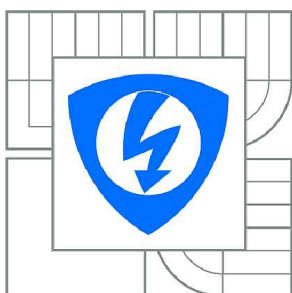


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

FYZICKÁ VRSTVA SYSTÉMU WIMAX

WIMAX PHYSICAL LAYER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

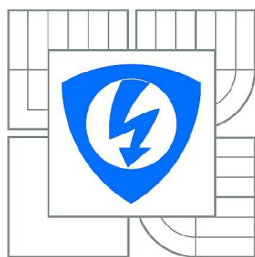
PETR VÍTEČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ MILOŠ

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Petr Víteček

ID: 125699

Ročník: 3

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Fyzická vrstva systému WiMAX

POKYNY PRO VYPRACOVÁNÍ:

Detailně prostudujte bezdrátovou technologii WiMAX. Zaměřte se na zpracování signálu ve fyzické vrstvě technologií OFDMA. Navrhněte a sestavte matematický model vysílače a přijímače systému WiMAX ve směru uplink. K realizaci použijte programové prostředí MATLAB

Po ověření funkčnosti modelu proveďte simulaci přenosu signálu i v rádiovém prostředí s nastavitelným C/N resp. C/I. Vyšetřete závislost chybovosti BER přenosu na poměrech C/N, resp. C/I.

DOPORUČENÁ LITERATURA:

[1] IEEE Std 802.16-2009, IEEE DRAFT Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems, IEEE, 2009, 2004s, ISBN 978-0-7381-5919-5 STD95914

[2] DOBEŠ, J., ŽALUD, V. Moderní radiotechnika. 1. vyd. Praha : BEN - technická literatura, 2006. 767 s. ISBN 80-730-0132-2.

Termín zadání: 6.2.2012

Termín odevzdání: 25.5.2012

Vedoucí práce: Ing. Jiří Miloš

Konzultanti bakalářské práce:

prof. Dr. Ing. Zbyněk Raida

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zabývá přenosovým systémem čtvrté generace WiMAX a modelováním jeho fyzické vrstvy ve směru uplink. V jednotlivých částech se blíže podíváme na samotný systém WiMAX, schéma vícenásobného přístupu OFDMA a modulaci OFDM. Dále na zpracování dat fyzickou vrstvou a v posledním bodě na její model v prostředí MATLAB. Výsledkem práce má být daný model fyzické vrstvy, který společně s modelem rádiového prostředí bude sloužit k vykreslování grafů chybovosti BER v závislosti na poměru signál/šum, a zjištění jak danou chybovost ovlivňují různě nastavené části modelu, jako jsou použité modulace, kódový poměr ochranných kódů a jejich typ, použité prokládání.

KLÍČOVÁ SLOVA

WiMAX, OFDM, OFDMA, Fyzická vrstva, MATLAB

ABSTRACT

This thesis deals with 4G communication system WiMAX and his model of physical layer in MATLAB in uplink direction. In individual parts of this thesis we will have closer look on WiMAX, multiple access scheme OFDMA and OFDM modulation. Further, the physical layer signal processing and as last the model of physical layer in MATLAB. Result of this thesis should be model of physical layer , which will be together with model of radio environment used to calculate graphs of BER depending on S/N ratio. The model should show how is BER affected by different parts of model such as used modulation, code rate of FEC code and used type of interleaving.

KEYWORDS

WiMAX, OFDM, OFDMA, Physical layer, MATLAB

VÍTEČEK, P. *Fyzická vrstva systému WiMAX*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2011. 42 s., 11 s. příloh. Semestrální projekt. Vedoucí práce: Ing. Jiří Miloš.

PROHLÁŠENÍ

Prohlašuji, že svůj semestrální projekt na téma *Fyzická vrstva systému WiMAX* jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu 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 uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto semestrálního projektu práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu semestrálního projektu Ing. Jiřímu Milošovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého semestrálního projektu.

V Brně dne

.....

(podpis autora)

OBSAH

OBSAH	IV
SEZNAM OBRÁZKŮ	VI
SEZNAM TABULEK	VIII
ÚVOD	1
1 WIMAX	2
1.1 STANDARDY	2
1.2 REFERENČNÍ MODEL SÍŤE 802.16	3
1.2.1 <i>Service-Specific Convergence Sublayer</i>	3
1.2.2 <i>MAC Common Part Sublayer</i>	3
1.2.3 <i>Security Sublayer</i>	4
1.2.4 <i>Fyzická vrstva</i>	4
1.3 ARCHITEKTURA SÍŤE	4
2 ORTHOGONAL FREQUENCY DIVISION MULTIPLE ACCESS	6
2.1 OFDM	6
2.2 MOŽNOSTI VÍCENÁSOBNÉHO PŘÍSTUPU	7
2.2.1 <i>Frequency Division Multiple Access</i>	8
2.2.2 <i>Time Division Multiple Access</i>	8
2.3 ORTHOGONAL FREQUENCY DIVISION MULTIPLE ACCESS	8
2.4 SOFDMA	10
2.5 OFDMA VYSÍLAČ A PŘIJÍMAČ	11
3 FYZICKÁ VRSTVA SYSTÉMU WIMAX	12
3.1 FYZICKÁ VRSTVA	12
3.2 RANDOMIZACE	12
3.3 KANÁLOVÉ KÓDOVÁNÍ	13
3.3.1 <i>RS kódy</i>	13
3.3.2 <i>Konvoluční kódy</i>	14
3.3.3 <i>Turbo kódy</i>	15
3.4 BITOVÉ PROKLÁDÁNÍ	16
3.4.1 <i>Pseudonáhodné prokládání</i>	16
3.4.2 <i>Maticové prokládání</i>	16
3.4.3 <i>Konvoluční prokládání</i>	16
3.5 REPETITION (OPAKOVÁNÍ)	16
3.6 DIGITÁLNÍ MODULACE	17
3.6.1 <i>Modulace BPSK</i>	17

3.6.2 QPSK	18
3.6.3 M-QAM	18
3.6.4 DFT a FFT	20
3.6.4 Raised Cosine Filtr	22
4 MODEL FYZICKÉ VRSTVY	24
4.1 MODEL VYSÍLAČE	24
4.1.1 Randomizace	24
4.1.2 Konvoluční kódování	25
4.1.3 RS kódování	25
4.1.4 Prokládání	26
4.1.5 Modulace	27
4.1.6 IFFT	29
4.1.7 Výstupní filtr	30
4.2 MODEL PŘIJÍMAČE	31
4.2.1 Vstupní Filtr	31
4.2.2 FFT	31
4.2.3 Demodulace	31
4.2.4 Opačné prokládání	32
4.2.5 RS dekódování	32
4.2.6 Konvoluční Dekódování	32
4.2.7 DeRandomizace	33
4.3 PROSTŘEDÍ PRO SIMULACE	33
4.4 VÝSLEDKY	33
4.4.1 Chybovost pro AWGN	34
4.4.2 Chybovost pro Rayleighův kanál	36
4.4.3 Chybovost pro Riciův kanál	37
4.4.4 Chybovost pro různé RS kodéry	37
4.4.5 Chybovost pro různé rate CC kodéry	38
5 ZÁVĚR	39
LITERATURA.....	40
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK.....	41
SEZNAM PŘÍLOH	43

SEZNAM OBRÁZKŮ

Obrázek 1.1: Referenční model standardu 802.16, převzato z [1]	3
Obrázek 1.2: Architektura sítě WiMAX, převzato z [4].....	5
Obrázek 2.1: Schéma OFDM vysílače, převzato z [3]	6
Obrázek 2.2: Spektrum OFDM signálu, převzato z [4].....	7
Obrázek 2.3: Oddělení jednotlivých účastníků v FDMA, převzato z [4]	8
Obrázek 2.4: Oddělení jednotlivých účastníků v TDMA, převzato z [4]	8
Obrázek 2.5: Schéma OFDMA vysílače, převzato z [3].....	9
Obrázek 2.6: Srovnání přístupu v OFDM a OFDMA, převzato z [11]	9
2.7 Struktura subnosných u OFDMA	10
2.8 Ukázka cyklického prefixu	10
Obrázek 2.9: a) Schéma OFDMA vysílače, b) schéma OFDMA přijímače, převzato z [4]	11
Obrázek 3.1 Proces zpracování dat fyzickou vrstvou, převzato z [3].....	12
Obrázek 3.2 Schéma PRBS generátoru použitého pro randomizaci, převzato z [1]	13
Obrázek 3.3 Princip blokového kódu ,viz [9]	14
Obrázek 3.4 Schéma konvolučního kodéru s kódovým poměrem 2/3	15
Obrázek 3.5 Principiální schéma turbo kodéru [10]	15
Obrázek 3.6 Konstelační diagram modulace QPSK, převzato z [2].....	18
Obrázek 3.7 Konstelační diagram funkce 16-QAM, převzato z [2].....	19
Obrázek 3.8 Operace vyjádřené pomocí grafů signálových toků	21
Obrázek 3.9 Algoritmus FFT typu DIT pro $N = 4$	21
Obrázek 3.10 Přenosová charakteristika filtru Raised Cosine.....	23
Obrázek 3.11 Impulsní charakteristika filtru Raised Cosine	23
Obrázek 4.1 Schéma OFDMA vysílače.....	24
Obrázek 4.2 Schématická značka hradla xor	24
Obrázek 4.3 Konstelační diagram modulace 64-QAM před normováním	28
Obrázek 4.4 Konstelační diagram modulace 64-QAM po normování	28
Obrázek 4.5 Spektrum na výstupu vysílače vytvořené pomocí IFFT.....	29
Obrázek 4.6 Přenosová charakteristika vytvořeného filtru.....	30
Obrázek 4.7 Binární signál délky 100 bitů po průchodu daným filtrem	30
Obrázek 4.8 Schéma OFDMA přijímače.....	31
Obrázek 4.9 Chybovost pro modulaci BPSK	34

Obrázek 4.10 Chybovost pro modulaci QPSK	34
Obrázek 4.11 Chybovost pro modulaci 16-QAM.....	35
Obrázek 4.12 Chybovost pro modulaci 64-QAM.....	35
Obrázek 4.13 Chybovost pro Rayleighův kanál	36
Obrázek 4.14 Chybovost pro Ricianův kanál	37
Obrázek 4.15 Chybovost pro různé RS kódy	37
Obrázek 4.16 Chybovost pro různé rate CC	38

SEZNAM TABULEK

Tabulka 1 Porovnání standardů 802.16, převzato z [1]	2
Tabulka 2 Hlavní části architektury systému WiMAX, převzato z [4].	4
Tabulka 3 Parametry SOFDMA	11
Tabulka 4: Porovnání jednotlivých fyzických vrstev a jejich vlastností, převzato z [1]	12
Tabulka 5 Pravdivostní tabulka funkce xor	24
Tabulka 6 Vlastnosti Modulátoru	27
Tabulka 7 Hodnoty koeficientů normování pro použité modulace	27
Tabulka 8 Vlastnosti jednotlivých kanálů	33
Tabulka 9 Nastavení modelu při simulacích v AWGN kanálu	36

ÚVOD

WiMAX je komunikační systém čtvrté generace založený na standardu IEEE 802.16. Má poskytovat vysokou datovou rychlost při pokrytí rozsáhlé oblasti. Zkratka WiMAX znamená Worldwide Interoperability for Microwave Access. Technologie WiMAX má splnit potřeby uživatelů, kteří chtějí mít vysokorychlostní připojení levně a bez nutnosti pořízení kabelového připojení. Dále je výhodný do oblastí kde není možné kabelové připojení kvůli vzdálenosti a nákladům.

Celkově se jedná o technologii, která je komplementární k IEEE 802.11 Wi-Fi, IEEE 802.16 však definuje standard pro sítě typu MAN (Metropolitan area network), které mohou poskytovat služby více uživatelům a na větší vzdálenost.

Cílem této bakalářské práce je vytvořit v programovém prostředí Matlab model fyzické vrstvy systému WiMAX. Následně spolu s modelem rádiového prostředí spočítat závislost chybovosti (BER) na odstupu signálu od šumu (SNR) pro různá nastavení modelu.

1 WiMAX

WiMAX začal vznikat v devadesátých letech dvacátého století. S narůstajícím datovým tokem v pevných telekomunikačních sítích a jejich cenou, zvláště v odlehlých oblastech a v mnoha zemích, kde žádné nebyly, začaly být zkoumány metody bezdrátové komunikace.

Zde začal projekt nazvaný „The last mile connectivity“, který měl za úkol navrhnout způsob poskytování vysoké datové rychlosti pro velký počet uživatelů, kteří nemusí mít existující kabelové připojení.

Dalším významným krokem byl vývoj standardu IEEE. První skupina byla založena v roce 1999 a první 802.16 standard byl schválen v roce 2001, tento standard byl následován dvěma dalšími známými jako 802.16a a 802.16c.

V září 2003 byla zahájena zásadní revize projektu. Ta měla za cíl sladit normy s evropským ETSI HiperMAN standardem. Projekt byl dokončen v roce 2004 a standard byl vydán jako 802.16d, ačkoli t je často označován jako 802.16-2004 s ohledem na datum vydání. S vydáním tohoto standardu byly předchozí 802.16 dokumenty, zahrnující verze 802.16- a, b, a c staženy. V roce 2005 následoval standard 802.16e (802.16-2005).

1.1 Standardy

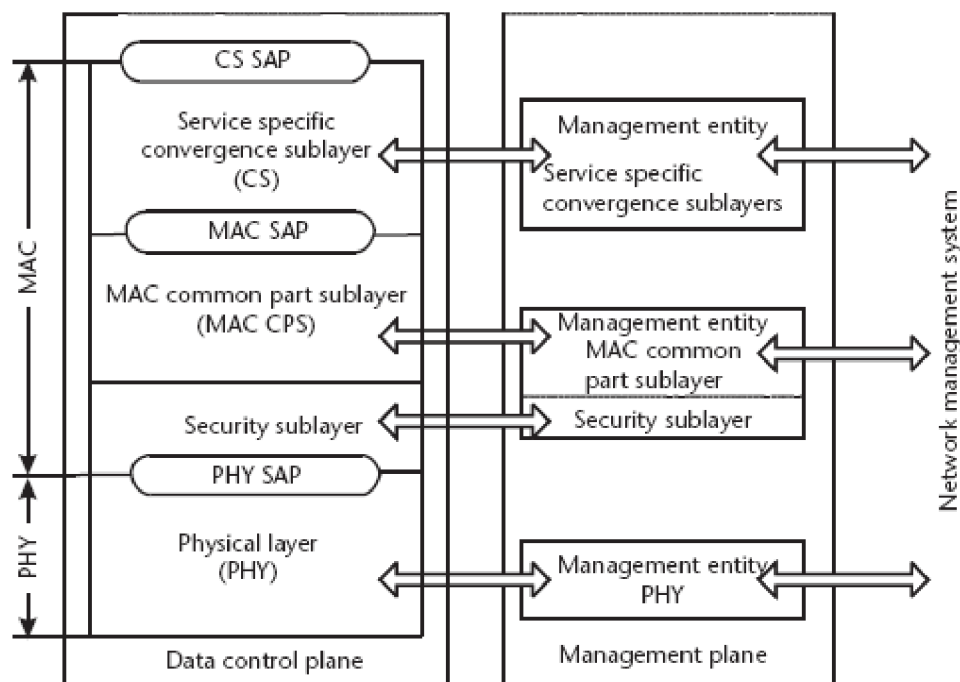
Jednotlivé standardy jsou uvedeny v tabulce č.1. V této práci se budeme zabývat standardem 802.16 – 2004, [1].

Tabulka 1: Porovnání standardů 802.16, převzato z [1]

Standardy	802.16-2001	802.16 a	802.16 - 2004, 16d	802.16 e
Šířka pásma	10 - 66 GHz, NLOS	2 - 11 GHz, NLOS, 10 - 66 GHz, LOS	2 - 11 GHz, NLOS, 10 - 66 GHz, LOS	2 - 11 GHz, NLOS
Fyzická vrstva	SC	Sca, OFDM, OFDMA	SC, Sca, OFDM, OFDMA	Sca, OFDM, OFDMA
Duplex	TDD, FDD	TDD, FDD	TDD, FDD	TDD, FDD
Mobilita	Fixní	Fixní	Fixní	Mobilní
Datum vydání	Duben 2002	Duben 2003	Říjen 2004	Únor 2006

1.2 Referenční model sítě 802.16

Referenční model se obvykle používá pro naznačení komunikace v dané síti či systému. Na rozdíl od klasického modelu počítačových sítí, sestávajících ze sedmi vrstev, obsahuje WiMAX pouze vrstvy dvě.



Obrázek 1.1: Referenční model standardu 802.16, převzato z [1]

Na obrázku 1.1 lze vidět referenční model standardu 802.16, který se skládá z MAC vrstvy a Fyzické vrstvy. MAC vrstva se skládá ze tří podvrstev. První podvrstvou je Service-Specific Convergence Sublayer (CS), druhou podvrstvou je The Common Part Sublayer (CPS), a poslední podvrstvou je Security Sublayer (nebo Privacy Sublayer).

1.2.1 Service-Specific Convergence Sublayer

Poskytuje transformaci nebo mapovací funkci na data externích sítí přijatá přes CS SAP, která jsou potom poslána do MAC CPS skrze MAC SAP. To zahrnuje klasifikaci jednotky externí sítě a přiřazení správné služby MAC vrstvy a identifikaci spojení. Může obsahovat i další funkce.

1.2.2 MAC Common Part Sublayer

Tato podvrstva je hlavní částí MAC vrstvy, která podporuje všechny možné typy služeb. Provádí základní operace jako přístup k systému, přidělování pásma, vytvoření spojení a udržování spojení. Vzhledem k tomu, že 802.16 používá sdílené médium pro poskytování služby více uživatelům, MAC CPS poskytuje mechanismus, který umožňuje všem uživatelům sdílet toto médium efektivně.

1.2.3 Security Sublayer

Bezpečnostní podvrstva má za úkol poskytovat soukromí, ověřování nebo utajení skrze spojení. Zabezpečení taktéž poskytuje silnou ochranu před neoprávněným přístupem. Hlavními dvěma komponenty pro zabezpečení jsou protokol zapouzdření , který slouží pro zabezpečení datových paketů v síti a protokol pro správu klíčů, který zabezpečuje distribuci klíčových dat z MS do BS.

1.2.4 Fyzická vrstva

Hlavním aspektem fyzické vrstvy, kterou se budeme zabývat, je zpracování signálů. Tedy kódování, modulace a jiné procesy. Fyzická vrstva standardu 802.16 bude detailně rozebrána v kapitole č.3

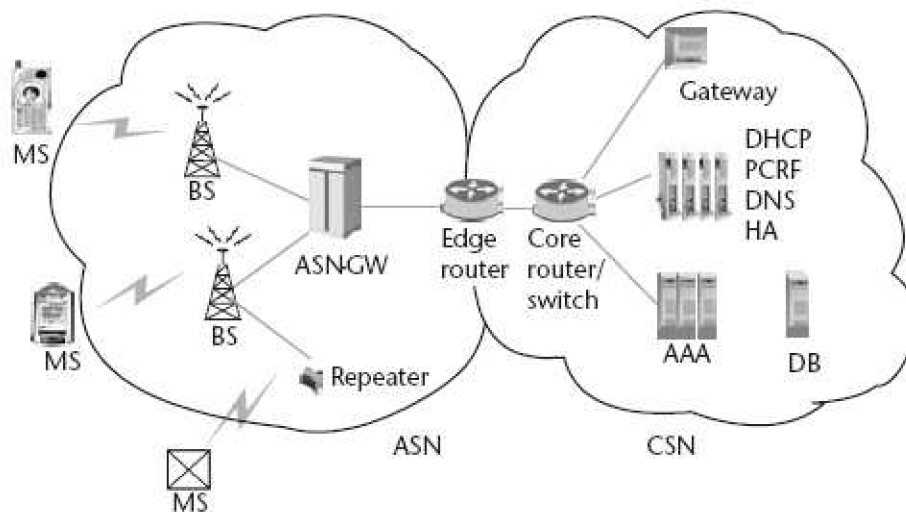
1.3 Architektura sítě

Základní norma 802.16 nedefinuje architekturu sítě. Nicméně je zapotřebí definovat standardní architekturu sítě WiMAX .To bylo realizováno pomocí pracovní skupiny WiMAX fórum, která vyvinula standard pro definování síťové architektury[16].

