

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY  
ŠTUDIJNÝ ODBOR: TELEKOMUNIKÁCIE

---

Bc. Miroslav Ďurčík

Simulovanie prenosu dát  
prostredníctvom priestorovo-časových  
kódov

Diplomová práca

Vedúci diplomovej práce: Ing. Jozef Oboňa

December 2004



## **Anotácia**

Slovenská technická univerzita v Bratislave

Fakulta elektrotechniky a informatiky

Študijný odbor: Telekomunikácie

Autor: Bc. Miroslav Ďurčík

Diplomová práca: Simulovanie prenosu dát prostredníctvom priestorovo-časových kódov

Vedúci diplomovej práce: Ing. Jozef Oboňa

Dátum odovzdania: December 2004

Cieľom práce je predstaviť časovo-priestorové kódy, relatívne nový spôsob získania dodatočnej kapacity pri prenose dát cez Rayleigho bezdrôtový komunikačný kanál s pomalými únikmi. Podané sú základné princípy niekoľkých známych časovo-priestorových schém, napríklad blokové časovo-priestorové kódy, mriežkové časovo-priestorové kódy, ako aj výsledky simulácií príkladov niektorých kódov v porovnaní s prenosom bez kódovania. Okrem toho je opísaný aj STSym - balík programov vyvinutý pre potrebu vykonávania časovo náročných simulácií časovo-priestorových a iných typov kódov za pomoci viacerých počítačov.



# Annotation

Slovak University of Technology, Bratislava

Faculty of Electrical Engineering and Information Technology

Degree Course: Telecommunications

Author: Bc. Miroslav Ďurčík

Thesis: Data transmission simulation of space-time codes

Supervisor: Ing. Jozef Oboňa

Date of delivery: December 2004

The aim of this work is to present space-time codes, a different approach for gaining additional capacity when transmitting data over flat fading Rayleigh wireless channel. Basic principles of known space-time coding schemes are discussed and simulation results for some selected space-time block codes as well as some space-time trellis codes are presented, compared to transmission without any coding. One part is devoted to STSym, package of windows applications developed for the purpose of space-time and other codes distributive simulations.



## **Čestné prehlásenie**

Čestne prehlasujem, že diplomovú prácu som vypracoval samostatne pod odborným vedením Ing. Jozefa Oboňu s pomocou vlastných vedomostí a čerpaním z dostupnej literatúry uvedenej na konci práce.

---

Bc. Miroslav Ďurčík



## **Pod'akovanie**

Ďakujem touto cestou vedúcemu diplomovej práce Ing. Jozefovi Oboňovi za cenné rady a usmernenie počas vypracovania a nasledujúcim osobám a subjektom za poskytnutie výpočtových prostriedkov:

- Firma Webglobe s.r.o.
- Roland Ragályi
- Peter Fázik



# Obsah

Úvod .....	15
1. MIMO .....	16
2. Diverzita.....	17
3. Modely kanálov .....	18
3.1. BSC kanál.....	18
3.2. Kanál AWGN .....	18
3.3. Rayleigho kanál .....	18
4. Typy časovo-priestorových kódov .....	20
4.1. MRRC .....	21
4.2. Alamoutiho schéma .....	21
4.3. STBC.....	23
4.4. Mriežkové časovo-priestorové kódy.....	24
4.5. D-Blast .....	25
4.6. V-BLAST .....	27
4.7. Diferenčné Cayleyho kódy.....	28
4.8. Sférické dekódovanie.....	29
4.9. Iné typy ST kódov .....	29
5. Simulačné prostredie .....	30
6. STSym .....	31
7. Použité pojmy a techniky.....	33
7.1. Normálové – Gaussovo rozdelenie.....	33
7.2. Generovanie náhodných čísel s normálovým rozdelením .....	33
8. Distribuované simulovanie .....	34
8.1. Architektúra distribuovaného simulovania .....	35
Princíp architektúry klient – XML server.....	36
8.2. Prevod, transformácie XML dokumentov.....	37
8.3. Komunikačný protokol medzi komunikujúcimi.....	38
8.4. Životný cyklus autonómneho simulátora .....	40
8.5. Databáza XML súborov úloh .....	43
8.6. Správa vykonávaných simulácií .....	43
8.7. Dohľad nad simulátorom, inštalácia .....	43
8.8. Automatická aktualizácia novou verziou .....	44

9.	Výsledky simulácie .....	45
9.1.	Prvotné simulácie STBC kódu na jednom počítači .....	45
9.2.	PSK modulácia a STBC matice .....	46
9.3.	Presnejšie výsledky STBC .....	47
9.4.	Počet intervalov v STBC matici .....	48
9.5.	Simulovanie mriežkových časovo-priestorových kódov .....	49
9.6.	Zvyšovanie počtu stavov, antén, intervalov .....	49
9.7.	Viac prijímacích antén či viac blokov v rámci ? .....	53
10.	Zhrnutie .....	55
	Prílohy .....	57
A.	Príklady výpisu súborov definovania kódov .....	59
B.	Príklady grafického definovania STTC kódov .....	60
C.	Príklad XML súboru .....	61
D.	Interné číslovanie a názvy stavov NT servisu .....	63
E.	Popis tabuliek a stĺpcov databázy úloh .....	64
F.	Dokumentácia k aplikácii STSym .....	65
	Výber parametrov Reed-Solomon kódu .....	68
	Výber parametrov blokového časovo-priestorového kódu .....	68
	Výber parametrov mriežkového časovo-priestorového kódu .....	69
	Parametre simulácie .....	69
G.	Rozhranie manažovania vykonávaných simulácií .....	71
	Sekcia „Import/Export“ .....	71
	Sekcia „Transformácie“ .....	71
	Sekcia „Spracovanie“ .....	72
	Zoznam použitých skratiek .....	73
	Použitá literatúra .....	75

## Zoznam obrázkov

Obr. 1: Schéma MIMO prenosového systému.....	16
Obr. 2: Model kanálu BSC .....	18
Obr. 3: Model kanálu AWGN.....	18
Obr. 4: Schéma modelu Rayleigho kanála pre prípad 2 vysielacích a 2 prijímacích antén .....	19
Obr. 5: Klasická MRRC schéma s dvoma vetvami.....	21
Obr. 6: Alamoutiho schéma s dvoma vysielacími anténami .....	22
Obr. 7: Grafický popis mriežkového časovo-priestorového kódu s moduláciou 4 PSK, 4 stavmi a 2 b/s/Hz.....	25
Obr. 8: Diagonálne vrstvy metódy D-BLAST.....	26
Obr. 9: Príklad použitia programu STSym.....	32
Obr. 10: Získaný výskyt hodnôt v jednotlivých priechinkoch .....	33
Obr. 11: Prepojenie komunikujúcich strán .....	36
Obr. 12: Legenda k diagramu stavov a prechodov .....	41
Obr. 13: Schéma stavov a prechodov medzi stavmi servisu .....	42
Obr. 14: Graf závislosti chybovosti prenášaných informačných bitov od $E_b/N_0$ .....	46
Obr. 15: Závislosť chybovosti od $E_b/N_0$ pre vybraný STBC kód 2x2 s rôznou PSK moduláciou .....	47
Obr. 16: Presnejší graf závislosti vybraných STBC kódov od $E_b/N_0$ .....	47
Obr. 17: Závislosť BER od $E_b/N_0$ , 3 antény, voliteľný počet intervalov .....	48
Obr. 18: Závislosť chybovosti vybraných mriežkových kódov .....	49
Obr. 19: Vplyv počtu prijímacích antén na efektivitu STTC kódu.....	50
Obr. 20: Vplyv počtu stavov STTC mriežky na BER.....	51
Obr. 21: Vplyv počtu blokov v rámci na STTC kód s mriežkou 4m04s2b.stt .....	51
Obr. 22: Vplyv počtu blokov v rámci na STTC kód s mriežkou 4m32s2b.stt .....	52
Obr. 23: Vplyv počtu blokov v rámci na STTC kód s mriežkou 8m16s3b.stt .....	52
Obr. 24: Súbor matice STBC kódu t2a2b2.stm .....	59
Obr. 25: Súbor matice STBC kódu t4a4b4.stm .....	59
Obr. 26: Súbor matice STBC kódu t8a8b8.stm .....	59
Obr. 27: Súbor matice STBC kódu t4a3b4.stm .....	59
Obr. 28: Súbor matice STBC kódu t8a4b4_c.stm.....	59
Obr. 29: Súbor matice STBC kódu t6a3s4_c.stm.....	59
Obr. 30: Súbor STTC kódu 4m04s2b.stt.....	59

Obr. 31: Súbor STTC kódu 4m08s4b.stt .....	59
Obr. 32: Súbor STTC kódu 8m08s3b.stt .....	59
Obr. 33: 2-ST kód, 4-PSK, 4 stavy, 2 b/s/Hz.....	60
Obr. 34: 2-ST kód, 4-PSK, 8 stavov, 2 b/s/Hz.....	60
Obr. 35: 2-ST kód, 8-PSK, 8 stavov, 3 b/s/Hz.....	60
Obr. 36: 2-ST kód, 8-PSK, 16 stavov, 3 b/s/Hz.....	60
Obr. 37: Časti okna programu STSym .....	65
Obr. 38: Dialógové okno výberu Reed-Solomon kódu.....	68
Obr. 39: Dialógové okno výberu STBC kódu .....	69
Obr. 40: Dialógové okno výberu STTC kódu a ďalších parametrov prenosu.....	69
Obr. 41: Dialógové okno voľby parametrov simulácie.....	70

## Zoznam tabuliek

Tab. 1: Príklad STBC matice (Alamoutiho schéma) .....	23
Tab. 2: Tabuľka možných architektúr distribuovaného simulovania .....	35
Tab. 3: Zoznam a popis úrovní rozbitia STSym dokumentu.....	38
Tab. 4: Časová náročnosť vykonaných a potrebných simulácií simulovaním na jednom počítači bez predčasného ukončenia simulácií .....	46
Tab. 5: Tabuľka chybovosti BER STTC kódu 4m32s2b.stt, pri $E_b/N_0$ 0dB, rôzny počet prijímacích antén a počet blokov v rámci.....	53
Tab. 6: Popis stavov STSymS servisu.....	63
Tab. 7: Popis tabuliek a polí databázy úloh.....	64
Tab. 8: Implementované modely v programe STSym .....	67

# Úvod

Zvyšujúce sa nároky na prenos informácií v mobilnej oblasti motivuje inžinierov a vedcov k vývoju dokonalejších metód a kódov na ich prenos. Turbo kódy, používané už vyše desať rokov sa dlho zdali byť tým najlepším čo je možné v tejto oblasti vymyslieť. Napriek tomu vývoj pokračoval ďalej. Časovo-priestorové kódy, na ktoré je tento projekt zameraný, umožňujú samé o sebe bez dodatočnej bitovej nadbytočnosti prenášať informačné bity efektívnejšie a sú alternatívou, alebo ako doplnok k zložitým samo-opravným kódom.

Za úlohy tejto práce sme si stanovili niekoľko cieľov. Urobiť prierez v doteraz známych typoch časovo-priestorových kódov, podať základné informácie, ich výhody a nevýhody. Následne tým lepšie pochopiť a vytvoriť model blokového (STBC), mriežkového (STTC) časovo-priestorového kódera a dekódera pre prenos cez Rayleigho prenosový kanál. Vytvoriť len jednoduchý model Rayleigho prenosového kanála. Vytvoriť aplikáciu na simulovanie STBC, STTC a iných kódov. Neskôr doplniť ďalšie modely iných spôsobov kódovania pre účely porovnávania výsledkov, efektivity prenosu.

Pokiaľ chceme, aby boli výsledné grafy dostatočne kvalitné a priebehy hladké, je potrebné uskutočniť prenos veľmi veľkého počtu informačných bitov, čo robí simulácie veľmi časovo náročné. Z tohto dôvodu sme sa rozhodli pre rozšírenie samotného simulačného programu o sadu ďalších doplnkových programov, ktoré umožňujú úlohy distribuovať na viaceré počítače, na ktorých už simulácie prebiehajú paralelne šetriac celkový potrebný čas na ich vykonanie. Vid' kapitola „distribuované simulovanie“.

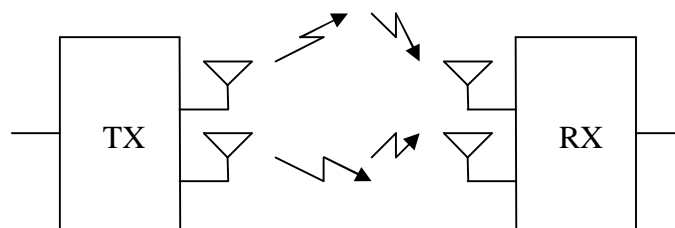
Výsledky simulovania prenosu dát rôznych kódov pri rôznych nastaveniach sme vyniesli do grafov a zhodnotili.

# 1. MIMO

MIMO, alebo aj „Multiple Input Multiple Output“, v preklade „viaceré vstupy a viaceré výstupy“ predstavuje novú oblasť prenosu dát bezdrôtovým kanálom. Jedná sa o systémy, ktoré na prenos používajú sústavu viacerých vysielacích a prijímacích antén. Patria sem aj systémy s jednou vysielacou a viacerými prijímacími anténami, alebo naopak viacerými vysielacími a jednou prijímacou anténou. Pokiaľ sme doteraz chceli vysielat' naraz viacerými anténami, museli predstavovať nezávislé prenosové kanály, napríklad s rozličnou modulačnou frekvenciou. Čo znamená, že na úkor väčšej energetickej spotreby a zväčšenia prenosového kanála sme získali väčšiu prenosovú rýchlosť alebo spoľahlivosť.

Odlišnosť týchto systémov je v tom, že prenos medzi vysielacími a prijímacími anténami prebieha v jednom komunikačnom kanály na zdieľanej frekvencii. Ktorý signál a z ktorej vysielacej antény je určený pre ktorú prijímaciu anténu stanovené nie je. Všetky signály sa prenášajú cez jedno médium – vzduch, v ktorom sa premiešajú a vzájomne ovplyvnia. Výsledný signál z vysielacej na prijímaciu stranu je formovaný samotným prostredím cez ktorý prechádza. Autá, stromy, kopce, budovy, pohybujúce sa i stojace predmety, všetky spôsobujú odrazy a oslabenie signálu. Takto vznikajú viaceré verzie signálu, ktoré pred tým, ako sa dostanú k niektorej prijímacej anténe, sa môžu tiež navzájom ovplyvniť – sčítať alebo odčítať. Odrazy prichádzajú aj v rôznom čase podľa dĺžky trasy, čo ešte viac komplikuje situáciu.

Zjednodušené schéma MIMO systému je na obr. 1.



Obr. 1: Schéma MIMO prenosového systému

## 2. Diverzita

Diverzita je pojem, ktorý charakterizuje princíp „porazenia“ nepriaznivého prenosového kanála. Všetky časovo priestorové kódovacie techniky sa snažia o vytvorenie diverzity – rozdielnosti vysielaných signálov. Spôsoby realizácie sú však rôzne. Jednotlivé vysielacie antény nevysielajú nezávislé toky symbolov. Vo vysielaní nastáva podstatná nadbytočnosť, opakovanie symbolov. Úspech dekódovania je možný len vtedy, ak sú vysielacie a prijímacie antény od seba dostatočne vzdialené. Pokiaľ by boli blízko seba, bola by len malá šanca, že prijaté signály sa budú odlišovať.

Medzi doteraz známe typy diverzity patria:

- **Časová diverzita** – Pre prenos viacerých nezávislých informácií cez spoločný prenosový kanál sa zadefinujú diskrétné časové intervaly počas ktorých môžu komunikujúci postupne jeden po druhom prenášať informácie.
- **Frekvenčná diverzita** – Komunikujúci vysielajú na rôznych frekvenciách.

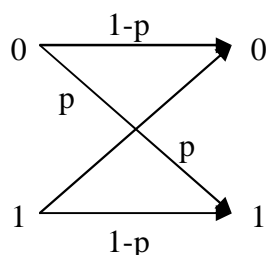
Nám však ide o inú diverzitu, nie o rozlíšenie viacerých nezávislých tokov (medzi rôznymi komunikujúcimi), ale skôr o lepšie využitie kapacity prenosového bezdrôtového kanála medzi dvoma komunikujúcimi. Ide o nový typ –**priestorovú diverzitu**. Respektíve:

- **Diverzita na strane prijímača**, kedy dodatočnú kapacitu získavame spracovaním signálov z viacerých prijímacích antén a odhadom charakteristík kanála. Možné je jednotlivé antény prepínať, vyberať vždy tú, ktorá prijíma najsilnejší signál, alebo použiť lineárnu kombináciu prijatých signálov.
- **Diverzita na strane vysieláča**, dosiahnutá zvýšením počtu vysielacích antén. Antény sa vo vysielaní môžu striedať, môže vysielateľ len tá ktorej signál je u prijímača najkvalitnejší (nutná spätná odozva od prijímača), alebo sú signály nimi vyslané fázovo posunuté. Sem patrí aj Alamoutiho schéma s jednou prijímacou anténou.
- **Kombinácia** – MIMO časovo priestorové schémy. Už spomínané STTC, STBC a iné časovo-priestorové schémy, ktorým sa ešte budeme podrobnejšie venovať v ďalších kapitolách.

### 3. Modely kanálov

#### 3.1. BSC kanál

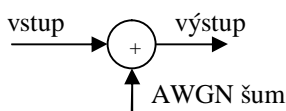
Binary Symmetric Channel (BSC) je kanál na prenos binárnych informácií s jediným parametrom  $p$  – pravdepodobnosť chybného prenosu bitu. Model je na obr. 2.



Obr. 2: Model kanálu BSC

#### 3.2. Kanál AWGN

Additive White Gaussian Noise (AWGN) kanál je najznámejší model kanála, používaný na simulovanie a porovnávanie efektivity prenosových systémov. V tomto kanáli nejde ani tak o zoslabenie prenášaného signálu, ale o vnesenie bieleho šumu do neho. Tento šum je s normálovým rozdelením funkcie hustoty náhodnej premennej, nulovou strednou hodnotou a jednotkou varianciou, ako aj smerodajnou odchýlkou. Schéma modelu je na obr. 3.



Obr. 3: Model kanálu AWGN

#### 3.3. Rayleighho kanál

Pre potreby simulácie časovo-priestorových kódov je potrebné simulovať prenosovú cestu medzi kóderom a dekóderom. Pre tieto potreby najlepšie vyhovuje Rayleighho model bezdrôtového kanála. Modeluje prenos medzi vysielačmi a prijímačmi, ktorí na seba nevidia. Každý ľubovoľný pár vysielačej a prijímačej antény, vid' obr. 4, predstavuje osobitnú cestu pre prenášaný signál, s vlastným tlmením a vzdialenosťou, ktorú musí signál prekonať kým dorazí k prijímaču. Vzdialenosť je závislá od existencie možných odrazov signálu na trase od väčších či menších predmetov. Výsledkom je, že každá prijímacia anténa prijme súčet

všetkých vyslaných signálov, no s rozdielnou mierou vplyvu tej ktorej vysielacej antény na celkový súčet. Jednotlivé prenosové cesty sa môžu a v praxi sa aj neustále menia. Prijatý signál sa vďaka týmto vplyvom môže javiť ako náhodný. Spôsoby ako prenášať dáta tak, aby s pomocou sústavy vysielacích a prijímacích antén bolo možné prenášať dáta aj cez opisované nepriaznivé prostredie však existujú. Rayleigho kanál všetky spomínané javy modeluje pomocou náhodného tlmenia medzi všetkými kombináciami prijímacích a vysielacích antén s normálovým rozdelením náhodnej premennej.

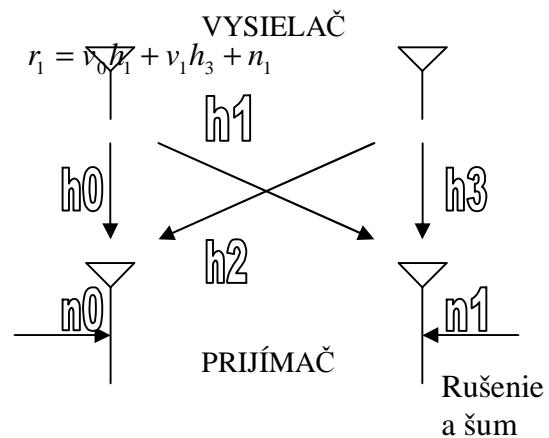
$$(3.3.1) \quad r_0 = v_0 h_0 + v_1 h_2 + n_0$$

Nutnou podmienkou úspechu týchto metód je dokonale poznať charakteristiky kanála. V praxi sa charakteristiky kanála pravidelne merajú tréningovými postupnosťami. Pre potreby našej simulácie toto nie je potrebné. Naš kanál bude

kvázistatický, kedy sa jeho charakteristiky počas prenosu jedného bloku nebudú meniť a zároveň vždy budeme poznať útlmové hodnoty jednotlivých ciest.

Tlmenie nie je jediný problém tohto kanála. Hlavným dôvodom vzniku chýb na prenose je šum. Ďalším javom ktorý pri prenose cez bezdrôtový kanál nastáva je medzi symbolová interferencia (ISI). Spôsobená je rozdielnosťou časov šírenia signálov jednotlivými cestami. Následkom je, že symboly vyslané časovo po sebe sa v prijímači môžu vzájomne ovplyvniť. Vysielané symboly sú reprezentované komplexným číslom, miera vplyvu signálu na prijatý signál je komplexné náhodné číslo s normálovým rozdelením a jednotkovou varianciou. Šum je komplexné náhodné číslo s normálovým rozdelením a varianciou závislou od pomeru SNR či  $E_b/N_0$ . Reálna zložka utlmí signál čo do veľkosti, imaginárna spôsobí oneskorenie. Matematický zápis modelu je popísaný vzorcami (3.3.1).

