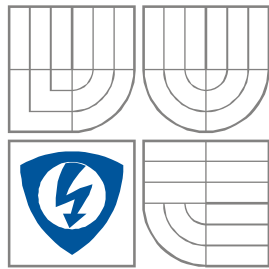


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

KOMUNIKAČNÉ PROTOKOLY PRE OPTICKÉ BEZDRÁTOVÉ SPOJE

COMMUNICATION PROTOCOLS FOR FREE-SPACE OPTICAL LINKS

DIPLOMOVÁ PRÁCA
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Ondrej Šedo

VEDÚCI PRÁCE
SUPERVISOR

prof. Dr. Ing. Zdeněk Kolka

BRNO, 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Ondřej Šedo

ID: 125653

Ročník: 2

Akademický rok: 2014/2015

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: 9.2.2015

Termín odevzdání: 21.5.2015

Vedoucí práce: prof. Dr. Ing. Zdeněk Kolka

Konzultanti diplomové práce:

doc. Ing. Tomáš Kratochvíl, Ph.D.

Předseda oborové rady

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, 2015. 22 s., 5 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

Zoznam obrázkov	viii
Zoznam tabuliek	ix
Úvod	1
1 Optické bezdrôtové spoje	2
1.1 Skladba optického bezdrôtového spoja.....	2
1.1.1 Vysielačový systém OBS	2
1.1.2 Prijímačový systém OBS	2
1.2 Energetická bilancia optického spoja	3
1.3 Atmosférické prenosové prostredie	4
2 Protokoly TCP/IP	6
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
3 Model OBS pre simuláciu	10
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.....	12
3.2.1 Generovanie okamžitej bitovej chybovosti.....	12
3.2.2 Implementácia do OMNeT++	12
3.3 Simulácia	13
4 Realizácia vyrovnávacej pamäte pomocou FPGA	15

4.1	Návrh vyrovnávacej pamäte vo vysieläči	15
4.2	Programovanie FPGA modulu.....	15
4.2.1	Vývojová doska	15
4.2.2	Obsluha sieťových konektorov	16
4.2.3	Prepojenie sieťových konektorov	17
4.2.4	FIFO na vysieläči SFP	17
4.2.5	Informácie o stave optického signálu	18
4.3	Testovanie funkčnosti	19
5	Záver	20
	Literatúra	21
	Zoznam skratiek	22
A	OMNeT++ simulátor	23
A.1	Zdrojový kód generátoru aktuálnej bitovej chybovosti	23
A.2	Priebeh simulácie	25
B	Konfigurácia FPGA	26
B.1	Zapojenie (PORT MAP) RAM blokov v FIFO pamäti	26
B.2	Zoznam portov eth_fifo_8	27

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 α_T	12
Obrázok 3.3: výsledky simulácie (bez vyrovnávacej pamäti)	13
Obrázok 3.4: výsledky simulácie (s vyrovnávacou pamäťou).....	14
Obrázok 4.1: Znázornenie vyrovnávacej pamäte	15
Obrázok 4.2: Vývojová doska ML505	16
Obrázok 4.3: Generovaný RAM blok.....	18

ZOZNAM TABULIEK

Tabuľka 1.1: Tabuľka stavov APP na základe V_M	4
Tabuľka 4.1: Test funkčnosti vyrovnávacej pamäte	19

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 simulátora OMNeT++, kde prebieha simulácia sieťovej komunikácie. Na základe výsledkov simulácií je navrhnutá vyrovnávacia pamäť umiestnená vo vysielači. Tá má za úlohu pozastaviť vysielanie v prípade výpadku linky, aby sa zabránilo strate prenášaných paketov a ich následnému nutnému opakovanému vysielaniu. 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

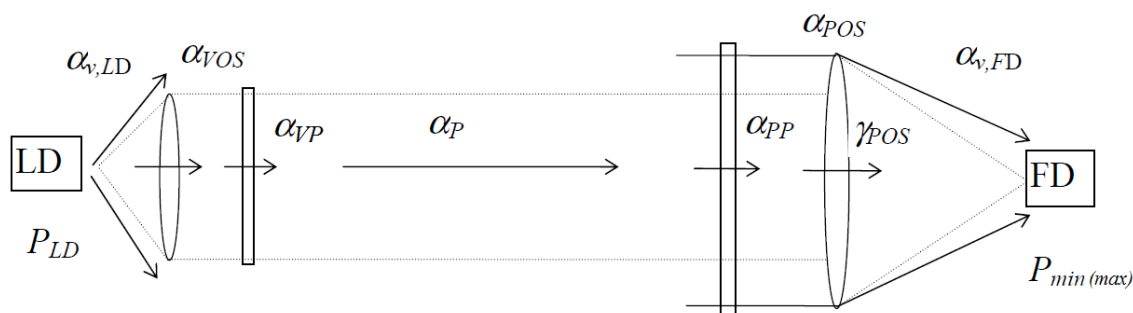
Prijímací systém OBS zabezpečuje príjem a demoduláciu signálu. Skladá sa z prijímacej optickej sústavy, fotodiódy, predzosilňovača a demodulátoru. Zameriavanie prijímacieho systému prebieha rovnako ako pri vysielacom systéme a na ochranu pred nečistotami je aj tu použitý optický priezor.