Architekturu sítě je možno popsat třemi hlavními částmi:

Tabulka 2 Hlavní části architektury systému WiMAX, převzato z [4].

Název	Zkratka	Funkce
Mobilní stanice	MS	Zařízení účastníka, které používá pro přístup k síti
Síť přístupových služeb	ASN	Tato oblast formuje radiové spojení se sítí a zahrnuje jednu nebo více MS a jednu nebo více ASN bran
Síť spojovacích služeb	CSN	Tato část sítě WiMAX poskytuje IP konektivitu a všechny IP funkce páteřní sítě.



Obrázek 1.2: Architektura sítě WiMAX, převzato z [4]

Celková síť WiMAX se skládá z řady objektů, které tvoří hlavní oblasti popsané výše. Tyto objekty jsou následující:

- Subscriber Station/Mobile Station (SS) – jsou často označeny jako předpokládané vybavení zákazníka (CPE). Mohou mít celou řadu forem a lze je brát jako vnitřní CPE nebo venkovní CPE. Venkovní CPE, má výhody díky lepšímu umístění antény. MS mohou být použity také, např. adaptér pro notebook.
- Basic Stations – Základnové stanice tvoří základní prvek sítě WiMAX. Jsou odpovědné za poskytování spojení. Také poskytuje další funkce, pokud jde o Mikro-Mobility Management funkce, jako je spouštění a předávka spojení zařízení, správa rádiových zdrojů, QoS , klasifikace provozu, DHCP (Dynamic Host Control Protocol) proxy, správu klíčů, Session Management a Multicast Group Management.
- ASN brána (ASN-GW) – Poskytuje funkce jako jsou paging, řízení přístupu, výběr účastnických profilů a šifrovacích klíčů.

Také může zahrnovat některé funkce AAA (Authentication, Authorization, Accounting server), jako zavedení a řízení mobilního spojení s BS, QoS, funkci FA pro mobilní IP a směrování na vybrané CSN.

- Home Agent – Home Agent slouží jako kotevní bod pro účastníky, zajišťuje roaming a funkce QoS. Home Agent funguje ve spojení s FA, jako je ASN Gateway, což je efektivnější řešení.
- AAA – Tak jako v komunikačním nebo bezdrátovém systému, i zde jsou potřeba služby autentizace, autorizace a účtování služeb, ty jsou zahrnuty do sítě spojovacích služeb (CSN).

2 ORTHOGONAL FREQUENCY DIVISION MULTIPLE ACCESS

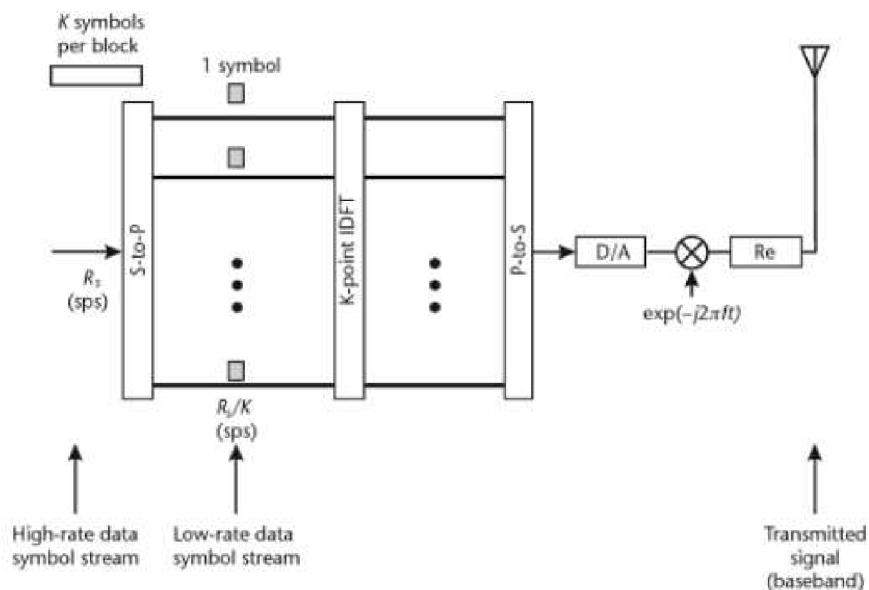
Jedná se o jednu nejmodernějších metod vícenásobného přístupu používanou v současných komunikačních systémech. Základem je OFDM modulace a její výhody.

2.1 OFDM

Orthogonal Frequency Division Multiplex je modulační formát, který dnes nachází velké uplatnění v různých aplikacích (Wi-Fi sítě, rozhlasové vysílání DAB, televizní vysílání DVB-T, sítě WiMAX) a je také používán další generací buňkových radiových sítí jako 3G LTE a UMB.

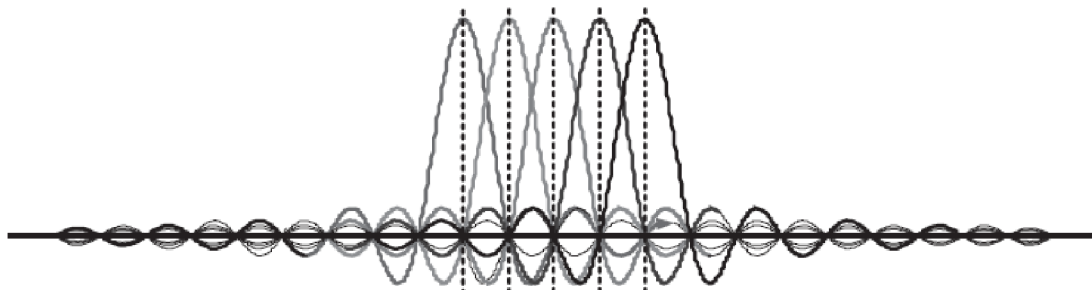
Hlavní myšlenkou OFDM je přenos dat za použití více subnosných. OFDM překonává problém potřeby velké šířky pásma, a to použitím matematického postupu DFT k vytvoření subnosných. Subnosné vytvořeny touto cestou nepotřebují další ochranná pásma a mohou být ve frekvenční oblasti mnohem blíže k sobě - ortogonalita.

Na začátku máme určitý datový tok s určitou přenosovou rychlostí R_S , kde každý symbol trvá $1/R_S$. Tento datový tok se skládá z určitých bloků a každý blok obsahuje K symbolů. Jelikož má být generováno právě K subnosných, rozdělí se datový tok pomocí sériově paralelního převodu na několik menších datových toků. Každý z nich má tedy přenosovou rychlost R_S/K . Po sériově-paralelním převodu tyto skupiny o K symbolech projdou inverzní DFT (IDFT) funkcí, která vytvoří dané spektrum. Po IDFT jsou data poslána skrz paralelně-sériový převodník. Blok K symbolů představuje jeden OFDM symbol. Po sobě jdoucí OFDM symboly jsou výstupem paralelně-sériového převodníku s přenosovou rychlostí R_S/K symbolů za sekundu, ale každý OFDM symbol trvá K/R_S .



Obrázek 2.1: Schéma OFDM vysílače, převzato z [3]

Na obrázku 2.2 je zobrazeno obecné spektrum OFDM signálu. Na rozdíl od FDM zde dochází ke znatelné úspoře šířky pásma díky ortogonalitě jednotlivých subnosných.



Obrázek 2.2: Spektrum OFDM signálu, převzato z [4]

Jelikož OFDM používá při přenosu více úzkopásmových subnosných je méně náchylné vůči ISI a vícecestným únikům. V podstatě má OFDM dvě hlavní výhody. První výhodou je použití FFT, tj. že nepotřebujeme k D/A převodníků a k oddělených modulátorů. Vše je nahrazeno IDFT funkcí ve vysílači a DFT funkcí v přijímači. Druhou výhodou je dosažení lepší spektrální účinnosti, než FDM, jelikož FDM vyžaduje ochranné intervaly mezi subnosnými, které OFDM nepotřebuje.

Lepší spektrální účinnost je umožněna díky ortogonalitě subnosných. To znamená, že každá subnosná je právě tak daleko od ostatních, aby se zabránilo v co největší míře interferenci. To umožňuje systémům s OFDM nastavit malý rozestup mezi subnosnými vlnami. Tato vzdálenost bývá obvykle definována jako $1/T$, kde T je doba trvání jednoho OFDM symbolu.

Nutností je použití lineárních zesilovačů ve třídě A, popřípadě dvojčinných zapojení ve třídě B, aby při zesilování signálu nedocházelo k narušení ortogonalitě subnosných vln.

2.2 Možnosti vícenásobného přístupu

V každém mobilním systému nebo technologii je nutné mít schéma, které umožňuje několika uživatelům získat přístup do daného přenosového kanálu a používat ho současně. S pokrokem mobilních technologií jsou používány různé techniky vícenásobného přístupu.

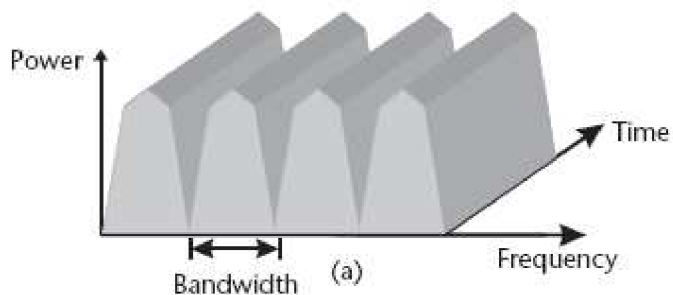
Existují čtyři hlavní techniky používané v mobilních systémech. Tyto schémata vícenásobného přístupu jsou známy jako FDMA, CDMA, TDMA a OFDM.

Požadavky, které musí schémata vícenásobného přístupu splňovat:

- Schopnost pracovat s více uživateli bez vzájemného rušení
- Schopnost maximalizace spektrální účinnosti
- Musí být robustní, umožňující snadný Handover mezi buňkami

2.2.1 Frequency Division Multiple Access

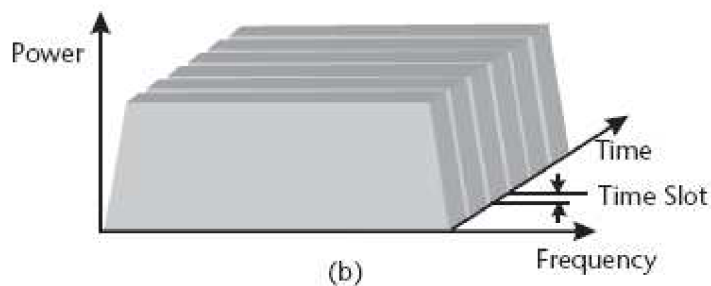
Je nejpřímější z technik pro vícenásobný přístup účastníků. Když účastník vstoupí do systému, popřípadě přejde z jedné buňky do druhé (tzv. handover), síť mu přidělí určitou frekvenci v kanálu. Tímto způsobem jsou různým účastníkům přiděleny různé sloty mající přístup do sítě. Jelikož jsou používány různé frekvence, nazývá se tato technika FDMA. Toto schéma bylo používáno všemi analogovými systémy.



Obrázek 2.3: Oddělení jednotlivých účastníků v FDMA, převzato z [4]

2.2.2 Time Division Multiple Access

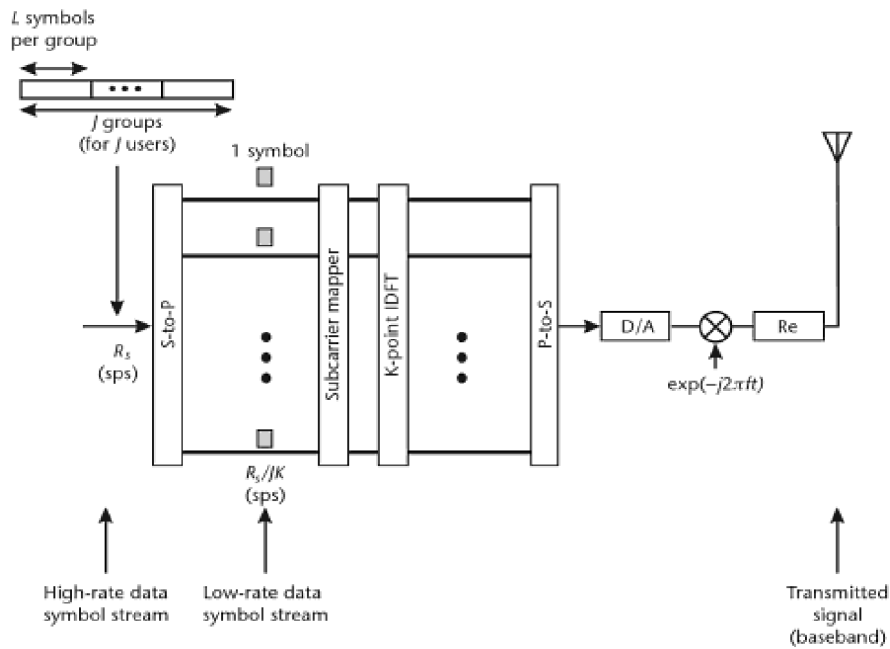
Tato technika vznikla s přechodem k digitálním mobilním systémům. Digitální data mohou být rozdělena v čase a odesílána jako dávky, shluky dat (tzv. bursty). V tomto případě jsou účastníkům přiděleny časové sloty, ve kterých mohou přijímat či vysílat data. Kvůli tomuto je možné mít v kanálu jen omezený počet uživatelů. Navíc zde může být použit i jiný kanál, takže systémy s TDMA mohou mít nějaké prvky FDMA operací. Tato smíšená schémata jsou používána například systémy GSM nebo UMTS.



Obrázek 2.4: Oddělení jednotlivých účastníků v TDMA, převzato z [4]

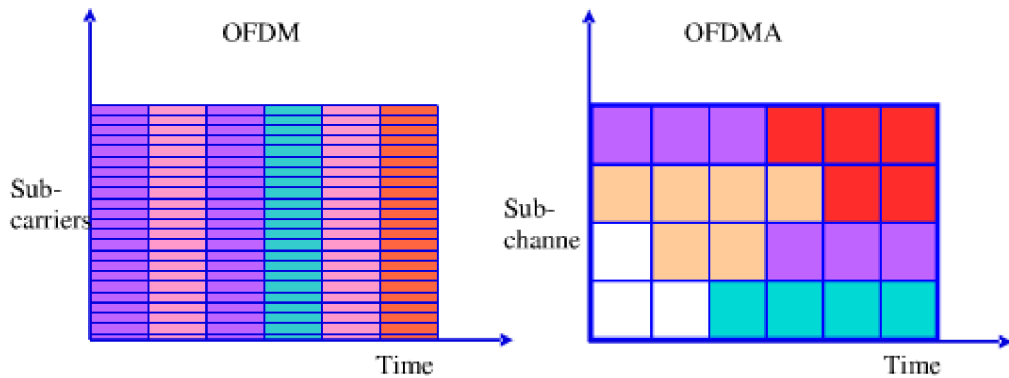
2.3 Orthogonal Frequency Division Multiple Access

OFDMA je metoda, která přiřazuje různé skupiny subnosných jednotlivým uživatelům. Touto cestou může mít více uživatelů přístup do systému ve stejný čas. V OFDMA namísto přiřazování OFDM symbolů v čase jednotlivým uživatelům, systém přímo přiřazuje dané subnosné jednotlivým uživatelům.



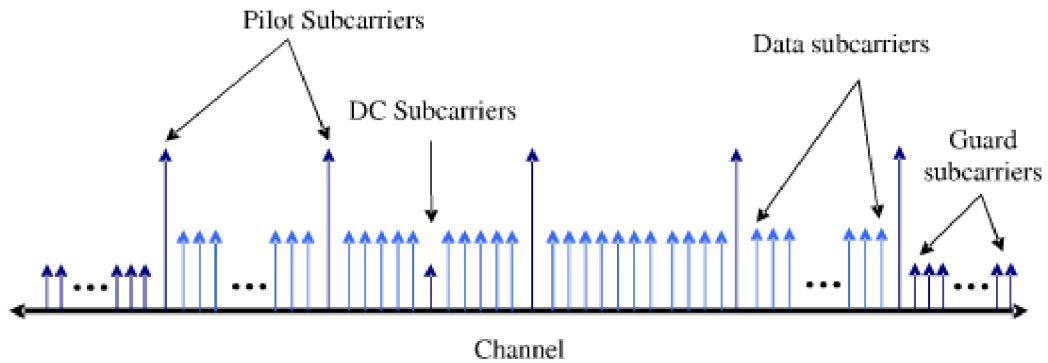
Obrázek 2.5: Schéma OFDMA vysílače, převzato z [3]

V OFDM může pouze jeden uživatel přenášet data v jednom timeslotu, v OFDMA může přenášet data více uživatelů v jednom timeslotu rozděleném na subnosné.



Obrázek 2.6: Srovnání přístupu v OFDM a OFDMA, převzato z [11]

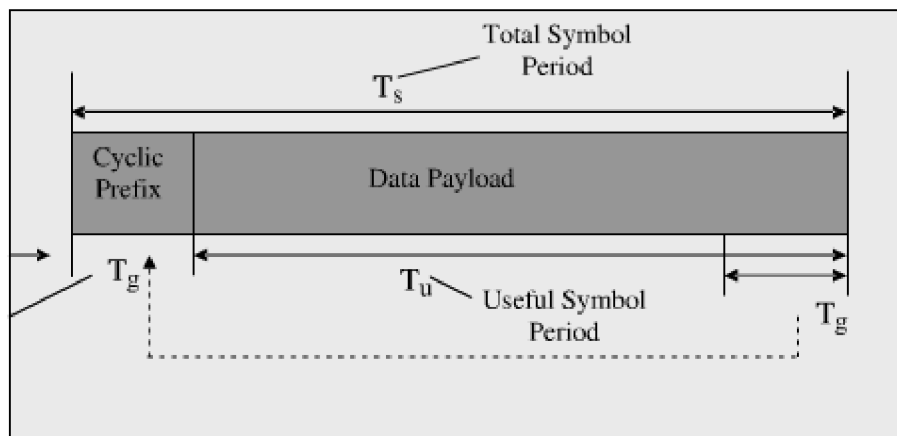
Lze vidět, že OFDM a OFDMA jsou strukturovány podobnou cestou. V OFDMA se každý symbol skládá ze sub-kanálů, které obsahují datové subnosné nesoucí informaci, pilotní subnosné sloužící jako referenční frekvence a pro různé odhady, DC subnosnou jako střední frekvenci a ochranné subnosné nebo ochranná pásma pro udržování odstupu mezi OFDMA signály, jak je naznačeno na obrázku [14].



2.7 Struktura subnosných u OFDMA

Aktivní subnosné (pilotní a datové) jsou rozděleny do podskupin subnosných, které se nazývají subkanály. Poté se definuje, které subkanály mohou být přiděleny účastnické stanici, to ale závisí na kondici kanálů a datových požadavcích.

Delší trvání symbolu u OFDMA zvyšuje zpoždění přenosu, zatímco ISI je kompletně eliminováno zavedením cyklického prefixu (CP). Cyklický prefix je opakování poslední části dat, která je připojena na začátek užitečných dat. ISI je kompletně eliminováno, dokud je délka CP větší, než je největší zpoždění šíření signálu v přenosovém prostředí [14].



