

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

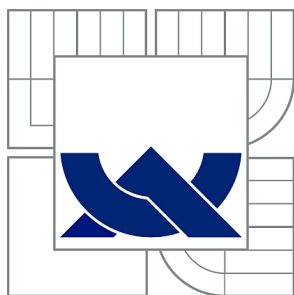
NÁVRH PŘEPÍNAČE VYUŽITELNÉHO V MODERNÍCH
KOMUNIKAČNÍCH SÍTÍCH

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

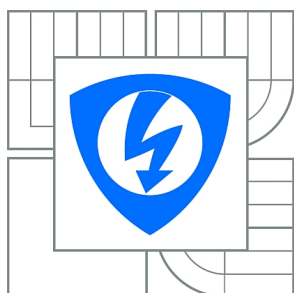
AUTOR PRÁCE
AUTHOR

Bc. LUBOMÍR MOJŽIŠ

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH PŘEPÍNAČE VYUŽITELNÉHO V MODERNÍCH KOMUNIKAČNÍCH SÍTÍCH

DESIGN OF SWITCH USED IN MODERN COMMUNICATION NETWORKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

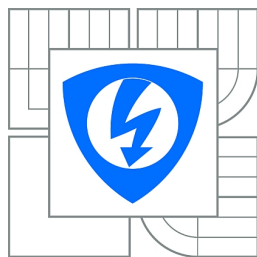
AUTOR PRÁCE
AUTHOR

Bc. LUBOMÍR MOJŽIŠ

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. VLADISLAV ŠKORPIL, CSc.

BRNO 2015



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Ľubomír Mojžiš

ID: 134562

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Návrh prepínače využiteľného v moderných komunikačných sítích

POKYNY PRO VYPRACOVÁNÍ:

Přepínače jsou jedním ze základních prvků moderních komunikačních sítí, seznámte se s jejich uspořádáním. Prostudujte architekturu přepínačů a zaměřte se na jejich výkonnost, podporu prioritního zpracování různých typů spojení a obecně na kvalitu služby QoS. Pro prioritní zpracování využijte standardní postupy a také neuronové sítě. Vytvořte algoritmus činnosti prvků a simulujte je pomocí simulačního programu MATLAB-SIMULINK, navrhnete model přepínače. Rozeberte různé alternativy optimalizace přepojování vstupních požadavků na výstupy. Výsledky připravte k využití v laboratorní výuce.

DOPORUČENÁ LITERATURA:

- [1] SKORPIL, V. Advanced Elements for Transmission Networks. The Journal of the Institute of Telecommunications Professionals, ISSN: 1447-4739, Volume 2, Part 2
- [2] SKORPIL, V. New Approach to Converged Networks. Proceedings of the Seventh International Conference on Networking. ICN 2008, April 13-18, Cancun,

Termín zadání: 9.2.2015

Termín odevzdání: 26.5.2015

Vedoucí práce: doc. Ing. Vladislav Škorpil, CSc.

Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato diplomová práce pojednává o přepínacích prvcích v moderní komunikační síti. V práci je popsána architektura přepínačů se zaměřením na kvalitu služeb a pomocí simulačního programu MATLAB-SIMULINK jsou navrženy a simulovány dva modely přepínače. První model je založen na klasickém přepínání a druhý je řízen neuronovou sítí. Na základě těchto modelů je v práci vytvořena laboratorní úloha vhodná pro výuku v oblasti komunikačních sítí.

KLÍČOVÉ SLOVÁ

přepínač, kvalita služeb, MAC adresa, Matlab, Simulink, DSCP, priorita, neuron, neuronová síť

ABSTRACT

This semestral paper deals with switching components of modern communication networks. A switch architecture focused on the quality of service is described in this paper and there are two switch models designed and simulated in the simulating program MATLAB-SIMULINK. The first model is based on classic switching and the other one is controlled by neural network. A laboratory exercise suitable for communication networks education is created in this paper on the basis of designed models.

KEYWORDS

switch, quality of service, MAC address, Matlab, Simulink, DSCP, priority, neuron, neural network

MOJŽIŠ, Lubomír *Návrh přepínače využitelného v moderních komunikačních sítích*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 79 s. Vedúci práce bol doc. Ing. Vladislav Škorpil, CSc.

PREHLÁSENIE

Prehlasujem, že som svoju diplomovú prácu na tému „Návrh přepínače využitelného v moderních komunikačních sítích“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím 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, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisejúcich s právom autorským a o zmene niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno

.....

(podpis autora)

POĎAKOVANIE

Rád by som sa poďakoval vedúcemu diplomovej práce doc. Ing. Vladislavu Škorpilovi, CSc. za metodickú pomoc pri spracovávaní diplomovej práce.

Brno

.....

(podpis autora)

POĎAKOVANIE

Výzkum popsaný v tejto diplomovej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno

.....
(podpis autora)

OBSAH

Úvod	12
1 Úvod do komunikačných sietí	13
1.1 Referenčný model komunikačných sietí	13
1.1.1 Fyzická vrstva	14
1.1.2 Spojová vrstva	14
1.1.3 Sieťová vrstva	15
1.1.4 Transportná vrstva	15
1.1.5 Relačná vrstva	15
1.1.6 Prezentačná vrstva	16
1.1.7 Aplikačná vrstva	16
1.2 Kvalita služieb v komunikačných sietí	17
1.2.1 Parametre komunikačnej siete	17
1.2.2 Mechanizmy zaistenia kvality služieb	18
1.2.3 Riadenie odosielania paketov	19
1.3 Výkonnosť aktívnych sieťových prvkov	24
2 Architektúra prepínača	25
2.1 Prepínač a QoS	26
2.1.1 Typ služby ToS	26
2.1.2 Diferencované služby	27
2.2 Klasifikácia architektúry prepínačov	29
2.2.1 Prepínače využívajúce časový multiplex	29
2.2.2 Prepínače využívajúce priestorový multiplex	31
2.3 Bloková schéma prepínača	33
3 Neurónové siete	34
3.1 Neurón	34
3.2 Modelovanie neurónu	35
3.3 Viacvrstvové neurónové siete	36
3.4 Využitie neurónových sietí v prepínacích prvkoch	37
4 Návrh prepínača	38
4.1 Topológia siete	38
4.2 Modelovanie topológie	40
4.2.1 Vstupná jednotka	41
4.2.2 Výstupná jednotka	42
4.2.3 Telo prepínača	43

4.3	Prepínač riadený neurónovou sieťou	46
4.3.1	Vytvorenie neurónovej siete	46
4.3.2	Aplikovanie neurónovej siete	49
5	Laboratórna úloha	51
5.1	Zadanie	51
5.2	Teoretický úvod	51
5.2.1	Prepínač a QoS	51
5.2.2	Topológia siete	53
5.2.3	Model prepínača	54
5.2.4	Neurónová sieť	57
5.3	Pracovný postup	58
5.4	Otázky	59
6	Vzorový protokol	60
6.1	Zadanie	60
6.2	Vypracovanie	60
6.2.1	Voľba DSCP hodnôt	60
6.2.2	Simulácia klasického prepínača	61
6.2.3	Analýza neurónovej siete a simulácia prepínača riadeného neurónovou sieťou	61
6.3	Odpovede na otázky	66
6.4	Záverečné zhodnotenie	66
7	Záver	67
	Literatúra	69
	Zoznam symbolov, veličín a skratiek	72
	Zoznam príloh	74
A	Obsah CD	75
B	Štruktúra prepínača	76

ZOZNAM OBRÁZKOV

1.1	Referenčný model ISO/OSI [3]	13
1.2	Porovnanie modelov ISO/OSI a TCP/IP 5	16
1.3	Riadenie odosielania paketov	19
1.4	Fronta typu FIFO	20
1.5	Prioritný systém front PQ	20
1.6	Prioritný systém front so spravodlivou obsluhou	21
1.7	Prioritný systém front s váženou cyklickou obsluhou	22
1.8	Systém front s váženou bitovou cyklickou obsluhou	23
2.1	Štruktúra Ethernetového rámca [6]	25
2.2	Hlavička IP paketu [11]	26
2.3	Štruktúra poľa DS	28
2.4	Architektúra prepínača so zdieľaným médiom	29
2.5	Architektúra prepínača so zdieľanou pamäťou	30
2.6	Krížový prepínač	31
2.7	Plne prepojené spojovacie pole	32
2.8	Bloková schéma prepínača [7]	33
3.1	Nervová bunka – neurón	34
3.2	Typy neurónových sietí [26]	36
4.1	Navrhnutá topológia	39
4.2	Topológia v prostredí Simulink	40
4.3	Vstupná dátová jednotka	41
4.4	Výstupná dátová jednotka	43
4.5	Telo prepínača	43
4.6	Zvýšenie priority	44
4.7	Funkcia hyperbolický tangens	46
4.8	Trénovanie neurónovej siete	48
4.9	Trénovacia chybová krivka	48
4.10	Riadenie prepínania pomocou neurónovej siete	49
4.11	Štruktúra neurónovej siete	49
4.12	Štruktúra vrstvy neurónovej siete	50
5.1	Štruktúra poľa DS	51
5.2	Topológia siete	53
5.3	Telo prepínača	56
5.4	Zvýšenie priority	56
5.5	Štruktúra neurónovej siete	57
5.6	Štruktúra vrstvy neurónovej siete	58
5.7	Nastavenie merania parametrov simulácie	59

5.8	Meranie doby dĺžky simulácie	59
6.1	Simulácia 1	61
6.2	Simulácia 2	61
6.3	Trénovanie neurónovej siete	62
6.4	Trénovacia chybová krivka	62
6.5	Parametre tréningu neurónovej siete	63
6.6	Regresia neurónovej siete	63
6.7	Porovnanie vstupných dát a dát naučených neurónovou sieťou	64
6.8	Simulácia 3 – prepínač riadený neurónovou sieťou	64
6.9	Simulácia 4 – prepínač riadený neurónovou sieťou	65
6.10	Závislosť doby simulácie na počte staníc dožadujúcich sa IPTV služby	65
B.1	Riadenie QoS	76
B.2	Klasifikácia rámcov	77
B.3	Riadenie prepínania	78
B.4	Váhové hodnoty neurónovej siete	79

ZOZNAM TABULIEK

2.1	Bežne používané DSCP hodnoty [12]	28
4.1	MAC adresy zariadení	39
4.2	Štruktúra vrstvy neurónovej siete	50
5.1	Bežne používané DSCP hodnoty	52
5.2	MAC adresy zariadení	54
6.1	Zvolené hodnoty DSCP	60
6.2	Doba simulácie pre počet staníc dožadujúcich sa IPTV služby	65

ÚVOD

Prepínače sú jedným zo základných prvkov moderných komunikačných sietí. Dnes sú počítačové siete nevyhnutnou súčasťou podnikania, školstva, štátnej správy a domácej komunikácie. Na rozsiahlej sieti internetu môžu okrem prenosu dát fungovať aj rôzne iné služby. Je nutné zaistiť nastavenie kvality služieb, aby podľa priority boli konkrétne služby vhodne uprednostňované.

V tejto práci sú navrhnuté dva modely 8-portového prepínača, ktorý podporuje kvalitu služieb. Prvý model je založený na klasickom prepínaní a druhý je riadený neurónovou sieťou. Návrh je realizovaný a simulovaný v prostredí MATLAB-SIMULINK. Navrhnutú topológiu tvoria okrem prepínača zariadenia, ktoré poskytujú služby vyžadujúce určitú kvalitu služieb.

Na základe tejto topológie je ďalej navrhnutá laboratórna úloha, v ktorej hlavnou úlohou je priradiť konkrétnej službe príslušnú hodnotu kvality tak, aby konkrétna služba bola adekvátne obslužená. Ďalej je v laboratórnej úlohe zahrnuté vytvorenie, tréning a testovanie neurónovej siete. Simulované sú oba modely prepínača. V laboratórnej úlohe je vypracovaná teória potrebná pre pochopenie činnosti prepínača, opis navrhnutej topológie, pracovný postup a otázky pre študentov. Pre vyučujúceho je zahrnutý vzorový protokol laboratórnej úlohy.

1 ÚVOD DO KOMUNIKAČNÝCH SIETI

1.1 Referenčný model komunikačných sietí

V prípade realizácii zložitého procesu je vhodné rozdelenie na skupiny jednoduchších procesov. Výsledná realizácia je potom omnoho pružnejšia, univerzálnejšia a rýchlejšia. Pri delení zložitého procesu je snaha vytvoriť skupiny tak, aby zahrňovali procesy navzájom blízke a aby skupina operácií bola čo najucelenejšia. Ucelenosť je dôležitá preto, aby jednotlivé skupiny boli čo najmenej závislé jedna na druhej a aby sa tak minimalizovala vzájomná komunikácia.

Z dôvodu zložitosti bol aj proces sieťovej komunikácie rozdelený na čiastkové skupiny. Bol vytvorený tzv. referenčný model skladajúci sa z jednotlivých vrstiev, kde každá vrstva predstavuje jednu skupinu operácií. [1]

Referenčný model ISO/OSI (*International Organization for Standardization/Open System Interconnection*) je najznámejší vrstvomý model sieťovej komunikácie popisujúci architektúru sietí. Model je definovaný normou ISO IS 7498 – prijatá v roku 1984. Jedná sa o sedemvrstvomý hierarchický model znázornený na obr. 1.1. [2]



Obr. 1.1: Referenčný model ISO/OSI [3]

V pravej časti sú názvy jednotlivých vrstiev a v ľavej sú príslušné dátové jednotky. Opis a funkcionality jednotlivých vrstiev a priradenie príslušných zariadení k vrstvám je uvedené nižšie.

1.1.1 Fyzická vrstva

Jedná sa o najnižšiu vrstvu modelu zaoberajúcu sa vlastným prenosom informácie fyzickým prostredím prostredníctvom elektromagnetického signálu. Je špecifikovaná parametrami:

- mechanickými – určujú mechanické vlastnosti fyzického prostredia, ktorým sa šíri elektromagnetický signál,
- elektrickými – určujú vlastnosti elektromagnetického signálu ako nosiča dát,
- funkčnými – význam signálov jednotlivých vodičov, párov, vlákien, prípadne určitých symbolov vo fyzickom rámci,
- procedurálnymi – jedná sa napríklad o identifikáciu obsadenosti fyzického spoja, aktiváciu a deaktiváciu fyzického spoja.

Medzi sieťové prvky pracujúce na fyzickej vrstve patria napríklad rozbočovač (*hub*), opakovač (*repeater*), modem, sieťová karta, atď. [2] Opakovač je zariadenie, ktoré má dva porty – vstupný a výstupný. Signál prijatý na vstupný port je regenerovaný na výstupný port. Jedná sa teda o tzv. opakovanie signálu. Rozbočovač obsahuje viacero portov – slúži teda na pripojenie viacero koncových zariadení do jedného sieťového bloku. Funkcionalita je taká, že signál prijatý na akýkoľvek port regeneruje na všetky ostatné porty okrem portu, na ktorý bol signál prijatý. Vďaka tomuto môže dôjsť ku kolíziám, ktorým je potrebné predchádzať. Hub je teda akýsi multiportový opakovač. Jeho výhodou je jednoduchá funkcionality, a teda nízka latencia v sieti. [4]

1.1.2 Spojová vrstva

Linková vrstva zaisťuje služby, ktoré riešia komunikáciu medzi susednými uzlami siete, prepojenými prvkami pracujúcimi na fyzickej vrstve. Je tvorená dvomi podvrstvami:

- podvrstva riadenia logického spoja – LLC (*Logical Link Control*)
- podvrstva riadenia prístupu ku komunikačnému kanálu – MAC (*Medium Access Control*)

Základnou jednotkou spojovej vrstvy je tzv. **rámec** (*frame*). Základný sieťový prvok pracujúci na linkovej vrstve je prepínač (*switch*), ktorého podrobný popis a realizácia je náplňou tejto práce. Prepínač na rozdiel od rozbočovača neregeneruje signál na všetky ostatné porty, ale iba na jeden príslušný port určený na základe cieľovej MAC adresy. Okrem fyzickej adresácie zaisťuje táto vrstva zabezpečenie dátových jednotiek, možnosť multiprotokolovej podpory, riadenie komunikácie, riadenie viacnásobného prístupu k zdieľanému médiu a iné. Bližší popis prepínača a procesu prepínania bude popísaný v ďalších kapitolách. [2]

1.1.3 Sieťová vrstva

Vyššia vrstva, ktorá je nezávislá na konkrétnej prenosovej technológii odpovedá sieťovej vrstve ISO/OSI. Často je taktiež nazývaná internetová vrstva, alebo IP vrstva podľa toho, že je realizovaná pomocou protokolu IP. Základnou jednotkou tejto vrstvy je **paket**. Účelom vrstvy je, aby sa jednotlivé pakety dostali od odosielaťa až k svojmu príjemcovi, spravidla cez medzilahlé smerovače (*router*), ktoré sú základnými sieťovými prvkami na tejto vrstve. [5]

Smerovač narozdiel od prepínača pracuje teda na tretej vrstve a je schopný korektne smerovať pakety na základe IP adresy, ktorá slúži na logické rozdelenie a adresovanie jednotlivých sieťových prvkov a komponentov. Smerovanie môže byť buďto statické (nastavené manuálne – administrátorom), alebo dynamické (zaistujú smerovacie protokoly). Podrobnejšie informácie sú nad rámec tohoto textu.

1.1.4 Transportná vrstva

Táto vrstva býva často označovaná taktiež ako TCP vrstva, pretože je najčastejšie realizovaná práve protokolom TCP (*Transmission Control Protocol*). Hlavnou úlohou tejto vrstvy je zaistiť prenos medzi dvoma koncovými účastníkmi, ktorými sú entity vyššej vrstvy. Podľa ich nárokov a požiadaviek môže transportná vrstva regulovať tok dát oboma smermi, zaisťovať spoľahlivosť prenosu a taktiež meniť nespojový charakter prenosu na spojový.

Ďalším používaným protokolom na tejto vrstve je UDP (*User Datagram Protocol*), ktorý narozdiel od TCP nezaistuje spoľahlivosť prenosu a charakter prenosu je nespojový. Je teda využívaný pri aplikáciách, ktoré nevyžadujú spoľahlivosť prenosu. [5] Dátovými jednotkami tejto vrstvy sú v prípade TCP protokolu **segment** a v prípade UDP protokolu **datagram**. Entity vyšších vrstiev (relácie) sa na jednotlivých sieťových adresách rozlišujú číslom TCP alebo UDP portu.

1.1.5 Relačná vrstva

Protokoly relačnej vrstvy riešia problematiku zaistenia korektného vedenia dialógov (relácií) aplikačných procesov. Vrstva realizuje naviazanie, udržovanie a rušenie relácií. Definuje typ komunikácie (simplex, poloduplex, duplex), zavádza synchronizačné body do prenášanej správy pre možné pokračovanie v prenose v prípade rozpadu a následnej obnovy relácie. Pri zobrazení relácie na transportné spojenie existuje najčastejšie pomer jedna k jednej, ale môže to byť aj viac k jednej a jedna k viacerým, čím sa napríklad zaistuje možnosť predania existujúceho transportného spojenia inej aplikácii. Ďalšou možnosťou je identifikácia komunikujúcich subjektov pre zaistenie bezpečnosti prístupu k informáciám.

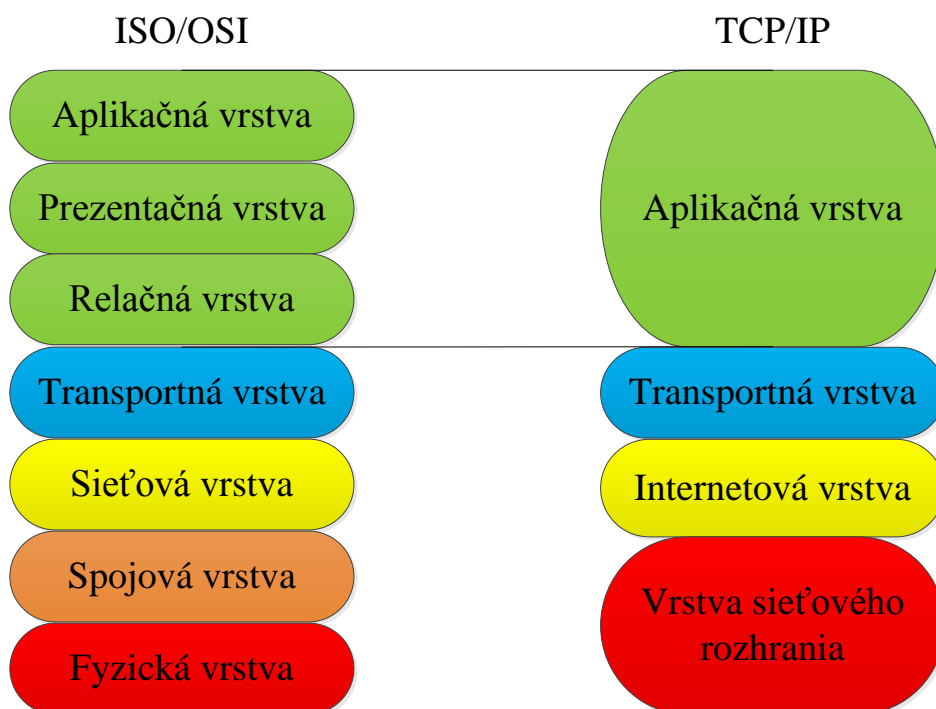
1.1.6 Prezentačná vrstva

Prezentačná vrstva zahŕňa niekoľko typov služieb – kódovanie, kompresiu dát a šifrovanie. Cieľom je predovšetkým upraviť formát správy do tvaru známeho obom komunikujúcim aplikačným programom. Spoločné formáty textu, čísel, statických obrázkov, audia a videa umožňujú komunikovať aplikáciám rôznych systémov s rôznou vlastnosťou prezentácií dátových typov. Kompresia zvyšuje prenosovú rýchlosť z pohľadu aplikačnej vrstvy. Šifrovanie poskytne prenos dôverných dát cez nedôveryhodné prostredie a zabezpečí proti odposluchu a modifikácii obsahu.

1.1.7 Aplikačná vrstva

Posledná vrstva ISO/OSI modelu implementuje protokoly tvoriace výkonné jadrá konkrétnych sieťových služieb a im odpovedajúcich aplikácií, napríklad pre prístup k webovým stránkam – HTTP (*Hyper Text Transfer Protocol*), prenos súborov – FTP (*File Transfer Protocol*), elektronická pošta – SMTP (*Simple Mail Transfer Protocol*) a iné. Aplikačná vrstva rieši teda problematiku užívateľov, sieťových zdrojov a synchronizáciu aplikácií. [2]

Okrem referenčného modelu ISO/OSI poznáme model TCP/IP. Rozdiely sú zrejmé z obr. 1.2. Fyzickú a spojovú vrstvu modelu ISO/OSI tvorí v TCP/IP jedna vrstva a posledné 3 vrstvy sú taktiež nahradené iba jednou vrstvou.



Obr. 1.2: Porovnanie modelov ISO/OSI a TCP/IP 5

1.2 Kvalita služieb v komunikačných sietí

Moderná komunikačná sieť poskytuje užívateľom široké spektrum služieb. Môže sa jednať o firemný informačný systém, zákaznícky riešené problémy pre online komunikáciu, špecifický software pre prenos hlasu, videa, atď. Každá služba má určité nároky na sieť, cez ktorú je poskytovaná. [7]

Kvalita služby QoS (*Quality of Service*) je termín pre riadenie dátových tokov. Cieľom QoS je rozlišovať typ prenosu a každému typu priradiť inú kvalitu. Kvalita služby je dôležitá napríklad pri real-time aplikáciách (VoIP, či video cez IP protokol), kedy je potrebné zaistiť, aby dátové jednotky boli doručené včas a bez strát. [8]

1.2.1 Parametre komunikačnej siete

V aplikáciách využívajúcich QoS sledujeme nasledujúce parametre:

- oneskorenie (*latency*),
- kolísanie oneskorenia (*jitter*),
- stratovosť,
- šírka pásma.

Oneskorenie (latencia) vzniká v konvergovaných sieťach ako dôsledok fyzikálnych vlastností šírenia signálu daným médiom. V rámci prenosu dát aktívnymi prvkami sa môže jednať o oneskorenie dané kódovaním a dekódovaním, zostavením rámca (resp. paketu), serializačné oneskorenie, oneskorenie spôsobené prekladaním, zabezpečovaním informácií, atď. Obvyklým zdrojom oneskorenia sú tzv. fronty v sieťových prvkoch. Oneskorenie patrí medzi najdôležitejšie parametre QoS používané pre špecifikáciu a zaistenie požiadaviek na prenos cez komunikačnú sieť. Predovšetkým pri prenose multimédií a v prípade ďalších služieb pracujúcich v reálnom čase je dodržanie určitej hodnoty oneskorenia kľúčové. Pre zaistenie dostatočnej kvality prenášaného audio signálu cez komunikačnú sieť bolo vypracované doporučenie ITU-T G.114, podľa ktorého má oneskorenie nasledovné veľkosti:

- 0-150 ms – akceptovateľné pre bežné užívateľské aplikácie,
- 150-140 ms – akceptovateľné pre medzinárodné hovory,
- viac ako 400 ms – neakceptovateľné pri projektovaní novej siete.

Kolísanie oneskorenia (*jitter*) je premenlivé oneskorenie, ktoré vzniká buď ako dôsledok chýb v konfigurácii siete alebo premenlivých podmienok na sieti. Často výraznejšie narastá pri preťažení siete, prípadne kvôli chybe sieťového prvku alebo jeho konfigurácie. Dôsledkom tohoto oneskorenia môže narastať stratovosť paketov, ktoré sú na koncových staniciach zahadzované kvôli doručení mimo poradia. *Jitter* je možné potlačiť pomocou vyrovnávajúcej pamäti. Voľba veľkosti vyrovnávajúcej pamäti do určitej miery ovplyvňuje kvalitu hovoru.

Stratovosť je jedným z ďalších parametrov ovplyvňujúcich QoS. Udáva sa v percentách. Príčinou stratovosti môže byť:

- jitter presahujúci medze podľa doporučenia,
- nestabilná sieť,
- vady na fyzickej vrstve, ktoré nemajú za následok úplné vyradenie linky z prevádzky,
- preťaženie siete.

Príčinou preťaženia siete býva nedostatočný výkon sieťových prvkov. Pripustnosť prvkov nie je daná rýchlosťou ich rozhraní, ale vnútornou vybavenosťou. [7] [9]

1.2.2 Mechanizmy zaistenia kvality služieb

Rada predovšetkým starších sieťových technológií založených na prepínaní rámcov (resp. paketov) poskytuje všetkým dátovým jednotkám rovnaký spôsob zachádzania. Tento prístup k spracovaniu dátových jednotiek je označovaný *Best-effort*. V rámci spracovania sa sieťový prvok snaží vyhovieť žiadosti dátových jednotiek o ich predanie na správny výstup, ale bez akýchkoľvek garancií. Preto nemôže zaručiť žiadne časové medze pre spracovanie dátovej jednotky, dokonca ani samotné prevedenie prepojenia dátovej jednotky.