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 α_{VOS} a α_{POS} , priepustnosť optických priezorov α_{VP} a α_{PP} , útlm šírením α_P , zosilnenie prijímacej optickej sústavy γ_{POS} , účinnosť väzby prijímacej optickej sústavy a fotodiódy $\alpha_{v,FD}$, útlm vyvolaný nedokonalosťou vzájomného zamierenia hlavíc α_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 škvŕny, ktorú v ohniskovej rovine prijímacej optickej sústavy vytvára prijatý optický signál. Pokiaľ je táto škvŕna 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 Δ_P , ktorá sa vypočíta nasledujúcim vzťahom:

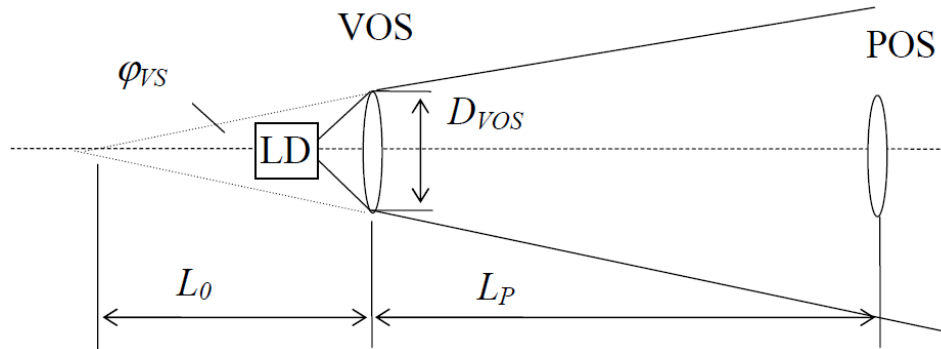
$$\Delta_P = P_{max} - P_0. \quad (1.1)$$

Útlm α_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 α 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 α pre ľubovoľnú dĺžku vlny sa vypočíta zo vzťahu

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

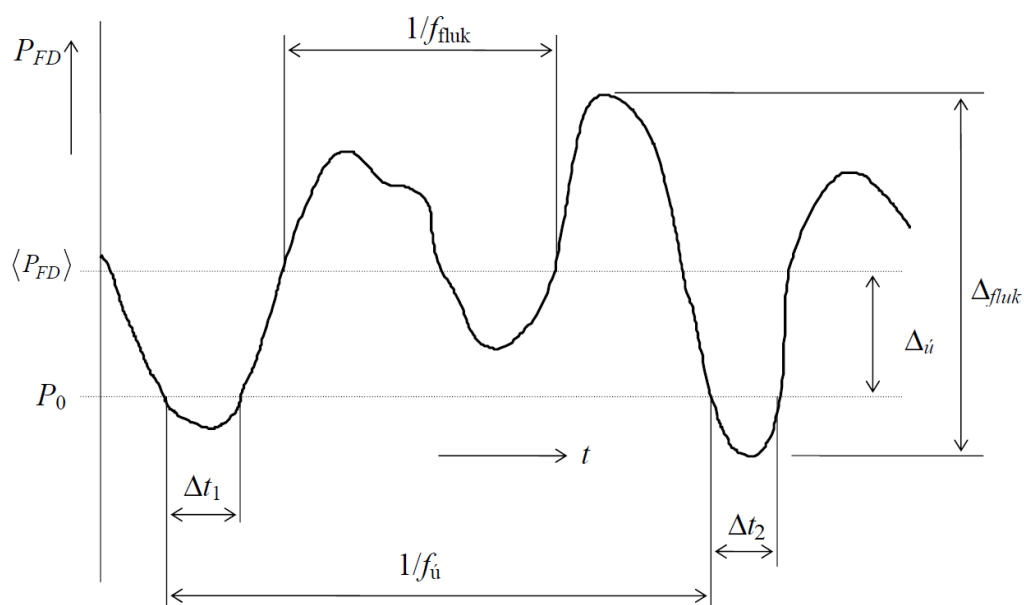
kde $q = 0,585V_M^{\frac{1}{3}}$ pre $V_M \leq 6$ km [1]. Podľa meteorologickej viditeľnosti a koeficientu extinkcie rozlišujeme rôzne stavy atmosféry (viď Tabuľka 1.1).

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

V_M [km]	α [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.

