenia sú napríklad mobilné telefóny, pagery, organizéry využívajúce CLDC konfiguráciu a špeciálne vytvorené pracovné prostredie KVM od Javy. Kapitola tiež približuje problematiku použitia sád programových rozhraní MIDP a Java API pre Bluetooth, potrebných pri vývoji aplikácií, ktoré využívajú pre bezdrôtový prenos technológiu Bluetooth.

V ďalšej časti bola navrhnutá koncepcia komunikačnej platformy, umožňujúca komunikáciu dvoch a viacerých mobilných zariadení medzi sebou. Táto koncepcia umožňuje prenos textových správ a dátových súborov medzi jednotlivými zariadeniami. Koncový bod u tejto koncepcie umožňuje plniť funkciu servera i klienta podľa potreby a tým je zabezpečená peer-to-peer komunikácia.

V rámci tejto diplomovej práce bol vytvorený zdrojový kód navrhutej koncepcie v programovacom jazyku Java vo vývojovom prostredí NetBeans 6.5.1 (GNU/GPL) na platforme J2ME s použitím konfigurácie CLDC 1.0 a profilov MIDP 1.0, Java API pre Bluetooth (súčasť J2ME). Pre vývoj bol použitý emulátor obecného mobilného telefónu obsiahnutý v Java Wireless Toolkit 2.5.2 (integrované vo vývojovom prostredí). Vytvorený zdrojový kód umožňuje výmeny súborov ako aj písomnú konverzáciu dvoch alebo viacerých ľudí v reálnom čase prostredníctvom Bluetooth spojenia.

V rámci testovania vytvoreného zdrojového kódu bola úspešne prevedená experimentálna komunikácia v reálnych podmienkach medzi mobilnými telefónmi. Pre test funkčnosti aplikácie výmeny textových správ a dátových súborov boli využité mobilné telefóny typu Sony Ericsson K550i, K530i, K608i.

Zoznam použitej literatúry

- [1] MAHMOUD, Qusay. H. *Naučte se Java 2 Micro Edition*. Přeložil Pavel Makovec. [s.l.]: GRADA Publishing, 2002. 264 s. ISBN 80-247-0444-7.
- [2] CORDEIRO, Carlos De Morais, AQRAWAL, Dharma Prakash. *Ad hoc and sensor networks: Theory and Application*. 2006th edition. USA: [s.n.], 2006. 664s. ISBN: 981-256-681-3.
- [3] MIKULKA, Ján. *Koexistence mobilních komunikačních systémů WLAN a Bluetooth*. [s.l.], 2008. 107s. Disertační práce na fakultě Elektrotechniky a komunikačních technologií na Ústavu radioelektroniky. Vedoucí disertační práce prof. Ing. Stanislav Hanus, CSc.
- [4] HOPKINS, Bruce, RANJITH, Antony. *Bluetooth for Java* [s.l.] : [s.n.], 2003. 352 s. ISBN 1590590783.
- [5] ECMA international. *Standard ECMA-368* [online]. 3-rd edition. [2008] , December 2008 [cit. 2008-12-10]. Standard. Dostupný z WWW: <<http://www.ecma-international.org/publications/standards/Ecma-368.htm>>.
- [6] ECMA international. *Standard ECMA-369* [online]. 3-rd edition. [2008] , December 2008 [cit. 2008-12-10]. Standart. Dostupný z WWW: <<http://www.ecma-international.org/publications/standards/Ecma-369.htm>>.
- [7] VAVRINA, Radovan. *Bluetooth: Káblom dávno odzvonilo* [online]. c2008 , posledná zmena 21.2.2007 [cit. 2008-12-10]. Dostupný z WWW: <<http://www.mobilmania.sk/Dalsierubriky/AR.asp?ARI=7741>>.
- [8] PUŽMANOVÁ, Rita. *Osobní síť – Bluetooth a IEEE 802.15* [online]. c1998-2008 , 14.5.2002 [cit. 2008-12-10]. Dostupný z WWW: <<http://www.lupa.cz/clanky/osobni-site-bluetooth-a-ieee-802-15/>>.
- [9] Bluetooth. *SIG Press Releases* [online]. c2008, 01-Dec-2008 [cit. 2008-12-10]. Dostupný z WWW: <<http://bluetooth.com/Bluetooth/Press/SIG>>.
- [10] PENN, Ivo. *Bezdrátová Bluetooth technologie* [online]. [2003] , [cit. 2008-12-10]. Dostupný z WWW: <http://fei1.vsb.cz/wofex/2003/paper/p2612/penn_ivo.pdf>.
- [11] Community Development of Java Technology Specification. *JSR 82: Java APIs for Bluetooth : Final Release 3* [online]. c1995-2008 [cit. 2008-12-10]. Dostupný z WWW: <<http://jcp.org/en/jsr/detail?id=82>>.
- [12] Palowireless. *Bluetooth Tutorial – Specification* [online]. [2007] [cit. 2008-12-10]. Dostupný z WWW: <<http://www.palowireless.com/INFOTOOTH/tutorial.asp>>.
- [13] URBAN, Robert. *Základní principy přenosu v sítích* [s.l.], 2007. 11 s. České vysoké učení technické v Praze. Semestrální práce. Dostupný z WWW: <http://radio.feld.cvut.cz/personal/mikulak/MK/MK07_semestralky/principy_prenosu_UWB.pdf>.