Tento model prenosového kanála je relatívne jednoduchý, avšak existujú dôkazy, že nemodeluje úniky úplne presne. Nie je ani frekvenčne selektívny (nie je dôležité na akej frekvencii je signál namodulovaný), medzi-symbolová interferencia nie je dobre modelovaná.



Obr. 4: Schéma modelu Rayleigho kanála pre prípad 2 vysielacích a 2 prijímacích antén

## 4. Typy časovo-priestorových kódov

Medzi známe časovo-priestorové kódy a schémy patria:

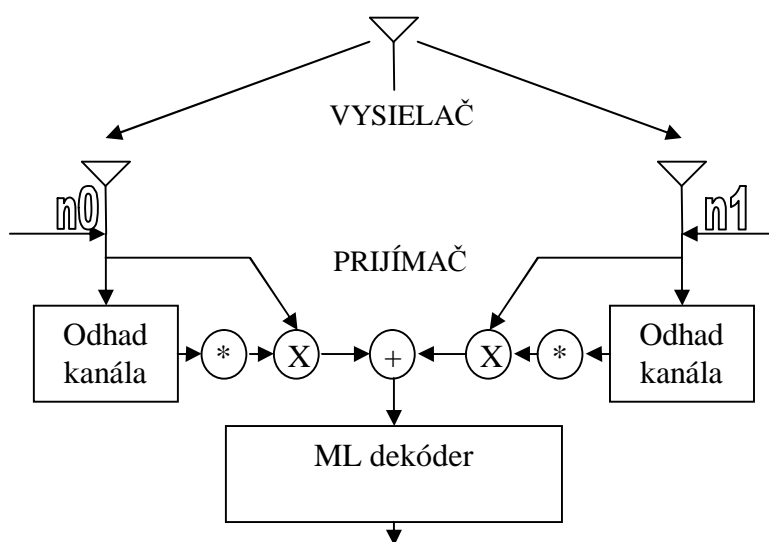
- **MRRC** – tradičný spôsob kombinovania prijatých signálov z rôznych antén využívaný aj v GSM bázových stanicách.
- **Alamoutiho schéma** – najjednoduchšia s dvoma vysielačmi a jednou alebo viacerými prijímacími anténami.
- **STBC** – blokové časovo-priestorové kódy. Kódovanie pomocou matice kódu. Dekódovanie je výberom symbolov, ktoré boli vyslané s najväčšou pravdepodobnosťou porovnaním s lineárnou kombináciou prijatých signálov v priestore a čase.
- **STTC** – Mriežkové časovo-priestorové kódy. Mapovanie vstupu do prechodov medzi stavmi v prechodovom diagrame. Princíp je podobný konvolučnému kódu, avšak jeden stav je definovaný cez viacero vysielačích antén. Dekódovanie je možné aj výberom najpravdepodobnejšej možnosti, alebo Viterbiho dekóderom.
- **Cayleyho kódy**; diferenčné časovo-priestorové kódy, nevyžadujú úplnú znalosť kanála na strane prijímača. Využíva Cayleyho transformáciu a špeciálne Hermitove matice, ktoré kód definujú.
- **BLAST** – „Bell Labs Layered Space-Time“, diagonálna BLAST schéma (D-BLAST), vertikálna BLAST schéma (V-BLAST). Kódovanie nie je potrebné, každá anténa vysiela vlastné dáta, avšak dekóder je o to náročnejší. Využíva štatistické nulovanie signálov.
- Modifikácie a kombinácie predošlých schém.
- **GLST**<sup>1</sup>
- Modifikovaná sférická metóda (zníženie náročnosti dekódovania) [13]
- A iné...

---

<sup>1</sup> Tao Meixia, Cheng S. R.: „Generalized Layered Space-Time Codes for High Data Rate Wireless Communications“, IEEE 2004

## 4.1. MRRC

Klasická metóda získavania diverzity, používaná veľmi dlho a často napríklad v GSM bázových stanicích. Bázovej stanici umožňuje kombinovať rôzne verzie, odrazy rovnakého signálu pre dosiahnutie menšej chybovosti, väčšej rýchlosti prenášaných dát z prenosného terminálu k bázovej stanici. Metóda má však obmedzenie v tom, že sa hlavne z ekonomických dôvodov nedá použiť na prenosnom terminály. Opis metódy spolu s porovnaním s Alamoutiho schémou je možné nájsť v [1]. Princíp prenosu touto metódou je zobrazený na obr. 5.



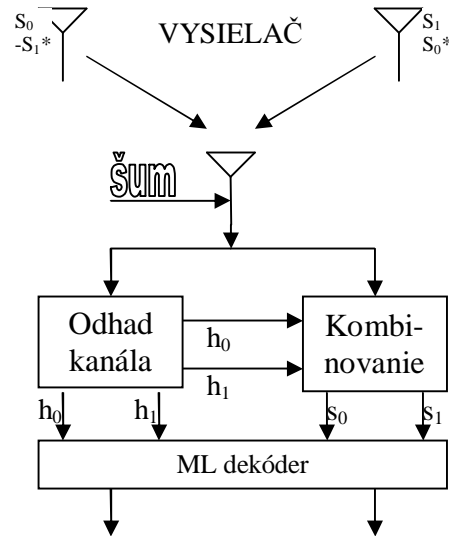
Obr. 5: Klasická MRRC schéma s dvoma vetvami

## 4.2. Alamoutiho schéma

Táto jednoduchá schéma s dvoma vysielačmi a jednou prijímacou anténou bola prezentovaná v článku [1]. Je ju možné rozšíriť aj na viacej vysielačích antén. Umožňuje, aby aj prenosné terminály mali zisk z použitia časovo-priestorových kódov bez nutnosti pridávania dodatočných antén na terminál. Dodatočné antény, tentoraz pre vysielač smer, sú pridané na bázovej stanici a slúžia tak všetkým komunikujúcim terminálom.

Princíp metódy je zobrazený na obr. 6. Vstupom do kódera sú dva symboly  $S_0$  a  $S_1$ , ktoré sú spracované počas dvoch časových intervalov. V prvom intervale sú symboly vyslané bez zmeny. Prvý symbol  $S_0$  prvou anténou a druhý symbol  $S_1$  druhou anténou. V druhom intervale sú signály navzájom vymenené, komplexne konjugované a signál  $S_1$  aj negovaný. V prijímači sú za pomoci úplnej znalosti prenosových ciest kanála výpočtom (4.2.1) spätne

odhadnuté. Úplná znalosť kanála sa získa odhadom tréningovými postupnosťami. Pri simuláciách, okrem prípadu navrhovania samotného systému na odhad kanála, toto nie je potrebné, keďže je známy.



Obr. 6: Alamoutiho schéma s dvoma vysielačimi anténami

$$(4.2.1) \quad S_0 = h_0^* r_0 + h_1 r_1^* + h_2^* r_2 + h_3 r_3^* \quad S_1 = h_1^* r_0 - h_0 r_1^* + h_3^* r_2 - h_2 r_3^*$$

### 4.3. STBC

Najjednoduchším typom časovo-priestorových kódov sú blokové časovo-priestorové (STBC) kódy. Informačné symboly sú vysielané v blokoch. Štandardná veľkosť bloku je 2, 4, alebo 8 symbolov na jeden blok. Počet vysielacích a prijímacích antén môže byť taký istý ako je počet symbolov, alebo menší. V každom časovom intervale sa cez kanál preniesie jeden informačný symbol. Vysielajú všetky antény, každá buď originálny - modifikovaný symbol daného časového okamihu, alebo symbol iného časového okamihu z bloku. Na druhej strane prijímač dokáže odhadnúť prijaté signály (4.2.1), pretože vie, symbol ktorého časového okamihu mal byť vysielaný z ktorej antény, a vie aj charakteristiku kanála. Potom pomocou kritéria najväčšej podobnosti (ML) vypočíta, čo bolo približne v každom okamihu vyslané. S narastajúcim počtom antén sa mu to oveľa lepšie. Princíp ML kritéria je popísaný nerovnicou (4.3.1), kde  $d^2$  je Euklidovská vzdialenosť dvoch signálov, symbolov. Signál  $S_0$  je odhadnutý, ku ktorému sa nájde najbližší symbol. Tab. 1 obsahuje príklad matice jedného STBC kódu, ktorým je Alamoutiho schéma.

	Vys.1	Vys.2
Čas 0	$S_0$	$S_1$
Čas 1	$-s_1^*$	$S_0^*$

Tab. 1: Príklad STBC matice (Alamoutiho schéma)

$$(4.3.1) \quad d^2(s_0, s_i) \leq d^2(s_0, s_k) \quad \text{pre všetky } i \neq k$$

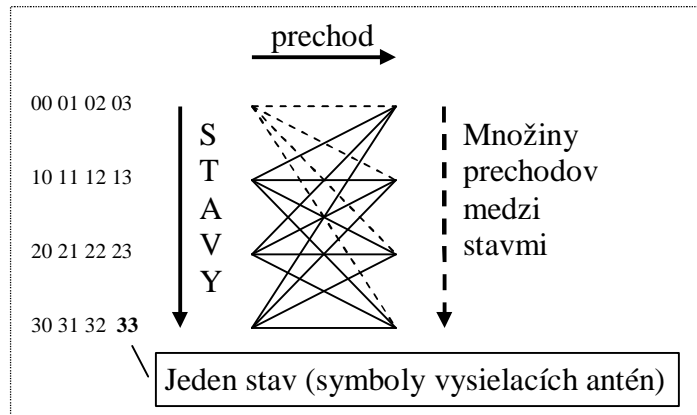
Každý špecifický STBC kód je popísaný svojou kódovacou maticou [3]. Matica má toľko riadkov, koľko je časových intervalov a zároveň aj počet prenášaných symbolov. Počet antén na oboch stranách je zvyčajne tiež zhodný s počtom prenášaných symbolov. Môže však byť aj menší, pokiaľ je kód ešte účinný. Číslo na danej pozícii znamená index vysielaného symbolu v aktuálnom bloku, vysielaný v časovom okamihu danom číslom riadku a vysielacou anténou danou číslom vysielacej antény ako stĺpca matice. Matica je potrebná aj pri dekódovaní, aby dekóder zistil, na akej pozícii sa ten ktorý vysielaný symbol nachádza v každom riadku, teda ktorá anténa ho v tom okamihu vyslala.

#### **4.4. Mriežkové časovo-priestorové kódy**

Mriežkové časovo-priestorové kódy (STTC) predstavujú jeden z prvých typov časovo-priestorových kódov. Spôsob kódovania je podobný mriežkovo kódovanej modulácii (TCM), s tým rozdielom, že jeden stav je definovaný cez všetky vysielacie antény.

Základným pojmom je stav. Stavom sa nazýva množina signálov, ktoré sú vysielané všetkými vysielacími anténami. V jednom časovom okamihu môže byť vysielateľ len v jednom stave. Počet stavov používaných v kódovaní či dekódovaní je obmedzený, jeho maximálna hodnota závisí od použitej modulácie, počtu vysielacích antén. Avšak nie každý stav môže byť použitý v hociktorom časovom intervale. Obmedzená je aj množina možných prechodov medzi stavmi. Potom kódovanie prenášaných informácií nie je mapovanie bitov na vysielané signály antén, ale mapovanie vstupných bitov na prechody medzi stavmi do nasledujúceho časového intervalu. Kódy sa definujú graficky - tabuľkou možných stavov (alebo aj vysielaných signálov anténami) v jednotlivých časových intervaloch jedného bloku spolu s grafickým zobrazením prechodov medzi stavmi (mriežka) podľa hodnoty vstupných bitov. Kód je možné charakterizovať aj stručne a to použitou moduláciou (počet stavov a typ), počtom stavov, počtom bitov vstupujúcich do kódera v jednom časovom intervale (bit/s/Hz). Na rozdiel od blokových kódov, počet vstupných bitov do kódera necharakterizuje použitá modulácia, ani počet stavov, ale počet možných prechodov z hociktorého stavu v časovom okamihu do ďalšieho časového okamihu. Príklad jednoduchého kódu je na obr. 7.

Otázkou je, ako určiť množiny možných stavov v jednotlivých časových intervaloch a množinu prechodov medzi stavmi. Riešenia je možné nájsť či už pokusným návrhom, alebo prehľadávaním všetkých možností distribuovanými počítačovými prostriedkami. Také vyhľadávania sú náročné na čas a množstvo použitých prostriedkov, ako aj množinu z ktorej vyberáme. Postupne vznikali viaceré algoritmy, ktoré matematickými dôkazmi zužovali počet potrebných operácií a aj veľkosť prehľadávanej množiny. Jeden z takýchto postupov je opísaný v [6], pričom v článku sú aj odkazy na ďalšie práce zaoberajúce sa vyhľadávaním vhodných mriežok.



Obr. 7: Grafický popis mriežkového časovo-priestorového kódu s moduláciou 4 PSK, 4 stavmi a 2 b/s/Hz.

Ďalšie príklady niektorých kódov, vrátane tých, ktoré sme použili pri simuláciách sú uvedené v prílohách.

#### 4.5. D-Blast

Pôvodným názvom BLAST („Bell Labs layered Space-Time Architecture“) [12], neskôr z dôvodu vzniku modifikácií bola metóda premenovaná na D-BLAST. Písmeno „D“ je odvodené od slova „diagonála“. Zdôrazňuje výstižný prvok pôvodnej metódy, a to diagonálne spracovanie vrstiev.

Všetky doteraz opisované metódy mali jednu nevýhodu. Nemožnosť dosiahnutia vysokých rýchlostí, ktoré by boli desiatky bitov za sekundu na jeden Herz. Práve potreba vysokých rýchlostí je to, čo dnešné aplikácie bezdrôtových sietí vyžadujú, vzhľadom na zväčšujúce sa potreby prenosu multimediálneho obsahu. Ich matematická výhoda v kódovaní pomocou matíc, mriežok umožňuje jednoduché dekódovanie (ortogonálne signály u STBC). Kódovanie však vnáša nadbytočnosť, čím prenosová rýchlosť klesá. Ďalej spracovávanie obrovských matíc vyžaduje rýchle obvody a procesory, avšak s ich veľkosťou náročnosť stúpa až do nereálnych hodnôt.

Popri popri predošlých metódach, ktoré tok dát spracovávali aj na strane vysieláča, boli objavované aj metódy, ktoré sa nesnažili prenášané dáta kódovať, ale skôr len nejako spracovávať na strane prijímača. Nasleduje stručný opis metódy.

Metóda vyžaduje rovnaký počet vysielacích ako aj prijímacích antén. Na strane vysieláča, sú vysielané dáta rozdelené do samostatných nezávislých kanálov. Pred tým ako sú vyslané príslušnou anténou, dochádza k pravidelnej cyklickej výmene vysielacích antén. Účelom je zabezpečiť, aby všetky prenosové kanály boli štatisticky rovnocenné. Treba pripomenúť,



Pravidlá spracovania každej časti sú takéto:

1. Rekonštruuj signály pre predošlé (už spracované) vrstvy a odčítaj ich ako rušenie od aktuálne spracovávanej časti vrstvy.
2. Predíd' rušeniu od horných (ešte nespracovaných) vrstiev sformovaním štatistických rozhodnutí, ktoré rušeniu od nich predchádzajú.

Pokiaľ pri prvej časti dôjde len k odčítaniu predošlých rozpoznaných spodných vrstiev, tak pri ďalších na ich úkor pribúda počet tých, ktoré je potrebné vynulovať pomocou štatistiky. Predchádzanie rušeniu je dosiahnuté projekciou signálových vektorov do upravenej prenosovej matice kanála. Jej tvar je možné nájsť v odporúčanej literatúre.

#### 4.6. V-BLAST

Vyššie opísaná metóda D-BLAST, ikeď prináša teoretickú efektívnosť využitia Shannonovej kapacity až na úroveň 90% [11], trpí niektorými implementačnými problémami znemožňujúcimi jej praktickú realizáciu. Preto bola metóda ďalej vyvíjaná a zjednodušená do novej metódy V-BLAST [14]. „V“ je od slova vertikálna. Túto metódu sa aj v roku 1999 podarilo autorom reálne vyskúšať v laboratórnych podmienkach (vnútorné priestory), kedy sa podarilo dosiahnuť spektrálnu efektívnosť na úrovni 40 bit/s/Hz [11].

Vysielanie je podobné pôvodnej metóde, kde sa vysielaná informácia nekóduje, ale rozdelí na toky a priamo vysielajú anténami bez dodatočného rotovania vysielacích antén. Prijímacích antén je podobne ako v predošlom prípade rovnaký počet, no môže ich byť i viacej. Prenosové prostredie a podmienky prenosu vrátane vyžiarenej energie sú bežné ako v ostatných prenosových schémach. Použitá modulácia je QAM. Podobne máme aj vysielací vektor pozostávajúci zo všetkých vyslaných symbolov v jednom momente  $\mathbf{a} = (a_1, a_2, \dots, a_m)^T$ . A podobne máme aj prijaté vektory na prijímacích anténach  $\mathbf{r}_1 = \mathbf{H}\mathbf{a} + \mathbf{v}$ .

Extrahovanie jednotlivých prijatých symbolov  $\mathbf{a}$  je vykonávané v poradí danom usporiadanou množinou  $S = \{k_1, k_2, \dots, k_M\}$ ; je to permutácia čísel  $1, 2, \dots, M$ , kde  $M$  je počet vysielacích antén. Od poradia spracovania závisí aj celkový výsledok. Z dôvodu závislosti jednotlivých medzivýsledkov, pokiaľ by hneď prvé  $\mathbf{y}_{k_l}$  (definované neskôr) malo najmenší (teda najhorší) pomer SNR, spôsobilo by ovládnutie celej chybovej štatistiky. Preto je vhodné aby kombinácia bola vybraná tak, aby minimálna SNR po detekcii bola čo najväčšia. Opis a dôkaz je v [14]. Stručne možno povedať, že signály sú extrahované od najsilnejšieho po najslabší.

Algoritmus detekcie má niekoľko krokov:

1. Použitím nulovacieho vektora  $w_{k_1}$  sformuluj rozhodovaciu štatistiku  $y_{k_1}$  :

$$(4.6.1) \quad y_{k_1} = w_{k_1} r_1$$

2. „Nakrájaj“  $y_{k_1}$  na získanie  $\hat{a}_{k_1}$  :

$$(4.6.2) \quad \hat{a}_{k_1} = Q(y_{k_1})$$

kde  $Q(\cdot)$  znamená kvantizačný operátor vhodný pre použitú sústavu.

3. Ak berieme do úvahy, že  $\hat{a}_{k_1} = a_{k_1}$ , zruš  $a_{k_1}$  z prijatého vektora  $r_1$ , pričom vznikne nový modifikovaný prijatý vektor  $r_2$  :

$$(4.6.3) \quad r_2 = r_1 - \hat{a}_{k_1} (H)_{k_1}$$

kde  $(H)_{k_1}$  znamená k-prvý stĺpec matice  $H$ . Kroky 1 až 3 sú následne vykonané

pre komponenty  $k_2, \mathbf{K}, k_M$  s použitím modifikovaných prijatých vektorov

$$r_2, r_3, \mathbf{K}, r_M .$$

Ako vidno v kroku 1, algoritmus závisí na výbere vhodných kritérií pre výpočet nulovacích vektorov  $w_{k_i}$ . Medzi najčastejšie používané kritériá patria:

1. MMSE (metóda najmenšej odchýlky) [15]
2. ZF [14]
3. Sférické dekódovanie [9]

Podrobnosti týchto kritérií sú nad rámec tejto práce, avšak čitateľ sa o nich môže dozvedieť napríklad v poukázanej odporúčanej literatúre.

## 4.7. Diferenčné Cayleyho kódy

Jednou z metód je aj metóda v ktorej informácia o kanáli nie je vôbec potrebná, teda je vhodná najmä v prípadoch kedy informácie o kanáli nie sú známe, alebo ich meranie z dôvodu častej zmeny či veľkých nákladov nie je možné. Princíp je známy už dlhšie, viaceré práce navrhli rôzne úspešné metódy, avšak nebolo známe ako nájsť kódy, fungujúce aj pri vyšších prenosových rýchlostiach, zmenách kanála a ľubovoľného počtu antén. Medzi predchodcov metódy patrí napríklad aj DPSK, kedy informácia o kanáli nie je nutná, pokiaľ kanál po nejakú dobu zostáva v približne rovnakom stave.

Stručný opis týchto kódov je možné nájsť v dokumente [8]. Okrem iného sa tu aj píše:

*„Cayleyho kódy: Sú veľmi jednoducho kódované.*

*Možné použiť ľubovoľný počet vysielačích ako aj prijímacích antén.*

*Dekódovateľné viacerými metódami vrátane techniky V-BLAST a sférického dekódovania.*

*Navrhované s ohľadom na vysielacie ako aj prijímacie antény.*

*Maximalizujú očakávanú vzdialenosť medzi párami matíc.“*

Kódy sú reprezentované jednomiestnymi štvorcovými maticami. Pri kódovaní sú vstupné dáta rozbité do samostatných dátových tokov a mapované do množiny jednomiestnych štvorcových matíc. Teda dáta tokov nie sú priamo vysielané ako je tomu u V-BLAST. Kódovanie je podľa vzorca  $X_k = X_{k-1}V_k$ . Kde  $V_k$  je Cayleyho transformácia definovaná ako

$$(4.7.1) \quad V_k = (I - iA_k)^{-1}(I + iA_k); \quad A_k = \sum_{q=1}^Q A_q a_q$$

Pričom  $A_{1..Q}$  sú vybrané Hermitove matice<sup>2</sup>, čím sa problém návrhu diferenčného kódu, hľadania množiny vhodných matíc, transformuje na problém hľadania takejto matice, aby boli zabezpečené aj potrebné kritériá zisku z kódovania. Individuálne kódové slová sú definované skalármi  $\alpha_1$  až  $\alpha_Q$ .

