



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV RADIOELEKTRIKY

DEPARTMENT OF RADIO ELECTRONICS

## KOMUNIKAČNÍ PROTOKOLY PRO OPTICKÉ BEZKABELOVÉ SPOJE

COMMUNICATION PROTOCOLS FOR FREE-SPACE OPTICAL LINKS

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Ondřej Šedo

### VEDOUCÍ PRÁCE

SUPERVISOR

prof. Dr. Ing. Zdeněk Kolka

BRNO 2016

# Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

**Student:** Bc. Ondřej Šedo

**ID:** 125653

**Ročník:** 2

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

## Komunikační protokoly pro optické bezkabelové spoje

**POKyny PRO VYPRACOVÁNÍ:**

Seznamte se s charakteristikou přenosového kanálu při komunikaci pomocí laserového svazku volným prostorem. Zaměřte se především na turbulenci, jejímž projevem je náhodné kolísání úrovně přijímaného signálu. Sestavte model generující náhodný časový průběh úrovně přijímaného signálu a implementujte jej do síťového simulátoru OMNeT++. Proveďte simulaci chování protokolu TCP/IP při průchodu toto linkou.

Navrhněte techniku využití vyrovnávací paměti na straně vysílače optické linky, která pozastaví vysílání v okamžicích výpadku spojení, čímž nevznikne požadavek na opakované vysílání paketů. Metodu realizujte buď jako program pro PC nebo pomocí FPGA.

**DOPORUČENÁ LITERATURA:**

[1] PEACH, R., BURDGE, G., REITBERGER, F., VISIONE, C., OYLER, M, et al. Performance of a 10 Gbps QoS-based buffer in a FSO/RF IP network, Proc. SPIE 7814, Free-Space Laser Communications X, 781402 (August 12, 2010); doi:10.1117/12.859257.

[2] MATHIS, M., SEMKE, J., MAHDAVI, J., OTT, T., 1997. The macroscopic behavior of the TCP congestion avoidance algorithm. SIGCOMM Comput. Commun. Rev. 27, 3 (July 1997), 67-82.

**Termín zadání:** 8.2.2016

**Termín odevzdání:** 19.5.2016

**Vedoucí práce:** prof. Dr. Ing. Zdeněk Kolka

**Konzultant diplomové práce:**

**doc. Ing. Tomáš Kratochvíl, Ph.D., předseda oborové rady**

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Táto diplomová práca sa zaoberá správaním protokolu TCP/IP pri prechode bezdrôtovým optickým spojom, ktorý je ovplyvnený atmosférickými turbulenciami. Tie spôsobujú fluktuáciu prijímaného optického výkonu a tým zmenu bitovej chybovosti spoja. Navrhnutý simulačný model využíva generovanie náhodnej úrovne prijímaného optického signálu a jeho prepočet na okamžitú bitovú chybovosť. Model je implementovaný do časovej analýzy sieťového simulátoru OMNeT++. Na základe simulácií je navrhnutá vyrovnávacia pamäť vo vysieláči, ktorá pozastaví vysielanie v prípadoch výpadku linky. Metóda vyrovnávacej pamäte je realizovaná pomocou FPGA.

## **ABSTRACT**

This master's thesis is focused on behaviour of the TCP/IP protocol on free-space optical link which is affected by the effect of atmospheric turbulence. This causes fluctuation of the received optical power and therefor bit error rate changes. In the simulation model short-time bit error rate is calculated from random generated received optical signal and then used in time-domain analysis of the OMNeT++ network simulator. A buffer in transmitter is designed based on simulation results. It pauses the data transmitting in cases of FSO link outage. This method is then implemented into FPGA device.

## **KĽÚČOVÉ SLOVÁ**

Optický bezdrôtový spoj, atmosférická turbulencia, protokol TCP/IP, sieťová simulácia, FPGA, vyrovnávacia pamäť.

## **KEYWORDS**

Free space optical link, atmospheric turbulence, TCP/IP protocol, network simulation, FPGA, buffer.

ŠEDO, O. *Komunikačné protokoly pre optické bezdrôtové spoje*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2016. 24 s., 3 s. příloh. Diplomová práce. Vedúci práce: prof. Dr. Ing. Zdeněk Kolka

# PREHLÁSENIE

Prehlasujem, že svoju diplomovú prácu na tému Komunikačné protokoly pre optické bezdrôtové spoje som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomí následkov porušenia ustanovení § 11 a nasledujúcich zákona č. 121/2000 Zb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných následkov vyplývajúcich z ustanovení druhej časti, hlavy VI. diel 4 Trestného zákoníku č. 40/2009 Zb.

V Brne dňa .....

.....

(podpis autora)

# POĎAKOVANIE

Ďakujem vedúcemu diplomovej práce prof. Dr. Ing. Zdeňkovi Kolkovi za odbornú pomoc a cenné rady pri spracovávaní mojej diplomovej práce. Ďalej Ing. Michalovi Kubíčkovi, Ph.D. za odbornú pomoc pri programovaní vývojovej dosky.

V Brne dňa .....

.....

(podpis autora)

# OBSAH

|  |             |
|--|-------------|
| <b>Zoznam obrázkov</b>                                   | <b>viii</b> |
| <b>Zoznam tabuliek</b>                                   | <b>ix</b>   |
| <b>Úvod</b>  | <b>1</b>    |
| <b>1 Optické bezdrôtové spoje</b>                        | <b>2</b>    |
| 1.1 Skladba optického bezdrôtového spoja.....            | 2           |
| 1.1.1 Vysielací systém OBS .....                         | 2           |
| 1.1.2 Prijímací systém OBS .....                         | 2           |
| 1.2 Energetická bilancia optického spoja .....           | 3           |
| 1.3 Atmosférické prenosové prostredie .....              | 4           |
| <b>2 Protokol TCP/IP</b>                                 | <b>6</b>    |
| 2.1 Vrstva sieťového rozhrania.....                      | 6           |
| 2.2 Sieťová vrstva (IP).....                             | 6           |
| 2.2.1 IP protokol .....                                  | 6           |
| 2.2.2 IP datagram .....                                  | 7           |
| 2.3 Transportná vrstva (TCP) .....                       | 8           |
| 2.3.1 Protokol TCP .....                                 | 8           |
| 2.3.2 TCP segment.....                                   | 8           |
| 2.4 Aplikačná vrstva .....                               | 9           |
| <b>3 Model OBS pre simuláciu</b>                         | <b>10</b>   |
| 3.1 Návrh modelu .....                                   | 10          |
| 3.1.1 Prijatý optický výkon.....                         | 10          |
| 3.1.2 Bitová chybovosť .....                             | 11          |
| 3.1.3 Modelovanie turbulencií .....                      | 11          |
| 3.2 Implementácia do simulátoru.....                     | 13          |
| 3.2.1 Generovanie okamžitej bitovej chybovosti.....      | 13          |
| 3.2.2 Simulovanie vyrovnávacej pamäte na vysielачi ..... | 13          |
| 3.3 Simulácia .....                                      | 13          |
| <b>4 Realizácia vyrovnávacej pamäte pomocou FPGA</b>     | <b>16</b>   |

|          |  |           |
|----------|--|-----------|
| 4.1      | Návrh vyrovnávacej pamäte vo vysielачi ..... | 16        |
| 4.1.1    | Vývojová doska .....                         | 16        |
| 4.1.2    | Návrh usmernenia dátového toku .....         | 17        |
| 4.1.3    | Informácie o stave optického signálu .....   | 18        |
| 4.2      | Realizácia návrhu vo VHDL.....               | 19        |
| 4.2.1    | Obsluha sieťových rozhraní.....              | 19        |
| 4.2.2    | Stavový automat .....                        | 19        |
| 4.2.3    | Informácia o stave signálu .....             | 21        |
| <b>5</b> | <b>Záver</b>                                 | <b>22</b> |
|          | <b>Literatúra</b>                            | <b>23</b> |
|          | <b>Zoznam skratiek</b>                       | <b>24</b> |
| <b>A</b> | <b>Priebeh simulácie v OmNET++</b>           | <b>25</b> |
| <b>B</b> | <b>Zdrojový kód stavového automatu</b>       | <b>26</b> |

# ZOZNAM OBRÁZKOV

|   |    |
|---|----|
| Obrázok 1.1: miesta útlmov a zosilnení v energetickej bilancii OBS .....        | 3  |
| Obrázok 1.2: význam pomocnej dĺžky $L_0$ .....                                  | 4  |
| Obrázok 1.3: parametre fluktuácií a únikov .....                                | 5  |
| Obrázok 2.1: vrstvy TCP/IP modelu.....  | 6  |
| Obrázok 2.2: IP datagram .....  | 7  |
| Obrázok 2.3: TCP segment .....  | 9  |
| Obrázok 3.1: okamžitá hodnota stredného optického výkonu pri ON-OFF kľúčovaní.. | 11 |
| Obrázok 3.2: generátor náhodných vzoriek koeficientov $\alpha T$ .....          | 12 |
| Obrázok 3.3: generátor dvoch vzájomne korelovaných náhodných postupností .....  | 12 |
| Obrázok 3.4: výsledky simulácie (bez vyrovnávacej pamäti) .....                 | 14 |
| Obrázok 3.5: výsledky simulácie (s vyrovnávacou pamäťou).....                   | 15 |
| Obrázok 4.1: Znázornenie vyrovnávacej pamäte .....                              | 16 |
| Obrázok 4.2: Vývojová doska ML505 .....   | 17 |
| Obrázok 4.3: tok dát cez pamäť DDR2 .....                                       | 18 |
| Obrázok 4.4: generovanie IP jadra na vstupe do stavového automatu .....         | 20 |
| Obrázok 4.5: grafické znázornenie navrhnutého stavového automatu .....          | 21 |



# ZOZNAM TABULIEK

|  |   |
|--|---|
| Tabuľka 1.1: Tabuľka stavov APP na základe $V_M$ ..... | 4 |
|--|---|

# ÚVOD

Optické bezdrôtové spoje už dnes nie sú ničím výnimočným. Ich veľkou výhodou sú najmä vysoké dosahované prenosové rýchlosti. Najväčší problém však nastáva pri zlých atmosférických podmienkach, kedy môže vplyvom napríklad hmly, dažďa alebo turbulentného prúdenia dôjsť k dočasnému výpadku optického spoja. Táto práca je zameraná hlavne na fluktuácie prijatého optického výkonu spôsobené atmosférickými turbulenciami. Kolísanie prijímaného optického výkonu vyvoláva zmeny bitovej chybovosti prenosového kanálu a pri prechode protokolu TCP/IP takouto linkou dochádza k stratám (zahadzovaniu) prenášaných paketov.

Simulačný model v tejto práci generuje náhodný časový priebeh úrovne prijímaného optického signálu a následne počíta aktuálnu bitovú chybovosť. Model je implementovaný do sieťového simulátoru OMNeT++, kde prebieha simulácia sieťovej komunikácie. Na základe výsledkov simulácií je navrhnutá vyrovnávacia pamäť umiestnená vo vysielači. Tá má za úlohu pozastaviť vysielenie v prípade výpadku linky, aby sa zabránilo strate prenášaných paketov a ich následnému nutnému opakovanému vysieleniu. Táto metóda je následne implementovaná pomocou FPGA.