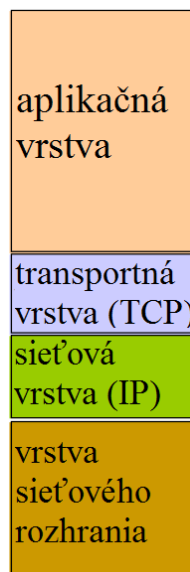


Obrázok 1.3: parametre fluktuácií a únikov [1]
 (P_{FD} – optický výkon na fotodióde prijímača, P_0 – citlivosť prijímača,
 f_{fluk} – frekvencia fluktuácií, $f_{\dot{u}}$ – frekvencia únikov, Δ_{fluk} – dynamika fluktuácií,
 $\Delta_{\dot{u}}$ – hĺbka únikov, t – čas, Δt_i – časové intervaly únikov)

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**, **príznamy** 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)			príznamy (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 **príznamov** záhlavia môžu byť nastavené nasledujúce príznaky:

- URG – TCP segment obsahuje nalievavé 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 – odosielať začína novú sekvenciu číslovania.
- FIN – odosielať 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, dopĺňa 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

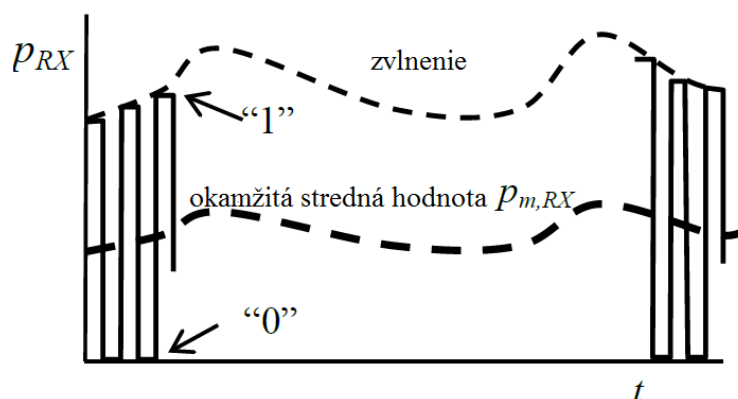
3.1.1 Prijatý optický výkon

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, α_{FSL} je bežný útlm spôsobený prechodom APP, α_{ATM} je absorpčný útlm nehostinnej atmosféry (hmla, dážď a pod.) a α_T reprezentuje efekt spôsobený atmosférickými turbulenciami. Koeficient α_{FSL} je pre daný OBS konštantný. Hodnota útlmu α_{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 α_T s určenou strednou hodnotou a rozptylom σ_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

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

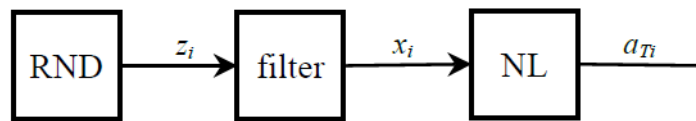
kde τ_0 je korelačná doba prenosového kanálu a koeficienty $a = 0,5$ a $b = 1$ pre Gauss-Markov model [2].

Zo vzniknutej postupnosti x_i sú nelineárnou transformáciou počítané koeficienty α_{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 σ_L^2 je logaritmický rozptyl vzťahnutý k PSI.



Obrázok 3.2: generátor náhodných vzoriek koeficientov α_T [2]

Pri implementácii takéhoto 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 čísel z_i a generovať vždy príslušný počet čísel podľa uplynutého času medzi dvoma paketmi.

3.2 Implementácia do simulátora

3.2.1 Generovanie okamžitej bitovej chybovosti

Generovanie náhodnej postupnosti z_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 vygenerovaná postupnosť je uložená do poľa $z[2*N+1]$, kde N predstavuje dĺžku odozvy FIR filtru. Toto pole následne prechádza spomínaným lineárnym filtrom, daným autokorelačnou funkciou zo vzťahu (3.7). Nakoniec je s využitím vzťahov (3.9), (3.8) a (3.5) vypočítaná okamžitá hodnota bitovej chybovosti.

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

3.2.2 Implementácia do OMNeT++

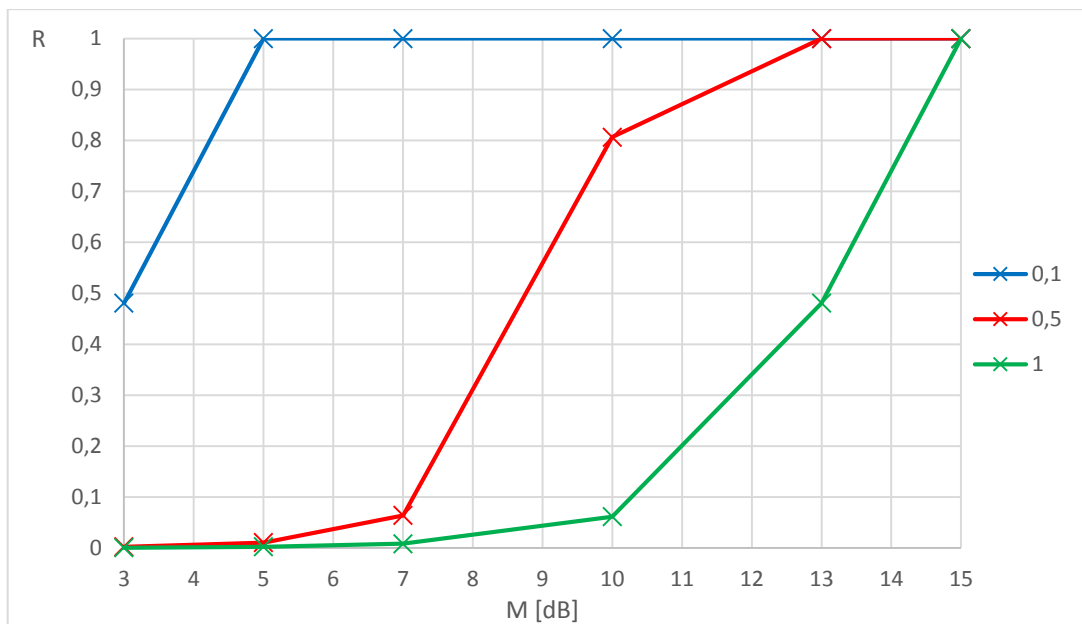
Pri prichádzajúcom pakete simulátor z vygenerovanej bitovej chybovosti vypočíta podľa vzťahu (3.6) pravdepodobnosť výskytu chyby v pakete a túto hodnotu porovnáva s náhodným číslom z intervalu $[0, 1)$. Na základe tohto porovnania sa rozhoduje, či bude