Z toho dôvodu vyplýva, že nedostatok sieťových prostriedkov má rovnaký vplyv na všetky dátové toky, čo ale môže vyvolať veľmi odlišný dopad pre rôzne typy sieťových služieb. Napríklad v prípade prenosu veľkého súboru má strata jedinej dátovej jednotky vplyv na celý súbor, ktorý môže byť znehodnotený. Naopak pri prenose multimédií je určitá strata akceptovateľná. Je teda potrebné zaistiť rozoznanie dátových služieb a následne ich zachádzanie.

Pre podporu rôznych tried kvality služieb sa na sieťovej vrstve používajú dva mechanizmy:

- IntServ (*Integrated Services*),
- DiffServ (*Differentiated Services*).

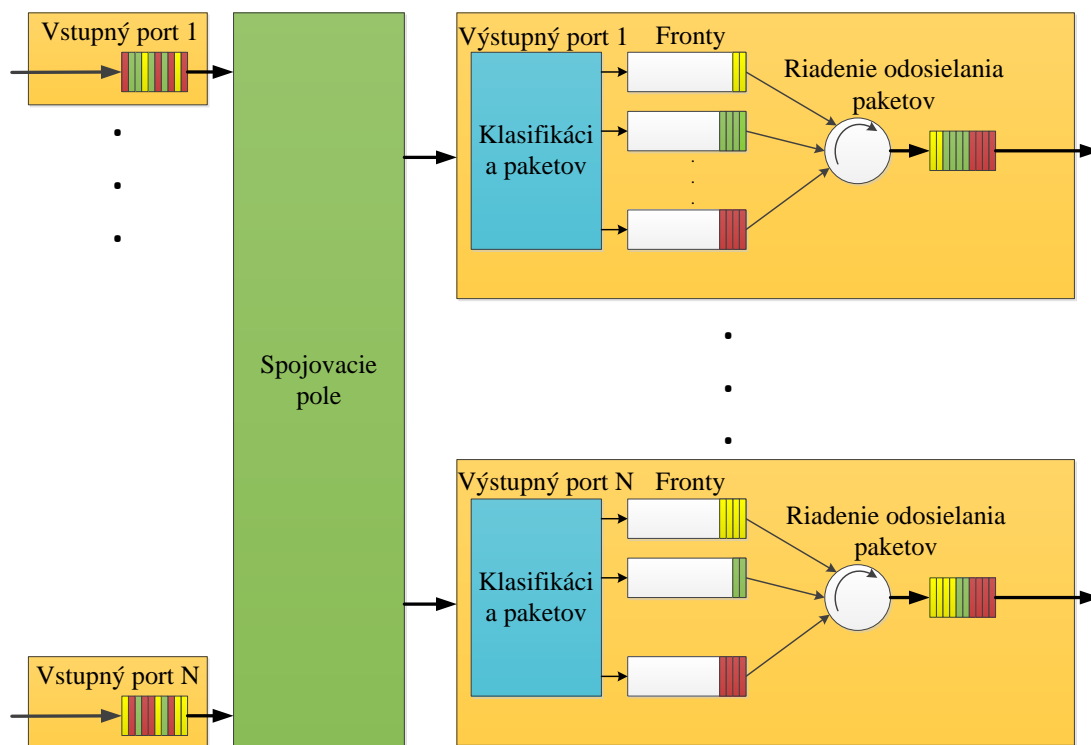
Technológia Integrovaných služieb je schopná rozlíšiť každý dátový tok na základe identifikátoru odosielateľa a príjemcu – sieťová adresa, identifikátor užívateľskej služby. Preto je schopná zaistiť riadenie kvality služieb po celej trase od zdroja až k cieľovej stanici.

Diferencované služby rozdeľujú prevádzku do niekoľko tried a potom zaistujú odlišné zachádzanie pre jednotlivé triedy. Tento mechanizmus preto nie je schopný garantovať parametre pre jednotlivé dátové toky, ale oveľa menej zťažuje procesor aktívneho prvku a je výrazne lepšie škálovateľný. [9]

V tejto práci je pri návrhu a implementácii sieťového prvku v zaistení QoS použitý mechanizmus diferencovaných služieb, viď nasledujúce kapitoly.

1.2.3 Riadenie odosielania paketov

Základom k zaisteniu odlišného zachádzania rôznych dátových tokov je radenie paketov do oddelených front a diferencovaný spôsob odosielania paketov z týchto front. Okrem samotného odosielania paketov podľa príslušného prioritného mechanizmu je ďalšou dôležitou úlohou dohľad nad dostupnými sieťovými prostriedkami, predovšetkým nad šírkou pásma výstupného portu. Pretože technológia prepínania paketov je založená na statickom multiplexe paketov, nedá sa zaručiť, aby nedochádzalo ku krátkodobému prekročeniu kapacity výstupnej linky. V takýchto prípadoch sú pakety s nižšou prioritou zadržané vo frontách a uprednostnené sú pakety s vyššou prioritou. Základná funkcia riadenia odosielania paketov je znázornená na obr. 1.3. Je teda vidno, že plánovanie prevádza každý výstupný port samostatne.



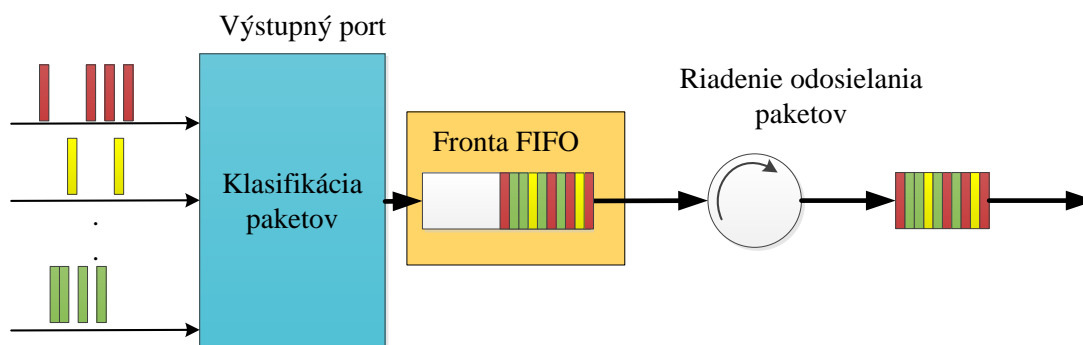
Obr. 1.3: Riadenie odosielania paketov

Poznáme nasledujúce metódy riadenia odosielania paketov:

- fronta typu FIFO (*First In First Out*),
- prioritný systém front PQ (*Priority Queuing*),
- systém front so spravodlivou obsluhou FQ (*Fair Queuing*),
- systém front s váženou cyklickou obsluhou WRR (*Weighed Round Robin*),
- systém front s váženou spravodlivou obsluhou WFQ (*Weighted Fair Queuing*),
- systém založený na triedach s váženou spravodlivou obsluhou CB WFQ (*Class-Based Weighted Fair Queuing*). [9]

Fronta typu FIFO

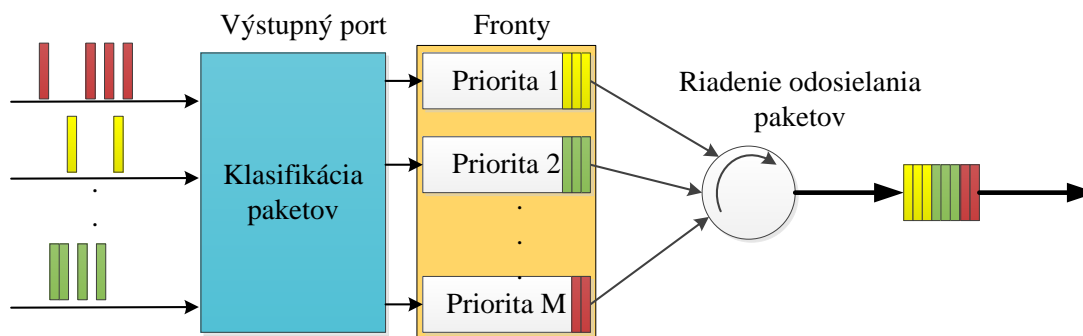
Táto fronta je historicky najstarším a základným typom v prípade, že nie je implementovaný žiadny špecifický algoritmus pre riadenie odosielenia. Pakety sú riadené do jednej fronty v takom poradí, v akom boli prijaté a v rovnakom poradí sú potom odoslané na výstupný port. Princíp tejto metódy je zobrazený na obr. 1.4. Najväčšou výhodou tohoto typu front je, že implementácia je veľmi jednoduchá a nevyžaduje žiadny algoritmus riadenia. Pretože tento spôsob obsluhy zaisťuje rovnaké zachádzanie pre všetky pakety, najviac vyhovuje pre sieťové technológie typu *Best-effort*. [9]



Obr. 1.4: Fronta typu FIFO

Prioritný systém front – PQ

Prioritný systém front predstavuje jednoduchý spôsob diferenciácie medzi jednotlivými triedami prevádzky. Ako vidno z obr. 1.5, pri tejto metóde existuje M rôznych front. Poradie odosielenia paketov potom odpovedá prioritě pridelenej fronte, v ktorej sa paket nachádza. Fronty s vyššou prioritou majú teda prednosť pred frontami s nižšou prioritou.



Obr. 1.5: Prioritný systém front PQ

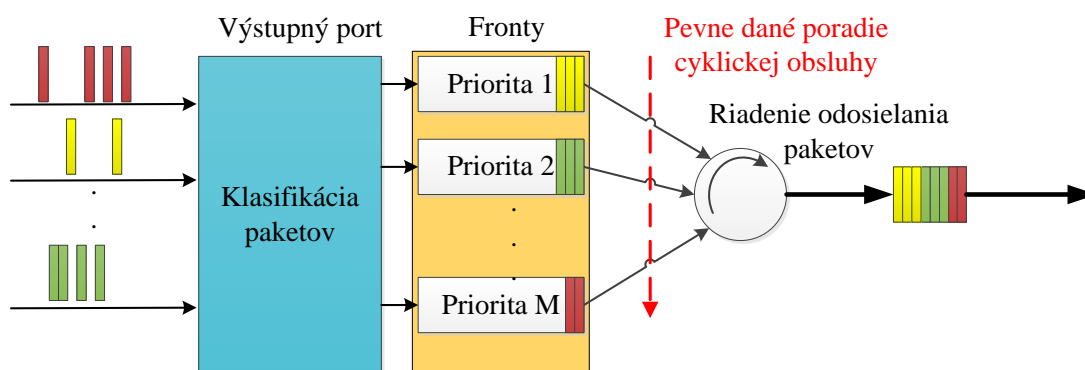
Podobne ako pri FIFO je aj pri tejto metóde výhodou jednoduchosť algoritmu riadenia. Naopak nevýhodou je potlačenie prevádzky s nižšou prioritou. V prípade, že fronty s vyššou prioritou sú doplňované pravidelne, fronty s nižšou prioritou tak strácajú možnosť odoslania paketov. V prípade vysokej latencie paketov môžu byť pakety považované za stratené a môžu byť poslané znovu, čo ešte viac zataží sieť.

Prioritné fronty je teda vhodné využívať v prípade, keď prevádzka s vysokou prioritou predstavuje iba malú časť celkovej prevádzky. Príkladom môžu byť dáta s vysokou dôležitosťou vyžadujúce okamžité spracovanie, ale naopak s nízkym objemom. Využitie je teda vhodné v prevádzke vyžadujúcej prenos dát v reálnom čase – prenos hlasu, videa a podobne. [9]

Systém front so spravodlivou obsluhou – FQ

Jedným z ďalších spôsobov diferenciácie tried prevádzky zaisťuje systém front so spravodlivou obsluhou, viď obr 1.6. V tomto systéme sú pakety taktiež radené do M front. Riadenie odosielania paketov cyklicky obsluhuje jednotlivé fronty, ktoré majú pakety k odoslaniu (algoritmus Round-robin). Behom každého cyklu je z jednej fronty odoslaný maximálne jeden paket. Implementácia tejto metódy je pomerne jednoduchá, pretože nevyžaduje komplexný algoritmus pre alokáciu šírky pásma jednotlivým frontám.

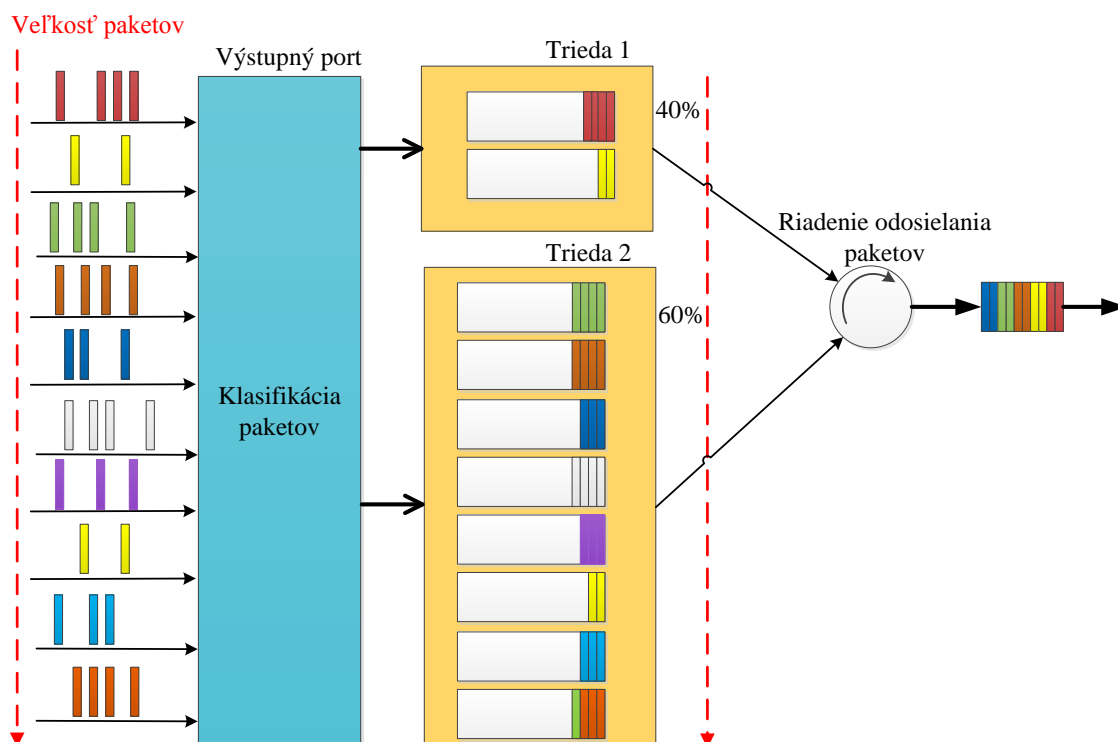
Systém obsluhy má ale dve nevýhody. Prvá z nich spočíva v tom, že každá fronta je obslužená s rovnakou pravidelnosťou. V prípade, že dátové toky majú odlišné požiadavky na šírku pásma, systém nebude schopný rozdeliť dostupnú šírku v takom relatívnom pomere, ako to dané toky vyžadujú. Druhým nedostatkom je, že behom obsluhy front je prenesený paket bez ohľadu na jeho veľkosť. Dátový tok teda môže obsadiť väčšiu šírku pásma v prípade väčšieho paketu. [9]



Obr. 1.6: Prioritný systém front so spravodlivou obsluhou

System front s váženou cyklickou obsluhou – WRR

Tento systém môže eliminovať nedostatok predchádzajúcej metódy obsluhy spočívajúcej v neschopnosti rozdelenia šírky pásma výstupného portu podľa požiadaviek tried prevádzky. Mechanizmus WRR je schopný zaistiť rôznym triedam rôzne veľkú časť dostupnej šírky pásma. Model riadenia fronty WRR je uvedený na obr. 1.7. Dátové toky sú najskôr rozdelené do niekoľkých tried a šírka pásma výstupného portu je rozdelená medzi tieto triedy úmerne k váhovej hodnote priradenej danej triede. Pritom súčet všetkých váhových hodnôt odpovedá 100 %.



Obr. 1.7: Prioritný systém front s váženou cyklickou obsluhou

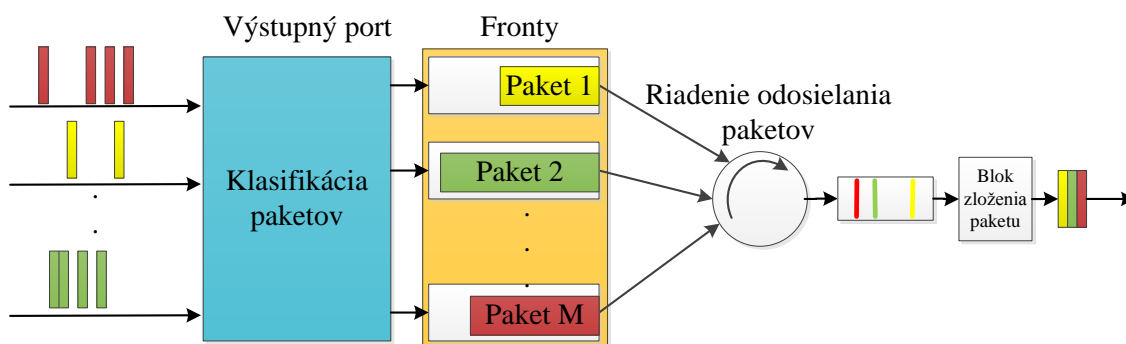
Ďalej je odoslanie paketov jednotlivých dátových tokov v jednej triede riadené obsluhou FQ. Ako vidíme na obr. 1.7, WRR využíva dvojúrovňové cyklické plánovanie Round-Robin. Prvá úroveň reprezentuje výber medzi jednotlivými triedami a druhá úroveň je využívaná v rámci konkrétnej triedy.

Pomerové delenie šírky pásma medzi jednotlivé triedy sa dá zaistiť percentuálnym rozdelením času tráveného obsluhou u danej triedy z obr. 1.7 vidíme, že cez daný odchádzajúci port vedie niekoľko dátových tokov.

Úlohou je zaistiť percentuálne rozdelenie šírky pásma medzi jednotlivé toky. Z obrázku je vidno, že 40 % šírky pásma je určených pre triedu 1 a zvyšných 60 % pre triedu 2. [9]

Systém front s váženou spravodlivou obsluhou – WFQ

Aj keď mechanizmus WRR čiastočne rieši nedostatky mechanizmu FQ, nevyrieši problém spočívajúci vo vplyve dĺžky paketu na šírku pásma, ktorú dátový tok bude mať k dispozícii. Fronta s váženou spravodlivou obsluhou WFQ je schopná tento vplyv eliminovať. Mechanizmus sa snaží odosielať pakety tak, aby odoslanie skončilo v dobe vypočítanej podľa pomocného teoretického modelu zohľadňujúceho dĺžku paketu. Tento model sa označuje ako vážená bitová cyklická obsluha a je znázornený na obr. 1.8. Riadenie odosielania obsahuje cyklicky každú frontu, ale jednotlivé pakety posiela na výstup bit po bite. Tieto bity sú potom zachytené v bloku zloženia paketu a následne sú odoslané na výstupný port. Preto väčší paket musí čakať dlhšiu dobu, aby bol znovu zložený. Ako už bolo uvedené, jedná sa o teoretický model, priama praktická realizácia by bola príliš neefektívna. [9]



Obr. 1.8: Systém front s váženou bitovou cyklickou obsluhou

Systém založený na triedach s váženou spravodlivou obsluhou – CB WFQ

Tento model je veľmi podobný modelu WRR, viď 1.7. Prichádzajúca prevádzka je taktiež rozdelená do niekoľkých tried a šírka pásma odchádzajúceho portu je rozdelená medzi triedy podľa váhovej hodnoty pridelenej týmto triedam. Súčet všetkých váhových hodnôt odpovedá celkovej šírke pásma, t.j. 100 %.

Rozdiel medzi WRR a CB WFQ je v obsluhu dátových tokov v rámci tried. U mechanizmu WRR boli dátové toky v rámci triedy obsluhované systémom FQ. V prípade mechanizmu CB WFQ sa pre obsluhu dátových tokov využíva systém WFQ.

Pre príklad stanovenia poradia odoslania paketov pre tento mechanizmus môžeme predpokladať 3 fronty. Ďalej predpokladajme 3 pakety rôznej veľkosti čakajúce v jednotlivých frontách. Mechanizmus nám presne stanoví, v akom poradí budú jednotlivé pakety odoslané. [9]

1.3 Výkonnosť aktívnych sieťových prvkov

Jednotka, ktorá vyjadruje rýchlosť, šírku pásma (*bandwidth*) a priepustnosť aktívneho prvku je bit za sekundu. Výkonnostné charakteristiky aktívneho prvku sa dajú popísať i ďalšími jednotkami. Existuje prevod medzi týmito jednotkami, v praxi ale vyjadrujú to isté. Prenosová rýchlosť obvykle udáva teoretické maximálne hodnoty dané použitou prenosovou technológiou. Ostatné jednotky sa obvykle vzťahujú na konkrétnu veľkosť rámca, resp. paketu a udávajú reálne výkony aktívnych prvkov za udaných podmienok.

V praxi sa používajú nasledovné jednotky:

- rámcová rýchlosť,
- paketová rýchlosť,
- počet spojení za sekundu,
- celkový počet súbežných spojení,
- počet transakcií.

Rámcová rýchlosť (*frame rate*) – jednotkou je rámec za sekundu. Udáva, koľko rámcov je aktívny prvok reálne schopný spracovať a priradiť zo vstupu na výstup. Rámcovú rýchlosť dostaneme vydelením prenosovej rýchlosti a veľkosti rámca. S narastajúcou veľkosťou rámca nám teda rámcová rýchlosť klesá. Reálna rámcová rýchlosť závisí na návrhu aktívneho prvku a výkonnosti všetkých jeho častí. Je taktiež závislá na charakteristike spracovanej prevádzky.

Paketová rýchlosť (*packet rate*) je obdobou rámcovej rýchlosti na tretej vrstve ISO/OSI modelu. Reálna paketová rýchlosť aktívneho prvku závisí na spracovávanom protokole, aktívnych službách na danom prvku atď.

Počet spojení za sekundu – udáva rýchlosť, resp. počet spojení, ktorý prvok schopný zostaviť za 1 s. Zostavovanie spojenia zaisťuje softwarová časť architektúry sieťového prvku. Samotné predávanie paketov, ktoré po zostavení spojenia nasleduje, je naopak obvykle hardwarovo akcelerované a prvok nijak zvlášť nezatažuje.

Celkový počet súbežných spojení – udáva, koľko spojení je zariadenie schopné súčasne spracovávať. Závisí najmä na veľkosti pamäti aktívneho prvku.

Počet transakcií – udáva počet akcií realizovaných za sekundu.

Aktívny prvok by mal byť navrhnutý tak, aby každý z jeho parametrov korešpondoval s ostatnými, aby sa nemohlo stať, že sa z niektorej vlastnosti stane zraniteľnosť – napríklad otvorením väčšieho počtu spojení, než je schopný prvok obslúžiť.

Výkon aktívnych prvkov je závislý jak na návrhu aktívneho prvku, tak aj na aktívnych službách. Rovnaký prvok s rôznou konfiguráciou môže dosahovať veľmi odlišné výsledky. Pri rovnakej konfigurácii sa môžu v štandardných prevádzkových podmienkach výsledky aktívneho prvku s rôznou prevádzkou líšiť – rámce resp. pakety mávajú rôznu veľkosť. [7]

2 ARCHITEKTÚRA PREPÍNAČA

Ako už bolo spomínané, prepínač pracuje na druhej (spojovej) vrstve referenčného modelu ISO/OSI. Na rozdiel od fyzickej vrstvy sú operácie na spojovej vrstve viazané na prenášané dátové jednotky – rámce, jak je spomínané v kapitole 1.1. V lokálnych počítačových sieťach je najčastejšie reprezentovaná rôznymi verziami technológie Ethernet. Ako už bolo spomínané, Ethernet rámce obsahujú okrem užívateľských dát aj zdrojovú a cieľovú adresu. [1] Štruktúra Ethernetového rámca je zobrazená na obr. 2.1. Rámec začína hlavičkou, ktorá obsahuje zdrojovú a cieľovú MAC adresu a typ



Obr. 2.1: Štruktúra Ethernetového rámca [6]

rámca. Ďalej obsahuje dátové pole (*Payload*) a kontrolnú postupnosť rámca FCS (*Frame Control Check*), čo je 4-bajtový cyklický redundantný súčet používaný pre detekciu poškodenia dát pri prenose. [6] Na základe MAC adres je teda možné identifikovať jednotlivé spojenia a prevádzať rôzne typy filtrovania prijatých dátových jednotiek. Vykonávané funkcie prepínača sú založené prevažne na odchádzajúcich a prichádzajúcich hardwarových adresách a sú implementované hardwarovo. Prepínač sleduje a vytvára databázu adres staníc. Databáza sa tvorí na základe adresy odosielateľa získanej z prijatého rámca. Prepínač skontroluje databázu a v prípade, že sa daná adresa nenachádza v databáze, bude tam následne priradená.