- [14] PUŽMANOVÁ, Rita. *UltraWideBand podruhé aneb mocenské spory* [online]. c1998-2008 ,29.4.2004 [cit. 2008-12-10]. Dostupný z WWW: <<http://www.lupa.cz/clanky/ultrawideband-podruhe-aneb-mocenske-spory/>>.
- [15] IEEE 802.15 Working Group for WPAN, *802.15 Documents* [online]. c2008 , modified: 14-Nov-2008 [cit. 2008-12-10]. Dostupný z WWW: <<http://ieee802.org/15/index.html>>.
- [16] IEEE 802.15 Wireless Personal Area Network. *DS-UWB Physical Layer Submission to 802.15 Task Group 3a* [online]. [2005] , september 2005 [cit. 2008-12-10] Dostupný z WWW: <<http://www.decawave.com/15-04-0137-05-003a-merger2-proposal-ds-uwband-update.doc>>.
- [17] HANUDEL, Christopher. *Wireless Universal Serial Bus* [online]. [2007] [cit. 2008-12-10]. Dostupný z WWW: <<http://www.clarkson.edu/class/cs463/wireless.sp2007/students/WirelessUSB.ppt>>.
- [18] GUTIERREZ, Jose. *IEEE P802.15 Working Group for Wireless Personal Area Network (WPANs)* [online]. [2003] , january – 2003 [cit. 2008-12-10]. Dostupný z WWW: <http://grouper.ieee.org/groups/802/15/pub/2003/Jan03/03036r0P802-15_WG-802-15-4-TG4-Tutorial.ppt>.
- [19] IEEE Standards Association. *802.15.4* [online]. c2008 , approved 12 May 2003 [cit. 2008-12-10]. Standart. Dostupný z WWW: <<http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>>.
- [20] KOTON, Jaroslav, ČIKA, Petr, KŘIVÁNEK, Vítězslav. *Standard nízkorychlostní bezdrátové komunikace ZigBee* [online]. [2008] , 18.04.2006 [cit. 2008-12-10]. Dostupný z WWW: <<http://access.feld.cvut.cz/view.php?cisloclanku=2006032001>>.
- [21] GOPI, Rajesh. *Introduction to Ultra Wide Band* [online]. [2006] , Aug 2006 [cit. 2008-12-10] Dostupný z WWW: <<http://www.nitw.ac.in/department/cse/archive/ubw.ppt>>.
- [22] CHAIRMAN, Bob Heile. *ZigBee Alliance Tutoriál* [online]. [2005] , September-November 2005 [cit. 2008-12-10]. Dostupný z WWW: <http://www.cs.ucdavis.edu/~aksoy/course/w06/slides/ZigBeeTutorial_05.ppt>.
- [23] Bluetooth. *Compare with Other Technologies* [online]. c2008 [cit. 2008-12-10]. Dostupný z WWW: <<http://bluetooth.com/Bluetooth/Technology/Works/Compare/>>.
- [24] GRIFFITH, Eric. *Freescall Quits UWB Forum* [online]. c2008 , April 7, 2006 [cit. 2008-12-10]. Dostupný z WWW: <<http://www.ultrawidebandplanet.com/resources/article.php/3597666>>.
- [25] WiMedia Alliance. *First Dual-Band UWB PHYs Achieve WiMedia Registration* [online]. [2008] , March 13, 2008 [cit. 2008-12-10]. Dostupný z WWW: <www.wimedia.org/imwp/idms/popups/pop_download.asp?contentID=12809>.

