

VĚDECKÉ SPISY VYSOKÉHO UČENÍ TECHNICKÉHO V BRNĚ

Edice PhD Thesis, sv. 775

ISSN 1213-4198

thesis
?
IS

Ing. Michal Polívka

**Kvalita služby
v konvergovaných systémech
s prvky řízenými neuronovou sítí**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

Ing. Michal Polívka

**KVALITA SLUŽBY V KONVERGOVANÝCH SYSTÉMECH
S PRVKY ŘÍZENÝMI NEURONOVOU SÍTÍ**

QUALITY OF SERVICE IN CONVERGED SYSTEMS
WITH ELEMENTS CONTROLLED BY NEURAL NETWORK

Zkrácená verze Ph.D. Thesis

Obor: Teleinformatika
Školitel: doc. Ing. Vladislav Škorpil, CSc..
Oponenti: doc. Ing. Jiří Sýkora, CSc.
Ing. Václav Křepelka, Ph.D.
Datum obhajoby: 15. prosince 2014

Klíčová slova:

Přepínač, switch, neuronová síť, softwarové fronty, hardwarové fronty, MATLAB, Simulink, VNUML.

Keywords:

Switch, neural network, software queues, hardware queues, MATLAB, Simulink, VNUML.

Místo uložení práce:

Disertační práce je k dispozici na Vědeckém oddělení děkanátu FEKT VUT v Brně,
Technická 10, 61600 Brno

OBSAH

ÚVOD.....	5
1 PŘEHLED SOUČASNÉHO STAVU	6
1.1 Statistika síťového provozu.....	6
2 CÍLE DISERTAČNÍ PRÁCE	7
3 ANALÝZA PROBLÉMU	9
3.1 Architektura	10
3.1.1 <i>Struktura spojovacího pole</i>	10
3.2 Aplikace neuronových sítí na řízení aktivního prvku	10
3.2.1 <i>Volba Neuronové sítě</i>	10
3.2.2 <i>Komparační test neuronové sítě</i>	10
4 NÁVRH PŘEPÍNAČE	11
4.1 Simulační model VNUML.....	11
4.1.1 <i>Realizace</i>	12
4.1.2 <i>Zhodnocení</i>	14
4.2 Modelovaný protokol.....	14
4.2.1 <i>Struktura protokolu</i>	14
5 KOMPLEXNÍ SIMULAČNÍ MODEL	15
5.1 Popis modelu.....	15
5.2 Generátor paketů	16
5.3 Vstupní fronty	18
5.4 Spojovací pole s řízením	19
5.4.1 <i>Popis spojovacího pole</i>	19
5.4.2 <i>Řízení spojovacího pole</i>	19
5.5 Výstupní FIFO fronty.....	21
5.6 Výsledky a testování	21
ZÁVĚR	24
SEZNAM POUŽITÝCH ZDROJŮ	26
SEZNAM VLASTNÍCH PRACÍ.....	27
ŽIVOTOPIS	29
ABSTRACT.....	30

ÚVOD

Vytyčeným cílem předložené disertační práce je navrhnout architekturu nového síťového prvku zohledňujícího pravidla QoS v moderních konvergovaných sítích. Disertační práce volně navazuje na projekt GA ČR 102/07/1503 „Pokročilá optimalizace návrhu komunikačních systémů pomocí neuronových sítí“, který byl v roce 2009 úspěšně uzavřen.

Vytčeným cílem disertační práce je využít v návrhu architektury síťového prvku neuronových sítí. Neuronovou síť implementovat jako řídicí prvek. Síťovým prvkem se rozumí přepínač (switch) nebo směrovač (router).

Výzkum probíhající v rámci řešení disertační práce měl ambice navrhnout nový směr, který by mohl předestřít v oblasti síťové infrastruktury nové možnosti návrhu a řízení síťových prvků, jakými jsou mj. přepínače Cisco Nexus 9000, případně Catalyst 6800 nebo směrovač Cisco CRS-3, resp. Brocade MLX.

Záměr využít k řízení umělé neuronové sítě vychází především z výsledků zmíněného předchozího výzkumu. Použití neuronových sítí se osvědčilo v úlohách, kde je třeba rychle a efektivně navrhnout řešení komplikovaných problémů, při nichž nelze využít konvenční algoritmy.

Výkon a další parametry současných aktivních síťových prvků umožňuje pokrytí všech zásadních požadavků kladených na soudobé konvergované sítě. V některých případech se aktivní prvky obtížně vyrovnávají se stavy, kdy je síť přetížena provozem a pracuje na hranici své propustnosti. V zásadě ale platí, že problémy na sítích nejsou obvykle způsobeny nedostupností dostatečně výkonných síťových prvků, ale spíše chybami v návrhu nebo konfiguraci dostupných řešení.

Mechanismy QoS se v konvergovaných sítích uplatňují nejen ve stavech, kdy zatížení sítě dosáhne maxima propustnosti, ale také při běžném provozu. Nasazení mechanismů QoS i při běžném provozním zatížení je obvyklé jako prostředek dosažení co nejlepších parametrů sítě z pohledu zákazníka. Provoz přenášený historickou čistě analogovou telekomunikační sítí se neukládal ve vyrovnávacích pamětech, zpoždění v celém přenosovém řetězci bylo minimální. Koncepce konvergovaných sítí i jejich klíčových prvků je od původních telekomunikačních sítí odlišná. Datové sítě jsou levné a rychlost spojování je ve srovnání s klasickými telekomunikačními sítěmi velmi velká, ale připojená zařízení „soutěží“ o zdroje, tedy o možnost přenášet data.

V IP sítích, které jsou v různých modifikacích nejpoužívanější technologií transportních sítí, je přepínán, resp. spojován, každý jeden paket [1]. Spojení je ustanovováno pouze na přenesení tohoto jednoho paketu. Pokud v klasické telefonii trvá spojení řádově minuty, v případě datových sítí se jedná o několik řádů méně – dle rychlosti sítě a velikosti paketu, např. pro 100 Gb/s síť a 64B paket cca 5,12 ns.

Problematika QoS v IP sítích je shrnuta v kapitole 1, která kromě toho shrnuje dostupné zdroje literatury a celkově současný stav řešené problematiky.

1 PŘEHLED SOUČASNÉHO STAVU

V rámci řešení disertační práce byla prostudována dostupná literatura. Snahou bylo sumarizovat zejména poznatky vztažené k aktuálně platným normám, doporučením RFC, doporučením výrobců a také manuálům na trhu dostupných aktivních prvků se vztahem k problematice QoS. Dále pak shrnout podporu QoS na různých vrstvách RM ISO/OSI (ISO 7498) [2] počínaje 2. vrstvou.

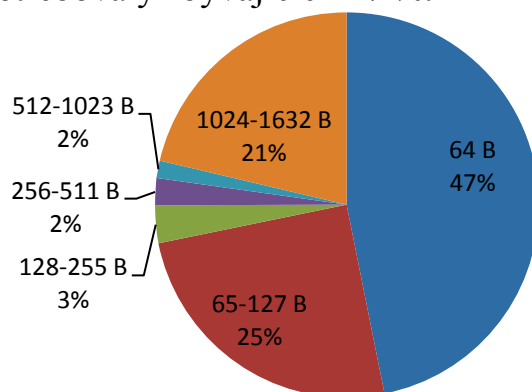
Pokud nebude uvedeno jinak, je veškerá popisovaná problematika vztažena k Ethernetu – IEEE 802.3 [3], ale často platí i pro ostatní paketové sítě.

Výrazné zaměření této kapitoly na aktivní prvky Cisco je dáno zejména jejich všeobecnou rozšířeností a možnostmi fungování popsaných algoritmů prakticky ověřit. Při výzkumu byly využity zejména prvky Cisco 2821, Cisco Catalyst 6500, Catalyst 3560E a Catalyst 2960G, které jsou výkonnostním a hierarchickým průřezem typické soudobé sítě. Popsané mechanismy jsou ale obecně známé a používané většinou současných výrobců síťových prvků. Po analytické stránce jsou popsány detailně v [4].

V rámci shrnutí současného stavu řešené problematiky byla provedena také měření aktivních prvků, publikovaná např. v [5], ale také statistiky provozu, na jejichž základě byl proveden návrh nového aktivního prvku, viz kap. 3–5.

1.1 STATISTIKA SÍŤOVÉHO PROVOZU

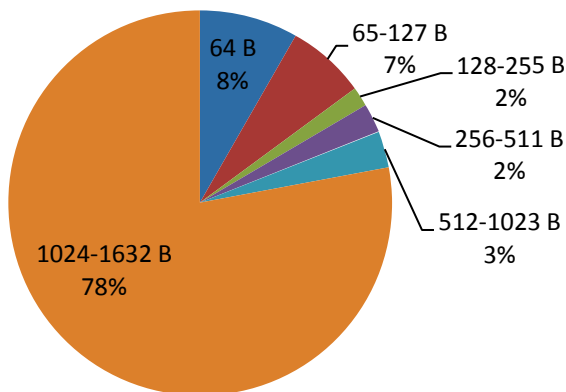
Jedním z měření, která byla v rámci vlastního výzkumu provedena, bylo měření zaměřené na sběr statistických údajů o síťovém provozu. Příklad rozložení velikostí rámců v síti 12 počítačů a 4 IP telefonů připojených k Internetu je zachycen na obr. 1.1. Měření bylo provedeno analytickými nástroji přepínače Cisco SGE 2010P a analyzátozem VeEX VePAL TX300/e na 1000Base-T Ethernetu. Přepínač tvořil střed hvězdicové topologie testované sítě. Měření bylo provedeno na propojovacím (trunk) portu, kterým byla síť připojena do Internetu. Měření probíhalo po dobu 24 dnů a 16 hodin kancelářské práce – prohlížení webových stránek, stahování PDF souborů atd. Protokol HTTP tvořil 83 % provozu, ostatní protokoly (RTP, RDP, FTP, DHCP,...) spotřebovaly zbývajících 17 %.