Prvá kapitola tejto práce je venovaná skladbe a energetickej bilancii optických bezdrôtových spojov a tiež atmosférickému prenosovému prostrediu. V druhej kapitole je stručne popísaná funkčnosť protokolu TCP/IP. Tretia kapitola obsahuje navrhovaný simulačný model a výsledky uskutočnených simulácií. Vo štvrtej kapitole je popísaná realizácia vyrovnávacej pamäte vo vysielači optickej linky pomocou FPGA. Zhrnutie a dosiahnuté výsledky práce sú uvedené v piatej kapitole.

# 1 OPTICKÉ BEZDRÔTOVÉ SPOJE

Pod pojmom optický bezdrôtový spoj (OBS) rozumieme plne duplexný spoj, ktorý k prenosu informácie v atmosférickom prenosovom prostredí (APP) využíva optickú nosnú vlnu obsahujúcu jeden alebo viac vlnovo delených kanálov. Optický výkon týchto kanálov je sústredený do jedného alebo viacerých zväzkov. Táto kapitola sa zaoberá skladbou a energetickou bilanciou takýchto spojov. Taktiež sú tu uvedené základné vlastnosti atmosférického prenosového prostredia a jeho vplyv na prenášaný signál.

## 1.1 Skladba optického bezdrôtového spoja

OBS sa skladá z dvoch hlavíc komunikujúcich medzi sebou duplexným spôsobom. Každá hlavica je ďalej pripojená k osobnému počítaču, serveru alebo ústredni väčšinou prostredníctvom optického alebo metalického káblu. Obe hlavice sú vybavené vysielacím a prijímacím systémom ako pre komunikáciu medzi sebou, tak aj pre komunikáciu s najbližším sieťovým počítačom. Väčšina OBS je navrhnutých nezávisle na použitom protokole a využíva ON-OFF kľúčovanie (OOK) [2].

### 1.1.1 Vysielací systém OBS

Vysielacím systémom OBS je časť hlavice, ktorá tvaruje vyžarovaný zväzok a zabezpečuje jeho moduláciu. Za jeho hlavné bloky sa považujú: modulátor, budič optického zdroja, laserová dióda, zameriavací systém a vysielacia optická sústava.

Priestorové tvarovanie optického zväzku vyžarovaného laserovou diódou zabezpečuje vysielacia osovo symetrická optická sústava. Tvarovaný zväzok ďalej prechádza optickým priezorom, ktorý slúži ako ochrana pred nečistotami prítomnými v atmosfére. Optický priezor pritom nesmie vyvolať nadmerný útlm alebo deformáciu optického zväzku. Hrubé nastavenie smerovania vysielacej sústavy sa vykonáva pomocou ďalekohľadu pripevneného k hlavici. O jemné smerovanie optickej osi sa stará mechanicky alebo elektronicky ovládaný zameriavací systém. K prúdovej stabilizácii optického výkonu slúži snímacia fotodióda zabudovaná v puzdre laserovej diódy.

OBS nevyžaduje kruhovú symetrickosť zväzku a pri energetickej bilancii sa pôvodný optický zväzok nahrádza energeticky ekvivalentným Gaussovým zväzkom. Ten má na ose zväzku rovnakú optickú intenzitu ako pôvodný zväzok a v jeho kruhovej stope je obsiahnutý rovnaký výkon ako v eliptickej stope pôvodného zväzku.

### 1.1.2 Prijímací systém OBS

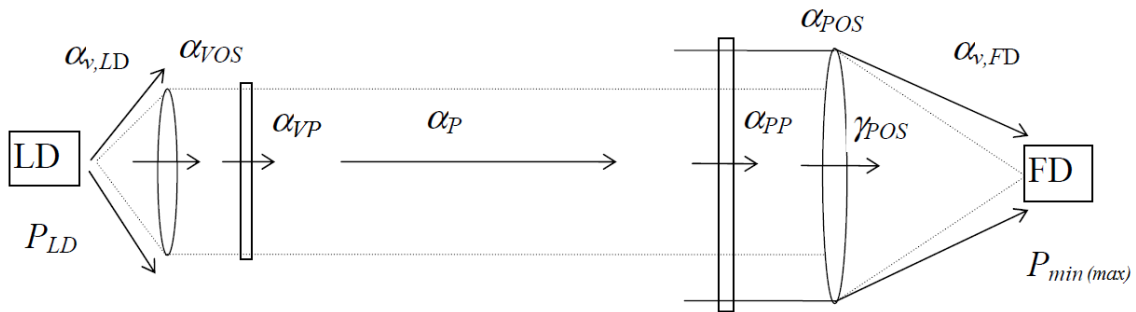
Prijímací systém OBS zabezpečuje príjem a demoduláciu signálu. Skladá sa z prijímacej optickej sústavy, fotodiódy, predzosilňovača a demodulátoru. Zameriavanie prijímacieho systému prebieha rovnako ako pri vysielacom systéme a na ochranu pred nečistotami je aj tu použitý optický priezor.

Zväzok dopadajúci na prijímací systém je sústredený optickou sústavou na aktívnu plochu fotodiódy. Predpokladá sa použitie PIN fotodiódy, ktorá konvertuje dopadajúci

optický výkon na fotoprúd. K zníženiu vplyvu žiarenia pozadia sa používa interferenčný filter, ktorý musí byť navrhnutý s ohľadom a použitú vlnovú dĺžku.

## 1.2 Energetická bilancia optického spoja

Energetická bilancia OBS zahŕňa hlavne (Obrázok 1.1) výkon laserovej diódy  $P_{LD}$ , účinnosť väzby laserovej diódy a vysielacej optickej sústavy  $\alpha_{v,LD}$ , priepustnosť vysielacej a prijímacej optickej sústavy  $\alpha_{VOS}$  a  $\alpha_{POS}$ , priepustnosť optických priezorov  $\alpha_{VP}$  a  $\alpha_{PP}$ , útlm šírením  $\alpha_P$ , zosilnenie prijímacej optickej sústavy  $\gamma_{POS}$ , účinnosť väzby prijímacej optickej sústavy a fotodiódy  $\alpha_{v,FD}$ , útlm vyvolaný nedokonalosťou vzájomného zamierenia hlavíc  $\alpha_Z$ , minimálny detekovateľný optický výkon fotodiódy ako aj výkon, pri ktorom dochádza k saturácii  $P_{min(max)}$  a citlivosť prijímacieho systému  $P_0$ .



Obrázok 1.1:miesta útlmov a zosilnení v energetickej bilancii OBS [1]

Účinnosť väzby  $\alpha_{v,LD}$  závisí na uhlovej šírke a rozložení zväzku vyžarovanom laserovou diódou a na numerickej apertúre vysielacej optickej sústavy. V praxi je možné predpokladať útlm väzby  $\alpha_{v,LD}$  približne 1,5 dB. Parameter  $\alpha_{v,FD}$  je závislý na pomere aktívnej plochy fotodiódy a veľkosti škvry, ktorú v ohniskovej rovine prijímacej optickej sústavy vytvára prijatý optický signál. Pokiaľ je táto škvra menšia ako aktívna plocha fotodiódy, je útlm väzby  $\alpha_{v,FD} = 0$  dB.

Zosilnenie prijímacej optickej sústavy je ovplyvnené pomerom plôch prijímacej a vysielacej apertúry a tiež umiestnením stredu prijímacej apertúry v ose Gaussovho zväzku. Citlivosť prijímacieho systému je definovaná ako minimálna úroveň prijatého optického výkonu nutná na dosiahnutie stanovenej minimálnej hodnoty SNR. Dôležitým parametrom je aj oblasť dynamiky prijímacieho systému  $\Delta_P$ , ktorá sa vypočíta nasledujúcim vzťahom:

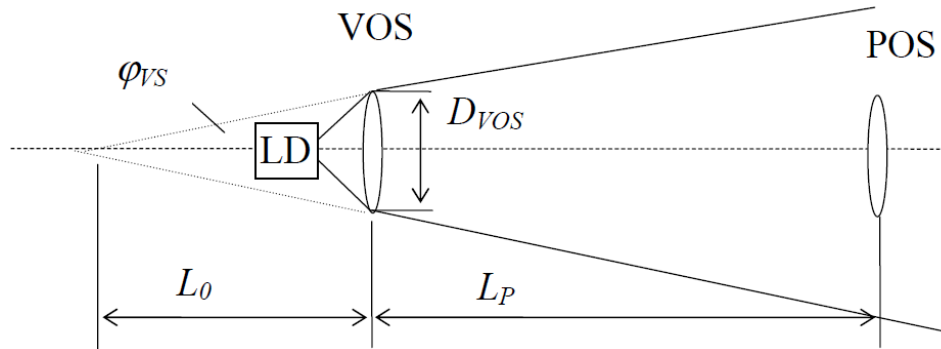
$$\Delta_P = P_{max} - P_0. \quad (1.1)$$

Útlm  $\alpha_Z$  spôsobujú rôzne vplyvy ako mechanické deformácie úchytu hlavice pri aretácii, teplotné deformácie konzol alebo aj nedostatočné skúsenosti obsluhy pri zameriavaní. Empiricky bolo zistené, že tento útlm neprevyšuje hodnotu 1,5 dB.

Najväčší útlm v energetickej bilancii OBS predstavuje útlm šírením. Je určený vzťahom

$$\alpha_P = \left| 20 \log \frac{L_0}{L_0 + L_P} \right|, \quad (1.2)$$

kde  $L_P$  je vzdialenosť medzi hlavicami a  $L_0$  je tzv. pomocná dĺžka (viď Obrázok 1.2).



Obrázok 1.2: význam pomocnej dĺžky  $L_0$  [1]

### 1.3 Atmosférické prenosové prostredie

Prenosové prostredie má veľký vplyv na kvalitu prenosu. Obecné je APP považované za nestacionárne a nehomogénne a jeho vplyv na kvalitatívne parametre prenosového kanálu má náhodný charakter. Okrem komunikácii vo vesmíre prebiehajú všetky OBS v troposfére, ktorá je charakteristická kondenzáciou vodnej pary, tvorením hmly a oblakov, podlieha vplyvom vetra, dažďa a sneženia a vznikajú v nej búrky a veterné víry (turbulencie). Optický zväzok prechádzajúci takýmto prostredím podlieha energetickým a tvarovým zmenám, ktoré môžu vyvolať zmeny úrovne prijímaného signálu.