Dekódovanie v prijímači je diferenčné, efektívnosťou blízke ML dekódovaniu pomocou eliminácie signálov alebo sférickej metódy so skoro polynomiálnou zložitou.

Neskôr bola metóda návrhu Cayleyho kódov ďalej rozšírená [10].

#### **4.8. Sférické dekódovanie**

Problém časovo-priestorových kódov je v ich dekódovaní. Očakávame aby efektivita dekódovania bola blízka ak nie rovná efektivite ML dekódera. Komplexnosť ML dekódera je však hlavne pri nekódovanom prenose obrovská. Metódy ako STBC využívajú kódovanie na dodanie nadbytočnosti, vytvárajú navzájom ortogonálne signály, ktoré je potom ľahké presne dekódovať pomocou lineárnych dekóderov. Toto však u nekódovaných prenosov nie je možné. Sub-optimálne riešenie prináša V-BLAST metóda.

Metóda sférického dekódovania [9] ide ešte ďalej a ukazuje, že je možné dosiahnuť ML efektivitu s malou komplexnosťou a dosiahnuť plnú diverzitu MIMO systému.

#### **4.9. Iné typy ST kódov**

Časovo-priestorové kódy sú veľmi sľubnou oblasťou, ktorá by mohla priniesť pokrok v bezdrôtovom prenose. Už len vymenovanie spomenutých objavených schém tomu dáva za pravdu. Ich vývoj tým ale nekončí, existujú aj ešte novšie modifikácie a kombinácie jednotlivých schém.

---

<sup>2</sup> Informácie o týchto maticiach dostupné napríklad na internete: <http://en.wikipedia.org/wiki/Hermitian> [online]

## 5. Simulačné prostredie

Pre porovnanie naštudovaných a implementovaných kódovacích techník je potrebné vykonať simulácie. Simuluje sa prenos náhodných dát cez komunikačný kanál, pričom úspešnosť prenosu sa vyhodnocuje. Pokiaľ chceme, aby naše výsledky boli presné, výsledné grafy bez skokov a nasimulovali sme aj chybovosť  $10^{-6}$ , tak je potrebné preniesť rádovo gigabity informačných bitov. Takéto simulácie vzhľadom na náročnosť algoritmov, generovania náhodných čísiel môžu trvať aj niekoľko dní nepretržitého rábania. Preto sme sa ako časť projektu rozhodli vytvoriť program, ktorý by takéto časovo náročné simulácie dokázal zvládnuť. Najjednoduchšou cestou je pravdaže robiť simulácie v programe Matlab, avšak ten natívne neumožňuje pokračovať vo výpočtoch po reštarte počítača.

Jeden priebeh pozostáva z maximálne 50 bodov, na ktorého výpočet hodnoty sú prenášané náhodné bity komunikačným prostredím. Nie je vždy potrebné vypočítať všetkých 50 bodov, pretože vzhľadom na to, že sa simuluje  $E_b/N_0$  od menšej hodnoty po väčšiu. Ak menšia hodnota obsahuje 0 chybných bitov a takýchto je za sebou niekoľko bodov, ďalšie body rátať je už zbytočné. Ďalšie urýchlenie je možné stanovením potrebného počtu chybných bitov, po ktorých dosiahnutí máme výslednú chybovosť bodu s určitou, dopredu stanovenou presnosťou. Takto je možné simuláciu aspoň pre prvé body vodopádovej krivky podstatne urýchliť. Najprv sme toto číslo stanovili na 1000, avšak ukázalo sa, že 1000 chýb je príliš málo pre  $E_b/N_0$  do 10 dB. Preto sme simulácie museli v tomto rozsahu zopakovať s minimálnym počtom chýb/bod 100000.

Kvôli vzájomnému porovnávaní výsledkov je energia symbolu vysielacích antén prepočítavaná z počiatočnej konštantnej energie informačného bitu. Táto hodnota energie na informačný bit je jednotková. Zväčšujúci počet vysielacích antén energiu symbolu znižuje. Bity nadbytočnosti ju znižujú. Viacstavová modulácia či počet prechodov v mriežke naopak zvyšuje. A nakoniec, keďže signál v skutočnosti pri prenose má nejaký konkrétny priebeh, pričom prijímač by tento signál mal vzorkovať, tak súvis medzi vysielacou energiou antény a hodnotou úrovne signálu nie je priama úmera. Hodnotu signálu chápeme ako odmocninu z tejto pridelenej energie. Tieto pravidlá možno nájsť aj v prácach [2], [5]. V reálnom prenose by sme prijatý signál museli navzorkovať a každej vzorke priradiť nejakú diskretnú hodnotu. Práce z ktorých sa aj naša práca odvíjala s týmto nepočítajú a z dôvodu možnej kontroly s uverejnenými výsledkami sme ani my s týmito nepočítali.

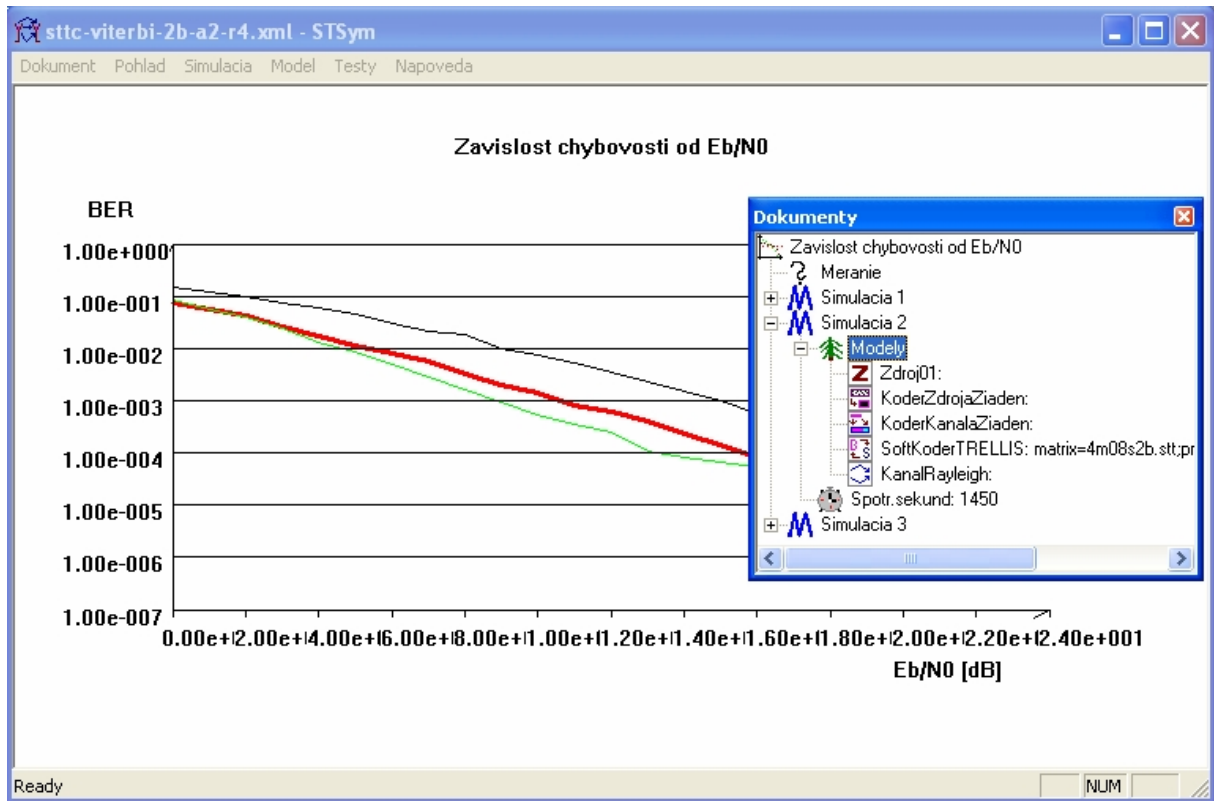
Ako prenosové prostredie sme implementovali Rayleigho kanál tak ako bol už skôr opísaný.

## 6. STSym

Schopnosti nášho programu zďaleka nekončia len pri schopnosti vykonávať časovo náročné simulácie časovo-priestorových kódov. Umožňuje vykonávať simulácie aj blokových kódov ako sú Reed-Solomon [16] kódy, prenosom cez rôzne typy kanálov ako Rayleigh, AWGN, či BSC. Je možné zvoliť si parametre použitého Reed-Solomon kódu. Takisto je možné vybrať si maticu pre STBC kód, počet bitov na symbol. Použitie Reed-Solomon a STBC kódu sa dá kombinovať. Dokonca je možné vybrať si aj typ zdroja dát a pozorovať ako sa prejavuje vplyv prenosovej cesty na hodnoty či rozdelenie prenášaných dát. Neskôr pribudol aj STTC kóder a dekóder s voľbou súboru mriežky, typu dekódera a všetkých potrebných parametrov STTC kódu vrátane dĺžky rámca.

Namerané výsledky simulácií sú v programe priamo zobrazované v grafe alebo zozname. Viaceré podobné simulácie je možné zoskupiť do dokumentu, ktorého priebehy sa v grafe zobrazia rozdielnymi farbami. Čím simulácia trvá dlhšie, tým je graf pravdaže krajší, hladší. Informácie o dokumentoch, priebehoch je možné priebežne zobrazovať už počas vykonávania simulácie. Všetky vytvorené dokumenty je možné uložiť do XML súboru.

Vykonanie simulácie v programe znamená splniť úlohu – *job*. Jednotlivé úlohy sa radia do radu úloh. Vykonávanie úloh je možné spustiť, pozastaviť, reštartovať. Priebeh vykonávania úloh sa pravidelne zobrazuje. Pri uložení dokumentov sa tieto uložia aj s úlohami, ktoré ešte neboli dokončené. Opätovné načítanie dokumentov spustí prerušenú simuláciu zo stavu v ktorom bola pozastavená. Automatické pravidelné ukladanie stavu vykonania úloh spolu s možnosťou načítania uložených dokumentov a úloh zabezpečujú, aby aj niekoľkodňová simulácia, alebo množina simulácií nevyšla nazmar - ak sa niekto pohrá s poistkami. Opisovaný program je možné vidieť na obr. 9. Jednoduchú príručku k tomuto programu je možné nájsť v prílohách.



Obr. 9: Príklad použitia programu STSym

## 7. Použité pojmy a techniky

### 7.1. Normálové – Gaussovo rozdelenie

Normálové rozdelenie patrí medzi spojité rozdelenia náhodnej premennej a je určené svojou funkciou hustoty pravdepodobnosti  $f(x)$ :

$$(7.1.1) \quad f(x) = \frac{1}{\sigma\sqrt{2\pi}} \times e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Kde  $\mu$  je stredná hodnota a  $\sigma$  smerodajná odchýlka.

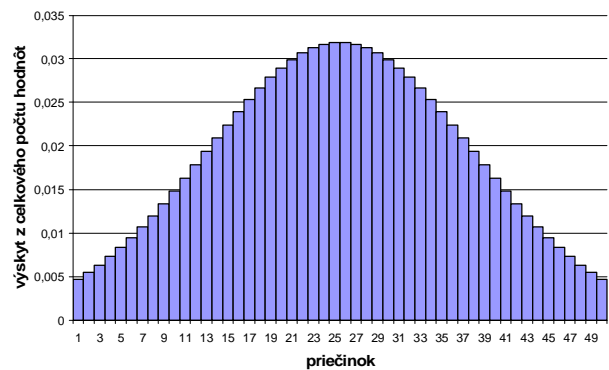
### 7.2. Generovanie náhodných čísel s normálovým rozdelením

Špeciálne rozdelenia ako opisované normálové rozdelenie nie sú štandardne podporované v programovacích jazykoch ako je napríklad C++. K dispozícii je zväčša len funkcia na generovanie náhodných hodnôt s rovnomerným rozdelením s obmedzeným počtom generovaných hodnôt. Inak je tomu v simulačných programoch ako MATLAB, ktoré však nemusia poskytovať všetky potrebné vlastnosti či funkcie.

Existujú viaceré metódy generovania náhodných čísel s normálovým rozdelením. Ich porovnanie je možné nájsť napríklad v [4]. Z porovnania vyplýva, že najlepšou metódou čo do presnosti a optimálnosti algoritmu je *polárna Box-Mullerova metóda*.

Postup metódy je nasledovný:

- a. Nech  $U_1$  a  $U_2$  sú dve nezávislé náhodné premenné s rovnomerným rozdelením od 0 po 1.
- b. Nech  $V_i = 2U_i - 1$  a definuj  $W = V_1^2 + V_2^2$
- c. Ak  $W > 1$  tak choď naspäť na krok b.
- d. Inak nech  $Y = \sqrt{(-2 \ln U_1) / W}$ ,  
 $X_1 = V_1 Y$  a  $X_2 = V_2 Y$ . Potom  $X_1$  a  $X_2$  sú navzájom nezávislé premenné s normálovým rozdelením  $N(0,1)$ .



Obr. 10: Získaný výskyt hodnôt v jednotlivých pričínkoch

Rozdelenie generovaných hodnôt s nulovou strednou hodnotou, jednotkovou varianciou v rozsahu od -2 po 2 je na obr. 10.

## 8. Distribuované simulovanie

Pre ďalšie vykonávanie simulácií, ktorých náročnosť môže spočívať vo väčšom počte, väčšej dĺžke je vhodné použiť na počítanie niekoľko počítačov súčasne – riešiť úlohy paralelne. Avšak aj samotná jedna úloha v podobe jedného priebehu v jednom grafe môže byť až príliš veľký oriešok pre jeden počítač. Budeme sa teraz ďalej baviť o rozšírení programu STSym samotného, ako aj o ďalších doplnujúcich programoch, ktoré takéto paralelné, alebo aj distribuované počítanie umožňujú.

### **V priebehu vývoja bolo potrebné vyriešiť niekoľko otázok:**

1. Architektúra, konfigurácia distribuovaného prostredia.
2. Spôsob prevodu úloh z programu STSym na jednoduchšieho menšie úlohy a naspäť.
3. Komunikačný protokol medzi komunikujúcimi.
4. Životný cyklus autonómneho simulačného programu.
5. Správa vykonávaných simulácií.
6. Dohľad nad simulačným programom, spôsob inštalácie a odinštalácie.
7. Automatická aktualizácia pri vytvorení novej verzie programu.

Jednotlivé časti celku si postupne vysvetlíme.

## 8.1. Architektúra distribuovaného simulovania

Od správneho výberu architektúry celého systému závisí počet potenciálnych simulačných staníc, efektívnosť pridelenia úloh, kontrola nad vykonávaním simulácií, rýchlosť reakcie na nové úlohy, zrušenie prebiehajúcich, ich neplatnosť a iné. Výhody a nevýhody riešení, ktoré pripadali v úvahu sú nasledovné:

Arch.	Výhody	Nevýhody
Rovný s rovným	Rýchla reakcia na zmeny. Uzol, ktorý spravuje počítanú úlohu môže rýchlo upozorniť uzol, ktorý počíta úlohu, že došlo k nejakej zmene.	Nefunkčnosť, alebo obmedzená funkčnosť komunikácie medzi sieťami, medzi ktorými je FIREWALL, prípadne len PROXY server.
	Každý uzol môže zadať do systému úlohu na rávanie.	Program, ktorý spravuje svoje úlohy v sieti musí neustále bežať.
	Celý systém je v jednom jedinom programe, napísaný v jednom programovacom jazyku.	Možné zahľtenie systému duplicitnými či nepotrebnými úlohami.
	Celistvosť zaručuje kompatibilitu komunikujúcich.	Veľká náročnosť implementácie a aj masívna zmena jestvujúceho programu.
Klient - server (vlastný protokol)	Potreba dvoch osobitných programov – jeden na počítanie, druhý na pridelenie a správu úloh – lepšia bezpečnosť systému.	Nefunkčnosť, alebo obmedzená funkčnosť komunikácie medzi sieťami, medzi ktorými je FIREWALL, prípadne len PROXY server.
	Kompaktnosť jedného programovacieho jazyka podporuje kompatibilitu komunikujúcich programov.	Vyššia náročnosť návrhu a implementácie vlastného komunikačného protokolu.
	Menej problémov s úpravou existujúceho programu STSym, avšak komunikačná klientská časť musí byť do programu implementovaná.	Program – server musí bežať NONSTOP. Reálne to môže robiť problém, nakoľko by sa nesmeli jednať o pracovnú stanicu ale server.
		Nutné priame napojenie na sieť Internet.
Klient – XML server (HTML protokol)	Systém je možno pozliepať z viacerých nezávislých programov v rôznych programovacích jazykoch. Dôležitú časť simulovanie v rýchlom C++, manažment v jednoduchom PHP.	Duplicita programovania – niektoré časti musia byť napísané a odladené viac krát; v každom použitom programovacom jazyku.
	Oddelenie počítania od správy úloh.	Nutnosť overovania kompatibility medzi jednotlivými programami.
	Funkčnosť v každej sieti v ktorej je aspoň PROXY server pre HTTP.	Výkonnosť a dostupnosť databázy úloh závisí od dostupnosti a poskytnutom výkone web servera
	Aplikácia STSym sa ponechá tak ako je, vytvoria sa len ďalšie doplnujúce programy.	Pomalá reakcia programov, ktoré vykonávajú simulácie na zmeny v správe úloh. Server nemá priame spojenie na klientov, preto len čaká, kedy sa klient opýta čo treba riešiť, prípadne či stále treba danú úlohu riešiť.
	PHP+MySQL umožňuje umiestnenie bázy úloh kdekoľvek v sieti INTERNET, kde bude k dispozícii 24 hodín denne.	

Tab. 2: Tabuľka možných architektúr distribuovaného simulovania

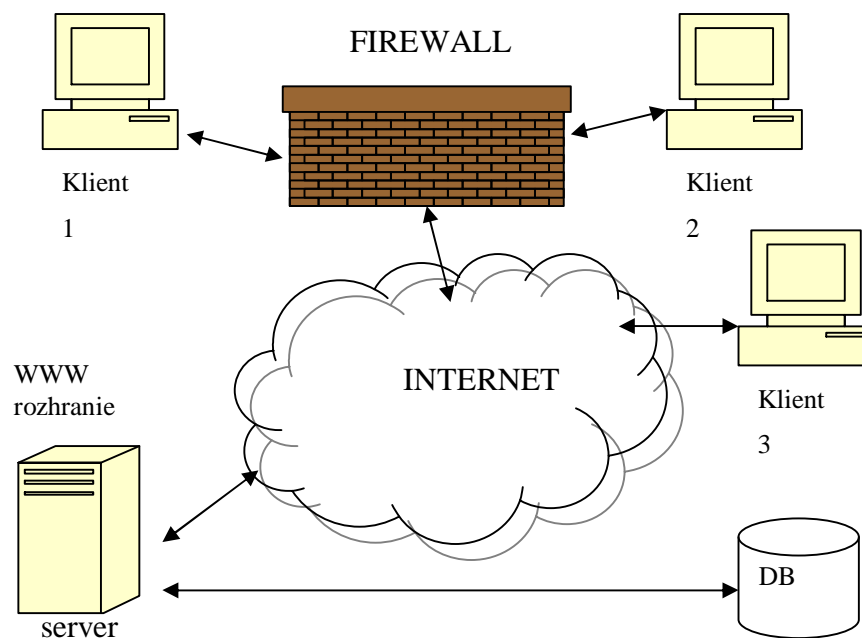
### Pričom za dôležité sme brali nasledovné kritériá:

1. Funkčnosť vo všetkých sieťach, ktoré majú prístup k Internetu. V sieťach na internátoch, školách, firmách atď.
2. NONSTOP prístupnosť k báze úloh.
3. Najvyššia rýchlosť simulačného programu.
4. Rýchlosť implementácie a odladenia programu na manažment pridelovaných úloh.

Jediné riešenie, ktoré spĺňa všetky kritériá je posledné a to Klient – XML server.

### **Princíp architektúry klient – XML server.**

Presnejšie sa jedná skôr o systém troch elementov: *Autonómny simulátor*, *PHP rozhranie* poskytujúce XML dokumenty, *MySQL server* uchovávajúci bázu úloh, a teda XML dokumentov.



**Obr. 11: Prepojenie komunikujúcich strán**

Ako na obr. 11 vyššie vidno, komunikujúce strany medzi sebou komunikujú prostredníctvom siete Internet. V tejto sieti je umiestnený WWW server, na ktorom medzi inými projektmi je aj projekt správy prebiehajúcich úloh a rozhranie pre klientské aplikácie. DB server v podaní MySQL servera môže ale nemusí byť priamo umiestnený na danom serveri. Klientské aplikácie prístupujú k rozhraniu buď priamo, alebo prostredníctvom PROXY servera, prípadne cez FIREWALL. Vďaka tomu, že rozhranie je program v jazyku

PHP, ktorý dynamicky generuje HTML alebo XML dokumenty, je možné pristúpiť k nemu z ľubovoľnej siete, pokiaľ prístup k serveru nie je explicitne zakázaný.

Komunikácia medzi klientom a rozhraním je jednoduchá a spočíva len v troch akciách: *vyžiadanie novej úlohy na spracovanie, potvrdenie pochopenia a prijatia úlohy a vrátenia výsledku spracovanej úlohy*. DB server je miesto na uloženie spracovaných ako aj nespracovaných XML dokumentov, zoznam aktuálne pridelených úloh ako aj zoznam registrovaných užívateľov a ich oprávnení. Klientská aplikácia je autonómny NT servis<sup>3</sup>, ktorý stále dokola získava a spracováva úlohy. Výpočet úloh je realizovaný pôvodnou aplikáciou STSym, ktorá bola len mierne upravená, aby mohla byť použitá v automatickom procese spracovania úloh.