daný paket zahodený.

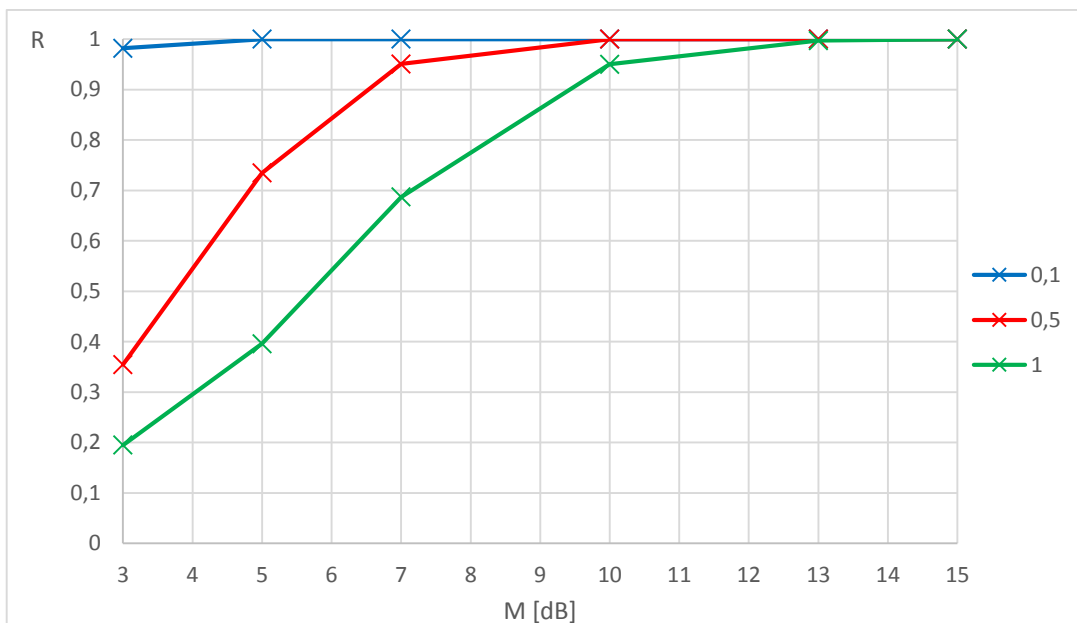
Pre porovnanie som do simulátora OMNeT++ implementoval aj možnosť ochrany pred náhlím vzrastom bitovej chybovosti vo forme fiktívnej vyrovnávacej pamäte vo vysielacom. Ak hodnota generovanej bitovej chybovosti prekročí stanovenú hranicu, simulátor ďalej generuje postupne hodnoty chybovosti, až kým nedôjde k jej opätovnému poklesu. Zároveň je s procesom generácie bitovej chybovosti počítaný počet uplynutých vzorkovacích období a tento čas je následne pripočítaný k dobe prenosu paketu. Tým je simulovaná vyrovnávacia pamäť vo vysielacom, ktorá zachytáva pakety po dobu zvýšenej hodnoty bitovej chybovosti nad stanovenú úroveň. Zdrojový kód implementácie premenlivej chybovosti je uvedený v prílohe A.1.

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 3.3 a Obrázok 3.4). Na x-ovej osi sú hodnoty linkovej rezervy a na osi y priemerná prenosová rýchlosť vzťahnutá 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 silné.



Obrázok 3.3: výsledky simulácie (bez vyrovnávacej pamäti)



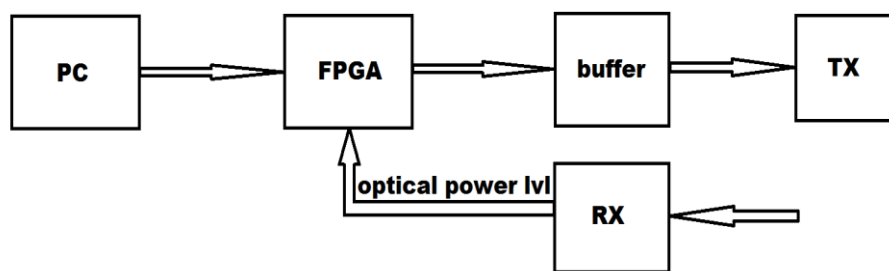
Obrázok 3.4: výsledky simulácie (s vyrovnávacou pamäťou)

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 A.2.

