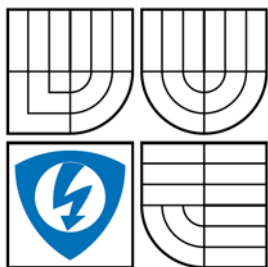


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ



FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

IMPLEMENTACE KVALITY SLUŽBY DO ŘÍZENÍ SÍŤOVÉHO PRVKU

IMPLEMENTATION OF QUALITY OF SERVICE IN THE CONTROL OF A NETWORK ELEMENT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

BC. MARTIN BOHÁČ

VEDOUCÍ PRÁCE

SUPERVISOR

DOC. ING. VLADISLAV ŠKORPIL, CSC.

BRNO 2008

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Martin Boháč
Bytem: Květná 14, Moravany 66448
Narozen/a (datum a místo): 20. 5.1984

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 602 00, Brno
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
 - ☐ diplomová práce
 - ☐ bakalářská práce
 - ☐ jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP:	Implementace kvality služby do řízení síťového prvku
Vedoucí/ školitel VŠKP:	Doc. Ing. Vladislav Škorpil, CSc.
Ústav:	Ústav telekomunikací
Datum obhajoby VŠKP:	11. 6.2008

VŠKP odevzdal autor nabyvateli v*:

- ☐ tištěné formě – počet exemplářů 2.
- ☐ elektronické formě – počet exemplářů 2.

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ☐ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 20. 5.2008.

.....
Nabyvatel

.....
Autor

ANOTACE

Cílem této práce bylo seznámit se s problematikou kvality služby v konvergovaných sítích především s užitím IP protokolu verze 6. Konvergované sítě jsou schopné efektivně přenášet různé typy dat – hlasový, datový nebo multimediální tok. Návrh aktivního síťového prvku jsem realizoval v prostředí Simulink, které je součástí programu Matlab. Navrhnutý model se skládá z jednoduché sítě několika počítačových stanic vzájemně propojených pomocí síťového prvku – přepínače. Model přepínače simuluje reálný provoz počítačových stanic, které odesílají data vzdáleným uživatelům. Pakety jsou v přepínači řazeny podle typu datového toku. S pakety je zacházeno tak, aby byla dodržena požadovaná kvalita služeb. Přednost mají datové toky, které potřebují být doručeny bez většího zpoždění, jako jsou hlasový nebo multimediální přenos. Pro řízení adresace přepínače jsem využil neuronovou síť. Neuronová síť reaguje na vstupní data a podle typu adresáta ovládá přepínač. Model přepínače lze využít v laboratorních cvičeních. K řešení této laboratorní úlohy je předpokladem základní znalost vývojového prostředí Simulink. Pracovní postup je koncipován tak, aby se úloha dala vypracovat během jednoho laboratorního cvičení.

Klíčová slova: konvergovaná síť, IP protokol verze 6, kvalita služby, neuronová síť, přepínač.

ABSTRACT

The main task of the Master Thesis is introduction into problems of quality of service in converged networks especially with use of IP protocol version 6. Converged networks are able to transfer different data types - voice, data or multimedia stream. Design of active network unit is realized in Matlab Simulink. Designed model consists of simple network with some computer terminals which are connected with network element - switch. Switch model simulates real traffic of computer terminals, that are sending data to remote users. Packets in switch are sorting by data stream type and QoS. Switch is managed by neuron network. Neuron network reacts to input data and controls switch depending on type of recipient. Switch model can be used in laboratory exercising. Solving this theme needs basic skill in Simulink and theme can be done in one laboratory exercise.

Keywords: converged network, IP protocol version 6, Quality of Service, neuron network, switch

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Implementace kvality služby do řízení síťového prvku“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 20. 5. 2008

.....

Autor

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Doc. Ing. Vladislavu Škorpilovi, CSc., za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

OBSAH:

1.	ÚVOD	10
2.	KVALITA SLUŽBY (QOS) V KONVERGOVANÝCH SÍTÍCH.....	11
2.1	KONVERGOVANÉ SÍTĚ	11
2.2	KVALITA SLUŽEB (QOS)	11
2.3	INTEGROVANÉ SLUŽBY	13
2.3.1	<i>Typy metod integrovaných služeb QoS.....</i>	<i>16</i>
2.3.2	<i>Limity pro sektor token a přenosovou rychlost</i>	<i>16</i>
2.3.3	<i>Funkce pro řízení provozu.....</i>	<i>17</i>
2.3.4	<i>Protokol RSVP.....</i>	<i>18</i>
2.4	METODY ODLIŠOVANÝCH (DIFERENCOVANÝCH) SLUŽEB QoS.....	19
2.4.1	<i>Faktory provozu QoS</i>	<i>20</i>
2.5	IP PROTOKOL VERZE 6.....	21
2.5.1	<i>Přenos hlasu přes protokol IP.....</i>	<i>22</i>
3.	ÚVOD DO NEURONOVÝCH SÍTÍ.....	25
3.1	VÝZNAM NEURONOVÝCH SÍTÍ	25
3.2	VÝVOJ NEURONOVÝCH SÍTÍ.....	25
3.3	NEURONY	26
3.3.1	<i>Biologický neuron</i>	<i>26</i>
3.3.2	<i>Umělý neuron</i>	<i>27</i>
3.4	PERCEPTRONY.....	28
3.4.1	<i>Princip činnosti perceptronu.....</i>	<i>29</i>
3.4.2	<i>Architektura dvouvrstvé perceptronové sítě.....</i>	<i>30</i>
3.4.3	<i>Algoritmus trénování perceptronové sítě</i>	<i>31</i>
3.4.4	<i>Využití perceptronové sítě</i>	<i>31</i>
4.	ANALÝZA VYUŽITÍ NEURONOVÝCH SÍTÍ.....	32
4.1	BIOLOGICKÁ NEURONOVÁ SÍŤ	33
4.2	UMĚLÁ NEURONOVÁ SÍŤ	34
4.2.1	<i>Topologie různých typů neuronových sítí.....</i>	<i>34</i>
4.2.1.1	<i>Vícevrstevný perceptron.....</i>	<i>35</i>
4.2.1.2	<i>Kohonenova samoorganizační síť</i>	<i>36</i>
4.2.1.3	<i>Neuronové sítě s přepínacími jednotkami.....</i>	<i>37</i>

4.2.1.4	Hopfieldova síť	38
4.3	ALGORITMY PRO UČENÍ NEURONOVÝCH SÍTÍ.....	41
4.3.1	<i>Back propagation</i>	42
5.	NÁVRH PŘEPÍNAČE ŘÍZENÉHO NEURONOVOU SÍTÍ.....	44
5.1	POPIS NÁVRHU	45
5.2	POČÍTAČOVÉ STANICE	46
5.3	PŘEPÍNAČ	47
5.3.1	<i>Klasifikace</i>	49
5.3.2	<i>Značkování</i>	53
5.3.3	<i>Adresace</i>	54
5.3.4	<i>Možné případy přijatých dat</i>	58
5.3.5	<i>Demonstrace funkce spínače</i>	59
6.	VYUŽITÍ V LABORATORNÍCH CVIČENÍCH	61
6.1	TEORETICKÝ ÚVOD	61
6.2	ZADÁNÍ	64
6.3	SCHÉMA ZAPOJENÍ	64
6.4	PRACOVNÍ POSTUP	65
6.5	ZÁVĚR LABORATORNÍ ÚLOHY	69
7.	ZÁVĚR.....	70

1. ÚVOD

V diplomové práci jsem se zabýval problematikou implementace kvality služby do konvergovaných sítí. Komunikační struktura konvergovaných sítí je schopna přenášet různé typy dat tak, aby byly splněny jejich požadavky. Výhody takových sítí jsou kompenzovány vysokými nároky na provoz.

V první části práce jsem obecně rozebral konvergované sítě s užitím IP protokolu verze 6, jejich vývoj a výhody oproti předcházejícímu protokolu IPv4. Dál jsem se věnoval problematice na téma: kvalita služeb, integrované služby a metody odlišovacích služeb.

V další části práce jsem popsal základní principy neuronových sítí, důvodem tohoto výkladu je použití spínacího prvku řízeného neuronovou sítí. V kapitole se zabývám významem neuronových sítí a popisem základních prvků jako je neuron nebo perceptron. V navazující kapitole jsem popsal různé typy neuronových sítí jejich topologie, algoritmy, základní vlastnosti a využití takovýchto sítí v praxi.

Návrh vlastního přepínače řízeného neuronovou sítí je popsán v následující kapitole. Tento návrh jsem provedl v prostředí Simulink, které je součástí programu Matlab. Je zde uveden popis návrhu přepínače, schéma zapojení a detailně rozvedeny jeho vlastnosti a funkce. Hlavní funkce přepínače jako je klasifikace, značkování nebo adresace, které přepínač provádí také pomocí neuronové sítě, jsou vysvětleny na obrázcích z programu Simulink.

V poslední kapitole je návrh využití přepínače v laboratorních cvičeních. Laboratorní úloha se skládá standardně z teoretického úvodu, zadání, schéma zapojení a pracovního postupu, podle kterého budou studenti pracovat. Pro řešení daných úloh je předpoklad základních znalostí programu Matlab resp. simulačního prostředí Simulink.

2. KVALITA SLUŽBY (QOS) V KONVERGOVANÝCH SÍTÍCH

2.1 Konvergované sítě

Konvergované sítě jsou spojovány v souvislosti s komunikační strukturou schopnou přenášet různé typy dat – hlas, data nebo multimédia. V současnosti dochází ke konvergenci i na úrovni služeb.

Termín „konvergence“, znamená přibližování se k sobě v oblasti telekomunikací a počítačových sítí, používá se v souvislosti s přenosovými sítěmi. Původně si tyto obě odvětví budovaly samostatné a oddělené přenosové sítě. Jedna sloužila potřebám přenosu hlasu (především pro telefonování) a druhá potřebám přenosu dat. Důvody byly historické i technické, protože přenosy hlasu i přenosy dat mají poněkud odlišné požadavky na přenosovou síť a její vlastnosti. Bylo to také neekonomické, protože budovat a provozovat dvě samostatné sítě vychází draž než u jedné společné sítě.

S růstem technologických možností, se ukázalo jako schůdné a možné vyhovět rozdílným potřebám přenosu dat i přenosu hlasu prostřednictvím jedné společné sítě. Taková síť se nazývá konvergovaná.

Konvergence služeb

Postupně dochází ke konvergenci služeb na vyšších úrovních. Objevily se takové služby, které v sobě kombinují prvky přenosu hlasu i přenosu dat. Jako například vkládání multimedií do WWW stránek – ty pak mohou obsahovat třeba různé předem připravené audio a videoklipy, či živé zvukové a obrazové přenosy. Zajímavým příkladem vzájemného prorůstání je i možnost iniciovat klasický telefonní hovor přímo z WWW stránky.

Příklady „datových“ služeb jsou přístup k Internetu nebo propojení lokálních sítí LAN v různých oblastech. U čistě hlasových služeb jde o klasické telefonování.

2.2 Kvalita služeb (QoS)

Pojem kvalita služeb (Quality of Services). Pro komunikační síť pojem QoS znamená, aby dokázala rozlišit typ přenášených dat, a aby splnila jejich požadavky na zpoždění, ztrátovost a jitter. Kvalita služeb je především spojována se sítí Internet. Internet je síť založena na protokolu TCP/IP a ten sám o sobě není schopen poskytovat kvalitu služeb. Nerozlišuje jednotlivé druhy služeb od sebe jako třeba real-time datové proudy od běžných datových přenosů a zachází s nimi zcela shodně.

Komunikační standardy jsou navrhovány tak, aby vytvářené sítě měly integrované služby. Tyto služby by měly zabezpečit správný přenos dat, hlasu nebo videa. Každá z těchto aplikací má na síť rozdílné požadavky. Datový přenos je náročný na šířku pásma a spolehlivost spojení. Hlas a video požadují relativně konstantní pásmo a garantovanou dobu doručení, částečná ztráta informace do určité míry tolerována a lze ji kompenzovat různými opravnými metodami. Následující tabulka zobrazuje rozdílné požadavky na síť [3].

	Hlas	Data	Multimédia
Zpoždění	nízká	vysoká	nízká
Kolísání zpoždění	nízké	vysoké	nízké
QoS	implicitní	minimální	přísná
Spolehlivost a dostupnost	vysoká	nízká	vysoká
Ztráty paketů	malé	velké	malé

Tab. 01. Potřebná a dostačující kvalita služby

Kvalita služeb je požadavek na síť, aby paket úspěšně překonal trasu mezi dvěma body v síti a bylo s ním zacházeno podle jeho typu. Internet pracuje na principu „Best Effort“. Po síti se posílají pakety s nevyužitými bity určujícími typ služby a prioritu. Pokud pakety poslané po síti tak se jim z hlediska vysílání a přijímání může dojít k následujícím poruchám.

- Ztracené pakety - směrovač může selhat při doručování paketu v případě, že jeho buffery jsou naplněné. Přijímající aplikace musí požádat o opětovné vyslání informace, což může představovat vážná zpoždění v celkovém přenosu.
- Zpoždění - paketu může trvat poměrně dlouhou dobu než dosáhne svého cíle vlivem čekání ve frontách, případně zvolením méně výhodně trasy, aby se zabránilo zahlcení. Na druhou stranu může jít velmi rychlou, přímou cestou, takže zpoždění je velmi proměnné.
- Rozptyl - pakety přicházejí do cíle s různým zpožděním. Tato variace ve zpoždění je označována jako rozptyl a vážně ovlivňuje kvalitu přenosu např. multimédií.
- Doručení mimo pořadí - v případě skupiny paketů proudících Internetem, každý z nich může být směrován jinou cestou, což pro ně může představovat různé zpoždění. Výsledkem toho je, že pakety mohou dorazit v jiném pořadí, než byly

vyslány. Tento problém vyžaduje existenci protokolů, které v cíli seřadí pakety podle pořadí, ve kterém byly vyslány.