Základnými veličinami na modelovanie vplyvu APP na kvalitu prenosu OBS sú koeficient extinkcie  $\alpha$  a index lomu prostredia  $n$ . Obe tieto veličiny sú obecné závislé na súradniciach priestoru, čase a dĺžke optickej vlny. Náhodné zmeny týchto veličín v priestore a v čase spôsobujú extinkciu a fluktuáciu optickej intenzity zväzku prechádzajúceho APP. Koeficient extinkcie je možné vyjadriť aj pomocou meteorologickej viditeľnosti  $V_M$ , ktorá sa definuje ako vzdialenosť, pri ktorej priepustnosť nadobúda hodnotu  $T = 0,02$  (pri  $\lambda = 555$  nm). Koeficient  $\alpha$  pre ľubovoľnú dĺžku vlny sa vypočíta zo vzťahu

$$\alpha(\lambda) = \frac{3,91}{V_M \cdot \left(\frac{555}{\lambda}\right)^q}; [\text{km}^{-1}], \quad (1.3)$$

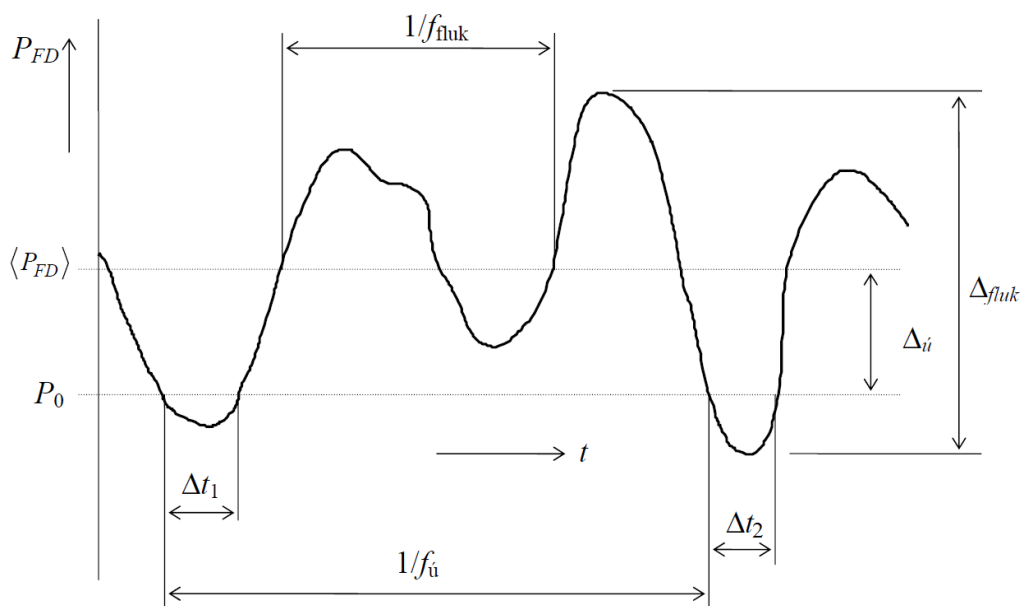
kde  $q = 0,585V_M^{\frac{1}{3}}$  pre  $V_M \leq 6$  km [1]. Podľa meteorologickej viditeľnosti a koeficientu extinkcie rozlišujeme rôzne stavy atmosféry (viď Tabuľka 1.1).

Tabuľka 1.1: Tabuľka stavov APP na základe  $V_M$  [1]

| $V_M$ [km] | $\alpha$ [ $\text{km}^{-1}$ ] | stav atmosféry |
|------------|-------------------------------|----------------|
| > 63       | < 0,062                       | veľmi čistá    |
| 26 – 63    | 0,062 – 0,15                  | čistá          |
| 1,3 – 26   | 0,15 – 3,07                   | opar           |

|            |             |            |
|------------|-------------|------------|
| 0,06 – 1,3 | 3,07 – 61,1 | hmla       |
| < 0,06     | > 61,1      | silná hmla |

V APP sa vyskytujú aj turbulencie, ktoré sa prejavujú najmä fluktuáciou prijímaného optického výkonu s rôznou frekvenciou a amplitúdou. Bežné javy ako je nástup hmly alebo dážď pôsobia s typickými časovými periódami 20 min. či 24 hod., zatiaľ čo teplotné turbulencie troposféry môžu vyvolať zmeny prijímaného optického signálu s frekvenciou rádovo  $10^2$  Hz. V prípade, že poklesne úroveň prijímaného výkonu pod stanovenú minimálnu hodnotu (citlivosť prijímača), sa jedná o tzv. únik. Pri štatistickom hodnotení OBS sa uvádza časový interval (percento času), počas ktorého došlo k únikom. Parametre fluktuácií a únikov možno vidieť na Obrázok 1.3.

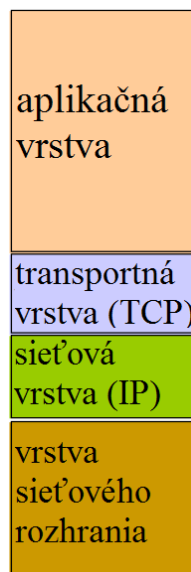


Obrázok 1.3: parametre fluktuácií a únikov [1]  
 ( $P_{FD}$  – optický výkon na fotodióde prijímača,  $P_0$  – citlivosť prijímača,  
 $f_{flukt}$  – frekvencia fluktuácií,  $f_{\dot{u}}$  – frekvencia únikov,  $\Delta_{fluk}$  – dynamika fluktuácií,  
 $\Delta_{\dot{u}}$  – hĺbka únikov,  $t$  – čas,  $\Delta t_i$  – časové intervaly únikov)

Práve kvôli týmto javom v APP je nutné pri návrhu OBS počítať s tzv. rezervou spoja na prenosové prostredie. Udáva sa v jednotkách [dB/km] a vyjadruje mieru možného zhoršenia prenosových vlastností APP, kým nedôjde k úniku.

## 2 PROTOKOL TCP/IP

Označenie TCP/IP zahŕňa celú sústavu protokolov a názorov, ako by mali počítačové siete fungovať. Predpokladá sa tu jednoduchá a rýchla komunikačná sieť, ktorá prepája inteligentné hostiteľské počítače bez vlastného zabezpečovania spoľahlivosti. Problém zabezpečenia spoľahlivosti tak pripadá na koncových účastníkov komunikácie a je teda riešený až na úrovni transportnej vrstvy (viď Obrázok 2.1). Komunikačná sieť tak môže byť naplno využívaná pre vlastný dátový prenos.



Obrázok 2.1: vrstvy TCP/IP modelu [3]

### 2.1 Vrstva sieťového rozhrania

Najnižšia vrstva má na starosti všetko, čo je spojené s riadením konkrétnej prenosovej cesty a priamym vysielaním a prijímaním dátových paketov. Táto vrstva nie je priamo definovaná sústavou TCP/IP a je závislá na konkrétnej použitej prenosovej technológii. Môže byť preto tvorená jednoduchým ovládačom, ale aj veľmi zložitým subsystémom s vlastným linkovým protokolom.

### 2.2 Sieťová vrstva (IP)

Sieťová vrstva zabezpečuje pri komunikácii smerovanie. To znamená, že jej úlohou je nájsť optimálnu trasu v sieti pre dopravu dát medzi koncovými systémami. V sústave TCP/IP to má na starosti protokol IP (InterNet Protocol).

#### 2.2.1 IP protokol

Protokol IP na rozdiel od bežných linkových protokolov dopravuje dáta medzi dvoma

ľubovoľnými počítačmi cez tzv. smerovače. Každý z týchto smerovačov samostatne rieši smerovanie dátového paketu k nasledujúcemu smerovaču, prípadne ku koncovému zariadeniu. Takýmto spôsobom je možné prenášať dáta aj cez mnohé siete LAN. Každé sieťové rozhranie je v IP protokole charakterizované aspoň jednou IP adresou. Pre IP protokol verzie 4 má veľkosť štyri bajty a pre verziu 6 to je šesťnásť bajtov. Dáta sa prenášajú vo forme IP datagramov.

## 2.2.2 IP datagram

IP datagram (Obrázok 2.2) tvorí záhlavie a prenášané dáta. Pokiaľ neobsahuje voliteľné položky, má záhlavie spravidla veľkosť 20 bajtov. Prvé 4 bity záhlavia sú vyhradené pre označenie verzie protokolu IP. Druhá 4-bitová položka obsahuje dĺžku záhlavia vyjadrenú v „štvorbajtoch“ (pre záhlavie veľkosti 20B je jeho dĺžka 5). Nasleduje 8 bitov pre typ služby. V praxi by mala táto položka slúžiť na špecifikáciu kvality prenosu IP datagramu, no smerovače ju väčšinou ignorujú. Ďalších 16 bitov je vyhradených pre celkovú dĺžku IP datagramu v bajtoch. Z toho plynie maximálna dĺžka IP datagramu rovná 65535 bajtov. Nasledujú položky identifikácia IP datagramu, príznaky a posunutie fragmentu (dokopy 32 bitov), ktoré sú využívané mechanizmom fragmentácie datagramu. Záhlavie ďalej obsahuje 8-bitovú informáciu o dobe života datagramu (time to live – TTL), ktorá zabraňuje nekonečnému túlaniu IP datagramu v sieti. Ak smerovač nemôže znížiť hodnotu TTL, IP datagram sa zahadzuje a odosielateľ je signalizovaný prostredníctvom protokolu ICMP. Ďalších 8 bitov obsahuje číselnú identifikáciu protokolu vyššej vrstvy, ktorý daný IP datagram využíva. Nasleduje kontrolný súčet (16 bitov) záhlavia a 32-bitové IP adresy odosielateľa a príjemcu IP datagramu. Voliteľné položky sú vo väčšine prípadov nevyužitú.

|                                  |                             |  |                                  |                                       |
|----------------------------------|-----------------------------|--|----------------------------------|---------------------------------------|
| verzia IP (4b)                   | dĺžka záhlavia (4b)         | typ služby (8b)                                | celková dĺžka IP datagramu (16b) |                                       |
| identifikácia IP datagramu (16b) |                             |  | príznaky (3b)                    | posunutie fragmentu od počiatku (13b) |
| doba života IP datagramu (8b)    | protokol vyššej vrstvy (8b) | kontrolný súčet zo záhlavia IP datagramu (16b) |                                  |                                       |
| IP adresa odosielateľa (32b)     |                             |  |                                  |                                       |
| IP adresa príjemcu (32b)         |                             |  |                                  |                                       |
| voliteľné položky záhlavia       |                             |  |                                  |                                       |
| prenášané dáta (nepovinné)       |                             |  |                                  |                                       |

Obrázok 2.2: IP datagram [4]

Protokol ICMP je súčasťou IP protokolu a funguje ako služobný protokol pre signalizáciu mimoriadnych stavov. Medzi najdôležitejšie patria:

- Echo – nástroj využívaný hlavne programom ping, ktorým môžeme testovať dostupnosť jednotlivých uzlov v sieti.
- Nedoručiteľný IP datagram – ak nemôže smerovač predať IP datagram ďalej k adresátovi, je tento zahodený a odosielateľ informovaný ICMP správou.
- Zníž rýchlosť odosielania – v prípade preťaženia siete niekde medzi



odosielateľom a príjemcom, vyšle túto správu smerovač, ktorý nie je schopný predávať ďalej všetky IP datagramy.