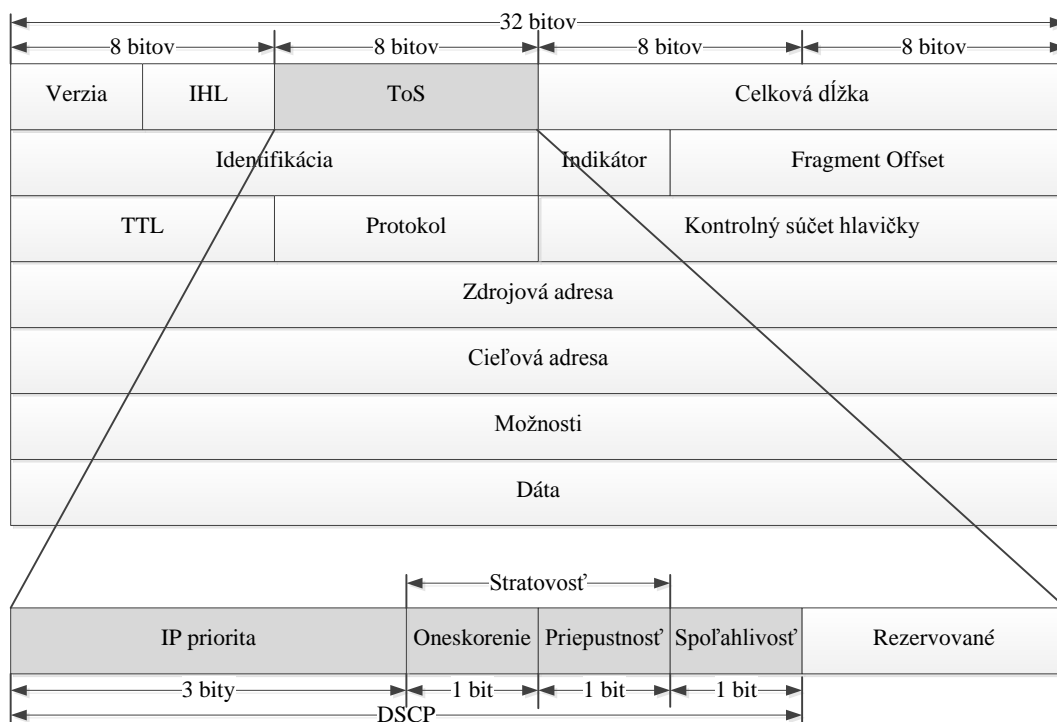
Dôležitou vlastnosťou prepínačov využívaných v sieti Ethernet je rozdelenie kolíznej domény. Kolízna doména označuje skupinu počítačov, ktoré sú prepojené buď priamo, alebo pomocou rozbočovača. V takomto prípade dáta vyslané z ľubovoľnej stanice sa môžu bez obmedzenia šíriť ku všetkým ostatným staniciam v doméne. Najväčšou výhodou tohoto riešenia je, že nezabráni šíreniu neplatných rámcov, ktoré najčastejšie vzniknú výsledkom kolízií. Neplatné rámce tak zbytočne blokujú nielen stanice, ktoré spôsobili kolízie, ale aj ostatné stanice v kolíznej doméne. Pretože prepínače prevádzajú kontrolu a filtrovanie prijatých dátových jednotiek, tak v prípade kolízie zistia, že sa jedná o porušené dáta a ignorujú ich. Zabraňuje tak šíreniu neplatných rámcov v sieti, a tak i zbytočnému blokovaniu ostatných staníc. Nárastom počtu staníc v jednej kolíznej doméne narastá i pravdepodobnosť výskytu kolízie, preto je snaha mať počet staníc v jednej kolíznej doméne čo najnižší. [1]

2.1 Prepínač a QoS

V dnešných moderných komunikačných sieťach poznáme prepínače, ktoré sú schopné pracovať taktiež na tretej vrstve modeli ISO/OSI, tzv. L3 prepínače. Boli vyvinuté pre zvýšenie výkonu smerovania vo väčšej sieti. Kľúčovým rozdielom medzi L3 prepínačom a smerovačom spočíva v technológii. Hardware L3 prepínača je postavený na princípe klasického prepínača pracujúceho na druhej vrstve. Je to teda vlastne prepínač, ktorý bol zdokonalený tým, že je schopný prevádzať smerovanie. Smerovač je zariadenie, ktoré je navrhnuté pre metódy smerovania. Sú navrhnuté tak, aby bolo možné použitie ako rozhranie medzi LAN (*Local Area Network*) a WAN (*Wide Area Network*) sieťou, a teda zvyčajne majú WAN rozhranie pre pripojenie do WAN siete. L3 prepínač z dôvodu spomínanej funkcionality a technológie toto rozhranie nemá. [10]

2.1.1 Typ služby ToS

Prepínač je teda schopný pracovať aj na tretej vrstve, a teda je schopný spracovávať pakety. Hlavička IP paketu je zobrazená na obr. 2.2. Zvýraznené je pole ToS (*Type*



Obr. 2.2: Hlavička IP paketu [11]

of Service), ktoré nám zaistuje značenie paketov, a teda kvalitu služieb. Pole ToS je 8-bitové, z toho 3 bity nám určujú prioritu podľa špecifikácie typu služby. Ďalšie

3 bity určujú požiadavky na prenos a zvyšné 2 bity sú nepoužívané a musia mať nastavenú hodnotu 0.

Bolo definovaných 8 úrovní (IP precedence), kde vyššie hodnoty znamenajú vyššiu prioritu:

- routine (0) – rutinná práca, zachádzanie typu Best-effort,
- priority (1) – pre správy, ktoré vyžadujú rýchle spracovanie,
- immediate (2) – pre správy dôležité z hľadiska bezpečnosti štátu, či armády,
- flash (3) – varovné a ďalšie extrémne dôležité správy,
- flash override (4) – správy rezervované pre využitie v prípade prelomenia obrany, či nukleárneho útoku,
- CRITIC/ECP (*Critical and Emergency Call Processing*) – pre správy posielané autorizovanými zdrojmi zo štátnych služieb civilnej obrany,
- internetwork control (6) – pre správy určené pre riadenie hraničných prvkov siete a spoluprácu s inými sieťami,
- network control (7) – pre správy spojené s riadením siete.

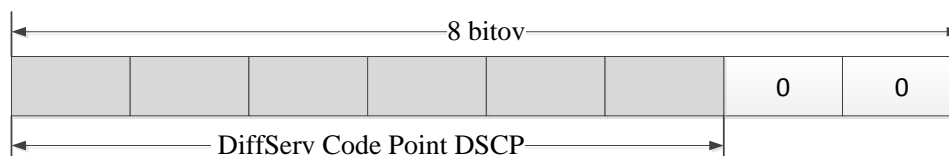
Ďalšie 3 bity poľa ToS signalizujú očakávaný spôsob spracovania daného IP paketu:

- 3. bit (D) – indikácia požiadavky na oneskorenie (*Delay*) vybranej trasy:
 - 0 = normálne oneskorenie
 - 1 = malé oneskorenie
- 4. bit (T) – indikácia požiadavky na priepustnosť (*Throughput*) vybranej trasy:
 - 0 = normálna priepustnosť
 - 1 = vysoká priepustnosť
- 5. bit (R) – indikácia požiadavky na spoľahlivosť (*Reliability*) vybranej trasy:
 - 0 = normálna spoľahlivosť prenosu
 - 1 = vysoká spoľahlivosť prenosu

Bity 6 a 7 sú rezervované pre budúce použitie. Pole ToS bolo pôvodne určené pre účely spracovania paketov, nebolo však bežne používané. Neskôr sa začalo používať pole pre podporu technológie DiffServ. [8] [9]

2.1.2 Diferencované služby

U mechanizmu diferencovaných služieb je spracovanie prevádzky riadené relatívnymi prioritami priradených jednotlivým triedam prevádzky. Špecifikácia uvádza nové pole DS (*Differentiated Services*), viď obr. 2.3. Zahrnutie tohoto poľa v rámci IP paketu je zrejmé z obr. 2.2. Prvých 6 bitov tohoto poľa je označovaných ako DSCP (*DiffServ Code Point*), zvyšné 2 bity sú taktiež nevyužívané. Na základe hodnoty DSCP je paketom zaistený požadovaný spôsob zachádzania – PHB (*Per-Hop Behavior*).



Obr. 2.3: Štruktúra poľa DS

Ľavý krajný bit tohoto poľa sa označuje 0. bit, ktorý je najvýznamnejší a naopak pravý bit je najmenej významný a označuje sa 5. bit. Týmito bitmi je teda možno vyjadriť 64 rôznych hodnôt. Zoznam bežne používaných hodnôt je uvedený v tab. 2.1. AF (*Assured Forwarding*) zaisťuje zaručené preposielanie dát, EF (*Expe-*

Tab. 2.1: Bežne používané DSCP hodnoty [12]

DSCP hodnota	Decimálna hodnota	Význam	Ekvivalentná IP priorita
000 000	0	BE	000
101 110	46	EF	101
001 010	10	AF11	001
001 100	12	AF12	001
001 110	14	AF13	001
010 010	18	AF21	010
010 100	20	AF22	010
010 110	22	AF23	010
011 010	26	AF31	011
011 100	28	AF32	011
011 110	30	AF33	011
100 010	34	AF41	100
100 100	36	AF42	100
100 110	38	AF43	100
001 000	8	CS1	1
010 000	16	CS2	2
011 000	24	CS3	3
100 000	32	CS4	4
101 000	40	CS5	5
110 000	48	CS6	6
111 000	56	CS7	7

ited Forwarding) znamená urýchlené preposielanie, zaisťuje nízku stratovosť, nízke oneskorenie a nízke kolísanie oneskorenia. CS (*Class Selector*) je označenie DSCP hodnoty, ktoré odpovedá určitej IP priorite a BE (*Best-effort*) je prednastavená (default) priorita, ktorá nám ale nezaručuje, že všetky dáta budú doručené. [8] [9]

2.2 Klasifikácia architektúry prepínačov

Na základe využitej technológie delíme prepínače nasledovne:

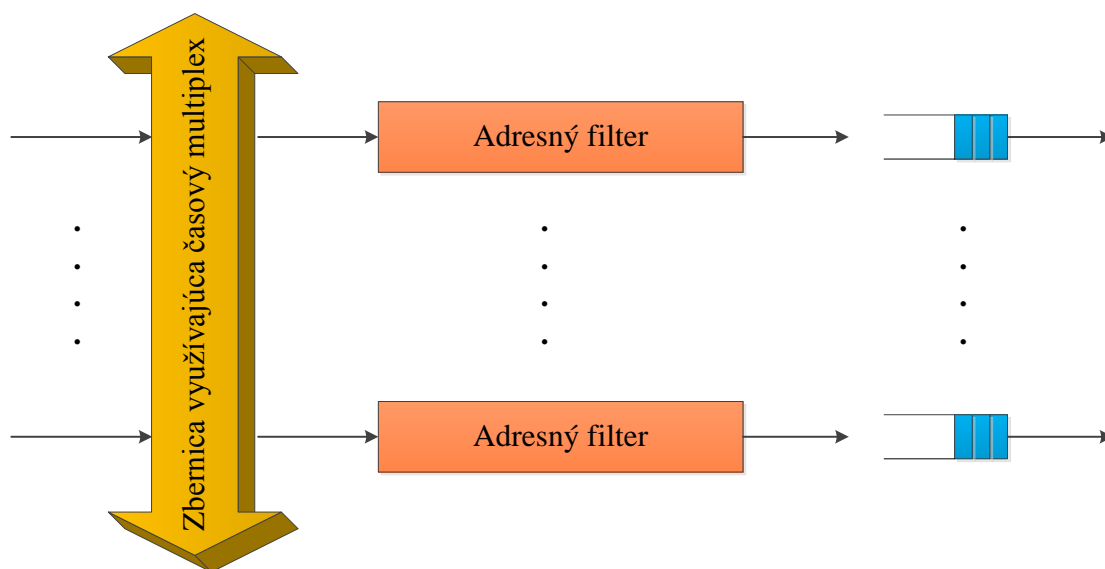
- prepínače využívajúce časový multiplex
 - prepínače využívajúce zdieľané médium
 - prepínače využívajúce zdieľanú pamäť
- prepínače využívajúce priestorový multiplex
 - jednocestné
 - viacestné

2.2.1 Prepínače využívajúce časový multiplex

Vnútrotná komunikačná štruktúra prepínačov tohoto typu je zdieľaná medzi všetky bunky prechádzajúce od vstupného portu na výstupný. Vnútrotná komunikačná štruktúra môže byť fyzicky zbernica, kruhová zbernica alebo pamäťový priestor. Hlavným problémom tejto štruktúry je obmedzenie rýchlosti vyplývajúce z maximálnej kapacity vnútornej komunikačnej štruktúry.

Prepínače využívajúce zdieľané médium

Vstupné porty prepínača sú pripojené k zdieľanému médiu pomocou časového multiplexu. Časový interval je teda rozdelený na úseky, pričom každý je pridelený príslušnému portu. Priepustnosť zdieľaného média teda určuje kapacitu prepínača. Architektúru takéhoto prepínača môžeme vidieť na obr. 2.4. Každá výstupná linka je



Obr. 2.4: Architektúra prepínača so zdieľaným médiom

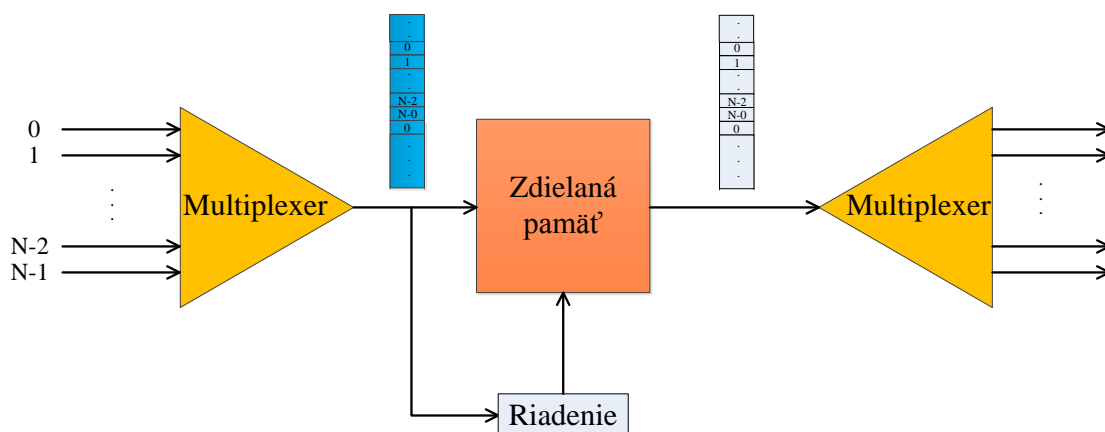
pripojená na zdieľané vysokorýchlostné médium cez adresný filter a vyrovnávaciu

pamäť FIFO. K bunke je bežne pripojená hlavička, ktorá obsahuje bitovú mapu výstupných portov. Nastavením bitu príslušného portu je potom signalizované, že bunka je určená práve pre tento výstupný port. Adresný filter overuje túto bitovú mapu buniek objavujúcich sa na zdieľanom médiu a v prípade splnenia kritérií filtrovania bunku skopíruje do vyrovnávajúcej pamäti.

Riadenie prepínača je decentralizované, a preto každý výstupný port môže pracovať samostatne, nezávisle na ostatných. Prepínače tohoto typu sa často využívajú v telekomunikačných systémoch, kde je šírka pásma vstupných liniek pomerne malá. [13]

Prepínače využívajúce zdieľanú pamäť

Prichádzajúce bunky sú radené do časového multiplexu a sú postupne zapisované do zdieľanej pamäteovej oblasti. Prepojovanie je potom realizované načítavaním buniek vo forme multiplexovaného toku dát, ktorý je potom demultiplexovaný a vedený k jednotlivým výstupom. Riadenie čítania a zápisu z pamäti je realizované modulom riadenia, na základe informácií získaných z hlavičky buniek. Architektúra tohoto typu je zobrazená na obr. 2.5. Všetky vstupy a výstupy zdieľajú jednu pamäťovú



Obr. 2.5: Architektúra prepínača so zdieľanou pamäťou

oblasť. Poznáme dve metódy zdieľania pamäťového priestoru medzi porty:

- úplné rozdelenie (*complete partitioning*)
- plné zdieľanie (*full sharing*)

V prípade úplného rozdelenia je pamäťový priestor rozdelený do N rovnako veľkých častí, kde každá časť je pridelená k jednému z výstupných portov. Bunka je potom zapísaná do pamäti podľa toho, pre ktorý výstupný port je určená.

V prípade plného rozdelenia je celý pamäťový priestor zdieľaný všetkými výstupnými portmi. Toto riešenie ponúka efektívnejšie využitie pamäte a vyžaduje algoritmus na dozeranie využitia pamäte medzi jednotlivými portmi. [13]

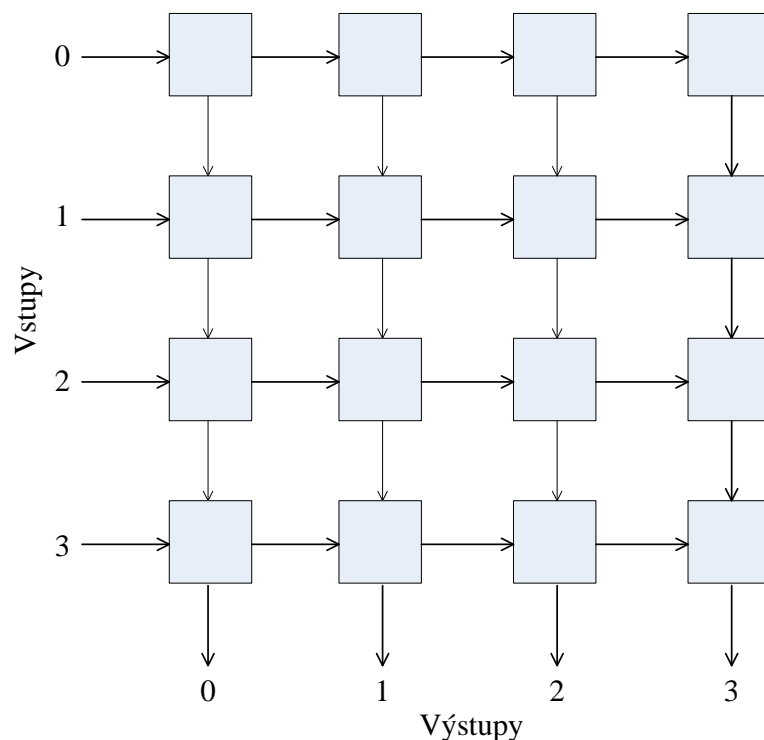
2.2.2 Prepínače využívajúce priestorový multiplex

V tejto skupine prepínačov existuje medzi vstupnými a výstupnými portmi viacero ciest. Tieto cesty je možno používať na súčasný prenos viacerých buniek. Preto je celková priepustnosť prepojovacieho uzlu násobkom šírky pásma cesty a počtu ciest, ktoré súčasne môžu prenášať bunky. Tieto prepínače delíme na 3 hlavné skupiny:

- krížové prepínače (*Crossbar switch*),
- plne prepojené štruktúry,
- spojovacie pole typu Banyan.

Krížový prepínač

Tento typ prepínača je zobrazený na obr. 2.6. Vodorovné linky odpovedajú vstu-

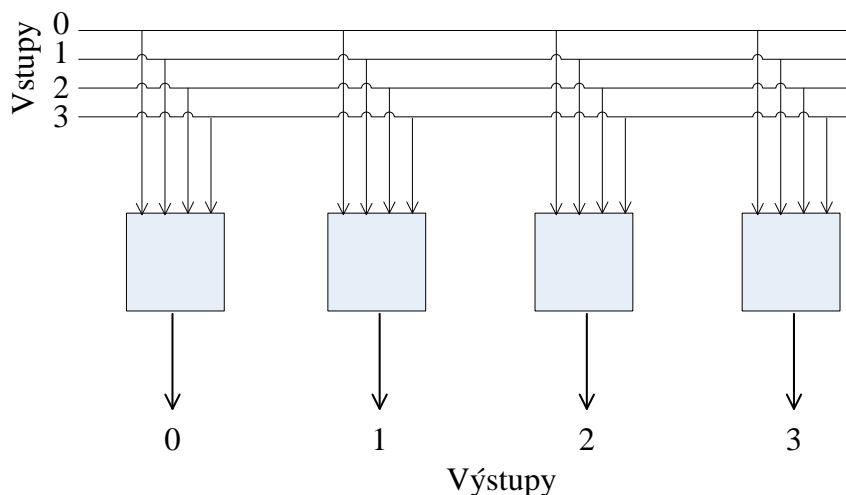


Obr. 2.6: Krížový prepínač

pom a zvislé výstupom. Prepínač obsahuje samostatne riadené spínacie prvky, kde každý môže byť buď v zopnutom alebo rozopnutom stave. Spojenie medzi vstupným a výstupným portom je dosiahnuté uvedením príslušných spínacích prvkov do zopnutého stavu. Tento stav je možno riadiť automaticky príchodom bunky s príslušnou adresou výstupu. Využitím tohoto riešenia sa riadenie spojovacieho pola značne zjednoduší a spojovacie pole sa stane samosmerovacím – riadiace funkcie sú distribuované medzi spínacie prvky. Aby krížový prepínač mohol spracovávať bunky, je treba doplniť vyrovnávajúcu pamäť. [13]

Plne prepojená štruktúra

Plne prepojená štruktúra sa vyznačuje tým, že každý vstup má samostatné spojenie s každým výstupom. Toto spojenie je spravidla dosiahnuté pomocou N zberníc, kde každá zbernica vedie od konkrétneho vstupného portu ku všetkým výstupným portom, viď obr. 2.7. Spojovacie pole vyžaduje N samostatných vyrovnávajúcich pamätí na vstupných portoch.



Obr. 2.7: Plne prepojené spojovacie pole

Plne prepojené spojovacie pole pracuje veľmi podobne ako spojovacie pole so zdieľaným médiom. Bunka z výstupného portu je rozoslaná na každý výstupný port. Architektúra umožňuje, aby bunky od rôznych vstupných portov boli súčasne privedené na rovnaký výstupný port. Preto musí výstupný port filtrovať bunky a ukladať ich do vyrovnávajúcej pamäti. [13]

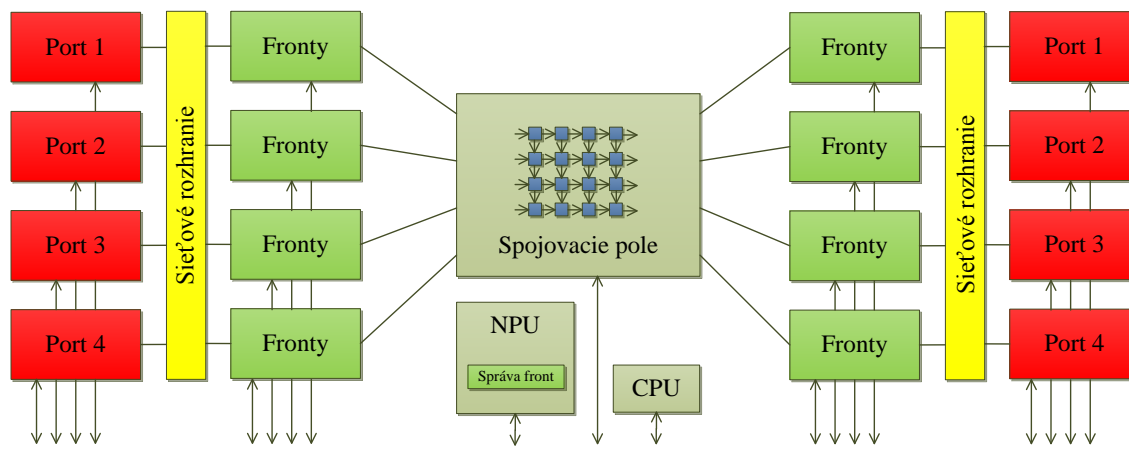
Banyan

Jedná sa o jednocestné samosmerovacie pole zostavené zo spínacích elementov. Počet spínacích elementov v závislosti na počtu vstupov a výstupov narastá logaritmicky. Samosmerovací algoritmus nevyžaduje prídavný riadiaci modul a ako všetky spojovacie polia so zdieľaným médiom umožňuje paralelný prenos viacerých buniek. Nevýhodou tejto štruktúry je vnútorné blokovanie a priepustnosť spojovacieho poľa rapídne klesá s nárastom veľkosti.

Existuje taktiež tzv. rozšírená Banyan sieť, kde viacestné spojovacie pole na rozdiel od predošlých štruktúr obsahuje niekoľko alternatívnych ciest medzi vstupným a výstupným portom. Nevýhodou je zložitejšie prepojovanie buniek. Funkcia prídavných vrstiev funguje tak, že bunka je testovaná v každej prídavnej vrstve, či bola doručená na správny výstupný port. [13]

2.3 Bloková schéma prepínača

V tejto práci bude vytvorený štvorportový prepínač. Bloková schéma takéhoto prepínača je zobrazená na obr. 2.8. Na schéme vidno oddelené vstupné a výstupné porty, bloky front, bloky sieťového rozhrania, spojovacie pole, CPU (*Central Processing Unit*) a NPU (*Network Processing Unit*).



Obr. 2.8: Bloková schéma prepínača [7]

Sieťový procesor NPU je kľúčovým riadiacim prvkom prepínača. Funkcie NPU sú nasledovné:

- riadenie prístupu k médiu,
- spracovanie paketov,
- klasifikácia paketov,
- regulácia a tvarovanie prevádzky.

Univerzálny procesor CPU má tiež riadiacu funkcionálnosť – správa aktívneho prvku, beh operačného systému IOS, manažment konfigurácií atď. Okrem toho plní taktiež niektoré úlohy spojené s primárnymi úlohami aktívneho prvku. Jedná sa o tie, ktoré je nutné plniť tak rýchlo, aby ich oneskorené plnenie znižovalo priepustnosť celého aktívneho prvku. [7]

Spojovacie pole je centrálnou štruktúrou prepínačov, patrí do dátovej oblasti aktívneho prvku. Je jednou z periférií NPU a býva implementované v rámci prepínacieho procesoru. Podrobnejšie informácie boli uvedené v kapitole 2.2.2.

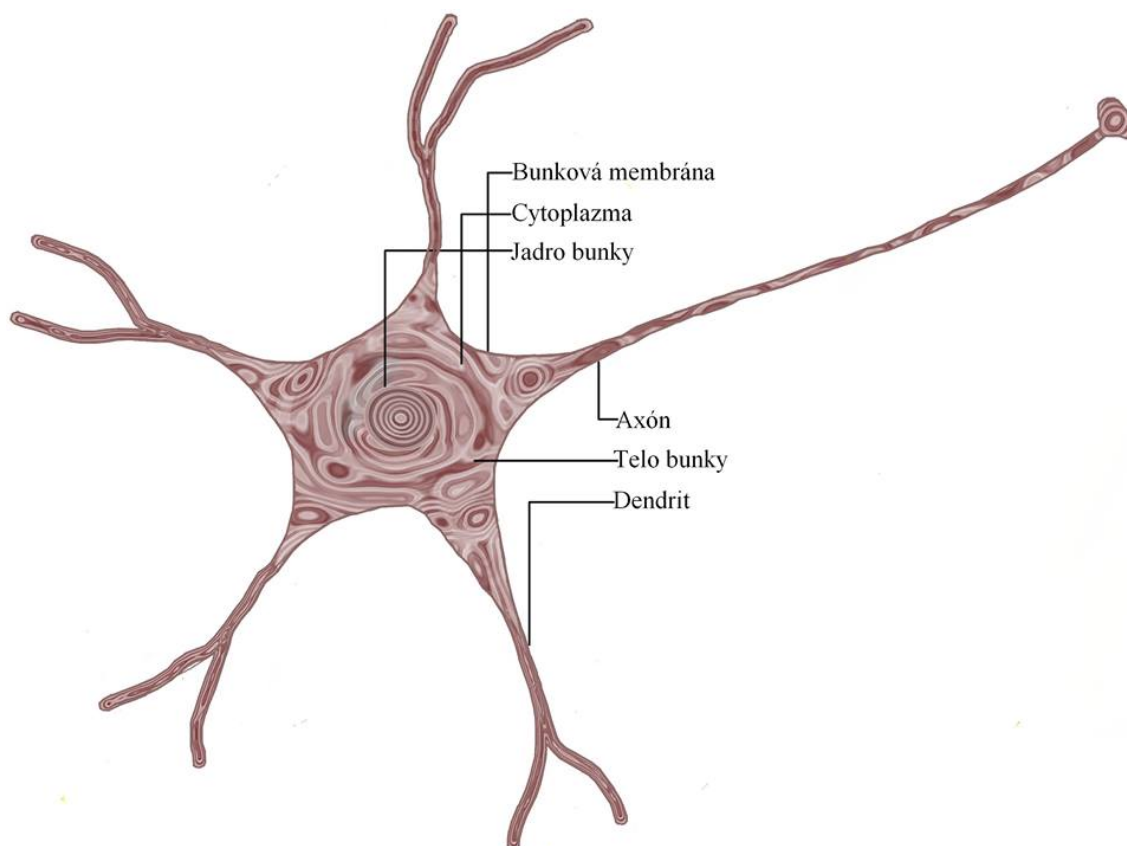
Sieťové rozhranie pracuje s protokolmi druhej vrstvy ISO/OSI modelu. Tvorí v podstate rozhranie medzi prepínačom a koncovými stanicami. Koncové stanice môžu byť počítače, servery, prístupové body, tlačiarne, atď. Porty väčšinou bývajú založené na technológii FastEthernet s rýchlosťou do 100 Mbit/s (využitie pre pripojenie koncových staníc), v novších modeloch GigabitEthernet s rýchlosťou až 1000 Mbit/s (využitie pre pripojenie do smerovača, prípadne iného prepínača). [14]

3 NEURÓNOVÉ SIETE

Teória neurónových sietí vychádza z neurofyziológických poznatkov. Snaží sa vysvetliť správanie sa na princípe spracovania informácií v nervových bunkách. Znalosti získané v oblasti výskumu umelých neurónových sietí majú veľký význam pre neurofyziológiu. V oblasti spracovania informácií nervovým systémom sa preto k niektorým poznatkom možno dopracovať len myšlienkovým experimentom (napr. počítačovým modelom). [26]

3.1 Neurón

Hlavnou funkciou nervového systému je riadiť organizmus. Podkladom tejto funkcie je schopnosť nervového systému spracovávať informácie. Informácie sa v nervovom systéme prenášajú vo forme zmien membránového potenciálu nervových buniek – neurónov. Model biologického neurónu je zobrazený na obr. 3.1. Tvar neurónu môže byť značne variabilný, na rôznych miestach v nervovom systéme sa nachádzajú bunky veľmi rozdielneho tvaru. Neuróny sú ako aj iné bunkové organizmy ohraničené bunko-



Obr. 3.1: Nervová bunka – neurón

vou membránou, ktorá je tvorená fosfolipidovou dvojvrstvou. Jedná sa o membránu polopriepustnú. Základnú stavbu tvorí telo (lat. *soma*) a niekoľko výbežkov. Tie možno rozdeliť na *dendrity* a *axón*. Dendrity predstavujú z informatického hľadiska vstupnú časť (predovšetkým na ne prechádza vzruch z iných buniek) a axón šíri vzruch k iným bunkám. [26]

3.2 Modelovanie neurónu

Jedna z tém v histórii neurónových sietí súvisí s modelovaním samotnej nervovej bunky. V roku 1952 Hodgkin a Huxley na základe meraní generovania a šírenia sa akčného potenciálu v axóne sépie navrhli sústavu diferenciálnych rovníc, ktoré popisujú dynamiku iónových prúdov na aktívnej excitabilnej membráne. Základom týchto rovníc je popis kinetiky membránových iónových kanálov riadených membránovým napätím a výsledkom je veľmi presná výpočtová reprodukcia generovania a šírenia sa akčných potenciálov.