- [26] GLYN, Roberts. *Developing Innovative Application for high Speed Bluetooth Technology* [online]. c2006 , 31 October 2007 [cit. 2008-12-10]. Dostupný z WWW: <www.wimedia.org/imwp/idms/popups/pop_download.asp?contentID=12413>.
- [27] WiMedia Alliance. *WiMedia UWB – Reality* [online]. c2007 [cit. 2008-12-10]. Dostupný z WWW: <www.wimedia.org/imwp/idms/popups/pop_download.asp?contentID=12329>.
- [28] WiMedia Alliance, *How it Works: UWB, WPAN and WiMedia Radio Space* [online]. c2008 [cit. 2008-12-10]. Dostupný z WWW: <www.wimedia.org/imwp/idms/popups/pop_download.asp?contentID=13836>.
- [29] Úvod do Java 2 Micro Edition [online] [cit 2008-12-10]. Dostupný z WWW: <<http://www.volny.cz/orlmar/Uvod.htm>>.
- [30] Sun Microsystems, Inc.. *J2ME Building Blocks for Mobile Devices* [online]. [2000] [cit. 2008-06-01]. Dostupný z WWW: <<http://java.sun.com/products/cldc/wp/KVMwp.pdf>>.
- [31] NetBeans Community. *NetBeans IDE* [počítačový program]. [cit. 2009-04-12]. Ver. 6.5.1. Dostupný z WWW <<http://www.netbeans.org/downloads/index.html>>. Vyžaduje Windows, Linux, Mac OS X, Solaris.

Zoznam použitých skratiek

| | |
|---------|--|
| 16QAM | (16 Quadrature Amplitude Modulation) → 16 stavová diskretná kvadratúrna modulácia |
| ACL | (Asynchronous Connection-Less) → asynchrónne bezspojovo orientované |
| API | (Application Programming Interface) → programové rozhranie aplikácie |
| AWT | (Abstract Windows Toolkit) → knižnica pre tvorbu grafického užívateľského rozhrania |
| BER | (Bit Error Rate) → bitová chybovosť |
| BP | (Beacon Period) → perióda beacon |
| BPSK | (Binary Phase Shift Keying) → binárne kľúčovanie s fázovým posuvom |
| BPST | (Beacon Period Start Time) → začiatok periódy Beacon |
| CAP | (Contention Access Period) → konkurenčná prístupová doba |
| CDC | (Connected Device Configuration) → konfigurácia v prostredí J2ME |
| CE | (Consumer Electronics) → spotrebnú elektroniku |
| CFP | (Contention Free period) → časová perióda alokovaných kanálov |
| CISC | (Complex Instruction Set Computer) → počítač s úplnou inštrukčnou sadou |
| CLDC | (Connected Limited Device Configuration) → konfigurácia v prostredí J2ME |
| CSMA/CA | (Carrier Sense Multiple Access/Colision Detecton) → viacnásobný prístup s odposluchom nosného kmitočtu a detekciou kolízií |
| CVM | (Compact Virtual Machine) → javovský virtuálna stroj určený pre CDC |
| DAA | (Detect And Avoid) → detekovanie a vyhnutie sa vysielaniu určitých zariadení |
| DECT | (Digital Enhanced Cordless Telecommunications) → digitálne bezšnúrové telekomunikácie |
| DEV | (Device) → zariadenie |
| DRP | (Distributed Reservation Protocol) → protokol určený na rezerváciu slotov prístupu k médiu |
| DS-CDMA | (Direct-Sequence Code-Division Multiple Access) → priame rozprestretie s kódom rozdeleným viacnásobným prístupom |
| DSSS | (Direct Sequence Spread Spectrum) → technika priameho rozprestretia spektra |
| DTP | (Data Transfer Period) → doba počas, ktorej sa prenášajú dáta |
| DVI | (Digital Video Interactive) → Interaktívne digitálne video |
| ECMA | (European Computer Manufacturers Association) → štandardizačná organizácia pre informačné a komunikačné systémy |
| EDGE | (Enhanced Data Rates for GSM Evolution) → umožňuje zvýšiť prenosovú rýchlosť systému GSM pri alokovaní všetkých 8 timeslotov |
| FFD | (Full Function Device) → plne funkčné zariadenie |
| FFI | (Fixed Frequency Interleaving) → stály frekvenčné prekladanie |
| FHSS | (Frequency Hopping Spread Spectrum) → rozprestretie spektra kmitočtovým skákaním nosnej vlny |
| FPP | (File Transfer Profile) → profile prenosu súboru |
| FTP | (File Transfer Protocol) → protokol určený pre prenos súborov |
| GAP | (Generic Access Profile) → základný profil, zaisťujúci kompatibilitu |
| GPRS | (General Packet Radio Services) → prenos dátových paketov cez rádiové rozhranie |
| GTS | (Guaranteed Time Slots) → rezervované časové sloty |