- Čas vypršal – táto správa odlišuje dva rôzne prípady. S kódom 0 znamená, že položka TTL bola znížená na nulu a IP datagram zahodený. Kód 1 signalizuje, že adresát nie je schopný v danom čase zostaviť z fragmentov celý IP datagram.

## 2.3 Transportná vrstva (TCP)

Transportná vrstva rieši komunikáciu medzi koncovými užívateľmi. Medzi jej typické funkcie patrí príjem dát z vyššej vrstvy, ich delenie na prenosové a dátové jednotky, predávanie paketov sieťovej vrstve a v prípade TCP aj zabezpečovanie spoľahlivosti prenosu dát.

### 2.3.1 Protokol TCP

Základné problémy, ktoré protokol TCP musí riešiť, sú:

- riadenie toku – odosielateľ nesmie zahltiť príjemcu. Pri preťažení je nutné obmedziť rýchlosť odosielania dát.
- chybové riadenie – musí byť zaistené opakované odoslanie dát v prípade ich straty alebo výskytu chyby.

Tieto mechanizmy sú založené na kladnom potvrdzovaní prijatých dát. To znamená, že ak odosielateľ nedostane potvrdenie prijatia, automaticky vyhodnotí situáciu ako stratu (chybu) a dáta odošle znovu. Pokiaľ sa potvrdzuje každý prijatý paket, jedná sa o metódu Stop-and-Wait. Pri metóde s plávajúcim oknom (Sliding Window) príjemca potvrdzuje naraz viacero prijatých paketov. Výhoda metódy s plávajúcim oknom spočíva v možnosti dočasnej nezávislosti rýchlosti vysielania paketov na rýchlosti ich odoberania príjemcom.

Kladné prijatie paketu príjemcom je potvrdzované odpoveďou ACK. Pri prijatí chybného paketu, posíla príjemca záporné potvrdenie NAK. Ak sa paket alebo potvrdenie stratí niekde v sieti, odosielateľ posíla odoslané dáta znovu po uplynutí časového intervalu (timeout). Dáta sú prenášané vo forme tzv. TCP segmentov.

### 2.3.2 TCP segment

Protokol TCP musí neformátovaný prúd bajtov rozdeliť do segmentov tak, aby sa zmestili do IP datagramov. To v praxi znamená, že maximálna dĺžka jedného segmentu je 65535 bajtov mínus dĺžka záhlavia TCP segmentu. Taktiež je vhodné voliť dĺžku TCP segmentov tak, aby nebola nutná fragmentácia výsledných IP datagramov prenášaných linkovými rámcami a tým sa nezvyšovali nároky na režiu dátového prenosu.

Obrázok 2.3 zobrazuje zloženie TCP segmentu. Zdrojový a cieľový port nesú identifikáciu portu odosielateľa a príjemcu. Poradové číslo odosielaného bajtu je číslo prvého bajtu TCP segmentu v toku dát od odosielateľa ku príjemcovi a poradové číslo prijatého bajtu je číslo nasledujúceho bajtu, ktorý je príjemca schopný prijať. Dĺžka záhlavia a kontrolný súčet fungujú u TCP segmentoch rovnako ako pri IP datagramoch. Dĺžka okna vyjadruje prírastok poradového čísla bajtu, ktorý bude príjemcom ešte akceptovaný. Ukazovateľ naliehavých dát môže byť nastavený len v prípade, že je

nastavený príznak URG. Znamená to, že TCP segment obsahuje naliehavé dáta, ktoré by mal príjemca spracovať prednostne (využíva napr. protokol Telnet).

|   |              |                                  |                  |
|---|--------------|----------------------------------|------------------|
| zdrojový port (16b)                     |              | cieľový port (16b)               |                  |
| poradové číslo odosielaného bajtu (32b) |              |                                  |                  |
| poradové číslo prijatého bajtu (32b)    |              |                                  |                  |
| dĺžka záhlavia (4b)                     | rezerva (6b) | príznamy (6b)                    | dĺžka okna (16b) |
| kontrolný súčet záhlavia (16b)          |              | ukazovateľ nalievavých dát (16b) |                  |
| voliteľné položky záhlavia              |              |                                  |                  |
| prenášané dáta (nepovinné)              |              |                                  |                  |

Obrázok 2.3: TCP segment [4]

V poli príznakových bitov záhlavia môžu byť nastavené nasledujúce položky:

- URG – TCP segment obsahuje naliehavé dáta.
- ACK – TCP segment má platné pole „poradové číslo prijatého bajtu“.
- PSH – spravidla signalizuje, že TCP segment obsahuje aplikačné dáta. (Tento príznak nemá ustálené používanie.)
- RST – odmietnutie TCP spojenia.
- SYN – odosielateľ začína novú sekvenciu číslovania.
- FIN – odosielateľ ukončil posielanie dát. (Uzatvorenie prenosu len v jednom smere.)

Voliteľné položky záhlavia sú zložené z typu, dĺžky a hodnoty danej voliteľnej položky. Ich maximálna veľkosť vyplýva z maximálnej dĺžky záhlavia. Tá je kvôli veľkosti poľa „dĺžka záhlavia“ obmedzená na 60 bajtov, z toho je 20 bajtov obsadených povinnými položkami, takže pre voliteľné položky ostáva maximálne 40 bajtov. Zároveň platí, že dĺžka záhlavia TCP segmentu musí byť deliteľná 4. V prípade, že nie je, dopĺňuje sa prázdnu voliteľnou položkou NOP.

## 2.4 Aplikačná vrstva

Aplikačná vrstva je najvyššou vrstvou protokolu TCP/IP. V sieťových aplikáciách pracuje veľké množstvo aplikačných protokolov, ktoré sa delia na užívateľské (HTTP, SMTP, Telnet, FTP, IMAP, atď.) a služobné (SNMP, S/MIME, SSL). Užívateľské protokoly sú využívané priamo užívateľskými aplikáciami, zatiaľ čo so služobnými protokolmi sa väčšinou užívatelia priamo nestretnú. Tie zabezpečujú správnu funkciu siete, nastavovanie smerovacích tabuliek v smerovačoch, prípadne majú na starosti zabezpečenie dát. V protokole TCP/IP komunikujú sieťové aplikácie v aplikačnej vrstve priamo s transportnou vrstvou, ktorej predávajú dáta určené na transport. Aby bol úplne určený cieľ v sieti, musia byť posielané dáta doplnené o IP adresu a port cieľa a tiež protokol, ktorý sa použije na prenos (najčastejšie TCP alebo UDP).

## 3 MODEL OBS PRE SIMULÁCIU

Mnoho štúdií preukázalo nevýhodné správanie protokolu TCP pri prechode prenosovým kanálom, ktorý je ovplyvnený turbulenciami. Mechanizmus TCP slúžiaci na zabránenie preťaženia linky nesprávne vyhodnocuje krátke úniky signálu a to následne vedie k poklesu prenosovej rýchlosti.

Analýza a simulácia komunikačných protokolov je zvyčajne založená na štatistickom modeli. To platí za predpokladu, že daný komunikačný protokol je možné analyticky popísať. V prípade komplexnejších protokolov je výhodnejšie použiť časovú analýzu, ktorá je však náročnejšia na výpočtový výkon.

### 3.1 Návrh modelu

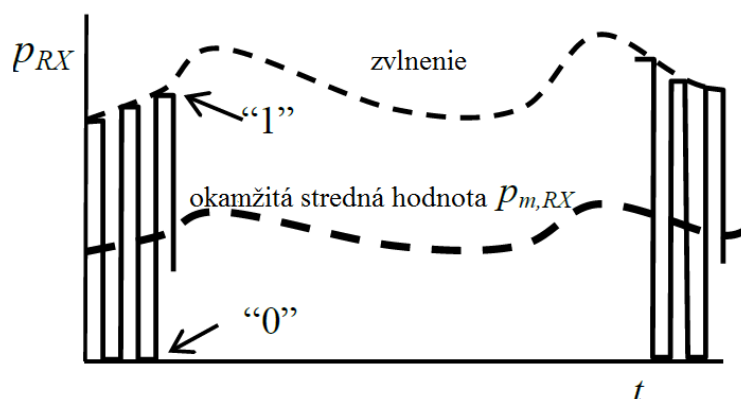
#### 3.1.1 Prijatý optický výkon

S ohľadom na predpokladanú linearitu kanálu môžeme vyjadriť strednú hodnotu prijatého optického výkonu ako [2]

$$p_{m,RX} = p_{m,TX} \alpha_{FSL} \alpha_{ATM} \alpha_T, \quad (3.1)$$

kde  $p_{m,RX}$  a  $p_{m,TX}$  predstavujú strednú hodnotu prijatého a vyžarovaného optického výkonu vztiahnutú k príslušnej apertúre,  $\alpha_{FSL}$  je bežný útlm spôsobený prechodom APP,  $\alpha_{ATM}$  je absorpčný útlm nehostinnej atmosféry (hmla, dážď a pod.) a  $\alpha_T$  reprezentuje efekt spôsobený atmosférickými turbulenciami. Koeficient  $\alpha_{FSL}$  je pre daný OBS konštantný. Hodnota útlmu  $\alpha_{ATM}$  je síce časovo premenná, ale jedná sa o veľmi dlhý proces (počas silnej hmly môže byť OBS nedostupný aj niekoľko hodín), a preto ju budeme pre potreby tohto modelu považovať za konštantnú.

Efekt atmosférických turbulencií spôsobuje prerozdelenie optického výkonu v optickom zväzku. Za predpokladu, že apertúra prijímacieho systému je menšia ako stopa zväzku v jej rovine, spôsobuje tento jav fluktuácie prijatého optického výkonu. Ak uvažujeme OBS s OOK a prenosovými rýchlosťami 100Mb/s a vyššími, je čas potrebný na prenos jedného bitu o niekoľko rádov kratší ako „perióda“ fluktuácií (viď Obrázok 3.1). Hodnoty  $p_{m,RX}$  a  $p_{m,TX}$  sú vyčísľované za čas niekoľkých bitových periód, no tento čas bude stále kratší ako koherenčný čas prenosového kanálu. Z tohto dôvodu môžeme tieto hodnoty považovať za okamžité.



Obrázok 3.1: okamžitá hodnota stredného optického výkonu pri ON-OFF kľúčovaní [2]

### 3.1.2 Bitová chybovosť

Bitová chybovosť OBS sa mení v závislosti na okamžitej hodnote prijímaného optického výkonu. Na základe predošlých predpokladov o dátovej rýchlosti je bitová chybovosť počas prenosu jedného paketu považovaná za konštantnú. Pri uvažovaní OOK s rovnakou pravdepodobnosťou výskytu symbolu „0“ a „1“ je možné okamžitú bitovú chybovosť vyjadriť vzťahom [2]

$$P_b = Q(p_{m,RX}/P_N), \quad (3.2)$$

kde  $P_N$  charakterizuje šumové vlastnosti prijímacieho systému a  $Q$  predstavuje Gaussov integrál pre výpočet pravdepodobnosti vzniku chyby.