## **8.2. Prevod, transformácie XML dokumentov**

Ako už bolo spomenuté, jeden XML dokument môže byť pre jeden počítač až príliš veľký oriešok na spracovanie. Preto bolo potrebné zaviesť spôsob zjednodušenia úloh, rozbitia na menšie časti. Tento proces môžeme nazvať aj ich transformáciou.

XML dokumenty z programu STSym sú najprv importované do databázy a pred označením na spracovanie sú transformáciami rozbité na menšie časti - transformácie. Navrhnutých bolo niekoľko úrovní rozbitia STSym dokumentu, nie všetky úrovne však boli aj naozaj implementované. Čím vyššia úroveň rozbitia, tým väčší počet najmenších častí na spracovanie. Zoznam týchto úrovní je v tab. 3.

Keď sú už dokumenty dostatočne atomické, zostáva ich len označiť na spracovanie. Prekopírujú sa do tabuľky spracovávaných úloh, kde čakajú na pridelenie používateľovi. Spolu s časom pridelenia a identifikátorom používateľa sa pri úlohe zaznačí aj dátum a čas do kedy je úloha užívateľovi pridelená, aby v jednom momente mohol na jednej úlohe pracovať len jeden používateľ a zároveň aby sa definoval maximálny rozumný čas do kedy sa používateľ musí ozvať že na danej úlohe stále pracuje. Keď úlohu úspešne dorieši, jej výsledok sa zapíše do tabuľky spracovávaných úloh a elementárna úloha sa tým uzavrie.

Po ukončení všetkých úloh najnižšej úrovne sa spätným transformovaním transformácie opäť spoja do jedného STSym XML dokumentu. Pritom dochádza ku kontrole, či všetky dokumenty patria k sebe a či je naozaj všetko dokončené. Štatistické údaje jednotlivých úloh sa spolu sčítajú. Tieto údaje zahŕňajú celkový spotrebovaný výpočtový čas v sekundách, počet prenesených bitov ako aj celkový počet chybné prenesených bitov.

---

<sup>3</sup> NT servis je autonómny program bežiaci v pozadí operačných systémov Windows® NT, 2000, XP

Po dokončení dokumentu je ho možné z bázy úloh exportovať a v programe STSym zobrazit' namerané priebehy jednotlivých priebehov.

Úroveň	Popis
<b>0F</b>	Najnižšia úroveň zodpovedajúca pôvodnému STSym XML dokumentu. Jeden XML dokument obsahuje jeden alebo viac <i>SymDoc</i> grafov, ktoré môžu obsahovať jeden alebo viac priebehov, atď.
<b>1D</b>	Úroveň samostatných <i>SymDoc</i> grafov. Na tejto úrovni majú transformácie len po jednom <i>SymDoc</i> grafe, tie však môžu mať ľubovoľný počet priebehov.
<b>2S</b>	Úroveň samostatných priebehov. Každý XML dokument má len jeden <i>SymDoc</i> graf, ktorý obsahuje len jeden priebeh, avšak so všetkými bodmi označenými na spracovanie.
<b>3P</b>	Úroveň samostatných pokusov, teda bodov jedného priebehu v jednom <i>SymDoc</i> grafe jedného XML dokumentu. Rozbitie dokumentov na túto úroveň je najväčším prínosom znižovania náročnosti úloh.
<b>4O</b>	Navrhovaná úroveň opakovaní prenosu bitov na výpočet jedného bodu. Implementácia by však pravdepodobne priniesla problém nepresnosti výsledkov generovania náhodných čísel pre potreby generovania chýb.

Tab. 3: Zoznam a popis úrovní rozbitia STSym dokumentu

### 8.3. Komunikačný protokol medzi komunikujúcimi

Komunikácia medzi klientom a WWW rozhraním k databáze pridelených úloh spočíva v troch požiadavkách:

1. Žiadosť o pridelenie úlohy z databázy úloh.
2. Potvrdenie pochopenia a prijatia úlohy, potvrdenie pokračovania spracovania úlohy po reštarte počítača.
3. Vrátenie výsledku simulácie do databázy úloh.

Komunikácia prebieha protokolom HTTP, pričom jeho detaily nie je vôbec potrebné poznať ani riešiť implementovanie, keďže prístup k rozhraniu zo strany klienta je riešený už implementovanou triedou systému *.Net*. Zo strany rozhrania sa jedná len o čítaním POST premenných a zapisovaním výsledku na štandardný výstup.

Ako už bolo spomenuté, všetky požiadavky sú riešené pomocou HTTP POST žiadosti na adresu rozhrania<sup>4</sup>.

Medzi zasielané premenné patria:

- **login** Prihlasovacie meno do databázy úloh.
- **passw** Prihlasovacie heslo do databázy úloh.
- **IDprocess** Identifikátor úlohy, ktorú je potrebnú potvrdiť, alebo ktorej výsledok vraciame do databázy úloh.
- **xmlko** Obsah celého výsledného XML vyriešeného dokumentu, ktorý zasielame naspäť do databázy úloh.

Výsledkom žiadosti je návratový kód, za ktorým nasleduje medzera a prípadný doplnkový alebo povinný text. Návratový kód identifikuje ako úspech tak aj neúspech operácie. Úspech operácie je identifikovaný kódom 100. Čokoľvek iné znamená neúspech.

Jednotlivé požiadavky podrobnejšie:

1. Žiadosť o pridelenie úlohy z databázy úloh.

Povinnými parametrami sú *login*, *passw*. Pri úspechu požiadavky je vrátený riadok „100 ID xxxx \n xml“, kde xxxx je identifikátor pridelenej úlohy, „\n“ znamená znak nového riadku a „xml“ obsah XML dokumentu pridelenej úlohy. Pri chybe je vrátený len návratový kód chyby.

2. Potvrdenie pochopenia a prijatia úlohy.

Povinnými parametrami sú *login*, *passw* a *IDprocess* potvrdzovanej úlohy. Pri úspechu požiadavky je vrátený len kód 100, inak nejaký iný chybový kód.

3. Vrátenie výsledku simulácie do databázy úloh.

Povinnými parametrami sú *login*, *passw*, *IDprocess* a *xmlko*. *IDprocess* je identifikátor vyriešenej úlohy a *xmlko* obsah vyriešeného XML dokumentu pridelenej úlohy.

Za chybu je považovaná aj nedostupnosť servera, ako aj nevrátenie návratového kódu, kedy klient použije štandardný chybový kód.

---

<sup>4</sup> Počas diplomovej práce bola adresa rozhrania [http://arki.host.sk/stsym\\_base/stsym.xml](http://arki.host.sk/stsym_base/stsym.xml)

## 8.4. Životný cyklus autonómneho simulátora

Zložitou časťou návrhu distribuovaného simulovania bolo navrhnúť autonómny program, ktorý beží na počítači v pozadí, získava a rieši úlohy, pričom sa vie vysporiadať so všetkými možnými problémami ktoré môžu nastať. Problém takéhoto programu je možné prirovnať k robotu, ktorého vyšleme na planétu Pluto, kde

1. Je absolútne sám odkázaný len na seba.
2. Musí udržiavať kontakt so Zemou, prijímať rozkazy a vysielat' odpovede.
3. Vedieť si poradiť s každou situáciou, nespanikáriť a ostať „na žive“.
4. Nesmie zbytočne plytvať energiou.
5. Musí poznať hranice svojich možností a poznať plán pri totálnom zlyhaní.
6. Nesmie ohroziť seba a ani planétu.
7. Po vykonaní úloh sa musí vrátiť naspäť na Zem.

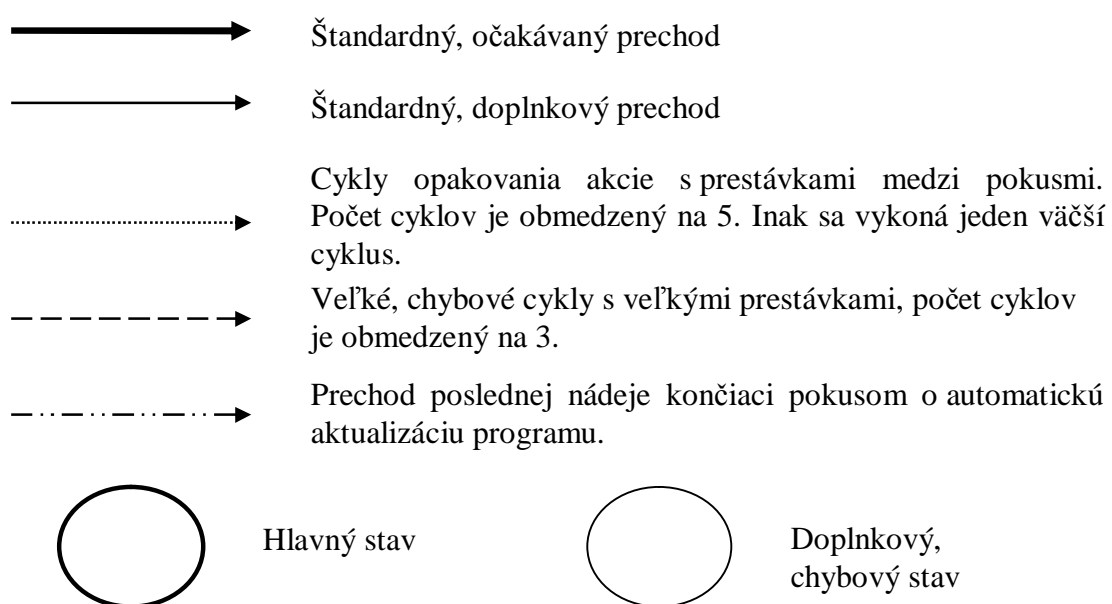
Tieto pravidlá teraz pretransformujeme do prípadu samostatného simulačného programu:

1. Po úspešnej inštalácii programu na počítači by mal byť sebestačný, nepretržite získavať a riešiť úlohy, prípadne sa aj nahradiť novšou verziou ak je to potrebné.
2. Úlohy prijíma žiadosťou servera umiestneného na WWW serveri. Výsledok tiež zasiela tomuto serveru. Udržovanie kontaktu je v podobe potvrdzovania pochopenia a prijatia úlohy na riešenie.
3. Riešenie možných problémov je vhodné riešiť implementovaním konečného automatu, ktorý by vedel riešiť bežné stavy i prípadné dopredu známe problémy, ktoré môžu nastať.
4. Pokiaľ nastane problém, ktorý si vyžaduje opakovanie požiadavky, alebo server nevie prideliť úlohu, nemôže či prípadne má iný problém, opakovanie musí byť v rozumných intervaloch a opakovaníach medzi ktorými program odpočíva.
5. Opakovania v predošlom prípade je vždy rozumné opakovať len po určitý hraničný čas po ktorom sa musí priznať že daná situácia je neriešiteľná. Môžu existovať aj stavy, priamo smerujúce k danému záveru. Avšak plán totálneho zlyhania v podobe automatickej aktualizácie rieši stavy, ktorých detekcia bola z dôvodu zlej verzie programu, nepoznania typu úlohy, zmeny parametrov či umiestnenia servera. Preto takéto stavy smerujú do automatickej aktualizácie. Servis bude po aktualizácii znovu spustený len za predpokladu, že nová verzia bola naozaj nájdená a úspešne nainštalovaná.

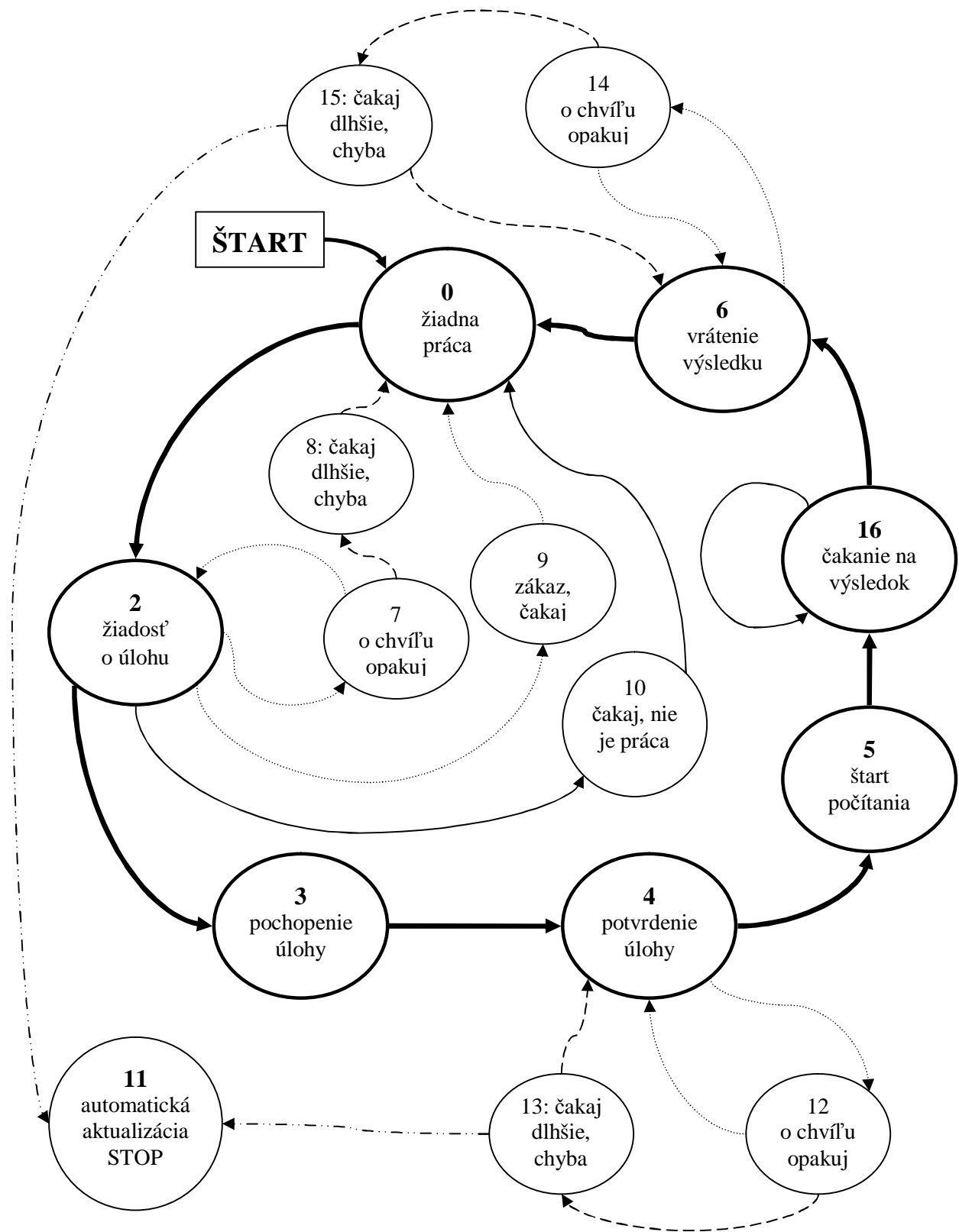
6. Neohroziť seba znamená napríklad neriešiť také úlohy, ktorým nerozumie, pretože by to mohlo spôsobiť spadnutie programu. Neohroziť planétu - neohroziť ostatné bežiacie programy, súbory na disku a hlavne výkon počítača, pokiaľ program beží na pracovnej stanici a nie na dedikovanom serveri. Preto je vhodné, aby priorita hlavného, ako aj dcérskych procesov bola najnižšia.
7. Program musí mať možnosť odinštalácie, zastavenia, opätovného spustenia, čiže kontroly. Ako aj možnosť automatickej aktualizácie.

Obr. 13 zobrazuje jednotlivé stavy servisu a prechody medzi nimi. Legenda k tomuto obrázku je na obr. 12. Detailnejší zoznam stavov s ich internými číslami a názvami je možné nájsť v prílohe D.

Stavy procesu sa delia na tri kategórie. V prvej kategórii sú stavy označené hrubou okrajovou čiarou. Tieto sú hlavné. Alebo tiež aj **žiadané**. Pokiaľ zmeny stavov postupne nastávajú z jedného hlavného stavu na druhý po hrubej čiare, všetko je v poriadku. Ďalšou kategóriou sú **upozornenia**, alebo aj krátke čakania. Nastávajú keď nejaká operácia zlyhá, pričom sa môže jednať len o dočasné zlyhanie, a preto bude operácia o niekoľko minút zopakovaná. Počet za sebou nasledujúcich takýchto krátkych čakaní je obmedzený. A poslednou kategóriou sú **závažné chyby**, kedy sa už predpokladá, že táto chyba môže byť dlhodobejšia alebo permanentná. Stavy druhej a tretej kategórie sú označené tenkou čiarou. Pokus o **automatickú aktualizáciu** je osobitným stavom.



Obr. 12: Legenda k diagramu stavov a prechodov



Obr. 13: Schéma stavov a prechodov medzi stavmi servisu

## **8.5. Databáza XML súborov úloh**

Databáza<sup>5</sup> obsahuje 4 tabuľky. Ich zoznam, popis, ako aj popis jednotlivých stĺpcov je možné nájsť v prílohe E. Obsahuje informácie o užívateľoch, ich štatistiky, zoznam oprávnení. Ďalej importované, exportované XML dokumenty, transformácie na všetkých použitých úrovniach a samotné atomické úlohy na riešenie. S každým záznamom je spojený aj dátum operácie, ktorý vedie k vzniku alebo modifikácii záznamu spolu s identifikátorom užívateľa. Nevýhodou daného návrhu tabuliek možno označiť duplicitu XML dokumentov medzi jednotlivými tabuľkami. Toto však vzhľadom na dnešné kapacity diskov nepredstavuje riziko.

## **8.6. Správa vykonávaných simulácií**

WWW rozhranie k databáze úloh<sup>6</sup> pre autonómne servery ako aj dynamické stránky pre manipuláciu úloh, vytváranie a mazanie, ako aj zobrazenie štatistiky vykonávania úloh, úspešnosť a rýchlosť jednotlivých užívateľov. Je naprogramované v jazyku PHP, ktorý sa v kombinácii s MySQL serverom využíva na internete veľmi často. Dosiahnutie výsledku je v PHP možné už s i malými skúsenosťami veľmi rýchlo. Pre prácu je potrebné sa prihlásiť prihlasovacími údajmi a od oprávnení prihláseného užívateľa závisí, ktoré operácie môže a ktoré nesmie vykonávať.

Popis sekcií a akcií, ktoré je v nich možné vykonať sa nachádza v prílohe G.

## **8.7. Dohľad nad simulátorom, inštalácia**

Oproti prvej verzii programu STSym, kedy ho tvoril len jeden spustiteľný binárny súbor a súbory matíc kódov sa rozšírením rozšíril aj počet potrebných súborov. Preto bol vytvorený aj inštalčný balík, ktorý všetky potrebné súbory nainštaluje na lokálny disk. Umiestnenie je možné nastaviť v priebehu inštalácie, má však jedno obmedzenie, a to že cesta nesmie obsahovať medzeru, ináč automatické úlohy ako aktualizácia nemusia fungovať. V priebehu inštalácie je potrebné zadať prihlasovacie meno, heslo a prípadnú zmenenú adresu www rozhrania k databáze úloh. Nezadaním posledne menovaného údaje sa použije štandardná adresa.

---

<sup>5</sup> Databáza úloh sa v čase písania práce nachádzala na serveri [db.host.sk](http://db.host.sk)

<sup>6</sup> Stránky na manažment databázy úloh sa počas diplomovej práce nachádzali na adrese [http://arki.host.sk/stsym\\_base/process.php](http://arki.host.sk/stsym_base/process.php)

Keďže autonómny servis, ktorý iniciuje a má pod kontrolou simulátor beží v pozadí pod systémovým užívateľom, pre bežného používateľa nie je možné manažovať tieto bežiace procesy. Štandardne bežia s minimálnou prioritou, ktorá by prakticky nemala vôbec obmedzovať chod ostatných aplikácií. Pokiaľ však z nejakej príčiny je potrebné zistiť, v akom stave sa nachádza simulátor, prípadne servis zastaviť alebo opätovne spustiť, je možné použiť pre tieto účely vytvorenú aplikáciu *STSymW*. Po spustení sa v notifikačnej oblasti objaví nová ikonka, cez ktorú je možný prístup k dialógu zmeny prihlasovacích nastavení, tých, ktoré boli zadané pri inštalácii. Takisto je možný prístup ku kontextovému menu na sledovanie práce servisu, jeho zastavenie ako aj opätovné spustenie.

### **8.8. Automatická aktualizácia novou verziou**

Získali sme povolenie na použitie počítačov pre vykonávanie simulácií, inštalácia prebehla úspešne, počítače úspešne odovzdávajú výsledky. Čo ale v prípade že doplníme simulačný program o nové funkcie, prípadne kódy a chceme zadať pracovným staniciam úlohu simulovať prenos pomocou tohto nového kódu? Stanice majú starú verziu programu, ktorý daný nový kód nepodporuje. Isteže je možnosť prejsť všetky stanice a na každej odinštalovať program a nainštalovať novú verziu. Je to však prácne riešenie. Preto je vhodné aj tento proces aktualizácie programu na novšiu verziu automatizovať.

Riešením je vytvoriť program, ktorý stiahne z internetu najnovšiu verziu programu (ak existuje novšia verzia), zastaví a odinštaluje aktuálnu verziu programu, nainštaluje novšiu verziu a opätovne spustí simuláciu.

Výsledkom je program *STSymU*, ktorý všetky vyššie spomínané operácie prevádza. Program je možné spustiť manuálne kedykoľvek je to potrebné. Program je spúšťaný aj automaticky servisom *STSymS*, pokiaľ servis zistí, že pridelená úloha vyžaduje novšiu verziu simulačného programu, alebo má problém komunikovať s www rozhraním či pochopiť úlohu.