2.8 Ukázka cyklického prefixu

2.4 SOFDMA

Scalable Orthogonal Frequency Division Multiple Access je verze OFDMA používaná v systému 802.16e (Mobile WiMAX). Škálovatelnost je podporována nastavením velikosti FFT, zatímco je pevně nastavena vzdálenost subnosných na 10,94 kHz. To umožňuje změnu šířky pásma, která se nyní pohybuje v rozmezí 1,25 MHz až 20 MHz.

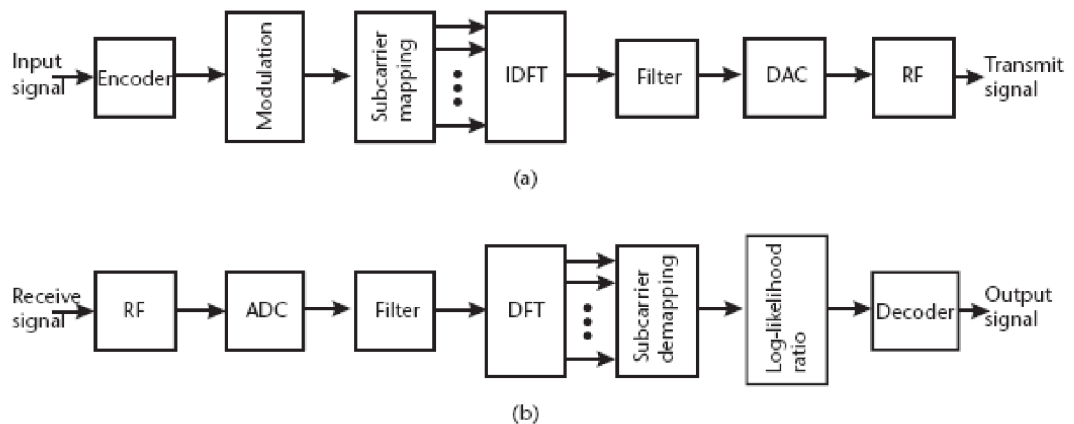
Základní principy SOFDMA jsou, že vzdálenosti subnosných jsou nezávislé na šířce pásma, počet subnosných vzrůstá s šířkou pásma. Počet subkanálů vzrůstá s šířkou pásma, zatímco kapacita každého individuálního subkanálu zůstává konstantní.

Tabulka 3 Parametry SOFDMA

PARAMETRY	HODNOTY			
Šířka pásma systému [MHz]	1,25	5	10	20
Vzorkovací frekvence F_p [MHz]	1,4	5,6	11,2	22,4
Velikost FFT	128	512	1024	2048
Počet subkanálů	2	8	16	32
Frekvenční oddělení subnosných	10,94 kHz			
Užitečná doba symbolu	91,4 μ s			
Ochranný čas	11,4 μ s			
Trvání OFDMA symbolu	102,9 μ s			
Počet OFDMA symbolů (5ms rámeček)	48			

2.5 OFDMA vysílač a přijímač

Na obrázku 2.9 je znázorněno schéma OFDMA vysílače a přijímače



Obrázek 2.9: a) Schéma OFDMA vysílače, b) schéma OFDMA přijímače, převzato z [4]

Podle tohoto schématu bude vytvořen matematický model fyzické vrstvy v prostředí MATLAB, viz. kapitola 4.

3 FYZICKÁ VRSTVA SYSTÉMU WIMAX

3.1 Fyzická vrstva

Fyzická vrstva je nejnižší vrstvou OSI modelu, definuje elektrické a fyzické specifikace pro zařízení. Zejména definuje vztah mezi zařízením a přenosovým médiem. Jejimi hlavními úkoly jsou sestavení a ukončení spojení, řízení při sdílení média mezi více účastníky, modulace nebo konverze určitých signálů.

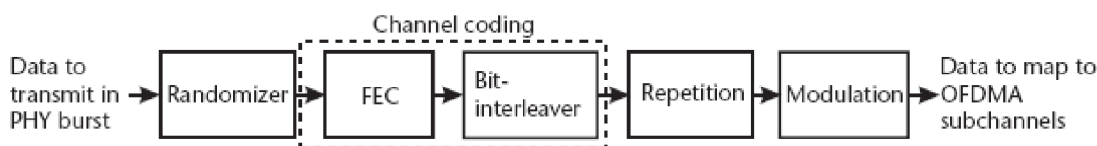
V případě standardu 802.16 - 2004 můžeme hovořit o pěti typech fyzické vrstvy a to následujících:

- WirelessMAN-SC PHY
- WirelessMAN-SCa PHY
- WirelessMAN-OFDM PHY
- WirelessMAN-OFDMA PHY
- Wireless HUMAN

Tabulka 4: Porovnání jednotlivých fyzických vrstev a jejich vlastností, převzato z [1]

Fyzická vrstva	Použitá šířka pásma	Možnosti	Duplex
WirelessMAN-SC	10 - 66 GHz		TDD,FDD
WirelessMAN-SCa	Licencovaná pásma pod 11 GHz	AAS,ARQ,STC	TDD,FDD
WirelessMAN-OFDM	Licencovaná pásma pod 11 GHz	AAS,ARQ,Mesh,STC	TDD,FDD
WirelessMAN-OFDMA	Licencovaná pásma pod 11 GHz	AAS,ARQ,STC	TDD,FDD
WirelessHUMAN	Licencovaná pásma pod 11 GHz	AAS,ARQ,Mesh,STC	TDD

Hlavním úkolem fyzické vrstvy, o kterou se budeme zajímat je zpracování signálů a jejich vyslání do radiového kanálu.

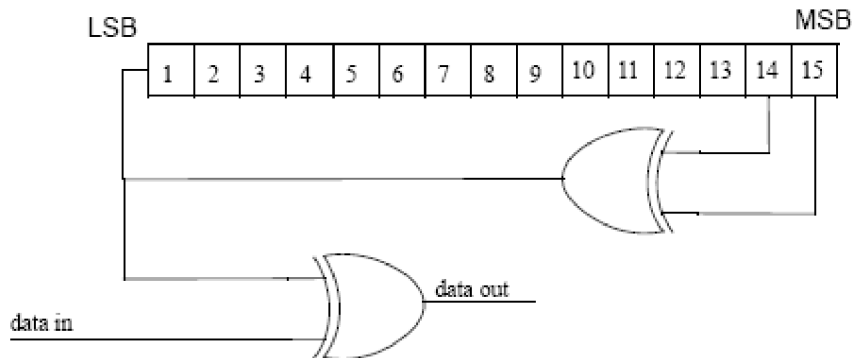


Obrázek 3.1 Proces zpracování dat fyzickou vrstvou, převzato z [3]

3.2 Randomizace

Randomizace je prováděna se všemi daty, jak v uplinku, tak i v downlinku. Randomizace je aplikována na každý FEC blok, tj. v našem případě blok o velikosti patnácti bitů. Pokud počet dat neodpovídá přesně potřebnému počtu, bude na konec datového toku přidáno potřebné množství nul (pouze jednou) tak, aby počty dat odpovídaly alokovanému prostoru pro data. Alokovaný prostor znamená počet

odpovídající $[N_S/R]$, kde N_S je počet přidělených slotů a R je použitý opakovací faktor.



Obrázek 3.2 Schéma PRBS generátoru použitého pro randomizaci, převzato z [1]

Každý datový bajt, který má být přenesen, postupně vstoupí do bloku Randomizéru, (MSB bit jako první). Preambule nejsou randomizovány. Počáteční hodnoty jsou použity pro výpočet randomizačních bitů, které jsou potom pomocí funkce XOR kombinovány s každým vysláním FEC bloku. Atomizační sekvence je aplikována pouze na informační bity.

Randomizer je inicializován s vektorem $v_{init} = [011011100010101]$.

3.3 Kanálové kódování

Jeho úkolem je ochrana dat před chybami vzniklými při přenosu radiovým prostředím. V základu jsou možné dva způsoby ochrany. První způsob pomocí ochranných kódů, druhý pomocí automatického opakování přenosu (ARQ). Je možné tyto metody zkombinovat - tzv. Hybridní ARQ). ARQ je ovšem zajišťováno transportní vrstvou, kterou standard 802.16 neobsahuje (viz kapitola 1.2).

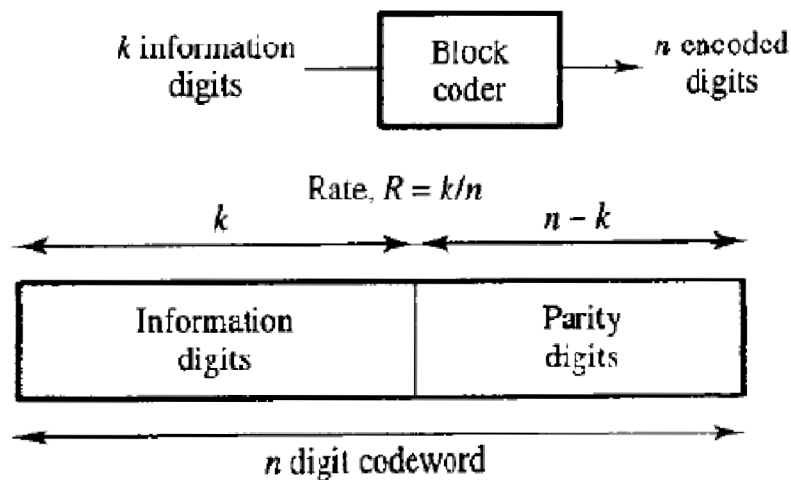
V našem případě se budeme zabývat možností s ochrannými kódy. V případě WiMAXu jsou použity RS kódy, konvoluční kódy, Turbo kódy.

3.3.1 RS kódy

Reed-Solomonovy kódy jsou jedny z nejpoužívanějších blokových kódů. K porozumění blokových kódů je nejprve zapotřebí určit několik definic.

Předpokládáme, že výstup z informačního zdroje je binární sekvence (tj. „1“ nebo „0“). Při blokovém kódování je tento výstup rozdělen na bloky o určité délce. Každý blok, který si označíme jako \mathbf{u} , se skládá z k informačních bitů.

K dispozici je celkem 2^k různých zpráv. Kodér, podle určitých pravidel, převádí každou vstupní zprávu do binární n -tice \mathbf{v} , kde $n > k$. Takle binární n -tice \mathbf{v} je označována jako kódové slovo nebo kódový vektor zprávy \mathbf{u} .



Obrázek 3.3 Princip blokového kódu ,viz [9]

Kvůli tomu, že existuje 2^k možných zpráv, existuje také 2^k možných kódových slov. Tento soubor 2^k kódových slov, se nazývá blokový kód. Aby byl daný blokový kód užitečný, musí být 2^k kódových slov odlišných.

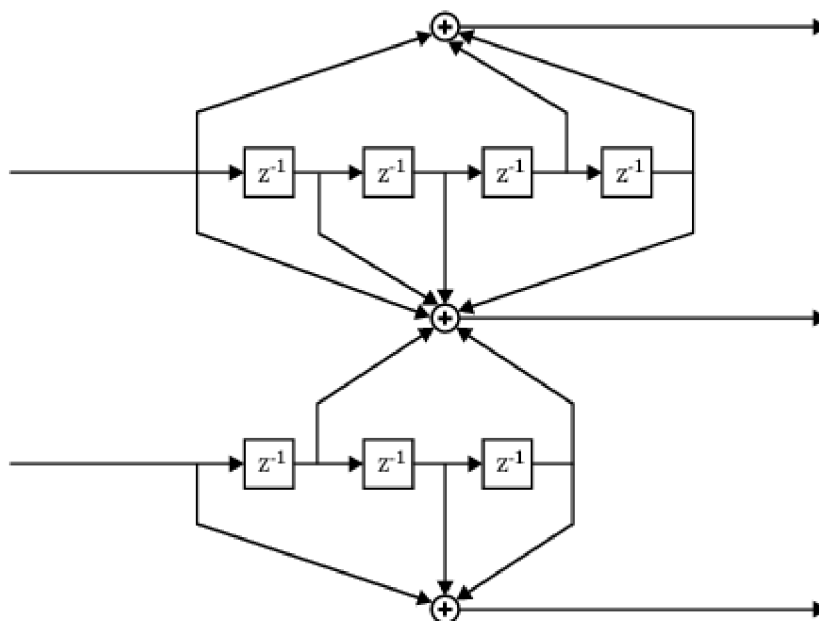
Blokový kód o délce n a počtu kódových slov 2^k je nazýván lineární (n,k) kód. Ve skutečnosti je ovšem blokový kód lineární pouze tehdy je-li modulo 2 součtu kódových slov, také kódové slovo. Příkladem blokových kódů jsou již zmiňované RS kódy, dále například Hammingovy kódy.

3.3.2 Konvoluční kódy

Konvoluční kódy jsou většinou specifikovány třemi parametry (n,k,m) , kde n je počet výstupních bitů, k je počet vstupních bitů a m je počet paměťových registrů.

Poměr k/n je nazýván kódový poměr a je to měřítko účinnosti kódu. Obvykle jsou n a k v rozmezí 1 až 8, m 2 až 10, a kódový poměr od 1/8 do 7/8. Často také výrobci čipů specifikují parametry (n, k, L) , kde L se nazývá omezující délka kódu[7] a je definováno jako:

$$L = k(m - 1) \quad (1)$$



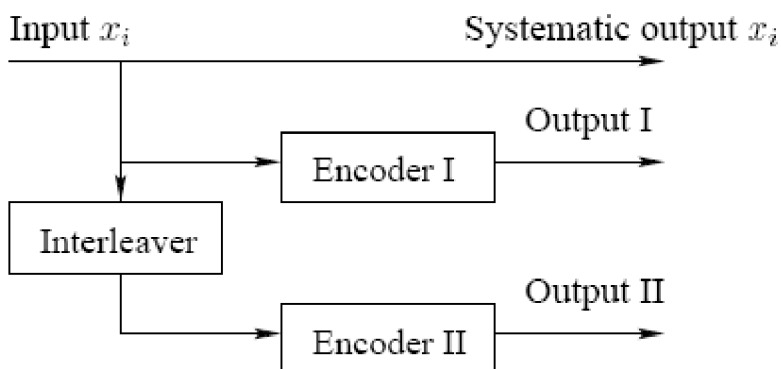
Obrázek 3.4 Schéma konvolučního kodéru s kódovým poměrem 2/3

Výstupní řetězec kodéru je tvořen určitým sloučením jednotlivých výstupních kódů.

3.3.3 Turbo kódy

Turbo kódy byly poprvé představeny v roce 1993. Bylo představeno schéma, které dosahuje pravděpodobnosti výskytu chyby 10^{-5} při použití kódu o poměru $1/2$ a kanálu s aditivním bílým Gaussovským šumem a BPSK modulace o $E_b/N_0 = 0,7$ dB.

Jednoduše řečeno, turbo kód je tvořen paralelním spojením dvou oddělených kódů, přes prokladač (interleaver).



Obrázek 3.5 Principiální schéma turbo kodéru [10]

3.4 Bitové prokládání

Prokládání slouží k přeskládání bitů, díky čemuž lze předcházet skupinovým (shlukovým) chybám, neboli burstům. Skupinová chyba je na přijímací straně rozložena na chyby menší, které by měly být schopny opravit opravné kódy.

Důležitými parametry uvedeného zpracování signálu jsou tzv. hloubka prokládání a rámeček vnějšího kódu. Čím větší je hloubka prokládání, tím větší může být skupinová chyba, kterou je schopen prokládací stupeň „rozprostřít“. Rámeček vnějšího kódu udává počet bitů, po kterých se budou opakovat vzniklé ojedinělé chyby.

3.4.1 Pseudonáhodné prokládání

Jednoduše přeřadíme dané bity v našem datovém toku, a to pomocí předem určené náhodné posloupnosti, tuto posloupnost poté aplikujeme ve vysílači k získání původních dat.

3.4.2 Maticové prokládání

Také nazýváno jako blokové prokládání, data jsou ve vysílači načtena po řádcích do matice a poté přečtena po sloupcích, dále následuje přenos radiovým kanálem a v přijímači se aplikuje opačný postup, tj. vložíme data po sloupcích a čteme je po řádcích

3.4.3 Konvoluční prokládání

Zde se data vkládají do paměťové matice po diagonálách a čtou po sloupcích, v přijímači opět následuje opačný proces.

3.5 Repetition (opakování)

Slouží ke zvýšení datového toku. Je to proces při kterém opakujeme identický „slot“ bitů R -násobně, kde R je „opakovací faktor“. Data získaná po randomizaci a kódování jsou rozdělena do několika slotů. Poté je každý slot opakován R -krát, čímž se vytvoří R za sebou jdoucích slotů určených k datovému mapování.

Například pokud máme $R = 2, 4$ nebo 6 , počet přidělených slotů NS je násobek R pro uplink, a číslo v rozsahu $[\min, \max]$ pro downlink, viz [1]:

$$\min = R \cdot K \quad (2), \quad \max = R \cdot K + (R - 1) \quad (3),$$

kde K je počet potřebných slotů před opakováním. Např. $K = 10, R = 6$ bude počet přidělených slotů číslo mezi 60 a 65. Toto schéma se aplikuje pouze pro QPSK modulaci.

3.6 Digitální modulace

Protože rádiový přenos v základním pásmu nelze realizovat, je nutné k jeho uskutečnění modulační signál, nesoucí informaci a ležící v základním pásmu, namodulovat na nosnou vlnu (vysokofrekvenční nebo mikrovlnou). Techniku modulace však lze využít i při přenosu po vedení, kde je možné formou FDM přenášet více vzájemně nezávislých modulačních signálů, aniž by docházelo k jejich vzájemnému rušení.

Modulační signál může mít různou podobu, odpovídající libovolné nekódované nebo kódované diskrétní modulaci v základním pásmu. Výhodné vlastnosti však mají pouze binární modulační signály PCM a DM a jejich varianty DPCM, ADPCM atd.

Ty je potom možné namodulovat na sinusovou vlnu buď amplitudově, frekvenčně nebo fázově, případně kombinací těchto základních způsobů. V případě systému WiMAX se budeme zabývat modulacemi BPSK, QPSK a modulacemi 16-QAM a 64-QAM.

3.6.1 Modulace BPSK

Jedná se o modulaci typu PSK tj. klíčování s fázovým posuvem, kde datový binární signál ovlivňuje fázi nosné vlny, přičemž její amplituda zůstává konstantní. V případě BPSK může fáze nabývat dvou stavů např. 0° a 180° . Příslušné dva signálové prvky je možné vyjádřit vztahy [2]:

$$s_{BPSK}(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t), \quad \text{pro } 0 \leq t \leq T_b \text{ (binární 1)} \quad (4)$$

$$s_{BPSK}(t) = -\sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t), \quad \text{pro } 0 \leq t \leq T_b \text{ (binární 0)} \quad (5)$$

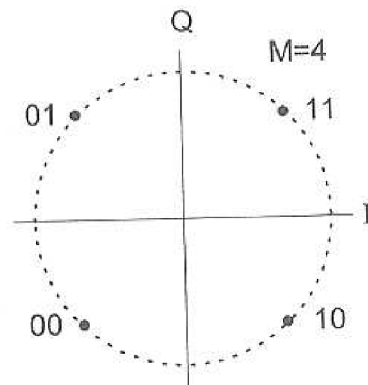
kde $\sqrt{2E_b/T_b}$ je amplituda modulovaného signálu, přičemž E_b je střední energie modulovaného signálu na 1 bit a T_b je bitová perioda. Modulační signál nemusí mít pravoúhlý průběh, nýbrž může být tvarován na vhodný průběh $m(t)$, potom lze signál BPSK psát v obecném tvaru [2]:

$$s_{BPSK}(t) = m(t) \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t), \quad \text{pro } 0 \leq t \leq T_b \text{ (binární 1 nebo 0)}, \quad (6)$$

přičemž v případě bipolárních modulačních impulzů NRZ je $m(t) = a_n = \pm 1$

3.6.2 QPSK

Quadrature Phase Shift Keying náleží mezi nejdůležitější kategorie vícecestavových PSK modulací, neboť se vyznačuje dobrou spektrální a výkonovou účinností. Nosná vlna má konstantní amplitudu a může nabývat čtyři různé fázové stavy, např. 45°, 135°, 225°, 315°. Existují zde tedy čtyři různé signálové prvky, přičemž každému z nich odpovídá jedna bitová dvojice. Konstelační diagram má různou konfiguraci, výhodná je varianta, kde je uplatněn princip Grayova kódování.