V mnohých prípadoch je uvádzaná hranica prijímaného optického výkonu  $p_{m0}$  pre dosiahnutie určitej chybovosti  $P_{b0}$ . Potom platí pre  $P_N$  výraz [2]

$$P_N = \frac{p_{m0}}{-F^{-1}(P_{b0})}, \quad (3.3)$$

kde  $F^{-1}$  predstavuje inverznú distribučnú funkciu. Prijatý optický výkon môže byť vyjadrený aj pomocou rezervy spoja  $M$  [2]:

$$p_{m,RX} = M\alpha_T p_{m0}. \quad (3.4)$$

Skombinovaním vzťahov (3.2 – 3.4) vznikne upravený výraz pre výpočet okamžitej bitovej chybovosti

$$P_b = Q(-M\alpha_T F^{-1}(P_{b0})). \quad (3.5)$$

Následne je možné vypočítať pravdepodobnosť, že paket o dĺžke  $n$  bitov bude stratený (zahodený) podľa vzťahu [2]

$$P_F = 1 - (1 - P_b)^n. \quad (3.6)$$

### 3.1.3 Modelovanie turbulencií

Atmosférické turbulencie sú reprezentované náhodne generovaným koeficientom útlmu  $\alpha_T$  s určenou strednou hodnotou a rozptylom  $\sigma_T^2$  (PSI), ktorý charakterizuje hĺbku vznikajúcich fluktuácií optického výkonu. Obrázok 3.2 zobrazuje proces generácie koeficientov útlmu. V prvom bloku je generovaná náhodná postupnosť  $z_i$ , ktorá má normálne rozloženie. Druhý blok predstavuje FIR filter, ktorý do vygenerovanej náhodnej

postupnosti zavádza autokorelačné vlastnosti prenosového kanálu. Autokorelačná funkcia je daná vzťahom

$$R_{ln \alpha_T}(\tau) = e^{-a|\tau/\tau_0|^b}, \quad (3.7)$$

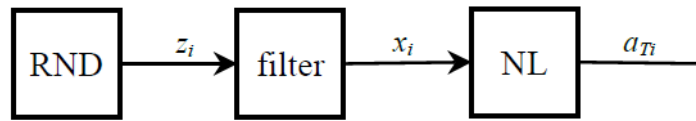
kde  $\tau_0$  je korelačná doba prenosového kanálu a koeficienty  $a = 0,5$  a  $b = 1$  pre Gauss-Markov model [2].

Zo vzniknutej postupnosti  $x_i$  sú nelineárnou transformáciou počítané koeficienty  $\alpha_{Ti}$  podľa vzťahu [2]

$$\alpha_{Ti} = e^{(x_i - \sigma_L^2/2)}, \quad (3.8)$$

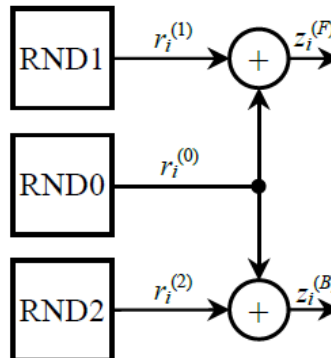
$$\sigma_L^2 = \ln(\sigma_T^2 + 1), \quad (3.9)$$

kde  $\sigma_L^2$  je logaritmický rozptyl vzťahnutý k PSI.



Obrázok 3.2: generátor náhodných vzoriek koeficientov  $\alpha_T$  [2]

Pri uvažovaní duplexného bezdrôtového optického spoja je však potrebné generovať zároveň dve postupnosti bitovej chybovosti. Jednu pre vysielací a druhú pre prijímací kanál. V praxi sú si tieto chybovosti veľmi podobné, ich vzájomná korelácia dosahuje 95% až 99%. To je dané ich umiestnením blízko seba. Je preto jasné, že základom takýchto generovaných bitových chybovostí musia byť dve postupnosti náhodných čísel, ktoré spĺňajú spomínané kritérium vzájomnej korelácie. Toho je možné dosiahnuť generovaním troch náhodných postupností a následným prepočtom, ako je uvedené na Obrázok 3.3.



Obrázok 3.3: generátor dvoch vzájomne korelovaných náhodných postupností [6]

Prepočet z troch náhodných postupností na dve vzájomne korelované je možné vykonať pomocou vzťahov [6]

$$z^{(F)} = \sqrt{k}r^{(0)} + \sqrt{1-kr^{(1)}} \text{ a } z^{(B)} = \sqrt{k}r^{(0)} + \sqrt{1-kr^{(2)}}, \quad (3.10)$$

kde  $r^{(0)}, r^{(1)}, r^{(2)}$  sú nezávislé náhodne generované postupnosti s normálnym rozložením a  $k$  je koeficient zmiešania. Ten je daný vzťahom [6]

$$k = \frac{\ln(\rho\sigma_T^2+1)}{\ln(\sigma_T^2+1)}, \quad (3.11)$$

kde  $\rho$  vyjadruje mieru vzájomnej korelácie (tzv. Pearsonov korelačný koeficient).

## 3.2 Implementácia do simulátoru

Pri implementácii navrhnutého generátora do sieťového simulátora ako je napr. OMNeT++ treba brať ohľad na to, že simulácia sa vykonáva na základe udalostí. V praxi to znamená, že procedúra je aktívna vždy len pri spracovávaní paketu a opäť sa „zobudí“ až pri príchode ďalšieho paketu. S tým je treba počítať pri generovaní náhodných postupností  $z$  a generovať vždy príslušný počet čísel podľa uplynutého času medzi dvoma paketmi.

### 3.2.1 Generovanie okamžitej bitovej chybovosti

Generovanie náhodných postupností  $r_i$  prebieha pomocou funkcie  $\text{normal}(0, 1)$ . Tá má ako vstupné parametre strednú hodnotu a rozptyl a na výstupe vracia náhodné číslo typu `double` podliehajúce normálnemu rozloženiu so zadanými parametrami. Takto vygenerované postupnosti sú uložené do polí `ran0`, `ran1` a `ran2` a následne podľa vzťahov (3.10) a (3.11) prepočítané na vzájomne korelované náhodné postupnosti `zrx` a `ztx` s veľkosťou  $[2*N+1]$ .  $N$  tu predstavuje dĺžku odozvy FIR filtru. Postupnosti `zrx` a `ztx` následne prechádzajú spomínaným lineárnym filtrom, ktorý je daný autokorelačnou funkciou zo vzťahu (3.7). Nakoniec je s využitím vzťahov (3.9), (3.8) a (3.5) vypočítaná okamžitá hodnota bitovej chybovosti (`berrx` a `bertx`).

V priebehu simulácie je počítaný uplynutý čas medzi dvoma paketmi a následne podľa toho generovaný potrebný počet náhodných čísel  $r_i$ . V prípade, že je uplynutá doba medzi dvoma paketmi dlhšia ako veľkosť polí  $[2*N+1]$  vynásobená vzorkovacou periódou, vygenerujú sa celé postupnosti nanovo, rovnako ako pri spustení simulácie.

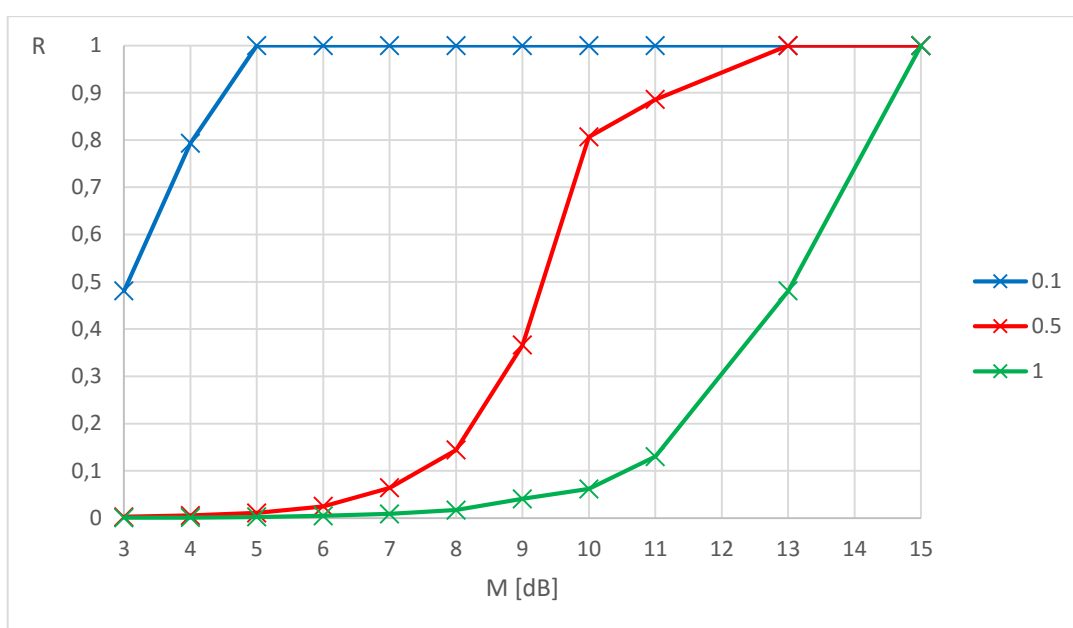
### 3.2.2 Simulovanie vyrovnávacej pamäte na vysielachi

Pri prichádzajúcom pakete simulátor z vygenerovanej bitovej chybovosti pre prijímač optického bezdrôtového spoja rozhoduje na základe predom určenej hraničnej hodnoty chybovosti, či bude paket odoslaný hneď, alebo bude zdržaný vo vyrovnávacej pamäti. V prípade, že je vygenerovaná chybovosť menšia ako zvolená hranica, simulátor vypočíta podľa vzťahu (3.6) pravdepodobnosť výskytu chyby v pakete a túto hodnotu porovnáva s náhodným číslom z intervalu  $[0, 1)$ . Na základe tohto porovnania sa rozhoduje, či bude daný paket zahodený. Tu je samozrejme použitá vygenerovaná bitová chybovosť na trase vysielacha. Ak však hodnota okamžitej bitovej chybovosti na prijímači prekročí stanovenú hranicu, dochádza ku postupnému generovaniu nových hodnôt tejto chybovosti a paket je odoslaný až po opätovnom znížení bitovej chybovosti na určenú hodnotu. Takýmto spôsobom je simulovaná vyrovnávacia pamäť na strane vysielacha.