Obr. 1.1: Příklad rozložení velikostí rámců v síti

Unikátnost změřeného provozu, resp. zásadní odlišnost oproti „obvyklému“ internetovému provozu, podtrhuje potřebu obezřetnosti při konfiguraci hardwarových front. Během měření byla, po přepočtení na přenesená užitečná data, transportována velkými rámci velikosti 1 024–1 632 B většina (78 %) obsahu. Z toho neměřená část paketů přesahuje velikost 1 518 B (maximální velikost ethernetového rámce bez preamble a oddělovače). Jedná se o tzv. „jumbo“ rámce. V rámci internetového provozu jsou však obvykle nejpoužívanější rámce velikosti 512 B, které tvoří přibližně 90 % celkového provozu [6]. Jejich podíl na celkovém přenosu byl v testovacím případě zanedbatelný, viz obr. 1.2.

Pakety velikosti 64 B obvykle přenáší informace o signalizaci, případně jsou používány protokoly jako je SSH, Telnet apod. K přenosům souborů se používají velké pakety, obvykle 1 500 B. Pakety velikosti 128–512 B jsou v obecném internetovém provozu nejobvyklejší a pokrývají nejširší spektrum služeb. Rozdílnost výsledků nedokazuje chybnost závěrů uvedených v [6], ale poukazuje na to, že zásadní význam na charakteristiku provozu má lokalita, ve které je prováděno měření a to zejména s postupem směrem k nižším úrovním v hierarchii sítě.



Obr. 1.2: Rozložení velikostí rámců po přepočtení na přenesená data

Nad hardwarovými frontami nejsou prováděny žádné operace QoS. Proto příliš velká hardwarová fronta může snížit účinnost nastavených pravidel QoS realizovaných softwarovými frontami. Malá hardwarová fronta může být zase příčinou nárůstu ztrátovosti, protože v případě přetečení její kapacity je paket zahozen.

2 CÍLE DISERTAČNÍ PRÁCE

Primárním cílem, jenž stanovuje samotné téma práce, je návrh nového aktivního síťového prvku s novou strukturou a řízením neuronovou sítí, na rozdíl od běžně dostupných aktivních prvků řízených konvenčními algoritmy. Počítá se s integrovanou podporou QoS. Průnikem vytipovaných problémů aktivních prvků, viz dále, a tématu disertační práce jsou cíle disertační práce:

1. Analyzovat vybrané aktivní prvky – prepínače.
2. Zvolit vhodnou neuronovou síť pro řízení prepínače.
3. Navrhnout nový prepínač řízený neuronovou sítí s podporou QoS.
4. Navrhnout efektivně modelovatelný protokol.
5. Navržený prepínač modelovat a optimalizovat.

Inspirací pro stanovení cílů byly problémy identifikované při zpracování kap. 1 a po provedení měření, např. viz kap.1.1, a 4.1. Vyřešení těchto problémů by mohlo přispět k celkovému růstu oboru. Jedná se zejména o:

1. Problém chybné identifikace provozu a jeho zařazení do správného toku, zejména pokud se jedná o provoz koncových uživatelů (provoz různých ne-standardních VoIP a videokonferenčních systémů, časté využívání HTTP službami, pro které nebyl navržen).
2. Problém doručování paketů mimo pořadí.
3. Problém výkonnosti velkých spojovacích polí při plné zátěži.
4. Potřeba reklasifikovat provoz na hraně sítě.
5. Problematika bezpečnosti.
6. Problematika energetické spotřeby aktivních prvků.

Z vyjmenovaných problémů si tato disertační práce klade za cíl zabývat se zejména řešením problémů definovaných v bodech 1–3, na jejichž základě byly definovány cíle disertační práce 1–5.

Nezbytným prvním krokem je provést analýzu v současné době používaných způsobů řízení aktivních prvků, jejich výkonnosti, algoritmů pro správu softwarových front, které s řízením provozu na výstupu aktivního prvku přímo souvisí. Bylo také třeba stanovit požadavky služeb na QoS s důrazem na VoIP, ale také obecně v konvergovaných datových a telekomunikačních sítích. Dalším z cílů bylo shrnout klíčové mechanismy a protokoly ovlivňující kvalitu služby v IP sítích.

Druhým a třetím z cílů je navrhnout přepínač řízený neuronovou sítí. K tomu je třeba vybrat vhodnou neuronovou síť a analyzovat vhodnost jejího použití k řešení stanoveného problému. Využití umělých neuronových sítí, jako jednoho z nástrojů umělé inteligence, se osvědčilo při řešení úloh, u nichž je obtížné algoritmizovat rozhodovací proces kvůli vysoké variantnosti možných vstupů a výstupů. Cílem je využít umělou neuronovou síť, jako efektivnější alternativu k v současné době používaným algoritmům řízení aktivního prvku. Integrovat řízení neuronovou sítí spolu s konvenčními algoritmy pro dosažení maximální přesnosti a rychlosti při rozhodování o cíli paketu, resp. rámce. Zajistit minimální latenci, upřednostnit provoz s vyšší prioritou a zabránit strádání provozu s prioritou nižší. Je třeba stanovit, které části aktivního prvku je vhodné řídit konvenčními algoritmy a pro které části řízení se lépe hodí použití zvolené neuronové sítě a to s důrazem na QoS.

Čtvrtým a pátým – konečným – cílem je vytvořit nový komplexní simulační model síťového prvku vybaveného novým typem řízení. Součástí je návrh technologií, které umožní samotný výzkum aktivního prvku a také prověření dosažených výsledků. Aktivní prvek může být buď modelován čistě softwarově, nebo může být provedena hardwarová implementace. Dominantní je softwarový model, v prvních fázích byla prověřována i možnost hardwarové implementace, kterou se stručně zabývá kap. 4.1.

Jakkoli je v rámci současných konvergovaných datových/telekomunikačních sítí nejpoužívanější protokol IP, pro použití v modelech se příliš nehodí. Proto je cílem i návrh nového protokolu, který ponese klíčové části zmíněného IP protokolu, ale

bude pro modelování dostatečně jednoduchý a tedy umožní snadnou detekci případných chyb v navrhovaných strukturách a algoritmech. Současně by modely postavené nad novým protokolem měly umožňovat přenos poznatků i zpět do reálných sítí postavených nad IP protokolem. Důraz je dán zejména na implementaci polí se vztahem k průchodu paketu/rámce aktivním prvkem a také na implementaci polí se vztahem ke QoS.

Pro ověření navrhovaných řešení je třeba vytvořit model aktivního prvku umožňující sledovat zkoumané parametry. Pro model budou mj. navrženy nové generátory provozu – vytvořený model musí umožňovat paralelní generování paketů, tedy simulovat několik různých zdrojů síťového provozu přicházejícího na různé vstupní porty aktivního prvku. Dále musí umožnit individuální nastavení velikostí vstupních a výstupních zásobníků, modifikovat zpracování průchozího provozu ve výstupních frontách atd. Jako podklad pro dílčí rozhodnutí o typu zvolené architektury, neuronové sítě apod., vznikne i několik dalších dílčích modelů, zaměřených na konkrétní zkoumaný problém.

Vzhledem k charakteru řešené problematiky není třeba vytvářet kompletní fyzikální model. Cílem je návrh algoritmů, architektury a ověření teorie komplexním modelem. Hardwarová implementace je cílem budoucího výzkumu v rámci širší skupiny.

3 ANALÝZA PROBLÉMU

Rozborem vytčených cílů bylo zřejmé, že vhodnější bude výzkum orientovat spíše na architekturu přepínačů než směrovačů. Přepínače svojí architekturou pokrývají maximálně oblast zájmu prováděného výzkumu. Moderní přepínače podporují i protokoly vyšších vrstev. Přepínače s podporou L3 RM ISO/OSI jsou často některými výrobci doporučovány i do rolí PE (což obecně nelze doporučit kvůli často nedostatečné kapacitě směrovacích tabulek v přepínačích). Přesto se ale jeví výhodnější pohlížet na vyvíjený prvek jako na přepínač. Pro výzkum v souladu s cíli vytyčenými v kap. 2 je nezbytná práce s více porty. Pokud je uvažován přepínač s podporou L3, jsou problémy QoS, které musí vyhodnotit, prakticky totožné s problematikou vyhodnocovanou nativním směrovačem.

Problémy definované v rámci kap. 2 ve vazbě na řešenou problematiku se týkají zejména:

- celkové architektury přepínače (počtu portů, velikostí pamětí, umístění zásobníků atd.),
- architektury spojovacího pole,
- řízení spojovacího pole,
- správy front, plánování, a zahazování paketů.

V počáteční fázi výzkumu (r. 2008–2009) byly provedeny dílčí testy a analýzy možností. Bylo rozhodnuto o vývoji modelu aktivního prvku řízeného neuronovými sítími.

3.1 ARCHITEKTURA

3.1.1 STRUKTURA SPOJOVACÍHO POLE

Pro potřeby zapojení neuronové sítě do řízení spojovacího pole se jeví jako výhodné použít modifikovanou strukturu křížového spínače. Křížový spínač v roli spojovacího pole se jeví jako výhodný proto, že umožňuje snadné zapojení ovládní externím algoritmem. Ačkoli lze křížový spínač navrhnout tak, aby se jednalo de facto o samosměrovací pole, z hlediska implementace požadované funkcionality je výhodnější centrální řízení prvku, se samostatným ovládním každého z N^2 spínacích prvků. Okrajově byla zvažována i spojovací pole typu „full mash“, tedy plně propojené porty. V případě 4portového modelu by se jednalo o výhodné zapojení bez problémů s vnitřním blokováním. Nárůst spojení s počtem portů je evidentní nevýhodou této topologie.

3.2 APLIKACE NEURONOVÝCH SÍTÍ NA ŘÍZENÍ AKTIVNÍHO PRVKU

Umělou neuronovou síť lze v rámci přepínače, resp. jeho řízení, využít několika způsoby. Využití neuronových sítí se nabízí jako jedno z možných řešení problémů, které musí aktivní prvek řešit. Neuronové sítě vynikají v úlohách, jejichž řešení lze popsat deterministickými algoritmy jen obtížně nebo vůbec.