Obrázek 3.6 Konstelační diagram modulace QPSK, převzato z [2]

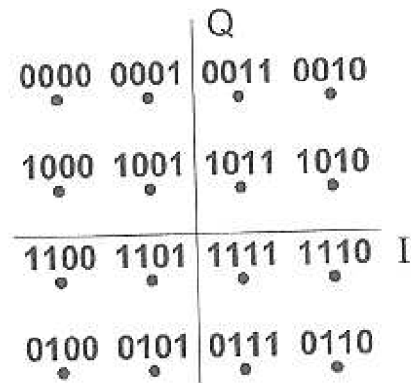
Zde dochází při chybném vyhodnocení symbolu v přijímači, záměně za sousední symbol, pouze k chybě v jediném datovém bitu. Signál QPSK je určen při pravoúhlém modulačním signálu v časové oblasti vztahem [2]:

$$s_{QPSK}(t) = -\sqrt{\frac{2E_s}{T_s}} \cos \left[2\pi f_c t + (2i - 1) \frac{\pi}{4} \right], \quad \text{pro } 0 \leq t \leq T_s, i = 1, 2, 3, 4, \quad (7)$$

kde E_s je střední energie signálu připadající na jeden symbol a $T_s = 2T_b$ je symbolová perioda, rovná dvojnásobku bitové periody T_b .

3.6.3 M-QAM

U kvadraturních amplitudových modulací se modulačním signálem ovlivňuje jak amplituda, tak fáze nosné vlny. Tyto formáty představují jednu z nejstarších, avšak stále důležitou variantu modulací proměnnou obálkou. Používají se ve variantách s více stavy, tedy v aplikacích, kde se požaduje velká spektrální účinnost. Konstelační diagram funkce 16-QAM viz obrázek.



Obrázek 3.7 Konstelační diagram funkce 16-QAM, převzato z [2]

U M-stavových modulací QAM se sdružují dva nebo více modulačních bitů do n-bitových kódových skupin, tj. symbolů, přičemž $M = 2^n$. Tyto symboly jsou vysílány ve formě signálových prvků $s_1(t)$, $s_2(t), \dots, s_n(t)$, které mají dobu trvání tj. symbolovou periodu T_s a jsou vyjádřeny obecným vztahem [2]:

$$s_{MQAM} = \sqrt{\frac{2E_{min}}{T_s}} a_i \cos(2\pi f_c t) + \sqrt{\frac{2E_{min}}{T_s}} b_i \sin(2\pi f_c t), 0 \leq t \leq T_s, i = 1, 2, M, (8)$$

kde a_i a b_i jsou páry nezávislých celých čísel, odpovídající poloze jednotlivých bodů konstelačního diagramu a E_{min} je střední energie signálového prvku s nejmenší amplitudou.

Signál M-QAM se skládá ze dvou kvadraturních nosných vln (tj. vln se vzájemnou fází 90°), z nichž každá je amplitudově klíčována soustavou diskretních modulačních signálů a sama představuje signál s klíčováním amplitudy ASK.

Například u modulace 16-QAM je synfázní vlna I klíčována čtyřmi modulačními úrovněmi a kvadraturní vlna rovněž čtyřmi úrovněmi. Signálové prvky zde nemají stejnou energii a v důsledku toho jsou jednotlivé symboly v přijímači detekovány s různou pravděpodobností výskytu chyby. Za předpokladu, že modulační impulzy mají pravoúhlý průběh, je možné signál $s_{MQAM}(t)$ vyjádřit také pomocí básových funkcí, definovaných [2]:

$$\theta_1(t) = \sqrt{\frac{2}{T_s}} \cos(2\pi f_c t) ; \theta_2(t) = \sqrt{\frac{2}{T_s}} \sin(2\pi f_c t), \text{ pro } 0 \leq t \leq T_s. (9)$$

Souřadnice i-tého bodu (symbolu) jsou $a_i \sqrt{E_{min}}$ a $b_i \sqrt{E_{min}}$, kde (a_i, b_i) je prvek matice $[L \times L]$ určené vztahem [2]:

$$\{a_i, b_i\} = \begin{vmatrix} (-L+1, L-1) & (-L+3, L-1) \dots & (L-1, L-1) \\ (-L+1, L-3) & (-L+3, L-3) \dots & (L-1, L-3) \\ (-L+1, -L+1) & (-L+3, -L+1) \dots & (L-1, -L+1) \end{vmatrix}, \quad (10)$$

přičemž $L = \sqrt{M}$ Výkonové spektrum modulací M-QAM je shodné s modulacemi M-PSK, shodná je i spektrální účinnost a potřebná šířka pásma. Výkonová účinnost je modulací M-QAM je však lepší, než u M-PSK. Pravděpodobnost chyby v přenosu symbolu je u modulací M-QAM, při přenosu kanálem se šumem AWGN a při koherentní demodulaci, určena přibližným vztahem [2]:

$$P_{EMQAM} \cong 2 \left(1 - \frac{1}{\sqrt{M}}\right) \operatorname{erfc} \left(\sqrt{\frac{E_{min}}{N_0}} \right). \quad (11)$$

Jak je patrné, se zvětšujícím se počtem stavů u modulací M-QAM se spektrální účinnost zvětšuje, avšak výkonová účinnost (při zachování stejné chybovosti) se zmenšuje.

3.6.4 DFT a FFT

Jsou zkratky diskrétní Fourierovy transformace a rychlé Fourierovy transformace. Jak již bylo řečeno, k vytvoření spektra se v OFDM systémech používá matematické operace IDFT, ovšem v případě použití n-bodové transformace lze použít IFFT, pokud

$$n = 2^x, \quad x = 1, 2, 3, \dots$$

V přijímači poté následuje opačný proces, kdy pomocí FFT získáváme modulovaný signál.

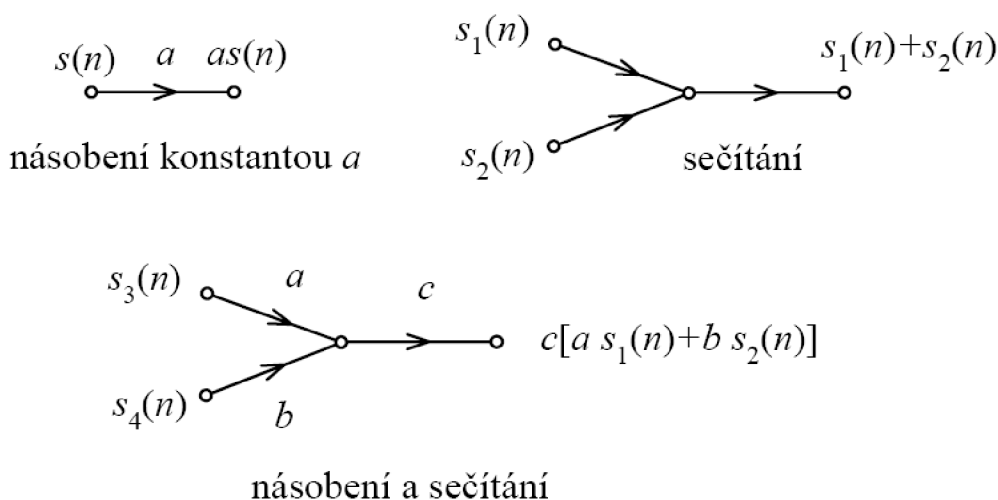
Diskrétní Fourierova transformace přiřazuje posloupnosti délky N jinou posloupnost délky N. Originál je $\{s(n)\}$, obraz je $\{S(k)\}$. Vztah pro výpočet obrazu DFT je následující

$$S(k) = R_N(k) \sum_{n=0}^{N-1} s[\operatorname{mod}_N(n)] \exp\left(-j \frac{2\pi}{N} kn\right) = R_N(k) \sum_{n=0}^{N-1} s(n) \exp\left(-j \frac{2\pi}{N} kn\right). \quad (12)$$

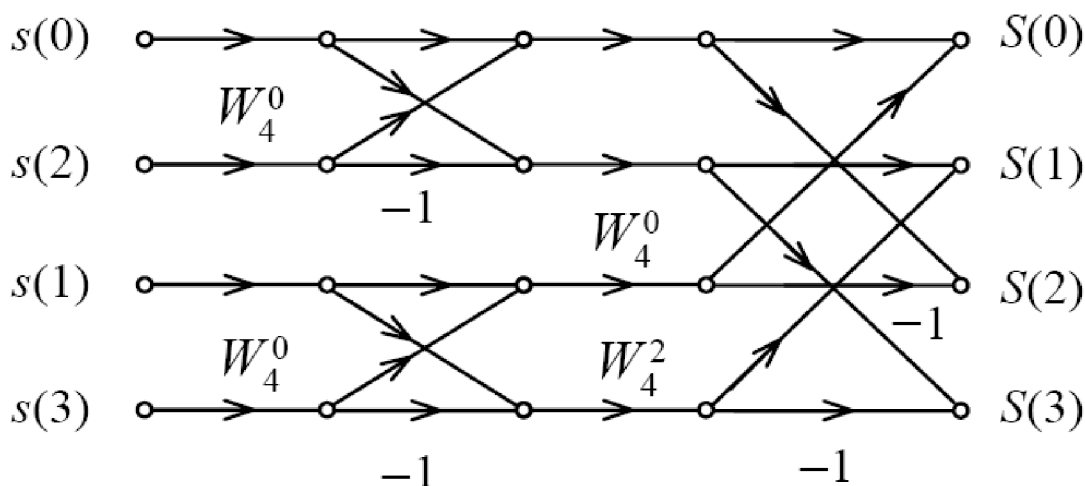
Z matematického hlediska je výraz $R_N(k)$ nadbytečný, proto se v literatuře často vynechává. Zpětnou Fourierovu transformaci (IDFT) lze zapsat následujícím vztahem:

$$s(n) = R_N(n) \frac{1}{N} \sum_{k=0}^{N-1} S(k) \exp\left(-j \frac{2\pi}{N} kn\right). \quad (13)$$

Rychlá Fourierova transformace je algoritmus pro výpočet DFT (vztahu 12), který využívá bitově reverzního pořadí dat. Na následujících obrázcích jsou základní operace násobení a sečítání pro grafy signálových toků, které použijeme pro grafické zobrazení výpočtu FFT.



Obrázek 3.8 Operace vyjádřené pomocí grafů signálových toků



Obrázek 3.9 Algoritmus FFT typu DIT pro $N = 4$

Algoritmus na obrázku patří do skupiny DIT. Vyplývá to z toho, že posloupnost vstupních dat dělíme na dvě posloupnosti. První z nich má členy se sudými indexy, druhá obsahuje členy s lichými indexy. Tyto dílčí posloupnosti dělíme dále stejným způsobem, až nám zůstanou dvoubodové základní posloupnosti. To platí v případě, že výchozí počet členů posloupnosti N je mocninou čísla dva, například 1024 je rovno 2^{10} .

Z tohoto důvodu jsou u OFDM či OFDMA systémů používány následující velikosti FFT – 128, 512, 1024, 2048. Velikost FFT také určuje počet subnosných a nastavení dalších parametrů systému.

3.6.4 Raised Cosine Filtr

V systému WiMAX je využit jako vstupní a výstupní filtr. Hlavním úkolem je kompenzace ISI. Nyquistova podmínka pro přenos bez ISI v základním pásmu [12].

$$\sum_{n=-\infty}^{\infty} H(f - f_s) = T_s. \quad (14)$$

Nejjednodušší způsob, jak tuto podmínku splnit je uvažovat v základním pásmu od $-B$ do B (zde $B = 0,5 f_s$) pouze jediný nenulový prvek řady z levé strany rovnice 13 [12].

$$H(f) = \begin{cases} T_s & |f| \leq B \\ 0 & |f| > B \end{cases} \quad (15)$$

Požadovanou impulsní odezvu filtru $h(t)$ získáme zpětnou Fourierovou transformací funkce $H(f)$ [12].

$$h(t) = \frac{\sin(2\pi Bt)}{2\pi Bt} = \text{sinc}(2\pi Bt) \quad (16)$$

Filtrem zaručujícím nulové ISI je tedy filtr s impulsní charakteristikou ve tvaru sinc, jeho nevýhodou je ovšem obdélníková forma amplitudové frekvenční charakteristiky. Možným řešením je mírné rozšíření frekvenčního přenosu použitého filtru. Za tímto účelem budeme uvažovat v přenosovém pásmu od $-B$ do B (kde $B = 0,5 f_s$) celkem tři nenulové prvky z levé strany rovnice 13. [12]

$$H(f) + H(f - 2f_s) + H(f + 2f_s) = T_s \quad (17)$$

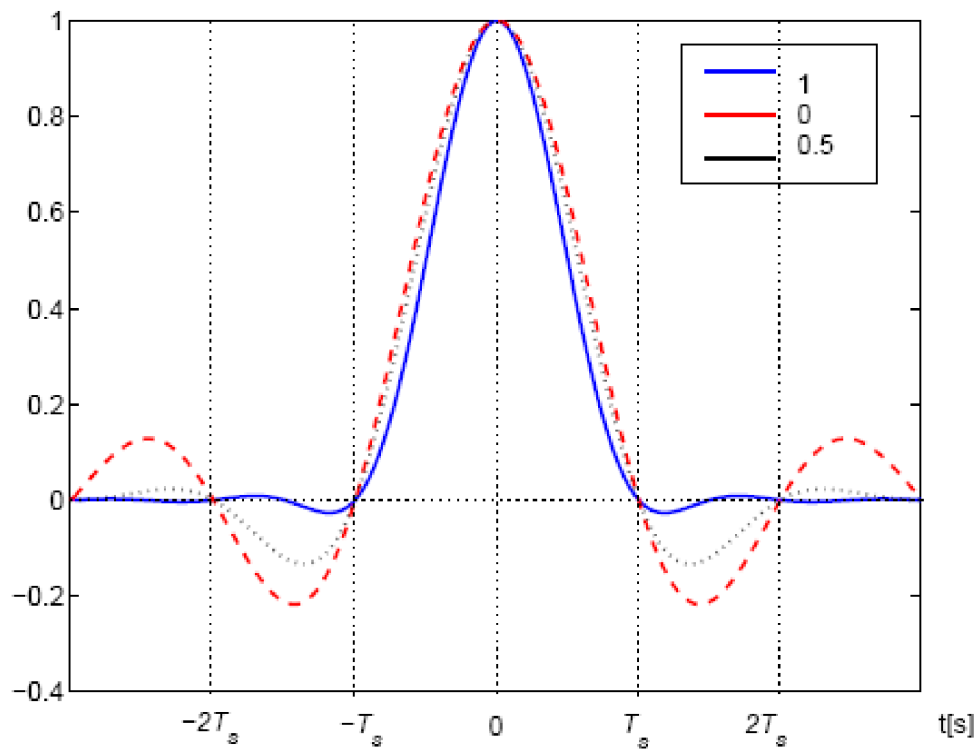
Jedním z filtrů splňujících tuto podmínku je právě Raised Cosine filtr s frekvenční přenosovou charakteristikou (19) [12].

$$H(f) = H_{rc}(f) \begin{cases} T_s & , 0 \leq |f| < \frac{1-\beta}{2T_s} \\ \frac{T_s}{2} \left[1 - \sin \frac{\pi T_s}{\beta} \left\{ |f| - \frac{1}{2T_s} \right\} \right] & , \frac{1-\beta}{2T_s} \leq |f| < \frac{1+\beta}{2T_s} \\ 0 & , \frac{1+\beta}{2T_s} \leq |f| \end{cases} \quad (18)$$

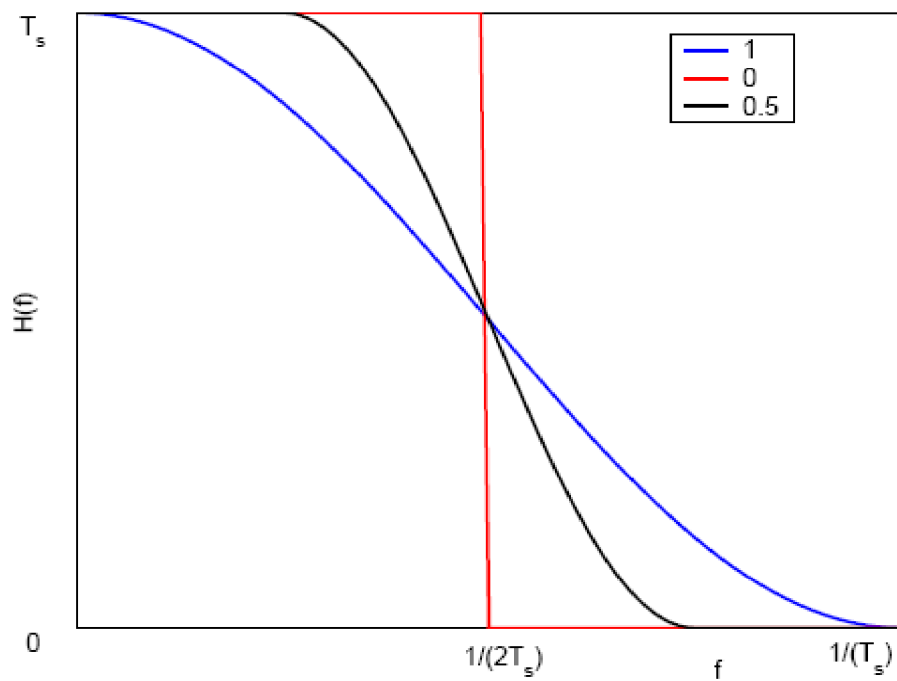
Činitel tvaru β určuje rozšíření přenosové funkce filtru vzhledem k filtru s impulsní charakteristikou z rovnice (15). Jeho význam je patrný z následujících obrázků, kde je zobrazena jak přenosová tak impulsní charakteristika tohoto filtru. Pro $\beta = 0$, odpovídá Raised cosine filtru z rovnice 15. S rostoucím β pak roste šířka pásma a dochází k rychlejšímu útlumu impulsní charakteristiky. Impulsní charakteristika filtru Raised Cosine je definována následujícím vztahem [12].

$$h(t) = h_{rc}(t) = \text{sinc}\left(\frac{\pi t}{T_s}\right) \frac{\cos\left(\frac{\pi \beta t}{T_s}\right)}{1 - \frac{\beta^2 t^2}{T^2}} \quad (19)$$

Celkový přenos filtru $H(f)$ je třeba rozdělit mezi vysílací filtr $H_T(f)$ a přijímací filtr $H_R(f)$.



Obrázek 3.10 Přenosová charakteristika filtru Raised Cosine



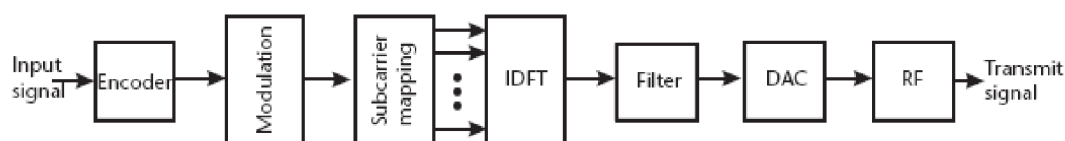
Obrázek 3.11 Impulsní charakteristika filtru Raised Cosine

4 MODEL FYZICKÉ VRSTVY

Daný model je realizován v prostředí MATLAB, použité funkce viz [10]. Popřípadě v příloze „Použité funkce – Matlab“.