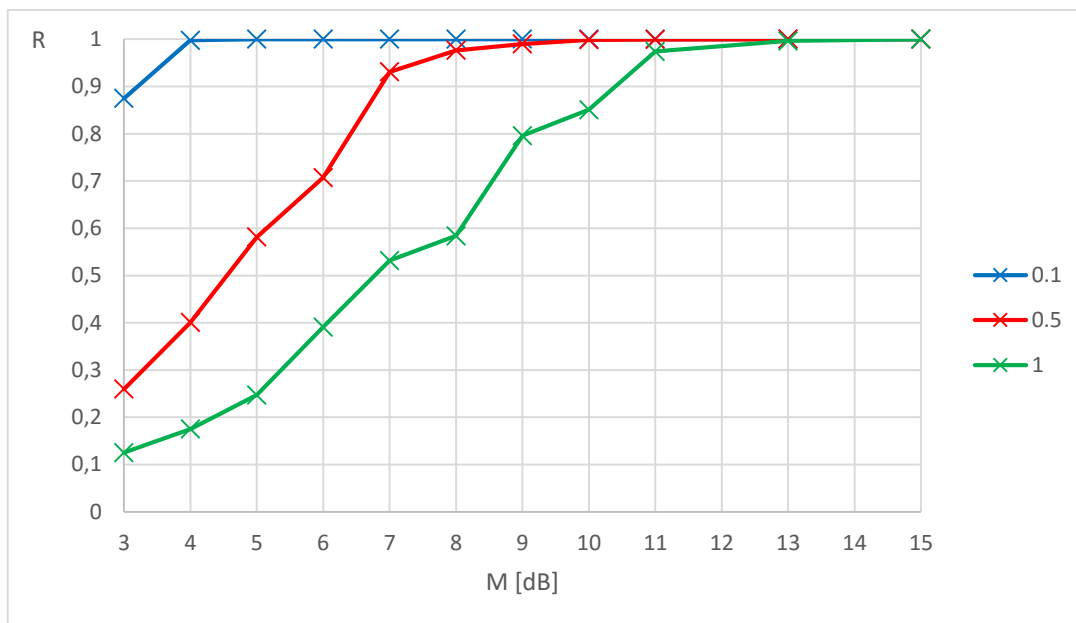
## 3.3 Simulácia

Simulovaná bola komunikácia client-server s použitým protokolom TCP/IP. Maximálna

prenosová rýchlosť bola nastavená na 1Gbps a veľkosť prenášaných dát predstavovala 1GB. To znamená, že pri bezchybnom prenose plnou rýchlosťou by mal byť prenos úspešne ukončený za približne 8 sekúnd. Vzhľadom k tomu, že vplyvom nastavovaných hodnôt linkovej rezervy  $M$  a hĺbky fluktuácií prijímaného optického výkonu  $PSI$  sa prenosová rýchlosť v niektorých prípadoch značne znížila, stanovil som maximálne trvanie simulácie na 60 sekúnd. K ukončeniu simulácie tak dôjde buď po prenesení objemu dát 1GB alebo po uplynutí 60 sekúnd simulačného času. Následne je vypočítaná priemerná prenosová rýchlosť z celkového objemu prenesených dát a konečnej hodnoty simulačného času. Takto získané hodnoty zo simulácií s rôznymi  $M$  a  $PSI$  sú zobrazené v grafoch závislostí (Obrázok 3.4 a Obrázok 3.5). Na x-ovej osi sú hodnoty linkovej rezervy a na osi y priemerná prenosová rýchlosť vztiahnutá ku jej maximálnej hodnote dosiahnutej pri bezchybnom prenose. Parametrom v oboch grafoch je hodnota  $PSI$ , pričom  $PSI = 0,1$  predstavuje slabé turbulencie a  $PSI = 1$  naopak veľmi silné.



Obrázok 3.4: výsledky simulácie (bez vyrovnávacej pamäti)



Obrázok 3.5: výsledky simulácie (s vyrovnávacou pamäťou)

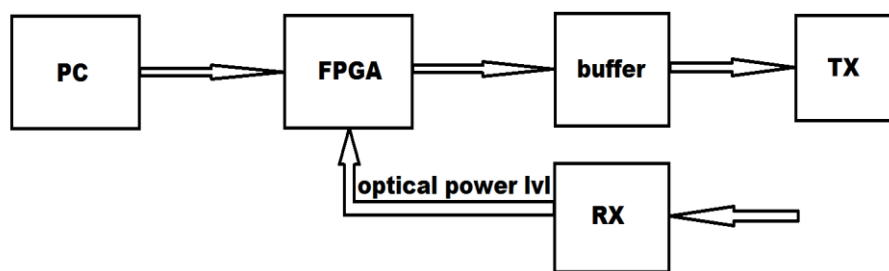
Z uvedených grafov závislostí vidno, že s vyrovnávacou pamäťou vo vysieláči bola priemerná prenosová rýchlosť výrazne vyššia. Zároveň simulácie ukazujú, že pri slabých turbulenciách ( $PSI = 0,1$ ) je na bezchybný prenos postačujúca linková rezerva na úrovni približne 5dB, zatiaľ čo pri silných turbulenciách ( $PSI = 1$ ) je to už 15dB. Priebeh simulácie je zobrazený v prílohe **Chyba! Nenašiel sa žiaden zdroj odkazov..**



## 4 REALIZÁCIA VYROVNÁVACEJ PAMÄTE POMOCOU FPGA

### 4.1 Návrh vyrovnávacej pamäte vo vysieläči

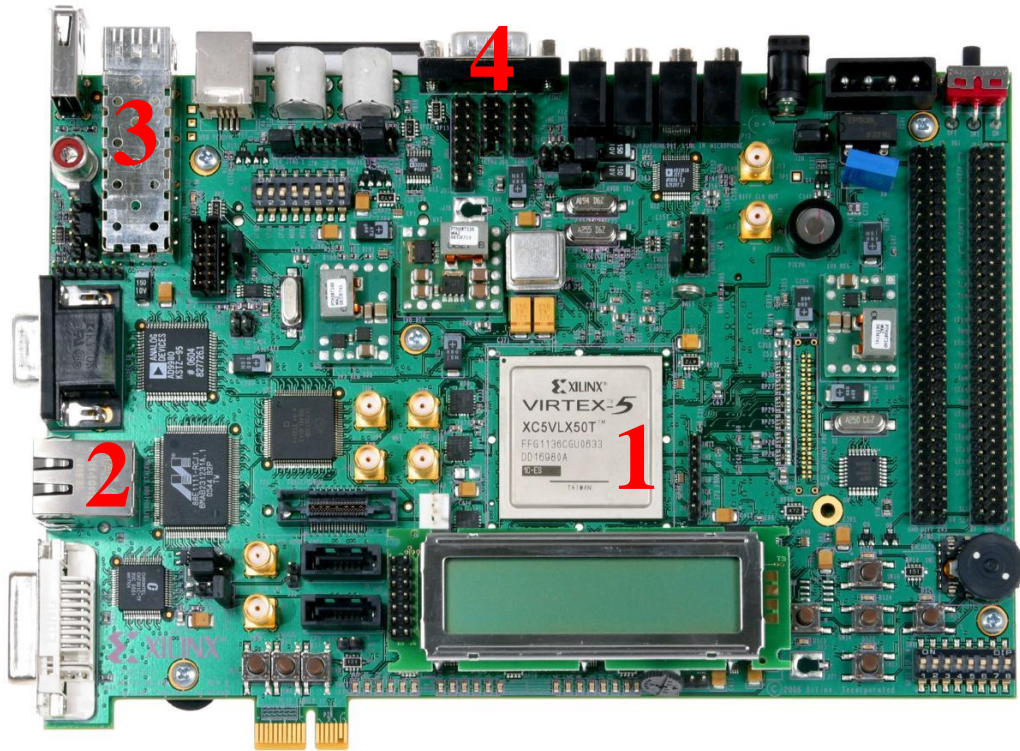
Aby sa zabránilo stratám prenášaných dátových paketov v prípade výpadku optickej linky, je vhodné využiť vyrovnávaciu pamäť typu FIFO na strane optickej vysieläča. Keď dôjde vplyvom atmosférických turbulencií k úniku optickej linky, pozastaví sa vysielanie a dátové pakety prichádzajúce do optickej vysieläča sa ukladajú do vyrovnávacej pamäte. Pri opätovnom obnovení optickej linky sú následne vysielané pakety vo vyrovnávacej pamäti v poradí, v akom boli do nej zapísané. Pamäť je riadená FPGA modulom. Ten prijíma informáciu o stave prijímaného optickej linky a na jej základe rozhoduje o tom, či dôjde k pozastaveniu vysielania nastavením príslušného príznakového bitu. Obrázok 4.1 zobrazuje blokové znázornenie použitia vyrovnávacej pamäte. Prijímaný optický signál nie je totožný so signálom vysielaným, no vzhľadom na vysokú úroveň vzájomnej korelácie (95% až 99%) ich budeme považovať za rovnaké.



Obrázok 4.1: Znázornenie vyrovnávacej pamäte

#### 4.1.1 Vývojová doska

Na realizáciu metódy vyrovnávacej pamäte vo vysieläči OBS som použil vývojovú dosku ML505 (Obrázok 4.2) od firmy Xilinx osadenú Virtex-5 FPGA čipom. Okrem štandardu ako je napríklad ethernet, či sériový port obsahuje táto vývojová doska aj slot pre SFP modul. Ten možno osadiť ako optickým, tak aj metalickým rozhraním, ktoré je jednoduchšie použiť pri vývoji a testovaní.



Obrázok 4.2: Vývojová doska ML505

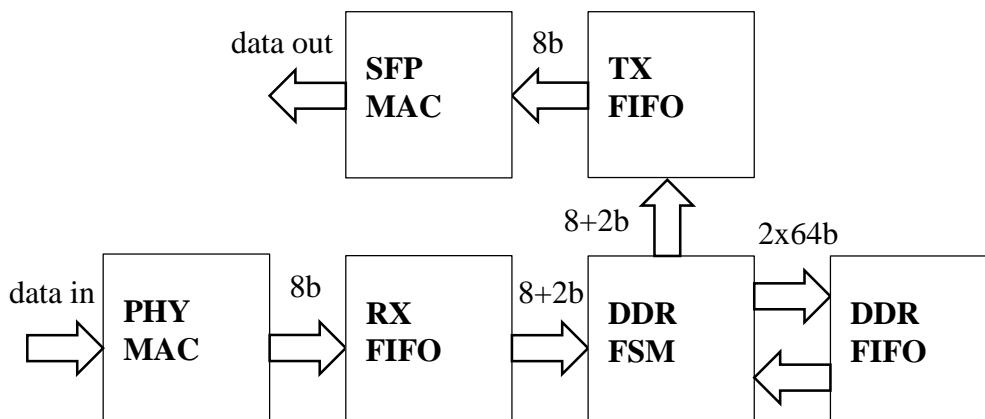
- Na obrázku vývojovej dosky sú označené:
- 1 – FPGA čip (XC5VLX50T)
  - 2 – Ethernet konektor
  - 3 – SFP slot
  - 4 – sériový port

Na spodnej strane sa nachádza slot pre externú pamäť RAM osadený 256MB DDR2 modulom. Samotný FPGA čip obsahuje aj integrovanú pamäť RAM, ktorej veľkosť dosahuje necelý megabajt. Keď vezmeme do úvahy prenosovú rýchlosť spoja na úrovni 1Gb/s a dobu výpadku približne okolo 10ms, dostávame sa k potrebnej veľkosti vyrovnávacej pamäti aspoň 1,25MB. Z toho plynie, že integrovaná pamäť v FPGA čipe nebude stačiť a je preto nutné použiť osadenú pamäť DDR2. Tú je možné implementovať len ako jednoportovú, čo znamená, že nie je možné vykonávať naraz operácie čítania a zápisu. Je preto nutné zvoliť vhodný systém časového multiplexovania prístupu k DDR2 pamäti.