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 optickeho vysieläča. Keď dôjde vplyvom atmosférických turbulencií k úniku optickeho signálu, pozastaví sa vysielanie a dátové pakety prichádzajúce do optickeho 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 optickeho signálu 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 optickej signál nie je totožný so signálom vysielaným, no vzhľadom na vysokú úroveň vzájomnej korelácie (95% až 99%) ich môžeme považovať za rovnaké.



Obrázok 4.1: Znázornenie vyrovnávacej pamäte

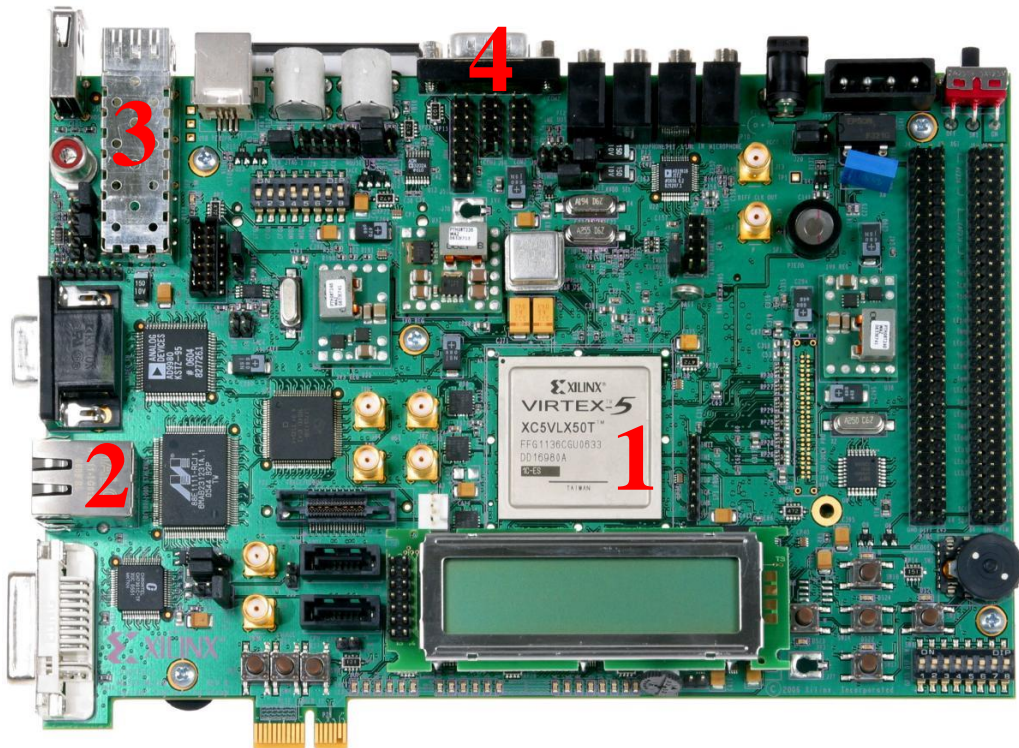
Zo schémy na obrázku 4.1 je zrejmé, že cez vyrovnávaciu pamäť prechádzajú dátové pakety aj v čase, kedy nedošlo k výpadku. Týmto sa jednoduchšie zabezpečí, aby nebolo pozastavené vysielanie uprostred paketu, pretože vo vyrovnávacej pamäti sú uložené informácie o presnom začiatku a konci každého prenášaného paketu.

4.2 Programovanie FPGA modulu

4.2.1 Vývojová doska

Na realizáciu metódy vyrovnávacej pamäti 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 som využil

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 šesťdesiat 36kb blokov RAM pamäte, ktoré možno spojiť do jedného bloku RAM o veľkosti 2160kb.