## 9. Výsledky simulácie

Pre potreby simulácií sme okrem iného implementovali dve netriviálne schémy časovo-priestorových kódov – STBC a STTC. STBC kód je definovaný svojou maticou, počtom intervalov, vysielačích antén a použitou moduláciou. Hľadanie optimálnej matice pre zvolené prenosové prostredie znamená vykonať stovky, tisíce simulácií so všetkými kombináciami symbolov v matici. Toto je úloha na mesiace, roky simulovania a nie je ju možné obsiahnuť v jednej diplomovej práci. Preto sme sa sústredili na simulovanie priebehov známych kódov, ktoré boli uverejnené v odporúčanej literatúre. Názvy matíc sa môžu odlišovať. Simulovanie efektivity kódov STTC schémy prebehlo podobne. Najprv sme vykonali niekoľko rýchlych prvotných simulácií STBC kódu, ktoré objasnili rýchlosť nami implementovaného modelu. Neskôr sme tieto priebehy spresnili (dlhšie simulácie), ďalej pridali prvotné priebehy STTC kódov a na koniec vykonali viacero sérií simulácií rôznych kódov za pomoci distribúcie.

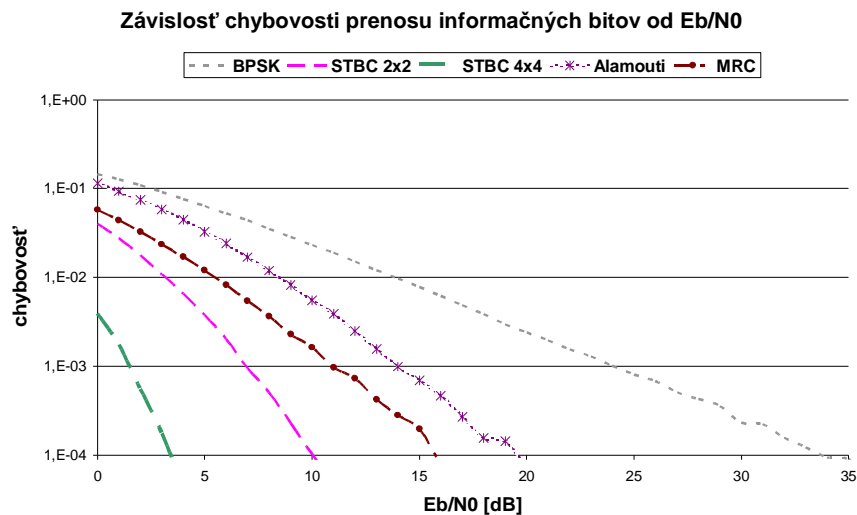
### 9.1. Prvotné simulácie STBC kódu na jednom počítači

Obr. 14 zobrazuje príklad grafu z vykonania simulácií na porovnanie efektivity blokových časovo-priestorových kódov. Ako možno na grafe vidieť, najväčšou chybovosťou sa vyznačuje BPSK prenos bez kódovania medzi 1 vysielačou a 1 prijímacou anténou. Oveľa lepšie je na tom prenos s 2 vysielačmi a 2 prijímacími anténami a maticou STBC kódu  $2 \times 2$ . Najlepší prenos, s najmenšou chybovosťou, je prenos s 8 vysielačmi a 8 prijímacími anténami, ktorý je vidno iba na obr. 16.

Naproti tomu tab. 4 ukazuje časovú náročnosť vykonaných simulácií na jednom počítači bez predčasného ukončenia simulácie. Kým prenos bez kódovania a STBC kód s maticou  $2 \times 2$  je relatívne rýchly, najlepší prenos je dosť pomalý, keďže sa pri prenose pracuje s maticou  $8 \times 8 = 64$ . Výsledky opisovanej simulácie nie sú úplne presné, pretože na výpočet jedného bodu grafu sa prenieslo len 500000 informačných bitov. Ako tab. 4 ukazuje, pokiaľ by sme chceli dosiahnuť 1000 chýb pri chybovosti  $10^{-6}$ , tak by simulácie trvali niekoľko dní či týždňov.

Simulácia	BPSK	STBC 2x2	STBC 4x4	STBC 8x8
Spolu prenesených informačných bitov	25000000	25000000	25000000	25000000
Spolu chybné prenesených informačných bitov	445817	55542	3230	25
Dĺžka simulácie v sekundách	545	452	577	1470
Rýchlosť prenosu informačných bitov [kbit/s]	45,87	55,31	43,333	17,01
Dĺžka simulácie pre dosiahnutie 1000 chýb pri chybovosti $10^{-6} = 20$ bodov x 1Gbit	5,04 dňa	4,185 dňa	5,34 dňa	2 týždne

Tab. 4: Časová náročnosť vykonaných a potrebných simulácií simulovaním na jednom počítači bez predčasného ukončenia simulácií



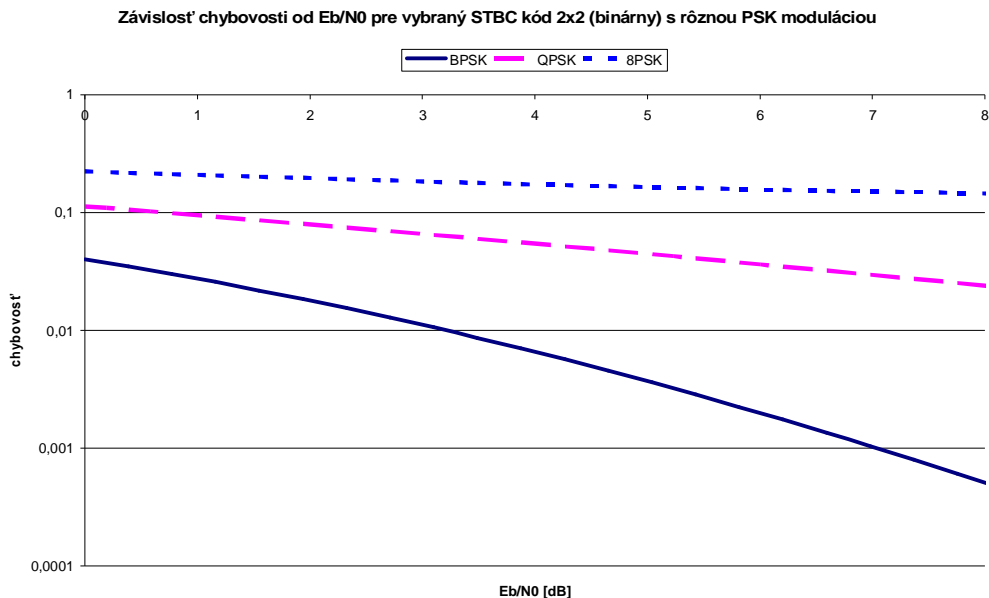
Obr. 14: Graf závislosti chybovosti prenášaných informačných bitov od  $E_b/N_0$

## 9.2. PSK modulácia a STBC matice

Doteraz prezentované simulácie STBC kódov používali iba BPSK moduláciu, pričom používané operátory nad symbolmi v matici boli len plus a mínus. Ako graf závislosti chybovosti od  $E_b/N_0$  nižšie ukazuje, efektívnosť takýchto kódov je optimálna len pre BPSK moduláciu. So zvyšujúcim sa počtom stavov PSK modulácie efektívnosť klesá, v podaní chybovosti prenášaných dát to znamená jej nárast. Na príklade STBC kódu 2x2 vidno, že pokiaľ BPSK a QPSK modulácia zabezpečuje klesanie vodopádovej krivky, ale pri 8PSK modulácii je už dekóder úplne nefunkčný.

Pre použitie modulácií QPSK, prípadne 8PSK, alebo PSK moduláciou nie mocninového tvaru je potrebné použiť maticu kódu, v ktorej kolmost' stĺpcov je zabezpečená ďalším rozmerom – imaginárnou osou. Vhodný je operátor komplexnej konjugácie alebo násobenie hodnôt násobkami základnej kruhovej frekvencie odvodené od použitej modulácie.

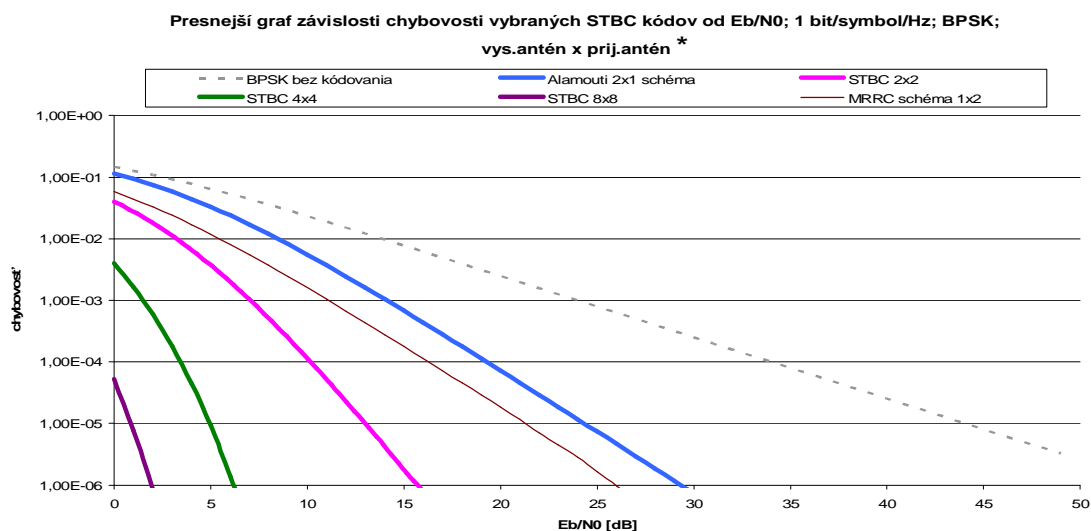
Tieto kódy nazývame komplexne ortogonálne, prípadne reálne blokové kódy.



Obr. 15: Závislosť chybovosti od  $E_b/N_0$  pre vybraný STBC kód 2x2 s rôznou PSK moduláciou

### 9.3. Presnejšie výsledky STBC

Potrebné simulácie s priebehmi o 50 bodoch a 1Gbit prenesených informačných bitov na bod, spomenuté v tabuľke vyššie, sme neskôr uskutočnili a výsledky sú zobrazené na obr. 16. Simulácie sme vykonali na piatich počítačoch s rôznym výkonom, pričom celkový spotrebovaný čas simulácii bol približne 20 dní. Oproti prvotným výpočtom jednotlivé priebehy trvali kratšie. Odvodzujúc od klesajúceho charakteru vodopádovej krivky sme implementovali predčasné ukončovanie simulácií.

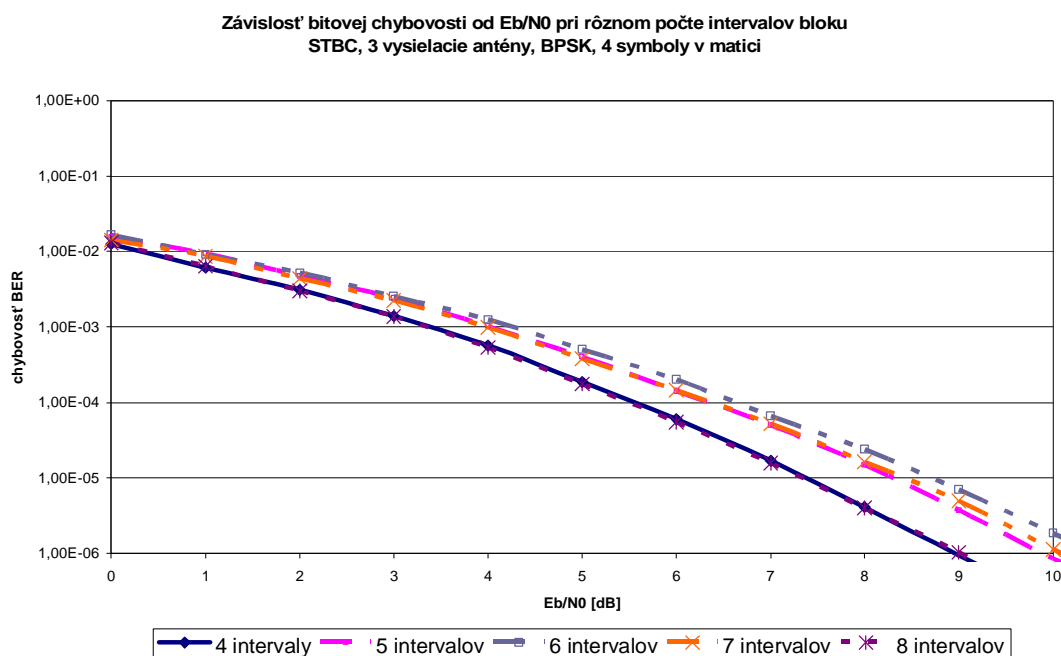


Obr. 16: Presnejší graf závislosti vybraných STBC kódov od  $E_b/N_0$

## 9.4. Počet intervalov v STBC matici

Doteraz simulované STBC matice mali štvorcový tvar, kde počet ich intervalov bol totožný s počtom vysielaných symbolov. Neskôr sme simulovali aj matice s väčším počtom intervalov ako je počet prenášaných symbolov v jednom bloku. Autori odporúčanej literatúry uvádzajú, že počet optimálnych (s úplnou diverzitou) štvorcových matíc je obmedzený, pre niektoré veľkosti existujú len sub-optimálne riešenia. Je možné ale použiť komplexné matice kódov s menšou rýchlosťou a to  $\frac{1}{2}$  alebo  $\frac{3}{4}$ . Jednoduchým spôsobom ako vytvoriť maticu rýchlosti  $\frac{1}{2}$  je zobrať napríklad maticu 4 krát 4 (obr. 25, strana 59) a pre ďalšie 4 intervaly konjugovať kópiu matice. Pri prepočte energie na jeden informačný symbol, teda tak ako sme simulovali, nedôjde k zmene vodopádovej krivky závislosti BER od  $E_b/N_0$ , avšak strácame polovicu prenosového pásma.

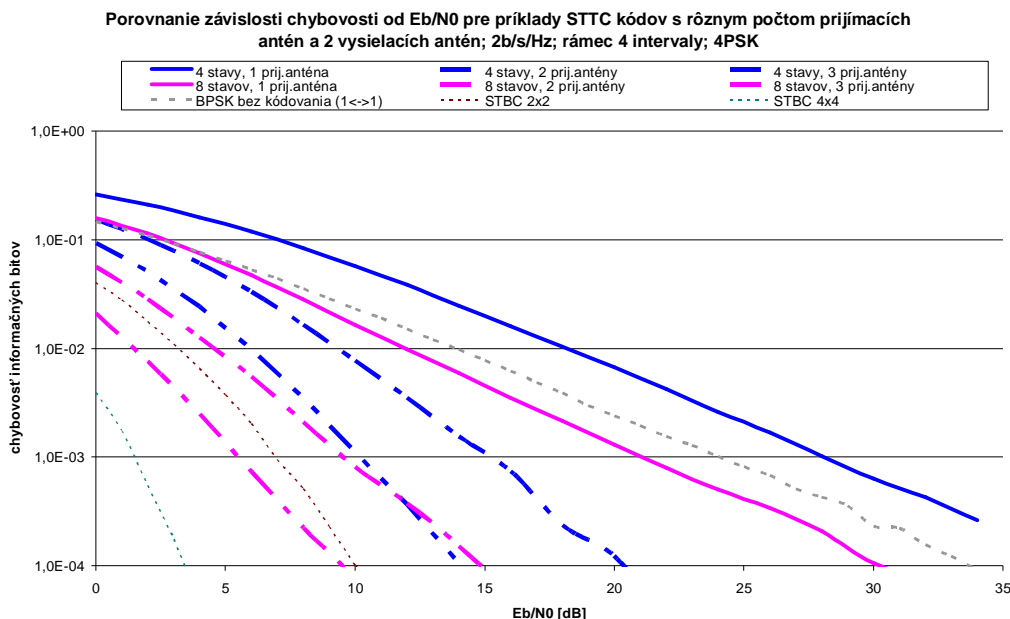
Položili sme si otázku, že keď spomínaná závislosť je rovnaká ako pre 4 tak aj pre 8 intervalov, či sa dá použiť aj menej intervalov, aby sa zväčšila energia na bit. Na obr. 17 je možné vidieť vodopádové krivky oboch kódov. Okrem nich sme simulovali aj priebehy pre počet intervalov 5, 6, 7, výberom potrebného počtu riadkov. Vidno, že takýmto jednoduchým výberom intervalov nie je možné získať rýchlejší lepší alebo rovnako dobrý kód. Pre kódy s rýchlosťou  $\frac{3}{4}$  je potrebné použiť zložitejšie STBC matice v ktorých sú použité desiatinné (aj mocniny) násobky symbolov. Náš program takéto kódy nepodporuje. Príklady kódov podľa Tarokha sú v dokumente [3].



Obr. 17: Závislosť BER od  $E_b/N_0$ , 3 antény, voliteľný počet intervalov

## 9.5. Simulovanie mriežkových časovo-priestorových kódov

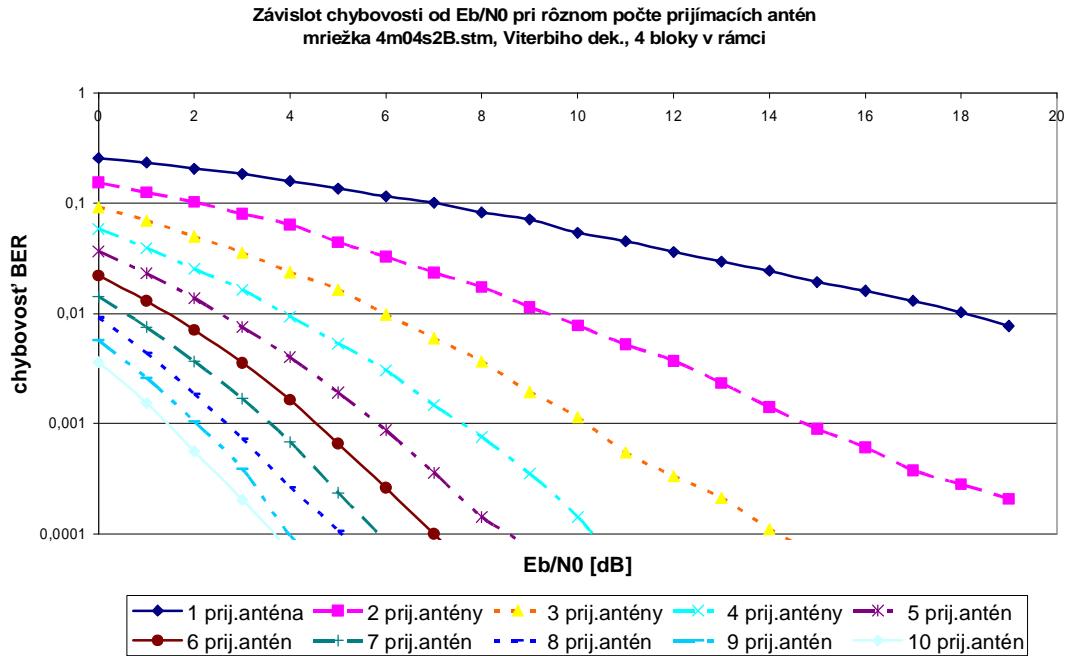
Simulácie vybraných mriežkových časovo-priestorových kódov možno vidieť na obr. 18. Použili sme dva príklady mriežok so 4 a 8 stavmi. Ďalej dve vysielačie antény, jednu, dve alebo viac prijímacích, moduláciu 4PSK, rýchlosť 2b/s/Hz a rámec so 4 intervalmi. Ako možno z grafu vidieť, so zvyšujúcim sa počtom prijímacích antén sa zisk z kódovania zvyšuje, pričom ale zároveň dodatočný zisk po pridaní ďalšej antény sa znižuje. Rozumným maximálnym počtom prijímacích antén je 6. Prvotná implementácia STTC dekódera nemala implementovaný Viterbiho dekóder, ale hľadala najpravdepodobnejšiu cestu zo všetkých možností, čím boli možnosti značne obmedzené (2b/s/Hz, 4 intervaly). Neskôr sme implementovali aj Viterbiho algoritmus, ktorý otvoril možnosti pre ďalšie, nové simulácie, väčšie rýchlosti a počty intervalov.



Obr. 18: Závislosť chybovosti vybraných mriežkových kódov

## 9.6. Zvyšovanie počtu stavov, antén, intervalov

Distribuovaným simulovaním priebehov sme zistili, že efektivitu kódu je možné vylepšovať zvyšovaním počtu použitých stavov, zvyšovaním počtu prijímacích antén (už bolo spomenuté). Zapájaním viacerých rotácií mapovania stavov STTC kódu za sebou do jedného rámca sa využije vlastnosť Viterbiho dekódera - lepšie dekódovať viacej symbolov za sebou.