- jiné chyby - např. chyba v paketu, apod.

Aplikace vyžadující QoS jako streamovaná multimédia - vyžadují garantovanou propustnost, nebo VoIP - telefonování přes IP vyžaduje přísné limity na zpoždění a rozptyl. Mezi hlavní parametry QoS patří míra propustnosti a zpoždění. Následující tabulka zobrazuje aplikace s rozdílnými požadavky na síť [3].

Aplikace	Požadavky na síť
Vysílání v reálném čase	velká šířka pásma směrem k uživateli (downstream)
Publikování z domova (vlastní webový server)	velká šířka pásma směrem od uživatele (upstream)
Interaktivní transakce, komunikace se systémy a zařízeními (monitorování)	trvalý přístup
Interaktivní hry	malé zpoždění

Tab. 02. Vztah mezi aplikacemi a požadavky na přístupovou síť

2.3 Integrované služby

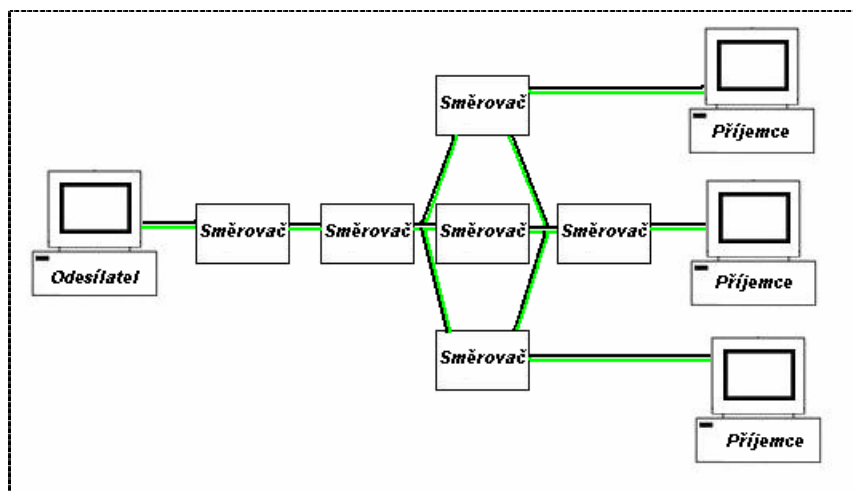
Integrované služby slouží k zajištění rezervace zdrojů pro konkrétní metodu QoS předtím, než jsou data zaslaná. Směrovače jsou ještě před samotným datovým přenosem mezi sebou propojeny a síť pak řídí průběžný přenos dat na základě metody QoS. Metoda QoS je sada pravidel, která určuje způsob přenosu. Jedná o seznam parametrů pro řízení přístupu. Žádost o rezervaci šířky pásma přichází z vysílací strany (klienta). Pokud všechny směrovače na trase přenosu s požadavkem přicházejícím od klienta souhlasí, dostane se žádost na server a je porovnána s metodou integrovaných služeb QoS. Pokud žádost odpovídá definovaných limitům pro metodu QoS, poskytne server QoS povolení pro připojení RSVP (Resource Reservation Protocol) a vyhradí pro aplikaci požadovanou šířku pásma.

Všechny síťové uzly, kterými přenos prochází, musí mít schopnost používat protokol RSVP. Směrovače realizují QoS pomocí následujících funkcí pro řízení provozu: plánování paketů, klasifikace paketů a řízení přístupu.

Nejdůležitější částí implementace metod integrovaných služeb QoS je schopnost řídit a předvídat zdroje v síti.

Příklad provozu v síti – může být směrován pouze na základě zdrojů. Pokud přenos prochází přes směrovače, které neumožňují RSVP, může dojít k problémům. Spojení se provede, ale výkon, který aplikace požaduje, tento směrovač nezaručí.

Na následujícím obrázku je znázorněno, jak proces integrovaných služeb QoS funguje.



Obr. 01. Trasa RSVP mezi odesílatelem (serverem) a příjemcem (klientem)

Aplikace používající RSVP na serveru zaznamená žádost o připojení klienta. Serverová aplikace vydá klientovi příkaz PATH. Příkaz obsahuje informaci o IP adrese směrovače. Příkaz obsahuje další informace o dostupných zdrojích na serveru a směrovačích na trase přenosu, a také informace o trase mezi serverem a klientem. Aplikace umožňující RSVP pošle zpátky po síťové trase příkaz RESV, tzn. že síťové zdroje byly přiděleny. Tento příkaz provádí vlastní rezervaci na základě směrovacích informací v příkazu PATH. Server a všechny směrovače na cestě přenosu si rezervují zdroje pro spojení RSVP. Když server obdrží příkaz RESV, aplikace spustí přenos dat klientovi. Data se přenášejí po stejné trase, jakou probíhala rezervace.

Služby integrované nejsou určeny pro krátkodobá spojení RSVP typu HTTP. Předpoklad pro aplikace používané v metodě integrovaných služeb QoS je, že musí být schopny používat protokol RSVP. Pokud server nemá aplikace umožňující RSVP, je potřeba napsat vlastní aplikaci, která RSVP umožňuje.

Pokud přicházejí pakety na server a pokoušejí se opustit síť, server určuje, zda má k dispozici zdroje potřebné k zaslání paketu. Potvrzení nebo odmítnutí závisí na množství prostoru v sektoru token. Pokud příchozí pakety způsobí, že sektor token přesáhne danou mezní

hodnotu, považují se pakety za nevyhovující. Tyto pakety může server zpracovat několika různými způsoby. Může pakety buď odložit, tvarovat, znovu přenést nebo uvolnit. Pokud jsou pakety vyhovující, server je posílá dál. Při integrovaných službách QoS se každému spojení poskytuje vlastní sektor token. U odlišovaných služeb QoS sdílí celá podsíť nebo skupina klientů jeden sektor token.

2.3.1 Typy metod integrovaných služeb QoS

Existují dva typy integrovaných služeb: služby řízeného zavádění a garantované služby.

Služby řízeného zavádění

Tyto služby podporují aplikace, citlivé na zahlcení sítě, např. aplikace v reálném čase. Tyto aplikace jsou odolné vůči malým ztrátám nebo zpožděním. Pokud aplikace využívá služby řízeného zavádění, její výkonnost se při zvýšení zátěže sítě nebude zhoršovat. Provoz bude zajištěn pomocí služby, udržující normální provoz v síti za omezenějších podmínek.

Úkolem směrovače je zajištění, aby služba řízeného zavádění obdržela definovanou šířku pásma a zdroje na zpracování paketů. Tyto směrovače musejí mít podporu pro integrované služby QoS. Funkce pro řízení provozu se skládají z těchto komponent: plánování paketů, klasifikace paketů a řízení přístupu.

Garantované služby

Garantované služby slouží k tomu, aby pakety byly doručeny v rámci stanovené doby. Aplikace jako audio a video vysílací systémy (streaming), potřebují garantované služby. Tyto služby kontrolují zpoždění ve frontě a zajišťují, aby se pakety nezpозdily nad stanovený časový limit. Všechny směrovače na trase přenosu musí podporovat RSVP, aby zajistil dodání paketu v souladu s touto metodou QoS. Garantované služby se definují stanovením mezi velikosti sektoru token a limity přenosové rychlosti.

2.3.2 Limity pro sektor token a přenosovou rychlost

Tyto limity výkonu zaručují garantované dodání paketů v rámci metod řízení šířky pásma u odchozích spojů, a to jak u metod integrovaných služeb QoS tak u metod odlišovaných služeb QoS.

Velikost sektoru token

Tato velikost určuje celkovou kapacitu vyrovnávací paměti, která obsahuje shluky dat. Shluk dat jsou informace, které aplikace předává serveru, aby ji odeslal vyšší rychlostí, než může server zpracovat. Pokud aplikace rychle zasílá shluky dat na server, vyrovnávací paměť se naplňuje. Naopak pokud aplikace posílá informace pomaleji, než jak ze serveru mohou odcházet, vyrovnávací paměť se vyprazdňuje. Když data ze serveru odcházejí stejně rychle jako do serveru vstupují, pak se velikost sektoru token nemění. Ve chvíli kdy je vyrovnávací paměť plná, metoda QoS s dalšími datovými pakety nakládá jako s pakety mimo profil. U metody QoS je možné nastavit, jak bude QoS zacházet s přenosy mimo profil.

Limity přenosové rychlosti

Limity přenosové rychlosti (šířky pásma) udávají průměrnou rychlost datových přenosů neboli počet bitů za sekundu, který může vstoupit do sítě. Při žádosti klienta od serveru o RSVP, žádá klient o konkrétní množství šířky pásma. Metoda QoS vyhodnotí požadovanou šířku pásma a porovná ji s limity pro přenosovou rychlost, které má definovány. Pokud by server překročil tyto limity, žádost odmítne. Limity přenosové rychlosti jsou v rámci metod integrovaných služeb QoS používány pouze při řízení přístupu. Měří se v Kbit/s. Hodnota tohoto parametru se může pohybovat od 10 Kbit/s do 1Gbit/s.

Limit průměrné přenosové rychlosti či šířky pásma musí být menší než limit maximální přenosové rychlosti či šířky pásma, tak abyste nevyčerpali celé rozhraní.

K nastavení limitů se používá funkce monitorování QoS. Metoda QoS se vytvoří s dostatečně velkým limitem přenosové rychlosti, aby pokryl datový provoz v síti. Spuštěním sběru dat pro tuto metodu QoS začne monitorování aktuálního stavu sítě. Sledováním dat získáme potřebné informace o přenosových rychlostech, které aplikace a síť používá. Na základě výsledků monitorování nastavíme limity přenosové rychlosti.

2.3.3 Funkce pro řízení provozu

Funkce pro řízení provozu souvisejí pouze metodou integrovaných služeb QoS. Pro zajištění požadovaných výsledků, musí být na trase přenosu hardware, který umožňuje RSVP. Směrovače musí mít určité funkce pro řízení provozu, aby mohly používat protokol RSVP. Server může fungovat pouze jako server nebo jako klient.

Funkce pro řízení provozu zahrnují tyto funkce.

- Plánování paketů - funkce řídí posílání paketů na základě informace v IP hlavičce. Tato funkce zajišťuje paketu, že bude doručen s parametry, které jsou nastaveny v metodě QoS. Aplikuje se v místě, kde se pakety řadí ve frontě.
- Klasifikace paketů - funkce identifikuje na základě informace v IP hlavičce pakety, které v rámci toku IP obdrží určitou úroveň služeb, a to opět. Každý příchozí paket je touto funkcí zmapován a zařazen do určité třídy. Pakety zařazené do stejné třídy jsou zpracovány stejným způsobem.
- Řízení přístupu - funkce řízení přístupu obsahuje algoritmus, pomocí kterého směrovač vyhodnocuje, zda má dost směrovacích zdrojů. Pokud jich nemá dostatek, je nový datový tok odmítnut. Jestliže je tok přijatý, směrovač pro daný paket zaktivuje funkce plánování a klasifikace, aby rezervoval požadovanou úroveň QoS.

2.3.4 Protokol RSVP

Protokol RSVP (Resource Reservation Protocol) rezervuje šířku pásma pro integrované služby QoS. Tuto rezervaci provádí spolu s rozhraními RAPI API nebo qtoq QoS sockets API. Všechny síťové uzly podél trasy přenosu podporovat protokol RSVP.

Protokolu RSVP se provádí rezervaci ve všech síťových uzlech na trase přenosu. Protokol udržuje rezervaci tak dlouho, aby poskytl metodě QoS požadovanou úroveň služeb. Rezervace definuje způsob zacházení s daty a šířku pásma, kterou dostanou přenášovaná data.

Rezervace se provádí pouze v jednom směru (od příjemce). U náročnějších spojení, jako jsou audio nebo video konference, je každý odesílatel zároveň příjemcem. Zde je nutno nastavit RSVP na obou stranách spojení.

Protokol RSVP je primárně využíván k získání kvality služeb pro aplikační datové proudy nebo toky. RSVP je také užíván směrovači k doručení QoS žádostí všem uzlům podél cesty datových toků. Rezervaci zdrojů je nutno realizovat v každém uzlu pro daný datový proud.

Multimediální aplikace vyžadují spolehlivé doručení dat a zároveň nemají přísné požadavky na včasnost doručení. Aplikace jako videokonference, IP telefonie, internetové rádio naopak vyžadují včasné doručení dat, které nemusí být nutně spolehlivé. Protokol RSVP má za úkol poskytnout IP sítím schopnost podporovat odlišné výkonové požadavky různým aplikacím.

Datový tok je u protokolu RSVP chápán jen jako sekvence datagramů, která má stejný zdroj, cíl a kvalitu služeb. U architektury založené na tomto protokolu existují dva základní prvky, zdroj a cíl. Na těchto prvcích běží procesy, které tvoří protokol RSVP a dochází mezi nimi k výměně RSVP zprávy. Vyměňují si dva druhy zpráv, PATH a RESV. Zprávu PATH vysílá zdroj. Pokud je tato zpráva doručena ke správnému příjemci, pak odpovídá zprávou RESV. RESV je zpět odeslána odesílateli stejnou cestou. V případě, že zpráva nedoručí ke správnému příjemci, pak je vygenerována a odeslána zpráva RESV ERROR. Úspěšná rezervace cesty se vykoná až v případě, kdy RESV zpráva dojde odesílateli. RSVP protokol slouží k zaručení kvality služeb pro daný případ použití.

2.4 Metody odlišovaných (diferencovaných) služeb QoS

Pomocí odlišovaných služeb se dělí provoz na síti do tříd. Při implementaci QoS je potřebné stanovit, jak klasifikovat síťový provoz a jak zacházet s různými provozními třídami.