V roku 1943 McCulloch a Pitts predstavili prvý model neuronovej siete. Aplikovali sieť zloženú z tzv. formálnych neurónov na symbolickú logiku, na výroky zložené z elementárnych logických operácií (AND, OR, NOT). V roku 1956 matematik John von Neumann vyriešil problém spoľahlivosti McCullochových-Pittsových sietí v prípade hardwarového poškodenia, a to zavedením redundancie neurónov.

V roku 1958 Frank Rosenblatt ukázal, že McCullochove-Pittsove siete s modifikovateľnými synaptickými váhami sa dajú natrénovať tak, aby vedeli rozpoznávať a klasifikovať objekty. Vymyslel pre ne názov **perceptróny**. Hlavná myšlienka jeho tréningovej procedúry je takáto: najskôr zaznamenajme odpoveď každého formálneho neurónu na daný podnet. Ak je odpoveď správna, nemodifikujeme váhy. Ak je odpoveď daného neurónu nesprávna, potom modifikujeme váhy všetkých aktivovaných vstupných synáps, a to nasledovným spôsobom: ak má byť neurón aktívny a nie je, zväčšíme ich, a naopak, ak má byť na výstupe neurónu 0 a nie je, zmenšíme ich. [26]

Základná funkcia neurónu spočíva v zbere elementárnych informácií z výstupu určitého počtu s ním spolupracujúcich neurónov, vyhodnotenia ich súhrnného stavu a vyhodnotenia zodpovedajúcej reakcie na výstupe. Takto zjednodušene chápaný neurón je definovaný nasledujúcim vzťahom:

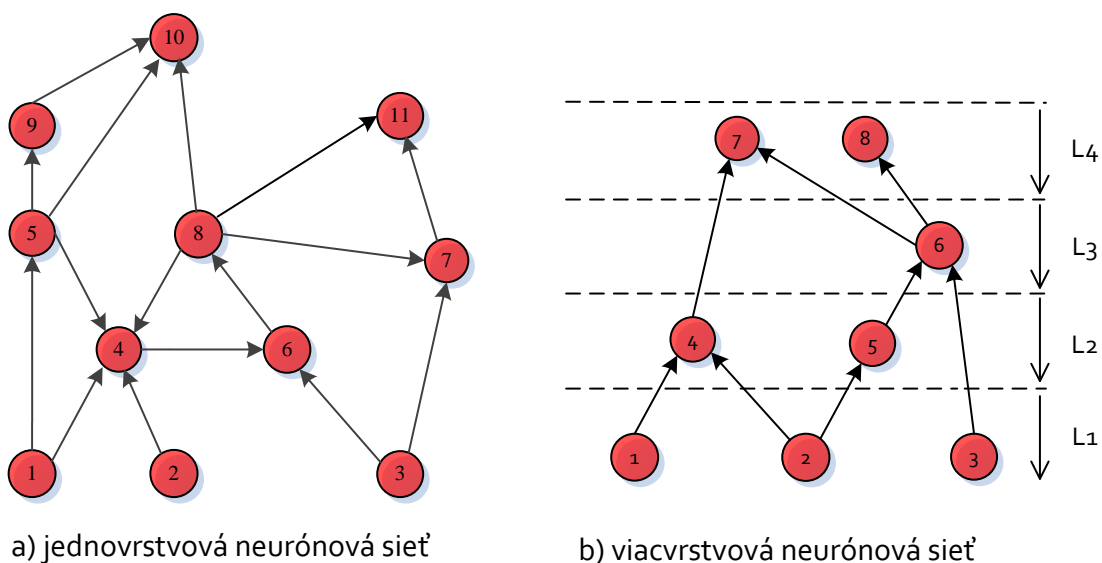
$$a = f\left(\sum_{i=1}^n w_i x_i + b\right), \quad (3.1)$$

kde a je výstupný signál neurónu, f je funkcia prenosu signálu (aktivačná funkcia neurónu), n je počet vstupov do neurónu, w_i sú váhy (*weights*) vstupných brán, x_i je vstupný signál neurónu snímaný zo vstupu alebo zo spolupracujúceho neurónu a b je hodnota prahu (*bias*). Existujú však aj oveľa zložitejšie modely neurónu. [28]

3.3 Viacvrstvé neurónové siete

Jednoduchá neurónová sieť (tzv. lineárny model neurónovej siete) môže byť jednovrstvová alebo viacvrstvová. Paradigma neurónovej siete je formulovaná pomocou grafovo-teoretického prístupu, pričom sa vychádza z analógie ľudského mozgu a koncept neurónovej siete je použitý na konštrukciu modelovej funkcie.

Neurónová sieť je definovaná ako orientovaný súvislý graf. Diagram a) na obr. 3.2 obsahuje orientovaný graf s jedným cyklom a teda nemôže byť použitý pre definíciu neurónovej siete s dopredným šírením. Diagram b) ilustruje možnosť rozkladu vrcholov (neurónov) acyklického orientovaného grafu na vrstvy $L_1 - L_4$, kde L_1 je



Obr. 3.2: Typy neurónových sietí [26]

vstupná vrstva, ktorá obsahuje len vstupné neuróny. Ak počet všetkých vrstiev je t , potom vrstvy $L_2 - L_{t-1}$ sú skryté vrstvy a L_t je vrstva výstupná. V tejto práci je využitá dvojvrstvová neurónová sieť.

Vrstva L_i , kde $1 \leq i \leq t$ je daná nasledujúcim spôsobom:

$$L_i = \{v \in V; d(v) = i + 1\} \quad (3.2)$$

kde vzdialenosť $d(v)$ sa rovná dĺžke maximálnej cesty, ktorá spája daný neurón so vstupným neurónom. Potom musí platiť $d(v) = 0$, pre $v \in V_1$. Neurónová sieť určená acyklickým grafom je obvykle volená tak, že neuróny z dvoch susedných vrstiev sú poprepájané všetkými možnými spojmi. Žiaľ takýto rozklad množiny neurónov na vrstvy je možný len pre neurónové siete reprezentované acyklickými grafmi, pre cyklické grafy vzdialenosť $d(v)$ môže nadobúdať ľubovoľnú kladnú celočíselnú hodnotu. [26]

3.4 Využitie neurónových sietí v prepínacích prvkoch

Prepínač podporujúci kvalitu služieb vykonáva nasledujúce základné funkcie:

- klasifikáciu prevádzky,
- správu front,
- riadenie spojovacieho poľa.

V prípade klasifikácie prevádzky je možné neurónovú sieť využiť k rozpoznávaniu typu prevádzky, protokolov a ich zaradovanie do tokov. Výhodou by bola možnosť zaradenia do triedy pre zariadenia neznámych protokolov na základe ich charakteristík a podobnosti s obdobnými protokolmi.

Pri využití v správe front môže byť plánovač nahradený neurónovou sieťou. Toto riešenie by nebolo veľmi efektívne oproti implementácii konvenčného algoritmu. Plánovač pracuje s presne definovanou množinou tokov – typov prevádzky. Použitím neurónovej siete by sme zvýšili mieru neistoty.

V prípade riadenia spojovacieho poľa je samotné pole, ktoré je riadené deterministickým algoritmom nahradené neurónovou sieťou. Vstupným vektorom neurónovej siete sú dáta prijaté na vstupných portoch prepínača, ktoré sú už patrične usporiadané v predošlom bloku na základe typu prevádzky pomocou deterministických algoritmov. Výstupný vektor neurónovej siete sú dáta radené na príslušné výstupné porty prepínača. Druhou možnosťou využitia neurónovej siete v riadení spojovacieho poľa je použitie krížového prepínača (viď 2.2.2), kde neurónová sieť by riadila jeho činnosť a zaistila by vytvorenie správnej cesty v spojovacom poli.

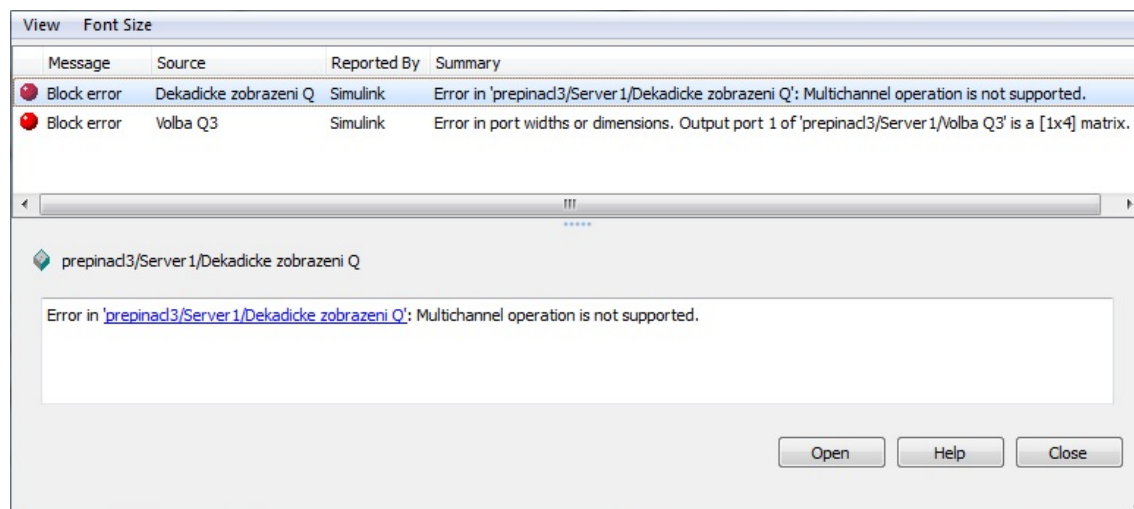
Na základe výskumov vykonaných v rámci 7 sa preukázalo, že využitie neurónovej siete je najefektívnejšie pri využití v riadení spojovacieho poľa. Využitie by bolo efektívne taktiež v klasifikácii prevádzky, bolo by ale potrebné zvoliť vhodnú neurónovú sieť pre každú aplikáciu, čo by spôsobilo značnú zložitosť hardvéru prepínača a značne zvýšené požiadavky na jeho výkonnosť.

V tejto práci je neurónová sieť využitá pre činnosť spojovacieho poľa, viď nasledujúca kapitola. Voľba vhodnej neurónovej siete pre riadenie tohoto poľa bola určená na základe 7. Pre riešenie úloh obsahujúcich problematiku rozpoznávania schém sa doporučuje použitie algoritmu spätného šírenia chyby (*feedforward backpropagation*). V rámci výskumu v 7 boli taktiež testované aj iné typy neurónových sietí, napríklad perceptronová sieť. Perceptrón je neurónová sieť, ktorá využíva jeden pracovný neurón nachádzajúci sa v skrytej vrstve. Počet neurónov vo vstupnej vrstve nie je obmedzený. Pracovný neurón je spojený so všetkými vstupnými neurónmi. Ukázalo sa, že využitie tohoto typu neurónovej siete sa vo využití riadenia spojovacieho poľa javí ako ideálne. [7]

4 NÁVRH PREPÍNAČA

Návrh a simulácia prepínača je prevádzaná v prostredí MATLAB-SIMULINK od firmy *MathWorks*. Simulink je prostredie pre vývoj blokových diagramov, ktoré je možné podľa potreby konfigurovať, a teda vytvárať blokové schémy, ktorých činnosť je možné simulovať a analyzovať. Simulink poskytuje grafický editor, knižnice zahŕňajúce blokové jednotky a modelové a simulačné dynamické systémy. V prostredí je taktiež možné vkladať entity Matlabu, v ktorých následne môžeme navrhovať (programovať) potrebné algoritmy. [15]

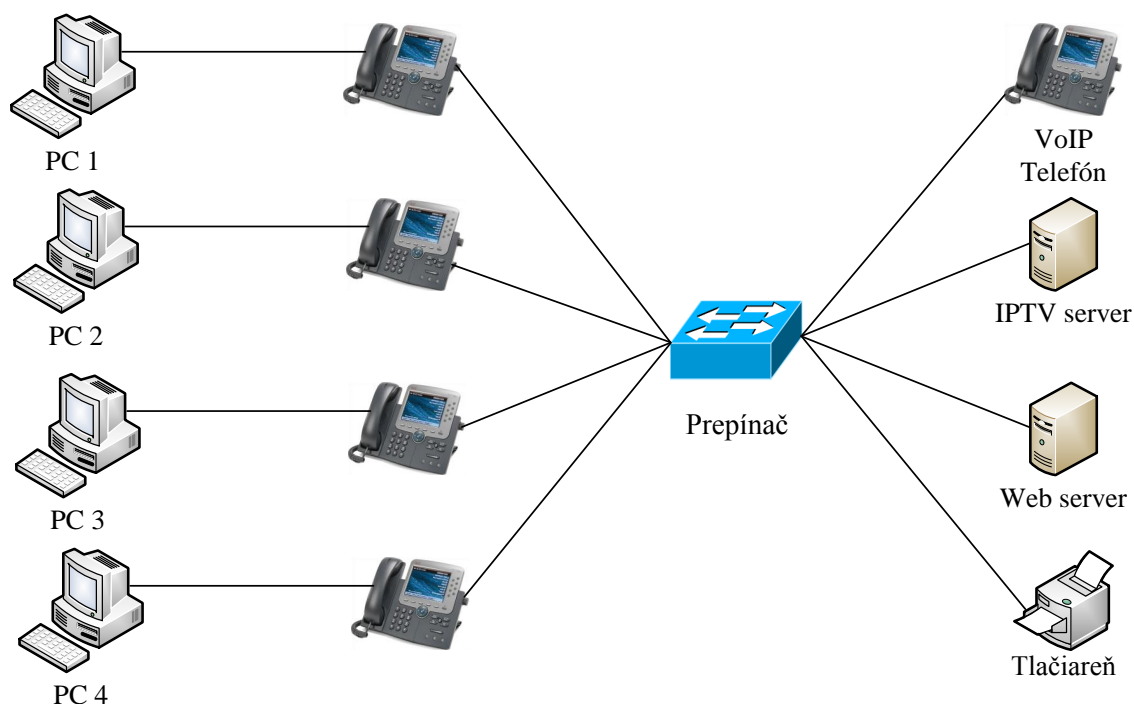
Simulácie v tejto práci sú prevádzané vo verzii R2010a 7.10.0. Použitý je kompilátor z prostredia *Visual Studio 2008 Professional edition*. Nástroj Simulink sa spúšťa príkazom **simulink**. Simulačné súbory sa otvárajú tradične pomocou File – Open (Ctrl+O). Pri spustení simulácie sa môže vyskytnúť nasledujúce chybové hlásenie:



Tejto chybe je možné predísť importovaním kompilátora z prostredia *Visual Studio 2008*, a to nasledovne: v Matlabe zadáme príkaz **mex -setup**, potvrdíme znakom **y**, vyberieme kompilátor pod číslom **1** a vznikne nám súbor typu **.bat**. V tomto momente je kompilátor úspešne importovaný a simulačné súbory je možné spustiť bez problémov (Simulation – Start). [16]

4.1 Topológia siete

Navrhnutá topológia siete je zobrazená na obr. 4.1. V topológii sú do prepínača pripojené 4 počítače cez IP telefóny. Na druhej strane je VoIP telefón, IPTV server, ktorý poskytuje službu sledovania televízie cez internet, webový server pre prístup



Obr. 4.1: Navrhnutá topológia

na webovú stránku a tlačiareň pre tlač dokumentov¹. V tomto scenári sa teda dá predpokladať, že 4 užívatelia môžu využívať 4 druhy prevádzky, na ktoré budeme v simulácii aplikovať QoS. Úlohou prepínača bude teda prepínať rámce na príslušný výstupný port v poradí podľa nastavenej hodnoty QoS.

Ako už bolo spomínané, prepínač prepína rámce na vstupné porty na základe MAC adresy. Ku každému zariadeniu pridáme fixnú MAC adresu, viď tab. 4.1. Ako

Tab. 4.1: MAC adresy zariadení

Zariadenie	MAC adresa	MAC adresa binárne	Dekadické zobrazenie
Telefón 1	00-24-97-34-0E-C5	00000000-00100100-10010111-00110100-00001111-11000101	197
Telefón 2	00-24-97-34-0D-C3	00000000-00100100-10010111-00110100-00001101-11000011	195
Telefón 3	00-24-97-34-0C-FB	00000000-00100100-10010111-00110100-00001100-11111101	251
Telefón 4	00-24-97-34-0A-E2	00000000-00100100-10010111-00110100-00001010-11100010	226
Telefón	00-24-97-34-0F-C4	00000000-00100100-10010111-00110100-00001111-11000100	196
IPTV server	20-16-D8-E3-70-CF	00100000-00010110-11011000-11100011-01110000-11001111	207
Web server	20-16-D8-E3-70-DE	00100000-00010110-11011000-11100011-01110000-11011110	222
Tlačiareň	00-30-C1-31-73-9D	00000000-00110000-11000001-00110001-01110011-10011101	157

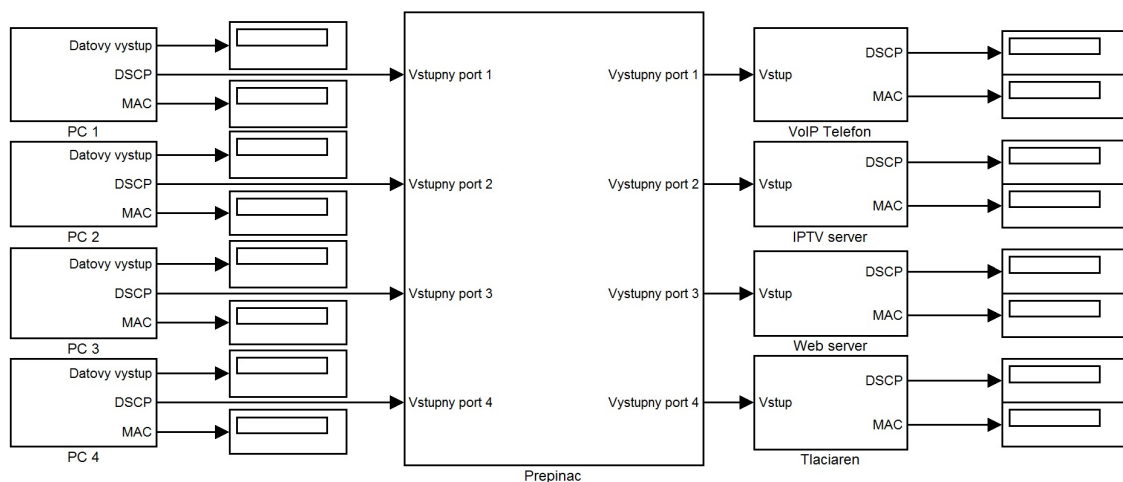
¹V reálnej infraštruktúre by prevádzka nebola aplikovaná cez iba 1 prepínač, ale cez viacero prepínačov či smerovačov, prípadne cez internet.

už bolo zobrazené na obr. 2.1 MAC adresa je 48-bitové číslo, kde prvých 24 bitov nám udáva výrobcu a ďalších 24 bitov udáva adresu konkrétnej stanice. Napríklad v prípade MAC adresy telefónu č. 1 – 00-24-97-34-0E-C5 je prvých 24 bitov „00-24-97“. [18] Podľa 19 prislúcha táto MAC adresa firme *Cisco systems*. V tomto prípade sa teda jedná o Cisco telefón.

Keďže v našej topológii je len obmedzený počet zariadení, nie je potrebné návrh prepínača implementovať pre reálnu 48-bitovú MAC adresu (návrh pre takúto adresu by bol oveľa zložitejší, z dôvodu obmedzenia funkcionality komponentov v prostredí Simulink). Pre zjednodušenie nám v našom prípade postačí posledných 8 bitov, ktorých dekadickú hodnotu budeme v simulácii zobrazovať. Tieto dekadické hodnoty sú taktiež uvedené v tab. 4.1.

4.2 Modelovanie topológie

Topológia navrhnutá v prostredí Simulink je zobrazená na obr. 4.2. Prepínač je

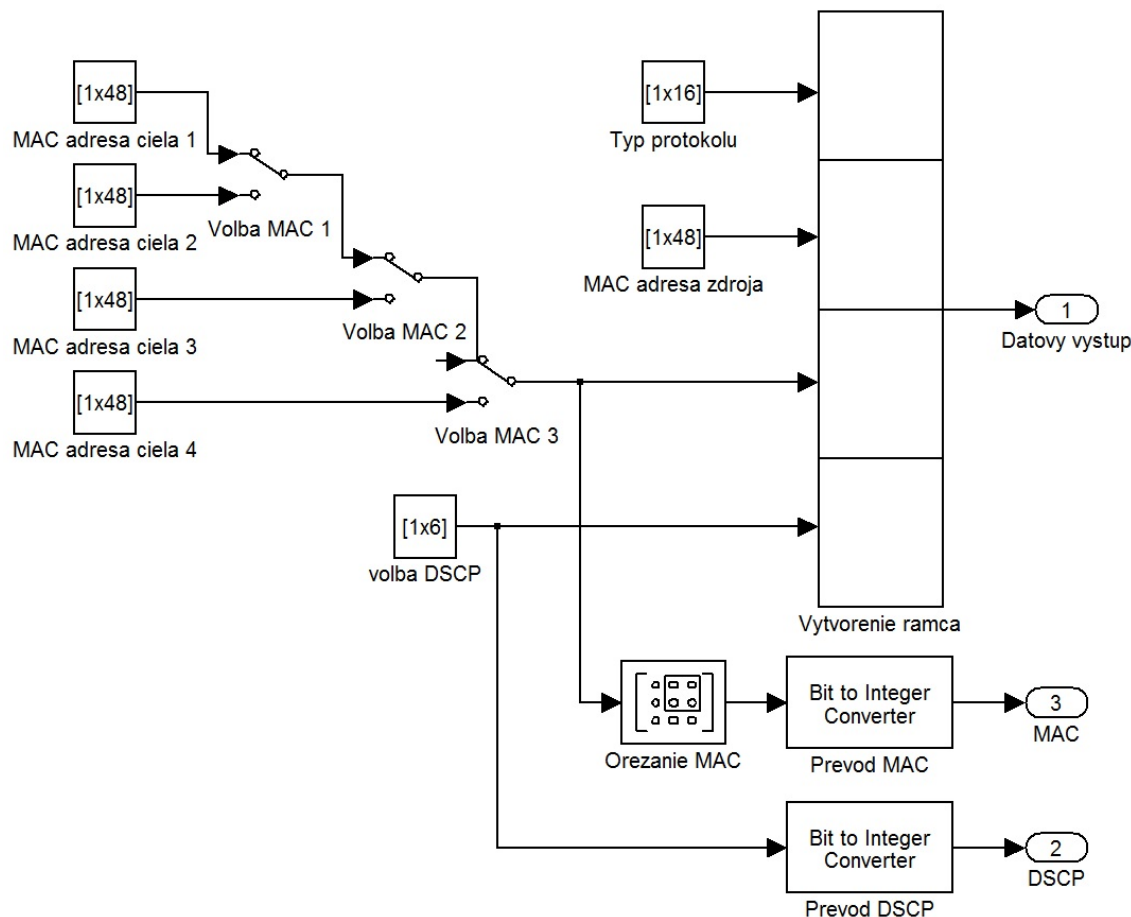


Obr. 4.2: Topológia v prostredí Simulink

tvorený 4 vstupnými a 4 výstupnými portmi. Na vstup prepínača sú pripojené 4 počítače – v praxi bývajú pripojené cez VoIP telefóny, jak je naznačené na obr. 4.1 (v prostredí Simulink sú zobrazené ako samostatná zdrojová jednotka). Na výstupe sú 4 výstupné jednotky predstavujúce spomínané sieťové prvky. Z každej vstupnej jednotky je vyvedený zobrazovač dekadického čísla, pomocou ktorého budeme zobrazovať zvolenú MAC adresu cieľovej stanice a nastavenú hodnotu DSCP. Z výstupných jednotiek sú taktiež vyvedené zobrazovače, jeden však slúži na zobrazenie prijatej hodnoty DSCP a MAC adresy fixne nastavenej na každej jednotke. Pre kontrolu správnej funkčnosti prepínača bude prijaté dáta predstavovať prijatá hodnota DSCP, na základe ktorej bude taktiež príslušná prevádzka adekvátne uprednostnená.

4.2.1 Vstupná jednotka

Úlohou vstupnej dátovej jednotky je v našom prípade predovšetkým generovanie dát (rámcov). Návrh tejto jednotky je zobrazený na obr. 4.3 Rámec bude vytvorený



Obr. 4.3: Vstupná dátová jednotka

spojením všetkých nastavených dát pomocou maticového zlučovača. V tomto komponente nastavujeme počet vstupov, v našom prípade sú to 4 vstupy, do ktorých privádzame nasledovné dáta:

- typ protokolu,
- zdrojová MAC adresa,
- cieľová MAC adresa,
- hodnota DSCP.