4.1 Model Vysílače

Na obrázku 4.1 je znázorněno schéma daného OFDMA vysílače. V této kapitole se dozvíme o tvorbě jeho modelu.

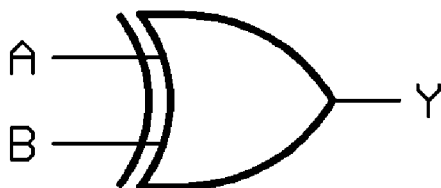


Obrázek 4.1 Schéma OFDMA vysílače

Blok Encoder zahrnuje čtyři části zpracování signálu a to randomizaci, kódování, prokládání a opakování. Blok Modulation provádí digitální modulaci. Blok Subcarrier mapping se zabývá sériově-paralelním převodem a mapováním subnosných. IDFT blok vytváří dané OFDMA spektrum signálu pomocí IFFT operace a realizuje paralelně-sériový převod. Blok Filter modeluje výstupní filtr, blok DAC digitálně-analogový převodník. Poslední blok RF vysílá signál do radiového prostředí.

4.1.1 Randomizace

Proces randomizace jsme realizovali podle obrázku 3.2. Bylo využito dvou funkcí XOR (viz obrázek 4.2). Pro doplnění je uvedena i stavová tabulka.



Tabulka 5 Pravdivostní tabulka funkce xor

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Obrázek 4.2 Schématická značka hradla xor

Samostatná funkce se realizuje v prostředí Matlab stejným příkazem, tedy xor. Zde uvádíme ukázkou zápisu této funkce:

$$Y = \mathbf{xor} (A, B);$$

4.1.2 Konvoluční kódování

U konvolučního kodéru (dále jen CC) je hlavním parametrem tzv. „rate“, který je definován následujícím vztahem:

$$rate = \frac{\text{počet vstupních bitů do kodéru}}{\text{počet výstupních bitů z kodéru}} \quad (20)$$

U systému WiMAX je, jak již bylo zmíněno, použito pět různých konfigurací rate pro konvoluční kódy, a to 1/2, 1/3, 1/6, 2/3 a 5/6.

K vytvoření CC je v našem modelu použito dvou funkcí. Jedné pro vytvoření struktury kodéru, funkce `poly2trellis`, kterou zároveň určujeme i rate kodéru. A druhé pro samotné kódování s použitím dané struktury, funkce `convenc`.

Pro jasnější představu uvádíme zápis této funkce, který odpovídá obrázku 3.4.

```
Struktura = poly2trellis([5 4],[23 35 0;0 5 13]);
Kódovaný_signál = convenc(msg,Struktura);
```

Zde `msg` považujeme za vstupní datový tok a funkci `convenc` použijeme k jeho zakódování, podle určené struktury kodéru. Funkce `poly2trellis` je na vysvětlení složitější. První závorka udává počet cest z jednotlivých vstupů do kodéru směrem k výstupu. Podle daného schématu (3.4) pro první cestu 5, pro druhou 4. Jelikož jsou vstupy jen dva, máme v první závorce jen dvě čísla. Druhá závorka určuje generátor kódu. K jejímu vypsání je zapotřebí znalosti první tabulky a číselných soustav, jelikož využívá osmičkové a binární soustavy. Počet čísel v dané závorce závisí na rate kodéru. Pokud označíme čísla v druhé závorce, která jsou oddělena středníkem, jako skupinu, pak lze říci, že počet skupin je roven počtu vstupních bitů kodéru, a počet čísel ve skupině odpovídá počtu výstupních bitů z kodéru. Hodnoty čísel ve skupině nám určuje první závorka, pokud máme například pět a čtyři cesty, rozdělíme čísla 5 a 4 na díly po třech následujícím způsobem:

$$\begin{array}{rcc}
 5 = & 5 & 4 ; & 3 & 2 & 1 ; \\
 & \underbrace{\quad} & & \underbrace{\quad} & & \\
 & 3 & & 7 & & = 37
 \end{array}
 \quad \Bigg| \quad
 \begin{array}{rcc}
 4 = & 4 & ; & 3 & 2 & 1 \\
 & \underbrace{\quad} & & \underbrace{\quad} & & \\
 & 1 & & 7 & & = 17
 \end{array}$$

Každá trojice reprezentuje binární čísla 0 a 1, tím pádem nejvyšší hodnota trojice je 7 v dekadické podobě, nejvyšší hodnota dvojice je 3 v dekadické podobě a nejvyšší hodnota jednoho čísla je 1. Tím dostáváme maximální číslo v první skupině 37, jelikož počet výstupních bitů kodéru je 3, musí být ve skupině ještě další dvě čísla, spadající do číselného rozmezí, které je určeno první závorkou.

4.1.3 RS kódování

Obdobně jako u konvolučního prokládání i zde využijeme dvě funkce. Jednu k vytvoření struktury kodéru, funkce `fec.rsenc(N,K)`, a druhou pro samotné kódování, funkce `encode`.

U funkce `fec.rsenc(N,K)` je K délka nekódované zprávy, N je délka základního kódu (tj. kódované zprávy). Jelikož jsou však RS kódy nebinární a náš model pracuje povětšinou jen s binárními kombinacemi, je třeba převést kódovaná data pomocí funkce `de2bi` z dekadické soustavy do soustavy binární. Následuje ukázka zápisu:

```
N = 255; K = 239 ;
Kodér = fec.rsenc(N,K) ;
Kódovaný_signál = encode(msg,Kodér) ;
```

Předtím než použijeme funkci `encode`, je zapotřebí se ujistit, že vstupní signál `msg` je celočíselným násobkem K a jde o sloupcový vektor o $\log_2(N+1)$ sloupcích, nikoliv řádkový, jinak dojde k chybě a vstupní data nepůjdou zakódovat.

4.1.4 Prokládání

V našem modelu využíváme tři typy prokládání - pseudonáhodné, maticové a konvoluční. Pro každý typ je použita jiná funkce, která jej realizuje.

Pseudonáhodné prokládání je realizováno pomocí funkce `intrlv`, která přeřadí vstupní datový tok podle předurčeného vektoru, který je vytvořen pomocí funkce `randperm`. Následuje ukázka zápisu:

```
Vektor_přeřazení = randperm(length(msg) ;
Proložený_signál = intrlv(msg,randperm) ;
```

Maticové prokládání realizujeme funkcí `matintrlv`. Při maticovém prokládání je potřeba určit nastavení prokládací matice, tj. počet řádků a sloupců. Dále je potřeba zajistit, aby počet sloupců krát počet řádků prokládací matice byl stejný, jako je délka vstupního datového toku. Ukázka zápisu pro vstupní datový tok `msg` o délce sto bitů:

```
Počet_řádků =10;
Počet_sloupců =10;
Proložený_signál= matintrlv(msg,Počet_řádků,Počet_sloupců) ;
```

Pro konvoluční prokládání je použita funkce `convintrlv`. Zápis je složitější, než u předchozích dvou typů:

```
Počet_řádků = 3;
Zpoždění = 3;
D = Počet_řádků · (Počet_řádků - 1) · Zpoždění;
msg = msg';
msg_upravená = [msg,zeros(D,1)];
Proloženo = convintrlv(msg_upravená,Počet_řádků,Zpoždění)' ;
```

Lze si všimnout, že je zapotřebí signál převést z řádkového na sloupcový vektor. Signál je také zapotřebí upravit tím, že k němu přidáme určitý počet nul. Tyto nuly jsou při opačném prokládání stejného typu opět odebrány.

4.1.5 Modulace

Realizujeme celkem čtyři modulace používané v systému WiMAX. K realizaci využijeme funkci modem. V podstatě se jedná o balík funkcí obsahující modulátor a demodulátor pro následující modulace: „DPSK, MSK, OQPSK, PSK, PAM, QAM, General QAM“.

V našem případě realizujeme modulace BPSK, QPSK, 16-QAM a 64-QAM. Tím pádem použijeme balíky pro modulace PSK a QAM. Jelikož se jedná o modulace, jsou použity funkce s koncovkou mod.

Modem.pskmod(vlastnost 1, hodnota 1,...) realizuje PSK modulátor s nastavitelnými vlastnostmi, obdobně pro QAM modulátor použijeme nadefinovanou funkci modem.qammod (vlastnost 1, hodnota 1, ...). Přehled vlastností jednotlivých modulátorů (viz tab. 4.4 a 4.5 – nebo jen jedna a typ podle modulace).

Tabulka 6 Vlastnosti Modulátoru

Vlastnost	Popis vlastnosti
Typ	Typ modulace, hodnota je fixně nastavena.
M	Počet stavů, jde o M-násobnou hodnotu. Pokud není nastavena, přidělí se jí hodnota M = 2.
Fázový offset	Fázový offset ideální signálové konstelace v radiánech, pokud není nastaveno, je přidělena hodnota 0.
Konstelace	Ideální konstelace signálu. Tuto možnost nelze zadávat, je automaticky vypočítána na základě M a fázového offsetu.
Pořadí symbolů	Typ mapování, který je použit k ideálnímu mapování symbolů. Možnosti jsou bingy (binární mapování), gray (Grayovo Mapování) a uživatelem nadefinované.
Mapování symbolů	Tuto položku je možno nadefinovat pokud je pořadí symbolů definované uživatelem. Z tohoto důvodu ji není třeba více rozepisovat.
Vstupní typ dat	Typ dat, která vstupují do modulátoru. Možnosti jsou bit (binární vstup) a integer (celočíslný vstup)

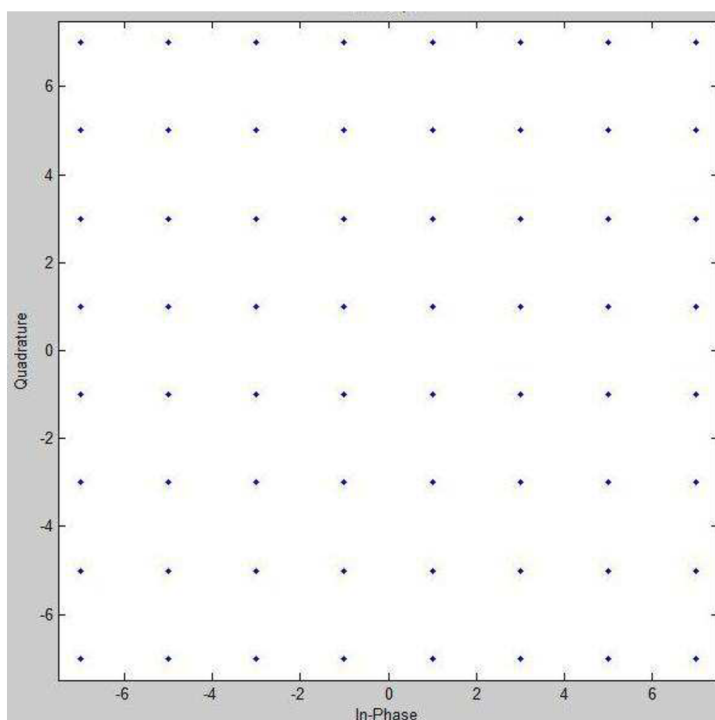
Před modulací je potřeba binární signál převést na dekadický tak, aby měl hodnoty 0 až M-1, tím se zaručí, že se obsadí všechny pozice v konstelačním diagramu.

Následně po modulaci dat následuje ještě proces normování, který zajišťuje, aby se data pohybovala v I-Q rovině na maximálních hodnotách 1 a -1. Normování je provedeno tak, že výstupní data z modulátoru jsou vynásobena určitým koeficientem, který je dán použitou modulací.(viz tabulka 7)

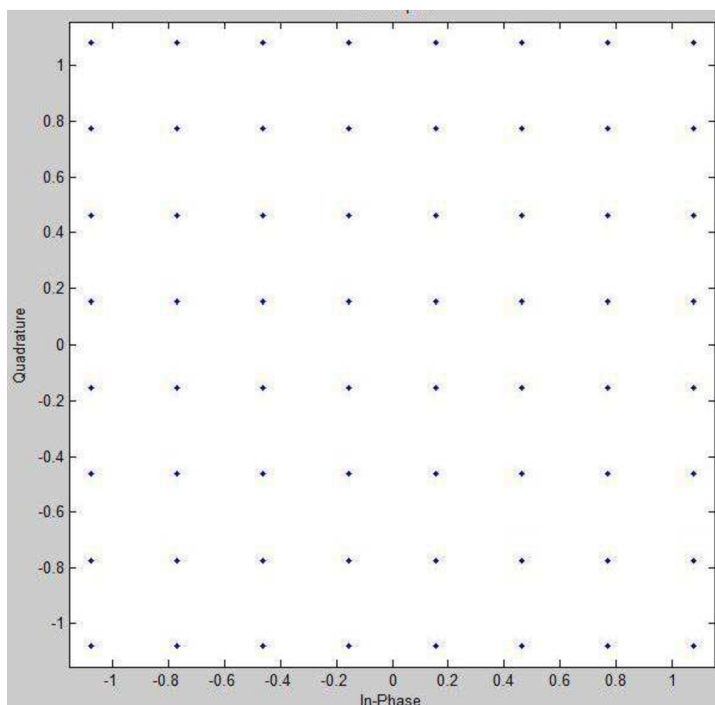
Tabulka 7 Hodnoty koeficientů normování pro použité modulace

Modulace	Koeficient	Modulace	Koeficient
BPSK	1	QPSK	1
16-QAM	$\frac{1}{\sqrt{10}}$	64-QAM	$\frac{1}{\sqrt{42}}$

Z dané tabulky je patrné, že pro kruhové modulace pohybující se na jednotkové kružnici je hodnota koeficientu pro normování 1, jelikož není potřeba upravovat jejich pozici. Ukázka dat před a po normování modulace 64-QAM.



Obrázek 4.3 Konstelační diagram modulace 64-QAM před normováním



Obrázek 4.4 Konstelační diagram modulace 64-QAM po normování

Ukázka zápisu kódu pro modulaci 64-QAM:

```
Modulátor = modem.qammod('M',64,'PhaseOffset',0,'SymbolOrder','binary','InputType','integer');  
Norm = 1/sqrt(42);  
Modulovaný_signál = modulate(Modulátor,msg);  
Normovaný_signál = Modulovaný_signál * norm;
```

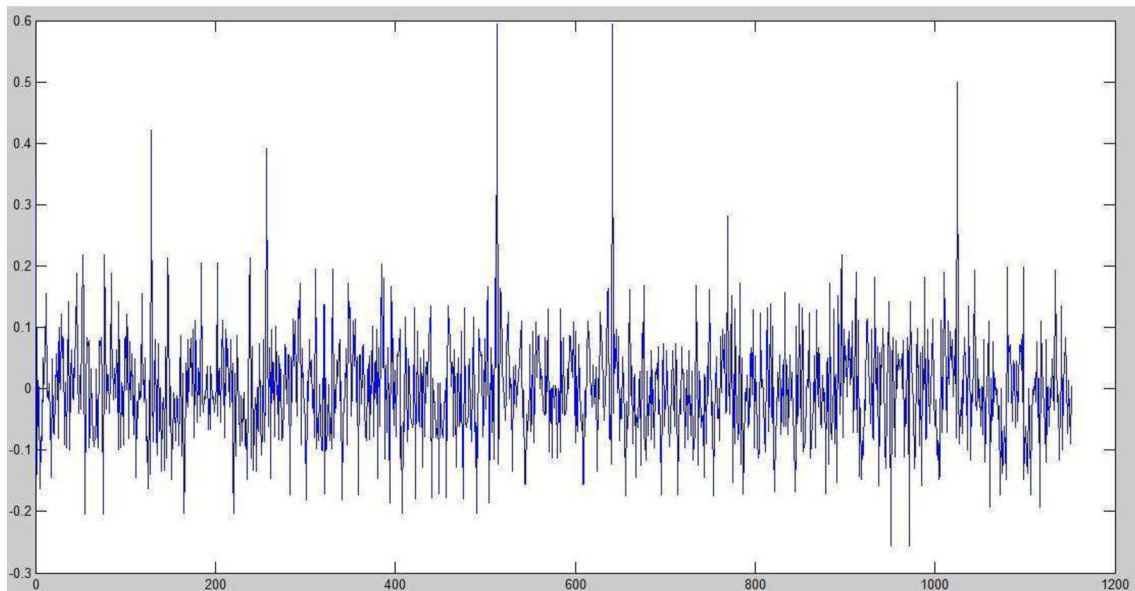
Zde předpokládáme, že signál msg obsahuje hodnoty 0 až 63, aby byly obsazeny všechny stavy v konstelačním diagramu. Pokud máme na vstupu binární signál, je potřeba jej před modulací převést na signál dekadický. Ukázka převodu signálu z binární do dekadické podoby.

```
Signal_bin = reshape(msg,[],log2(M));  
Signal_dec = bi2de(Signal_bin);  
Signál_pro_modulaci = reshape(Signal_dec,1,[]);
```

4.1.6 IFFT

Jak již bylo řečeno, inverzní Fourierovu transformaci použijeme k vytvoření spektra signálu, který budeme přenášet. Naprogramování této části v matlabu plně odpovídá schématu OFDM(A) modulátoru. Využijeme funkci `reshape(zpráva,řádky,sloupce)` k realizaci SPC a PSC, a funkci `ifft` právě k realizaci již zmíněné transformace. Následuje ukázka zápisu a náhled na vytvořené spektrum (obrázek 4.7).

```
N_fft = 128;  
y = reshape(msg,N_fft,[]);  
x = ifft(y,N_fft,...);  
spektrum = reshape(x,1,[]);
```



Obrázek 4.5 Spektrum na výstupu vysílače vytvořené pomocí IFFT

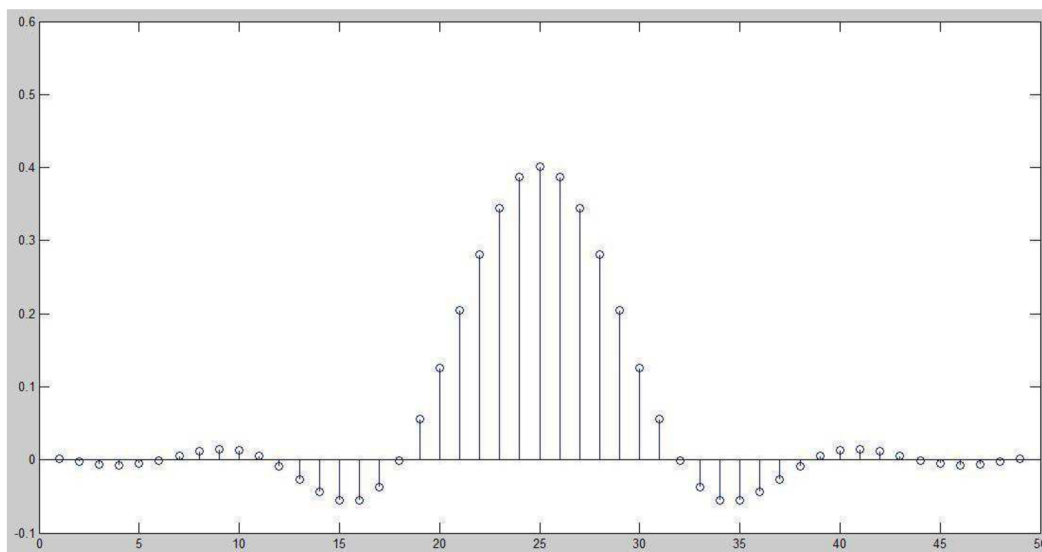
4.1.7 Výstupní filtr

Jako výstupní filtr je používán Raised-Cosine filtr. V našem modelu nakonec nebyl použit, jelikož z nějakého důvodu do něj zanašel dosti velkou chybovost. Pro ilustraci však uvádíme jak jej realizovat.