#### 4.1.2 Návrh usmernenia dátového toku

Blokový diagram návrhu vyrovnávacej pamäte je zobrazený na Obrázok 4.3. Dáta prijaté ethernetovým rozhraním smerujú v blokoch o veľkosti 8b do vstupnej vyrovnávacej pamäte FIFO. Tu prebieha kontrola chýb v paketoch a vytvorenie príznakových bitov sof (start of frame) a eof (end of frame). Dáta spolu s príznakovými bitmi následne vstupujú do stavového automatu, ktorý má na starosti riadenie a komunikáciu s DDR2 pamäťou. Ten obsahuje dva bloky pamäte RAM vytvorené z internej pamäte FPGA čipu. Jeden je vstupný, z neho sa dáta zapisujú do DDR2 pamäte

a druhý je výstupný, do neho sú dáta z DDR2 pamäte vyčítané. Oba tieto RAM bloky musia byť vygenerované ako dvojportové, pretože šírka dátovej zbernice pamäte DDR2 nie je rovnaká so šírkou dátovej zbernice vstupnej a výstupnej vyrovnávajúcej pamäte FIFO. Stavový automat teda musí sledovať objem dát vo vstupnom bloku RAM a povoliť zápis do DDR2 pamäte len ak je dostatočný na čo najväčšie využitie šírky dátovej zbernice. Tu je treba počítať s hodnotou minimálne 2 krát 64b, pretože zápis do pamäte DDR2 prebieha vždy aspoň v dvoch slovách po sebe (s vzostupnou aj zostupnou hranou hodinového signálu). Rovnakým spôsobom prebieha aj čítanie dát z pamäte DDR2 do výstupného RAM bloku stavového automatu. Odtiaľ sú dáta smerované do výstupnej FIFO pamäte, kde opäť prebehne kontrola paketu, ktorý je následne odosielaný na výstupné rozhranie SFP modulu.



Obrázok 4.3: tok dát cez pamäť DDR2

### 4.1.3 Informácie o stave optického signálu

Informáciu o úrovni optického signálu je možné posielat' do FPGA čipu sériovou linkou. Predpoklad je, že FPGA pozná hraničnú hodnotu úrovne prijímaného optického signálu pre daný OBS. Túto hodnotu porovnáva s prijatou aktuálnou hodnotou a na základe výsledku nastavuje príslušný príznakový bit. V tomto prípade sa jedná o tx\_enable pre SFP modul.

Pri poklese úrovne prijímaného optického výkonu pod stanovenú hodnotu, je tx\_enable nastavené na nízku úroveň, čo zapríčini zastavenie vysielania SFP modulu. Z vyrovnávacej pamäti je súčasne zakázané čítať ale prichádzajúce pakety sa do nej stále zapisujú. Keď úroveň optického výkonu opäť stúpne nad stanovenú hodnotu je vysielanie znova povolené a z vyrovnávacej pamäti je možné čítať dáta. Pre potrebu konkrétneho spoja je taktiež možné nastaviť hysteréziu vypínania vysielateľa.

Dáta s informáciou o úrovni prijímaného optického signálu sú posielané rýchlosťou 9600baud/s, čo odpovedá vzorkovacej frekvencii úrovne signálu 1kHz.

## 4.2 Realizácia návrhu vo VHDL

### 4.2.1 Obsluha sieťových rozhraní

Správne nastavenie ethernetu a SFP modulu má na starosti vložený mikrokontrolér PicoBlaze. Prístup k samotným signálom zabezpečuje vygenerované IP jadro Ethernet MAC. To disponuje potrebnými hodinovými signálmi, ako aj signálmi prijímača a vysielča.

```
-- Client Receiver Interface - EMAC0
EMAC0CLIENTRXD          : out  std_logic_vector(7 downto 0);
EMAC0CLIENTRXDVLD      : out  std_logic;
EMAC0CLIENTRXGOODFRAME : out  std_logic;
EMAC0CLIENTRXBADFRAME  : out  std_logic;
EMAC0CLIENTRXFRAMEDROP : out  std_logic;

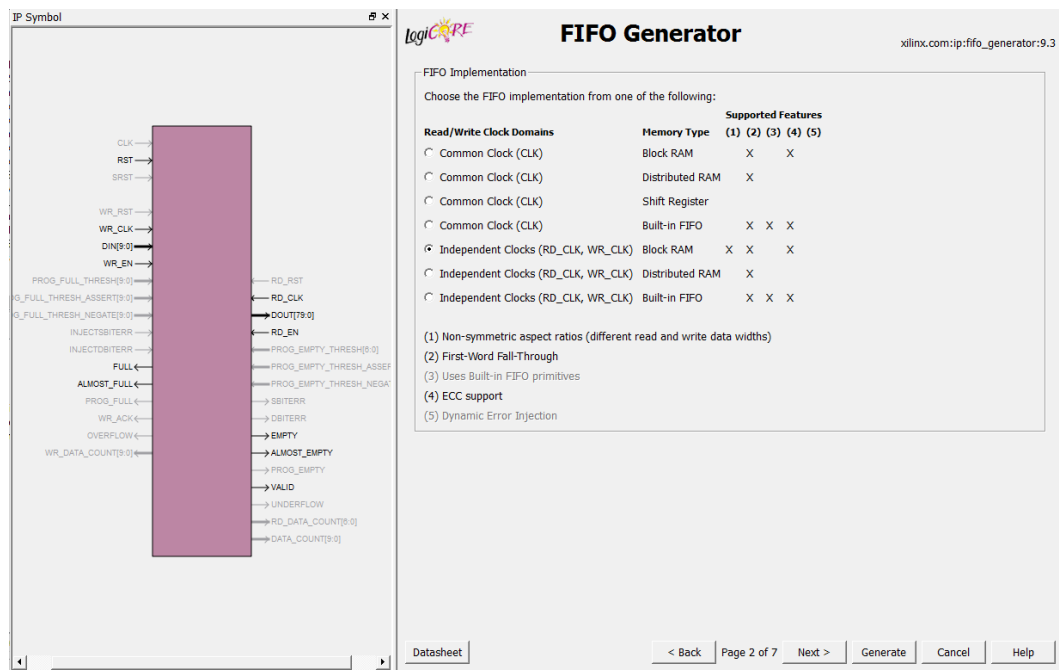
-- Client Transmitter Interface - EMAC0
CLIENTEMAC0TXD          : in   std_logic_vector(7 downto 0);
CLIENTEMAC0TXDVLD      : in   std_logic;
EMAC0CLIENTTXACK        : out  std_logic;
CLIENTEMAC0TXFIRSTBYTE : in   std_logic;
CLIENTEMAC0TXUNDERRUN  : in   std_logic;
EMAC0CLIENTTXCOLLISION : out  std_logic;
EMAC0CLIENTTXRETRANSMIT : out  std_logic;
```

Signály RXD a TXD obsahujú prenášané dáta a ich platnosť je určená príslušným DVLD signálom. Z prijímača sú vyvedené signály GOODFRAME a BADFRAME, z ktorých je práve jeden aktívny vždy súčasne s posledným bajtom daného paketu. FRAMEDROP je aktívny v prípade, že je paket určený na zahodenie.

Vysieláč vždy po odoslaní prvého bajtu paketu čaká na potvrdenie ACK a až potom odosiela zvyšné bajty paketu. Signál FIRSTBYTE by mal byť stále na nízkej úrovni [5]. UNDERRUN slúži ako odkaz pre Ethernet MAC, aby daný paket znehodnotil. COLLISION a RETRANSMIT signalizujú kolíziu pri pokuse o prenos a požiadavku na znovuodoslanie paketu (neplatné pre duplexný mód).

### 4.2.2 Stavový automat

Navrhovaný stavový automat riadi zápis a čítanie z pamäti DDR2. Jeho základom sú vstupná a výstupná pamäť RAM, ktoré pracujú na princípe FIFO. Ich hlavnou úlohou je zmena šírky slova z 10b (8b dáta + sof + eof) na 80b a naopak kvôli efektívnejšiemu využitiu DDR2 pamäte (bity 80 až 127 sú pri zápise doplnené nulami). Pri generovaní IP jadier (Obrázok 4.4) som využil možnosť nezávislých hodinových signálov pre zápis a čítanie, čo umožnilo zvoliť pre tieto dve operácie aj rôzne šírky slova. Maximálny počet uložených 10b slov som ponechal na 1024, čo považujem za dostatočnú hodnotu vzhľadom na fakt, že tieto FIFO pamäte slúžia hlavne na optimalizovanie šírky slova a dáta by sa v nich preto nemali hromadiť.



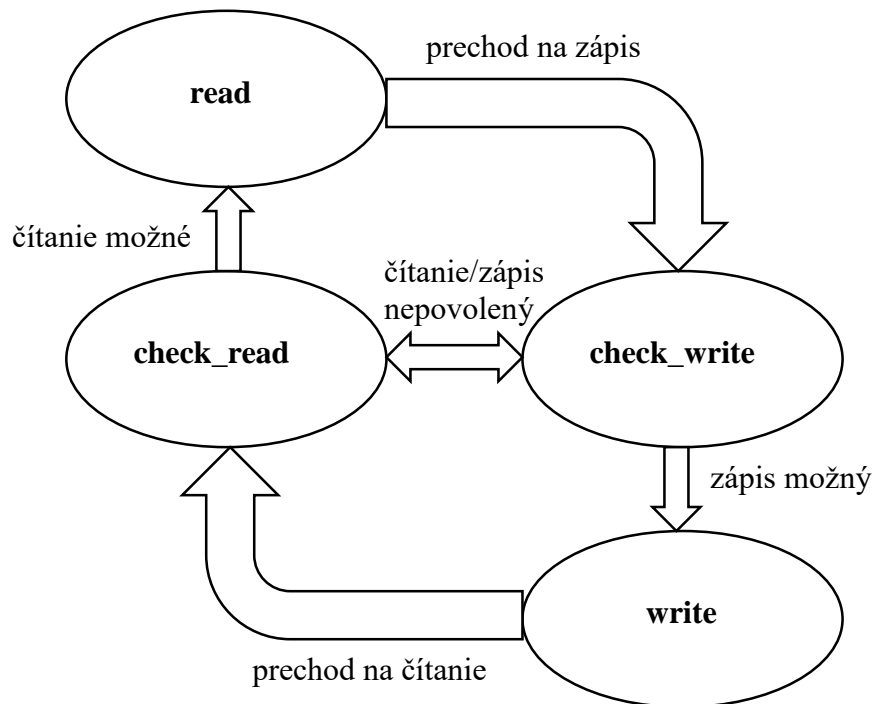
Obrázok 4.4: generovanie IP jadra na vstupe do stavového automatu

Medzi dvoma FIFO pamäťami už pracuje samotný stavový automat (Obrázok 4.5). Jeho základným princípom je pravidelné striedanie operácii čítania a zápisu pri prístupe k DDR2 pamäti. Navrhol som preto 4 možné stavy:

- **check\_read** – Kontroluje sa, či je možný presun dát z DDR2 do výstupnej FIFO pamäte. Ak áno, nasleduje „read“ stav, v opačnom prípade „check\_write“.
- **check\_write** – Kontroluje sa, či je možný presun dát zo vstupnej FIFO pamäte do DDR2. Ak áno, nasleduje „write“ stav, v opačnom prípade „check\_read“.
- **read** – Prebieha čítanie dát z DDR2 pamäte a ich uloženie do výstupnej FIFO pamäte. Nasledujúci stav je „check\_write“.
- **write** – Prebieha zápis dát z vstupnej FIFO pamäte do DDR2 pamäte. Nasledujúci stav je „check\_read“.