3.2.1 VOLBA NEURONOVÉ SÍTĚ

Volba vhodné neuronové sítě použité pro řízení spojovacího pole byla provedena na základě publikací [7], [8], [9] a zejména [10]. Pro řešení úloh obsahujících problematiku rozpoznávání schémat (pattern recognition) doporučuje literatura [10] použití algoritmus zpětného šíření chyby. Vzhledem k výsledkům výzkumů popsaných v [7], [8], [9] a v doporučeních uvedených v [10], byla pozornost zaměřena zejména na sítě založené na algoritmu zpětného šíření chyby (feedforward backpropagation).

3.2.2 KOMPARAČNÍ TEST NEURONOVÉ SÍTĚ

Všechny typy neuronových sítí, které byly v rámci výzkumu použity, byly podrobeny testům, jejichž úkolem bylo ověřit správnou funkci parametrů a proměnných, které lze při vytváření a trénování sítě, resp. jejího modelu, použít. Bylo třeba ověřit chování modelovacího prostředí, specifika implementace numerických integračních metod, nastavení simulačního algoritmu a zejména zvolit vhodnou neuronovou síť pro konkrétní aplikaci.

Nejjednodušším modelem, který byl vytvořen a použit pro otestování neuronových sítí, ověření základní funkčnosti simulačního softwaru a nastavitelnosti všech parametrů, byl model booleovské funkce antivalence [11]. Ta bývá někdy označována jako operace exkluzivní disjunkce, exkluzivní OR nebo, v technické praxi nejčastěji, XOR.

Logická operace XOR byla zvolena pro svoji jednoduchost a s tím spojenou snadnou implementovatelnost. Neuronová síť typu feedforward backpropagation

potřebuje ke korektnímu modelování operace XOR pouze dva neurony ve skryté (pracovní) vrstvě. Operace XOR byla zvolena také pro svůj samotný charakter – blízkost řešené problematice – v případě XOR, stejně jako v případě řízení přepínače, záleží na pořadí.

Tréninkovou množinu pro modelování XOR tvořily 2 matice. Vstupní matice \mathbf{P} , viz (3.1) (input data) a cílová matice \mathbf{T} , viz (3.2) (target data).

$$\mathbf{P} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad (3.1)$$

$$\mathbf{T} = (0 \quad 1 \quad 1 \quad 0) \quad (3.2)$$

Testování bylo zahájeno návrhem a modelováním XOR neuronovou sítí typu feedforward backpropagation. Tréninková funkce (training function) byla vybrána TrainLM (Levenberg – Marquardt, viz dále), adaptační funkce byla zvolena s klesajícím gradientem (gradient descent), dynamickými vahami a biasem (momentum weight and bias) – LearnGDM [10]. Výkonová funkce – MSE, viz dále. Síť byla vytvořena se 2 vrstvami, přičemž pracovní vrstva (layer 1) obsahovala 2 neurony. Přenosová funkce (transfer function) byla zvolena sigmoidní „hyperbolický tangens“ (tan-sigmoid), v Matlabu funkce `tansig`.

Pro trénování je použito 90 % dat vstupní matice, pro ověření a testování pak 5 %. Funkce `divideParam` je popsána v [10].

Síť dosáhne cílového stavu po přibližně 14 tréninkových epochách. Dosažený výkon (resp. střední kvadratická odchylka mse) sítě byl $mse = 7,9656 \cdot 10^{-18}$.

Podobně jako možnosti nastavení a vlastnosti sítě feedforward backpropagation byly zkoumány i vlastnosti perceptronu a sítě s rozpoznáváním vzoru (pattern recognition), kterou literatura [10] doporučuje jako náhradu perceptronu.

Provedené testy prokázaly, že pro řízení aktivního prvku se nejlépe hodí síť typu feedforward backpropagation, která bude v komplexním modelu použita přednostně.

4 NÁVRH PŘEPÍNAČE

Při naplňování cílů výzkumu vzniklo několik dílčích a dva komplexní modely síťového aktivního prvku přepínače. Jako první vznikly modely postavené nad systémem VNUML, průběžně pak komplexní modely vytvořené v MATLABu.

4.1 SIMULAČNÍ MODEL VNUML

Během výzkumu byly části výsledků ověřovány zejména na open source řešení VNUML, resp. UML, kde byl nahrazen původní přepínací program a byly prověřovány možnosti jeho vylepšení. Jako nejperspektivnější řešení se jeví instalace fyzické akcelerační karty obsahující přepínací jádro a jeho poskytnutí virtuálním počítačům.

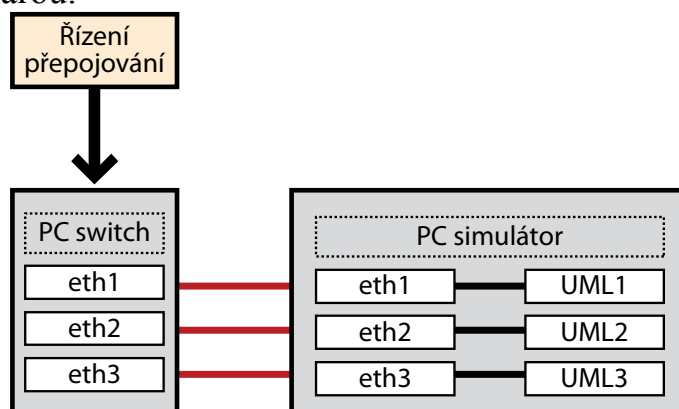
4.1.1 REALIZACE

Platforma VNUML byla zvolena kromě rychlé upravitelnosti také proto, že dávala naději na možné snadné převedení vytvořených modelů, resp. modifikací programu „UML switch“ do reálného provozu. Proto byl navržen a následně i sestaven emulátor složený ze 2 PC, viz obr. 4.1. Každé z PC bylo vybaveno 4 síťovými kartami, pro samotnou emulaci byla využita rozhraní eth1–3. Rozhraní eth0 bylo ponecháno pro kontrolní účely.

První z použitých PC, na obr. 4.1 označené „PC switch“, bylo využíváno jako emulátor přepínače. Byl na něm provozován upravený program „UML switch“, přičemž jeho vstupy/výstupy byly mapovány na fyzická rozhraní počítače eth1–3.

Druhé z použitých PC, na obr. 4.1 označené „PC simulátor“, sloužilo jako protistrana – umožňovalo emulaci 3 dalších počítačů – v tomto případě byly provozovány v rámci VNUML. Původním předpokladem bylo další rozšíření v rámci „PC simulátoru“, tedy simulace rozsáhlejší sítě uvnitř VNUML s tím, že rozhraní eth1–3 by byla použita jako trunk porty.

Celá snaha o fyzickou realizaci byla motivována možností jak testování, tak také provádět reálná měření. Ta probíhala v bodech, resp. propojeních vyznačených na obr. 4.1 červenou čarou.



Obr. 4.1: Emulátor přepínače

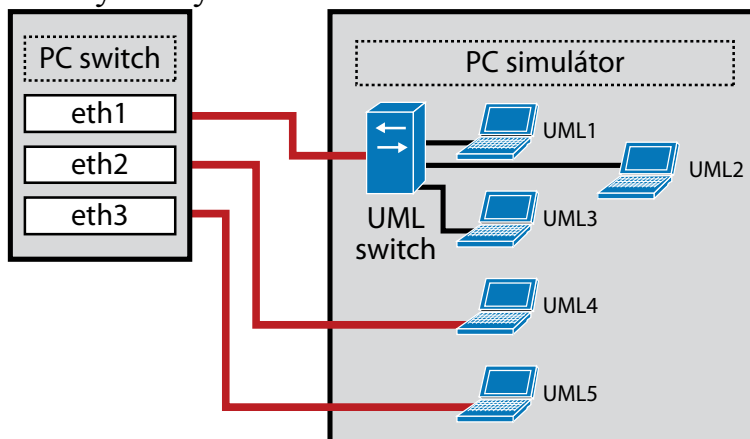
Konfigurace zachycená na obr. 4.1 se neosvědčila – prokázal se příliš velký vliv topologie, resp. počet virtuálních uzlů v rámci „PC simulátoru“ na zpoždění, viz graf na obr. 4.3.

Konfigurace 1 uvedená v grafu, obr. 4.3, svojí topologií odpovídá konfiguraci uvedené na obr. 4.1. Každému rozhraní v rámci systému „PC switch“ odpovídá rozhraní v rámci „PC simulátor“.

Konfigurace 2 uvedená v grafu, obr. 4.3, je zachycena na obr. 4.2. V této konfiguraci byla v rámci „PC simulátoru“ vytvořena virtuální síť postavená nad VNUML. Virtuální síť byla složena ze 3 uzlů označených na obr. 4.2 UML1–3. Jeden ze 4 portů virtuálního přepínače „UML switch“ byl mapován na fyzické síťové rozhraní „PC simulátoru“. Na zbývajících 2 fyzická rozhraní „PC simulátoru“ byly mapovány uzly UML4 a UML5, podobně jako v případě konfigurace 1, resp. obr. 4.1.

Měření zpoždění konfigurace 2 zachycené v grafu na obr. 4.3 proběhlo opět programem Ping. Měřena byla trasa mezi uzly UML1 a UML5.