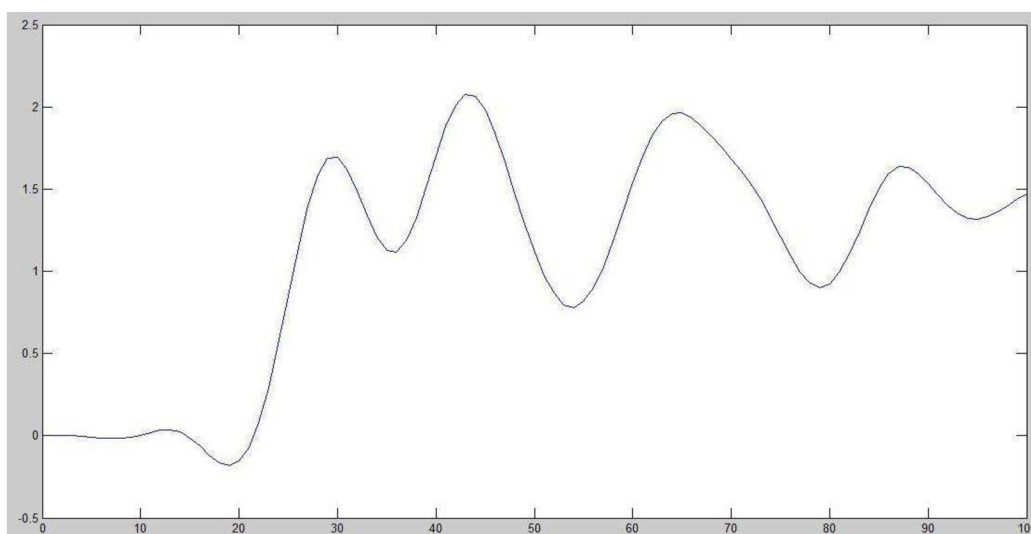
Pro jeho realizaci existuje několik možných funkcí, v našem případě jsme si vybrali funkci `rcosine`.

```
filtr = rcosine(Fd,Fs,typ_filtru);  
filtrovany_signál = filter(filtr,1,msg);
```

F_d je vzorkovací frekvence vstupního signálu, F_s je vzorkovací frekvence výstupního signálu a `typ_filtru` nám určuje, zda se jedná o `fir`, či `iir` filtr a zda `raised cosine` či `square root raised cosine`. Pro ukázkou průchod binárního signálu `sqrt` filtrem.



Obrázek 4.6 Přenosová charakteristika vytvořeného filtru

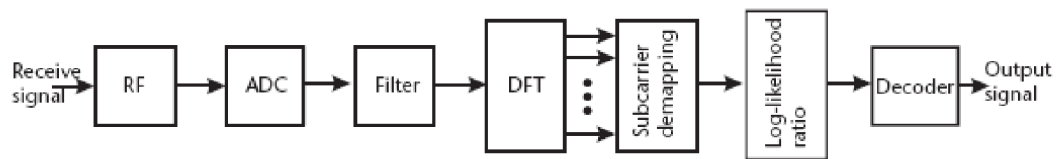


Obrázek 4.7 Binární signál délky 100 bitů po průchodu daným filtrem

Při vytvoření filtru musí platit podmínka, že F_s je minimálně dvakrát větší, než F_d . V podstatě jedná o splnění vzorkovacího teorému.

4.2 Model přijímače

V modelu přijímače dochází k opačným krokům zpracování signálu, než je tomu u vysílače. To znamená, že RF blok představuje příjem signálu, ADC představuje analogově-digitální převod, Filter představuje vstupní filtr, DFT realizuje FFT operaci. V následujícím bloku dochází k demapování subnosných. Nakonec následuje blok dekodování signálu sestávající z inverze „opakování“ a prokládání, dekodování konvolučním a RS kódem a nakonec de-randomizace.



Obrázek 4.8 Schéma OFDMA přijímače

4.2.1 Vstupní Filtr

Jako vstupní filtr je používán taktéž Raised-Cosine filtr. Jak již bylo řečeno, nakonec nebyl použit, zde uvádíme zápis, jak jej realizovat. Nastavení je zcela shodné jako ve výstupním filtru, pouze při filtraci se liší.

```

Filtr = rcosine(Fd, Fs, typ_filtru);
Filtrovaný_signál = filter(1, filtr, msg);
  
```

Ze signálu na obrázku 4.9 bychom po průchodu stejným filtrem dostali binární signál.

4.2.2 FFT

Postup je téměř stejný jako u IFFT ve vysílači, s jediným rozdílem, že použijeme FFT, abychom z vytvořeného spektra získali modulovaný signál.

```

N_fft = 128;
y = reshape(msg, N_fft, []);
x = fft(y, N_fft, ...);
signál_pro_demodulování = reshape(x, 1, []);
  
```

4.2.3 Demodulace

Předtím než data vstoupí do demodulátoru, je potřeba provést opačný proces k normování. Vstupní data tedy vydělíme normovacím koeficientem podle použité modulace, teprve potom následuje samotný proces demodulování. Z procesu demodulace dostáváme dekadický signál, který převedeme na binární, a to pomocí funkce `de2bi`, abychom ho mohli nadále používat v našem modelu.

Ukázka zápisu, pro demodulaci s použitím 64-QAM.

```
Modulátor = modem.qammod('M',64,'PhaseOffset',0,'SymbolOrder','binary','InputType','integer');  
Norm = 1/sqrt(42);  
Odnormovaný_signál = msg./ norm;  
Demodulováno = demodulate(Modulátor,Odnormovaný_signál);
```

Následně je ještě potřeba převést demodulovaná data z dekadické podoby do podoby binární funkcí `de2bi`.

4.2.4 Opačné prokládání

Opačné prokládání je realizováno taktéž třemi funkcemi, pro každé prokládání jedna. Pseudonáhodné opačné prokládání je nejjednodušší, použijeme stejný vektor, podle kterého jsme data přeřadili.

```
De_proložená_data = deintrlv(msg,vektor_přeřazení);
```

Maticové opačné prokládání je realizováno taktéž jako maticové prokládání, viz zápis.

```
Počet_řádků = 10;  
Počet_sloupců = 10;  
De_proloženo = matdeintrlv(msg,Počet_řádků,Počet_sloupců);
```

Konvoluční opačné prokládání je lehčí, než prokládání, ovšem některé části jsou stejné.

```
Počet_řádků = 3;  
Zpoždění = 3;  
D = Počet_řádků * (Počet_řádků - 1) * Zpoždění;  
X = convdeintrlv(msg,Počet_řádků,Zpoždění);  
De_proložená_data = X(D+1:end);
```

Jak lze vidět, dojde k odstranění dat, která byla přidána při prokládání.

4.2.5 RS dekodování

Pro dekodování jsou vytvořeny struktury dekodéru. Vstupní signál je taktéž rozdělen do $\log_2(N+1)$ sloupců a musí být celočíselným násobkem K . Pro ukázkou zápis RS dekodéru:

```
N = 255; K = 239 ;  
Dekodér = fec.rsdec(N,K);  
msg = reshape(msg,[],log2(N+1));  
Dekodovaný_signál = decode(msg,Kodér);
```

4.2.6 Konvoluční Dekodování

Pro konvoluční dekodování je potřeba stejná struktura, která byla použita ke kódování, vytvořená stejným postupem. K dekodování je ovšem použita jiná funkce a to `vitdec` (viterbiho dekodování). Pro ukázkou uvádíme dekodér pro rate $2/3$, stejná struktura jakou jsme naznačili ve vysílači.

```

Struktura = poly2trellis([5 4],[23 35 0;0 5 13]);
Dekódovaný_signál = vitdec(msg,Struktura,5,'trunc','hard');

```

Jak si lze všimnout, je zde několik dalších proměnných, které jsou závislé na podobě dat, která kódujeme.

4.2.7 DeRandomizace

Probíhá v podstatě stejně jako randomizace. Máme stejný PRBS generátor, který nám vytvoří randomizační sekvenci, která potom projde obvodem xor společně s dekódovanými daty. Po derandomizaci by výsledkem měl být binární signál, stejný jaký byl na vstupu vysílače v případě, že neuvažujeme chyby.

4.3 Prostředí pro simulace

Model radiového prostředí byl nahrazen třemi typy kanálů. Daný kanál byl vložen mezi vysílač a přijímač a byly u něho nastaveny určité hodnoty. Použity byly následující druhy kanálů (viz tabulka 7).

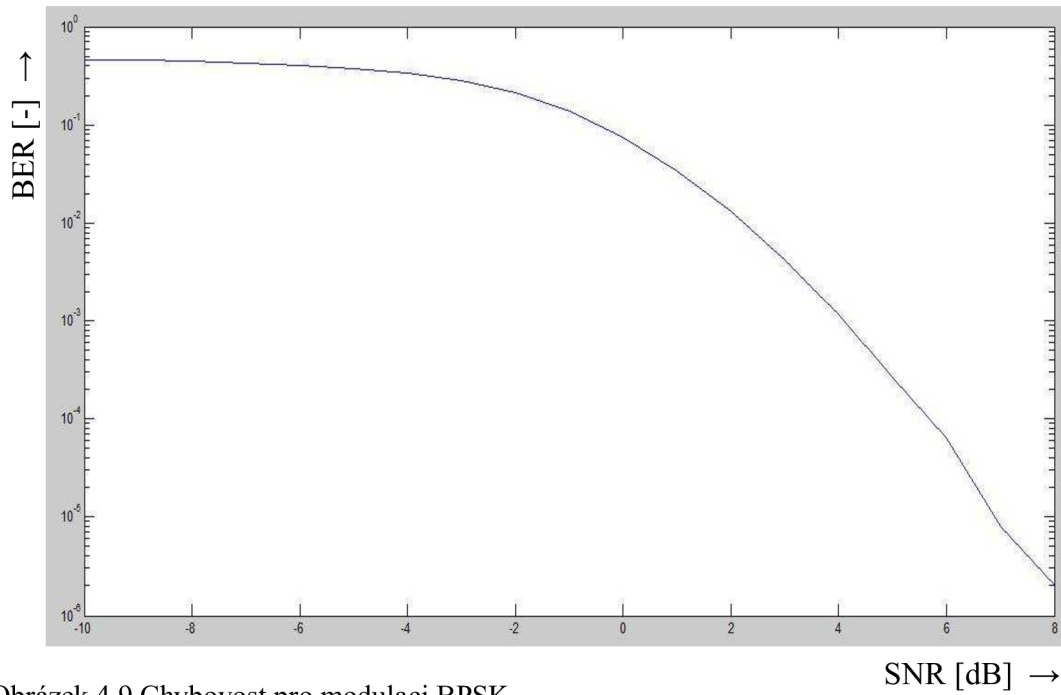
Tabulka 8 Vlastnosti jednotlivých kanálů

Druh kanálu	Popis
AWGN	Přidává do signálu bílý gaussovský šum, který simuluje širokopásmové rušení. Hlavním nastavením je zde SNR.
Rayleigh	Modeluje případ, kdy se na přenosu signálu z vysílače podílejí pouze nepřímé (odražené) cesty
Rician	Modeluje případ, kdy se na přenosu signálu z vysílače podílejí jak přímá, tak i nepřímé (odražené) cesty

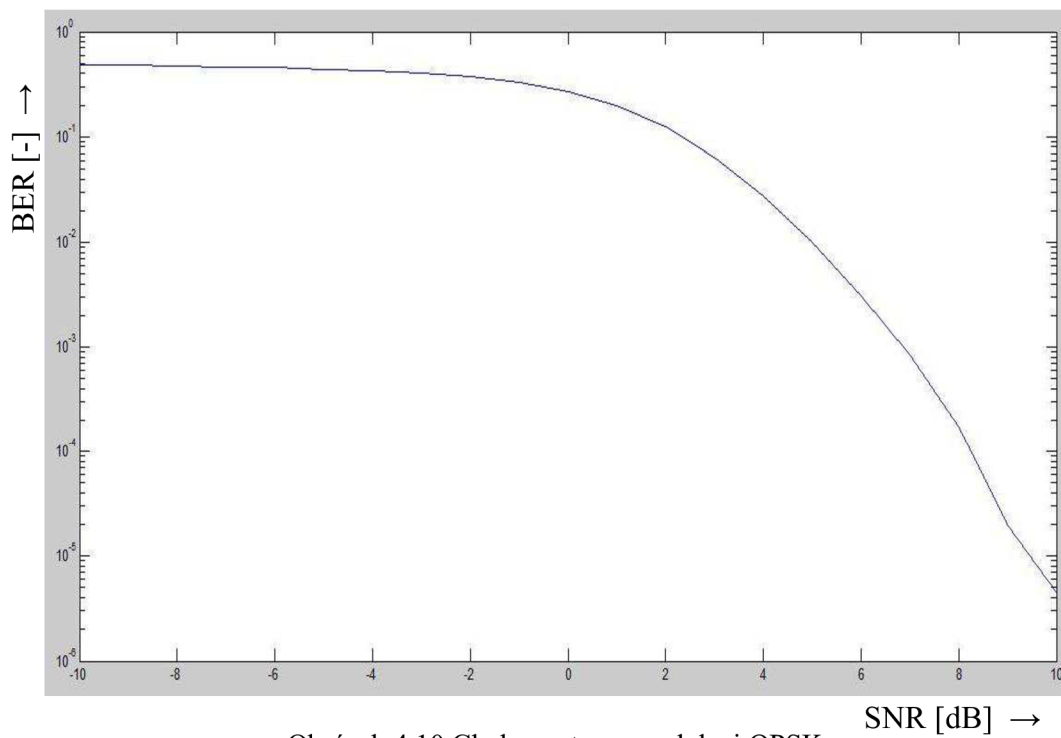
4.4 Výsledky

V této části práce lze nalézt grafy chybovosti v závislosti na SNR popřípadě další ukázky vlivu použitého nastavení na chybovost. Je potřeba říci, že ideální je použití turbo kódů namísto zřetěženého kódování.

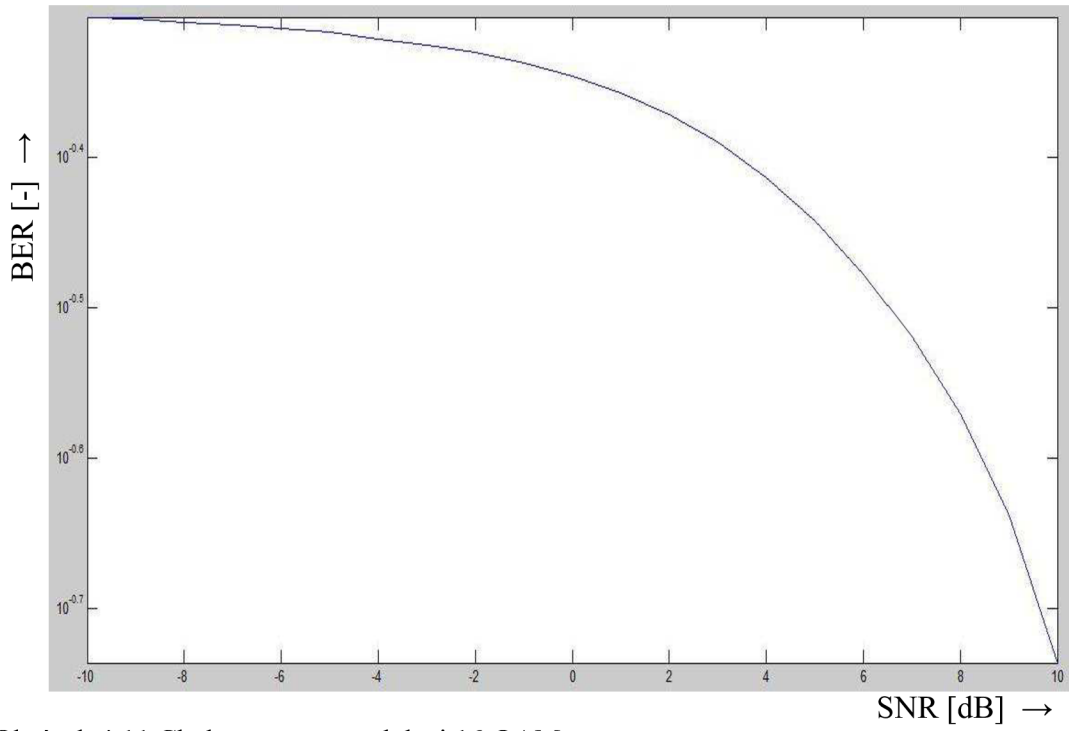
4.4.1 Chybovost pro AWGN



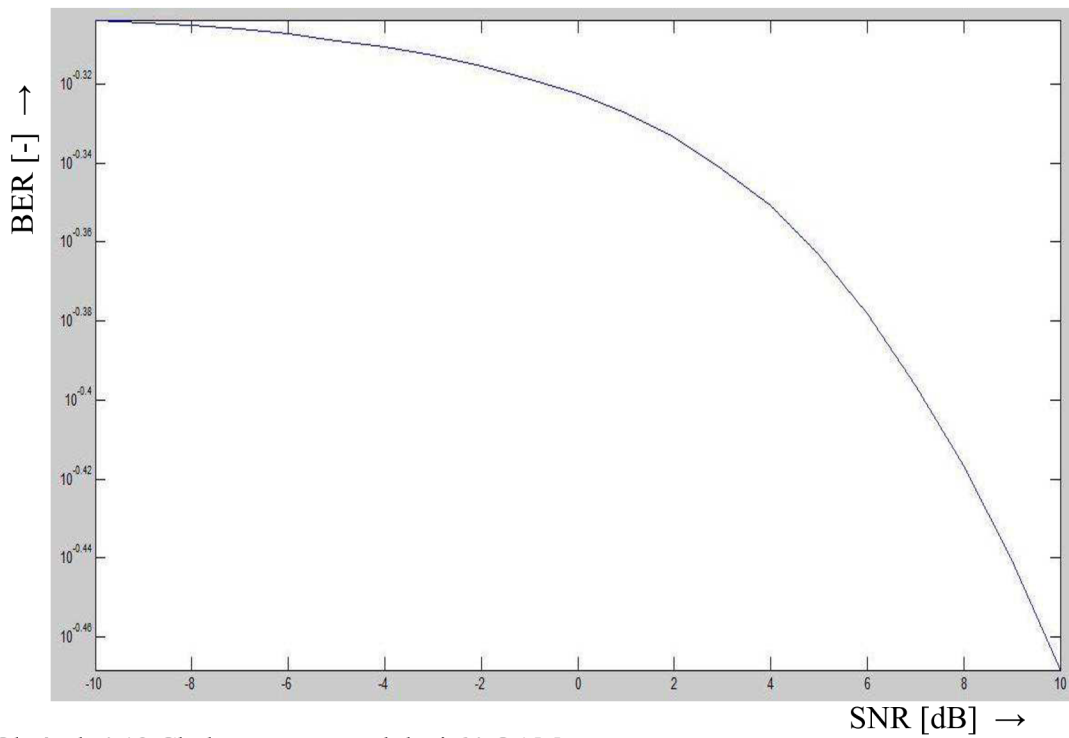
Obrázek 4.9 Chybovost pro modulaci BPSK



Obrázek 4.10 Chybovost pro modulaci QPSK



Obrázek 4.11 Chybovost pro modulaci 16-QAM

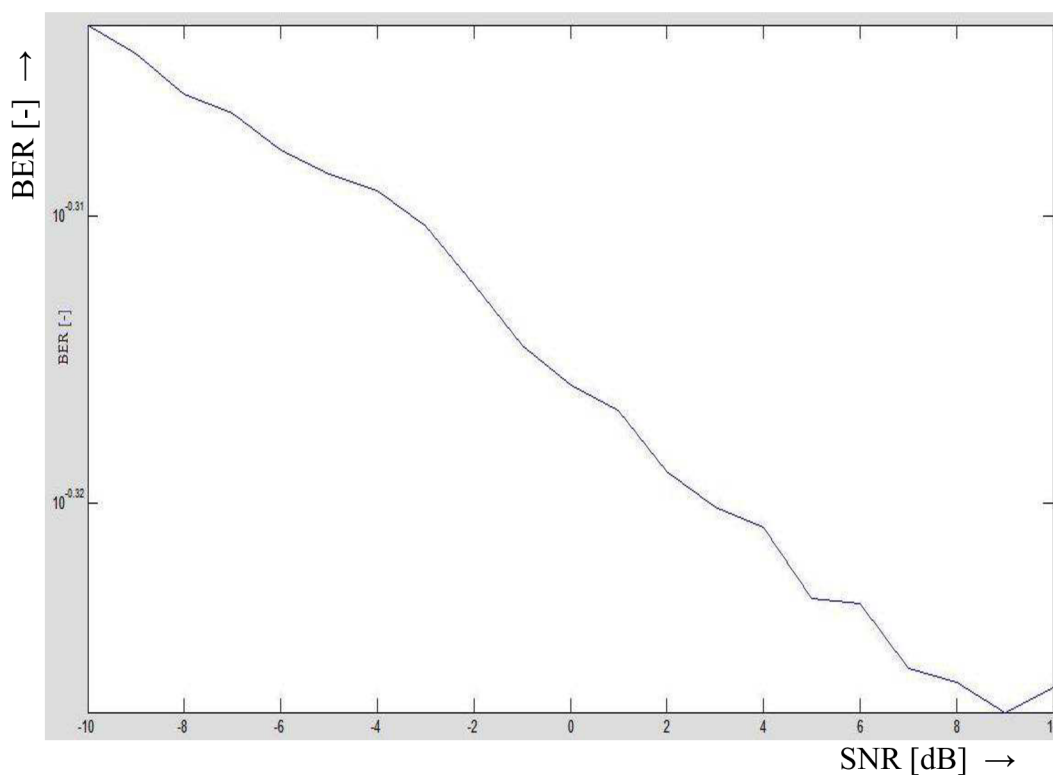


Obrázek 4.12 Chybovost pro modulaci 64-QAM

Tabulka 9 Nastavení modelu při simulacích v AWGN kanálu

Modulace	CC rate	RS	Prokládání
BPSK	1/2	255/239	Maticové
QPSK			
16-QAM			
64-QAM			