Jednotlivé dáta sú implementované pomocou komponenty konštantnej hodnoty, kde nadefinujeme príslušnú dátovú postupnosť bitov, ktoré vytvoria jeden vektor. Maticový zlučovač nám teda spojí jednotlivé vektory do jedného veľkého vektora, ktorý je vlastne postupnosť bitov a predstavuje zdrojový rámec.

Napríklad v prípade definovania MAC adresy tlačiarne zadáme do komponenty konštantnej hodnoty nasledujúcu postupnosť bitov: [0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 1 0 0 0 1 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 1]. Cieľové MAC adresy sú fixne nastavené v 4 komponentoch (podľa tab. 4.1) a pomocou spínačov môžeme voliť požadovanú adresu cieľového zariadenia. Zdrojová MAC adresa je taktiež fixne definovaná podľa tab. 4.1.

Ďalej do vytvoreného rámca definujeme pole typ protokolu. Toto pole nebude v tejto simulácii využité, definujeme ho len z ilustračných dôvodov. Podľa obr. 2.1 je toto pole 16-bitové a v našom prípade definujeme protokol ipv4 – podľa 20 je tomuto typu protokolu pridelená hexadecimálna hodnota 0x0800.

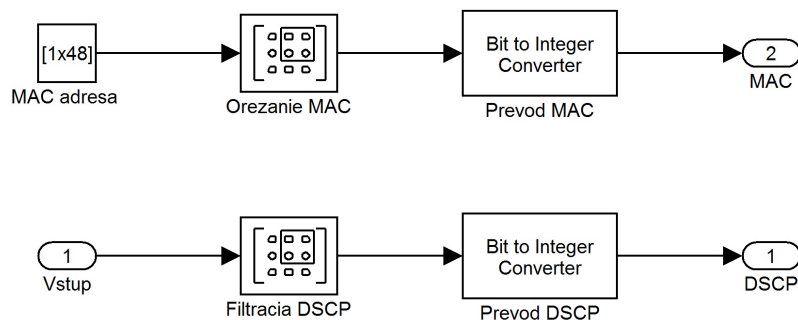
Posledným polom navrhnutého rámca je dôležité pole – DSCP. Pole taktiež odpovedá reálnej veľkosti, a teda je tvorené 6 bitmi, ktoré zvolíme podľa tab. 2.1. Toto pole narozdiel od cieľovej MAC adresy môžeme voliť ľubovoľne, keďže počet možných hodnôt DSCP je podstatne väčší (cieľové MAC adresy sú fixne definované a pomocou spínačov volíme cieľovú stanicu). Túto možnosť voľby využijeme v návrhu laboratórnej úlohy, kde úlohou bude vhodne zvoliť hodnotu DSCP podľa príslušného typu prevádzky, viď nasledujúca kapitola.

V predchádzajúcej kapitole bola opísaná štruktúra reálneho rámca, resp. paketu. Takýto rámec by mal oveľa väčšiu dĺžku ako rámec navrhnutý v tomto modeli a návrh modelu by bol oveľa zložitejší. Model prepínača je ale navrhovaný pre edukatívne účely, a teda nám postačí zjednodušená varianta vstupných dát (nepotrebné časti rámca sú v modeli vynechané).

4.2.2 Výstupná jednotka

Do výstupnej dátovej jednotky (obr. 4.4) je privedený výstupný port prepínača. Pre overenie správnej činnosti zobrazíme prijatú hodnotu DSCP. Tak, ako boli pri tvorení rámca jednotlivé polia spojené pomocou maticového zlúčovača, tak potrebujeme požadované hodnoty z vytvoreného rámca vyfiltrovať. Použijeme k tomu komponent *Submatrix*, v ktorom definujeme index začiatku a konca dát, ktoré chceme z vektora vyfiltrovať. Napríklad v prípade filtrácie hodnoty DSCP bude index začiatku vektora (*Starting column index*) 113 a index konca (*Ending column index*) 118, viď rozloženie bitovej postupnosti na obr. 4.3.

Ďalej je vo výstupnej jednotke fixne definovaná MAC adresa, ktorá je vždy fixne pridelená sieťovému rozhraniu. Budeme ju zobrazovať vo výstupnom zobrazovači dekadického čísla pre porovnanie s požadovanou cieľovou adresou nastavovanou v zdrojovej jednotke. Z MAC adresy je taktiež odrezaných posledných 8 bitov, aby bolo možné určiť totožnosť s MAC adresou, s ktorou náš prepínač v skutočnosti pracuje.

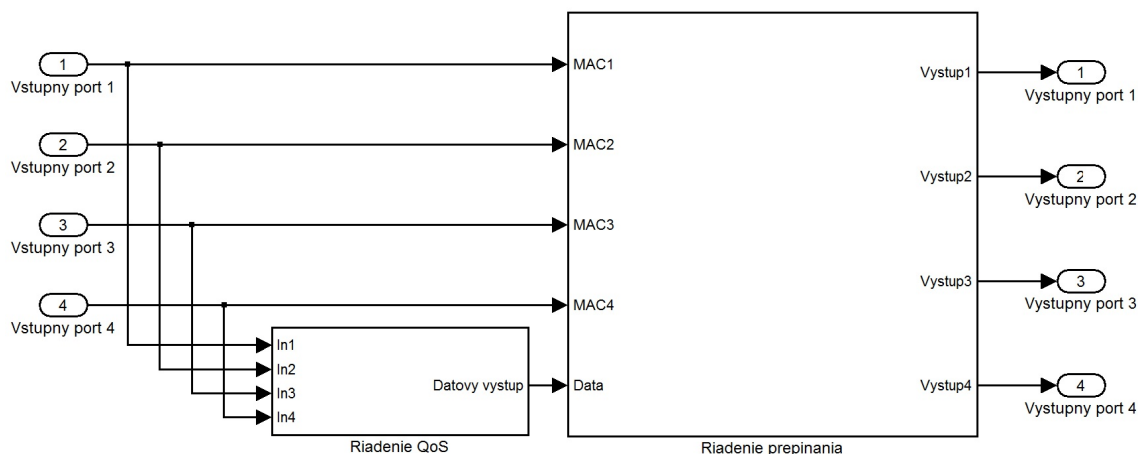


Obr. 4.4: Výstupná dátová jednotka

4.2.3 Telo prepínača

Model prepínača je rozdelený na niekoľko subsystémov, kde jednotlivé subsystémy plnia určitú funkcionálnu. V prípade nepoužitia týchto subsystémov by bola schéma prepínača príliš rozsiahla a neprehľadná. Dva základné subsystémy prepínača sú:

- riadenie QoS,
- riadenie prepínania.



Obr. 4.5: Telo prepínača

Riadenie QoS má za úlohu určiť, v akom poradí budú dáta doručené do subsystému riadenie prepínania, kde budú dáta následne priradené na príslušný výstupný port.

Riadenie QoS

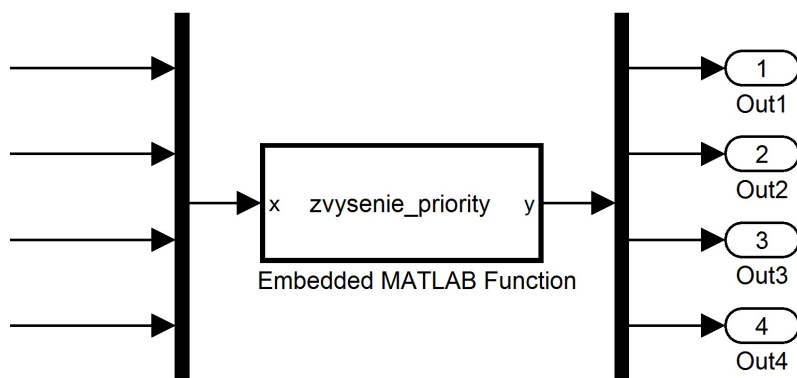
Štruktúra tohoto subsystému je zobrazená v prílohe B.1. V subsystéme sa zo vstupných dát filtruje cieľová MAC adresa, ktorá je následne prevedená do dekadického tvaru. Filtruje sa iba potrebná časť MAC adresy, čiže posledných 8 bitov – z vytvoreného rámca sa jedná o bity 105-112.

Kedže subsystém riadenie QoS určuje, v akom poradí budú rámce prepínané, je potrebné rámce klasifikovať. Implementácia tejto klasifikácie je zobrazená v prílohe B.2. Signály sú ďalej privedené do subsystému radenie priorít.

Okrem klasifikácie rámcov sú signály ešte privedené do subsystému priorit, ktorý rieši situáciu, v ktorej by 2 rámce s rovnakou prioritou mali byť prepnuté na rovnaký výstupný port. Jeden z týchto rámcov je treba uprednostniť, a teda zvýšiť mu prioritu. V tomto subsystéme je jednak implementované preskladanie do poradia FIFO, a ďalej sú signály pomocou multiplexéru privedené do komponentu *Embedded MATLAB Function* (viď obr. 4.6), kde v prípade detekcie 2 rámcov s rovnakou prioritou je jednému z nich zvýšená priorita o 1. Implementovaný je nasledujúci algoritmus:

```
function y = zvyssenie_priority(x)
for i=1:3
    for j=1:(4-i)
        if x(i)==x(i+j)
            x(i+j)=x(i+j)+1;
        end;
    end;
end;
y=x;
```

Algoritmus je navrhnutý pre práve 4 možné fronty, resp. 4 vstupné porty. V prípade návrhu prepínača pre viac portov by bolo potrebné algoritmus modifikovať, ideálne globalizovať. Algoritmus porovnáva každý vstupný signál s každým ďalším vstupným signálom – prvý porovná so zvyšnými tromi, druhý so zvyšnými dvoma a tretí so zvyšným štvrtým. V prípade rovnosti signálov je pôvodnému signálu zvýšená priorita o 1. Signály sú následne opäť demultiplexované a privedené na 4 výstupy, ktoré sú taktiež privedené do subsystému radenie priorít.



Obr. 4.6: Zvýšenie priority

V subsystéme radenie priorit sú jednotlivé priority zoradené vzostupne. Zoradenie je realizované pomocou *Embedded MATLAB Function*, kde je naprogramovaný algoritmus zoradenia. Implementovaný je algoritmus *Bubblesort*. [21] Ďalej sú príslušné páry signálov navzájom porovnávané, aby subsystém postupne obslúžil aj rámce s najnižšou prioritou. [22]

Pre implementáciu poradia prepínania je potrebné vložiť časovač (*trigger*), ktorý riadi pomocný spínač. Tento prepínač vysiela do spínača poradia postupnosť riadiacich pulzov podľa príslušnej kvality služby. Spínače sú zoradené v kaskáde a dáta sa prenášajú sériovo. Subsystém radenie priorit generuje signály pre spínač poradia, ktorý ma hodnotu 1-4 a bude ovládať v zoradenom poradí hlavný spínač. Tým hlavný spínač zopne v určitom poradí hlavný dátový tok do subsystému riadenie prepínania. [23]

Riadenie prepínania

Reálny prepínač si ukladá spoznané MAC adresy do tzv. ARP tabuľky. Akonáhle rozpozná prepínač novú MAC adresu, zapíše ju do ARP tabuľky. Protokol ARP (*Address Resolution Protocol*) slúži na mapovanie (preklad) IP adresy tretej vrstvy ISO/OSI na MAC adresu – adresu druhej vrstvy. [24]

V našom modeli prepínača je ARP tabuľka implementovaná v subsystéme riadenie prepínania. V tomto prípade nám postačí, ak budú v ARP tabuľke uložené záznamy o zariadení, ktoré sú pripojené na výstupných portoch prepínača. Následne sú tieto záznamy z tabuľky načítavané a porovnávané s prijatými MAC adresami nastavenými na zdrojovej stanici. Taktiež pracujeme iba s poslednými 8 bitmi, ktoré ďalej konvertujeme do dekadického podoby.

Štruktúra tohoto subsystému je zobrazená na obr. B.3. Funkcia pre porovnávanie adries je nasledovná:

```
function y = prepinanie(MAC_prijata, MAC_arp)
if MAC_prijata==MAC_arp;
    y=2;
else
    y=1;
end;
```

Keďže v modeli môžu byť len 4 MAC adresy, prepojí sa odpovedajúca adresa tým spôsobom, že portový spínač dostane povel na prepnutie dát, inak prepojí nulu. [23] Vo výstupe sú nakoniec signály dát prepnutých podľa príslušnej MAC adresy nastavenej na zdrojovej stanici a v poradí podľa príslušnej hodnoty DSCP. Výsledky simulácie sú uvedené v rámci laboratórnej úlohy.

4.3 Prepínač riadený neurónovou sieťou

Pre využitie neurónovej siete v modelovaní prepínača je vytvorený druhý model. Topológia a vnútorná štruktúra je totožná s predošlým návrhom, viď obr. 4.1 a 4.2. Telo prepínača je taktiež rozdelené na 2 základne časti – riadenie QoS a riadenie prepínania. Rozdiel je v bloku riadenie prepínania, kde je použitá neurónová sieť.

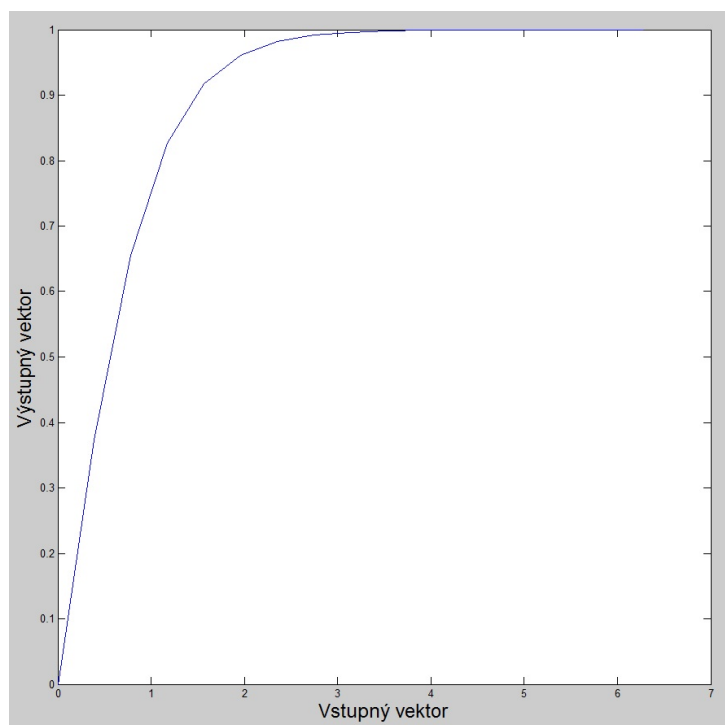
4.3.1 Vytvorenie neurónovej siete

Tvorba neurónovej siete vychádza zo vzťahu 3.1. Základom je aktivačná funkcia, ktorá je matematickou transformáciou súhrnu váh vstupov, ktoré tvoria výstup neurónu. Táto funkcia má 2 časti:

- kombinačná funkcia (*combination function*),
- prenosová funkcia (*transfer function*).

Úlohou kombinačnej funkcie je zlúčiť všetky vstupy neurónu do jednej jednotky. Výstup tejto funkcie je vstupom prenosovej funkcie, ktorá nad vstupmi prevádza obvykle nelineárnu transformáciu a generuje výstup neurónu. [7]

Štandardná aktivačná funkcia je sigmoidná. Príkladom takejto funkcie je hyperbolický tangens. Túto funkciu nadefinujeme do vstupného a výstupného vektora, kde výstupný vektor (*TargetVector*) predstavuje hyperbolický tangens vstupného vektora (*InputVector*), viď obr. 4.7.



Obr. 4.7: Funkcia hyperbolický tangens

Ďalej vytvoríme objekt neurónu pomocou nasledujúcich príkazov:

```
PR = minmax(InputVector);
Layers = [2 1];
TransFcns = {'tansig' 'purelin'};
BTF = 'traingdx';
BLF = 'learnngdm';
PF = 'mse';
net = newff(PR, Layers, TransFcns, BTF, BLF, PF);

pause;
```

Jednotlivé položky predstavujú nasledovné:

- *PR* – predstavuje rozsah vstupných hodnôt (minimálnu a maximálnu hodnotu elementov vstupných vektorov),
- *Layers* – udáva počty neurónov jednotlivých vrstvách,
- *TransFcns* – udáva prenosovú (aktivačnú) funkciu neurónov v jednotlivých vrstiev,
- *BTF* – udáva názov tréningovej funkcie siete (*beachtraining*),
- *BLF* – udáva názov funkcie siete (*steptraining*),
- *PF* – udáva názov funkcie pre výpočet chyby pri tréningu,
- *net* – premenná, v ktorej je uložený vytvorený objekt neurónovej siete.

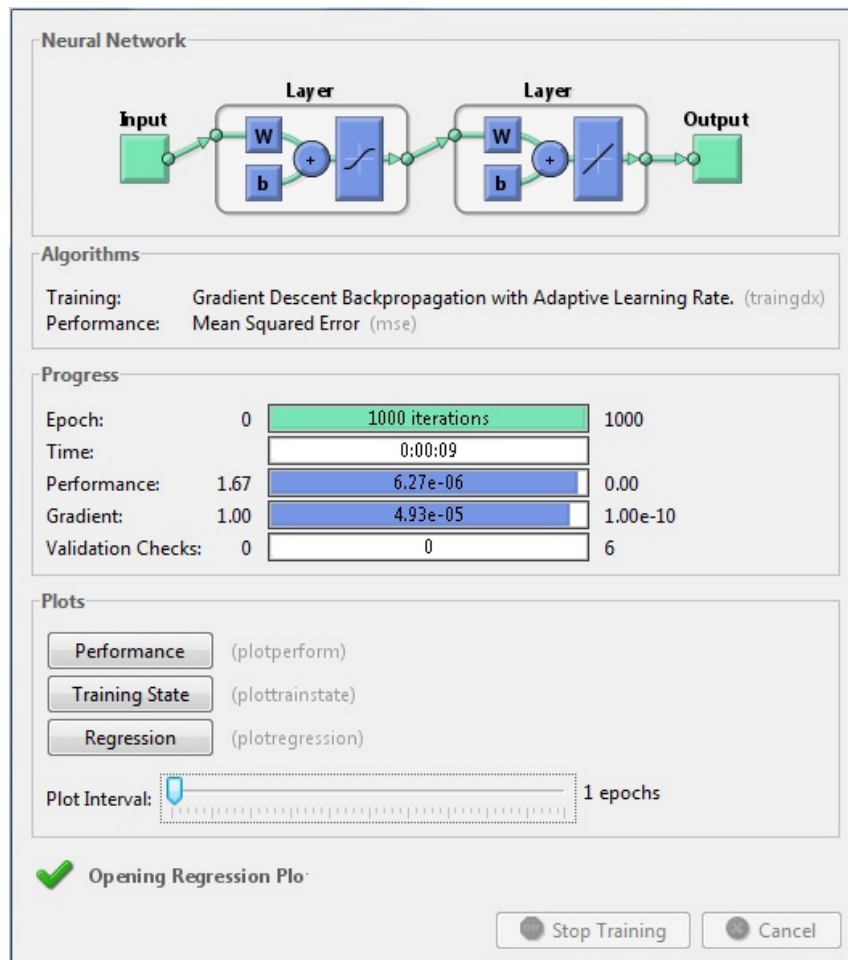
V objekte *net* sú uložené všetky paradigmy neurónovej siete, ako napríklad informácie o topológii, prenosových funkciách, chybových funkciách, hodnoty váh, prahov. Pred procesom tréningu je potrebné sieť inicializovať *init()*. Behom inicializácie dochádza k nastaveniu váh a prahov neurónovej siete. Po inicializácii môžeme spustiť proces tréningu pomocou príkazu *train()*:

```
net = init(net);
[net, tr] = train(net, InputVector, TargetVector);
```

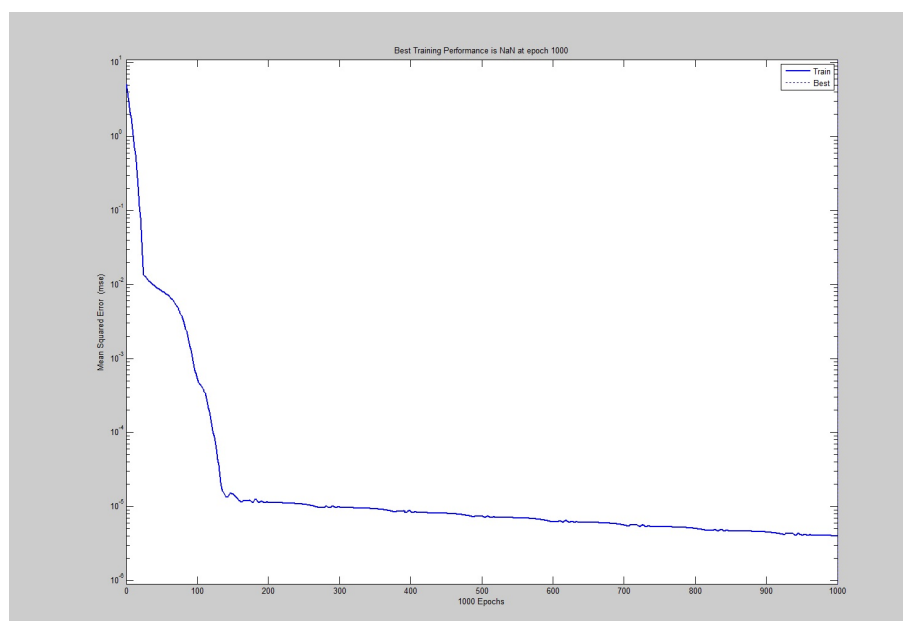
Jednotlivé parametre predstavujú nasledovné:

- *net* – názov objektu siete, ktorá sa má natréňovať,
- *InputVector* – vstupný vektor dát vzorovej množiny,
- *tr* (*training record*) – dáta reprezentujúce priebeh tréningového procesu.

Výsledok tréningu neurónovej siete môžeme vidieť na obr. 4.8. Behom tréningu sa vykresľuje tréningová chybová krivka (obr. 4.9), ktorá ukazuje, ako postupne klesá chybová funkcia tréningu. Vytvorenú neurónovú sieť teraz môžeme importovať do Simulinku pomocou príkazu *gensim(net, st)*, viď ďalšia kapitola. 29



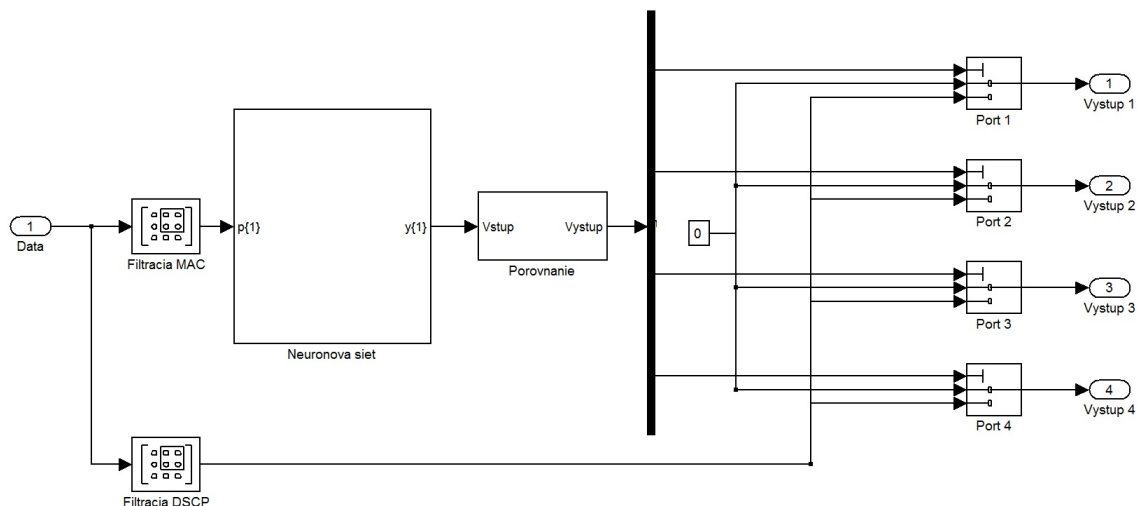
Obr. 4.8: Trénovanie neurónovej siete



Obr. 4.9: Trénovacia chybová krivka

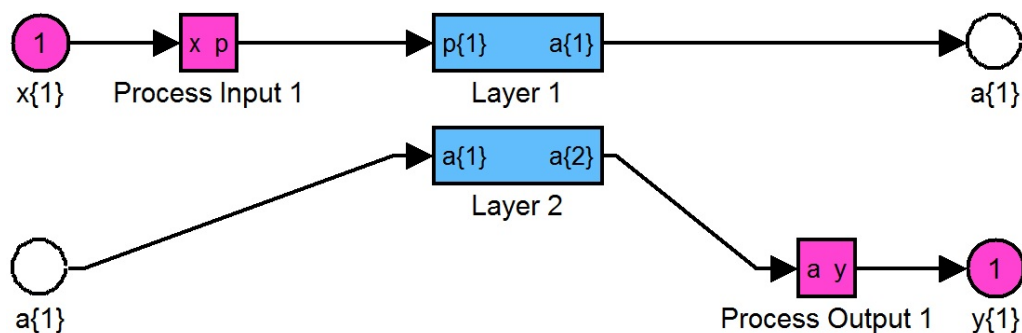
4.3.2 Aplikovanie neurónovej siete

Ako už bolo spomínané, neurónová sieť je použitá v bloku riadenie prepínania. Tento blok je zobrazený na obr. 4.10. Zo vstupného signálu je vyfiltrovaná jak MAC ad-



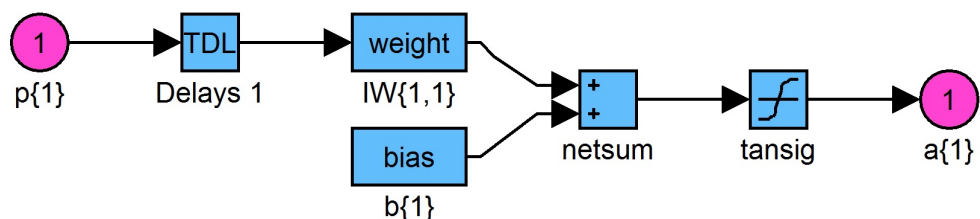
Obr. 4.10: Riadenie prepínania pomocou neurónovej siete

resa, tak hodnota DSCP, ktorá bude nakoniec zobrazená na výstupných jednotkách. Vyfiltrovaná MAC adresa je vstupnou jednotkou neurónovej siete. Štruktúra tejto siete je zobrazená na obr. 4.11. Neurónová sieť bola importovaná do Simulinku pomocou funkcie *gensim(net,st)*, kde parameter *net* predstavuje vytvorenú neurónovú sieť a *st* predstavuje časový interval (defaultná hodnota je 1). [27]



Obr. 4.11: Štruktúra neurónovej siete

Neurónová sieť je tvorená vstupnou a výstupnou vrstvou, tak ako je popísané v 3.3. Každá z vrstiev je tvorená blokmi charakteristickými pre neurónovú sieť podľa vzťahu 3.1 – oneskorovací člen, váhy (*weights*), hodnotu prahu (*bias*), súčet a funkciu prenosu signálu (aktivačnú funkciu neurónu), viď obr. 4.12.