Pomocí bitů v IP hlavičce server identifikuje úroveň služeb pro daný IP paket. Směrovače přidělují své prostředky na základě informace obsažené v poli TOS (Type of Service) v IP hlavičce. Kód určí způsob, jakým je paket v jednotlivých síťových uzlech přeposílán. K tomu, aby určitý paket obdržel požadovanou úroveň služeb, musí každý uzel sítě podporovat odlišované služby. Zařízení musí být schopno provádět zpracování kódu. Základ pro zpracování kódu je to, že uzel sítě musí umět používat plánování front (Queue Scheduling) a správu výstupních priorit (Outbound Priority Management), což znamená, že zařízení podporuje odlišované služby.

Pokud paket prochází přes směrovač, který nepodporuje odlišované služby QoS, ztratí paket svoji úroveň služeb. Paket bude zpracován, ale může dojít k neočekávanému způsobu doručení. Na serveru můžete použít buď standardní kódové pozice, nebo si můžete vytvořit své vlastní třídy.

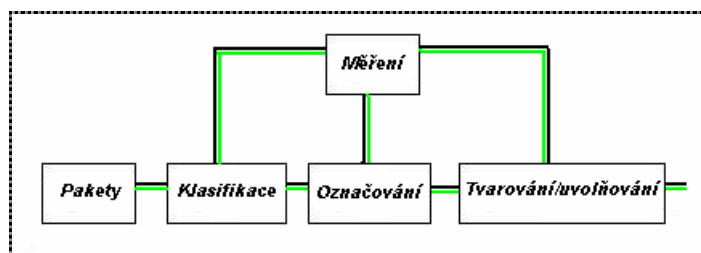
Rozdíl v přenosu od integrovaných služeb spočívá v tom, že nevyžadují rezervaci nebo manipulaci s jednotlivými toky dat. Se všemi přenosy zařazenými do určité třídy se zachází stejně.

Požítí metody odlišovaných služeb je využíváno k řízení přenosů do serveru a ze serveru. Server může používat odlišované služby k omezení výkonu. Omezení méně důležitých aplikací umožňuje, přednostní odchod důležitých aplikací ze serveru. Při vytváření metody QoS, je nutné nastavit různé mezní hodnoty pro server. Mezi tyto limity výkonu patří velikost sektoru token (Token Bucket Size), maximální přenosová rychlost (Peak Rate Limit) a průměrná přenosová rychlost (Average Rate Limit).

2.4.1 Faktory provozu QoS

Síťové zařízení využívající metody QoS, musí mít určité schopnosti, které QoS podporují. Směrovače nebo přepínače musí mít schopnosti týkající se práce s pakety, jako jsou klasifikace, měření, označování, tvarování a uvolňování. Tyto schopnosti se nazývají faktory provozu QoS. Aby se síťové zařízení mohlo označit jako podporující QoS, musí splňovat všechny faktory provozu.

Na následujícím obrázku jsou znázorněny jednotlivé faktory provozu QoS.



Obr. 02. Faktory provozu QoS

Jednotlivé faktory provozu QoS.

- Klasifikace - klasifikace provozu znamená, že se pakety v datovém toku vybírají na základě obsahu v jejich IP hlavičkách. Pakety se třídí na základě kódové pozice odlišovaných služeb (DSCP). Klasifikaci MF (Multi-field) - třídí pakety na základě hodnoty kombinace jednoho nebo více polí hlavičky.
- Měření - při měření provozu se zjišťuje, zda IP pakety poslané dál po klasifikaci odpovídají profilu IP hlavičky přenosu. Informace v IP hlavičce je určena hodnotami nastavenými v metodě QoS. Měření se provádí pro každý paket.
- Označování - při označování paketů se nastavuje pole odlišovaných služeb. Nastavení kódové pozice odlišovaných služeb a převede se na bajty. Označování lze nakonfigurovat tak, že se všechny pakety označují jednou kódovou pozicí nebo sadou kódových pozic používaných pro volbu chování při jednotlivých přechodech.
- Tvarování - při tvarování dochází ke zpoždění některých paketů v toku přenosu. Tok musí probíhat v souladu s profilem přenosu. Tvarování využívá vyrovnávací paměť, a pokud je paměť plná, pakety mohou být vyřazeny.
- Uvolňování - při uvolňování paketů dochází k vyřazení některých paketů z toku přenosu. Dochází k tomu proto, aby se tok přenosu uvedl do souladu s profilem přenosu.

2.5 IP protokol verze 6

Protokol Internetu verze 6 (IPv6) je aktualizovanou verzí protokolu Internetu verze 4 (IPv4) a bude postupně nahrazovat IPv4 jako standard sítě Internet. IPv6 je pokračováním vývoje protokolu Internetu. Ve větší části sítě Internet se v současné době používá IPv4. Tento protokol je spolehlivý a odolný více než 20 let. IPv4 však má závažná omezení, která budou s rozvojem Internetu způsobovat další problémy.

Prohlubuje nedostatek adres IPv4 potřebných pro všechna nová zařízení připojovaná k Internetu. Podstatou zdokonalení IPv6 je rozšíření prostoru IP adres z 32 bitů na 128 bitů, které umožňuje fakticky neomezený počet jedinečných IP adres. Nový formát textu adres IPv6 je.

XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

Obr. 03. Formát textu adres IPv6

Každé *x* představuje hexadecimální číslici reprezentující 4 bity. Rozšířená schopnost adresování IPv6 je řešením problému vyčerpání adres. V současnosti se stále víc používají přenosné počítače a mobilní telefony. Rostoucí požadavky uživatelů s bezdrátovým připojením přispívají k vyčerpání adres IPv4. Rozšíření prostoru IP adres v IPv6 řeší tento problém tím, že bude k dispozici dostatečný počet IP adres i pro rostoucí počet zařízení.

Adresování IPv6 poskytuje nové funkce, které zjednodušují konfigurování a správu adres v síti. Konfigurace a údržba sítě IPv4 je poměrně náročná. IPv6 snižuje pracovní zátěž automatizací některých úkolů správce sítě.

U IPv6 při přechodu k jinému poskytovateli služeb sítě Internet (ISP) zaniká povinnost přechíslovat adresy zařízení. Stávající adresy můžeme zachovat, protože jsou globálně jedinečné.

Funkce automatické konfigurace zajišťuje automatické nakonfigurování adres rozhraní a směrovačů. Při bezstavové automatické konfiguraci vytvoří IPv6 z adresy MAC počítače a z prefixu sítě poskytnutého lokálním uzlem novou jedinečnou adresu IPv6. Tato funkce odstraňuje potřebu serveru DHCP.

Funkce IPv6

IPv6 je v současné době implementován v platformě pro vývoj aplikací, která slouží k vývoji a testování aplikací IPv6. Funkce IPv6 jsou pro stávající aplikace TCP/IP transparentní a existují společně s funkcemi IPv4.

Hlavní funkce serveru ovlivněné IPv6 jsou tyto.

- Konfigurace - proces konfigurace se u IPv6 liší od procesu konfigurace u IPv4. Při použití funkce IPv6, musíme změnit konfiguraci TCP/IP serveru tím, že nakonfigurujeme linku pro IPv6. IPv6 můžeme nakonfigurovat na lince Ethernet nebo na tunelové lince. Pokud pro provoz IPv6 nakonfigurujeme linku Ethernet, budou pakety IPv6 posílány sítí IPv6.
- Sokety - k vývoji a testování aplikací typu soket slouží rozhraní API a nástroje IPv6. IPv6 vylepšuje sokety tak, že aplikace mohou používat IPv6 s využitím nové skupiny adres. Tato vylepšení neovlivňují stávající aplikace IPv4. Můžete vytvářet aplikace, které podporují souběžný provoz IPv4 a IPv6, nebo pouze provoz IPv6.
- DNS - (Domain Name System) podporuje adresy typu AAAA a novou doménu pro zpětná vyhledávání: IP6.ARPA. Přestože DNS dokáže číst informace IPv6, server musí ke komunikaci s DNS používat IPv4.
- Odstraňování problémů s TCP/IP - pro sítě a tunely IPv6 můžeme používat standardní nástroje pro odstraňování problémů, například příkazy PING a NETSTAT, trasování přenosové cesty a trasování komunikace. Tyto nástroje nyní podporují formát adres IPv6.

2.5.1 Přenos hlasu přes protokol IP

Přenášet digitalizovaný lidský hlas po sítích na bázi protokolu IP teoreticky nebyl možný, protože architektura byla navržena na principu přepojování paketů a nedokáže garantovat jejich včasnost doručení. Není tedy vhodný pro potřeby IP telefonie, a už vůbec ne pro potřeby internetové telefonie. Přenos hlasu je umožněn díky těmto faktorům.

- Zvyšuje se celková propustnost přenosových cest, neboli kapacita Internetu.
- Protokoly snižují nároky na objem přenesených dat, a také snižují citlivost na případné nerovnoměrnosti v čase doručování.

Tyto faktory nemění základní princip, ale snižují pravděpodobnost, že se všechna data nestihnou přenést včas. Přenášení dat v Internetu se časem zlepšuje a díky tomu je možné

přenášet živý hlas. Internetová telefonie byla testována již kolem roku 1995. Další vývoj se ubíral směrem ke zlepšování přenosových vlastností Internetu, a tím byla kvalita internetové telefonie reálnější.

Příčina problémů s hlasovými přenosy v IP sítích je v tom, že síť nikoho neupřednostňuje a chová se vůči všem přenášeným datům stejně. Existují takové datové pakety, jako například elektronická pošta, kterým časové zpoždění nevadí. Vedle nich pak existují i takové datové pakety, které jsou na případná zpoždění nesmírně citlivé. IP protokol je zkonstruován tak, že všechny pakety se snaží doručit v minimální možné době, a pokud se mu to nedaří, zpoždění je u všech stejné. Protokolu IP chybí schopnost poskytnout některým paketům své služby v jiné kvalitě než ostatním. Nejlépe takovým způsobem, aby jejich kvalita mohla být garantována tomu, kdo je potřebuje.

V současnosti se oblast přenos hlasu po IP síti (VoIP - Voice over IP), resp. kompletní řešení telefonie přes IP (Telephony over IP) mohutně rozšiřuje do mnoha firem. Jedná se o oblast spojenou s možností ekonomických úspor a lepších služeb při telefonování.

Princip fungování VoIP

Princip fungování je vcelku jednoduchý. Při telefonování se analogový hlas převede do datové podoby a rozdělí se do IP paketů, ty se dál šíří jako klasický IP provoz. Směrem k příjemci je opět podobná konverze, kde se IP pakety rozbalí, dojde k převodu dat na analogovou podobu. Možnosti zapojení jsou různé. Před ústřednu, místo ústředny a nebo za ústřednu. Záleží na použití IP telefonů nebo typu připojení k ústředně (data/hlas).

Přenos probíhá v následujícím pořadí - volající musí vědět, které telefonní číslo má jakou IP adresu, případně na jakou IP adresu hovor předat dál. Je nutné provést signalizaci, tedy zazvonit u volaného, případně volajícímu nějak oznámit nedostupnost. Je důležité tento hovor zaúčtovat, zejména pokud jde do jiné telefonní sítě.

- SoftSwitch - nezávislá část na použitém protokolu. Řídí hovory, provádí směrování provozu, poskytuje autentizační služby.
- Gatekeeper - prvek zodpovědný za správu uživatelů (klientů) a bran ve své oblasti. Řešeno jako software na serveru nebo jako součást jiných zařízení (např. bran).
- Gateway - signalizační brána nebo brána mezi médii. Provádí konverzi mezi různými síťovými protokoly nebo technologiemi.

Sít' pro přenos VoIP

IP telefonie potažmo technologie VoIP nepotřebuje ke své činnosti nutně Internet. Vystačí libovolná IP síť např. vnitrofiremní. Každá síť není pro přenos hlasu vhodná. Je nutno, aby byla k dispozici dostatečná šířka pásma. Dalším důležitým požadavkem je i dostatečně malá odezva sítě. Je také vhodné, aby zpoždění na síti bylo konstantní. Pokud není některý z těchto požadavků splněn, dochází k různým narušením kvality přenášeného hlasu.

- Ztráta paketů - se projeví výpadkem hovoru. Může být způsobeno nedostatečnou šířkou pásma nebo nekvalitní linkou.
- Příliš velká odezva - dochází ke zpoždění v komunikaci. Doporučená doba zpoždění musí být méně než 150 ms, aby lidské ucho nepoznalo rozdíl proti analogovým telefonům.
- Nekonstantní doba zpoždění - dochází ke zpoždění a k opakování hlasu - došlé pakety nedorazí ve správném pořadí, kodeky často počítají s konstantním přítokem dat. Tento jev se označuje jako „jitter“ a dá se mu předcházet dostatečnou velikostí vyrovnávací paměti na straně příjemce.

V současnosti je možné provést telefonní hovor i přes mobilní GRPS spojení (General Packet Radio Service), což je ovšem případ krajní nouze a obě zúčastněné strany se musí smířit s nedostačující kvalitou hlasu, velkými prodlevami mezi doručením hlasu a případně i ztrátami některých částí hovoru.

3. ÚVOD DO NEURONOVÝCH SÍTÍ

3.1 Význam neuronových sítí

Umělé neuronové sítě vznikly na podkladě výzkumu mozku, kde se pomocí umělých neuronových sítí modelovaly procesy, které probíhají v mozku. Neuronové sítě rozšířily do mnoha oborů technické praxe. Základním prvkem mozku je neuron, do něhož vstupuje velký počet vstupů – tzv. dendritů a vystupuje pouze jeden výstup – axon. Ten se ovšem může dále rozvětňovat do tzv. terminálů, které se připojují na dendrity pomocí synapsí. Na základě tohoto biologického popisu byl definován umělý neuron, který tvoří výkonný prvek umělé neuronové sítě.

Neuronové počítače jsou velmi zjednodušeně matematické modely neuronových systémů živých organismů. Současné aplikace umělých neuronových sítí využívají selektivní a generalizační vlastnosti. Složitější struktury jsou schopné řešit i úlohy typu jako jsou optimalizační úlohy.