4.2.2 Obsluha sieťových konektorov

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 vysielачa.

```
-- 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.

Vysielač 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 (neplatí pre duplexný mód).

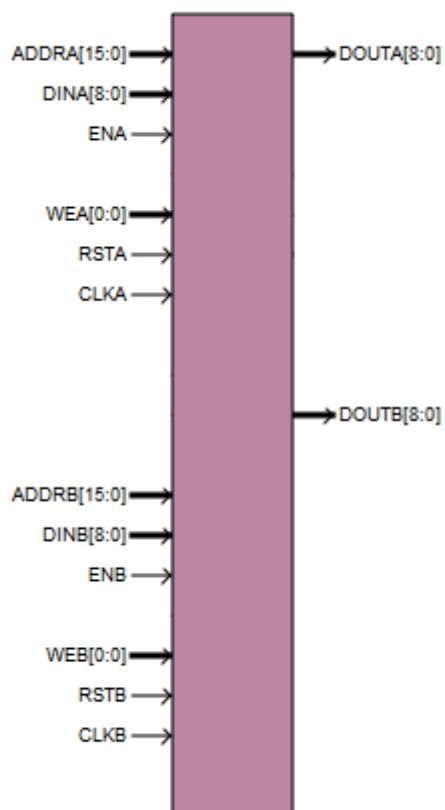
4.2.3 Prepojenie siet'ových konektorov

Na obsluhu prepojenia ethernet konektoru s SFP modulom som využil komponent eth_fifo_8 (zoznam portov uvedený v prílohe B.2), ktorý obsahuje 4096B fifo pamäť na prijímači aj vysielači. Každý prechádzajúci paket má pridelené príznakové bity sof (začiatok paketu) a eof (koniec paketu) a je uložený v pamäti vysielača ešte aj po odoslaní, kým neuplynie čas na požiadavku o znovuodoslanie. Taktiež sú tu zahadzované neúplné pakety, prípadne pakety s nastaveným BADFRAME alebo FRAMEDROP.

Každá 4096B fifo pamäť je tvorená z dvoch 2048B blokov. Šírka slova v pamäti je 8 bitov plus jeden paritný bit, ktorý tu však slúži na označenie eof. Mieru naplnenia pamäti je možné sledovať prostredníctvom 4-bitového signálu wr_fifo_status v šestnástinách celkovej veľkosti fifo pamäti.

4.2.4 FIFO na vysielači SFP

Aby vyrovnávací pamäť splňala funkciu ochrany pred výpadkami optickej linky vplyvom atmosférických turbulencií, bolo nutné 4096B FIFO pamäť na vysielači nahradiť väčšou. Každý z použitých 2048B blokov RAM som nahradil vygenerovaným IP jadrom Block Memory Generator (Obrázok 4.3). To som vygeneroval podľa pôvodných RAM blokov. Generovaný RAM blok má šírku slova 9b (horných 8 bitov sú dáta a spodný bit je eof) a hĺbku 65536 slov. Celkovú FIFO pamäť tvoria dva takéto bloky, takže výsledná veľkosť vyrovnávacej pamäte je 128kB. Každý RAM blok má šírku adresy 16 bitov. Celkovo má však FIFO pamäť šírku adresy 17 bitov. Najvyšší bit preto rozhoduje, do ktorého z dvoch RAM blokov sa budú dáta zapisovať. Kompletne pripojenie blokov RAM je uvedené v prílohe B.1.



Obrázok 4.3: Generovaný RAM blok

Port A je používaný na zapisovanie do pamäte zatiaľ čo port B na čítanie. Tým je zabezpečená možnosť rozdielneho hodinového signálu pre čítanie a zapisovanie.

Obsluha FIFO je napísaná tak, že nie je možné čítať z pamäte paket, ktorý v nej ešte nie je uložený celý. Týmto mechanizmom sa predchádza začatiu odosielaniu paketov, ktoré majú nastavený napríklad BADFRAME príznakový bit.

4.2.5 Informácie o stave optického signálu

Informáciu o úrovni optického signálu je možné poslať 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 nulu, 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 (možná hysterezia) je vysielanie znova povolené a z vyrovnávacej pamäti je možné čítať dáta.

Pri testovaní funkčnosti je možné generovať náhodný priebeh prijímaného optického výkonu rovnako ako je popísané v kapitole 3 a tieto údaje poslať do FPGA. Ja som pre

potrebu rýchleho testovania priradil nastavovanie tx_enable bitu priamo na tlačidlo, ako vidno v nasledujúcom úryvku z VHDL kódu. Následne som krátkymi stlačeniami simuloval výpadky spojenia.

```
-- Ethernet FIFO 8 (from PHY to SFP)
PHYtoSFP_eth_fifo_8: eth_fifo_8_big PORT MAP (
    tx_clk           => CLK_TX_SFP,
    tx_reset        => '0',
    tx_enable       => NOT GPIO_BTN(0),
```

4.3 Testovanie funkčnosti

Pre otestovanie funkčnosti VHDL kódu som pripojil vývojovú dosku medzi PC a bod pripojenia do siete LAN. Pripojenie bolo smerované tak, aby vysielač na SFP module posielal dáta smerom do PC. Z pohľadu PC užívateľa išlo o testovanie v smere download. Pripojenie do siete bolo obmedzené na maximálnu rýchlosť 100Mb/s.

Z umiestnenia na lokálnej sieti som následne sťahoval 1,2GB veľký súbor na lokálny disk PC. Celkovo som sťahovanie zopakoval trikrát. Prvý raz bez simulácie výpadkov, druhýkrát s výpadkami a nastavenou vyrovnávacou pamäťou a nakoniec s výpadkami a bez vyrovnávacej pamäti. Výsledky sú uvedené v nasledujúcej tabuľke.

Tabuľka 4.1: Test funkčnosti vyrovnávacej pamäte

Charakter spoja	Priemerná rýchlosť
bez výpadkov	8,1 MB/s
s výpadkami + FIFO	7,8 MB/s
s výpadkami	6,4 MB/s

Z výsledkov testu je zrejmé, že konfigurácia s implementovanou vyrovnávacou pamäťou dosahuje vyššie prenosové rýchlosti pri simulovaných výpadkoch spojenia ako konfigurácia bez vyrovnávacej pamäte.

5 ZÁVER

V prvých dvoch kapitolách tejto semestrálnej 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 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 a realizáciu vyrovnávacej pamäti pomocou FPGA. Test funkčnosti v prípade simulácie výpadkov linky preukázal nárast prenosovej rýchlosti so zaradenou vyrovnávacou pamäťou. Pre dosiahnutie lepších výsledkov aj pri vyšších prenosových rýchlostiach by bolo vhodné zväčšiť veľkosť vyrovnávacej pamäte, napríklad využitím externej DDR2 RAM pamäte.

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: Euroment, 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

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 zvlnenia)
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 OMNET++ SIMULÁTOR

A.1 Zdrojový kód generátoru aktuálnej bitovej chybovosti