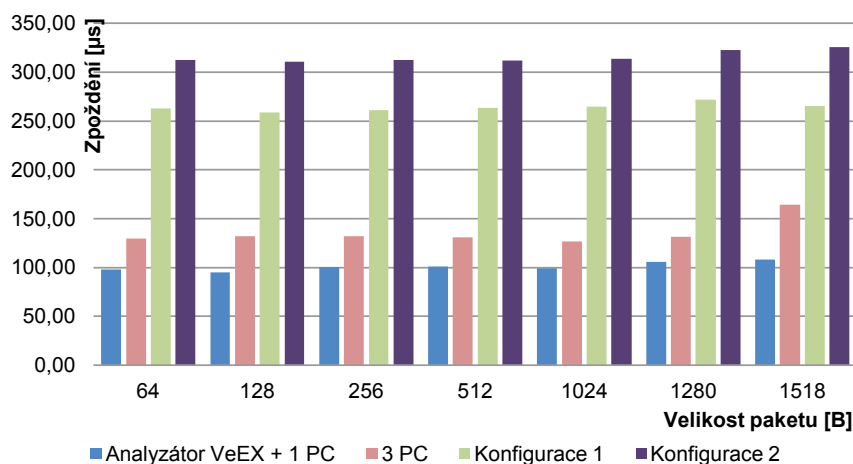
Výsledky měření jsou vyneseny do sloupcového grafu, obr. 4.3. Z grafu je patrné, že zpoždění je pro různě velké pakety/rámce prakticky neměnné. Systém pracoval při měření stabilně. Současně je ale z grafu patrný zásadní vliv topologie, ale i metodiky měření na výsledky.



Obr. 4.2: VNUML konfigurace 2

Původním předpokladem bylo, že měření provedená v rámci „konfigurace 1“, měření v konfiguraci „3 PC“ a měření analyzátozem VeEX VePAL TX300/e poskytnou přibližně stejné výsledky. Ve všech třech vyjmenovaných případech se jednalo o hvězdicovou topologii s centrálním prvkem „PC switch“. Konfigurace 2 posloužila jako potvrzení domněnky, že zásadní zpoždění do měření vnáší „PC simulátor“.

Při měření analyzátozem VeEX byl centrálním prvkem „PC switch“, analyzátozem VeEX byl jedním portem připojen k jeho rozhraní eth1, druhým k rozhraní eth2. Datový tok (náhodná data L2) vytěžující zátěž 30 % rozhraní byl generován rovněž analyzátozem VeEX a byl veden mezi rozhraním eth3 a rozhraními eth1 a eth2. Zajímavý je rozdíl zpoždění v případě měření analyzátozem VeEX a s použitím konfigurace „3 PC“, který je cca $30 \mu s$ s výjimkou velkých rámců 1 518 B, kde byl změněn rozdíl absolutního zpoždění $\Delta l_a = 56,2 \mu s$.



Obr. 4.3: Vliv topologie VNUML modelu na zpoždění

Kvůli rozdílným výsledkům dosažených při použití „PC simulátoru“, bylo od jeho dalšího použití a vývoje upuštěno.

4.1.2 ZHODNOCENÍ

Obtížná opakovatelnost výsledků, závislost na vedlejších okolnostech – výkon PC, procesy operačního systému, propustnost sběrnic atd. ukázaly, že modely vytvořené v rámci systému VNUML lze použít spíše pro výukové potřeby. Vytvořené algoritmy sice lze přenést do praktické realizace, ale výkon navržených algoritmů prakticky nelze vzájemně porovnávat. Z popsaných důvodů byl výzkum na platformě VNUML zastaven.

4.2 MODELOVANÝ PROTOKOL

Pro účely modelování problematiky spojené s QoS byl navržen zjednodušený protokol, resp. paket. Při návrhu byla snaha, aby obsahoval všechna pole, která mohou mít na QoS vliv. Navržený paket byl inspirován protokolem IP. Jeho cílem je tedy modelovat provoz na L3 RM ISO/OSI. Návrh obsahuje obdobu IP adresy zdroje a cíle, obdobu pole DSCP a datovou část.

Požadavkem na paket bylo, aby jeho hlavička měla konstantní velikost – to umožňuje zjednodušit model. Naopak část datová by měla mít velikost proměnlivou, což přiblíží charakteristikou a chováním paket vzorovému protokolu IP. Generátory paketů, viz kap. 5.2, by navíc měly umožnit konfigurovat datovou část tak, aby ve vybraných experimentech mohla být velikost paketu konstantní včetně datové části.

4.2.1 STRUKTURA PROTOKOLU

Schematický návrh nového paketu určeného pro použití v modelech je zachycen na obr. 4.4. Jeho hlavičku tvoří pole „Zdrojový port“, „Cílový port“ a „Priorita“.

0	1	2	3	4	5
Zdrojový port	Cílový port	Priorita		Data	

Obr. 4.4: Model paketu

Zdrojový port je obdobou zdrojové IP adresy. Může nabývat celočíselných hodnot 1–4. Pole identifikuje port, resp. generátor paketů, ze kterého byl paket odeslán.

Cílový port je obdobou cílové IP adresy. Nabývá celočíselných hodnot 1–4, označuje port, na který je paket směřován.

Priorita – pole je složeno ze dvou částí, tak aby odpovídalo poli DSCP. První číslice dvouciferného čísla označující prioritu nabývá celočíselných hodnot 1–4 (reprezentuje prioritu), přičemž vyšší číslo znamená vyšší přednost. Druhá číslice je z celočíselného intervalu 1–3 a určuje pravděpodobnost zahození paketu. V případě pravděpodobnosti zahození znamená vyšší hodnota větší pravděpodobnost zahození. Hodnota celého pole „priorita“ je počítána podle

vztahu (4.1), kde P specifikuje prioritu (přednost) paketu a d označuje pravděpodobnost jeho zahození. Hodnoty pole odpovídají hodnotám DSCP AF.

$$priorita = P \cdot 10 + d \quad (4.1)$$

Data – poslední pole modelového paketu je rovněž celočíselné. Pole zastupuje reálně přenášená data, přičemž pole má v binární reprezentaci proměnlivou velikost, ale může být definováno i staticky. Pole „Data“ má kromě simulace reálného obsahu **význam identifikátoru paketu**. Ve většině simulací je pole naplňováno náhodnými čísly z intervalu 1–10 000. Pravděpodobnost nežádoucího zopakování paketu se stejnými poli „Zdrojový port“, „Cílový port“ a „Data“ během jejich náhodného generování v požadovaném intervalu je malá. Jednotlivé pakety se shodným zdrojem a cílem lze díky poli „Data“ snadno odlišit.

Navržený paket byl implementován do paketových generátorů vytvořených a implementovaných jako součást komplexního modelu přepínače.

5 KOMPLEXNÍ SIMULAČNÍ MODEL

Klíčovou částí výzkumu byla tvorba návrhu aktivního prvku (přepínače) a jeho modelu. Přepínač (switch) je systém, jehož fyzikální podstata byla zjednodušena – byl vytvořen zjednodušený komplexní model, pracující s diskretním časem. Vytvořený model pracuje s pakety jako s kvanty. Simulační model byl vytvářen s cílem shrnout komplexně většinu proměnných, které mohou ovlivnit průchod paketu aktivním prvkem a umožnit jejich libovolnou modifikaci pro nalezení optimálních parametrů a řídicích algoritmů.

5.1 POPIS MODELU

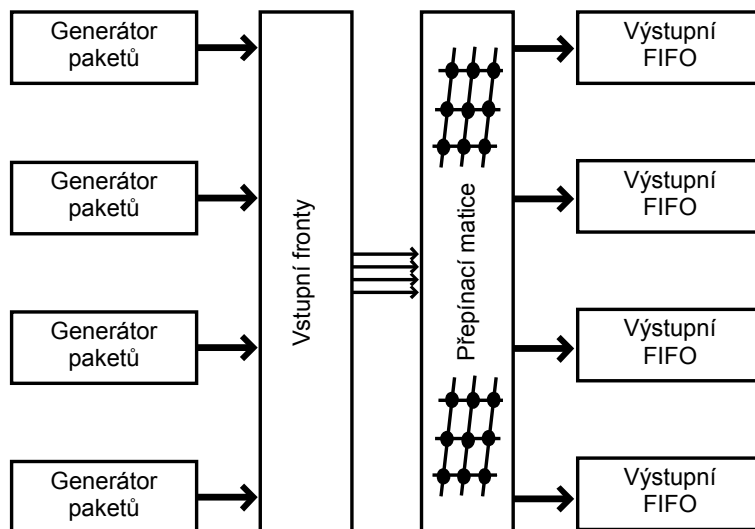
Modelován byl přepínač se 4 vstupními a 4 výstupními porty tak, jak bylo zmíněno v kap. 3.1. Přepínač byl navržen jako plně duplexní – tedy pro provoz s odděleným vysílacím a přijímacím kanálem. Provoz s polovičním duplexem není navrženým modelem podporován. Počet portů byl zvolen tak, aby byla současně zachována jednoduchost modelu a možnost simulovat všechny požadované jevy a situace a kolizní stavy.

Navržený přepínač je částečně založen na konvenční. Model je navíc doplněn o generátory provozu, blok monitorování a nově navržené řízení spojovacího pole. Celý přepínač je navržen s ohledem na QoS a na minimalizaci negativních důsledků jeho činnosti na přenášený síťový provoz.

Unikátní je řízení spojovacího pole, které je řízeno neuronovou sítí „Feedforward Backpropagation“ a správa vstupních front, která v rámci komplexního modelu používá upravený algoritmus MDRR.

Základní blokové schéma navrženého modelu je zachyceno na obr. 5.1. Hlavními bloky přepínače jsou:

- generátory paketů,
- vstupní fronty,
- přepínací matice (včetně bloků řízení neuronovou sítí),
- a výstupní fronty (bloky výstupních front jsou na obr. 5.1 označeny jako „Výstupní FIFO“).



Obr. 5.1: Základní blokové schéma modelu

5.2 GENERÁTOR PAKETŮ

Model obsahuje 4 stejné paketové generátory. Každý z generátorů je zcela nezávisle konfigurovatelný. Generátory generují paket ve formátu popsaném v kap. 4.2.1.

Generátory jsou synchronizovány a inicializovány blokem „Spouštěč“. Blok „Spouštěč“ je uživatelsky konfigurovatelný a je také složen ze 4 subsystémů, přičemž může být každý ze 4 těchto subsystémů konfigurován samostatně a odlišně od ostatních. Tato vlastnost je výhodná pro simulaci sítí s různými charakteristikami propojenými modelovaným aktivním prvkem. Pakety mohou být generovány náhodně, nebo uživatelsky předdefinované. To, že v daném okamžiku není vyslán žádný paket, je modelováno vygenerováním paketu majícího všechna pole rovna 0.