3.2 Vývoj neuronových sítí

Počátek prací na neuronových sítích se datuje k roku 1943, kdy s velmi jednoduchým modelem neuronu přišli pánové Warren McCulloch a Walter Pittse.

Frank Rosenblatt zobecnil model neuronu v roce 1957 na tzv. perceptron, který počítal s reálnými čísly. Jedná se o pevnou architekturu jednovrstvé sítě s „ n ” vstupními a „ m ” výstupními neurony. Zároveň navrhl učicí algoritmus, který v konečném čase nalezne odpovídající váhový vektor parametrů nezávisle na počáteční konfiguraci. Na základě tohoto výzkumu spolu s Charlesem Wightmanem sestrojil během let 1957 a 1958 první neuropočítač.

V 80. letech John Hopfield ukázal souvislost některých modelů s fyzikálními modely magnetických materiálů. Později podle něj byly nazvány tzv. Hopfieldovy sítě, a ty fungují na principu autoasociativní paměti. Síť má pevnou topologii s „ n ” neurony, které jsou zapojeny cyklicky. Všechny neurony jsou zároveň výstupní.

V roce 1986 badatelé sdružení do PDP skupiny s zástupci Davidem Rumelhartem, Geoffreyem Hintonem a Ronaldem Williamsem publikovali učicí algoritmus zpětného šíření chyby tzv. backpropagation pro vícevrstvou neuronovou síť.

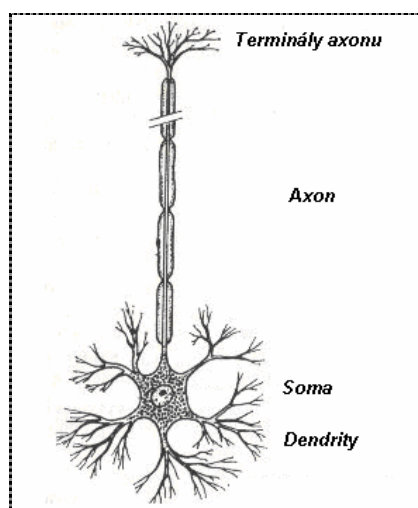
V dnešní době se především výpočty simulují na klasických PC stanicích. Některé nadějné neuronové sítě jsou zaváděny do praxe, ať již v podobě automatického řidiče nebo náhrada u měřicích přístrojů.

3.3 Neurony

3.3.1 Biologický neuron

Neuron je základní jednotkou nervové soustavy, je to nervová buňka se svými výběžky. Hlavní funkcí neuronu je tvorba a přenos nervových signálů, jejichž fyzikální podstatou je pohyb iontů, který lze registrovat jako elektrické děje. Integrací nervových signálů vznikají odpovědi nervové soustavy, které představují povely k činnosti jednotlivých orgánů.

Obrázek převzat z elektronických skript pro předmět Umělá inteligence.



Obr. 04. Biologický neuron

Biologický neuron se skládá z těchto částí.

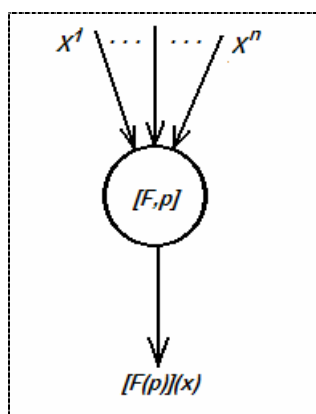
- Soma – buněčné tělo neuronu.
- Dendrity – buněčné výběžky vážou se k nim axony jiných neuronů.
- Axon – výběžek, kterým se šíří signál k ostatním neuronům na konci se bohatě větví.
- Terminál axonu – zakončení axonu uvolňuje neurotransmitery - šíření signálu chemickou cestou.

Podle tohoto modelu sbírá typický neuron signály od ostatních neuronů biologického mozku pomocí velkého množství jemných vláken zvaných dendrity. Neuron vysílá elektrické impulzy do okolí prostřednictvím dlouhého tenkého vlákna zvaného axon, který se rozbíhá do

tisíců větví, tzv. terminálů. Na konci každé větve se nachází spoj zvaný synapse, který přemění aktivitu axonu na elektrický impulz, a který buď vyvolá nebo utlumí aktivitu v přilehlém neuronu. Jestliže neuron dostane dostatečně velký aktivační elektrický impulz ve srovnání s tlumícím, aktivuje svůj axon. Proces učení se děje pomocí změn existujících synapsí, čímž se mění vliv neuronu na ostatní. V lidském mozku má jeden neuron v průměru spojení s 10 až 100 tisíci jinými neurony.

3.3.2 Umělý neuron

Umělý neuron vznikl na základě předlohy biologického neuronu.



Obr. 05. Umělý neuron

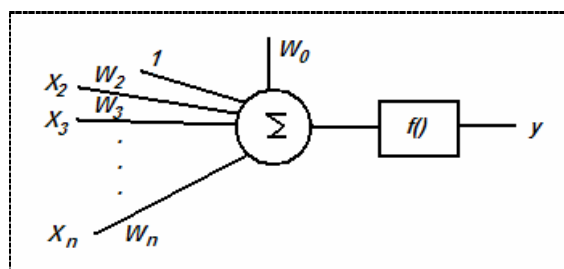
Umělý neuron se skládá z těchto částí.

- x^1 až x^n – jsou výstupy z předchozích neuronů.
- n – počet vstupů.
- $[F,p]$ – tělo neuronu, zde jsou vstupy sečteny a vynásobeny zadanou funkcí.
- $[F(p)](x)$ – požadovaný výstup neuronu.

Princip umělého neuronu je inspirován činností podle biologické předlohy. Výstupy z neuronů (resp. vstupy), na které je připojen, jsou vynásobeny váhami, které určují vztah spojení mezi neurony. Vážené vstupy jsou sečteny, je od nich odečten práh a výsledný signál je zpracován přechodovou funkcí. Výstup takové funkce představuje výstupní signál neuronu. Samotný umělý neuron je v podstatě nejjednodušší umělá neuronová síť tzv. perceptron. Přechodová funkce perceptronu je funkce signum, výstup neuronu je tedy 1 nebo -1 podle znaménka jeho potenciálu. Tento typ neuronu tedy každý vstup (n -rozměrný vektor, kde n je počet vah neuronu) klasifikuje do jedné z tříd 1 nebo -1.

3.4 Perceptrony

V této kapitole jsou uvedeny obecné vzorce, které jsem převzal z literatury [9].



Obr. 06. Model perceptronu

Model perceptronu se skládá z těchto částí.

- x_1-x_n – jsou vstupy perceptronu.
- n – je počet vstupů.
- w_i – je váha spoje.
- w_0 – je prahová hodnota.
- f – je aktivační funkce.
- y – je požadovaný výstup perceptronu.

Z předkládaných vzorových dat je perceptron schopen se naučit jejich klasifikaci. Omezující podmínkou je lineární separabilita množin, které bude perceptron klasifikovat do rozdílných tříd. Perceptron hledá dělící nadrovinu mezi předkládanými vektory. Rovina vektory rozděluje podle požadované klasifikace. Používá tzv. delta-pravidlo. Pokud taková rovina neexistuje, pokud se například shluky vektorů v n -rozměrném prostoru prolínají, není perceptron schopen korektně se naučit a klasifikovat všechny předložené vstupy. Různými modifikacemi algoritmu učení a nastavením vah a prahu neuronu se minimalizuje počet chybných klasifikací.

Využití perceptronu při rozpoznávání znaků je tedy značně omezené. Požadavek na lineární separabilitu klasifikovaných množin činí perceptrony náchylnými na transformace znaků nebo šum v datech.

Perceptron je tedy základní jednotka perceptronových sítí. Aktivační funkce perceptronu je omezená funkce, její výstupní hodnoty nabývají hodnoty 0 nebo 1 a to v závislosti na vstupních hodnotách.

3.4.1 *Princip činnosti perceptronu*

Všechny vstupy v perceptronu jsou ohodnoceny příslušnou vahou, ta souvisí s daným spojením. Sumarizace vážených vstupů je předána jako vstupní informace pro aktivační funkci, jejímž výstupem je výstup perceptronu. Jeho vztah je definován takto [9].

$$y = f\left(\sum_{i=1}^n w_{1,i}x_i + w_0\right). \quad (1)$$

V závislosti na hodnotách vstupu jsou výstupní hodnoty omezeny v rozsahu 0 nebo 1. Pokud je prahová hodnota 0 znamená to, že pro všechny záporné vstupní hodnoty bude na výstupu neuronu 0. Naopak pro všechny nezáporné hodnoty bude na výstupu neuronu hodnota 1. Hodnota prahu způsobuje posun vstupní hodnoty, kdy dochází ke změně výstupní hodnoty z 0 na 1.

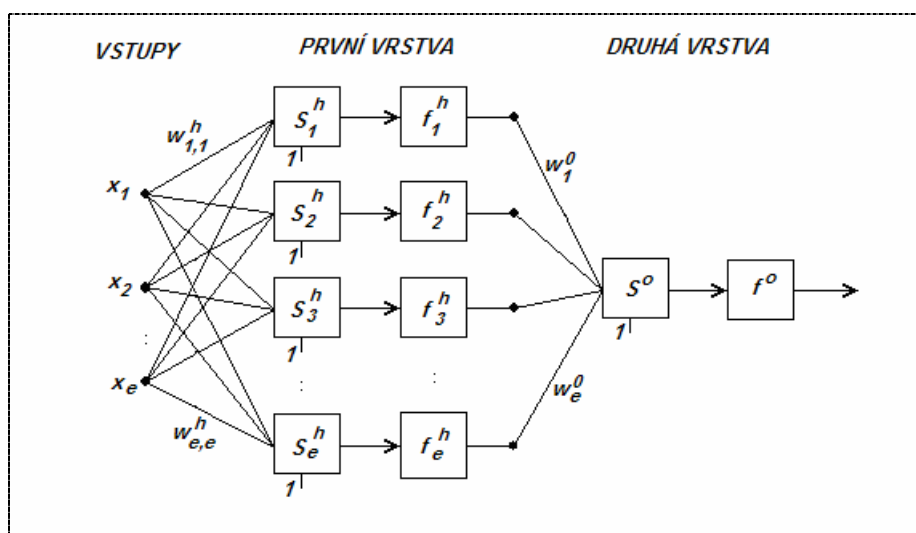
3.4.2 Architektura dvouvrstvé perceptronové sítě

Vícevrstvé perceptronové sítě se skládají z několika neuronových jednotek podobných jako perceptron. Obecně se tyto sítě označují jako dopředné vícevrstvé sítě.

Dvouvrstvá perceptronová síť, která je dána matematickým vztahem [9].

$$y_{NN} = f^o \left(\sum_{i=1}^n w_i^o f_i^h \left(\sum_{j=1}^n w_{ij}^h x_j + w_{0i}^h \right) + w_0^o \right), \quad (2)$$

kde váhy jsou spojení mezi vstupní a skrytou vrstvou nebo spojení mezi skrytou a výstupní vrstvou. Prahové hodnoty mezi odpovídajícími neuronovými vrstvami jsou aktivační funkce pro skrytou a výstupní vrstvu. Hodnoty označují součty vstupů ohodnocené váhovými a prahovými hodnotami.



Obr. 07. Architektura dvouvrstvé perceptronové sítě

Takovéto vícevrstvé sítě často využívají ve skrytých vrstvách logsigmoidální aktivační funkci, ta generuje výstupy mezi 0 a 1. Vstupní hodnoty neuronové sítě nabývají hodnot od záporného do kladného nekonečna. Pokud výstupy mají být v rozsahu -1 až $+1$ používá se tansigmoidální aktivační funkce. Ve výstupní vrstvě se zpravidla používá lineární aktivační funkce, která upravuje výstup neuronové sítě na požadovaný rozsah hodnot v pracovní oblasti.

3.4.3 Algoritmus trénování perceptronové sítě

Při trénování perceptronových sítí se používají algoritmy zpětného šíření (Back-propagation) nebo jejich alternativní typy. Důležité je znát využití perceptronové sítě. Trénování algoritmus Back-propagation je nejvhodnější pro tzv. statické neuronové sítě. Při modelování dynamických systémů se používají dynamické neuronové sítě, u kterých se adaptují jejich parametry v průběhu učení. Důležitá je volba rekurzivních algoritmů pro trénování sítě. Nejčastěji využívaným algoritmem pro trénování dynamických neuronových sítí je rekurzivní algoritmus nejmenších čtverců (RLS algoritmus).

3.4.4 Využití perceptronové sítě

Hojně používanou architekturou je síť se dvěma vrstvami (jednou vstupní a jednou skrytou vrstvou). U adaptivního trénování neuronových sítí je z hlediska rychlosti zpracování výpočtu nutné vystačit s co nejmenším počtem vrstev a neuronů v síti.

4. ANALÝZA VYUŽITÍ NEURONOVÝCH SÍTÍ

Neuronové sítě se využívají při modelování složitých systémů, ve kterých nejsou známy závislosti mezi jednotlivými veličinami, problém je popsán pouze množinou realizací. Uplatňují se především tam, kde klasické počítače selhávají. Jde o úlohy, kde není znám algoritmus nebo je analytický popis pro počítačové zpracování příliš komplikovaný. Zejména se uplatní tam, kde jsou k dispozici rozsáhlé soubory příkladových dat, pokrývajících celou oblast problému.

V této kapitole jsou uvedeny obecné vzorce, které jsem převzal z literatury [8], [10].