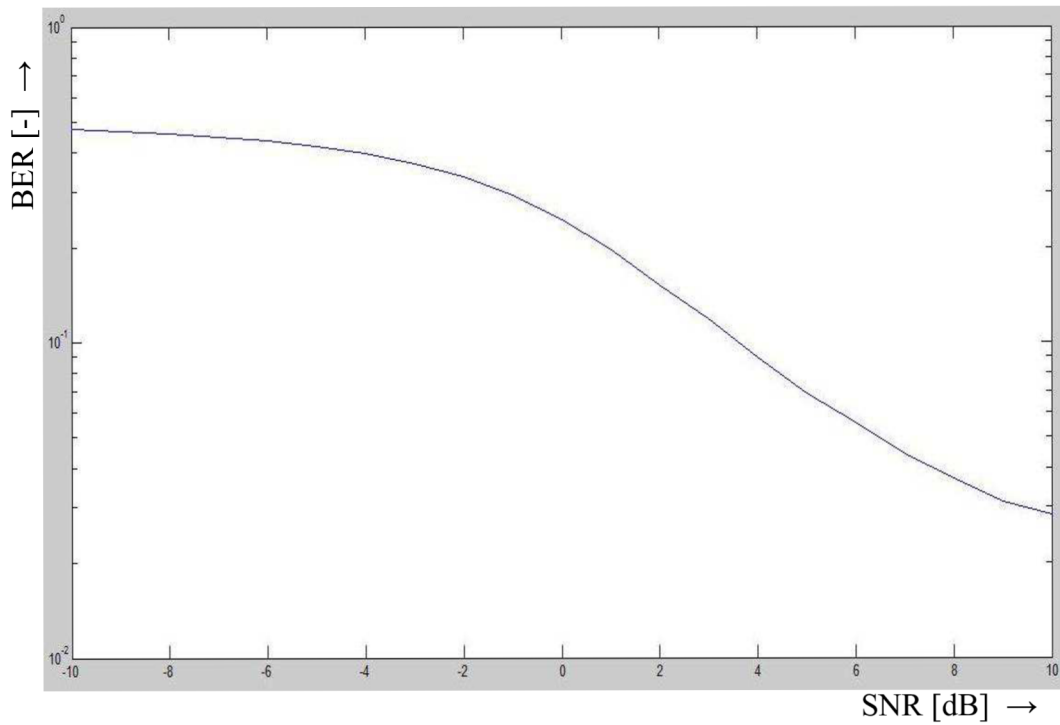
4.4.2 Chybovost pro Rayleighův kanál



Obrázek 4.13 Chybovost pro Rayleighův kanál

Simulace pro Rayleighův kanál byla pořízena s nastavením RS kodéru na hodnotu 255,239. Konvoluční kodér byl nastaven na rate 1/2 , bylo zvoleno maticové prokládání a použita BPSK modulace.

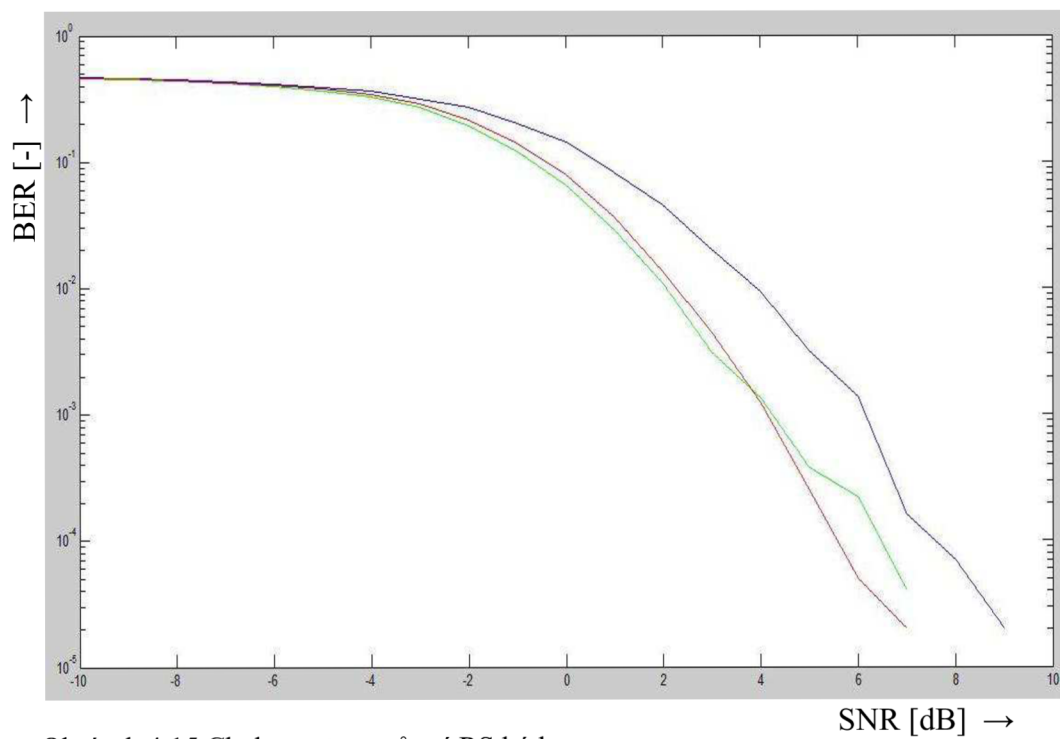
4.4.3 Chybovost pro Riciův kanál



Obrázek 4.14 Chybovost pro Ricianův kanál

Nastavení ponecháno stejné jako pro Rayleighův kanál.

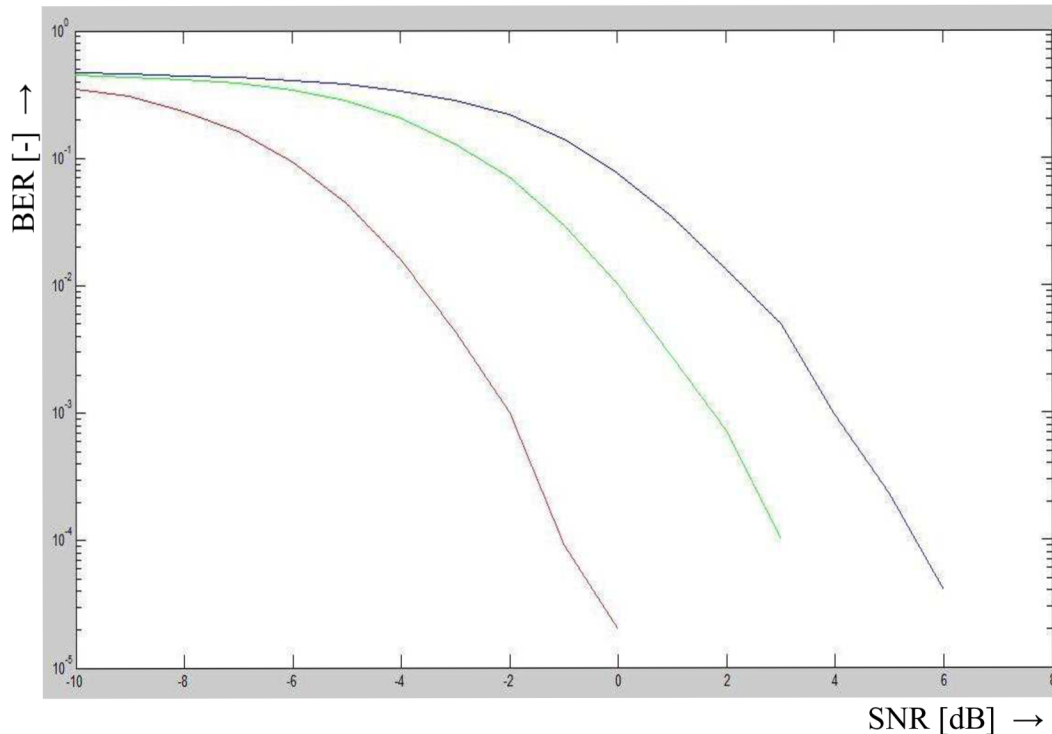
4.4.4 Chybovost pro různé RS kodéry



Obrázek 4.15 Chybovost pro různé RS kodéry

Modrá barva označuje kód o nastavení 7,3; zelená barva 127,119; červená barva 255,239. Konvoluční kodér byl při simulacích nastaven na rate 1/2 , bylo zvoleno maticové prokládání a použita BPSK modulace.

4.4.5 Chybovost pro různé rate CC kodéru



Obrázek 4.16 Chybovost pro různé rate CC

Modrá barva označuje kodér o rate 1/2, zelená 1/3 a červená 1/6. Z výsledků lze pozorovat, že kodéry přidávající větší redundanci chrání proti chybám lépe. Následující výsledky byly pořízeny pro BPSK modulaci, s použitím RS kódu 255,239 a maticového prokládání.

5 ZÁVĚR

Úspěšně se podařilo realizovat model fyzické vrstvy systému WiMAX v programovém prostředí Matlab. Model jsme vytvářeli jako skupinu skriptů, kde každý skript realizuje část vysílače či přijímače. Na všechny tyto skripty je odkazováno z hlavního programu, který určuje pořadí těchto bloků, upravuje signál na potřebnou velikost a také slouží jako model radiového prostředí.

Pokud přejdeme k samotnému modelu a jeho částem, realizovány byly všechny části přijímače a vysílače, až na konvoluční dekodér o rate 5/6. Byla tedy realizována randomizace, konvoluční kodéry o rate 1/2, 1/3, 1/6, 2/3, Reed-Solomův kodér o velikostech 255,239; 127,119; 7,3; tři druhy prokládání – konvoluční, maticové a pseudonáhodné. Dále digitální modulátory BPSK, QPSK, 16-QAM, 64-QAM a IFFT operaci, která nám vytváří spektrum.

Simulace byly pořízeny pro tři druhy radiových kanálů, jejichž popis lze nalézt v kapitole 4.3. Je patrné, že u simulací únikových kanálů vykazuje náš model poměrně vysokou chybovost. To je způsobeno absencí cyklického prefixu, vstupního a výstupního filtru.

Na přiloženém CD (zadní strana desek) lze nalézt celý zdrojový kód práce. A to hlavní program, všechny skripty realizující části přijímače či vysílače, elektronickou verzi textu bakalářské práce a návod k obsluze modelu. Zdrojové kódy daného modelu je též možné nalézt v příloze.

LITERATURA

- [1] IEEE Std 802.16-2009, IEEE DRAFT Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems, IEEE, 2009, 2004s, ISBN 978-0-7381-5919-5 STD95914
- [2] DOBEŠ, J., ŽALUD, V. Moderní radiotechnika. 1. vyd. Praha : BEN - technická literatura, 2006. 767 s. ISBN 80-730-0132-2.
- [3] C. YANG, Samuel. *OFDMA : System analysis and design*. [s.l.] : ARTECH HOUSE, 2010. 289 s. ISBN 978-1-60807-076-3.
- [4] GI LEE, Byeong; CHOI, Sunghyun. *Broadband Wireless Access and Local Networks : Mobile WiMAX and WiFi*. 2008. [s.l.] : ARTECH HOUSE, 2008. 618 s. ISBN 978-1-59693-293-7.
- [5] S. R. DINIZ, Paulo; A. B. DA SILVA, Eduardo; L. NETTO, Sergio. *Digital Signal Processing : System Analysis and Design*. [s.l.] : Cambridge University Press 2002, 2010. 877 s. ISBN 978-0-511-78983-0.
- [6] LIU, Hui a Guoqing LI. *OFDM-Based Broadband Wireless Networks: Design and Optimization*. Canada: John Wiley, 2005. ISBN 0-47L-7234b-0. <http://www.complextoreal.com/convo.htm>
- [7] Radio-Electronics.com: Resources and analysis for electronic engineers. [online]. [cit. 2011-12-29]. Dostupné z: <http://www.radio-electronics.com/>
- [8] Linear block codes. [online]. [cit. 2011-12-29]. Dostupné z: [http://www.uotechnology.edu.iq/dep-eee/lectures/4th/Communication/Information theory/5.pdf](http://www.uotechnology.edu.iq/dep-eee/lectures/4th/Communication/Information%20theory/5.pdf)
- [9] KÄSPER, Emilia. Turbo Codes. [online]. [cit. 2011-12-29]. Dostupné z: <http://users.tkk.fi/pat/coding/essays/turbo.pdf>
- [10] Function Reference: Communication System Toolbox. *Mathworks: MATLAB and Simulink for technical computing* [online]. [cit. 2012-01-01]. Dostupné z: <http://www.mathworks.com/help/toolbox/comm/ref/a1037894415.html>
- [11] OFDM, OFDMA, and SOFDMA in WiMAX: page 2 of 3 OFDMA. *Conniq.com: A beginner's Guide to connectivity and Networking* [online]. [cit. 2012-01-01]. Dostupné z: <http://www.conniq.com/WiMAX/fdm-ofdm-ofdma-sofdma-02.htm>
- [12] MARŠÁLEK, Roman. *Teorie rádiové komunikace: rozpracovaná verze II*. Skripta VUT v Brně.
- [13] ŠEBESTA, Vladimír a Zdeněk SMÉKAL. *Signály a systavy*. Skripta VUT v Brně.
- [14] SHARMA, Deepak. *Study paper on scalable OFDM Access: SOFDMA* [online]. 2010-04-05 [cit. 2012-05-21]. Dostupné z: <http://www.slideshare.net/deepakecrbs/ofdma-basics>
- [15] WiMAX Forum. [online]. 2001, 2012 [cit. 2012-05-21]. Dostupné z: <http://www.wimaxforum.org/>
- [16] *WiMAX Forum: Network Architecture*. 3.2. 2009. WMF-T32-002-R010v05. Dostupné z: <http://www.wimaxforum.org/resources/documents/technical/T32>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

<i>4G</i>	Sít' čtvrté generace
<i>ADC</i>	A/D převodník
<i>ARQ</i>	Automatic Repeat-reQuest
<i>BER</i>	Bit Error Rate
<i>BPSK</i>	Binary PSK
<i>BS</i>	Basic Station, Základnová stanice
<i>BW</i>	Bandwith, Šířka pásma
<i>CC</i>	Konvoluční kodér
<i>CDMA</i>	Code-Division Multiple Access
<i>CP</i>	Cyklický prefix
<i>CPS</i>	Common Part Sublayer
<i>CSN</i>	Connectivity Service Network
<i>DAC</i>	D/A převodník
<i>DFT</i>	Discrete Fourier Transformation
<i>DHCP</i>	Dynamic Host Control Protocol
<i>FDD</i>	Frequency Division Duplex
<i>FDMA</i>	Frequency-Division Multiple Access
<i>FEC</i>	Forward Error Correction
<i>FFT</i>	Fast Fourier Transformation
<i>FIR</i>	Finite Impulse Response
<i>GSM</i>	Global System for Mobiles
<i>HARQ</i>	Hybrid Automatic Repeat-reQuest
<i>IDFT</i>	Inverse Discrete Fourier Transformation
<i>IFFT</i>	Inverse Fast Fourier Transformation
<i>IIR</i>	Infinite Impulse Response
<i>ISI</i>	InterSymbol Interference
<i>MAC</i>	Media Access Control
<i>MS</i>	Mobile Station, Mobilní stanice
<i>NLOS</i>	None Line of Sight
<i>LOS</i>	Line of Sight
<i>OFDM</i>	Orthogonal Frequency Division Multiplex
<i>OFDMA</i>	Orthogonal Frequency Division Multiple Access

<i>PCM</i>	Pulse Code Modulation
<i>PHY</i>	Physical Layer
<i>PSC</i>	Paralel to Serial Converter
<i>PSK</i>	Phase Shift Keying
<i>QAM</i>	Quadrature Amplitude Modulation
<i>QoS</i>	Quality of Services
<i>QPSK</i>	Quadrature PSK
<i>RS</i>	Reed-Solomon
<i>SAP</i>	Service Access Point
<i>S/N</i>	Signal / Noise
<i>SNR</i>	Odstup signálu od šumu
<i>SOFDMA</i>	Scalable Orthogonal Frequency Division Multiple Access
<i>SPC</i>	Serial to Paralel Converter
<i>SQRT</i>	Square Root Raised Cosine
<i>TDD</i>	Time Division Duplex
<i>TDMA</i>	Time-Division Multiple Access
<i>UMTS</i>	Universal Mobile Telecommunications System

SEZNAM PŘÍLOH

A. zdrojové kódy

A.1	Hlavní program	44
A.2	Parametry Systému	47
A.3	Randomizace	48
A.4	Konvoluční kodér	48
A.5	RS kodér	49
A.6	Prokládání	49
A.7	Modulátory	50
A.8 I	FFT	51
A.9	FFT	51
A.10	Demodulátory	51
A.11	Opačné prokládání	52
A.12	RS dekodér	53
A.13	Konvoluční dekodér	53
A.14	Derandomizace	54