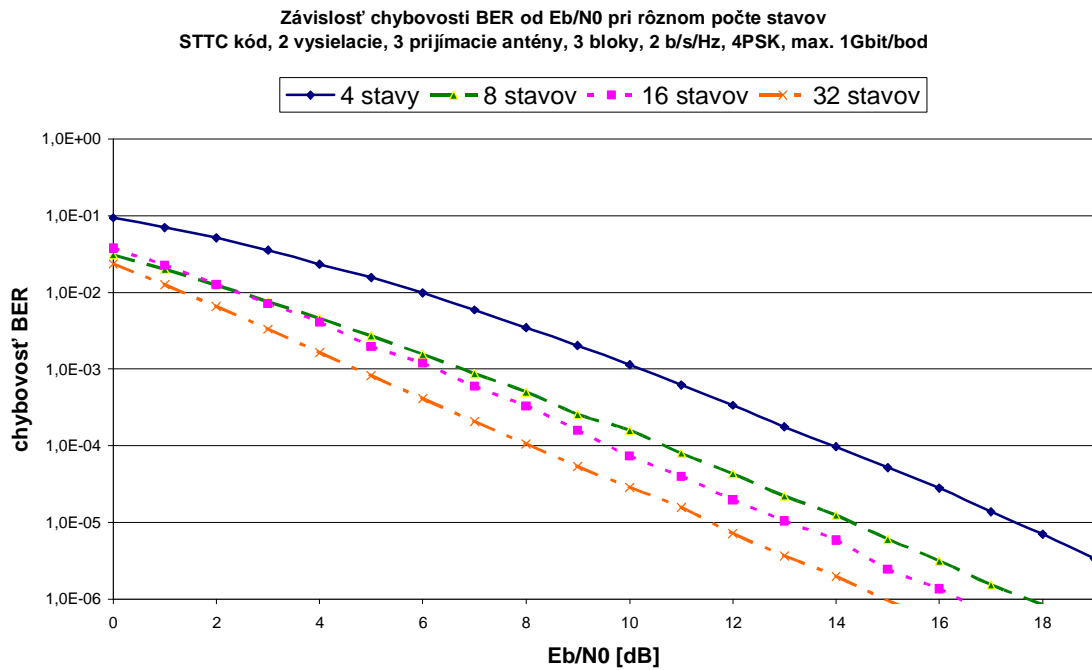
Obr. 19: Vplyv počtu prijímacích antén na efektivitu STTC kódu

Vplyv počtu prijímacích antén na priebehy efektivity STTC kódu zobrazuje obr. 19. Ako už bolo spomenuté vyššie, zvyšovanie počtu antén zlepšuje BER kódu. Spomenutý bol aj počet 6 ako rozumné maximum týchto antén. Ako vidíme, kým je počet antén menší, zisk z pridania dodatočnej antény je značný (vid' úsečka chybovosti 0,001), postupne sa znižuje a pri prechode z 9 na 10 antén je už malý. Tento prechod sa s ohľadom na iné ekonomickejšie možnosti zvýšenia efektivity už neoplatí.

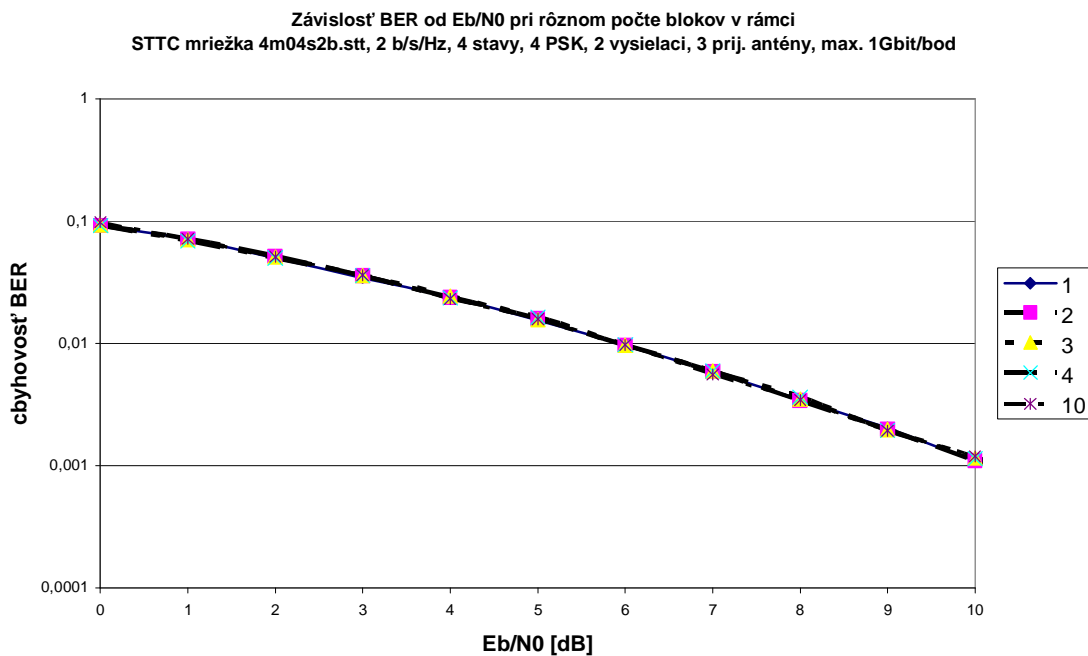
Zvýšenie počtu stavov je druhým, menej ekonomicky náročným spôsobom zvýšenia efektivity kódu. Tento prechod si môže vyžadovať zmenu použitej modulácie, pokiaľ použitá modulácia už neposkytuje požadovaný nový počet stavov. Pri STTC je možné popri PSK použiť aj QAM moduláciu. Vplyv zvyšovania počtu stavov je možné vidieť na obr. 20.

Možnosti zvýšiť efektivitu STTC kódu spojením viacerých blokov do jedného rámca už nie sú jednoznačné. O nekatastrofických konvolučných kódach, v ktorých sú použité mriežky je známe, že s narastajúcim počtom intervalov, ktoré spolu dekoduje Viterbiho dekóder sa zlepšuje aj efektivita kódu. Táto dĺžka môže narastať až do nekonečna. Simulácie STTC kódov s Viterbiho algoritmom ukázali, že veľmi závisí od konkrétnej mriežky, či spojením blokov kód zlepšíme alebo naopak zhoršíme. Pokým obr. 22 zobrazuje príklad očakávaného výsledku, kde s narastajúcim počtom blokov v rámci sa efektivita zlepšuje, obr. 21, obr. 23 ukazujú príklady kde tomu tak nie je. V prvom spomenutom príklade počet blokov

v rámci na priebeh BER nemá skoro žiaden vplyv, v druhom príklade má dokonca degradujúci efekt. Aj v kladnom príklade z obr. 22 je dodatočný zisk významný len do počtu blokov v rámci 10.

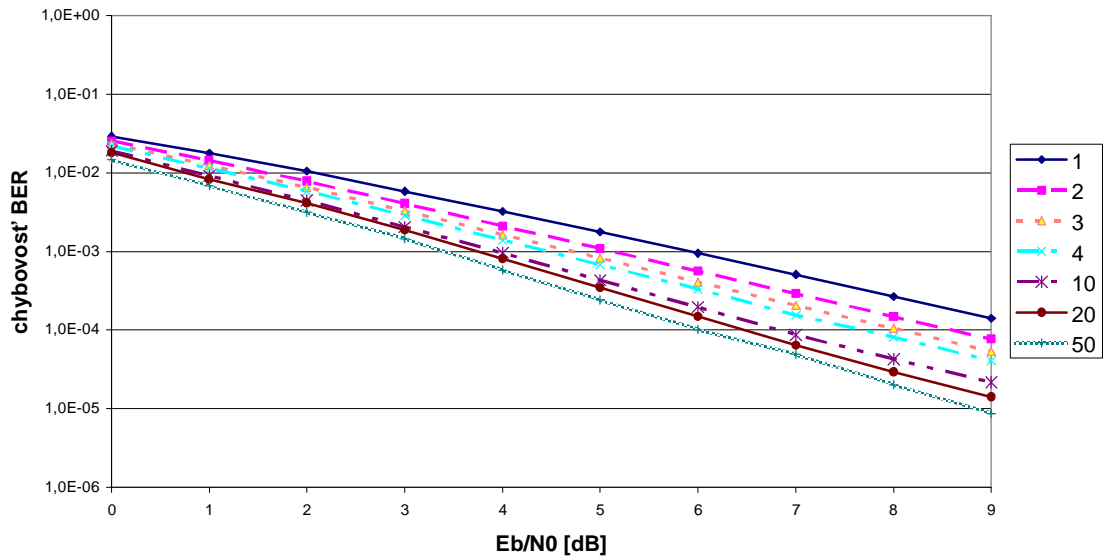


**Obr. 20: Vplyv počtu stavov STTC mriežky na BER**



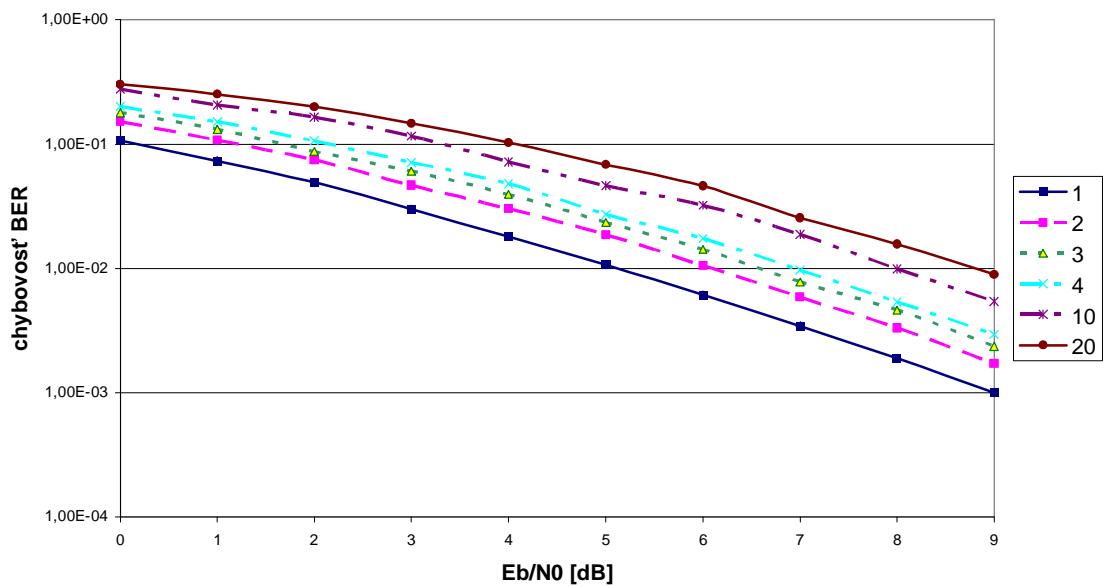
**Obr. 21: Vplyv počtu blokov v rámci na STTC kód s mriežkou 4m04s2b.stt**

Závislosť BER od  $E_b/N_0$  pri rôznom počte blokov v rámci  
 STTC mriežka 4m32s2b.stt, 2 b/s/Hz, 32 stavov, 4 PSK, 2 vysielacie,  
 3 prijimacie antény, max. 1 Gbit/bod



Obr. 22: Vplyv počtu blokov v rámci na STTC kód s mriežkou 4m32s2b.stt

Závislosť BER od  $E_b/N_0$  pri rôznom počte blokov v rámci STTC mriežka 8m16s3b.stt, 3 b/s/Hz, 16  
 stavov, 8 PSK, 2 vysielacie, 3 prijimacie antény, max. 1 Gbit/bod



Obr. 23: Vplyv počtu blokov v rámci na STTC kód s mriežkou 8m16s3b.stt

## 9.7. Viac prijímacích antén či viac blokov v rámci ?

Počet blokov v rámci	Počet prijímacích antén	Chybovosť BER
1	3	0,028912
2	3	0,025646
3	3	0,022905
4	3	0,022537
10	3	0,018099
20	3	0,017962
1	4	0,010018
2	4	0,007115
3	4	0,005691
4	4	0,005054
10	4	0,003633
1	5	0,003548
20	4	0,003305
2	5	0,002279
3	5	0,00169
4	5	0,001472
1	6	0,001364
10	5	0,000865
2	6	0,000756
1	7	0,000535
20	5	0,000534
3	6	0,000528
4	6	0,000348
2	7	0,00029
10	6	0,000208
3	7	0,000206
1	8	0,000184
20	6	0,000152
4	7	0,000151
2	8	0,000119

Tab. 5: Tabuľka chybovosti BER STTC kódu 4m32s2b.stt, pri  $E_b/N_0$  0dB, rôznych počtoch prijímacích antén a počtoch blokov v rámci

Máme známy pomer  $E_b/N_0$ , prípadne aj chybovosť BER ktorú chceme dosiahnuť. Koľko treba prijímacích antén, aký STTC kód a koľko blokov v rámci? Pre tento účel sme pokusne zmapovali priebehy závislosti chybovosti BER od pomeru  $E_b/N_0$  pre rôzne STTC mriežky. Použili sme všetky kombinácie počtu prijímacích antén (1 až 10) s počtom blokov v rámci (1, 2, 3, 4, 10, 20).

Zo vzniknutých priebehov je možné vytvoriť tabuľky podobné tab. 5 pre rôznych pomer  $E_b/N_0$ . Uvedená tabuľka zobrazuje parametre STTC kódu s mriežkou 4m32s2b.stt ak chceme dosiahnuť požadovanú chybovosť BER. Ako bolo v predošlej časti spomenuté, zlepšovanie efektivity prenosu je možné zvyšovaním počtu antén ako aj počtu blokov v rámci. Zvyšovanie počtu antén je finančne náročnejšie. Tabuľka ukazuje vzájomnú zameniteľnosť dvojíc – počet blokov a počet antén. Tabuľka je len výberom najzaujímavejšej časti z pôvodnej tabuľky. Možno si v nej všimnúť, že pri nízkom počte antén (3) žiaden počet blokov v rámci nedokáže nahradiť akúkoľvek efektívitu BER pri väčšom počte prijímacích antén s hoc i len jedným blokom v rámci.

Čím väčší je počet prijímacích antén, tým menší je náskok oproti dvojici s menším počtom antén no väčším počtom blokov. Zvýraznené riadky tabuľky tvoria hranicu, kedy väčší počet blokov v rámci nahrádza potrebu použitia dodatočnej antény. Ako vidno, počet antén v nižších riadkoch už nie je hlavným kritériom poradia efektivity BER.

Všetky priebehy je možné nájsť v dokumentoch *sttc.xls* a *simulacie.xls* na priloženom optickom dátovom médiu.



## 10. Zhrnutie

Úlohou tejto diplomovej práce bolo oboznámiť sa so známymi metódami, schémami prenosu dát pomocou viacerých vysielacích a prijímacích antén, nazývaných časovo-priestorové kódy. Ďalej vytvoriť model prenosového kanála a vytvoriť program pre nameranie a vyhodnotenie efektivity vybraných časovo-priestorových kódov. Myslíme si, že všetky body zadania sa nám splniť podarilo.

V krátkej, ani nie desať ročnej histórii časovo-priestorových kódov bolo vymyslených viacero metód. V práci je možné sa dozvedieť o princípoch najhlavnejších z nich. Ich použiteľnosť je rôzna. STBC kódy sú vďaka matematickému pozadiu ortogonálnych matic jednoducho a rýchlo dekódovateľné, avšak nájsť vhodnú maticu môže byť ťažké, pričom prenosová rýchlosť nemusí byť v porovnaní s inými metódami lákavá. STTC kód pri použití modulácie s väčším počtom stavov, intervalov dosahuje väčšie rýchlosti, avšak dekódovanie Viterbiho dekóderom je náročnejšie. Najvyššiu rýchlosť dosahujú BLAST metódy, dekódovacia náročnosť je však najvyššia. Diferenčné kódy tvoria samostatnú oblasť.

Pre simulovanie prenosu bezdrôtovým kanálom je najvhodnejšie použiť Rayleigho model prenosového kanála. Je jednoduchý a pri prvotnej implementácii novej schémy takmer nenahraditeľný, keďže aj autori metód vykonávajú simulácie pomocou tohto modelu. Je to však len teoretický model, tlmenie a šum kanála popisuje štatisticky. Je kvázistatistický. Pre lepšie simulovanie by bolo potrebné implementovať zložitejší model, zahrnúť modulovanie v určitom pásme, vzorkovať prenášaný signál, implementovať tréningové postupnosti a podobne. Pre naše simulácie tento model postačoval.

Výskum v oblasti nových spôsobov prenosu dát vyžaduje vykonávanie časovo náročných simulácií. Rýchlosť nami vytvorenej aplikácie bola vzhľadom na potrebné simulácie a výpočtové prostriedky malá, a pred nasimulovaním presných priebehov bolo potrebné popracovať na optimalizácii vykonávaných úkonov a obmedzeniach zastavenia simulácie, keď už ďalšie simulovanie nie je potrebné. Ďalším vylepšením bolo vytvorenie programov pre distribuované simulovanie, ktoré umožnilo na simulácie využiť aj iné počítače, na ktorých bol simulačný program nainštalovaný so súhlasom ich majiteľov. Simulovanie na týchto počítačoch prebiehalo samostatne a automaticky, riadené diaľkovo z rozhrania na Internete. Do aplikácie sme implementovali viaceré modely kanála (AWGN, Rayleigh, BSC), viaceré časovo-priestorové schémy (Alamouti, STTC, STBC, MRRC), RS kóder a dekóder.

Ako je možné vidieť z porovnania efektivity STBC, STTC kódov s rôznymi parametrami a maticami, prenos pomocou viacerých antén prináša dodatočné vylepšenie efektivity prenosu

bez straty rýchlosti, nutnosti zvýšenia energetickej spotreby. Je ale potrebných viacero vysieláčov a prijímačov umiestnených od seba v dostatočnej vzdialenosti. Umiestniť dve klasické antény na mobilný telefón by asi spôsobovalo problémy, zato v bázovej stanici určite nie. Možným riešením na mobilnej stanici by mohli byť anténne polia. Namerané priebehy ukázali, že chybovosť časovo-priestorových kódov je tým menšia, čím väčší počet prijímacích antén použijeme. Toto platí všeobecne pre všetky metódy. Ďalšie vylepšenia sú závislé od jednotlivých metód. Efektivita – chybovosť nami implementovaných metód bola závislá od tvaru matice či mriežky, počte intervalov a použitej modulácii. Všetky spomenuté parametre daný kód definujú. Výsledky sú rôzne od prípadu k prípadu a najst' optimálnu maticu, mriežku, moduláciu je možné len masívnym simulovaním všetkých možností po vylúčení duplicit. Toto nebolo úlohou práce, preto sme simulácie vykonali so známymi maticami, mriežkami kódov uverejnených v odporúčanej literatúre. Dobré kódy s malou chybovosťou je možné kombinovať aj s vhodným samo-opravným kódom, kedy už na úkor straty rýchlosti je možné dosiahnuť takmer nulovú chybovosť.

Na výsledky našej práce by v budúcnosti mohli nadväzovať ďalšie práce, dodatočne vyvíjať a zlepšovať simulačný program, implementovať nové algoritmy, spôsoby simulácie, prenosové kanály. Prípadne sa podujat' hľadať nové kódy za pomoci masívneho simulovania desiatkami či stovkami počítačom, hľadaním vhodnej STBC matice, STTC mriežky, modulácie.

# Prílohy



## A. Príklady výpisu súborov definovania kódov

```
symbolov_bloku=2
cas_slotov=2
vys_anten=2

+01. +02.
-02. +01.
```

**Obr. 24: Súbor matice  
STBC kódu t2a2b2.stm**

```
symbolov_bloku=4
cas_slotov=4
vys_anten=4

+01. +02. +03. +04.
-02. +01. -04. +03.
-03. +04. +01. -02.
-04. -03. +02. +01.
```

**Obr. 25: Súbor matice  
STBC kódu t4a4b4.stm**

```
symbolov_bloku=8
cas_slotov=8
vys_anten=8

+01. +02. +03. +04. +05. +06. +07. +08.
-02. +01. +04. -03. +06. -05. -08. +07.
-03. -04. +01. +02. +07. +08. -05. -06.
-04. +03. -02. +01. +08. -07. +06. -05.
-05. -06. -07. -08. +01. +02. +03. +04.
-06. +05. -08. +07. -02. +01. -04. +03.
-07. +08. +05. -06. -03. +04. +01. -02.
-08. -07. +06. +05. -04. -03. +02. +01.
```

**Obr. 26: Súbor matice STBC kódu t8a8b8.stm**

```
symbolov_bloku=4
cas_slotov=4
vys_anten=3

+01. +02. +03.
-02. +01. -04.
-03. +04. +01.
-04. -03. +02.
```

**Obr. 27: Súbor matice  
STBC kódu t4a3b4.stm**

```
symbolov_bloku=4
cas_slotov=8
vys_anten=4

+01. +02. +03. +04.
-02. +01. -04. +03.
-03. +04. +01. -02.
-04. -03. +02. +01.
+01* +02* +03* +04*
-02* +01* -04* +03*
-03* +04* +01* -02*
-04* -03* +02* +01*
```

**Obr. 28: Súbor matice  
STBC kódu t8a4b4\_c.stm**

```
symbolov_bloku=4
cas_slotov=6
vys_anten=3

+01. +02. +03.
-02. +01. -04.
-03. +04. +01.
-04. -03. +02.
+01* +02* +03*
-02* +01* -04*
```

**Obr. 29: Súbor matice STBC kódu  
t6a3s4\_c.stm**

```
psk=4
states=4
bshz=2
rotations=4
vys_anten=2

0 0,0 1,0 2,0 3
1 0,1 1,1 2,1 3
2 0,2 1,2 2,2 3
3 0,3 1,3 2,3 3

0 1 2 3
0 1 2 3
0 1 2 3
0 1 2 3
```

**Obr. 30: Súbor STTC  
kódu 4m04s2b.stt**

```
psk=4
states=8
bshz=2
rotations=4
vys_anten=2

0 0,0 1,0 2,0 3
1 0,1 1,1 2,1 3
2 0,2 1,2 2,2 3
3 0,3 1,3 2,3 3
2 2,2 3,2 0,2 1
3 2,3 3,3 0,3 1
0 2,0 3,0 0,0 1
1 2,1 3,1 0,1 1
```

```
0 1 2 3
4 5 6 7
0 1 2 3
4 5 6 7
0 1 2 3
4 5 6 7
0 1 2 3
4 5 6 7
```