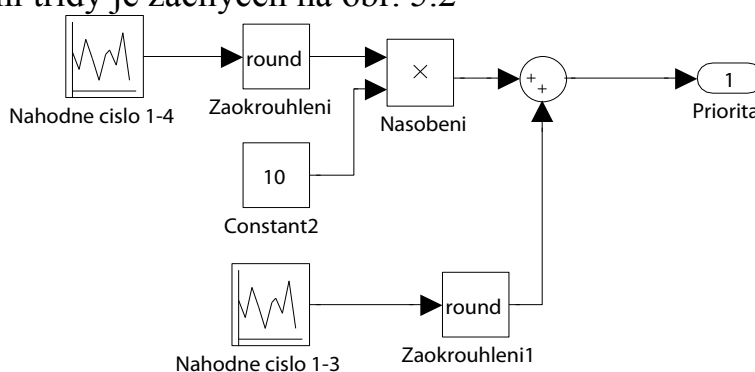
V bloku „Spouštěč“ lze měnit jak rychlost generování paketu, tak také pravděpodobnost, se kterou budou generovány prázdné pakety, tedy kdy nebude na daném portu do systému vyslán paket. Přesné nastavení lze provést v bloku „perioda 0 paketu“ – samostatně pro každý z paketových generátorů. Lze nastavit generování s uniformní pravděpodobností generování prázdného paketu, nebo lze zvolit Gaussovo rozložení pravděpodobnosti. Schopnost externě řídit rychlost generování paketů pro každý z generátorů umožňuje v budoucnu modelovat i chování protokolů vyšších vrstev nebo pokročilých mechanismů zahazování paketů, jako např. Modified Tail Drop aj.

Přepínačem „náhodná perioda/linková rychlost“ lze vkládání prázdných paketů zastavit a testovat systém plnou linkovou rychlostí. Výjimečné výskyty prázdných

paketu na výstupu generátoru v tomto režimu jsou zapříčiněny filtrací kolizních paketů – tedy těch, které mají shodný zdrojový a cílový port.

Na výstupu bloku „Generátor“ je ověřováno, zda se neshodují pole zdrojového a cílového portu. Vzhledem k tomu, že oba tyto porty mohou být (dle konfigurace) voleny náhodně, může ke shodě dojít. Protože pakety se shodným zdrojovým a cílovým portem jsou defektní a v reálném provozu se nevyskytují, jsou zahazovány (resp. nahrazovány nulovými pakety). Ačkoli lze v generátoru vytvářet pakety s libovolnou zdrojovou i cílovou adresou (tedy i mimo rozsah modelovaného přepínače), model ve své předkládané podobě tyto stavy neošetřuje.

Výpočet hodnoty pole „priorita“ je prováděn podle vztahu (4.1) tak, aby bylo co nejpodobnější poli DSCP AF, jak je popsáno v kap. 4.2.1. Každá ze dvou částí pole P a d může být definována uživatelsky, nebo může být volena náhodně. Náhodný generátor prioritní třídy je zachycen na obr. 5.2



Obr. 5.2: Blok náhodného generování prioritní třídy

Reálný ethernetový provoz může být z hlediska časování považován aktivním prvkem za pseudonáhodný. Pakety mohou být do aktivního prvku doručovány stochasticky – zpoždění paketů se může lišit. Kvůli simulaci tohoto jevu – jitteru – jsou pakety po vygenerování v rámci bloku „Generátor“ uloženy do fronty FIFO, dokud nejsou odeslány ven z generátoru v náhodném nebo uživatelsky definovaném čase.

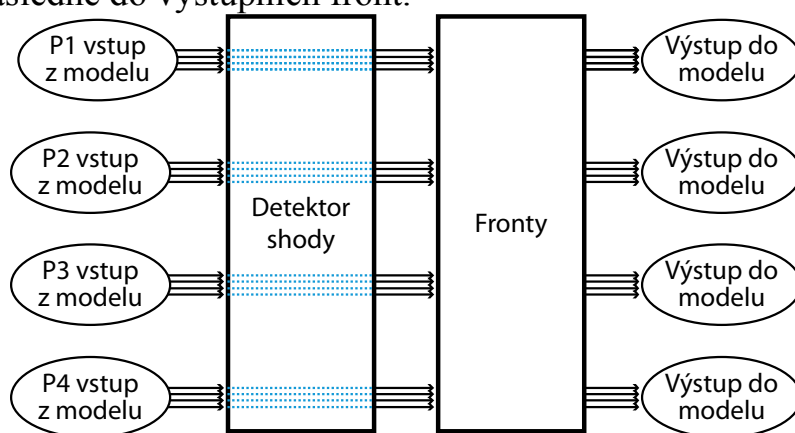
Podstatným problémem, který bylo třeba v rámci modelu vyřešit, byla detekce začátku a konce paketu, resp. rámce. Celá problematika je v rámci Ethernetu a TCP/IP vyřešena, ale z důvodů popsaných v kap. 4.2 nebylo praktické modelovat celou sadu protokolů. Vkládat do pro model vytvořeného protokolu preambuli se neosvědčilo při úvodních testech. Model musel na mnoha místech paketu provádět syntaktickou analýzu celého protokolu (parsing). Celý problém byl vyřešen s použitím paralelního (sběrnicevého) systému datového přenosu v rámci modelu. V místech, kde je to možné, je celý provoz uložen do sběrnice (blokem Bus Creator), reprezentovaným modulem „Koncentrátor“. V bloku, kde je třeba s některou z částí paketu pracovat, lze snadno tuto část vydělit (blok Bus Selector). Každé z polí protokolu je tedy v rámci celého modelovaného přepínače přenášeno důsledně odděleně. Ačkoli reálně (na binární vrstvě) mají pakety protokolu proměnlivou délku, přenos sběrnice

s nimi umožňuje pracovat jako s „kontextem“, který není v rámci takto řešeného přepínače třeba vytvářet.

5.3 VSTUPNÍ FRONTY

Blok „Vstupní fronty“ je klíčovým blokem řešícím problematiku priority provozu v rámci celého přepínače. Umožňuje volit na nejvyšší úrovni ze dvou možností správy vstupních front, resp. řešení kolizí paketů směřujících do stejného cíle. Zjednodušené blokové schéma základních součástí je zachyceno na obr. 5.3.

Součástí bloku vstupních front je podblok „Detektor shody“, jehož účelem je detekovat a označovat pakety se shodným cílem, tedy pakety kolidující. Podblok „Detektor shody“ pracuje s kopií původního paketu (na obr. 5.3 je volný průchod paketu blokem naznačen modrou čárkovanou čarou). Detektor shody označí kopie kolidujících paketů hodnotou -1 v poli „Priorita“. Pokud blok neidentifikuje kolizi paketů, jsou všechny pakety bezprostředně předávány do spojovacího pole (bloku „Přepínací matice“) a následně do výstupních front.



Obr. 5.3: Blokové schéma správy vstupních front

Pokud je detekována kolize, umožňuje komplexní model dva různé způsoby nakládání s kolidujícími pakety.

První možností, kterou lze v bloku řízení vstupních front volit pro případ kolize paketů, je zahození paketu s nastavenou nižší prioritou P . Pokud koliduje více paketů, je vždy propouštěn pouze paket s nejvyšší prioritou, ostatní jsou zahozeny (zahození je realizováno stejně, jako v případě zahazování v generátoru, tedy nahrazením paketu prázdným paketem). Jedná se o rychlé a transparentní řešení, ale se zbytečnou ztrátovostí paketů. Tato metoda řešení kolizí je implementována v bloku „Zahazování shody“.

Druhá možnost zacházení s kolidujícími pakety v bloku vstupních front je, ve srovnání s volbou první, sofistikovanější. Cílem návazných podbloků je plně podpořit QoS. Dva a více paketů s detekovanou kolizí jsou odeslány do podbloku „Řazení QoS“, který je jedním z podbloků bloku „Fronty“ na obr. 5.3.

V rámci podbloku „Řazení QoS“ jsou ukládány pakety do front. Bloky „QoS řazení 1–4“ rozdělí pakety do skupin.

Pro každý vstupní port je definována jedna skupina front pro uložení 8 paketů. Pokud je ve frontě více než 8 paketů, všechny nově příchozí pakety jsou zahozeny (mechanismus Tail Drop). Každá skupina front je složena z jednotlivých front – pro každou prioritní třídu je dedikována 1 samostatná fronta. Do front jsou pakety rozdělovány podle priorit.

V případě, že je výstupním rozhraní (port) volné a připravené pro odeslání paketu, jsou čteny a postupně odesílány pakety z front. Základním algoritmem pro čtení paketů z front je algoritmus se striktní prioritou MDRR – začíná s odesíláním paketů z fronty s nejvyšší prioritou. V případě, že je tato fronta prázdná, algoritmus pokračuje s frontou s nižší prioritou atd. Blokem následujícím za vstupními frontami je spojovací pole.

5.4 SPOJOVACÍ POLE S ŘÍZENÍM

Blok nazvaný v modelu „Krizovy_spinac_s_rizenim“ se skládá ze dvou hlavních částí – řízení (generátoru konfigurační matice) a spojovacího pole – křížového spínače. Spojovací pole je modelováno jako pole 4×4 jednoduchých externě řízených spínačů.

5.4.1 POPIS SPOJOVACÍHO POLE

Na vstupy jednotlivých spínačů jsou přiváděny sběrnice (pakety) – vždy stejný vstupní paket na jeden řádek spínačů. Každý ze sloupců spojovacího pole směřuje do jednoho výstupního portu. Blok „Vystupni filtr1–4“ slouží k identifikaci platného paketu (odlišení od prázdných paketů) a sloučení 4 výstupů do jednoho cílového směřujícího do výstupní fronty.

Součástí bloku „Krizovy_spinac_s_rizenim“ je subsystém monitoringu stavu spojovacího pole.