A. ZDROJOVÉ KÓDY

A.1 Hlavní program

```
%WiMAX 1.1.0
%-----
clear all
clc
%Základní nastavení signálu:
config = params(1);

switch(config.base.modulace);
    case 'BPSK'
        max = 1;
        signal = randi([0, max],1,config.base.N);
    case 'QPSK'
        max = 3;
        signal_de = randi([0, max],1,config.base.N);
        signal = reshape(de2bi(signal_de),1,[]);

    case '16-QAM'
        max = 15;
        signal_de = randi([0, max],1,config.base.N);
        signal = reshape(de2bi(signal_de),1,[]);
    case '64-QAM'
        max = 63;
        signal_de = randi([0, max],1,config.base.N);
        signal = reshape(de2bi(signal_de),1,[]);
end

%-----
%VYSÍLAČ
%Randomizace
signal_randomized = randomizator(config,signal);
%Kanálové kódování
%FEC
%Konvoluční Kodér
signal_coded_CC =
Konvolucni_kodery(config,signal_randomized);
%RS kodér
switch(config.base.RS_type)
    case '7,3'
        x_RS = length(signal_coded_CC);
        RS = ceil(x_RS/3);
        RS_new = RS*3;
        RS_add = RS_new - x_RS;
        signal_coded_CC_new =
[signal_coded_CC,zeros(1,RS_add)];
    case '127,119'
        x_RS = length(signal_coded_CC);
        RS = ceil(x_RS/119);
        RS_new = RS*119;
```

```

        RS_add = RS_new - x_RS;
        signal_coded_CC_new =
[signal_coded_CC, zeros(1,RS_add)];
        case '255,239'
            x_RS = length(signal_coded_CC);
            RS = ceil(x_RS/239);
            RS_new = RS*239;
            RS_add = RS_new - x_RS;
            signal_coded_CC_new =
[signal_coded_CC, zeros(1,RS_add)];
        end

        signal_coded_RS = RS_koder(config,signal_coded_CC_new);
%Interleaving
switch(config.base.Interleaving_type)
case 'pseudo_random'
    signal_coded_RS_new = signal_coded_RS;
    prok_add = 0;
case 'matrix'
    x_prok = length(signal_coded_RS);
    prok = ceil(x_prok/3);
    prok_new = prok*3;
    prok_add = prok_new - x_prok;
    signal_coded_RS_new =
[signal_coded_RS, zeros(1,prok_add)];
case 'conv'
    signal_coded_RS_new = signal_coded_RS;
    prok_add = 0;
end

    signal_prokladany =
Interleaving(config,signal_coded_RS_new);
%Repetition - Zvýšení dat. toku
%vynechavka ATM
%OFDMA
%Symbol Mapping (Modulation)
switch(config.base.modulace)
case 'BPSK'
    sig_1 = signal_prokladany;
    Mod_4 = 0;
case 'QPSK'
    Mod = length(signal_prokladany);
    Mod_2 = ceil(Mod/2);
    Mod_3 = Mod_2*2;
    Mod_4 = Mod_3 - Mod;
    sig_1 = [signal_prokladany, zeros(1,Mod_4)];
case '16-QAM'
    Mod = length(signal_prokladany);
    Mod_2 = ceil(Mod/4);
    Mod_3 = Mod_2*4;
    Mod_4 = Mod_3 - Mod;

```

```

        sig_1 = [signal_prokladany, zeros(1, Mod_4)];
    case '64-QAM'
        Mod = length(signal_prokladany);
        Mod_2 = ceil(Mod/6);
        Mod_3 = Mod_2*6;
        Mod_4 = Mod_3 - Mod;
        sig_1 = [signal_prokladany, zeros(1, Mod_4)];
    end
    signal_modulovany = Modulatory(config, sig_1);
    %Subcarrier Mapping
        %tuto položku nerealizujeme
    %IFFT
        N_fft = config.base.N_fft;
        IFFT_x = length(signal_modulovany);
        IFFT_y = ceil(IFFT_x/N_fft);
        prid = (IFFT_y*N_fft) - IFFT_x;
        signal_pom = [signal_modulovany, zeros(1, prid)];

        signal_IFFT = IFFT(config, signal_pom);
    %Filter
        %filter_out = Filtr_vystupni(config, signal_IFFT);
    %DAC
        %tuto položku nelze v programovém prostředí realizovat
    %RF-TX
        %--
%-----
    %RÁDIOVÉ PROSTŘEDÍ
    SNR = 30;
    sum = awgn(signal_IFFT, SNR);

%-----
    %PŘÍJÍMAČ

    %RF-RX
        %--          %vytvorit fci
    %ADC
        %tuto položku nerealizujeme
    %Filter
        %filter_in = Filtr_vstupni(config, sum);
    %OFDMA
        %FFT
        signal_FFT = FFT(config, sum);
        signal_FFT_new = signal_FFT(1:end-prid);
        %Subcarrier Demapping
            %tuto položku nerealizujeme
        %Symbol DeMapping (Demodulace)
            signal_demodulovany = Demodulatory(config, signal_FFT_new);
            signal_demodulovany_new = signal_demodulovany(1:end -
Mod_4);
        %DE-Repetition
            %vynechavka ATM

```

```

%Kanálové Kódování
    %Deinterleaving
        signal_deinterleaved =
DeInterleaving(config,signal_demodulovany_new);
        signal_deinterleaved_new= signal_deinterleaved(1:end -
prok_add );
    %FEC
        %RS DeKodér
            signal_RS_decoded =
RS_Decoder(config,signal_deinterleaved_new);
            signal_RS_decoded_new = signal_RS_decoded(1:end -
RS_add);
        %Konvoluční DeKodér
            signal_CC_decoded =
CC_Decoder(config,signal_RS_decoded_new);
    %DeRandomizace
        signal_new = De_randomizator(config,signal_CC_decoded);

```

A.2 Parametry systému

```

function config = params(in)
%%||||| Určení základních parametrů systému
|||||
BW = 23e6;           %šířka pásma
n = 8/7;           %sampling faktor
G = 1/32;          % CP time = 1/32 , 1/16 , 1/8, 1/4
N_fft = 128;       %velikost fft = 128,512,1024,2048
F_s = floor((n*BW/8000))*8000;      %vzorkovací frekvence
delta_f = F_s/N_fft;      %ochranný interval
T_b = 1/delta_f;         % užitečná doba symbolu
T_g = G*T_b;            % CP čas
T_s = T_b + T_g;        % čas OFDMA symbolu
T_samp = T_b/N_fft;     % vzorkovací čas ???
CC_rate = '1/2';       % CC koder rate
modulace = 'QPSK';     % Typ modulace
Interleaving_type = 'matrix';
N = 100;
RS_type = '7,3';       % dalsi 255,239 a 127,119

%=====
config.base.N = N;
config.base.BW = BW;
config.base.n = n;
config.base.G = G;
config.base.N_fft = N_fft;
config.base.F_s = F_s;
config.base.delta_f = delta_f;
config.base.T_b = T_b;
config.base.T_g = T_g;
config.base.T_s = T_s;
config.base.T_samp = T_samp;

```

```

config.base.CC_rate = CC_rate;
config.base.modulace = modulace;
config.base.Interleaving_type = Interleaving_type;
config.base.RS_type = RS_type;
%=====
end

```

A.3 Randomizace

```

function [out] = randomizator(config,signal)
%Randomizace
L1 = length(signal);
%výpočet randomizačního vektoru
randomizer_start_sequence=[1,0,0,1,0,1,0,1,0,0,0,0,0,0,0];

randomizer_new = zeros(1,L1);
    for i=1:L1

        y =
xor(randomizer_start_sequence(14),randomizer_start_sequence(15));

        %Bitové posuny realizující novou sekvenci
randomizer_start_sequence(15) = randomizer_start_sequence(14);
randomizer_start_sequence(14) = randomizer_start_sequence(13);
randomizer_start_sequence(13) = randomizer_start_sequence(12);
randomizer_start_sequence(12) = randomizer_start_sequence(11);
randomizer_start_sequence(11) = randomizer_start_sequence(10);
randomizer_start_sequence(10) = randomizer_start_sequence(9);
randomizer_start_sequence(9) = randomizer_start_sequence(8);
randomizer_start_sequence(8) = randomizer_start_sequence(7);
randomizer_start_sequence(7) = randomizer_start_sequence(6);
randomizer_start_sequence(6) = randomizer_start_sequence(5);
randomizer_start_sequence(5) = randomizer_start_sequence(4);
randomizer_start_sequence(4) = randomizer_start_sequence(3);
randomizer_start_sequence(3) = randomizer_start_sequence(2);
randomizer_start_sequence(2) = randomizer_start_sequence(1);
randomizer_start_sequence(1) = y;
        %Uložení nového bitu do paměti pro použití
        randomizer_new(i) = y;
    end

    %Výstupní signál
    out = xor(randomizer_new,signal);
end

```

A.4 Konvoluční kodér

```

function [out] = Konvolucni_kodery(config,signal_randomized)

```

```

%Konvolucni kodery
CCrate = config.base.CC_rate;
switch(CCrates)
    case '1/2'
        trel = poly2trellis(5,[37 33]); % 11111 , 11011
    case '1/3'
        trel = poly2trellis(5,[37 33 31]);
    case '1/6'
        trel = poly2trellis(5,[37 35 31 30 27 23]);
    case '2/3'
        trel = poly2trellis([5 4],[23 35 0;0 5 13]);
    case '5/6'
        trel = poly2trellis([5 4 3 5 4],[37 35 31 30 27 23;17 16
15 14 13 12;7 6 5 4 3 2;37 35 31 30 27 23;17 16 15 14 13 12;]);
end
%Kódování:
    out = convenc(signal_randomized,trel);
end

```

A.5 RS kodér

```

function [out] = RS_koder(config,signal_coded_CC_new)
RS_type = config.base.RS_type;
switch(RS_type)
%RS_koder
    case '7,3'
        signal_for_coding = reshape(signal_coded_CC_new,[],1);
        coder = fec.rsenc(7,3);
        signal_coded_1 = encode(coder,signal_for_coding);
        sig_1 = reshape(signal_coded_1,1,[]);
        sig_2 = de2bi(sig_1);
        out = reshape(sig_2,1,[]);
    case '127,119'
        signal_for_coding = reshape(signal_coded_CC_new,[],1);
        coder = fec.rsenc(127,119);
        signal_coded_1 = encode(coder,signal_for_coding);
        sig_1 = reshape(signal_coded_1,1,[]);
        sig_2 = de2bi(sig_1);
        out = reshape(sig_2,1,[]);
    case '255,239'
        signal_for_coding = reshape(signal_coded_CC_new,[],1);
        coder = fec.rsenc(255,239);
        signal_coded_1 = encode(coder,signal_for_coding);
        sig_1 = reshape(signal_coded_1,1,[]);
        sig_2 = de2bi(sig_1);
        out = reshape(sig_2,1,[]);

end
end

```

A.6 Prokládání

```

function [out] = Interleaving(config,signal_coded_RS_new)

```

```

%Interleaving
Interleaving_type = config.base.Interleaving_type;
switch(Interleaving_type)
    case 'pseudo_random'
        konst = length(signal_coded_RS_new);
        permutace_vektor=randperm(konst);
        out = intrlv(signal_coded_RS_new,permutace_vektor);
    case 'matrix' % Doplnit vypočet delky signalu
        Nrows=3;
        Nc = length(signal_coded_RS_new);
        Ncols=Nc /3;
        out = matintrlv(signal_coded_RS_new,Nrows,Ncols);
        permutace_vektor = 0;
    case 'conv'
        Nrows=3;
        Slope = 3;
        D = Nrows*(Nrows-1)*Slope
        signal_new = signal_coded_RS';
        signal_added = [signal_new;zeros(D,1)];
        out = convintrlv(signal_added,Nrows,Slope);
        permutace_vektor = 0;
end
end

```

A.7 Modulátory

```

function [out] = Modulatory(config,sig_1)
%Modulatory
modulace = config.base.modulace;
%Start
switch(modulace)
    case 'BPSK'
        %Vytvoreni modulatoru BPSK
        modulator =
modem.pskmod('M',2,'PhaseOffset',0,'SymbolOrder','binary','InputType'
e','bit');
        norm = 1;
        signal_pom = sig_1;
    case 'QPSK'
        %Vytvoreni modulatoru QPSK
        modulator =
modem.pskmod('M',4,'PhaseOffset',0,'SymbolOrder','gray','InputType'
,'integer');
        norm = 1;
        sig_2 = reshape(sig_1,[],2);
        sig_3 = bi2de(sig_2);
        signal_pom = reshape(sig_3,1,[]);
    case '16-QAM'
        %Vytvoreni modulatoru 16-QAM
        modulator =
modem.qammod('M',16,'PhaseOffset',0,'SymbolOrder','binary','InputTy
pe','integer');

```



```

        norm = 1/sqrt(10);
        sig_2 = reshape(sig_1,[],4);
        sig_3 = bi2de(sig_2);
        signal_pom = reshape(sig_3,1,[]);
    case '64-QAM'
        %Vytvoreni modulatoru 64-QAM
        modulator =
modem.qammod('M',64,'PhaseOffset',0,'SymbolOrder','binary','InputType',
'integer');
        norm = 1/sqrt(42);
        sig_2 = reshape(sig_1,[],6);
        sig_3 = bi2de(sig_2);
        signal_pom = reshape(sig_3,1,[]);
    end
    %Vlastni modulace
        signal_modulated = modulate(modulator, signal_pom);
        out = signal_modulated.*norm;
    end
end

```

A.8 IFFT

```

function [out] = IFFT (config,signal_pom)
    N_fft = config.base.N_fft;
    %převod dat serial -> paralel
        paralel=reshape(signal_pom,N_fft,[]);
    %Fourierovy transformace
        paralel_pom = ifft(paralel,N_fft,'nonsymmetric');
    %převod dat paralel -> serial
        serial=reshape(paralel_pom,1,[]);
        out = serial;
    end
end

```

A.9 FFT

```

function [out] = FFT (config,sum)
    N_fft = config.base.N_fft;
    %převod dat serial -> paralel
        paralel=reshape(sum,N_fft,[]);
    %Fourierovy transformace
        paralel_pom = fft(paralel,N_fft);
    %převod dat paralel -> serial
        pom=reshape(paralel_pom,1,[]);
        out = pom;
    end
end

```

A.10 Demodulátory

```

function [out] = Demodulatory(config,signal_FFT)

```

```

modulace = config.base.modulace;
switch(modulace)
    case 'BPSK'
        %Vytvoreni demodulatoru BPSK
        demodulator =
modem.pskdemod('M',2,'PhaseOffset',0,'SymbolOrder','binary','Output
Type','bit');
        norm = 1;
        output = demodulate(demodulator,signal_FFT);
    case 'QPSK'
        %Vytvoreni demodulatoru QPSK
        demodulator =
modem.pskdemod('M',4,'PhaseOffset',0,'SymbolOrder','gray','OutputTy
pe','bit');
        norm = 1;
        signal_pom = signal_FFT./norm;
        signal_demodulated = demodulate(demodulator, signal_pom);
        output = signal_demodulated(2,:);
    case '16-QAM'
        %Vytvoreni demodulatoru 16-QAM
        demodulator =
modem.qamdemod('M',16,'PhaseOffset',0,'SymbolOrder','binary','Outpu
tType','bit');
        norm = 1/sqrt(10);
        signal_pom = signal_FFT./norm;
        signal_demodulated = demodulate(demodulator, signal_pom);
        output = signal_demodulated(4,:);
    case '64-QAM'
        %Vytvoreni demodulatoru 64-QAM
        demodulator =
modem.qamdemod('M',64,'PhaseOffset',0,'SymbolOrder','binary','Outpu
tType','bit');
        norm = 1/sqrt(42);
        signal_pom = signal_FFT./norm;
        signal_demodulated = demodulate(demodulator, signal_pom);
        output = signal_demodulated(6,:);
end
    %Vystupni signal
    out = output;
end

```

A.11 Opačné prokládání

```

function [out] = DeInterleaving(config,signal_demodulovany_new)
Interleaving_type = config.base.Interleaving_type;
switch(Interleaving_type)
    case 'pseudo_random'
        out = deintrlv(signal_demodulovany_new,konst);
    case 'matrix'
        Nrows=3;
        Nc = length(signal_demodulovany_new);
        Ncols=Nc /3;

```

```

        out = matdeintrlv(signal_demodulovany_new,Nrows,Ncols);
    case 'conv'
        Nrows = 3;
        Slope = 3;
        D = Nrows*(Nrows-1)*Slope;
        b1 = convdeintrlv(signal_demodulovany_new,Nrows,Slope);
        out = b1(D+1:end);
    end
end
end

```

A.12 RS dekodér

```

function [out] = RS_Decoder(config,signal_deinterleaved_new)
RS_type = config.base.RS_type;
switch(RS_type)
    case '7,3'
        decoder = fec.rsdec(7,3);
        signal_pom_1 = reshape(signal_deinterleaved_new,[],3);
        signal_decoded = decode(decoder,signal_pom_1);
        out = reshape(signal_decoded(:,1),1,[]);
    case '127,119'
        decoder = fec.rsdec(127,119);
        signal_pom_1 = reshape(signal_deinterleaved_new,[],7);
        signal_decoded = decode(decoder,signal_pom_1);
        out = reshape(signal_decoded(:,1),1,[]);
    case '255,239'
        decoder = fec.rsdec(255,239);
        signal_pom_1 = reshape(signal_deinterleaved_new,[],8);
        signal_decoded = decode(decoder,signal_pom_1);
        out = reshape(signal_decoded(:,1),1,[]);
    end
end
end

```

A.13 Konvoluční dekodér

```

function [out] = CC_Decoder(config,signal_RS_decoded_new)
CC_rate = config.base.CC_rate;
switch(CC_rate)
    case '1/2'
        trel = poly2trellis(5,[37 33]); % 11111 , 11011
        out=vitdec(signal_RS_decoded_new,trel,5,'trunc','hard');
    case '1/3'
        trel = poly2trellis(5,[37 33 31]);
        out=vitdec(signal_RS_decoded_new,trel,5,'trunc','hard');
    case '1/6'
        trel = poly2trellis(5,[37 35 31 30 27 23]);
        out=vitdec(signal_RS_decoded_new,trel,5,'trunc','hard');
    case '2/3'
        trel = poly2trellis([5 4],[23 35 0;0 5 13]);
        out=vitdec(signal_RS_decoded_new,trel,5,'trunc','hard');
    case '5/6'
        trel = poly2trellis([5 4 3 5 4],[37 35 31 30 27 23;17 16
15 14 13 12;7 6 5 4 3 2;37 35 31 30 27 23;17 16 15 14 13 12;]);

```

```

        out=vitdec(signal_RS_decoded_new,trel,5,'trunc','hard');
end
end

```

A.14 DeRandomizace

```

function [out] = De_randomizator(config,signal_CC_decoded)
%Randomizace
L1 = length(signal_CC_decoded);
%výpočet randomizačního vektoru
randomizer_start_sequence=[1,0,0,1,0,1,0,1,0,0,0,0,0,0,0];
derandomizer_new = zeros(1,L1);
    for i=1:L1
        y =
xor(randomizer_start_sequence(14),randomizer_start_sequence(15));
        %Bitové posuny realizující novou sekvenci
        randomizer_start_sequence(15) = randomizer_start_sequence(14);
        randomizer_start_sequence(14) = randomizer_start_sequence(13);
        randomizer_start_sequence(13) = randomizer_start_sequence(12);
        randomizer_start_sequence(12) = randomizer_start_sequence(11);
        randomizer_start_sequence(11) = randomizer_start_sequence(10);
        randomizer_start_sequence(10) = randomizer_start_sequence(9);
        randomizer_start_sequence(9) = randomizer_start_sequence(8);
        randomizer_start_sequence(8) = randomizer_start_sequence(7);
        randomizer_start_sequence(7) = randomizer_start_sequence(6);
        randomizer_start_sequence(6) = randomizer_start_sequence(5);
        randomizer_start_sequence(5) = randomizer_start_sequence(4);
        randomizer_start_sequence(4) = randomizer_start_sequence(3);
        randomizer_start_sequence(3) = randomizer_start_sequence(2);
        randomizer_start_sequence(2) = randomizer_start_sequence(1);
        randomizer_start_sequence(1) = y;
        %Uložení nového bitu do paměti pro použití
        derandomizer_new(i) = y;
    end
    %Výstupní signál
    out = xor(derandomizer_new,signal_CC_decoded);
end

```