**Obr. 31: Súbor STTC  
kódu 4m08s4b.stt**

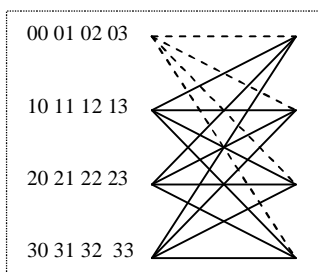
```
psk=8
states=8
bshz=3
rotations=8
vys_anten=2

0 0,0 1,0 2,0 3,0 4,0 5,0 6,0 7
5 0,5 1,5 2,5 3,5 4,5 5,5 6,5 7
2 0,2 1,2 2,2 3,2 4,2 5,2 6,2 7
7 0,7 1,7 2,7 3,7 4,7 5,7 6,7 7
4 0,4 1,4 2,4 3,4 4,4 5,4 6,4 7
1 0,1 1,1 2,1 3,1 4,1 5,1 6,1 7
6 0,6 1,6 2,6 3,6 4,6 5,6 6,6 7
3 0,3 1,3 2,3 3,3 4,3 5,3 6,3 7
```

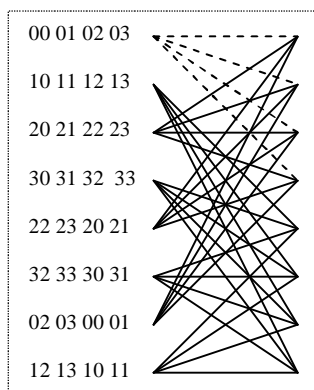
```
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
```

**Obr. 32: Súbor STTC kódu 8m08s3b.stt**

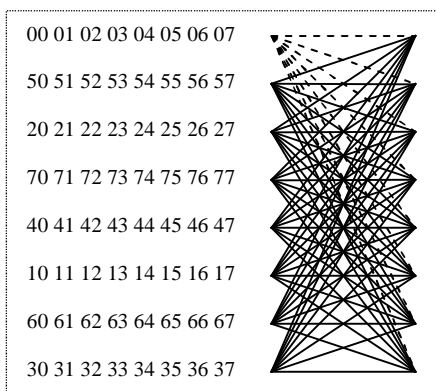
## B. Príklady grafického definovania STTC kódov



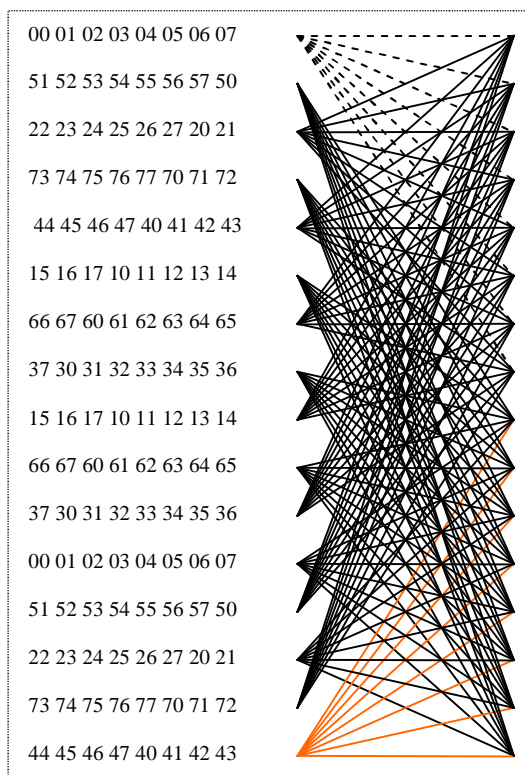
**Obr. 33: 2-ST kód, 4-PSK,  
4 stavy, 2 b/s/Hz**



**Obr. 34: 2-ST kód, 4-PSK,  
8 stavov, 2 b/s/Hz**



**Obr. 35: 2-ST kód, 8-PSK,  
8 stavov, 3 b/s/Hz**



**Obr. 36: 2-ST kód, 8-PSK,  
16 stavov, 3 b/s/Hz**

## C. Príklad XML súboru

```
<?xml version="1.0" encoding="UTF-8" ?>
- <SYMDOCS SAVE_TIME="0">
- <SYMDOC NAZOV="Zavislost" ID="ATGCTQLFCZFEYRKA">
- <MERANIE TYP="SNR" NAZOV="Zavislost">
- <X OS="LINEAR" MIN="0.000000e+000" MAX="1.900000e+001" MENO="Eb/NO
  [dB]" OPAKOVANI="100000000" HODNOT="50" />
- <Y OS="LOG" MENO="chybovost" />
  </MERANIE>
- <SIMULACIE>
- <SIMULACIA ID="UNYUTMFHOCRI ZLFW">
- <MODEL>
- <ZDROJ MENO="Zdroj01" PARAMETRE="" />
- <KODERZDROJA MENO="KoderZdrojaZiaden" PARAMETRE="" />
- <KODERKANALA MENO="KoderKanalaZiaden" PARAMETRE="" />
- <SOFTKODER MENO="BlockST" PARAMETRE="matrix=t2a2b2.stm;symbol_bits=3;" />
- <KANAL MENO="KanalRayleigh" PARAMETRE="" />
  </MODEL>
- <HODNOTY POCET="50">
- <HODNOTA INDEX="0" VALUE="2.223365e-001" />
- <HODNOTA INDEX="1" VALUE="2.165829e-001" />
- <HODNOTA INDEX="2" VALUE="2.110297e-001" />
- <HODNOTA INDEX="3" VALUE="2.056890e-001" />
- <HODNOTA INDEX="4" VALUE="2.005265e-001" />
- <HODNOTA INDEX="5" VALUE="1.955870e-001" />
- <HODNOTA INDEX="6" VALUE="1.908547e-001" />
- <HODNOTA INDEX="7" VALUE="1.863453e-001" />
- <HODNOTA INDEX="8" VALUE="1.820448e-001" />
- <HODNOTA INDEX="9" VALUE="1.779569e-001" />
- <HODNOTA INDEX="10" VALUE="1.740850e-001" />
- <HODNOTA INDEX="11" VALUE="1.704313e-001" />
- <HODNOTA INDEX="12" VALUE="1.669772e-001" />
- <HODNOTA INDEX="13" VALUE="1.637257e-001" />
- <HODNOTA INDEX="14" VALUE="1.606575e-001" />
- <HODNOTA INDEX="15" VALUE="1.577776e-001" />
- <HODNOTA INDEX="16" VALUE="1.550874e-001" />
- <HODNOTA INDEX="17" VALUE="1.525708e-001" />
- <HODNOTA INDEX="18" VALUE="1.502204e-001" />
- <HODNOTA INDEX="19" VALUE="1.480246e-001" />
- <HODNOTA INDEX="20" VALUE="1.459728e-001" />
- <HODNOTA INDEX="21" VALUE="1.440643e-001" />
- <HODNOTA INDEX="22" VALUE="1.422863e-001" />
- <HODNOTA INDEX="23" VALUE="1.406388e-001" />
- <HODNOTA INDEX="24" VALUE="0.000000e+000" />
- <HODNOTA INDEX="25" VALUE="0.000000e+000" />
- <HODNOTA INDEX="26" VALUE="0.000000e+000" />
- <HODNOTA INDEX="27" VALUE="0.000000e+000" />
- <HODNOTA INDEX="28" VALUE="0.000000e+000" />
- <HODNOTA INDEX="29" VALUE="0.000000e+000" />
- <HODNOTA INDEX="30" VALUE="0.000000e+000" />
```

```

<HODNOTA INDEX="31" VALUE="0.000000e+000" />
<HODNOTA INDEX="32" VALUE="0.000000e+000" />
<HODNOTA INDEX="33" VALUE="0.000000e+000" />
<HODNOTA INDEX="34" VALUE="0.000000e+000" />
<HODNOTA INDEX="35" VALUE="0.000000e+000" />
<HODNOTA INDEX="36" VALUE="0.000000e+000" />
<HODNOTA INDEX="37" VALUE="0.000000e+000" />
<HODNOTA INDEX="38" VALUE="0.000000e+000" />
<HODNOTA INDEX="39" VALUE="0.000000e+000" />
<HODNOTA INDEX="40" VALUE="0.000000e+000" />
<HODNOTA INDEX="41" VALUE="0.000000e+000" />
<HODNOTA INDEX="42" VALUE="0.000000e+000" />
<HODNOTA INDEX="43" VALUE="0.000000e+000" />
<HODNOTA INDEX="44" VALUE="0.000000e+000" />
<HODNOTA INDEX="45" VALUE="0.000000e+000" />
<HODNOTA INDEX="46" VALUE="0.000000e+000" />
<HODNOTA INDEX="47" VALUE="0.000000e+000" />
<HODNOTA INDEX="48" VALUE="0.000000e+000" />
<HODNOTA INDEX="49" VALUE="0.000000e+000" />
  </HODNOTY>
- <STATISTIKA STAV="BUSY">
  <PRENESENYCH>24000000048</PRENESENYCH>
  <CHYBNYCH>4161472782</CHYBNYCH>
  <SECONDS>87145</SECONDS>
  </STATISTIKA>
  <VIZUAL PENCOLOR="65280" PENSTYLE="0" PENWIDTH="1" />
  </SIMULACIA>
  </SIMULACIE>
  </SYMDOC>
- <SYMJOBS>
- <SYMJOB ID="QCXRFHEFXKCOCHXFE" STATUS="BUSY" REFRESH="0">
  <PROCESSED POKUSOV="24" OPAKOVANI="68119758"
  PRENESENYCH_BITOV="68119758" CHYBNE_PRENESENYCH_BITOV="9483094" />
  <IDENTIFICATION SYMDOCID="ATGCTQLFCZFEYRKA"
  SIMULACIAID="UNYUTMFHOCRI ZLFW" />
  </SYMJOB>
</SYMJOBS>
</SYMDOCS>

```

### D. Interné číslovanie a názvy stavov NT servisu

Číslo stavu	Názov stavu	Popis stavu
0	NO_ACTION	Počiatkový štartovací stav, v ktorom nie je definovaná žiadna akcia ani úloha.
2	XML_GET	Prebieha žiadosť o získanie úlohy na spracovanie.
3	XML_PARSE	Prebieha načítanie, pochopenie úlohy.
4	XML_ACK	Prebieha potvrdzovanie prijatej úlohy.
5	XML_CALC	Štart vykonávania úlohy.
6	XML_POST	Prebieha potvrdzovanie vykonanej úlohy.
7	XML_GET_WAIT	Krátke čakanie medzi opakovaniami pokusu o získanie úlohy na spracovanie z dôvodu nešpecifikovanej chyby.
8	XML_GET_WAIT_ERROR	Dlhšie čakanie medzi opakovaniami pokusu z dôvodu nešpecifikovanej chyby.
9	XML_GET_WAIT_UNABLE	Dlhé čakanie. Server zistil príliš veľa úloh pridelených jednému používateľovi.
10	XML_GET_WAIT_NOJOB	Dlhšie čakanie. Server nemá žiadnu úlohu na spracovanie, preto je potrebné čakať.
11	AUTO_UPDATE	Iniciovanie automatickej aktualizácie a vypnutie seba samého, teda servisu a jeho častí.
12	XML_ACK_WAIT	Krátke čakanie medzi dvoma pokusmi o potvrdenie pridelennej úlohy.
13	XML_ACK_WAIT_LONG	Dlhšie čakanie medzi pokusmi o potvrdenie prijatej úlohy z dôvodu nešpecifikovanej chyby.
14	XML_POST_WAIT	Krátke čakanie medzi pokusmi o vrátenie výsledku z dôvodu nešpecifikovanej chyby.
15	XML_POST_WAIT_LONG	Dlhšie čakanie medzi pokusmi o vrátenie výsledku z dôvodu nešpecifikovanej chyby.
16	XML_CALC_WAIT	Čakanie na ukončenie simulácie, poruchu simulátora, prípadne na iný stav.

Tab. 6: Popis stavov STSymS servisu

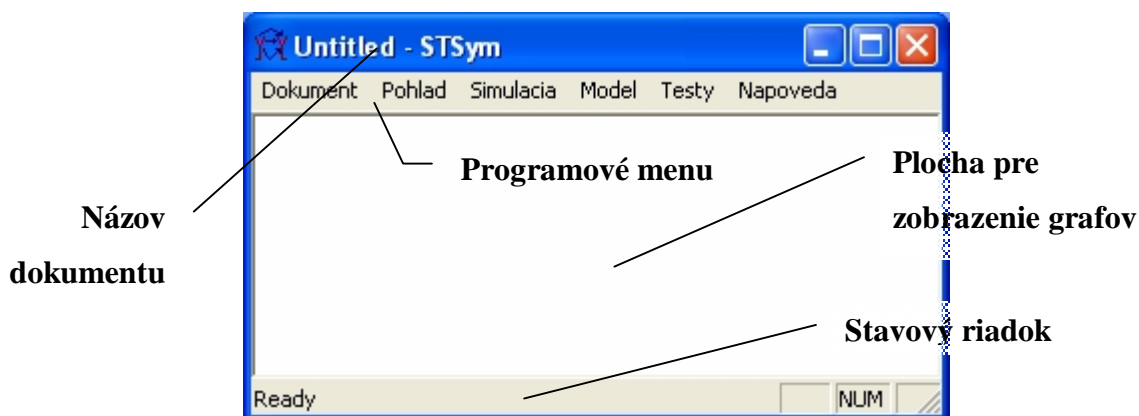
## E. Popis tabuliek a stĺpcov databázy úloh

Tabuľka	Stĺpec	Popis
users	<b>Tabuľka povolených používateľov aj s ich oprávneniami.</b>	
	<i>IDuser</i>	Identifikátor používateľa
	<i>Name</i>	Celé meno používateľa
	<i>Login</i>	Prihlasovacie meno používateľa
	<i>Passw</i>	Prihlasovacie heslo používateľa
	<i>Permissions</i>	Množina pridelených oprávnení
Xml_sydocs	<b>Primárna tabuľka vložených XML dokumentov</b>	
	<i>IDdocs</i>	Identifikátor XML dokumentu
	<i>Import_date</i>	Dátum a čas vloženia dokumentu do databázy
	<i>Import_user</i>	Identifikátor používateľa ktorý dokument vložil
	<i>Import_xml</i>	Obsah vloženého XML dokumentu
	<i>Export_date</i>	Dátum a čas kompletizácie dokumentu
	<i>Export_xml</i>	Vyriešená verzia XML dokumentu
	<i>Priority</i>	Priorita riešenia dokumentu od 1 po 5, kde 5 je najvyššia, štandardne 3
Xml_transform	<b>Tabuľka transformácií dokumentov</b>	
	<i>IDtransform</i>	Identifikátor transformácie
	<i>IDdocs</i>	Identifikátor pôvodného vloženého XML dokumentu
	<i>Xml_split</i>	Neriešený XML obsah
	<i>Xml_join</i>	Vyriešený XML obsah
	<i>IDparent</i>	Identifikátor rodiča tejto transformácie
	<i>level</i>	Úroveň danej transformácie
	<i>Children_count</i>	Počet priamych potomkov danej transformácie
Xml_process	<b>Tabuľka riešených úloh, ich pridelenia a stavu riešenia</b>	
	<i>IDprocess</i>	Identifikátor riešenej úlohy
	<i>IDdocs</i>	Identifikátor pôvodného XML dokumentu
	<i>IDtransform</i>	Identifikátor príslušnej transformácie
	<i>Xml_todo</i>	Neriešené XML
	<i>Xml_done</i>	Vyriešené XML
	<i>Assigned_user</i>	Identifikátor prideleného používateľa
	<i>Assigned_date</i>	Dátum a čas pridelenia úlohy
	<i>Completed_date</i>	Dátum a čas vyriešenia úlohy
	<i>timeout</i>	Dátum a čas vypršania pridelenia úlohy

**Tab. 7: Popis tabuliek a polí databázy úloh**

## F. Dokumentácia k aplikácii STSym

Program STSym (*STSym.exe*) je samostatným a jediným programom, ktorý vykonáva simulácie. Vytvorený bol ako prvý. Ostatné programy sú len doplnkové pre potreby distribuovaného simulovania. Na nasledujúcich riadkoch si popíšeme ovládanie tohto programu.



Obr. 37: Časti okna programu STSym

Predchádzajúci obr. 37 zobrazuje hlavné časti okna programu. Najväčšiu plochu zaberá plocha pre vykresľovanie grafov s nameranými priebehmi závislostí. V spodnej časti sa nachádza stavový riadok, ktorý informuje o aktuálne označenej položke v menu, ako aj o rýchlosti simulovania priebehu a čase, ktorý zostáva do konca vykonávanej úlohy. Vo vrchnej lište je hneď vedľa názvu programu zobrazený aj názov dokumentu XML, pod akým bol naposledy uložený na disk. Všetky operácie a voľby sa vykonávajú cez menu alebo cez dialógové okná otvorené cez menu:

- **Dokument**                      Otvorenie a uloženie súboru s priebehmi a úlohami.
- **Pohľad**                              Zmena pohľadu medzi grafom a textovým výpisom hodnôt.
  - *Výpis*                                  Prepnutie pohľadu na textový výpis hodnôt priebehov.
  - *Graf*                                      Prepnutie pohľadu na grafické zobrazenie priebehov.
  - *Okno dokumenty*                      Aktivovanie okna s popisom grafov, priebehov, štatistiky a konfigurácii použitých modelov.
  - *Okno úloh*                              Aktivovanie okna so zoznamom vykonávaných alebo v rade čakajúcich úloh, teda simulácií.
- **Simulácia**                              Akcie pre spustenie a zastavenie simulácie.
  - *Simuluj...*                              Prídanie úlohy simulovania prvého priebehu nového grafu. V dialógovom okne sa nastavujú potrebné parametre simulácie.

- *Pridaj priebeh*            Pridanie nového priebehu do aktuálneho grafu s použitím aktuálne nastavených modelov. Parametre simulácie zostávajú totožné s ostatnými priebehmi v grafe.
- *Start*                        Spustenie vykonávania v rade čakajúcich simulácií.
- *Stop*                         Zastavenie vykonávania aktívnej simulácie.
- *TODO*                        Zobrazenie počtu čakajúcich úloh z celkového počtu.
- **Model**                      Výber modelov jednotlivých častí simulovaného prostredia.
  - *Zdroj*                        Výber modelu pre generovanie informačných bitov.
  - *Kóder zdroja*                Výber modelu pre kompakciu informačných bitov pred ich kódovaním a prenosom.
  - *Kóder kanála*                Výber modelu pre kódovanie (dodanie nadbytočnosti) informačných bitov pred ich prenosom.
  - *Soft-kóder kanála*        Výber modelu časovo-priestorového kódu, ako aj použitej modulácie či počtu použitých antén na prenos.
  - *Kanál*                        Výber modelu prenosového kanála. Nie všetky modely podporujú prenos pomocou viacerých prijímacích či vysielačích antén.
- **Testy**                        Testovanie častí programu (ladenie).
- **Nápoveda**                 Informácie o autorovi a verzii programu STSym.

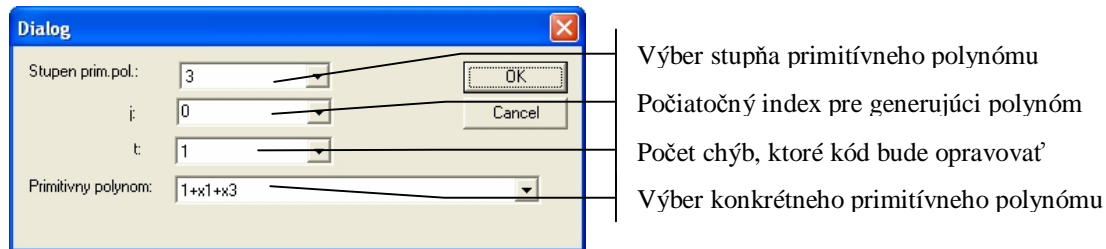
Nasleduje tab. 8 s popisom implementovaných modelov:

Model	Názov	Popis	Okno
<b>Zdroj</b>	<i>Rnd(rovnomerne)</i>	Generované sú celé čísla v celom rozsahu funkcie RND, prenášané sú bity ich binárnej reprezentácie.	
	<i>Rnd(normálovo)</i>	Generované sú celé čísla s normálovým rozdelením náhodnej premennej. Prenášané sú bity ich binárnej reprezentácie.	
	<i>Rnd 01 (rovnomerne)</i>	Generované sú jednotlivé bity s rovnomerným rozdelením, ktoré sú aj prenášané.	
<b>Kóder zdroja</b>	<i>Žiaden</i>	Kompakcia bitov sa neaplikuje.	
<b>Kóder kanála</b>	<i>Žiaden</i>	Ochrana bitov sa neaplikuje.	

	<i>Reed-Solomon</i>	Použije sa Reed-Solomonov kód.	Áno
<i>Soft-kóder kanála</i>	<i>Žiaden (BPSK)</i>	Prenos bude jedným kanálom pomocou dvojstavovej BPSK modulácie.	
	<i>Block Space-Time</i>	Použije sa STBC časovo-priestorový kód s voliteľným počtom antén.	Áno
	<i>Alamouti</i>	Alamoutiho kód s dvoma vysielačmi a jednou prijímacou anténou.	
	<i>Maximum Ratio...</i>	MRRC schéma s jednou vysielačou a dvoma prijímacími anténami.	
	<i>Trellis</i>	STBC časovo-priestorový kód s voliteľným počtom prijímacích antén. Počet vysielačích antén stanovuje použitá mriežka.	Áno
<i>Kanál</i>	<i>Prepojenie</i>	Prenášané symboly nie sú oslabené ani ovplyvnené šumom. Pri použití viacerých antén je možná len kombinácia M-M, pričom jednotlivé páry vytvárajú osobitné, nezávislé prenosové cesty.	
	<i>BSC</i>	Binárny symetrický kanál. Nie je možné použiť časovo-priestorový kód.	
	<i>AWGN</i>	Pridanie bieleho šumu k prenášaným symbolom. Signály nie sú zoslabené. Avšak nie je použiteľný pre časovo-priestorové kódy.	
	<i>Rayleigh</i>	Rayleighov prenosový kanál pre prenos cez bezdrôtové prostredie za pomoci viacerých prijímacích a vysielačích antén. Symboly sú navzájom ovplyvnené, zoslabené a pod vplyvom šumu.	