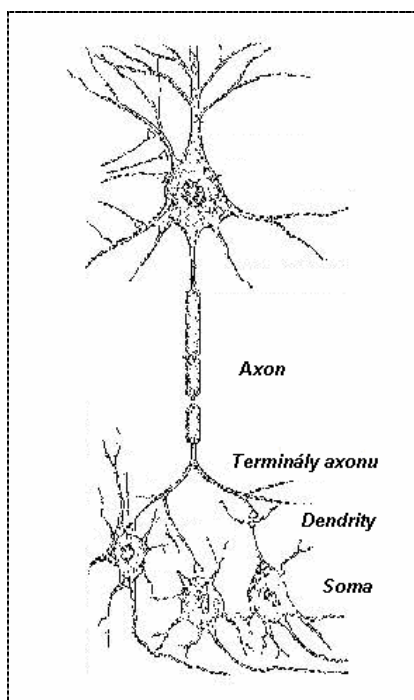
Příklady použití neuronových sítí v praxi.

- Rozpoznávání obrazu (řeči) – klasifikace dat.
- Řízení procesu výroby – aproximace.
- Transformace signálů.
- Komprese dat.
- Expertní systémy.

4.1 Biologická neuronová síť

Neuronové sítě vycházejí z biologické předlohy uspořádání neuronů v lidském mozku. Mozková kůra obsahuje 13 až 15 miliard neuronů, z nichž každý může být propojen až s 5000 jinými neurony.

Obrázek převzat z elektronických skript pro předmět Umělá inteligence.

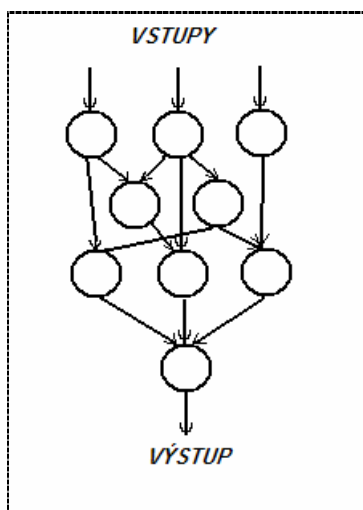


Obr. 08. Biologická neuronová síť

Propojení mezi neurony je realizováno tak, že terminály axonu se stýkají s výběžky - trny dendridů jiných neuronů. K přenosu informace slouží mezineuronové rozhraní nazývané synapse, které mohou být buďto excitační, rozšiřující vzruch, nebo inhibiční, vzruch tlumící. Pokud je neuron na vstupech podrážděn a toto podráždění přesáhne určitou hraniční mez nazývanou práh, sám generuje impuls a tím je zajištěno šíření informace.

4.2 Umělá neuronová síť

Umělá neuronová síť se může skládat z několika vzájemně propojených neuronů.



Obr. 09. Umělá neuronová síť

Mezi základní vlastnosti neuronových sítí patří.

- Schopnost učení - schopnost nastavit parametry případně změnit topologii tak, aby síť nejlépe modelovala daný problém.
- Robustnost - stabilita vůči chybám vstupních dat.
- Paralelismus - paralelní architektury umožňující rychlejší výpočty.

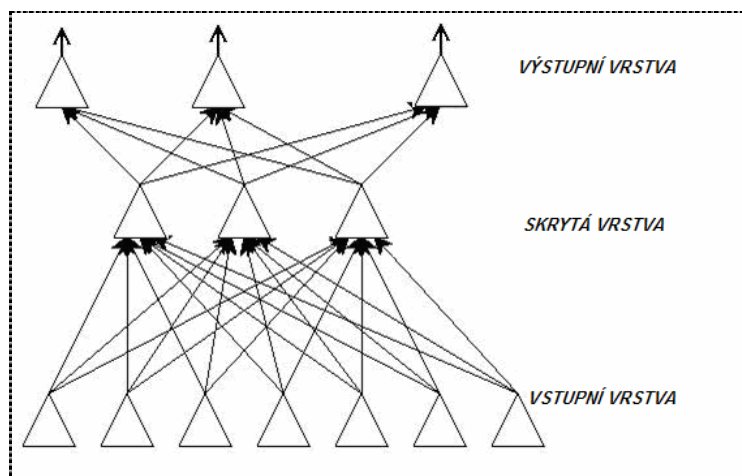
4.2.1 Topologie různých typů neuronových sítí

Neuronová síť je systém složený z většího počtu procesních jednotek (neuronů), které jsou mezi sebou navzájem propojeny (topologie). Učení neuronové sítě spočívá v nastavení parametrů jednotlivých neuronů popřípadě v modifikaci topologie sítě.

Vybrané modely neuronových sítí.

- Vícevrstevný perceptron.
- Kohonenovy mapy.
- Neuronové sítě s přepínacími jednotkami.
- Hopfieldova síť.

4.2.1.1 Vícevrstevný perceptron

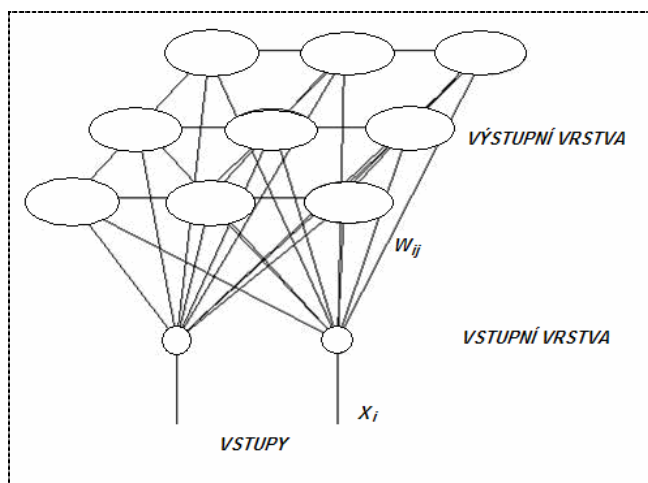


Obr. 10. Vícevrstevný perceptron

Vlastnosti a charakteristika vícevrstevného perceptronu.

- Topologie: po sobě jdoucí vrstvy představují bipartitní grafy.
- Spojité přechodové funkce (sigmoidy).
- Gradientní metoda učení (Back-propagation).

4.2.1.2 Kohonenova samoorganizační síť

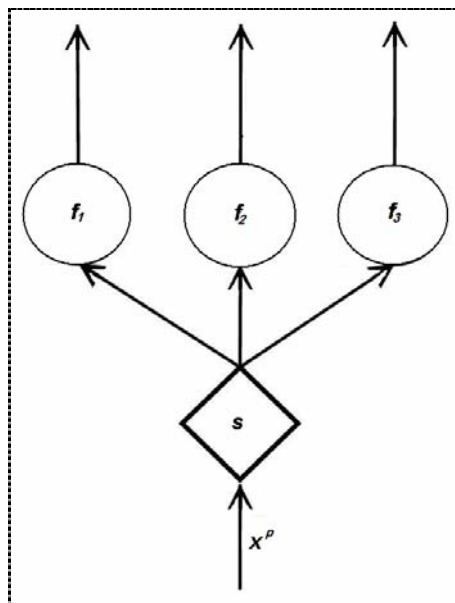


Obr. 11. Kohonenova samoorganizační síť

Vlastnosti a charakteristika Kohonenovy samoorganizační sítě.

- Jednovrstvá topologie.
- Váhy každého neuronu ve výstupní vrstvě definují polohu neuronu v prostoru.
- Výpočet probíhá na základě kompetice (pro každý vstup je aktivní právě jeden neuron).
- Učení – po zpracování vstupu se upraví váhy neuronu (j).

4.2.1.3 Neuronové sítě s přepínacími jednotkami



Obr. 12. Neuronová síť s přepínacími jednotkami

Neuron s přepínací jednotkou se skládá z těchto částí.

- Jedné přepínací jednotky.
- Několika výpočetních jednotek.
- Přepínací jednotka distribuuje vstupy mezi výpočetní jednotky.
- Výpočetní jednotka představuje „klasický“ neuron.

4.2.1.4 Hopfieldova síť

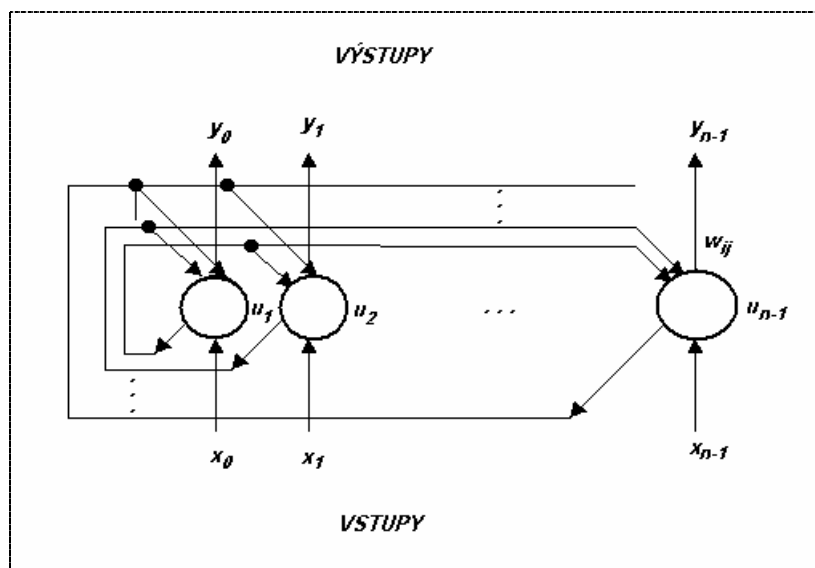
Vlastnosti a charakteristika Hopfieldova sítě.

- Úplná topologie (cyklická).
- Přípustné pouze binární vstupy.
- Výpočet probíhá do ustálení stavů.
- Používá se:
 - jako asociativní paměť,
 - klasifikátor,
 - k optimalizaci.

Hopfieldova neuronová síť (HNS) patří do kategorie tzv. autoasociativních sítí. Síť lze využít jako klasifikátor, k řešení optimalizačních problémů. Využití HNS jakožto asociativní paměti se používá pro rozpoznávání obrazového vstupu.

Struktura Hopfieldovy sítě

Hopfieldovu neuronovou síť ztvárňuje graf, kde každý neuron představuje jeden vstup a jeho výstup je přímo propojen přes váhy se vstupy všech ostatních neuronů sítě. Síť je symetrická s diagonálně symetrickou matice vah. Struktura HNS je následující.



Obr. 13. Struktura Hopfieldovy sítě

Popis struktury Hopfieldovy sítě.

- $x_0, x_1, \dots, x_{n-1}$ představují vstupy sítě.
- $u_0, u_1, \dots, u_{n-1}$ jsou stavy v jednotlivých fázích činnosti sítě, jenž se v následující fázi stávají opět vstupy.
- $y_0, y_1, \dots, y_{n-1}$ jsou výstupy sítě, které jsou vlastně totožné se stavy $m_0, m_1, \dots, m_{n-1}$. Jelikož vybavování je iterační proces, teprve ustálený stav neuronů je směřodátný.
- Váhy w_{ij} jsou v obou směrech shodné ($w_{ij} = w_{ji}$), pro je váhová matice diagonálně symetrická. Jelikož na neurony nepřivádíme jejich vlastní výstup, diagonála váhové matice je nulová. Každý neuron obsahuje přenosovou funkci a práh.
- Prah je u HNS nulový. Reálné neuronové sítě mají přenosovou funkci neuronů podobnou sigmoidě, pro zjednodušení lze použít i skokovou funkci.

Učení vzoru probíhá jednorázově. Vzor se naučí modifikací vah ve váhové matici sítě. Sít nemá žádné skryté vrstvy a proto ji nelze použít pro kódování dat.

Učení

Pro vzor o N elementech mějme čtvercovou matici vah $N \times N$. Každý prvek vznikne vynásobením i -tého vstupu s j -tým vstupem, pro $i = j$ je váhový koeficient roven nule. Jelikož elementy nabývají hodnot $+1$ či -1 , i matice vah bude obsahovat pouze hodnoty $-1, +1, 0$ (diagonála). Výsledná matice je součtem dílčích matic jednotlivých vzorů, kterých je M . Formálně lze tento proces zapsat takto [8].

$$w_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s, & \text{pro } i \neq j \\ 0, & \text{pro } i = j, 1 \leq i, j \leq N \end{cases} \quad (3)$$

Označení x^s zde neznamena s -tou mocninu x , ale označení elementu x z s -tého vzoru, kde w_{ij} je váha mezi neurony i a j . x_i resp. x_j je s -tý element vstupního vzoru, jenž nabývá hodnot $+1$ nebo -1 . Tudíž výsledkem učení je ustavení synaptických vah, přičemž váha narůstá, pokud jsou hodnoty mezi dvojicí neuronů shodné. Klesá, pokud jsou rozdílné.

Vybavování

Vybavování je na rozdíl od učení iteračním procesem. Postup je následující.

- Na vstupy neuronů předložíme neznámý vzor (inicializace).
- Provedeme výpočet výstupů neuronů sítě.
- Po každém cyklu jsou výstupy upraveny a slouží jako vstupy v následujícím cyklu.
- Pokračujeme tak dlouho, až se dojde ke stabilizaci sítě, tj. stav se během dvou cyklů nezmění na žádném neuronu.

Formálně tento algoritmus lze zapsat.

Krok 1: Inicializace stavů [10].

$$\mu_i(0) = x_i, 0 \leq i \leq N-1, \quad (4)$$

kde

- $\mu_i(t)$ – je stav i -tého neuronu v čase t .
- x_i – je element obrazce (s hodnotou +1 nebo -1).

Krok 2: Iterace až do ustálení (nalezení odpovědi) [10].

$$\mu_i(t+1) = f_p \left[\sum_{j=0}^{N-1} w_{ij}, \mu_j(t) \right], 0 \leq i \leq N-1, \quad (5)$$

kde

- f_p – je přenosová funkce (obecně může být nelineární).

Krok 2 se provádí tak dlouho, dokud neplatí [10].

$$\forall i : \mu_i(t+1) = \mu_i(t), \quad (6)$$

potom výstupy neuronů $y_i = \mu_i(t)$ jsou odpovědi - elementy výstupního obrazce.

Základní principem HNS spočívá v porovnávání obrazců podle Hammingovy metriky a správná odpověď představuje stav, jenž má Hammingovu vzdálenost k předloženému obrazci nejmenší.