Obr. 4.12: Štruktúra vrstvy neurónovej siete

Sieť má nastavené váhové a prahové hodnoty tak, aby konkrétnej adrese priradila príslušný výstup (viď obr. B.4). Neurónová sieť teda zvolí, na ktorý port bude daný rámec prepnutý.

Pre prehľadnosť je možné výstup neurónovej siete zobrazit napríklad pridaním komponentu *Display*. Pre efektívnejšie zobrazenie a prípadné ďalšie spracovanie je vhodnejšie dáta exportovať do súboru typu *.mat*, a to pomocou komponentu *To File*. V tomto komponente definujeme názov súboru a premennej, do ktorej budú dáta uložené. Vzniknutý *.mat* súbor môžeme zobazit klasickým otvorením alebo je možné jeho obsah načítať do *workspace* a následne zobrazit zadáním názvu vytvorenej premennej do príkazového riadka matlabu. Obsah súboru je zobrazený v tab. 4.2. Výsledné hodnoty sú buď veľmi blízke 0 alebo 1. Na základe tejto hodnoty ďalej

Tab. 4.2: Štruktúra vrstvy neurónovej siete

	1	2	3	4	5	6
1	0	1	2	3	4	5
2	-8.57E-11	-8.57E-11	0.99999999	1.96E-11	-1.18E-10	-8.57E-11
3	-7.21E-11	-7.21E-11	-5.20E-11	-3.61E-11	0.99999999	-7.21E-11
4	0.99999999	0.99999999	-4.73E-12	1.33E-11	5.39E-11	0.99999999
5	-2.17E-11	-2.17E-11	-4.59E-11	0.99999999	9.67E-11	-2.17E-11

prepínač určí, či má byť rámec na daný výstupný port poslaný alebo nie. Preto je potrebné hodnotu zaokrúhliť na presnú hodnotu 0 resp. 1. Toto zaokrúhlenie je implementované v bloku *Porovnanie*, kde prebieha prevod na absolútnu hodnotu a následne porovnanie s referenčnou hodnotou a vrátenie hodnoty 1 v prípade, že je hodnota väčšia ako nastavená referenčná hodnota.

Signál je následne privedený do 4 spínačov, ktoré pošlú na výstup vyfiltrovanú hodnotu DSCP, pokiaľ sa na ich riadiacom porte objaví hodnota 1. [22] Výstupy týchto spínačov sú ďalej už privedené do cieľových staníc, kde je na displeji zobrazená príslušná hodnota DSCP, na základe čoho je jednoducho a prehľadne viditeľná správna funkčnosť prepínača. Výsledky simulácie sú opäť zahrnuté v rámci laboratórnej úlohy, viď nasledujúca kapitola.

5 LABORATÓRNA ÚLOHA

5.1 Zadanie

1. Preštudujte teóriu prepínania, neurónových sietí, QoS a architektúru prepínača.
2. Navrhните hodnoty kvality služby pre zadané typy služieb.
3. Simulujte rôzne možnosti prevádzky pre rôzny počet užívateľov.
4. V prostredí Matlab vytvorte a analyzujte neurónovú sieť.
5. Simulujte model prepínača riadeného neurónovou sieťou.
6. Porovnajte oba modely prepínačov.
7. Výsledky analyzujte a vyhodnoťte.

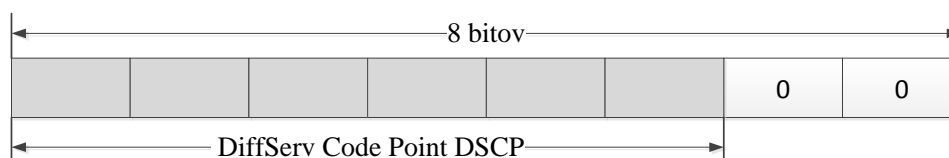
5.2 Teoretický úvod

5.2.1 Prepínač a QoS

Prepínače sú jedným zo základných prvkov moderných komunikačných sietí. Pracujú na druhej (spojovej) vrstve referenčného modelu ISO/OSI a zaisťujú prenos dátových jednotiek – rámcov.

Dnešné moderné prepínače sú schopné pracovať aj na tretej vrstve, a teda sú schopné spracovávať pakety. Hlavička paketu zahŕňa pole ToS (*Type of Service*), ktoré nám zaisťuje značenie paketov, a teda kvalitu služieb. Pole ToS je 8-bitové, z toho 3 bity nám určujú prioritu podľa špecifikácie typu služby. Ďalšie 3 bity určujú požiadavky na prenos a zvyšné 2 bity sú nepoužívané a musia mať nastavenú hodnotu 0.

U mechanizmu diferencovaných služieb je spracovanie prevádzky riadené relatívnymi prioritami priradených jednotlivým triedam prevádzky. Špecifikácia uvádza nové pole DS (*Differentiated Services*), viď obr. 5.1. Prvých 6 bitov tohoto poľa je



Obr. 5.1: Štruktúra poľa DS

označovaných ako DSCP (*DiffServ Code Point*), zvyšné 2 bity sú taktiež nevyužívané. Na základe hodnoty DSCP je paketom zaistený požadovaný spôsob zachádzania – PHB (*Per-Hop Behavior*).

Ľavý krajný bit tohoto poľa sa označuje 0. bit, ktorý je najvýznamnejší a naopak pravý bit je najmenej významný a označuje sa 5. bit. Týmito bitmi je teda možno vyjadriť 64 rôznych hodnôt. Zoznam bežne používaných hodnôt je uvedený v tab. 5.1. AF (*Assured Forwarding*) zaisťuje zaručené preposielanie dát, EF (*Expedited*

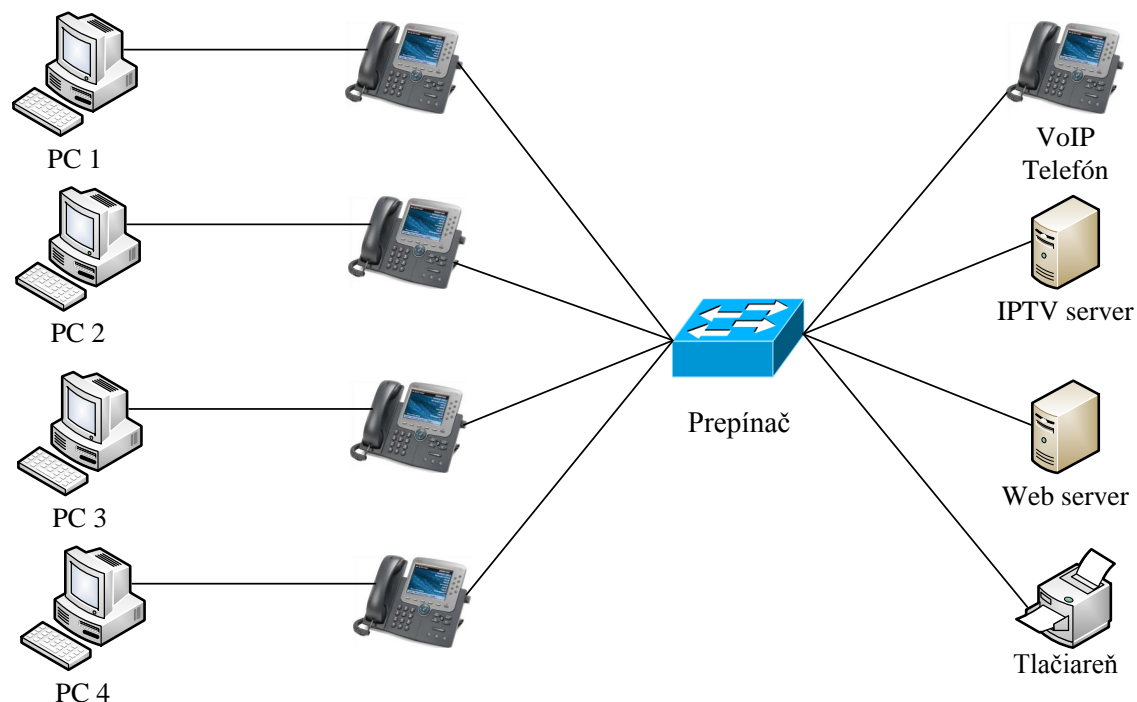
Tab. 5.1: Bežne používané DSCP hodnoty

DSCP hodnota	Decimálna hodnota	Význam	Ekvivalentná IP priorita
000 000	0	BE	000
101 110	46	EF	101
001 010	10	AF11	001
001 100	12	AF12	001
001 110	14	AF13	001
010 010	18	AF21	010
010 100	20	AF22	010
010 110	22	AF23	010
011 010	26	AF31	011
011 100	28	AF32	011
011 110	30	AF33	011
100 010	34	AF41	100
100 100	36	AF42	100
100 110	38	AF43	100
001 000	8	CS1	1
010 000	16	CS2	2
011 000	24	CS3	3
100 000	32	CS4	4
101 000	40	CS5	5
110 000	48	CS6	6
111 000	56	CS7	7

Forwarding) znamená urýchlené preposielanie, zaisťuje nízku stratovosť, nízke oneskorenie a nízke kolísanie oneskorenia. CS (*Class Selector*) je označenie DSCP hodnoty, ktoré odpovedá určitej IP priorite a BE (*Best-effort*) je prednastavená (*default*) priorita, ktorá nám ale nezaručuje, že všetky dáta budú doručené.

5.2.2 Topológia siete

Topológia siete je zobrazená na obr. 5.2. V topológii sú do prepínača pripojené 4 počítače cez IP telefóny. Na druhej strane je VoIP telefón, IPTV server, ktorý poskytuje službu sledovania televízie cez internet, webový server pre prístup na webovú stránku a tlačiareň pre tlač dokumentov.



Obr. 5.2: Topológia siete

Každé zariadenie má pridelenú fixnú MAC adresu, viď tab. 5.2. MAC adresa je 48-bitové číslo, kde prvých 24 bitov nám udáva výrobcu a ďalších 24 bitov udáva adresu konkrétnej stanice. Napríklad v prípade MAC adresy telefónu č. 1 – 00-24-97-34-0E-C5, je prvých 24 bitov „00-24-97“. Podľa 19 prislúcha táto MAC adresa firme *Cisco systems*. V tomto prípade sa teda jedná o Cisco telefón.

Keďže v našej topológii je len obmedzený počet zariadení, nie je potrebné, aby model prepínača pracoval s celou 48-bitovou MAC adresou. Prepínač je teda navrhnutý tak, že pracuje len s poslednými 8 bitmi MAC adresy, ktorých dekadickú hodnotu budeme v simulácii zobrazovať. Tieto dekadické hodnoty sú taktiež uvedené v tab. 5.2.

Tab. 5.2: MAC adresy zariadení

Zariadenie	MAC adresa	MAC adresa binárne	Dekadické zobrazenie
Telefón 1	00-24-97-34-0E-C5	00000000-00100100-10010111-00110100-00001111-11000101	197
Telefón 2	00-24-97-34-0D-C3	00000000-00100100-10010111-00110100-00001101-11000011	195
Telefón 3	00-24-97-34-0C-FB	00000000-00100100-10010111-00110100-00001100-11111011	251
Telefón 4	00-24-97-34-0A-E2	00000000-00100100-10010111-00110100-00001010-11100010	226
Telefón	00-24-97-34-0F-C4	00000000-00100100-10010111-00110100-00001111-11000100	196
IPTV server	20-16-D8-E3-70-CF	00100000-00010110-11011000-11100011-01110000-11001111	207
Web server	20-16-D8-E3-70-DE	00100000-00010110-11011000-11100011-01110000-11011110	222
Tlačiareň	00-30-C1-31-73-9D	00000000-00110000-11000001-00110001-01110011-10011101	157

5.2.3 Model prepínača

Model prepínača je umiestnený na pracovnej ploche – súbor *prepinac.mdl*. Prepínač je tvorený 4 vstupnými a 4 výstupnými portmi. Na vstup prepínača sú pripojené 4 počítače. Na výstupe sú 4 výstupné jednotky predstavujúce spomínané sieťové prvky. Z každej vstupnej jednotky je vyvedený zobrazovač dekadického čísla, pomocou ktorého budeme zobrazovať zvolenú MAC adresu cieľovej stanice a nastavenú hodnotu DSCP. Z výstupných jednotiek sú taktiež vyvedené zobrazovače, jeden však slúži na zobrazenie prijatej hodnoty DSCP a MAC adresy fixne nastavenej na každej jednotke. Pre kontrolu správnej funkčnosti prepínača bude prijaté dáta predstavovať prijatá hodnota DSCP, na základe ktorej bude taktiež príslušná prevádzka adekvátne uprednostnená.

Vstupná jednotka

Úlohou vstupnej dátovej jednotky je v tomto prípade generovanie dát (rámcov). Rámec bude vytvorený spojením všetkých nastavených dát pomocou maticového zlučovača. V tomto komponente nastavujeme počet vstupov, v našom prípade sú to 4 vstupy, do ktorých privádzame nasledovné dáta:

- typ protokolu,
- zdrojová MAC adresa,
- cieľová MAC adresa,
- hodnota DSCP.

Jednotlivé dáta sú implementované pomocou komponenty konštantnej hodnoty, kde je nadefinovaná príslušná dátová postupnosť bitov, ktoré vytvoria jeden vektor. Maticový zlučovač nám teda spojí jednotlivé vektory do jedného veľkého vektora, ktorý je vlastne postupnosť bitov a predstavuje zdrojový rámec. Cieľové MAC adresy sú

fixne nastavené v 4 komponentoch (podľa tab. 5.2) a pomocou spínačov môžeme voľiť požadovanú adresu cieľového zariadenia. Zdrojová MAC adresa je taktiež fixne nastavená podľa tab. 5.2.

Ďalej do vytvoreného rámca definujeme pole typ protokolu. Toto pole nebude v tejto simulácii využité, definujeme ho len z ilustračných dôvodov. Pole 16-bitové a v našom prípade definujeme protokol ipv4 – podľa 20 je tomuto typu protokolu pridelená hexadecimálna hodnota 0x0800.

Posledným poľom navrhnutého rámca je dôležité pole – DSCP. Pole taktiež odpovedá reálnej hodnote, a teda je tvorené 6 bitmi, ktoré zvolíme podľa tab. 5.1. Toto pole narozdiel od cieľovej MAC adresy môžeme voľiť ľubovoľne, keďže počet možných hodnôt DSCP je podstatne väčší.

Výstupná jednotka

Do výstupnej dátovej jednotky je privedený výstupný port prepínača. Pre overenie správnej činnosti zobrazíme prijatú hodnotu DSCP. Tak, ako boli pri tvorení rámca jednotlivé polia spojené pomocou maticového zlúčovača, tak potrebujeme požadované hodnoty z vytvoreného rámca vyfiltrovať. Použijeme k tomu komponent *Submatrix*, v ktorom definujeme index začiatku a konca dát, ktoré chceme z vektora vyfiltrovať. Napríklad v prípade filtrácie hodnoty DSCP bude index začiatku vektora (*Starting column index*) 113 a index konca (*Ending column index*) 118.

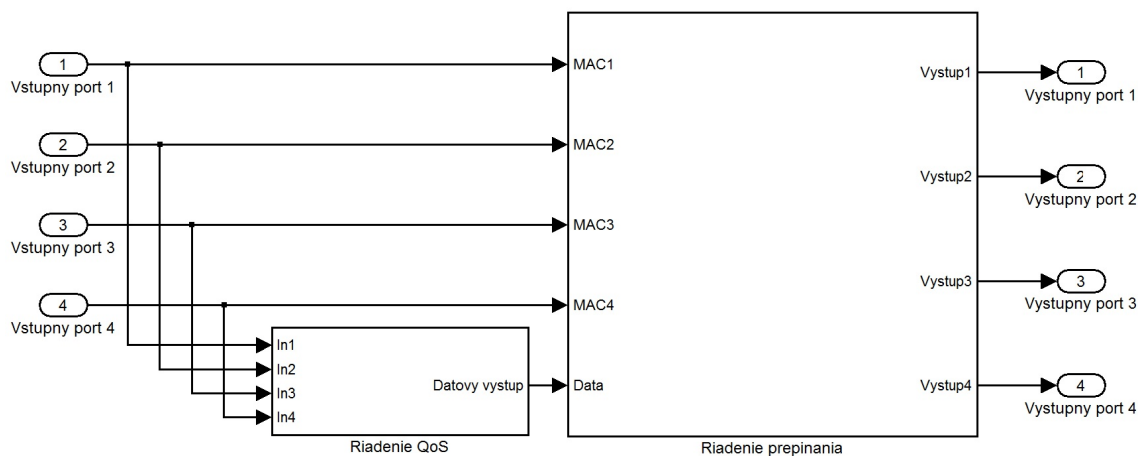
Ďalej je vo výstupnej jednotke fixne definovaná MAC adresa, ktorá je vždy fixne pridelená sieťovému rozhraniu. Budeme ju zobrazovať vo výstupnom zobrazovači dekadického čísla pre porovnanie s požadovanou cieľovou adresou nastavovanou v zdrojovej jednotke. Z MAC adresy je taktiež odrezaných posledných 8 bitov, aby bolo možné určiť totožnosť s MAC adresou, s ktorou prepínač v skutočnosti pracuje.

Telo prepínača

Model prepínača je rozdelený na niekoľko subsystémov, kde jednotlivé subsystémy plnia určitú funkcionálnosť. V prípade nepoužitia týchto subsystémov by bola schéma prepínača príliš rozsiahla a neprehľadná. Dva základné subsystémy prepínača sú:

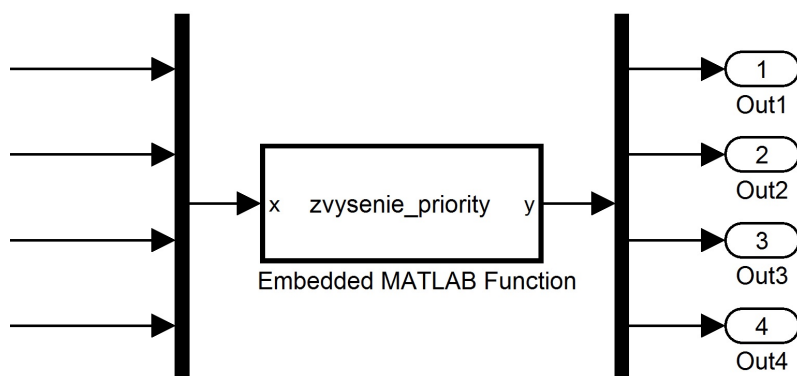
- riadenie QoS,
- riadenie prepínania.

Riadenie QoS má za úlohu určiť, v akom poradí budú dáta doručené do subsystému riadenie prepínania, kde budú dáta následne priradené na príslušný výstupný port. V subsystéme sa zo vstupných dát filtruje cieľová MAC adresa, ktorá je následne prevedená do dekadického tvaru. Filtruje sa iba potrebná časť MAC adresy, čiže posledných 8 bitov – z vytvoreného rámca sa jedná o bity 105-112.



Obr. 5.3: Telo prepínača

Okrem klasifikácie rámcov sú signály ešte privedené do subsystému priorit, ktorý rieši situáciu, v ktorej by 2 rámce s rovnakou prioritou mali byť prepnuté na rovnaký výstupný port. Jeden z týchto rámcov je treba uprednostniť, a teda zvýšiť mu prioritu. Algoritmus je implementovaný v komponente *Embedded MATLAB Function*, viď obr. 5.4.



Obr. 5.4: Zvýšenie priority

V subsystéme radenie priorit sú jednotlivé priority zoradené vzostupne. Zoradenie je realizované pomocou algoritmu *Bubblesort*. Ďalej sú príslušné páry signálov navzájom porovnávané, aby subsystém postupne obslúžil aj rámce s najnižšou prioritou.

Pre implementáciu poradia prepínania je potrebné vložiť časovač (*trigger*), ktorý riadi pomocný spínač. Tento prepínač vysiela do spínača poradia postupnosť riadiacich pulzov podľa príslušnej kvality služby. Spínače sú zoradené v kaskáde a dáta sa prenášajú sériovo. Subsystém radenie priorit generuje signály pre spínač poradia, ktorý má hodnotu 1-4 a bude ovládať v zoradenom poradí hlavný spínač.

V modeli prepínača je ARP tabuľka implementovaná v subsystéme riadenie prepínania. V tomto prípade nám postačí, ak budú v ARP tabuľke uložené záznamy o zariadeniach, ktoré sú pripojené na výstupných portoch prepínača. Následne sú tieto záznamy z tabuľky načítavané a porovnávané s prijatými MAC adresami nastavenými na zdrojovej stanici. Taktiež pracujeme iba s poslednými 8 bitmi, ktoré ďalej konvertujeme do dekadického podoby.

Keďže v modeli môžu byť len 4 MAC adresy, prepojí sa odpovedajúca adresa tým spôsobom, že portový spínač dostane povel na prepnutie dát, inak prepojí nulu. Vo výstupe sú nakoniec signály dát prepnutých podľa príslušnej MAC adresy nastavenej na zdrojovej stanici a v poradí podľa príslušnej hodnoty DSCP.

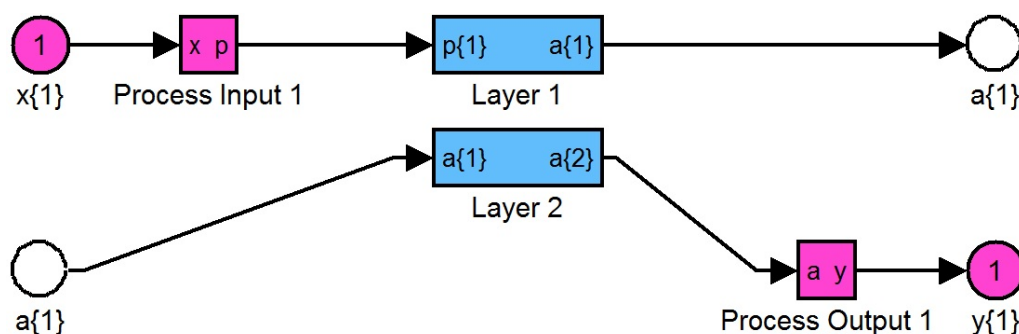
5.2.4 Neurónová sieť

Neurónová sieť je výpočtový model zostavený na základe abstrakcie vlastností biologických nervových systémov. Modelovanie neurónovej siete vychádza z nasledujúceho vzťahu:

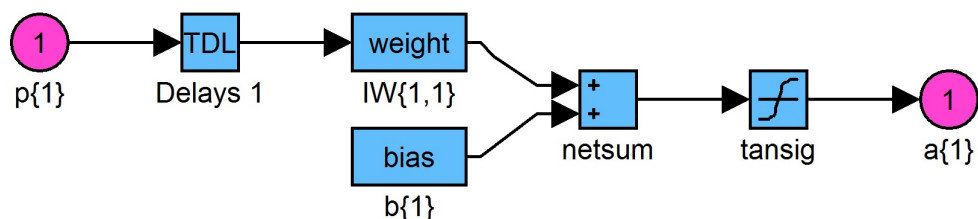
$$a = f\left(\sum_{i=1}^n w_i x_i + b\right), \quad (5.1)$$

kde a je výstupný signál neurónu, f je funkcia prenosu signálu (aktivačná funkcia neurónu), n je počet vstupov do neurónu, w_i sú váhy (*weights*) vstupných brán, x_i je vstupný signál neurónu snímaný zo vstupu alebo zo spolupracujúceho neurónu a b je hodnota prahu (*bias*).

Jednoduchá neurónová sieť (tzv. lineárny model neurónovej siete) môže byť jednovrstvová alebo viacvrstvová. V tejto laboratórnej úlohe je použitá dvojvrstvová neurónová sieť, ktorú je možné vytvoriť pripraveným skriptom *script.m*. Jej štruktúra je zobrazená na obr. 5.5. Neurónová sieť je tvorená vstupnou a výstupnou vrstvou. Každá z vrstiev je tvorená blokmi charakteristickými pre neurónovú sieť podľa vzťahu 5.1 – oneskorovací člen, váhy (*weights*), hodnota prahu (*bias*), súčet a funkcia prenosu signálu (aktivačná funkcia neurónu), viď obr. 5.6.



Obr. 5.5: Štruktúra neurónovej siete



Obr. 5.6: Štruktúra vrstvy neurónovej siete

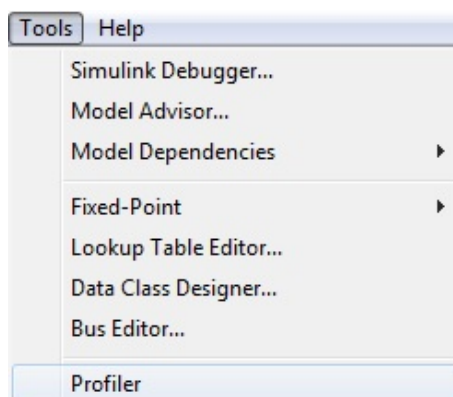
Sieť má nastavené váhové a prahové hodnoty tak, aby konkrétnej adrese priradila príslušný výstup. Neurónová sieť teda zvolí, na ktorý port bude daný rámec prepnutý. Výstupom neurónovej siete je reálne číslo veľmi blízke hodnote 0 alebo 1. Na základe tejto hodnoty ďalej prepínač určí, či má byť rámec na daný výstupný port poslaný alebo nie. Preto je potrebné hodnotu zaokrúhliť na presnú hodnotu 0, resp. 1. Toto zaokrúhlenie je implementované v bloku *Porovnanie*, kde prebieha prevod na absolútnu hodnotu a následne porovnanie s referenčnou hodnotou a vrátenie hodnoty 1 v prípade, že je hodnota väčšia ako nastavená referenčná hodnota.

Signál je následne privedený do 4 spínačov, ktoré pošlú na výstup vyfiltrovanú hodnotu DSCP, pokiaľ sa na ich riadiacom porte objaví hodnota 1. Výstupy týchto spínačov sú ďalej už privedené do cieľových staníc, kde je na displeji zobrazená príslušná hodnota DSCP.

5.3 Pracovný postup

1. Spustíte program Matlab a otvorte v ňom model prepínača umiestnený na ploche (*prepinac.mdl*).
2. Zoznámte sa s jednotlivými blokmi a preštudujte ich činnosť.
3. Navrhnete vhodnú prioritu a priradíte príslušné hodnoty DSCP podľa tab. 5.1 pre služby VoIP, IPTV, surfovanie na webe a tlač dokumentov.
4. V blokoch vstupných zariadení nastavte vybrané hodnoty DSCP tak, aby každý užívateľ používal práve jednu službu.
5. Podobne nastavte hodnoty tak, aby jednu zo služieb využívalo viac užívateľov.
6. Pomocou skriptu *script.m* vytvorte a natrénujte neurónovú sieť a analyzujte jej parametre a výkonnosť (skript je možné spustiť z príkazového riadka Matlabu, skúste si aj postupné zadávanie jednotlivých príkazov zo skriptu a snažte sa porozumieť funkcionalite jednotlivých príkazov).
7. Simuláciu z bodov 4 a 5 preveďte aj s prepínačom riadeným neurónovou sieťou (*prepinac-neural.mdl*).
8. Zmerajte dobu dĺžky simulácie postupne pre 1 – 4 počítače využívajúce službu IPTV, viď obr. 5.8 (*Tools -> Profiler* – obr. 5.7).