| | |
|---------|--|
| HDMI | (High-Definition Multi-media Interface) → digitálne rozhranie používané medzi zdrojom audio a video signálu a televíziou |
| HDTV | (High Definition TV) → digitálna televízia s vysokým rozlíšením |
| HSCSD | (High Speed Circuit Switched data) → umožňuje prenos dát v sieti GSM vyšou rýchlosťou bez hardwarového zásahu do jej štruktúry |
| ICI | (Inter-Channel Interference) → medzikanálová interferencia |
| IEEE | (The Institute of Electrical and Electronics Engineers) → inštitút pre elektrotechnické a elektronické inžinierstvo |
| IFFT | (Inverse Fast Fourier transform) → inverzná rýchla fourierová transformácia |
| IrDA | (Infrared Data Association) → asociácia pre infračervený prenos |
| ISI | (Inter-Symbol Interference) → medzi-symbolová interferencia |
| ISO | (International Organization for Standardization) → medzinárodná štandardizačná organizácia |
| ISP | (Internet Service Provider) → poskytovateľ internetových služieb |
| JRE | (Java Runtime Environment) → prostredie pre beh Javy |
| KVM | (Kilo Virtual machine) → javovský virtuálny stroj určený pre CLDC |
| L2CAP | (Logical Link Control and Adaptation Protocol) → protokol vrstvy dátových spojení |
| LLC | (Logical Link Control) → riadenie logického spoja |
| LMP | (Link Manager Protocol) → protokol spojovej vrstvy |
| MAC | (Medium Access Control) → Riadenie prístupu k médiu |
| MAS | (Medium Access Slot) → slot prístupu k médiu |
| MBOA | (MultiBand OFDM Alliance) → názov skupiny |
| MB-OFDM | (MultiBand Ortoogonal Frequency Division Multiplexing) → mutlipásmový ortogonálny multiplex s kmitočtovým delením |
| MIDP | (Mobile Information Device Profile) → profil poskytujúci API v zariadeniach ako je mobilný telefón |
| MIDP | (Mobile Information Device Profile) → profil poskytujúci API |
| MTS | (Management Time Slots) → riadiace časové sloty |
| OEM | (Original Equipment Manufacturer) → vybavenie definované výrobcom |
| OFDM | (Ortoogonal Frequency Division Multiplexing) → ortogonálny multiplex s kmitočtovým delením |
| O-QPSK | (Offset – Quadrature Phase Shift Keying) → ofsetová modulácia QPSK |
| PAC | (Piconet Acquisition Codeword) → získané pikosiet'ové kódové slovo |
| PAL | (Protocol Adaptation Layer) → protokolovo adaptačná vrstva |
| PCA | (Prioritized-based Channel Access) → konkurenčný |
| PDA | (Personal Digital Assistants) → vreckový počítač |
| PHY | (Physical) → fyzická vrstva |
| PNC | (Piconet Coordinator) → koordinátor pikosiete |
| PPDU | (Physical Protocol Data Unit) → dátová jednotka protokolu fyzickej vrstvy |
| PRN | (Pseudo Random Code) → pseudonáhodný kód |
| QoS | (Quality of Services) → kvalita služieb |
| QPSK | (Quadrature Phase Shift Keying) → štvorstavové fázové kľúčovanie |
| RFCOMM | (Radio Frequency Communication) → protokol emuláciu sérového portu |
| RFD | (Reduced Fncion Device) → zariadenie s redukovanými funkciami |