4.3 Algoritmy pro učení neuronových sítí

Algoritmy učení neuronových sítí jako např. back-propagation jsou založeny na metodě nejmenších čtverců. Na množině funkcí vstup-výstup sítí daného typu minimalizují funkcionál empirické chyby určený vzorkem empirických dat [8].

$$z = \{(u_i, v_i) \in R^d \times R, i = 1, \dots, m\}, \quad (7)$$

kde

- m – je velikost vzorku.
- R – značí množinu reálných čísel.

Při jeho minimalizaci se hledá funkce f z množiny funkcí vstup-výstup, pro kterou je hodnota co nejmenší. Předpoklad tohoto postupu je že tvar, který známe jen na základě vzorku empirických dat zatížených chybami měření, se dá vyjádřit jako funkce vstup-výstup zvoleného typu sítě. Volba typu sítě tedy znamená volbu hypotézy o tvaru, který hledáme. Schopnost generalizace lze docílit omezením množiny všech funkcí vstup-výstup sítí daného typu pouze na funkce, které mají nějaké vlastnosti vystihující určitou znalost o povaze úlohy, kterou se síť učí. Téměř všechny typy sítí používané v neuropočítání mají tzv. univerzální aproximační vlastnost. Funkce vstup-výstup těchto sítí mohou aproximovat s libovolnou přesností všechny reálné funkce (např. spojité funkce). Zvyšování přesnosti aproximace samozřejmě zvyšuje nároky na velikost sítě, takže omezení počtu výpočetních jednotek sítě automaticky omezuje množinu hypotetických funkcí, které lze sítí daného typu s dostatečnou přesností aproximovat. Geometrické vlastnosti takových hypotetických funkcí mají ale dost složitý popis, který nenabízí dostatečně intuitivní náhled do jejich vlastností a proto se nehodí k modelování generalizace.

4.3.1 Back propagation

Algoritmus Back-propagation je jedním z nejúčinnějších nástrojů ze všech modelu neuronových sítí. Český výraz pro Back-propagation je zpětné šíření. Používá se pro neuronovou síť i pro algoritmus vícevrstevných dopředných neuronových sítí. Zde se chyba výstupní vrstvy zpětně přepočítává do předchozích vrstev a podle její hodnoty se upravují váhy. Back-propagation je tedy vícevrstvá síť s učitelem.

Základní vztah pro obecný neuron [8].

$$y = f\left(\sum_{i=1}^n w_i x_i + w_0\right), \quad (8)$$

kde

- x_i – jsou vstupy neuronu (celkem n).
- w_i – jsou synoptické váhy.
- f – nelineární přenosová funkce neuronu.
- w_0 - práh.
- y – výstup neuronu.

Vzorec popisuje funkci neuronové sítě. Pro realizaci obecné transformační funkce T je třeba použít síť, která má alespoň tři vrstvy s dopřednými vazbami. První z nich je vstupní vrstva s n prvky, která uskutečňuje pouze rozdělení vstupních signálů na všechny prvky druhé vrstvy. Tato vrstva (skrytá) má $2n+1$ výkonných prvků. Třetí (výstupní) má m prvků.

Algoritmus zpětného šíření šíří odchylku od vrstvy výstupní až k vrstvě vstupní. Algoritmus zpětného šíření se snaží minimalizovat energii neuronové sítě. Energie je mírou naučenosti, tedy odchylky mezi skutečnými a požadovanými hodnotami výstupů neuronové sítě pro danou trénovací množinu, můžeme ji vyjádřit pomocí energetické funkce.

Energetická funkce je definována následovně [8].

$$E = \frac{1}{2} \sum_{i=1}^m (y_i - d_i)^2 , \quad (9)$$

kde

- m – počet výstupů sítě, i -tý výstup.
- y_i – i -tý výstup.
- d_i – požadovaný výstup.

Funkci si lze představit jako plochu v prostoru. Okamžitá hodnota vah a prahů je pak bodem na této ploše. Z tohoto bodu se snažíme dosáhnout minima energie. Minimum energie, tedy minimální odchylka skutečných dat od požadovaných dat. V algoritmu se objevují ještě dva další parametry - rychlost učení, koeficient, který určuje délku kroku a tzv. setrvačnost, délka cesty určitého směru dokud nenastane nová změna gradientu.

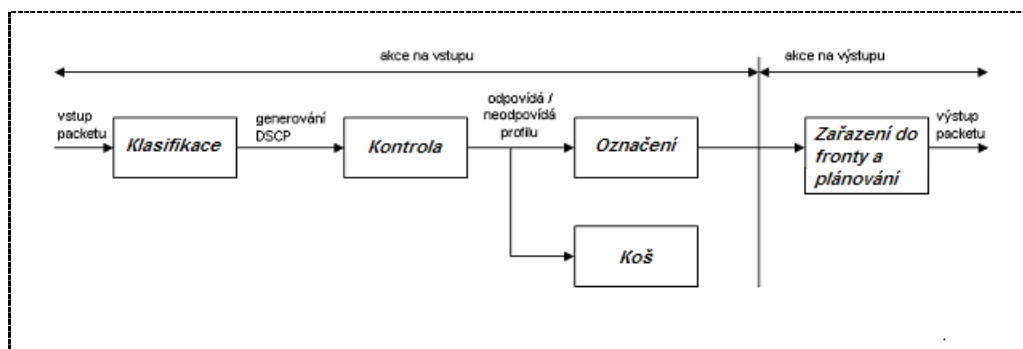
5. NÁVRH PŘEPÍNAČE ŘÍZENÉHO NEURONOVOU SÍTÍ

U klasické ethernetové sítě se každý paket doručuje v co nejkratší době. Všechny pakety mají stejnou šanci na doručení. Pokud dojde k zahlcení sítě, tak mají i stejnou šanci k zahození.

Pokud bude správně nakonfigurovaná služba QoS (kvalita služby), je možnost vybírat jednotlivé datové toky a priorizovat je podle jejich důležitosti před ostatními. Preferovanému datovému toku lze s použitím správy zahlcení přiřadit větší prioritu pro zpracování, což má následně pozitivní vliv na výkon aplikací a efektivnější využití šířky pásma.

Základní QoS model při zpracování paketu

Na obrázku jsou naznačeny akce probíhající na vstupu a výstupu.

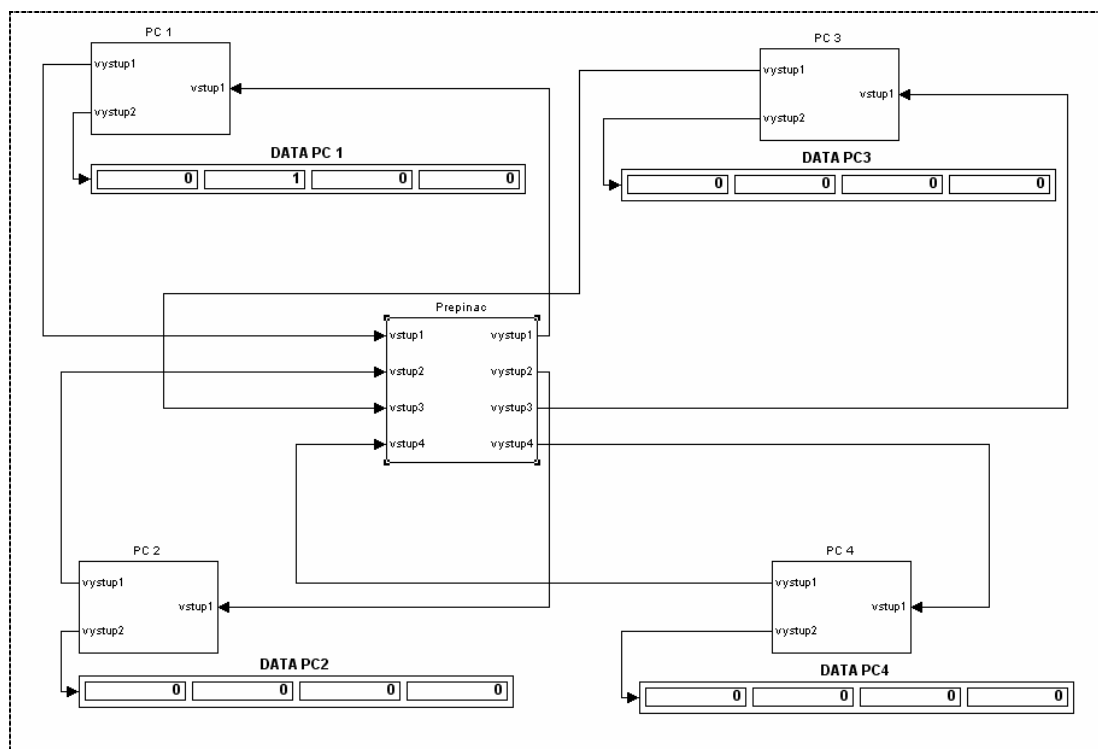


Obr. 14. Akce při zpracování paketu

Na vstupu dochází nejprve ke klasifikaci paketu. Je tím myšleno prozkoumání obsahu (dat) paketu a určení hodnoty DSCP. Následující akcí je porovnání DSCP hodnoty s hodnotami přednastavenými a dochází zde k rozhodnutí zda paket vyhovuje. Pokud paket nevyhovuje nakonfigurovaným parametrům je zahozen. Pokud vyhovuje je překlasifikován a je mu upravena hodnota DSCP. Akce na výstupu – zařazení do fronty a plánování. Zde je na základě hodnoty DSCP rozhodnuto, do které výstupní fronty se paket zařadí. Následně je fronta dle nastavených vah obsloužena.

5.1 Popis návrhu

Vlastní návrh jsem provedl v prostředí Simulink, který je součástí programu Matlab. Jedná se o jednoduché zapojení čtyř počítačových stanic propojených pomocí přepínače. Stanice jsou pracovně označeny PC1, PC2, PC3 a PC4.

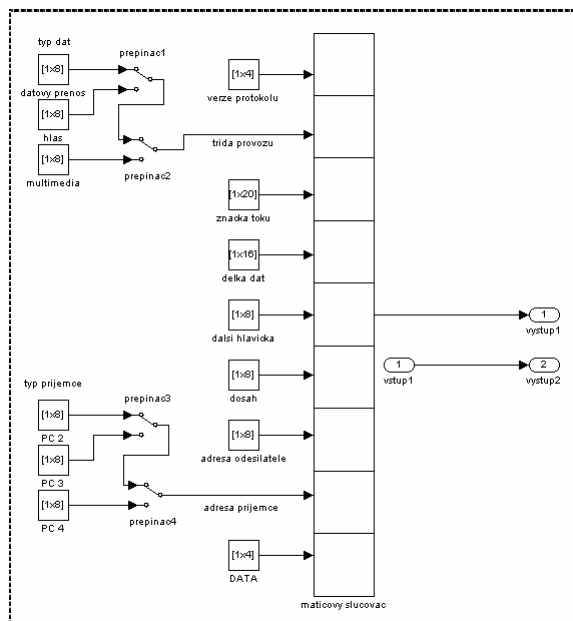


Obr. 15. Zapojení testovací sítě

Jak vidíme na obrázku, výstupy ze stanic jsou přivedeny na vstupy přepínače, zde jsou dále zpracovány. Výstupy přepínače jsou pak navedeny do vstupů stanic. Stanice obsahují ještě jeden výstup ten je připojen na *display*, který slouží pouze pro ilustrační zobrazení přijatých dat.

5.2 Počítačové stanice

Výstupem z každé stanice je paket navržený dle IPv6 složený z několika částí. Na následujícím obrázku si přiblížíme jednotlivé části.



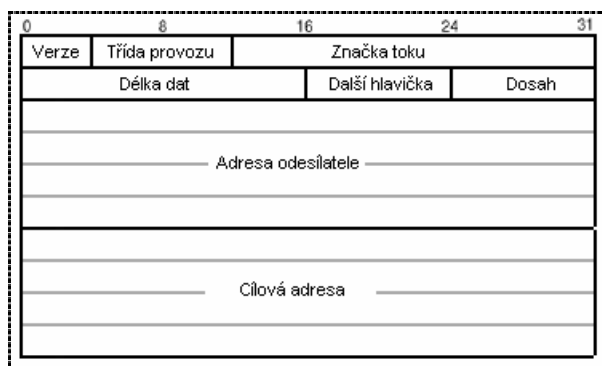
Obr. 16. Vysílaný paket ze stanice

Po spuštění aplikace se na *vystup1* objeví vektor (paket) o velikosti 84bitů. Tento vektor je tvořen dílčími vektory, které jsou seskupeny pomocí prvku *maticovy slučovac*.

- Verze protokolu (4bity).
- Třída provozu (hodnota DSCP - 8bitů).
- Značka toku (20bitů).
- Délka dat (16bitů).
- Další hlavička (8bitů).
- Dosah (8bitů).
- Adresa odesílatele (8bitů).
- Adresa příjemce (8bitů).
- Vysílaná data (4bity).

V části stanice je propojen *vstup1* na *vystup2* a ten je dále připojen na display sloužící k zobrazení přijatých dat.

Ve skutečnosti se hlavička paketu IPv6 od znázorněného liší a to hlavně z pohledu velikosti. V hlavičce datagramu IPv6 má adresa odesílatele vyhrazené místo o velikosti 128bitů. Totéž platí o velikosti cílové adresy (adresa příjemce).



Obr. 17. Základní hlavička IPv6

U protokolu IPv6 nese informaci s hodnotou DSCP třída provozu (Traffic Class). Ta vyjadřuje prioritu paketu a jeho zařazení. Slouží k tomu, aby tato položka umožnila IP poskytovat služby se zaručenou kvalitou.

Položka verze identifikuje verzi IP paketu (verze 6). Značka toku (Flow Label) je zde označen proud paketů se společnými vlastnostmi (odesílatel, adresát, apod.) a slouží k tomu, aby přepínač rychle rozpoznal, že paket je součástí určitého toku. Délka dat nese informace o délce paketu. Adresa odesílatele a cílová adresa zabírá 80% velikosti hlavičky.