Tab. 8: Implementované modely v programe STSym

## Výber parametrov Reed-Solomon kódu



Obr. 38: Dialógové okno výberu Reed-Solomon kódu

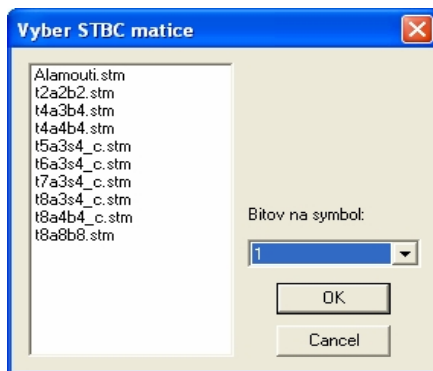
Na obr. 38 je vyobrazené dialógové okno, ukazujúce sa, keď používateľ vyberie model Reed-Solomon (ďalej len RS) kódu. RS kód podobne ako blokové kódy je definovaný svojim generujúcim polynómom, dĺžkou a nadbytočnosťou. Výber kódu prebieha v nasledovnom poradí:

1. Výber stupňa primitívneho polynómu. RS kód pracuje so symbolmi, ktorých počet je obmedzený (zvyčajne označený ako  $q$ ) a tvoria tzv. konečné Galoisove pole. Stupeň určuje, koľko informačných bitov je možné reprezentovať jedným symbolom. Zároveň určuje stupeň primitívneho polynómu, ktorý generuje binárne reprezentácie použitých symbolov.
2. Výber počiatočného indexu. Generujúci polynóm pre vytvorenie kódu závisí od počtu kódom opravovaných chybných symbolov a od počiatočného symbolu. Podrobnosti je možné nájsť v príslušnej literatúre, ktorá sa týmito kódmi zaoberá.
3. Výber počtu symbolov, ktoré RS kód je schopný opraviť, pokiaľ došlo k chybe v počte symbolov nie väčšom ako je zadané. Číslo ovplyvňuje dĺžku generujúceho polynómu, ako aj parametre kódu (myslí sa  $n, k$ ). Kde  $n=q-1$ ;  $k=n - 2 * t$
4. Pole symbolov o potrebnom počte je možné vytvoriť viacerými primitívnymi polynómami. Zo zoznamu je možné vybrať ľubovoľný z takýchto polynómov.

Potvrdenie výberu RS kódu je kliknutím na tlačidlo OK.

## Výber parametrov blokového časovo-priestorového kódu

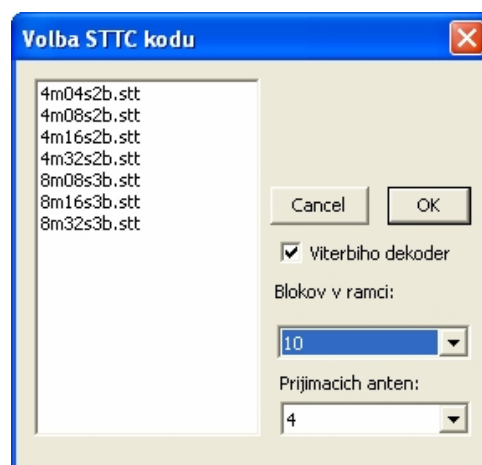
Obr. 39 zobrazuje dialógové okno pre výber konkrétneho STBC časovo-priestorového kódu. Kódy sú definované svojimi parametrami a maticou kódu, ktorá je zapísaná v súbore s príponou „.stm“ v podadresári „STBC“ adresára projektu. V súbore je definovaná matica kódu, počet intervalov, vysielacích, prijímacích antén, ako aj počet symbolov, ktoré matica prenáša. Použitá modulácia je PSK, pričom počet bitov jedného symbolu sa zadáva cez rozbaľovací zoznam „bitov na symbol“. Ak dané číslo označíme ako  $x$ , potom použitá modulácia je  $2^x$  PSK.



Obr. 39: Dialógové okno výberu STBC kódu

## Výber parametrov mriežkového časovo-priestorového kódu

Dialógové okno výberu STTC kódu je na obr. 40. Podobne ako v predošlom prípade, aj STTC kód je definovaný v súbore, ktorý je umiestnený v podadresári „STTC“ adresára projektu a má príponu „.stt“. Všetky takéto súbory v tomto adresári sú zobrazené v zozname. V súbore je definovaná matica stavov v jednotlivých časových intervaloch, matica prechodov do ďalších stavov podľa prenášaných bitov, počet intervalov, použitá PSK modulácia, počet stavov mriežky. V okne je ďalej možné určiť model dekódera. Buď ML dekóder, alebo Viterbiho dekóder. Cez rozbaľovacie zoznamy je možné potrebné vybrať počet prijímacích antén, ako aj počet blokov prenášaných za sebou a dekódovaných naraz. Jeden blok je definovaný ako jeden prechod cez všetky definované časové intervaly mriežky, pričom viaceré bloky formujú jeden rámeč.

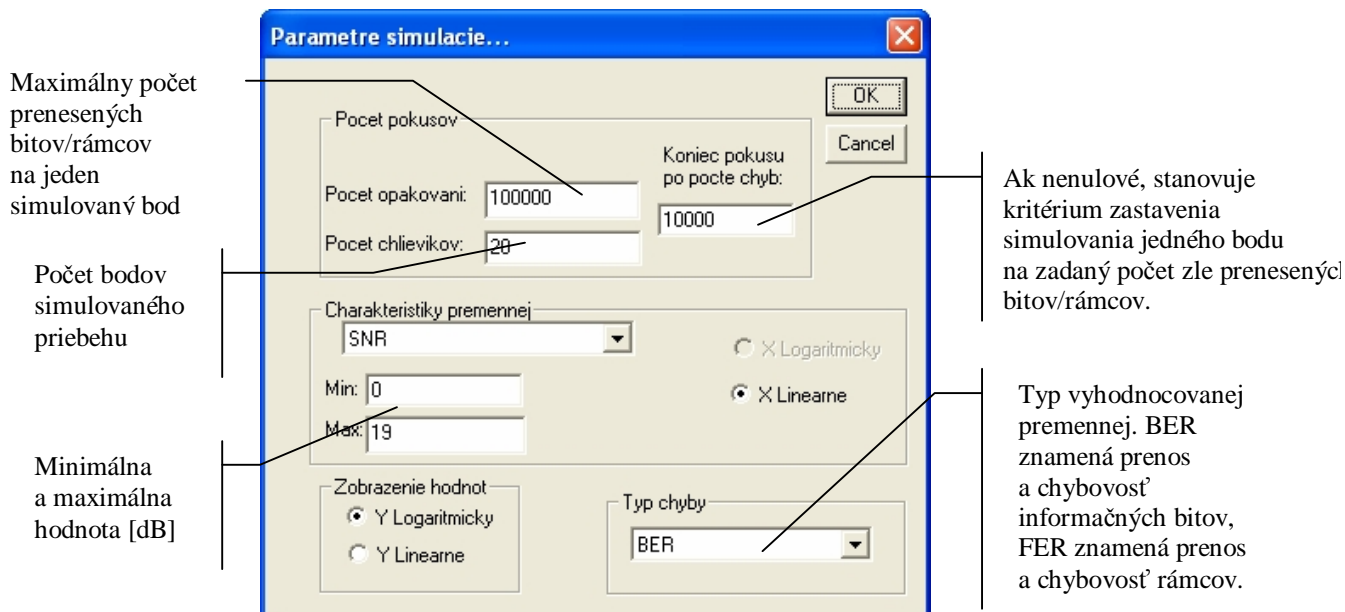


Obr. 40: Dialógové okno výberu STTC kódu a ďalších parametrov prenosu

Potvrdenie voľby je možné vykonať kliknutím na tlačidlo „OK“.

## Parametre simulácie

Po výbere potrebných modelov simulovaného prostredia simuláciu vytvoríme kliknutím na položku „Simulacia/Simuluj..“ v aplikačnom menu. Zobrazí sa okno z obr. 41.



Obr. 41: Dialógové okno voľby parametrov simulácie

Programom je možné simulovať závislosť chybovosti od  $E_b/N_0$ , ako aj iné priebehy, ktoré sú len pre účely testovania modelov a nebudeme sa nimi teraz zaoberať. Pre simulovanie spomenutého priebehu je v okne potrebné nastaviť SNR v charakteristikách premennej. Nastaviť je ďalej potrebné rozsah pomeru  $E_b/N_0$  minimálnou a maximálnou hodnotou v dB.

Zobrazenie hodnôt je vhodné prepnúť na logaritmické zobrazenie.

Typ chyby určuje, či sa bude merať chybovosť prenášaných informačných bitov (BER), alebo chybovosť celých rámcov (FER).

Presnosť, hladkosť priebehov ako aj rýchlosť vykonania simulácie ovplyvňujú parametre počtu pokusov. Tu je možné stanoviť počet bodov priebehu od minimálnej až po maximálnu hodnotu vrátane. Ďalej počet opakovaní stanovuje maximálny počet prenosov bitu alebo rámca. A napokon kritérium ukončenia simulovania jedného bodu, pokiaľ sa zistí stanovený počet chybných bitov alebo rámcov. Ak je kritérium nulové, potom sa neaplikuje.

Potvrdenie a vytvorenie nového grafu s prvým priebehom je možné kliknutím na tlačidlo „OK“. Pokiaľ chceme pridať ďalší priebeh do aktuálneho grafu (keď chcem len zmeniť výber modelov pre porovnanie), vyberieme v aplikačnom menu položku „Simulacia/Pridaj priebeh“. Novým otvorením tohto dialógového okna sa vždy vytvorí nový graf.

## **G. Rozhranie manažovania vykonávaných simulácií**

### **Sekcia „Import/Export“**

Zobrazuje sumárne informácie o vložených úlohách po celkoch. Dátum a čas vloženia úlohy do systému, linku na obsah pôvodného XML, linku na obsah dokončeného XML ak je už úloha uzatvorená. Ďalej obsahuje číselné údaje ako počet transformácií danej úlohy, celkový počet spracovávaných úloh v skupine, počet už odovzdaných výsledkov. Cez osobitný formulár sa do systému vkladá nový XML súbor, ktorý je možné potom v inej sekcii rozbiť na menšie úlohy.

Vložený súbor XML musí v systéme prejsť niekoľkými štádiami:

1. Vloženie XML súboru do systému (import)
2. Prekopírovanie XML do transformačnej tabuľky.
3. Rozbitie XML na dostatočný počet menších úloh.
4. Prekopírovanie najmenších častí dokumentu z transformačnej tabuľky do tabuľky spracovávaných úloh.
5. Vykonanie všetkých úloh.
6. Prekopírovanie výsledkov naspäť do transformačnej tabuľky.
7. Spätné spojenie výsledkov do jedného XML dokumentu.
8. Spätné prekopírovanie výsledku z transformačnej tabuľky do tabuľky dokumentov (exportovanie výsledku).
9. Vyhodnotenie výsledku.

### **Sekcia „Transformácie“**

V tejto sekcii sa vykonávajú rozbíjania úloh na množiny menších úloh, ako aj spätné spájania spracovaných množín úloh do jedného dokumentu. Stránka zobrazuje detaily o počtoch úloh v jednotlivých úrovniach. Rozbiť je možné len úlohy, ktoré ešte nemajú žiadnych potomkov. Spojenie množiny je možné ak všetci potomkovia jedného dokumentu na vyššej úrovni majú už výsledok, následne vznikne výsledok jedného rodiča o úroveň vyššie. Po spojení do jedného dokumentu je možné v predošlej sekcii vykonať export do hlavnej tabuľky dokumentov.

Pokiaľ sa zo skupiny úloh vykonala len časť, spojené jednotlivé časti (priebehy) je možné prehliadnúť ešte pred exportovaním osobitnou linkou s identifikátorom rodičovskej transformácie.

## Sekcia „Spracovanie“

Obsahuje niekoľko zoznamov štatistík vykonávania úloh.

- Pracovná štatistika: Obsahuje zoznam všetkých užívateľov a ich pracovné štatistiky. Počet pridelených ako aj už ukončených úloh, dátum a čas odovzdanie prvého výsledku, dátum a čas odovzdania posledného výsledku a najbližší dátum a čas vypršania pridelenia jednej úlohy.
- Štatistiky časov vykonávania úloh: Zoznam skupín úloh (teda importovaných dokumentov), počet najmenších úloh, percentuálny stav spracovania, minimálny, maximálny a priemerný čas medzi pridelením a odovzdaním jednej úlohy. Na konci riadku sa môže nachádzať linka „@“ na zrušenie neodovzdaných výsledkov, pokiaľ ich výpočet už nie je potrebný. V riadku sa môže nachádzať aj linka „xml“ – náhľad priebehu s už vypočítanými bodmi.
- Posledne odovzdané výsledky: Zoznam posledných 15 odovzdaných úloh spolu s menom užívateľa, času odovzdania a spotrebovaným časom od pridelenia po odovzdanie úlohy. Na konci riadku sa nachádza linka na XML súbor odovzdanej úlohy.

## Zoznam použitých skratiek

Skratka	Význam	Slovenský popis
AWGN	Additive White Gaussian Noise	Aditívny biely Gausovský šum
BLAST	Bell-labs Layered Space-time	Schéma časovo-priestorových kódov vyvinutá v laboratóriách Bell
D-BLAST	Diagonal Bell Labs Layered Space Time	Diagonálna BLAST časovo-priestorová schéma
V-BLAST	Vertical Bell Labs Layered Space Time	Vertikálna BLAST časovo-priestorová schéma
BER	Bit Error Rate	Chybovosť informačných bitov
BPSK	Binary Phase-Shift Keying	2 stavová fázová modulácia
BSC	Binary Symmetric Channel	Binárny (dvojstavový) symetrický kanál
C++		Programovací jazyk C++
Eb/N0		Pomer energie na bit k energii šumu
FER	Frame Error Rate	Chybovosť rámcov
GSM	Global System For Mobile Communications	Štandard celulárnej mobilnej siete
HTML	Hyper text Markup Language	Štandard/jazyk pre uchovávanie informácií na internete s dôrazom na logické členenie textu
http	Hyper Text Transfer Protocol	Protokol pre prenos HTML dokumentov
IID	Independent identically-distributed random variables	Nezávislé premenné s rovnomerným rozdelením náhodnej premennej
Matlab		Simulačné prostredie
MIMO	Multiple In Multiple Out	Viacero vstupov a viacero výstupov (antén)
ML	Maximum Likelihood	Princíp najväčšej pravdepodobnosti
MRRC	Maximum Receiver Ratio Combining	Princíp kombinovania prijatého signálu rovnakého zdroja z viacerých antén pre zlepšenie kvality prenášaných dát
N(0,1)	Normal(Gaussian)	Normálové rozdelenie pravdepodobnosti s nulovou strednou hodnotou a jednotkovou smerodajnou odchýlkou
PSK	Phase-Shift Keying	Fázová modulácia
Rayleigh channel model		Rayleighho model bezdrôtového kanála
RS	Reed-Solomon codes	Reed-Solomon kódy, ako špeciálny prípad BCH kódov

SNR	Signal to Noise Ratio	Pomer výkonu signálu k výkonu šumu
STBC	Space-time Block Codes	Blokové časovo-priestorové kódy
STSym	STSym	Simulačný program STSym
STSymS	STSym WIN32 NT Service	Autonómny NT servis pre distribuované simulovanie
STSymU	STSym Automatic Updater	Program pre automatickú aktualizáciu STSym
STSymW	STSym Watch	Program na ovládanie STSymS servisu
STTC	Space-time Trellis Codes	Mriežkové časovo-priestorové kódy
TCM	Trellis Coded Modulation	Mriežkovo kódovaná modulácia
XML	Extensible Markup Language	Jazyk na popis dokumentov na výmenu informácií a serializáciu objektov

## Použitá literatúra

- [1] Siavash M. Alamouti, „*A Simple Transmit Diversity Technique for Wireless Communications*”, IEEE Journal on select areas in Communications, vol. 16, no. 8, October 1998, pp. 1451-1458.
- [2] Bölcskei H., Weinrichter H., „*An overview of coding techniques for fading channels using multiple antenna elements*“, e&i vol. 11, November 2002, pp. 376-380.
- [3] Tarokh V., Jafarkhani H., Calderbank A. R., „*Space-Time Block Codes from Orthogonal Designs*“, IEEE Transactions on Information Theory., vol. 45, no. 5, July 1999, pp. 1456-1467.
- [4] Di Wu: „*Gaussian Distributed Random Variables Generation*“, Winlab, Rutgers University, K dispozícii na adrese <http://www.winlab.rutgers.edu/~diwu/Document/projectF.pdf>.  
Verzia zo dňa 23.2.2004
- [5] Tarokh V., Seshadri Nambi: „*Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction*“, IEEE Transactions on Information Theory, vol. 44, no.2, March 1998, pp. 744-753
- [6] Ananta Narayanan T., Rajan Sundan B.: „*A General Construction of Space-Time Trellis Codes for PSK signal sets*“. GLOBECOM 2003, pp. 1978 – 1983.
- [7] Zhihong Hong, Ke Liu, Akbar M. Sayeed, Health R. W. Jr.: „*A General Approach to Space-Time trellis Codes*“
- [8] B. Hassibi, B. M. Hochwald: „*Cayley Differential Unitary Space-Time Codes*“. Department of Electrical Engineering California Institute of Technology. August 2001. IEEE Transactions on Information Theory, vol. 48, No. 6, June 2002, pp. 1485 – 1503.
- [9] O. Damen, A. Chkeif, J. C. Belfiore: „*Lattice Code Decoder for Space-Time Codes*“. IEEE Communications Letters, vol. 4, No. 5, May 2000.
- [10] El Gamal, H.; Damen, M. O.: „*Universal Space-Time Coding*“. IEEE Transactions On Information Theory, vol. 49, No. 5, May 2003. pp. 1097 - 1119.
- [11] G. D. Golden, G. J. Foschini, R. A. Valenzuela a P.W. Wolnianski: „*Detection algorithm and initial laboratory results using v-blast space-time communication architecture*“, Electronics letters col. 35, pp. 14-16, January 1999.
- [12] G. J. Foschini: „*Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multi-Element Antennas*“. Bell Labs Technical Journal Autumn 1996, pp. 41-59.

- [13] R. Gowaikar, B. Hassibi: „*Efficient Statistical Pruning for Maximum Likelihood Decoding*“, California Institute of Technology, powerpoint prezentácia z dňa 3. júla 2003, verzia stiahnutá dňa 9. 10. 200 z adresy [www.ee.caltech.edu/~gowaikar/pubs/rqisit03.ppt](http://www.ee.caltech.edu/~gowaikar/pubs/rqisit03.ppt)
- [14] Wolniansky, P.W.; Foschini, G.J.; Golden, G.D.; Valenzuela, R.A.: „*V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel*“. Proc. IEEE ISSSE-98. Pisa, Taliansko, 30. september 1998. Verzia stiahnutá dňa 15.10.2004 z adresy [www.bell-labs.com/project/blast/vblast-issse-98.pdf](http://www.bell-labs.com/project/blast/vblast-issse-98.pdf) .
- [15] Golden, et al.: „*A four-element adaptive antenna array for IS-136 PCS base stations*“, IEEE VTC-97, May 1997, pp. 1577-1581.
- [16] Ďurčík Miroslav: „*Viacúčelový testovací interaktívny web pre pedagogické účely*“, záverečný projekt, FEI STU Bratislava, Katedra telekomunikácií, Máj 2003, strany 12-15, 19-21.

## Poznámky





## Obsah priloženého média

```
-----\
|      FEI STU BRATISLAVA      |
|      ODBOR: TELEKOMUNIKÁCIE  |
|                               |
|      Bc. Miroslav Ďurčík     |
|                               |
|      Simulovanie prenosu dát  |
|      prostredníctvom        |
|      priestorovo-časových kódov |
|                               |
|      DIPLOMOVÁ PRÁCA        |
|      December 2004         |
|      vedúci práce: Ing. Jozef Oboňa |
|                               |
|-----/
```

Obsah CD média:

```
-----
/source      Zdrojové súbory WIN32 programov.
/RSRP       Reed-Solomon systematický kóder a dekóder
/STSym      STSym simulátor.
/STSym_service WIN NT Autonómny servis pre distribuované
            simulovanie.
/STSymUpdater Program pre automatickú aktualizácia ostatných
            programov.
/STSymWatch Program pre kontrolu STSymS servisu.
/STSymINOUT Konzolový program pre manuálne získanie,
            potvrdenie a vrátenie vypočítanej úlohy.
/STSymInstall Projekt inštalačného balíka.
/upload     Adresár s poslednou verziou súborov automatickej
            zálohy na serveri.
/stsym_base PHP súbory rozhrania pre manažment bázy
            distribuovaných úloh.
/diplomovka Text diplomovej práce [MS WORD 2003] + MS Excel
            dokumenty s grafmi.
/db         SQL skripty na vytvorenie bázy distribuovaných úloh.
/install
  /service_pack Servisný balík .Net prostredia.
  /Release      Posledná verzia inštalačného balíka STSym
            aplikácií pre distribuované simulovanie.
/cvs       Súbory správy verzií programov.
/svoc     Prezentácia ŠVOČ 2004 o časovo-priestorových kódoch.
/kody
  /STBC       Súbory matíc STBC kódov (.stb).
  /STTC       Súbory mriežok STTC kódov (.stt).
/simulacie XML súbory zadaných úloh na simulovanie + vypočítané
            (.done.xml).
```