5.4.2 ŘÍZENÍ SPOJOVACÍHO POLE

Prostřednictvím vstupů „Rizeni“ každého ze spínačů je celé spojovací pole ovládáno tzv. konfigurační maticí **C**. Konfigurační matice je generována v každém simulačním kroku. Může být generována jak konvenčním algoritmem, tak umělou neuronovou sítí.

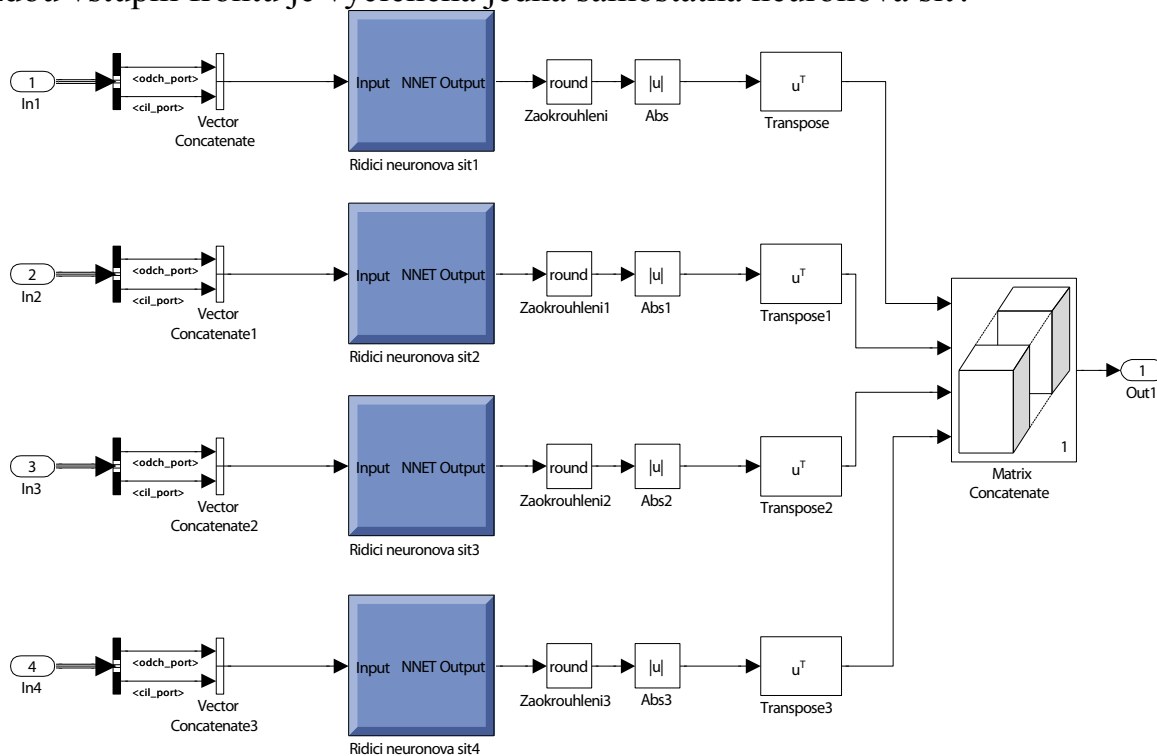
Konvenční přepínače jsou obvykle řízeny explicitně programováním definovaným algoritmem, který řídí síťový procesor a další součásti aktivního prvku. Navržený aktivní prvek je řízen alternativně – neuronovou sítí, viz kap. 3.2. Matice (5.1) je příkladem generované konfigurační matice.

$$\mathbf{C} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (5.1)$$

Přepínač je navržen tak, aby pracoval zcela paralelně, z čehož by v ideálním případě plynulo, že s rostoucím počtem portů nebude klesat výkon. V praxi je počet

portů promítnut do velikosti paralelních řídicích neuronových sítí, z čehož plyne pokles výkonu s přibývajícím počtem portů. Další pokles výkonu je závislý na zastoupení kolidujících paketů v provozu.

Paralelní struktura návrhu je zachycena na obr. 5.4, který zobrazuje blok „Gen_konf_mat“. Čtyři řídicí neuronové sítě jsou na obr. 5.4 vyznačeny modře. Pro každou vstupní frontu je vyčleněna jedna samostatná neuronová síť.



Obr. 5.4: Struktura bloku řídicích neuronových sítí

Vektor vstupující do neuronové sítě \mathbf{P} se skládá ze dvou částí (čísel) – zdrojového a cílového portu. Výstupní (cílový) vektor \mathbf{T} je složen ze 4 binárních čísel. Příkladem vstupního a výstupního vektoru jsou matice (5.2).

$$\mathbf{P} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}; \mathbf{T} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad (5.2)$$

Struktura neuronové sítě je odvozena od otestované struktury popsané v rámci kap. 3.2.2. Síť je typu „Feedforward Backpropagation“. Volba, resp. vhodnost použití tohoto typu sítě byla potvrzena výsledky testů popsanými v kap. 3.2.2 a rovněž během trénování sítě pro řešení přepínání paketů. Ve skryté vrstvě sítě bylo použito 100 neuronů, ze 4 neuronů se pak skládá výstupní vrstva.

Tréninková množina je rozdělena:

- 90 % celé tréninkové množiny je použito k trénování,
- 5 % tréninkové množiny je použito k ověřování (validaci),
- 5 % tréninkové množiny použito k testování.

Vytvořená neuronová síť rychle konvergovala k požadované střední kvadratické chybě mse (mean squared normalized error, viz kap. 3.2.2. Po třech epochách byla dosažena požadovaná $mse = 4,306 \cdot 10^{-21}$.

5.5 VÝSTUPNÍ FIFO FRONTY

Každému výstupnímu portu je přiřazena jedna výstupní FIFO fronta – v rámci obr. 5.1 označena „Výstupní FIFO“. Výstupní FIFO fronty svým určením odpovídají HW frontám. Velikost fronty je nastavena na 8 paketů, stejně jako v případě vstupních FIFO front.

Pakety jsou ukládány do výstupních FIFO a čteny pouze tehdy, je-li cílový port na opozičním zařízení připraven odesílat paket přijmout. To je modelováno samostatným řídicím vstupem na každé frontě, na který lze připojit samostatný spouštěč. Protože většinou byly řešeny úlohy spojené s QoS a řízením spojovacího pole, bylo nezáhodoucí, aby na výstupních FIFO frontách docházelo ke ztrátám paketů nebo dalšímu zpoždování provozu. Proto byly obvykle provozovány výstupní fronty v synchronním režimu – do bloku monitorování byl odeslán stejný počet paketů, jako byl přijat na vstupu.

5.6 VÝSLEDKY A TESTOVÁNÍ

Navržený model přepínače je vybaven testovacími a vizualizačními nástroji sloužícími pro ověření výsledků simulací. Kromě bloku monitorování, který je součástí bloku „Krizovy_spinac_s_rizenim“, viz kap. 5.4.1, je v modelu i další monitorovací blok umístěný přímo v první vrstvě modelu, označený „Monitoring“.

Podrobným ověřovacím nástrojem v bloku „Monitoring“ je „srovnávací tabulka“. Srovnávací tabulka. Kromě podrobného zobrazení formou tabulky jsou do simulinkového modelu zabudovány statistické výstupy odesílané ke zpracování skriptem „vyhodnoceni_v3.m“ do MATLABu. Statistiky jsou sbírány na výstupech generátorů a na cílových portech. Statistiky jsou sbírány v rámci bloku „Blok pocitadel paketu“, který je tvořen 32 subsystemy. Úkolem je filtrovat a sčítat počty vygenerovaných paketů všech priorit. Obdobně jsou zpracovávány pakety přijaté, resp. pakety zaslané do výstupních rozhraní po průchodu přepínačem.

Měření jsou zcela nezávislá jak na zmíněné vygenerované „srovnávací tabulce“, tak na zbytku modelu. Počty vygenerovaných a ztracených paketů lze tedy kontrolovat na několika místech a různými nezávislými způsoby.

Byl modelován průchod staticky definovaných paketů přepínačem. Výsledky jsou zpracovány v tab. 5.1 kde:

- Z – označuje zdrojový (odchozí port).
- C – označuje cílový port.
- P – označuje prioritu.
- D – označuje data.

Z uvedeného příkladu je zřejmé, že ve všech případech došlo na cílových portech ke ztrátě paketů. Zajímavé je ale rozložení – které pakety byly zahozeny.

V předloženém příkladě byl v modelu použit navržený algoritmus – striktní MDRR. Ztrátovost paketů s nejvyšší prioritou byla na všech portech nulová. Fronty s nižší prioritou (P2) ztratily 1–2 pakety.

Vzhledem k charakteru navrženého algoritmu je ztrátovost na portech s prioritou 3 a 4 očekávaná. Vysoká celková ztrátovost na všech portech je dána vysokým zatížením vstupních front – aplikuje se „Tail Drop“.

Cílový port č.	Odesláno pak.	Ztraceno celkem	Ztrátovost [%]	Ztraceno P 1	Ztraceno P 2	Ztraceno P 3	Ztraceno P 4
1	10	1	10,0	0	0	1	0
2	23	4	17,4	0	2	2	0
3	16	1	6,3	0	1	0	0
4	14	3	21,4	0	2	1	0

Tab. 5.1: Měření průchodu paketů přepínačem SMDRR

Pro srovnání byla sada paketů, zpracována druhým z bloků vstupních front, tedy „Zahazování shody“. Výsledky po zpracování jsou uvedeny v tab. 5.2.

Oproti měření uvedenému v tab. 5.1 je patrná ztrátovost i u paketů s prioritou P1 – byly ztraceny 2 pakety směřující do portu 2 a 2 pakety směřující do portu 3. Obecně lze – v souladu s očekáváním – pozorovat rovnoměrnější rozložení ztrátovosti mezi jednotlivé prioritní třídy.