5.3 Přepínač

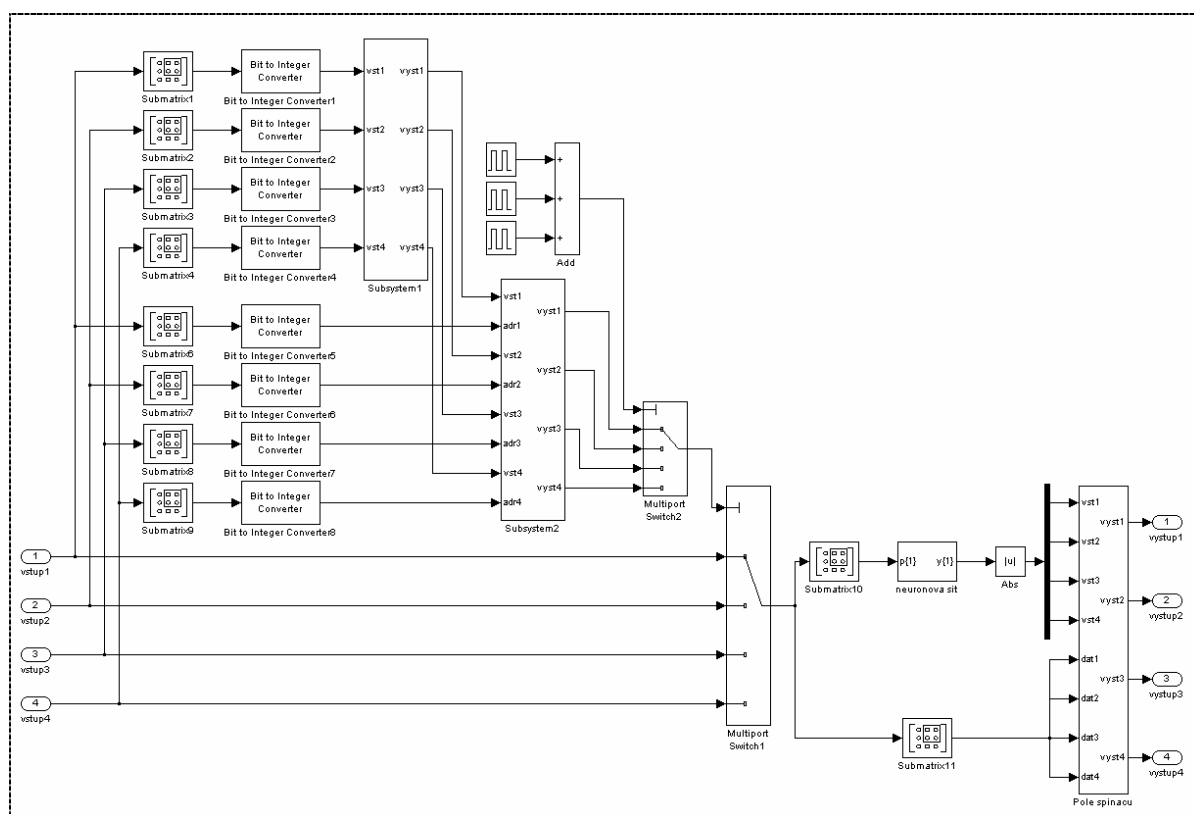
Pro upřednostnění jednoho datového toku před ostatními, je potřeba si toky nějakým způsobem označit a rozdělit do tříd (tzv. klasifikovat). To, do jaké třídy je paket klasifikován je přenášeno pomocí osmi bitů. Těchto osm bitů nese hodnotu DSCP (Differentiated Services Code Point value). Tato hodnota reprezentuje QoS klasifikaci a má rozsah 0 až 63.

V mém případě jsem k prioritizaci použil jednoduché pravidlo. Mezi stanicemi jsou vysílány tři typy dat. Jedná se o prostý datový přenos, přenos hlasu a nebo multimediální přenos. Aplikace je navržena tak, že upřednostní nejprve toho, který chce přenášet multimédia. Jako další se dostane na řadu stanice, která přenáší hlas. A jako poslední bude obslužen ten, který přenáší data.

Pokud více stanic posílá stejný typ dat, která se načtou do bufferu přepínače, je jim postupně zvyšována priorita podle pořadí příchodu.

Samotná data jsou připojena až na konec paketu a budou se lišit podle odesílatele. Data jsou pro jednodušší demonstraci na displeji o velikosti 4bitů. V praxi jsou samozřejmě data hlavní částí paketu a svojí velikostí několikanásobně přesahují hlavičku paketu.

Při pohledu na zapojení přepínače jej můžeme rozdělit podle funkcí na dvě základní části. Jedna část má za úkol provést klasifikaci tzn. na základě typu přenášených dat, upřednostní obsluhu daného toku dat. Druhá část je řízena neuronovou sítí a má za úkol poslat data správnému adresátovi.



Obr. 18. Zapojení přepínače

Jednotlivé vstupy jsou přivedeny do prvků *Submatrix1-4* ty vyberou z celého paketu požadovanou informaci (hodnotu DSCP) o typu přenášených dat. Signál je dále převeden z posloupnosti bitů na hodnotu *integer*, pro další zpracování.

5.3.1 Klasifikace

Jde o proces rozlišování jednoho typu toku od druhého zkoumáním obsahu paketů. Klasifikace je prováděna pouze pokud má přepínač podporu QoS.

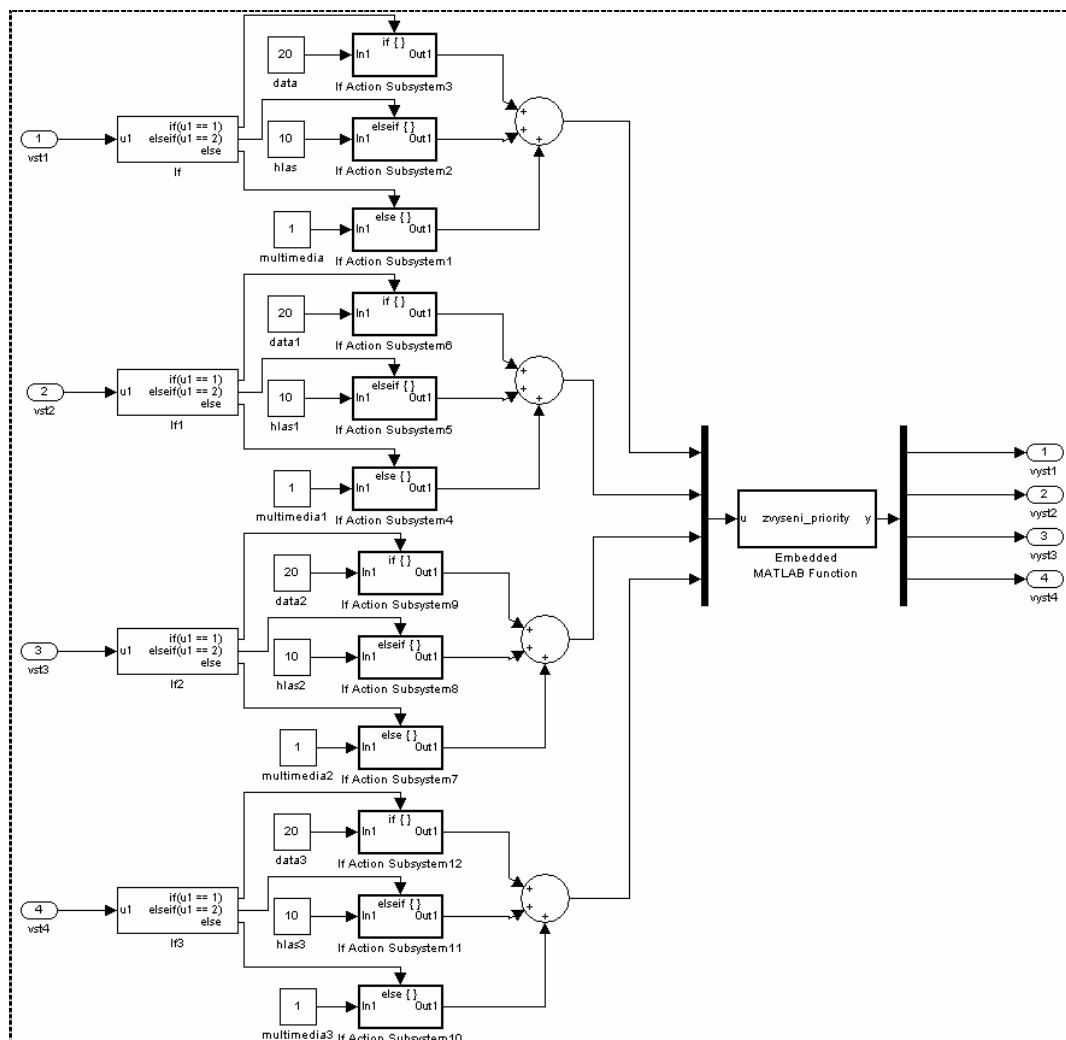
V mém případě je paket klasifikován z hodnoty DSCP v příchozím paketu a tato hodnota je dále využívána pro interní potřebu přepínače.

Rozlišovány budou pakety s obsahem dat, hlasu nebo multimédií.

Typ dat	Hodnota DSCP	Hodnota priority
DATA	00000001	20
HLAS	00000010	10
MULIMÉDIA	00000011	1

Tab. 03. Typ dat a jejich hodnota DSCP

Aplikace je navržena tak, že upřednostňuje multimédia před hlasem a datový tok je obslužen s nejmenší prioritou. Po separaci potřebného pásma a převedení signálu na desítkové číslo dále signál vede do prvku *Subsystem1*. Tento prvek je dílčí podsystém, který obsahuje další obvody.



Obr. 19. Prvek *Subsystem1*

Každý podsystém má svoje vstupy a výstupy. Zde jsou vstupy přivedeny do prvku *if*, ten simuluje podmínku, kterou musí signál splnit. Při detekci typu datového přenosu je signálu přidělena priorita s hodnotou 20, při přenosu hlasu je přidělena hodnota 10 a při přenosu multimédií hodnota 1. Přičemž čím nižší je hodnota tím vyšší priorita pro obslužení. Po přidělení priority ještě signál prochází prvkem *Embedded MATLAB Function*, tento prvek slouží pro aplikaci libovolné funkce napsané ve formátu M-file. Funkce *zvyseni_priority* slouží k postupnému zvyšování hodnoty priority o hodnotu 1, v případě shodných typů

vysílaných dat. Např. pokud odešlou všechny stanice hlasový paket, bude jim přiřazena hodnota priorit 10-13.

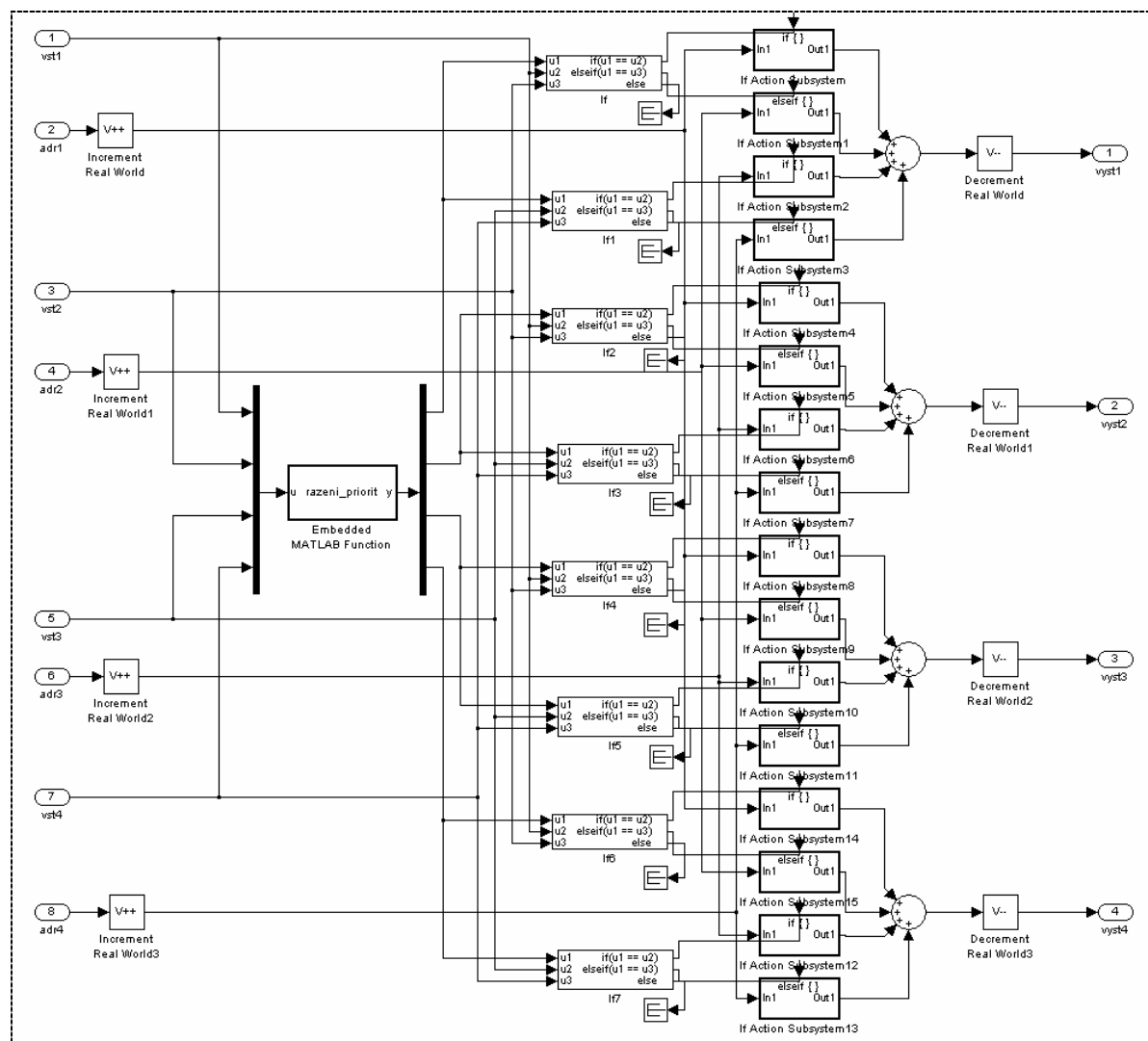
Z výstupu *Subsystem1* jsou signály přivedeny do prvku *Subsystem2*. Na vstupy *Subsystem2* jsou také přivedeny adresy odesílatelů ve formě desítkového čísla – *adr1* - 4.