```
static simtime_t oldt = 0; //last packet finish time
const int N = 32; //FIR response
double tau = 5e-3; //Correlation time
double Ts = tau/50; //sample period
double para = 0.5; //parameters of autocorrelation function
double parb = 1;
double PSI = 1; //scintillation index
static double z[2*N+1]; //random generated numbers
double x; //z after FIR
double at; //atmospheric attenuation (x after NL)
double time[2*N+1]; //time for autocorrelation function
double R[2*N+1]; //autocorrelation function
double pm0=-7.0345; //normal inverse cdf of PBE0=1e-12
double MdB=5; //link fade margin [dB]
double M=pow(10, MdB/10); //link fade margin absolute
double ber = 1e-6; //bit error rate
// bit error modeling
if ((ber > 0.0) && msg->isPacket())
{
    cPacket *pkt = (cPacket *)msg;
    double ksig = 0;
    double sigma = 0;
    for (int i=(-2*N-1); i<=(2*N+1); i++) //computing time for R
        time[i+2*N+1] = i*Ts;
    for (int i=0; i<(2*N+1); i++) //computing autocorrelation function
    {
        double ti;
        if (time[i]<0) //absolute value for double
            ti = -1*time[i];
        else
            ti = time[i];
        R[i] = exp(-1*para*pow(ti/tau, parb));
        ksig = ksig + (R[i]*R[i]);
    }
    ksig = sqrt(ksig);
    sigma = sqrt(log(PSI+1))/ksig;
    if ((oldt == 0) || (t-oldt > ((Ts)*(2*N+1))))
    {
        for (int i=0; i<(2*N+1); i++)
            z[i] = normal(0, 1);
        x = 0;
        for (int j=2*N; j>=0; j--)
            x = x + R[2*N-j]*z[j]; //FIR filter
    }
    else
    {
        int cnt=1;
        while ((t-oldt)>Ts) //number of Ts elapsed
        {
            oldt = oldt - Ts;

```



```

        cnt++;
    }
    for (int i=0; i<cnt; i++)
    {
        for(int j=0; j<(2*N); j++)
            z[j] = z[j+1];
        z[2*N] = normal(0, 1);
        x = 0;
        for (int k=2*N; k>=0; k--)
            x = x + R[2*N-k]*z[k];
    }
}
at = exp(x*sigma - (pow(sqrt(log(PSI+1)), 2)/2)); //atm. attenuation
ber = normCFD(M*at*pm0); //calculating bit error rate
if (dblrand() < 1.0 - pow(1.0-ber, (double)pkt->getBitLength()))
{
    pkt->setBitError(true);
}

```

A.2 Priebeh simulácie

The screenshot displays the OMNET++ simulation environment. At the top, the title bar reads "OMNET++/Tkeniv - General #0 - omnetpp.ini - D:\omnetpp-4.5\samples\ISO_01\simulations". The main window is divided into several sections:

- Network Topology:** A central diagram shows a network with nodes labeled "client", "server", "configurator", and "server". IP addresses are assigned to each: "10.0.0.1/30" for the client, "10.0.0.2/30" for the server, and "10.0.0.1/30" for the server. A "link" connects the client and server.
- Event Log:** A table listing simulation events with columns for "Event#", "Time", "Name", "Src/Dest", and "Info". The events are primarily TCP connections and data transmissions between the client and server.
- Packet Details:** A detailed view of a selected packet (Event #879) showing its structure:
 - Protocol: (PPF)Frame (tcpseg(l=152))
 - Fields: Contents(1)
 - Control: info = I432 (PPF)Frame
 - Encapsulated: encapsulatedPacket = (IPv4)DataGram (tcpseg(l=1452) (Packet))
 - Base: message
 - Packet: packet
 - Standing: standing
- Simulation Controls:** Buttons for "File", "Simulate", "Inspect", "View", and "Help" are visible at the top left. A status bar at the bottom indicates "Next: tcpseg(l=152) (TCP segment, id=9396)".

B KONFIGURÁCIA FPGA

B.1 Zapojenie (PORT MAP) RAM blokov v FIFO pamäti