Správne prepínanie medzi jednotlivými stavmi zabezpečuje prvá kombinačná časť na začiatku zdrojového kódu stavového automatu (viď. príloha B). Okrem súčasného stavu (*pres\_state*) je rozhodovanie o nasledujúcom stave podmienené signálmi *input\_empty* (prázdna vstupná FIFO pamäť), *ddr2\_full* (plná pamäť DDR2), *output\_full* (takmer plná výstupná pamäť FIFO) a *ddr2\_empty* (prázdna pamäť DDR2). Nasledujúci stav je následne aktivovaný vzostupnou hranou hodinového signálu (*clk\_main*).

Druhá kombinačná časť zabezpečuje funkčnosť súčasného stavu automatu. Dôležité sú tu predovšetkým informácie o adrese čítania (*rd\_addr*) a zápisu (*wr\_addr*). Stavový automat nesmie dovoliť čítanie z adresy DDR2 pamäte, do ktorej neboli pred tým uložené dáta. V prípade pretečenia adresy zápisu je nastavený pomocný signál *addr\_jump*, ktorý značí, že sa začína zapisovať znova od začiatku pamäte. Keď následne pretečie aj adresa čítania, je signál *addr\_jump* opäť vynulovaný.



Obrázok 4.5: grafické znázornenie navrhnutého stavového automatu

Správne inkrementovanie adries zápisu a čítania po vykonaní príslušnej operácie zabezpečujú signály `after_wr` a `after_rd`. K navýšeniu hodnoty adresy dohádza vždy v nasledujúcom stave, aby sa predišlo zápisu/čítania bunky s chybnou adresou.

### 4.2.3 Informácia o stave signálu

Pri prijímaní informácie o úrovni prijímaného optického signálu predpokladám, že do FPGA vstupuje už len jednobitová informácia, či má alebo nemá byť vysielanie zastavené. Vyhradil som pre túto informáciu MSB bit z prijatých dát po sériovej linke. Následne je na základe tohto bitu zastavené vysielanie na SFP rozhraní. Funkciu zastavovania vysielania, je možné vypnúť jednoduchým prepnutím DIP prepínača `GPIO_DIP(2)`. VHDL kód, ktorý toto zabezpečuje je nasledovný.

```

PROCESS (UART_Data_out, Buffer_active) BEGIN
  IF Buffer_active = '1' AND UART_Data_out(7) = '1' THEN
    STOP_tx <= '1';
  ELSE
    STOP_tx <= '0';
  END IF;
END PROCESS;

```

## 5 ZÁVER

V prvých dvoch kapitolách tejto diplomovej práce som popísal základné vlastnosti optických bezdrôtových spojov, atmosférického prenosového prostredia a tiež funkčnosť protokolu TCP/IP. V tretej kapitole som predstavil návrh časovej analýzy s generátorom náhodného priebehu prijímaného a vysielaného optického výkonu pre použitie v simulácií správania sa protokolu TCP/IP pri prechode OBS.

Z výsledkov vykonaných simulácií plynie, že fluktuácie optického výkonu vplyvom atmosférických turbulencií spôsobujú značné zníženie prenosovej rýchlosti. Simulácie tiež ukázali, že použitím vyrovnávacej pamäti na strane vysielateľa je možné obmedziť vplyv fluktuácií na prenosovú rýchlosť.

V štvrtej kapitole som predstavil návrh realizácie vyrovnávacej pamäte pomocou FPGA s využitím externej DDR2 pamäte osadenej na vývojovej doske ML505 od firmy Xilinx. Hlavnú funkčnosť tvorí stavový automat, ktorý preskladáva prijímané dáta do slov s väčšou šírkou a následne ich striedavo zapisuje a vyčíta zo spomínanej DDR2 pamäte systémom FIFO.

# LITERATÚRA

- [1] WILFERT, O.: *Optoelektronika. Skripta*. FEKT VUT v Brně, Brno, 2002.
- [2] KOLKA, Z.; BIOLKOVÁ, V.; BIOLEK, D.: *Model of FSO Path for Network Simulation*. In Proceedings of the 18th International Conference on Circuits, Systems, Communications and Computers (CSCC 2014). Santorini, Greece: Europment, 2014. s. 46-49. ISBN: 978-1-61804-245-3.
- [3] KOLKA, Z.: *Počítačové a komunikační sítě. Skripta*. FEKT VUT v Brně, Brno, 2007.
- [4] DOSTÁLEK, L.; KABELOVÁ, A.: *Velký průvodce protokoly TCP/IP a systémem DNS. 2. aktualiz. vyd.* Praha: Computer Press, 2000, 426 s. ISBN 80-722-6323-4.
- [5] Xilinx: Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC User Guide. 2011
- [6] KOLKA, Z.; BIOLKOVÁ, V.; BIOLEK, D. *Optimizing Link Layer of Terrestrial FSO Links Using Channel Reciprocity*. Proceedings of The International Conference on New Trends in Signal Processing 2014. L. Mikuláš, SR: AFCEA, 2014.s. 60-65.



## ZOZNAM SKRATIEK

|      |   |
|------|---|
| OBS  | optický bezdrôtový spoj   |
| APP  | atmosférické prenosové prostredie                                 |
| OOK  | ON-OFF keying (kľúčovanie)  |
| TCP  | transmission control protocol (protokol riadenia prenosu)         |
| IP   | internet protocol (internetový protokol)                          |
| TTL  | time to live (doba života paketu)                                 |
| PSI  | power scintillation index (index sily zvlňenia)                   |
| FIFO | first-in first-out  |
| SFP  | small form-factor pluggable (malý pripojiteľný vysielač/prijímač) |
| SOF  | start of frame (začiatok paketu)                                  |
| EOF  | end of frame (koniec paketu)                                      |

# A PRIEBEH SIMULÁCIE V OMNET++

The screenshot displays the OMNET++ simulation environment. At the top, the title bar reads "OMNET++ / [kenv - General #0 - omnetpp.ini - D:\omnetpp-4.5\samples\ISO\_01\simulations". The interface is divided into several panels:

- Network Diagram:** Shows a "client" node connected to a "server" node via a link labeled "ppp0" with IP addresses "10.0.0.1/30" and "10.0.0.2/30". A "configurator" node is also present.
- Event Log:** A list of events with columns for Time, Name, Src/Dest, and Info. The events include TCP and ACK packets between IP addresses 10.0.0.1 and 10.0.0.2.
- Packet Capture:** A table of captured packets with columns for Time, Content, and other details. The captured content is a "controlInfo" field containing "controlInfo = NULL (cObjPtr)" and "encapsulatedPacket = (IP-dtogram) tpsseq(l=1452) (Pv4dkt)".
- Simulation Controls:** Includes buttons for "Simulate", "Inspect", "View", and "Help", along with various status indicators.

The event log shows a sequence of TCP connections and acknowledgments. For example, event #679 is a TCP packet from 10.0.0.1 to 10.0.0.2 with sequence number 10221 and window length 1452. Subsequent events show ACKs and further data transfers.

## B ZDROJOVÝ KÓD STAVOVÉHO AUTOMATU

```
-- STATE MACHINE -
-- Next state determination --
PROCESS (pres_state, input_empty, ddr2_full, output_afull, ddr2_empty) BEGIN
  CASE pres_state IS
    WHEN check_wr => IF input_empty = '0' AND ddr2_full = '0' THEN
                      next_state <= do_wr;
                    ELSE
                      next_state <= check_rd;
                    END IF;
    WHEN check_rd => IF output_afull = '0' AND ddr2_empty = '0' THEN
                      next_state <= do_rd;
                    ELSE
                      next_state <= check_wr;
                    END IF;
    WHEN do_wr => next_state <= check_rd;
    WHEN do_rd => next_state <= check_wr;
  END CASE;
END PROCESS;

-- Next state application --
PROCESS (clk_main) BEGIN
  IF rising_edge(clk_main) THEN
    pres_state <= next_state;
  END IF;
END PROCESS;

-- Present state functionality --
PROCESS (pres_state, after_rd, after_wr, rd_addr, wr_addr) BEGIN
  CASE pres_state IS
    WHEN check_wr => command <= X"0";
                    address <= STD_LOGIC_VECTOR(wr_addr);
                    data_en <= '0';
                    addr_en <= '0';
                    input_en <= '0';
                    output_en <= '0';
                    IF after_rd = '1' THEN
                      IF STD_LOGIC_VECTOR(rd_addr) = X"7FFFFFFF" THEN
                        IF STD_LOGIC_VECTOR(wr_addr) = X"00000000" THEN
                          ddr2_empty <= '1';
                        END IF;
                        rd_addr <= X"00000000";
                        addr_jump <= '0';
                      ELSE
                        IF STD_LOGIC_VECTOR(wr_addr) =
                          STD_LOGIC_VECTOR(rd_addr + 1) THEN
                          ddr2_empty <= '1';
                        END IF;
                        rd_addr <= rd_addr + 1;
                      END IF;
                    END IF;
    WHEN check_rd => command <= X"1";
                    address <= STD_LOGIC_VECTOR(rd_addr);
                    data_en <= '0';
                    addr_en <= '0';
                    input_en <= '0';
                    output_en <= '0';
                    IF after_wr = '1' THEN
                      IF STD_LOGIC_VECTOR(wr_addr) = X"7FFFFFFF" THEN
                        IF STD_LOGIC_VECTOR(rd_addr) = X"00000000" THEN
                          ddr2_full <= '1';
                          addr_jump <= '0';
                        ELSE
                          addr_jump <= '1';
                        END IF;
                      ELSE
                        wr_addr <= X"00000000";
                      END IF;
                    IF STD_LOGIC_VECTOR(rd_addr) =
```

```

                                STD_LOGIC_VECTOR(wr_addr + 1) THEN
                                ddr2_full <= '1';
                                END IF;
                                wr_addr <= wr_addr + 1;
                                END IF;
                                END IF;
                                WHEN do_wr =>
                                addr_en <= '1';
                                data_en <= '1';
                                input_en <= '1';
                                data_ddr2_in <= data_fsm & (OTHERS => '0');
                                after_wr <= '1';
                                WHEN do_rd =>
                                addr_en <= '1';
                                output_en <= '1';
                                data_fsm <= data_ddr2_out(79 downto 0);
                                after_rd <= '1';
                                END CASE;
                                END PROCESS;

```