| | |
|------|---|
| RISC | (Reduced Instruction Set Computer) → počítač s redukovanou inštrukčnou sadou |
| SCO | (Synchronous Connection-Oriented) → synchronne spojovo orientované |
| SIG | (Special Interest Group) → zájmová skupina |
| SP | (Synchronization Profile) → synchronizačný profil |
| SPP | (Serial Port Profile) → profil emulácie sériovej linky |
| TDMA | (Time Division Multiple Access) → viacnásobný prístup s časovým delením |
| TFC | (Time-Frequency Codes) → časovo-frekvenčné kódy |
| TFI | (Time-Frequency Interleaving) → časovo-frekvenčné prekladanie |
| UART | (Universal asynchronous receiver/transmitter) → univerzálni asynchrónny prijímač a vysielateľ |
| USB | (Universal Serial Bus) → univerzálna sériová zbernica |
| UWB | (Ultra-Wide Band) → technológia vysielania signálu rozprestretého v šírke pásma až niekoľkých GHz |
| WiFi | (Wireless Fidelity) → bezdrôtová sieť |
| WLAN | (Wireless Local Area Network) → bezdrôtová miestna sieť |
| WLP | (WiMedia Link Layer Control protocol) → protokol riadenie logického spoja od WiMedie |
| WPAN | (Wireless Personal Area Network) → bezdrôtová osobná sieť |
| WXP | (WiMedia eXtended Platform) → rozširujúca platforma od WiMedie |

Zoznam príloh

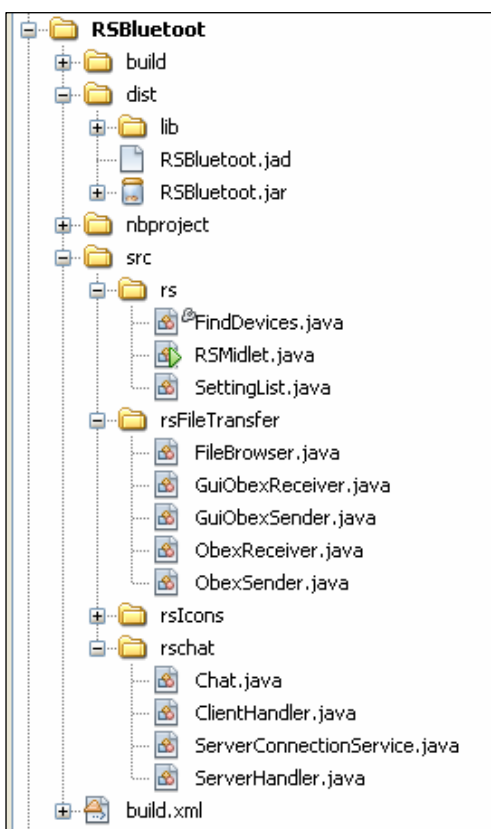
| | | |
|----------|-----------------------------------|-----------|
| A | OBSAH PRILOŽENÉHO DVD..... | 82 |
|----------|-----------------------------------|-----------|

A Obsah priloženého DVD

| | |
|--------------------------------------|---|
| ./diplomova_praca.pdf | - text diplomovej práce v elektronickej podobe |
| ./netbeans-6.5.1-ml-java-windows.exe | - open-source vývojové prostredie [31] |
| ./jdk-6u13-windows-i586-p.exe | - súbor základných nástrojov pre vývoj aplikácií pre platformu Java |
| ./RSBluetooth/* | - vytvorená aplikácia |

Pre kompiláciu aplikácie je doporučené použiť program *J2ME Wireless Toolkit 2.5.2*, v ktorom sa dá aplikácia taktiež emulovať (emulácia mobilného telefónu). Tento program je súčasťou vývojového prostredia *NetBeans 6.5.1* dostupný z [31] na platforme J2ME, ktorý bol využitý pri vývoji (pre operačný systém Windows, dátum stiahnutia verzie -12.4.2009). Toto programové vybavenie je tiež k použitiu na priloženom DVD (./netbeans-6.5.1-ml-java-windows.exe), ktoré je súčasťou diplomovej práce.

Adresár RSBluetooth obsahuje všetko potrebné pre úspešnú kompiláciu. Jedná sa o zdrojové súbory (./RSBluetooth/src/*.java); obrázky použité v aplikácii (./RSBluetooth/src/rsIcons/*.png); JAD súbor - Java Application Descriptor (./RSBluetooth/dist/RSBluetooth.jad) a skomprimovaný *jar* súbor (./RSBluetooth/dist/RSBluetooth.jar), ktorý obsahuje preložené *.class súbory, obrázky a iné podporné súbory, ktoré sú nutné k chodu aplikácie. Tento súbor je určený pre nahratie do mobilného telefónu.



Adresárová štruktúra projektu RSBluetooth.