Cílový port č.	Odesláno pak.	Ztraceno celkem	Ztrátovost [%]	Ztraceno P 1	Ztraceno P 2	Ztraceno P 3	Ztraceno P 4
1	10	1	10,0	0	1	0	0
2	23	2	8,7	2	0	0	0
3	16	7	43,8	2	3	2	0
4	14	0	0,0	0	0	0	0

Tab. 5.2: Měření průchodu paketů přepínačem bez QoS

Při měřeních, viz tab. 5.1 a tab. 5.2 bylo vygenerováno vždy celkem 63 paketů (pakety jsou generovány a přijímány porty paralelně) ve 20 taktech, resp. jednotkách simulačního času s velikostí kroku 0,1. Během těchto 20 taktů model umožňuje vygenerovat a přijmout maximálně 40 paketů (reálně je simulací, resp. generátory, vygenerováno 41 skupin paketů – poslední 41. skupina ale není zpracována).

Simulováno bylo zatížení přepínače náhodným provozem – bylo celkem odesláno 1 464 paketů při 1 000 simulačních taktech. Výsledek je zachycen v tab. 5.3. Z výsledků měření uvedených v tab. 5.3 je opět patrná práce algoritmu SMDRR – nejméně jsou zahazovány pakety s nejvyšší prioritou. Protože ale byly při tomto měření odchozí pakety odesílány náhodně, nelze vyloučit zahazování mechanismem „Tail Drop“. Proto jsou počty zahozených paketů v polovině případů větší v prioritní třídě 3 oproti prioritě 4.

Cílový port č.	Odesláno pak.	Ztraceno celkem	Ztrátovost [%]	Ztraceno P 1	Ztraceno P 2	Ztraceno P 3	Ztraceno P 4
1	526	293	55,7	14	20	150	109
2	227	131	57,7	19	32	17	63
3	211	157	74,4	12	57	72	16
4	500	260	52,0	8	10	105	137

Tab. 5.3: Měření průchodu paketů přepínačem SMDRR – 1 000 simulačních taktů

Pokud je snížena rychlost generování paketů (úpravou periody v bloku „Spousteč_g“), úměrně dochází ke snížení ztrátovosti. Pokud je perioda snížena na polovinu – oproti příkladu uvedenému v tab. 5.1, pokles ztrátovosti je patrný z tab. 5.4.

Ztráta 2 paketů priority P2 směřujících na port C3 byla identifikována ve „srovnávací tabulce“ jako důsledek 5 paketů směřujících na port C3 – 2 + 2 + 1 ve 3 simulačních taktů.

Cílový port č.	Odesláno pak.	Ztraceno celkem	Ztrátovost [%]	Ztraceno P 1	Ztraceno P 2	Ztraceno P 3	Ztraceno P 4
1	2	0	0,0	0	0	0	0
2	14	0	0,0	0	0	0	0
3	14	2	14,3	0	2	0	0
4	20	2	10,0	0	0	0	2

Tab. 5.4: Měření průchodu paketů přepínačem SMDRR – 50% zatížení

Ovlivnit ztrátovost navrženého přepínače mohou rovněž délky vstupních softwarových front. Prodloužením softwarových front lze snížit ztrátovost, ale zásadním způsobem může dojít k nárůstu zpoždění. Zpoždění lze v rámci modelovaného přepínače pozorovat nejlépe v rámci „srovnávací tabulky“.

Prodloužení vstupních softwarových front bylo modelováno – výsledek je uveden v tab. 5.5. Jako srovnávací provoz byl opět použit stejný staticky definovaný paket, jako v předchozích případech. Prioritním třídám P1 a P2 byly délky front ponechány, jak bylo zmíněno v kap. 5.3 délky 8. Fronty P3 a P4 byly prodlouženy na 10 – to odpovídá ¼ všech možných vygenerovaných paketů (při délce simulace 20 simulačních taktů, která byla použita).

Cílový port č.	Odesláno pak.	Ztraceno celkem	Ztrátovost [%]	Ztraceno P 1	Ztraceno P 2	Ztraceno P 3	Ztraceno P 4
1	10	1	10,0	0	0	1	0
2	23	4	17,4	0	2	2	0
3	16	1	6,3	0	1	0	0
4	14	0	0	0	0	0	0

Tab. 5.5: Měření průchodu paketů přepínačem SMDRR – prodloužení front

Ztrátovost je závislá také na rozložení provozu – pokud je provoz generován náhodně, statisticky významná jsou až měření provedená na delších simulacích. V případě uvedeném v tab. 5.5 se na výsledku negativně projevil důsledek krátké simulace (cílový port č. 3) – pakety zůstávají uloženy ve frontách – nejsou ztraceny, ale protože simulace je po 20 simulačních taktech ukončena a pakety nebyly doručeny, jsou mezi ztracené, resp. nedoručené počítány.

Předejít problému lze buď prodloužením simulace, kdy bude délka simulace výrazně delší než je délka front, podobně, jako je tomu u reálného provozu a tím se ztrátovost posledních paketů stane statisticky nevýznamná nebo lze manuálně definovat odesílání pouze např. do poloviny délky simulace (a následně odesílat pouze prázdné pakety), tak aby se fronty vyprázdnily v průběhu simulace. Pro manuální analýzu chyb „srovnávací tabulkou“ je tato metoda vhodnější.

Cílový port č.	Odesláno pak.	Ztraceno celkem	Ztrátovost [%]	Ztraceno P 1	Ztraceno P 2	Ztraceno P 3	Ztraceno P 4
1	3	0	0,0	0	0	0	0
2	11	0	0,0	0	0	0	0
3	7	0	0,0	0	0	0	0
4	8	0	0,0	0	0	0	0

Tab. 5.6: Měření průchodu paketů přepínačem SMDRR – zkrácení odesílání paketů

Výsledky měření potvrdily funkčnost navrženého přepínače. Navržený a simulovaný aktivní prvek dokázal pracovat bez ztrát. Při nastavování a provozu simulace je ale třeba zohlednit zmíněná specifika vytvořeného modelu.

ZÁVĚR

V rámci řešení disertační práce byl navržen nový aktivní prvek – přepínač řízený neuronovou sítí. Všechny cíle definované v kapitole 2 se podařilo v plném rozsahu naplnit. V první fázi řešení disertační práce proběhl výzkum v současnosti dostupných technologií a aktuálního stavu poznání v oboru přepínaných sítí a konstrukci síťových prvků. Pro vývoj nového způsobu řízení byly zvažovány 2 síťové prvky – přepínače a směrovače. Vzhledem k postupnému prolínání byl zvolen k návrhu aktivní prvek – přepínač. Konceptně je cílen do rolí prvku na hraně operátora (PE). Přepínač je navržen tak, aby respektoval soudobé požadavky na QoS.

Prvním vytčeným cílem disertační práce byla analýza a měření dostupných aktivních prvků. Výsledky měření jsou shrnuty v kap. 5.6. Dílčí měření jsou také obsažena v kap. 1.1 a 4.1. Byly proměřeny vlastnosti přepínačů Cisco, Hewlett-Packard, Micronet a platformy VNUML. Platforma VNUML byla do testů zařazena zejména proto, že byla v úvodu výzkumu problematiky řešené v rámci disertační práce zvažována jako perspektivní pro testovací implementaci navrženého řešení. Postupně se ale platforma VNUML ukázala jako nevhodnou volbou. Problematikou návrhu modelu v rámci VNUML se zabývá kap. 4.1.

V rámci návrhu nového prepínače a zejména jeho řízení bylo třeba zvolit neuronovou síť, která by byla pro řízení aktivního prvku nejvhodnější. O této problematice pojednává kap. 3.2. Jednalo se o druhý z cílů disertační práce. Po provedení komparačních testů (kap. 3.2.2) byla pro řízení navrhovaného prepínače zvolena neuronová síť „feedforward backpropagation“.

Dalším cílem disertační práce bylo navrhnout koncepčně nový prepínač. Touto problematikou se zabývá jak část kap. 3, tak kap. 4. Byly analyzovány typy spojovacích polí – kvůli možnosti externího řízení a kontrolovatelnosti bylo rozhodnuto o použití křížového spínače (kap. 3.1.1). Byla také navrhována celková koncepce aktivního prvku – bylo rozhodnuto o návrhu čtyřportového prepínače. Bylo také rozhodnuto o umístění softwarových front. Oproti běžně používanému umístění za spojovací pole před hardwarové fronty byly softwarové fronty situovány naopak před spojovací pole. Toto umístění má řadu nevýhod – např. problémy blokování paketů uvnitř fronty. Z hlediska navrhovaného modelu se ale ukázalo zvolené řešení jako nejlepší. Pro správu paketů ve frontách byl navržen algoritmus Modified Deficit Round-Robin (modifikovaný deficitní okruh), který byl upraven do podoby striktní priority. Při analýze algoritmů pro správu softwarových front byla zkoumána i možnost jejich správy neuronovou sítí. Postupně se ale ukázalo použití neuronové sítě jako komplikovanější a výsledky hůře predikovatelné než v případě použití konvenčních algoritmů. Proto bylo řízení správy softwarových front ponecháno na zmíněném konvenčním algoritmu.

Protože modelování celého RM ISO/OSI, resp. TCP/IP se ukázalo být neefektivní, byl jako jeden z cílů disertační práce stanoven návrh výrazně zjednodušeného protokolu vhodného pro řešení stanovené problematiky. Byl navržen protokol obsahující nejvýznamnější pole se vztahem ke QoS a směrováním paketu v rámci aktivního prvku. Navržený protokol má v hlavičce pouze 3 pole, datová část je čtvrtá. V hlavičce jsou obsažena pouze pole adres (zdroj a cíl paketu) a pole priority, které je inspirováno polem DS, resp. DSCP IP protokolu. Datová část slouží také k identifikaci paketu při kontrole práce navrženého prepínače.

Posledním z cílů disertační práce bylo všechny navržené koncepce ověřit v rámci komplexního simulačního modelu. Vzhledem k tomu, že výzkum, resp. řešení disertační práce, probíhal nelineárně, v prvotních fázích bylo uvažováno spíše o modelování v rámci VNUML. Později byly všechny modely tvořeny pouze v rámci MATLABu a Simulinku. Postupnými úpravami modelů, simulačních časů, hustoty generování paketů atd., se podařilo dosáhnout nulové ztrátovosti paketů v navrženém prepínači. Navržený prepínač a jeho model potvrdily, že síťový prvek lze neuronovou sítí řídit.