```
GND(0) <= '0';
VCC <= '1';
GND_BUS <= (others => '0');

RAM_block_l : BRAM_SFP_TX -- lower addresses
PORT MAP (
  clka => wr_clk, -- write clock
  rsta => wr_sreset, -- write reset
  ena => VCC, -- port A enable
  wea => wr_en_l, -- write enable
  addra => std_logic_vector(wr_addr(15 downto 0)), -- address A
  dina => data_in, -- data input A
  douta => open, -- data output A
  clkb => rd_clk, -- read clock
  rstb => rd_sreset, -- read reset
  enb => rd_en_bram, -- port B enable
  web => GND, -- port B write enable
  addrb => std_logic_vector(rd_addr(15 downto 0)), -- address B
  dinb => GND_BUS(8 downto 0), -- data input B
  doutb => data_out_l -- data output B
);

RAM_block_u : BRAM_SFP_TX -- upper addresses
PORT MAP (
  clka => wr_clk,
  rsta => wr_sreset,
  ena => VCC,
  wea => wr_en_u,
  addra => std_logic_vector(wr_addr(15 downto 0)),
  dina => data_in,
  douta => open,
  clkb => rd_clk,
  rstb => rd_sreset,
  enb => rd_en_bram,
  web => GND,
  addrb => std_logic_vector(rd_addr(15 downto 0)),
  dinb => GND_BUS(8 downto 0),
  doutb => data_out_u
);

data_in <= wr_data_bram & wr_eof_bram; -- data + eof input
rd_data_bram_l <= data_out_l(8 downto 1); -- lower data output
rd_eof_bram_l <= data_out_l(0 downto 0); -- lower data eof
rd_data_bram_u <= data_out_u(8 downto 1); -- upper data output
rd_eof_bram_u <= data_out_u(0 downto 0); -- upper data eof
```

B.2 Zoznam portov eth_fifo_8

```
-- Transmit FIFO MAC TX Interface
tx_clk      : in  std_logic; -- MAC transmit clock
tx_reset    : in  std_logic; -- Synchronous reset (tx_clk)
tx_enable   : in  std_logic; -- Clock enable for tx_clk
tx_data     : out std_logic_vector(7 downto 0); -- Data to MAC
                                                    transmitter
tx_data_valid : out std_logic; -- Valid signal to MAC transmitter
tx_ack      : in  std_logic; -- Ack signal from MAC transmitter
tx_underrun : out std_logic; -- Underrun signal to MAC transmitter
tx_collision : in  std_logic; -- Collision sig from MAC transmitter
tx_retransmit : in  std_logic; -- Retransmit sig from MAC transmitter

-- Transmit FIFO Local-link Interface
tx_ll_clock  : in  std_logic; -- Local link write clock
tx_ll_reset  : in  std_logic; -- synchronous reset (tx_ll_clock)
tx_ll_data_in : in  std_logic_vector(7 downto 0); -- Data to Tx FIFO
tx_ll_sof_in_n : in  std_logic; -- sof indicator to FIFO
tx_ll_eof_in_n : in  std_logic; -- eof indicator to FIFO
tx_ll_src_rdy_in_n : in  std_logic; -- src ready indicator to FIFO
tx_ll_dst_rdy_out_n : out std_logic; -- dst ready indicator from FIFO
tx_fifo_status : out std_logic_vector(3 downto 0); -- FIFO
                                                    memory status
tx_overflow   : out std_logic; -- FIFO overflow indicator from FIFO

-- Receive FIFO MAC RX Interface
rx_clk      : in  std_logic; -- MAC receive clock
rx_reset    : in  std_logic; -- Synchronous reset (rx_clk)
rx_enable   : in  std_logic; -- Clock enable for rx_clk
rx_data     : in  std_logic_vector(7 downto 0); -- Data from MAC
                                                    receiver
rx_data_valid : in  std_logic; -- Valid signal from MAC receiver
rx_good_frame : in  std_logic; -- Good frame indicator from MAC
                                                    receiver
rx_bad_frame : in  std_logic; -- Bad frame indicator from MAC
receiver
rx_overflow   : out std_logic; -- FIFO overflow indicator from FIFO

-- Receive FIFO Local-link Interface
rx_ll_clock  : in  std_logic; -- Local link read clock
rx_ll_reset  : in  std_logic; -- synchronous reset
                                                    (rx_ll_clock)
rx_ll_data_out : out std_logic_vector(7 downto 0); -- Data from
                                                    Rx FIFO
rx_ll_sof_out_n : out std_logic; -- sof indicator from FIFO
rx_ll_eof_out_n : out std_logic; -- eof indicator from FIFO
rx_ll_src_rdy_out_n : out std_logic; -- src ready indicator from FIFO
rx_ll_dst_rdy_in_n : in  std_logic; -- dst ready indicator to FIFO
rx_fifo_status : out std_logic_vector(3 downto 0) -- FIFO memory
                                                    status
```