9. Porovnajete oba modely prepínača. Výsledky analyzujete, vyneste do grafu a prezentujete vyučujúcemu.



Obr. 5.7: Nastavenie merania parametrov simulácie

Simulink Profile Report: Summary

Report generated 11-Apr-2015 00:49:01

Total recorded time:	0.76 s
Number of Block Methods:	311
Number of Internal Methods:	6
Number of Nonvirtual Subsystem Methods:	7
Clock precision:	0.00000005 s
Clock Speed:	2200 MHz

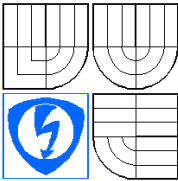
To write this data as prepinacl3ProfileData in the base workspace click [here](#)

Obr. 5.8: Meranie doby dĺžky simulácie

5.4 Otázky

1. Z koľkých bitov sa skladá DSCP?
2. V akých aplikáciách je dôležitá kvalita služby?
3. Ktorý transportný protokol používa VoIP a prečo?
4. Ktoré dva základné prvky tvoria biologický neurón?
5. V ktorých blokoch prepínača by bolo možné využiť neurónové siete?
6. V ktorom bloku prepínača bola neurónová sieť využitá v tejto laboratórnej úlohe a prečo?
7. Z ktorých prvkov sa skladá neurónová sieť vytvorená v tejto laboratórnej úlohe a popíšte ich funkcionality.

6 VZOROVÝ PROTOKOL

	Predmet	
	Meno	
	Ročník	Študijná skupina
	Spolupracoval	Merané dňa
Kontroloval	Hodnotenie	Dňa
Číslo úlohy	Názov úlohy <h3 style="text-align: center;">SIMULÁCIA PREPÍNAČA</h3>	

6.1 Zadanie

1. Preštudujte teóriu prepínania, neurónových sietí, QoS a architektúru prepínača.
2. Navrhните hodnoty kvality služby pre zadané typy služieb.
3. Simulujte rôzne možnosti prevádzky pre rôzny počet užívateľov.
4. V prostredí Matlab vytvorte a analyzujte neurónovú sieť.
5. Simulujte model prepínača riadeného neurónovou sieťou.
6. Porovnajete oba modely prepínačov.
7. Výsledky analyzujte a vyhodnoťte.

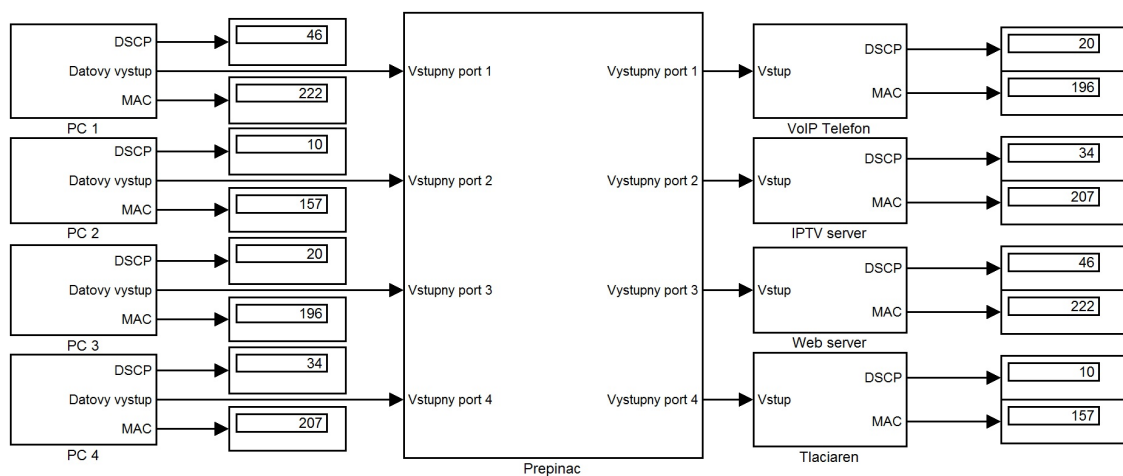
6.2 Vypracovanie

6.2.1 Voľba DSCP hodnôt

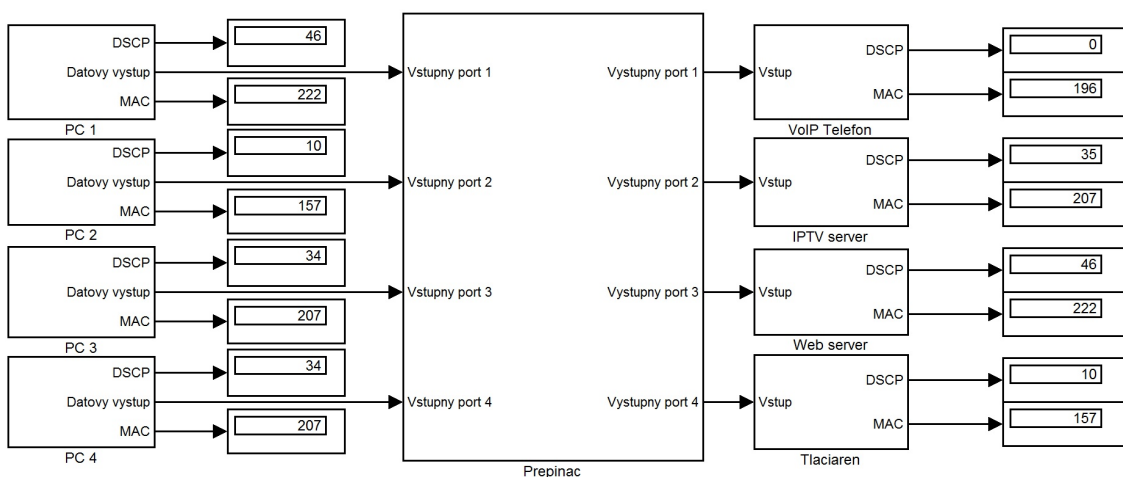
Tab. 6.1: Zvolené hodnoty DSCP

Služba	DSCP hodnota	Decimálna hodnota	Význam
VoIP	101 110	46	EF
IPTV	100 010	34	AF41
Web	010 100	20	AF22
tlač	001 010	10	AF11

6.2.2 Simulácia klasického prepínača



Obr. 6.1: Simulácia 1

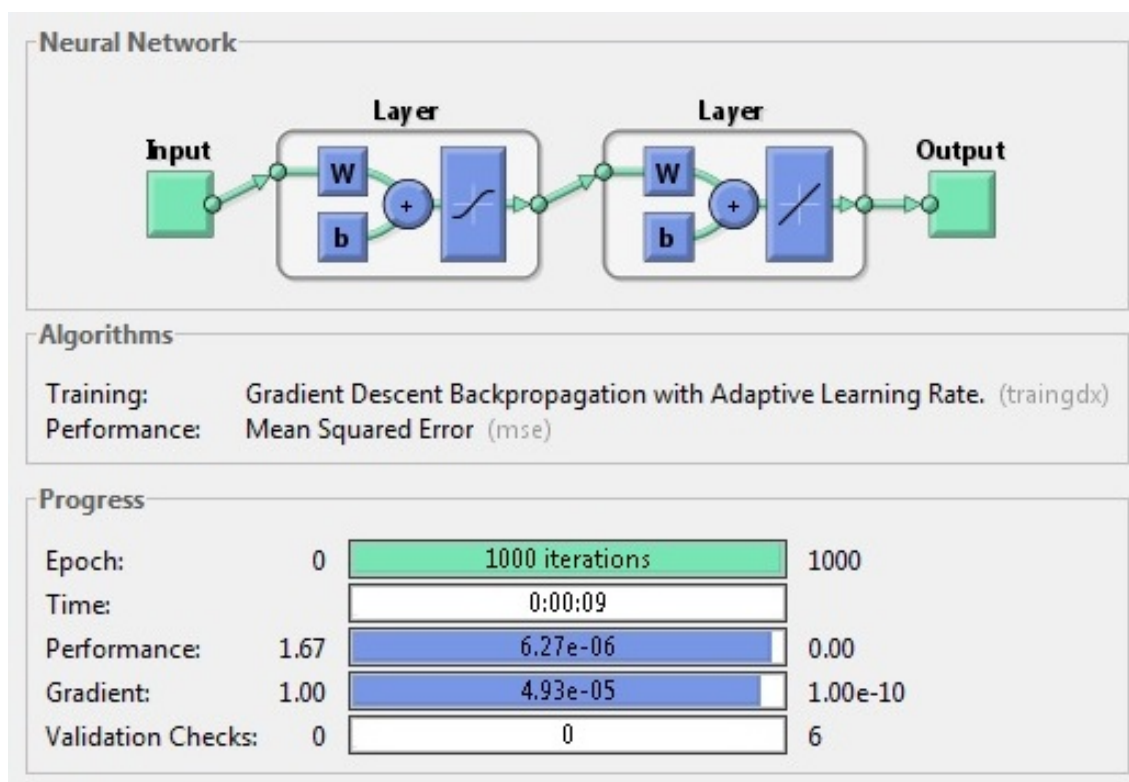


Obr. 6.2: Simulácia 2

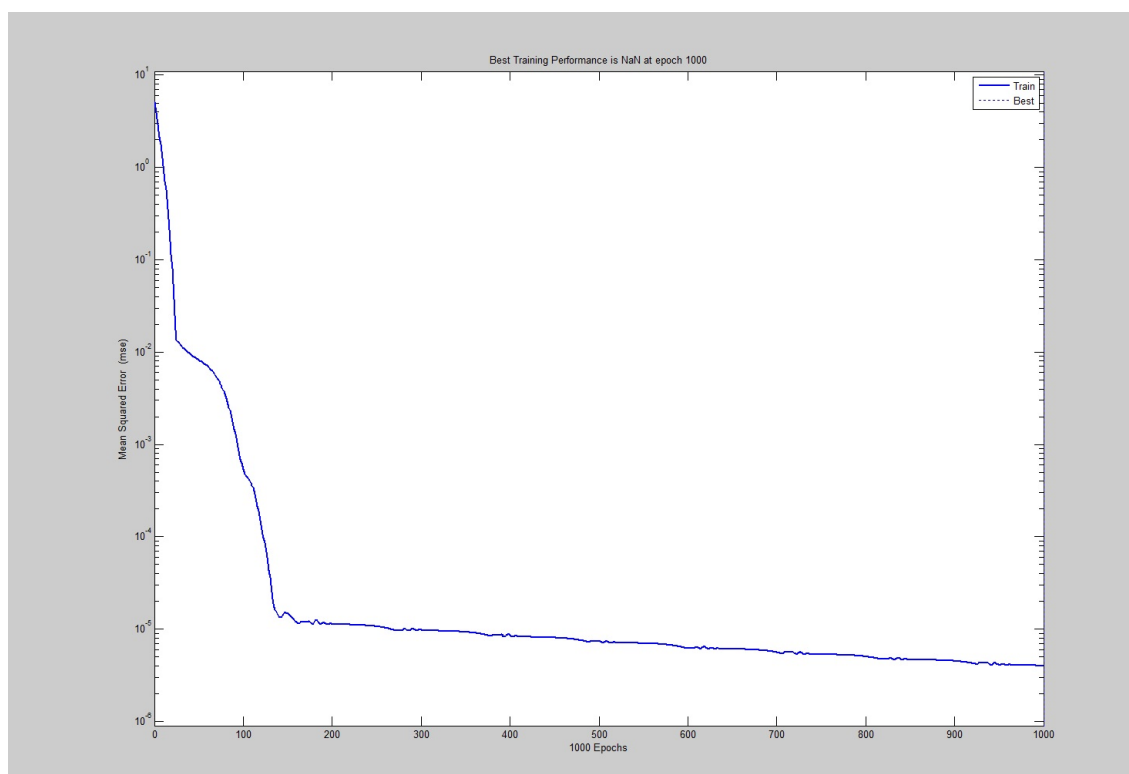
6.2.3 Analýza neurónovej siete a simulácia prepínača riadeného neurónovou sieťou

Zobrazenie vstupných dát a dát výstupných naučených neurónovou sieťou (obr. 6.7):

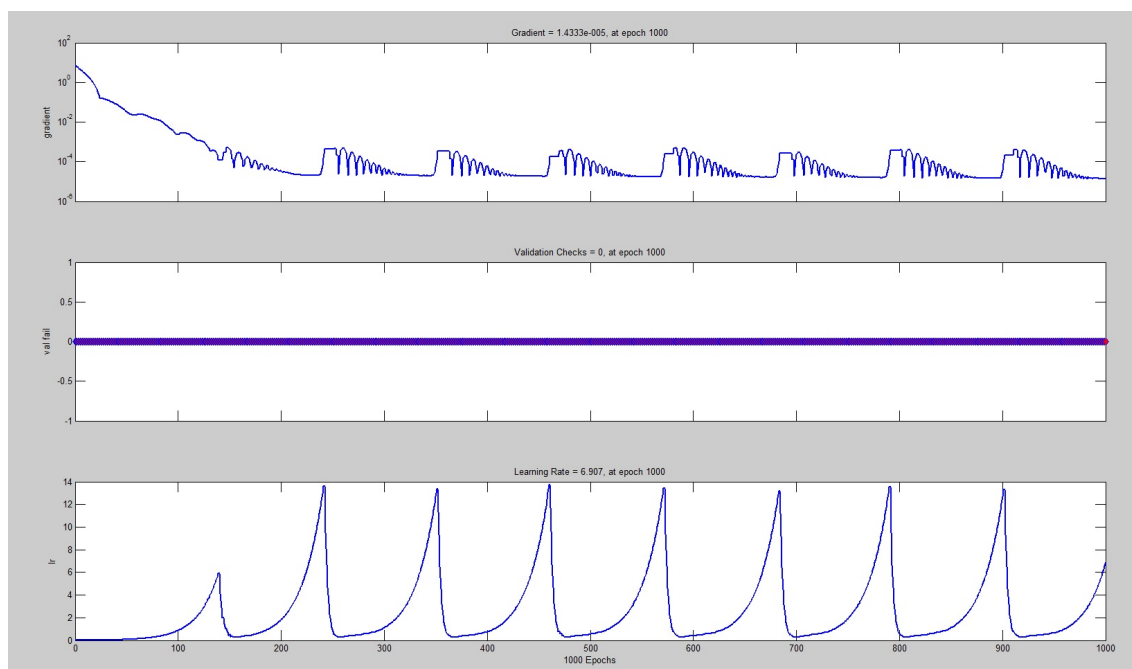
```
[OutputVector] = sim(net,InputVector);
subplot(2,1,1)
plot(InputVector,TargetVector,'x-');
subplot(2,1,2);
plot(InputVector,OutputVector,'x-');
```



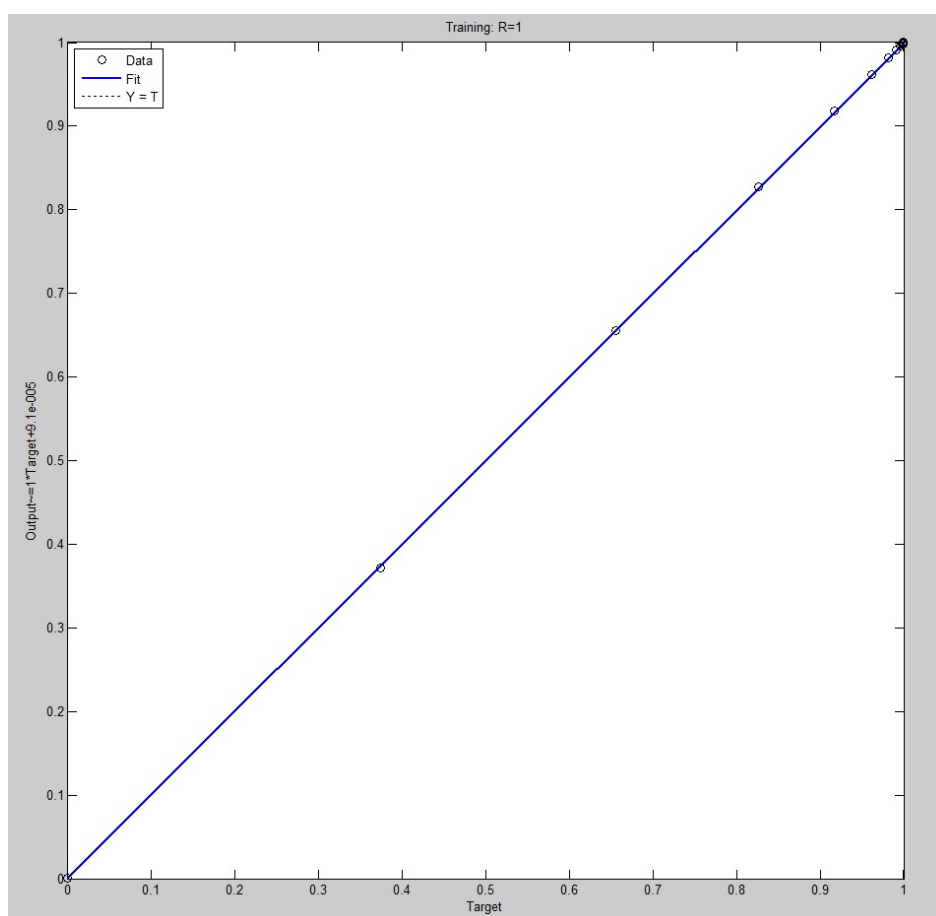
Obr. 6.3: Trénovanie neurónovej siete



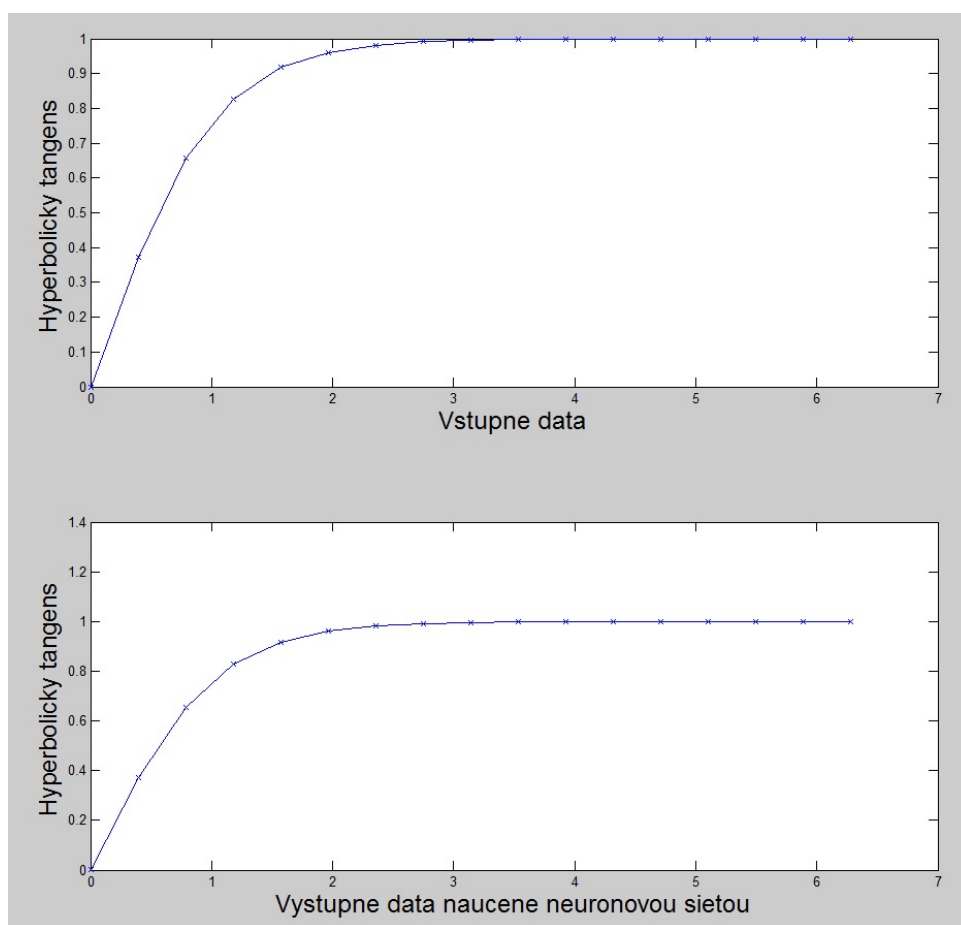
Obr. 6.4: Trénovacia chybová krivka



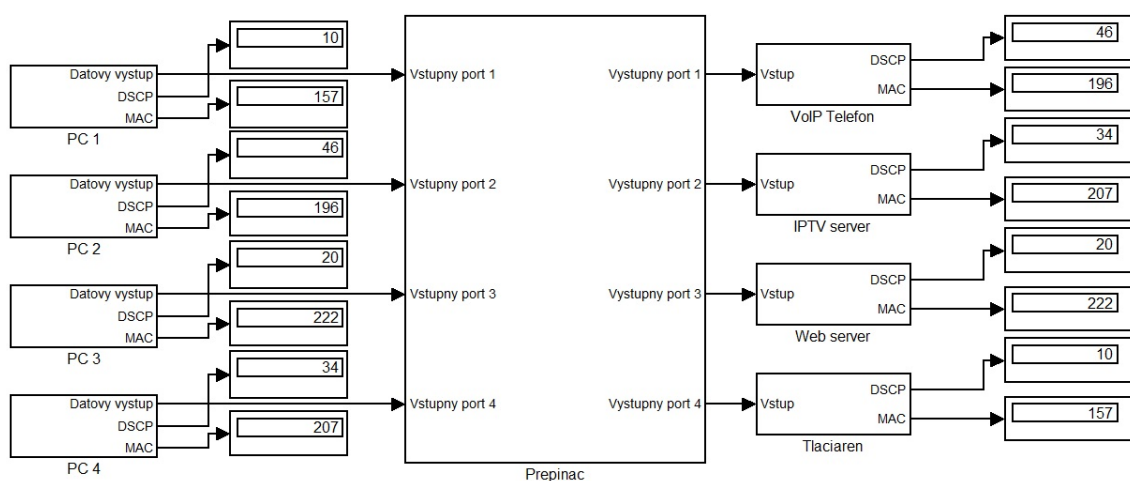
Obr. 6.5: Parametre tréningu neurónovej siete



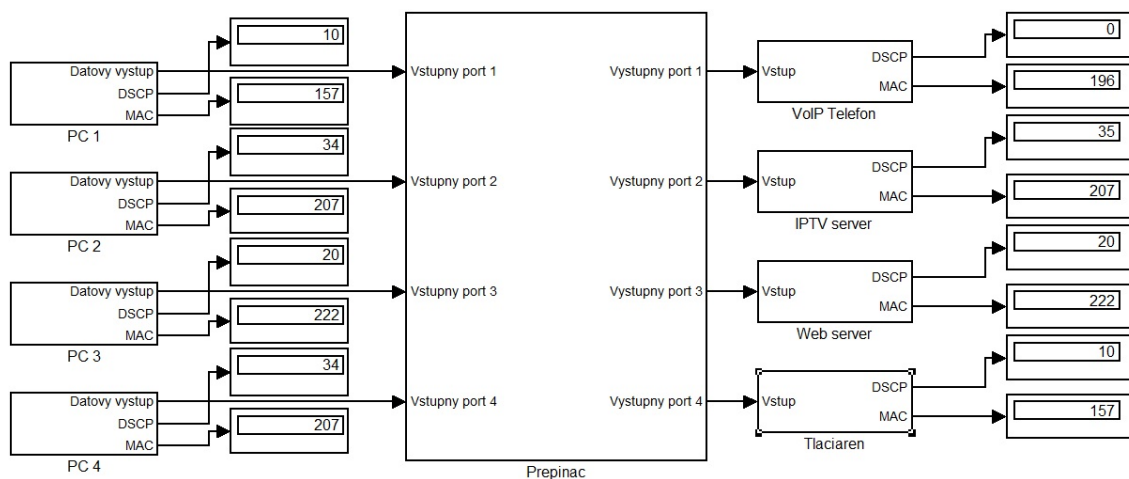
Obr. 6.6: Regresia neurónovej siete



Obr. 6.7: Porovnanie vstupných dát a dát naučených neurónovou sieťou



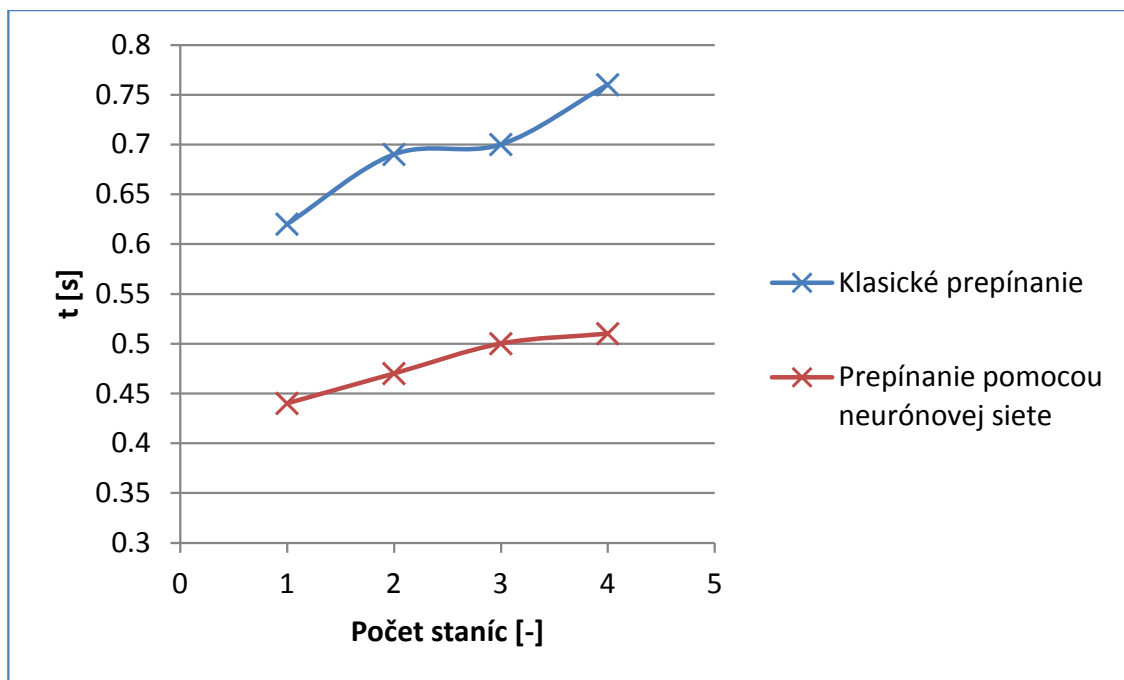
Obr. 6.8: Simulácia 3 – prepínač riadený neurónovou sieťou



Obr. 6.9: Simulácia 4 – prepínač riadený neurónovou sieťou

Tab. 6.2: Doba simulácie pre počet staníc dožadujúcich sa IPTV služby

Počet počítačov	Doba simulácie [s]	
	Klasický prepínač	Prepínač riadený neurónovou sieťou
1	0.62	0.44
2	0.69	0.47
3	0.7	0.5
4	0.76	0.51



Obr. 6.10: Závislosť doby simulácie na počte staníc dožadujúcich sa IPTV služby

6.3 Odpovede na otázky

1. Zo 6.
2. V aplikáciách, ktoré poskytujú služby prebiehajúce v reálnom čase, ako napríklad VoIP, videokonferencia, IPTV.
3. Protokol UDP, pretože zaistuje i keď nespoľahlivý, ale kontinuálny a rýchly prenos dát.
4. Dendrity a axón.
5. Využitie by bolo možné v klasifikácii prevádzky, správe front a v riadení spojovacieho poľa.
6. V riadení spojovacieho poľa, pretože je toto riešenie najviac efektívne.
7. Váhy (*weights*), hodnota prahu (*bias*) – slúžia na určenie príslušného výstupného portu. Sigmoidná aktivačná funkcia neurónu, ktorá je matematickou transformáciou súhrnu váh, ktoré tvoria výstup neurónu.