Pomocí provedených simulací se podařilo model optimalizovat a připravit jej pro hardwarovou implementaci. Na výzkum provedený v rámci disertační práce by v budoucnu měla navázat implementace do vývojových karet INVEA-TECH Combo-20G, tedy použití v reálném síťovém prostředí.

Dílní výsledky byly průběžně prezentovány v rozličných publikacích. Všechny vytčené cíle disertační práce byly dosaženy.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] DOSTÁLEK, Libor a KABELOVÁ, Alena. *Velký průvodce protokoly TCP/IP a systémem DNS*. Praha : Computer Press, 2000. ISBN: 80-7226-323-4.
- [2] International Organization for Standardization. ISO/IEC 7498: Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. Genève : International Organization for Standardization, 1994.
- [3] LAW, David J., a další. IEEE 802.3-2012: IEEE Standard for Ethernet. [Online] 28. 12 2012. [Citace: 10. 3 2014.] <http://standards.ieee.org/about/get/802/802.3.html>. ISBN 973-07381-7312-2.
- [4] GEBALI, Fayez. *Analysis of Computer and Communication Networks*. New York : Springer Science+Business Media, LLC, 2008. ISBN 978-0-387-74436-0.
- [5] POLÍVKA, Michal a PELKA, Tomáš. Computer network switch testing and modelization. *Elektrorevue - Electrotechnics magazine*. 2, 2011, Sv. 2, June 2011, stránky 32–38.
- [6] Cisco Systems Inc. Understanding and Tuning the tx-ring-limit Value. *IP to ATM Class of Service*. [Online] 14. 12 2007. [Citace: 27. 11 2011.] http://www.cisco.com/en/US/tech/tk39/tk824/technologies_tech_note09186a00800fbafc.shtml.
- [7] ŠKORPIL, Vladislav a ŠŤASTNÝ, Jiří. Alternatives of Converged Network Control. *Elektrorevue*. 2010, Sv. 1, 1, stránky 19–23.
- [8] *Network Elements Controlled by Artificial Neural Network*. ŠKORPIL, Vladislav a NOVÁK, David. Corfu Island, Greece : WSEAS, 2010. Proceedings of 14th WSEAS International Conference on Communications. stránky 145–148. ISBN 978-960-474-200-4.
- [9] *Back Propagation and Genetic Algorithms for Control of the Network Element*. ŠKORPIL, Vladislav a KAMBA, Stanislav. Budapest, Hungary : IEEE, 2011. 34th International Conference on Telecommunications and Signal Processing. stránky 240–243. ISBN 978-1-4577-1410-8.
- [10] BEALE, Mark Hudson, HAGAN, Martin T. a DEMUTH, Howard B. *Neural Network Toolbox 7*. Natic : The MathWorks, Inc., 2010.
- [11] BARTSCH, Hans Jochen. *Matematické vzorce*. Praha : Mladá fronta, 2002. ISBN 80-204-0607-7.

SEZNAM VLASTNÍCH PRACÍ

- [1] POLÍVKA, Michal; ŠKORPIL, Vladislav. Přístupové a transportní sítě – laboratorní cvičení. Brno: Vysoké učení technické v Brně, 2013. s. 1-80. ISBN: 978-80-214-4726- 4.
- [2] ŠKORPIL, Vladislav; POLÍVKA, Michal. Parallel WAN Switch Based on Neural Network. In Recent Researches in Communications and Computers. WSEAS Press, 2012. s. 68-72. ISBN: 978-1-61804-109- 8.
- [3] POLÍVKA, Michal; ŠKORPIL, Vladislav. Modeling of Network Switch Controlled by Neural Network. In 2012 35th International Conference on Telecommunications and Signal Processing (TSP) (id 20316). 2012. s. 165-168. ISBN: 978-1-4673-1116- 8.
- [4] POLÍVKA, Michal; ŠKORPIL, Vladislav. Modeling Logical Function Antivalence Using Neural Network in MATLAB. In 2011. ISBN: 978-80-214-4283- 2.
- [5] POLÍVKA, Michal; PELKA, Tomáš. Computer network switch testing and modelization. Elektrorevue – Internetový časopis (<http://www.elektrorevue.cz/>), 2011, roč. 2011, č. 2, s. 32-38. ISSN: 1213- 1539.
- [6] POLÍVKA, Michal; PERROTON, Laurent; PELKA, Tomáš. VNUML – application in computer network, switch modeling. In 6th International Conference on Teleinformatics. 2011. s. 136-141. ISBN: 978-80-214-4231- 3.
- [7] POLÍVKA, Michal; PELKA, Tomáš. Unified Communications – Possible Risks and Their Solutions. In Telecommunications and signal processing. 2010. s. 1-4. ISBN: 978-963-88981-0- 4.
- [8] POLÍVKA, Michal; PELKA, Tomáš. Simulátor rozsáhlých sítí – SimPP. Elektrorevue – Internetový časopis (<http://www.elektrorevue.cz/>), 2010, roč. 2010, č. 9, s. 1-5. ISSN: 1213- 1539.
- [9] PELKA, Tomáš; POLÍVKA, Michal; HAJNÝ, Jan. SimPP – Wide Area Network Simulator. In The 33rd International Conference on Telecommunication and Signal Processing, TSP 2010. 2010. s. 1-4. ISBN: 978-963-88981-0- 4.
- [10] POLÍVKA, Michal; PELKA, Tomáš; KACÁLEK, Jan. Moderní telekomunikační služby ve výuce. In International WORKSHOP RTT 2009 Research in Telecommunication Technology. 2009. s. 1-3. ISBN: 978-80-01-04411- 7.

- [11] PELKA, Tomáš; POLÍVKA, Michal. Clustering a jeho využití. In International WORKSHOP RTT 2009 Research in Telecom-munication Technology. 2009. s. 1-4. ISBN: 978-80-01-04411- 7.
- [12] PELKA, Tomáš; POLÍVKA, Michal. Comparison of Python virtual machines. Linux+, 2008, roč. 2008, č. 11, s. 1-8. ISSN: 1733- 4209.
- [13] POLÍVKA, Michal. Možnosti moderního síťového zabezpečení zastaralé výpočetní techniky. Elektrovue – Internetový časopis (<http://www.elektrovue.cz/>), 2008, roč. 2008, č. 27, s. 1-7. ISSN: 1213- 1539.
- [14] POLÍVKA, Michal; PELKA, Tomáš; KYSELÁK, Martin. Using terminal server in education. In 9-th International Conference – Re-search in Telecommunication Technology RTT - 2008. 1. 2008. s. 1-7. ISBN: 978-80-227-2939- 0.
- [15] PELKA, Tomáš; POLÍVKA, Michal. AAA protokoly. Hakin9, 2008, roč. 2008, č. 3, s. 48-52. ISSN: 1214- 7710.
- [16] ŠKORPIL, Vladislav; MOLNÁR, Karol; NOVOTNÝ, Vít; JEŘÁBEK, Jan; HANUS, Stanislav; POLÍVKA, Michal; ČÍKA, Petr; ZEMAN, Václav.; ČÍŽ, Radim. Encyklopedie Teleinformatiky. Encyklopedie Teleinformatiky. VUT. Brno: VUT Brno, 2008. s. 1-751.

ŽIVOTOPIS

VZDĚLÁNÍ

- 2007–dosud** student Vysokého učení technického v Brně (<http://www.vutbr.cz/>), **Fakulta elektrotechniky a komunikačních technologií** – doktorský studijní program (elektrotechnika a komunikační technologie – obor teleinformatika).
- 2009** pětíměsíční stáž na Université Paris-Est, LIGM, Equipe A3SI, Computer Science Dept. **ESIEE Paris** (L'École d'ingénieurs des sciences et technologies de l'information et de la communication), Cité Descartes, BP 99, 93162 Noisy-le-Grand Cedex France, <http://www.esiee.fr/> – projekt v rámci doktorského studia.
- 2009–2010** doplňující **pedagogické studium** pro zaměstnance VUT.
- 2001–2007** Vysoké učení technické v Brně (<http://www.vutbr.cz/>), **Fakulta elektrotechniky a komunikačních technologií** – magisterský studijní program (elektrotechnika a informatika – obor elektronika a sdělovací technika).

ÚČAST NA PROJEKTECH

- 2011** člen projektového týmu „**Výzkum elektronických komunikačních systémů**“, FEKT-S-11-15, Vysoké učení technické v Brně – Vnitřní projekty VUT – plně financující (01. 01. 2012 – 31. 12. 2014).
- 2010** spoluřešitel projektu „**Simulátor transportních sítí**“, 3046/F1a, Ministerstvo školství, mládeže a tělovýchovy ČR – Fond rozvoje vysokých škol (FRVŠ) – plně financující (01. 01. 2010 – 31. 12. 2010).
- 2010** člen projektového týmu „**Výzkum komunikačních systémů a sítí**“, FEKT-S-10-16, Vysoké učení technické v Brně - Vnitřní projekty VUT – plně financující (01. 01. 2010 – 31. 12. 2010)
- 2009** hlavní řešitel projektu „**Inovace stávajících a příprava nových laboratorních úloh předmětu Služby telekomunikačních sítí**“, 1033/G1, Ministerstvo školství, mládeže a tělovýchovy ČR – Fond rozvoje vysokých škol (FRVŠ) – plně financující.

ABSTRACT

The Quality of Service (QoS) is in converged systems an important parameter. The dissertation thesis deals with research of QoS implementation into a newly developed network element. There was designed and implemented new protocol, based on the IP. The dissertation thesis deals with proposal of a new network element – the switch controlled by a neural network. During the research have been measured switches cross a performance classes. On the base of the measurement was designed the new four-port switch with switch fabric build on crossbar switch with an external control. The switch was designed with maximum QoS support. The switch fabric is controlled by the feedforward backpropagation neural network. The designed switch was modeled in the MATLAB and Simulink. The simulations prove that developed solution is functional.