6.4 Záverečné zhodnotenie

V prvej simulácii je simulovaný prípad, kde PC1 vyžaduje surfovanie na webe, PC2 tlač dokumentov, PC3 telefonovanie prostredníctvom VoIP a PC4 sledovanie televízie prostredníctvom IPTV. Vo výstupe je vidno, že priorita odpovedá prioritě nastavenej na strane vstupu pre príslušnú MAC adresu cieľovej stanice, a teda každá služba bola podľa nastavenej priority adekvátne obslužená.

V druhom prípade predpokladáme, že PC3 a PC4 vyžadujú súčasne službu IPTV. Vo výstupe je vidno, že priorita je pre túto službu zvýšená o 1. V prípade VoIP služby je na výstupe 0, keďže táto služba v tomto prípade nebola využitá.

Pomocou príkazov zo skriptu sme vytvorili, trénovali a analyzovali neurónovú sieť. Počas trénovania sme sledovali výkon neurónovej siete, ktorý bol vynesý do grafu 6.4. Trénovacia krivka má v závislosti na počte iterácií klesajúci charakter. Ostatné parametre tréningu neurónovej siete sú zobrazené na obr. 6.5 a 6.6. Overenie funkčnosti neurónovej siete sme vykonali transformáciou vstupnej funkcie na výstupnú funkciu cez neurónovú sieť ($[OutputVector] = sim(net, InputVector)$). Vstupnú a výstupnú funkciu sme vyniesli do grafu 6.7, v ktorom vidíme totožnosť oboch funkcií, z čoho je zrejmé, že neurónová sieť bola natrénovaná správne.

Rovnakým spôsobom sme simulovali aj prepínač s neurónovou sieťou. Výsledok ukazuje správnu funkčnosť pre obe varianty simulácie. V poslednom bode sme zmerali dĺžku doby simulácie, ktorá ilustračne odpovedá reálnym prepínačom – so stúpajúcim počtom počítačov dožadujúcich sa služby IPTV, ktorá spôsobí najväčšie zahľtenie siete, je doba simulácie väčšia. Doba simulácie je všeobecne väčšia pre prepínač s klasickým prepínaním – prepínač s neurónovou sieťou sa javí výkonnejší.

7 ZÁVER

Úvodom je v práci opísaný model ISO/OSI, v ktorom je patrične zaradený prepínač a jeho opis. Opísaná je jak vnútorná architektúra prepínača, tak jeho výkonnosť a parametre. Ďalej sú v práci opísané požiadavky na kvalitu služieb, mechanizmy integrácie a spôsoby riadenia odosielania paketov. Ďalej je uvedená štruktúra rámcov a paketov a možnosti zaistenia kvality služieb na základe hodnoty DSCP. Teoretická časť zahŕňa taktiež teóriu neurónových sietí opísanú od biologického neurónu až po vytvorenie a popis umelej neurónovej siete.

V prostredí MATLAB-SIMULINK sú navrhnuté dva modely prepínača, ktoré predstavujú zjednodušenú verziu reálneho prepínača vhodnú pre edukatívne účely. Prvý model je založený na klasickom prepínaní a druhý je riadený neurónovou sieťou. Prepínač je zaradený v topológii, v ktorej sú na vstupných portoch pripojené 4 počítače používateľov, ktorí môžu využívať služby vyžadujúce rôznu kvalitu služby. Na strane výstupu prepínača je teda VoIP telefón koncového užívateľa, IPTV server, webový server a tlačiareň. Sú teda možné 4 typy služieb s rôznou požiadavkou na QoS.

Oba modely prepínača sú principiálne rozdelené na dve základne časti, kde prvá zaisťuje poradie, v ktorom budú dáta prepínané na základe nastavenej hodnoty DSCP a druhá časť realizuje samotné prepínanie na základe nastavenej cieľovej stanice na strane zdrojového zariadenia. Keďže sú v topológii 4 možné cieľové stanice, tak na zdrojovej stanici je možné pomocou spínačov nastaviť 4 rôzne MAC adresy cieľových staníc. Hodnota DSCP sa nastavuje manuálne v bitovej podobe na základe pripravenej tabuľky. Ošetrený je taktiež prípad, v ktorom by viacerí užívatelia vyžadovali rovnaký typ služby. V tomto prípade je dátam jednej z týchto služieb zvýšená priorita pomocou navrhnutého algoritmu.

Riadenie prepínania je v prvom modele realizované na základe porovnávania MAC adresy z prijatých dát a MAC adresy uloženej v ARP tabuľke prepínača. ARP tabuľka reálneho prepínača obsahuje každú rozpoznanú MAC adresu. V tomto prípade sú v tabuľke uložené práve 4 adresy cieľových staníc.

V druhom prípade je riadenie prepínania realizované pomocou neurónovej siete, ktorá bola vygenerovaná a natrénovaná pomocou Matlabu. Počas tréningu sa vykresľuje tréningová chybová krivka, ktorá ukazuje, ako postupne klesá chybová funkcia tréningu. Správna funkčnosť neurónovej siete bola overená transformáciou vstupnej funkcie na výstupnú funkciu cez neurónovú sieť. Výsledky boli vynesené do grafu, v ktorom je vidno totožnosť oboch funkcií, z čoho je zrejmé, že neurónová sieť bola natrénovaná správne. Sieť je tvorená vstupnou a výstupnou vrstvou, pričom každá vrstva má nastavené hodnoty váh a prahu tak, aby konkrétnej adrese priradila príslušný výstup, a teda zaistila správne prepnutie rámcu.

Na základe navrhnutých modelov je v práci vytvorená laboratórna úloha. Zahrnutý je štandardne teoretický úvod, v ktorom je opísaná kvalita služieb a štruktúra navrhnutých modelov prepínača. Cieľom laboratórnej úlohy je vhodne určiť hodnoty DSCP pre konkrétny typ služby a návrh odsimulovať. Ďalej je v laboratórnej úlohe zahrnuté vytvorenie a analýza neurónovej siete. Pre túto časť bol vytvorený skript, pomocou ktorého je možné vytvoriť a otestovať neurónovú sieť jedným príkazom. Tento skript je vhodný ako pomôcka pre vyučujúceho. Študentom je v návode laboratórnej úlohy doporučené aplikovať príkazy zo skriptu postupne pre lepšie porozumenie problematike neurónových sietí. Pracovný postup laboratórnej úlohy je písaný tak, aby bolo možné úlohu splniť v časovom rozsahu dvoch vyučovacích hodín. Na záver sú uvedené kontrolné otázky pre študentov.

Oba modely boli v rámci laboratórnej úlohy odsimulované. Správna funkčnosť modelov je zrejmá z totožnosti dát zobrazených na vstupných a výstupných jednotkách topológie siete. V rámci laboratórnej úlohy je výkonnosť oboch modelov porovnaná pomocou dĺžky doby simulácie. Hodnoty sú rádovo v sekundách, takže porovnanie je iba ilustračné – v reálnom prepínači by prepnutie jedného rámca trvalo oveľa menej (v závislosti na výkonnosti prepínača, rýchlosti portov prepínača a veľkosti rámca). Testovanie bolo prevedené pre službu IPTV, ktorá spôsobuje najväčšie zahltenie siete, a teda má nastavenú hodnotu DSCP s najvyššou prioritou. Výsledok ukazuje, že so stúpajúcim počtom počítačov dožadujúcich sa služby IPTV je doba simulácie väčšia, čo odpovedá reálnej situácii, kde sa so zvýšením zahltenia siete znižuje prenosová rýchlosť dát. Doba simulácie je všeobecne väčšia pre prepínač s klasickým prepínaním, takže prepínač s neurónovou sieťou sa javí výkonnejší. Všetky výstupy simulácií sú zahrnuté v rámci vzorového protokolu, v ktorom sú taktiež odpovede na otázky a záverečné zhodnotenie.

LITERATÚRA

- [1] MOLNÁR, K. *Praktikum z informačních sítí*. Brno: Vysoké učení technické v Brně, 2013. s. 1-135. ISBN: 978-80-214-4715-8. (cs)
- [2] NOVOTNÝ, V. *Architektura sítí*. Brno: Vysoké učení technické v Brně, 2011. s. 1-152. ISBN: 978-80-214-4715-8. (cs)
- [3] Networking Transport [online]. [cit. 2014-11-14]. Dostupné z WWW: <<http://fab.cba.mit.edu/classes/MIT/863.07/people/david/guru/index.html>>.
- [4] Hub Definition [online]. 2005 [cit. 2014-11-18]. Dostupné z WWW: <<http://www.linfo.org/hub.html>>.
- [5] JEŘÁBEK, J. Pokročilé komunikační techniky. Vysoké učení technické v Brně, 2012. s. 1-195. ISBN: 978-80-214-4636-6. (cs)
- [6] JANEČEK, J., BÍLÝ, M.: Lokální sítě. Vydavatelství ČVUT, Praha, 2004. ISBN: 80-01-02900-X.
- [7] POLÍVKA, M. *Kvalita služby v konvergovaných systémech s prvky řízenými neuronovou sítí*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 120 s. Vedoucí dizertační práce doc. Ing. Vladislav Škorpil, CSc.
- [8] MOJŽIŠ, Lubomír. *Perspektivní využití technologie CISCO*: bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 89 s. Vedúci práce bol doc. Ing. Vladislav Škorpil, CSc.
- [9] KOTON, J. Moderní síťové technologie. Vysoké učení technické v Brně, 2012. s. 1-195. ISBN: 978-80-214-4636-6. (cs)
- [10] Differences between a Switch, a Router, And a Layer 3 Switches [online]. 2013 [cit. 2014-11-27]. Dostupné z: <www.cisco1900router.com/tutorial-on-differences-between-a-switch-a-router>.
- [11] Introduction to QoS – Classification and Marking [online]. [cit. 2014-11-27]. Dostupné z WWW: <http://fengnet.com/book/ios_mpls/ch13lev1sec1.html>.
- [12] *Cisco.com* [online]. [cit. 2014-11-27]. DSCP and Precedence Values. Dostupné z WWW: <http://www.cisco.com/en/US/docs/switches/datacenter/nexus1000/sw/4_0/qos/configuration/guide/qos_6dscp_val.pdf>.

- [13] MOLNAR, K. *Hardware počítačových sítí*, Brno: Vysoké učení technické v Brně, 2011, s. 1-157, ISBN: 978-80-214-4449-2 (cs)
- [14] *Cisco.com* [online]. [cit.2014-11-28]. Introduction to Gigabit Ethernet. Dostupné z WWW: <http://www.cisco.com/en/US/tech/tk389/tk214/tech_brief09186a0080091a8a.html>.
- [15] *Mathworks.com* [online]. 2014 [cit.2014-11-28]. Simulink overview. Dostupné z WWW: <<http://www.mathworks.com/products/simulink/>>.
- [16] *Mathworks.com* [online]. 2014 [cit.2014-11-28]. Supported and Compatible Compilers – Release 2010a. Dostupné z WWW: <<http://www.mathworks.com/support/compilers/R2010a/win32.html>>.
- [17] Simulink Basics Tutorial [online]. 2012 Dostupné z WWW: <http://ctms.engin.umich.edu/CTMS/index.php?aux=Basics_Simulink>.
- [18] MAC Address Definition [online]. 2005 [cit.2014-12-07]. Dostupné z WWW: <http://www.linfo.org/mac_address.html>.
- [19] MAC address standards [online]. 2014 [cit.2014-12-07]. Dostupné z WWW:<<http://www.ieee.org/netstorage/standards/oui.txt>>.
- [20] Ethernet types standards [online]. 2014 [cit.2014-12-07]. Dostupné z WWW:<<http://standards.ieee.org/develop/regauth/ethertype/eth.txt>>.
- [21] Bubble sort [online]. 2014 [cit.2014-12-08]. Dostupné z WWW:<<http://www.algoritmy.net/article/3/Bubble-sort>>.
- [22] BOHÁČ, Martin. *Implementace kvality služby do řízení síťového prvku*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2008. 73 s. Vedúci práce bol doc. Ing. Vladislav Škorpil, CSc.
- [23] ŠVEC, Adam. *Optimalizace přepínače v konvergované síti*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 88 s. Vedúci práce bol doc. Ing. Vladislav Škorpil, CSc.
- [24] *Cisco.com* [online]. [cit.2014-12-10]. Dynamic ARP Inspection. Dostupné z WWW: <<http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/dynarp.html>>.

- [25] *Cisco.com* [online]. [cit. 2014-12-11]. Understanding QoS as Implemented in the Solution. Dostupné z WWW: <http://www.cisco.com/en/US/products/ps6902/products_implementation_design_guide_chapter09186a0080666638.html>.
- [26] KVASNIČKA V., BEŇUŠKOVÁ L., POSPÍCHAL J., FARKAŠ I., TIŇO P. a KRÁL A. 1997 [cit. 2015-04-09]. *Úvod do teórie neurónových sietí*. Iris: Bratislava.
- [27] *Mathworks.com* [online]. 2015 [cit. 2015-04-11]. Simulink overview. Dostupné z WWW: <<http://www.mathworks.com/help/nnet/ref/gensim.html>>.
- [28] KAJAN S. *Úvod do teórie umelých neurónových sietí*. [online]. 2007, s. 40 [cit. 2015-04-15]. Dostupné z WWW: <http://www.kasr.elf.stuba.sk/predmety/is/prednasky/is11_p5doplnok.pdf>.
- [29] TUČKOVÁ J. *Algoritmy a struktury neuropočítačů*. [online]. 2014 [cit. 2015-04-26]. Dostupné z WWW: <<http://amber.feld.cvut.cz/ssc/>>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

AF	Assured Forwarding
ARP	Address Resolution Protocol
BE	Best-effort
CB WFQ	Class-Based Weighted Fair Queuing
CPU	Central Processing Unit
CS	Class Selector
DS	Differentiated Services
DSCP	Differentiated Services Code Point
EF	Expedited Forwarding
FCS	Frame Control Check
FIFO	First In First Out
FQ	Fair Queuing
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol
IOS	Internetwork Operating System
IP	Internet Protocol
IPTV	Internet Protocol Television
ISO	International Organization for Standardization
LAN	Local Area Network
L3	Layer 3
LLC	Logical Link Control
MAC	Medium Access Control
NPU	Network Processing Unit
OSI	Open System Interconnection

PHB	Per-Hop Behavior
PQ	Priority Queuing
QoS	Quality of Service
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
ToS	Type of Service
UDP	User Datagram Protocol
VoIP	Voice over IP
WAN	Wide Area Network
WFQ	Weighted Fair Queuing
WRR	Weighed Round Robin

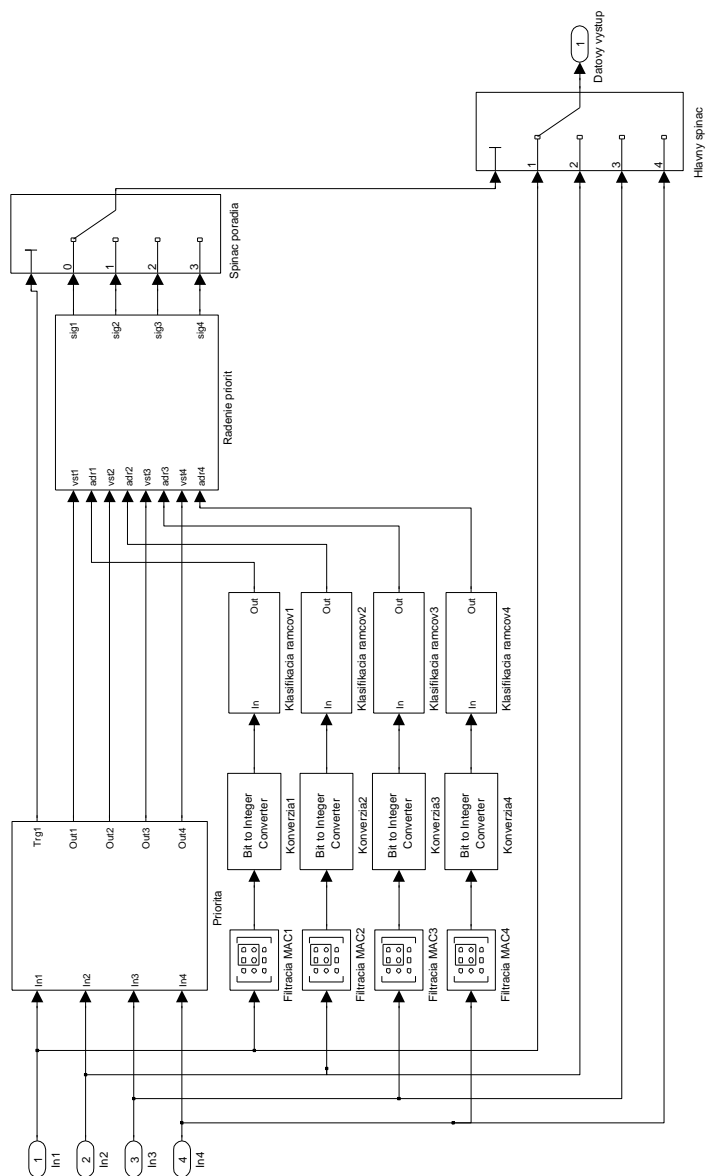
ZOZNAM PRÍLOH

A	Obsah CD	75
B	Štruktúra prepínača	76

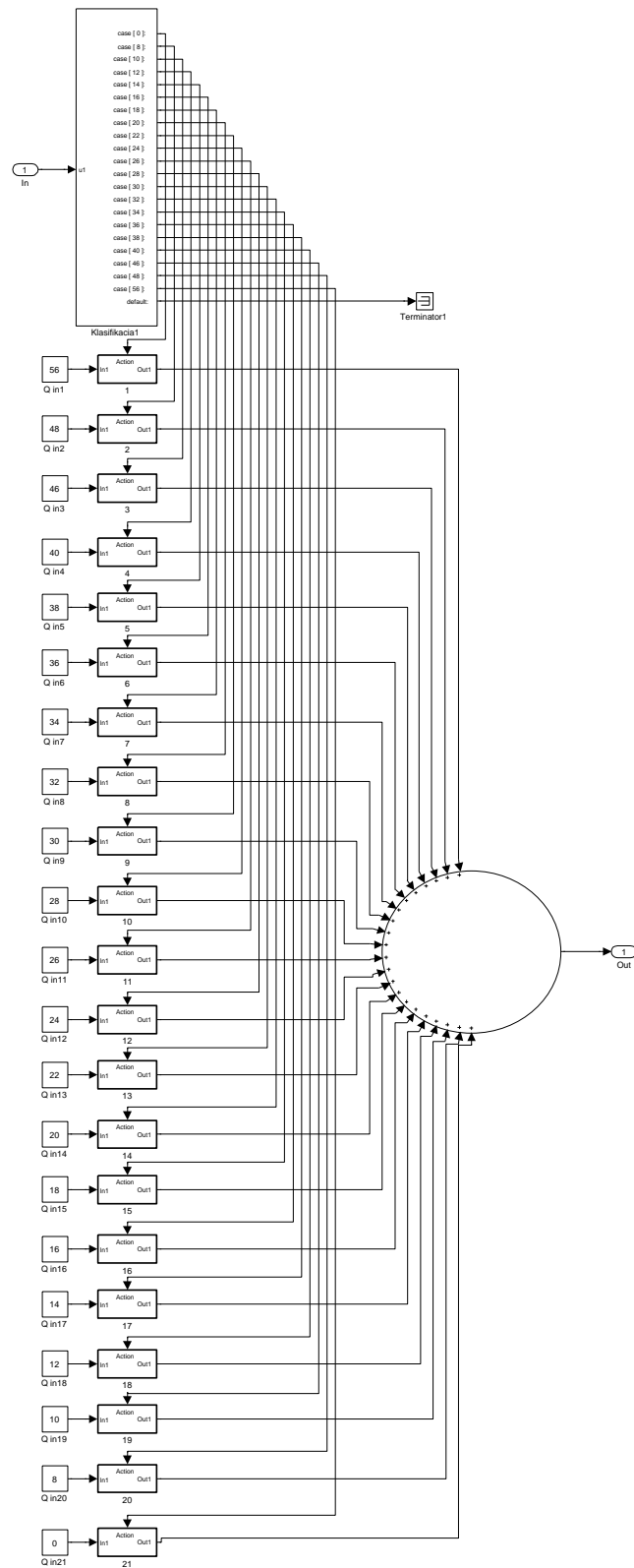
A OBSAH CD

- Elektronická verzia diplomovej práce,
- model prepínača s klasickým prepínaním,
- model prepínača riadeného neurónovou sieťou,
- skript na vytvorenie a testovanie neurónovej siete,
- laboratórna úloha,
- vzorový protokol.

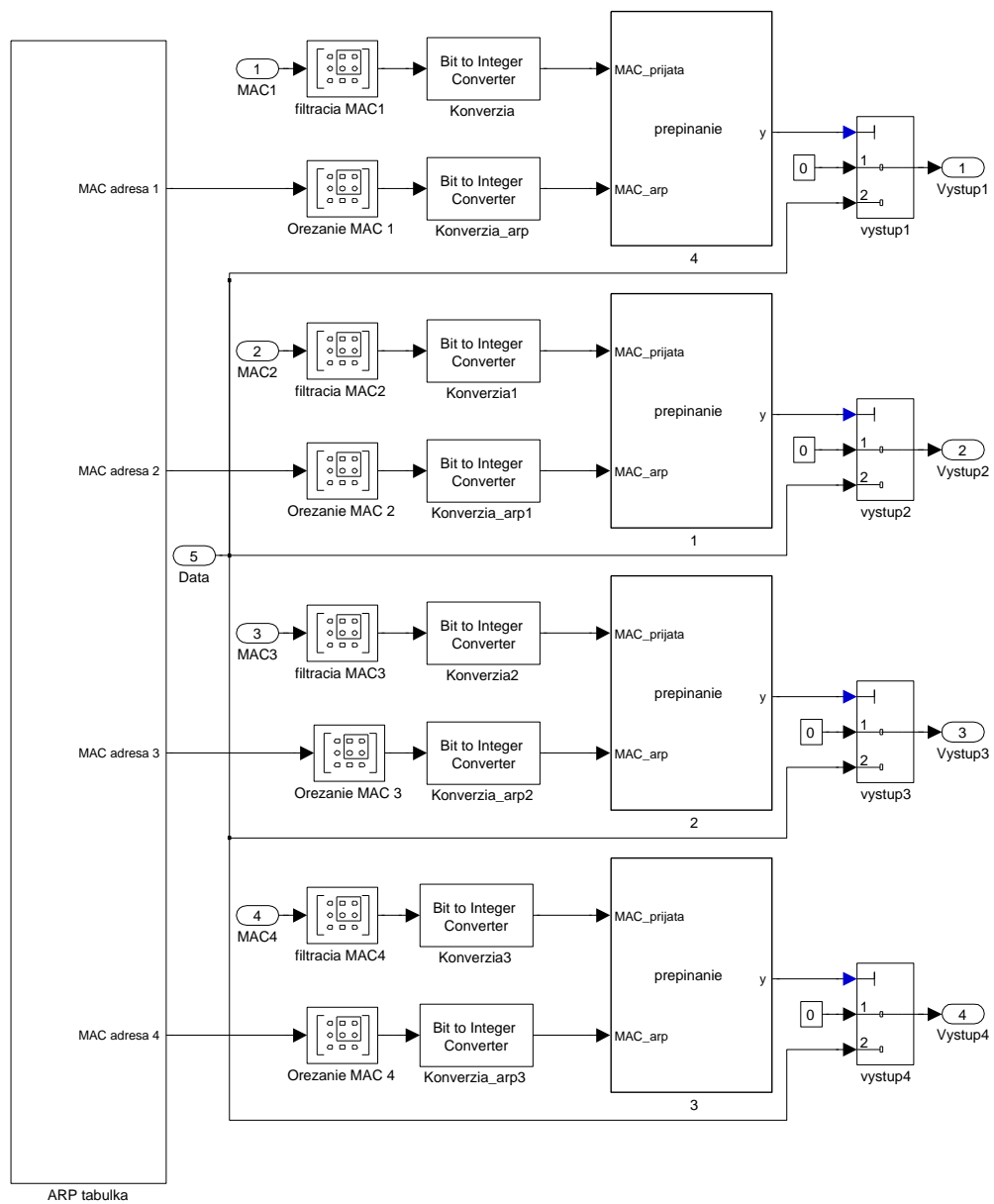
B ŠTRUKTÚRA PREPÍNAČA



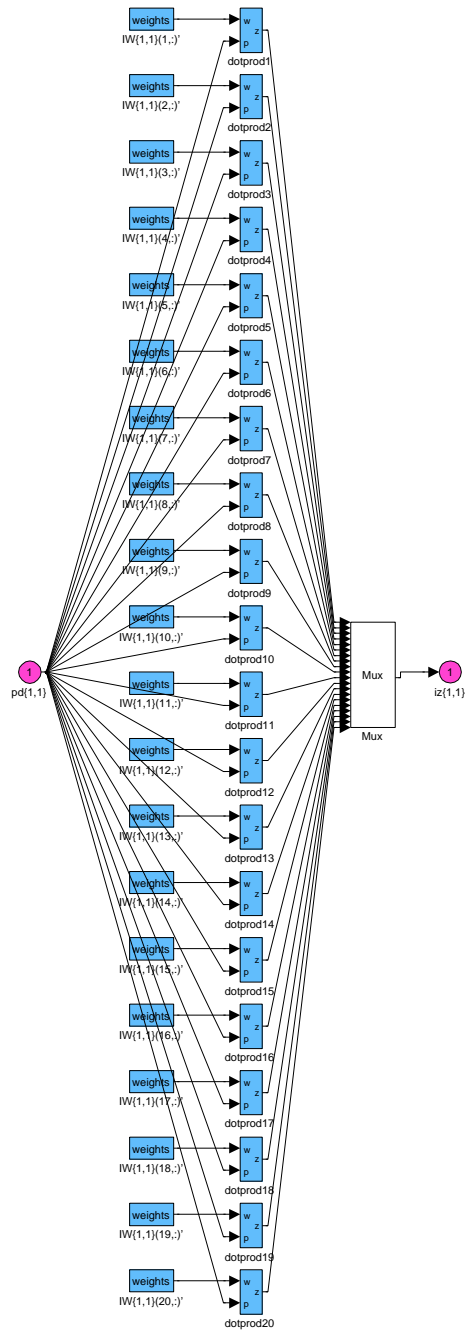
Obr. B.1: Riadenie QoS



Obr. B.2: Klasifikácia rámcov



Obr. B.3: Riadenie prepínania



Obr. B.4: Váhové hodnoty neurónovej siete