Adresy odesílatelů.

Stanice	Adresa
PC 1	00000000
PC 2	00000001
PC 3	00000010
PC 4	00000011

Tab. 04. Tabulka adres odesílatelů

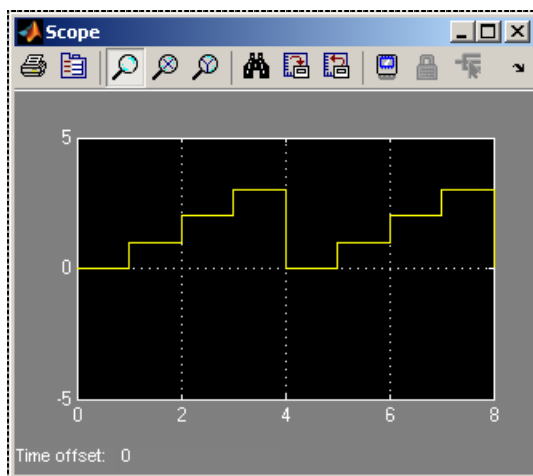
Zapojení podsystému *Subsystem2*.



Obr. 20. Prvek *Subsystem2*

V tomto podsystému jsou jednotlivé hodnoty priorit seřazeny vzestupně v prvku *Embedded MATLAB Function*, který obsahuje funkci *razeni_priorit* ta realizuje samotný proces řazení. Dále je signál porovnáván se vstupním signálem, aby podsystém postupně obsloužil linky s nejnižší hodnotou priority.

Hodnoty výstupních signálů *vyst1 – 4* z prvku *Subsystem2* jsou přivedeny na více-vstupní spínač *Multiport Switch2*. Tento spínač je ovládán třemi generátory, které generují signál o amplitudě 1, 2 a 3 po nastavené periodě. Signál je dále sloučen pomocí bloku *Add* (suma) jehož výstupní signál vypadá takto.



Obr. 21. Výstup signálu z bloku *Add*

Znamená to, že vstupy spínače *Multiport Switch2* jsou obslouženy postupně sešhora dolů. Výstupní hodnoty tohoto spínače jsou přivedeny na řídicí port spínače *Multiport Switch1*, který obsluhuje linku PC1 až PC4 s právě nejnižší hodnotou priority.

5.3.2 Značkování

Po klasifikaci je paket prověřován zda nepřekročuje nastavenou šířku pásma. Pokud k tomuto dojde jsou pakety označeny jako neodpovídající profilu. Paket může být dále pře-klasifikován nebo zahozen. K tomuto účelu slouží v aplikaci podmínka v bloku *if* v *Subsystem1* ten porovnává, zda jsou doručená data ve formátu, v jakém je přepínač očekává.

5.3.3 Adresace

Pro adresaci využívá navrhnutý přepínač individuálních (unicast) adres. Každá z adres identifikuje jedno síťové rozhraní – jednoho adresáta. Data mají být doručována právě jedné stanici. Standardní IPv6 adresa je v tomto formátu.

123A:0000:0000:0000:0000:0000:0000: 0000.

Obr. 22. Formát adresy IPv6

Protože je 0 častou hodnotou, která se vyskytuje u adres. Je možné zápis zkrátit na hodnotu.

123A::

Obr. 23. Zkrácený formát adresy

V navrhnuté síti se jedná o propojení 4 počítačových stanic. Jejich adresy jsou v následující tabulce.

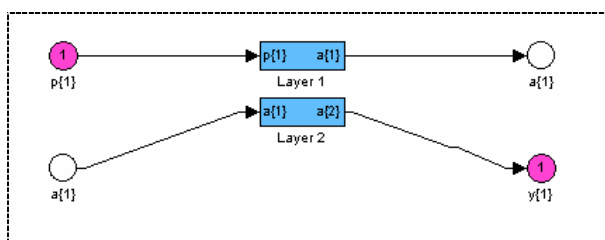
Spojení PC stanic	Adresa spoje
PC 1 – 2	00000001
PC 1 – 3	00000011
PC 1 – 4	00000010
PC 2 – 1	00000111
PC 2 – 3	00000101
PC 2 – 4	00000100
PC 3 – 1	00001001
PC 3 – 2	00001011
PC 3 – 4	00000110
PC 4 – 1	00001000
PC 4 – 2	00001010
PC 4 – 3	00001100

Tab. 05. Tabulka adresace spojení

Přepínač vybírá podle priority pořadí linek, které budou obslouženy přednostně. Na výstupu prvku *Multiport Switch1* se objevují celé obsluhované pakety.

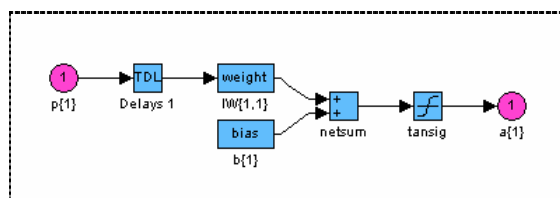
Dále je paket nutno rozdělit do dvou větví. V jedné větvi se vyfiltrují z celého paketu přenášená data, k tomuto slouží prvek *Submatrix11*. Přenášená data jsou přivedena jako vstupy *dat1–4* do pod systému *Pole spinacu*. V druhé větvi obvodu je vyfiltrován adresát, kterému má být vysílaný paket doručen. Z výstupu prvku *Submatrix10* je přiveden na vstup pod systému *neuronova sit*.

Tento pod systém zabezpečuje správnou adresaci a doručování dat. Pod systém byl vygenerován z M-file souboru pomocí příkazu *gensim(net,st)*. Tento příkaz je schopen vygenerovat blok neuronové sítě do prostředí Simulink. První parametr příkazu, který je v závorce *net* představuje naučenou neuronovou síť, která bude součástí bloku. Druhý parametr *st* představuje vzorový čas (implicitně 1). Neuronová síť má jednu základní vrstvu a vrstvu skrytou.



Obr. 24. Dvouvrstvá neuronová síť (jedna vrstva skrytá)

Každá z vrstev se skládá z bloků, které jsou pro neuronové sítě charakteristické. Jedná se o váhy, prahové hodnoty, sumu a aktivační funkci.



Obr. 25. Bloky vrstvy (Layer 1)

Vstupem do neuronové sítě jsou binární adresy pro jednotlivá spojení tak, jak je zaznamenáno v tabulce adresace spojení (viz výše). Každý možný spoj má svoji vlastní adresu.

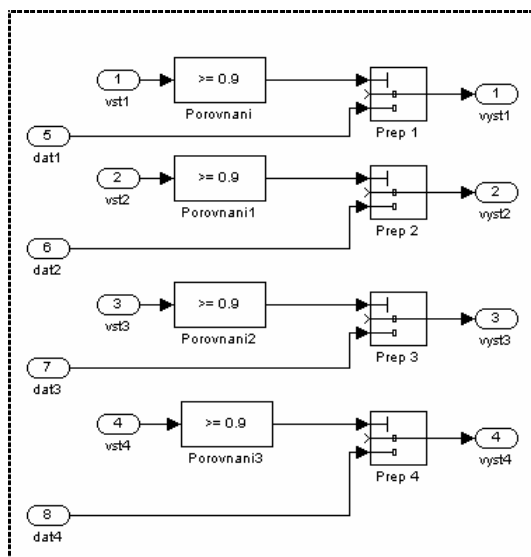
Neuronová síť má nastaveny váhy a prahy tak, aby reagovala na danou adresu požadovaným výstupem. Výstupní hodnoty neuronové sítě ovládají podsystém *Pole spinacu*.

Vstup NS	Výstup NS
00000001	01000000
00000011	00100000
00000010	00010000
00000111	10000000
00000101	00100000
00000100	00010000
00001001	10000000
00001011	01000000
00000110	00010000
00001000	10000000
00001010	01000000
00001100	00100000

Tab. 06. Pravdivostní tabulka pro NS

Prakticky to znamená, že neuronová síť zvolí výstupní port, na který je připojen cílový adresát.

Výstupy z neuronové sítě nejsou celá čísla, avšak čísla s přesností na několik desetinných míst. Prakticky se však jedná o hodnoty 0 nebo 1. V podsystému *Pole spinacu* je prvek *Porovnani*, který identifikuje hodnoty vyšší než 0,9 a zaokrouhlí je na hodnotu 1.

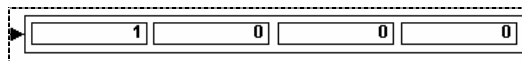


Obr. 26. Prvek Pole spinacu

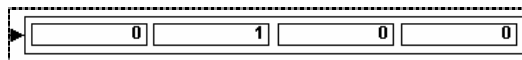
Přepínače *Prep1* – *4* pošlou na výstup vstupní data pokud se objeví na jejich řídicím portu hodnota 1. Výstupy *Pole spinacu* jsou zároveň výstupy celého přepínače. Výstupy přepínače jsou již přiváděny na vstupy jednotlivých PC stanic. K uživateli jsou již přiváděna pouze odesílaná data a ty jsou zobrazeny na display stanice.

5.3.4 Možné případy přijatých dat

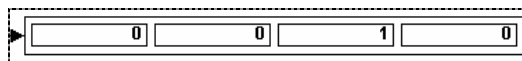
Přijatá data mohou přicházet od různých adresátů. Data vyslaná od jednotlivých PC stanic jsou v tomto tvaru.



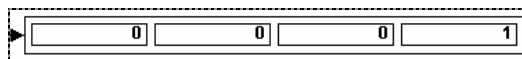
Obr. 27. Přijatá data od PC1



Obr. 28. Přijatá data od PC2

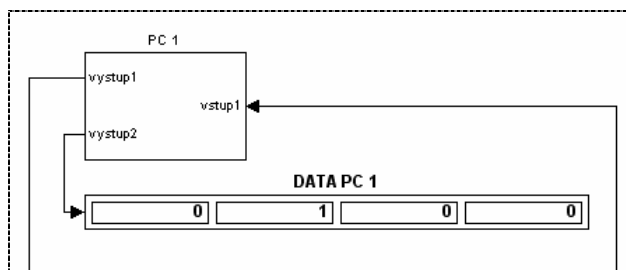


Obr. 29. Přijatá data od PC3



Obr. 30. Přijatá data od PC4

Díky tomuto principu okamžitě poznáme, od kterého odesílatele data přišly.



Obr. 31. Přichozí data v PC1

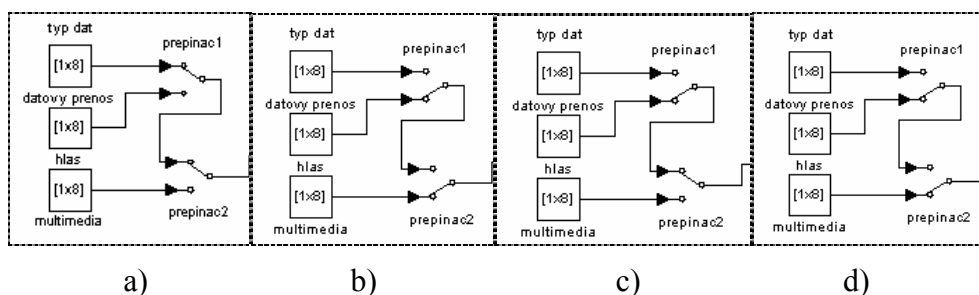
Na obrázku je zobrazen display PC 1, který zobrazuje přijatá data od stanice PC 2.

Přepínač dynamicky reaguje na vysílaná data, které je možno modifikovat pomocí manuálních spínačů.

5.3.5 Demonstrace funkce spínače

Typ vysílaných dat

Pro demonstraci funkcí spínače musíme začít u vysílaných dat. Každá ze stanic v daném okamžiku vysílá různý typ dat. Stanice 1. vysílá datový přenos, stanice 2. vysílá multimediální přenos, stanice 3. vysílá hlasový přenos a stanice 4. vysílá také multimédia.

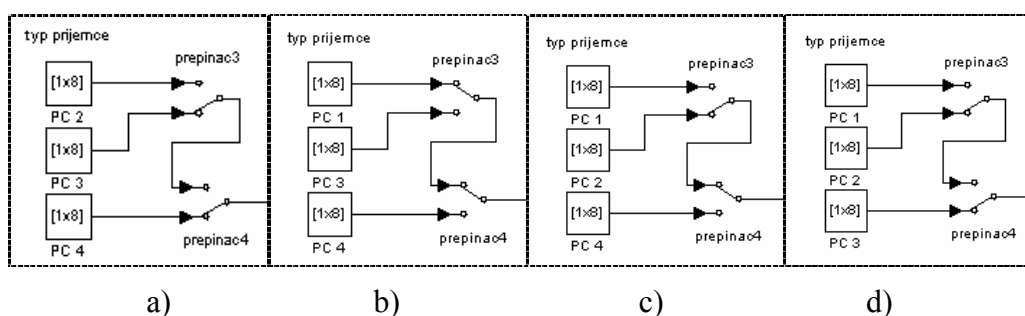


Obr. 32. Výběr vysílaných dat a) stanice 1. b) stanice 2. c) stanice 3. d) stanice 4.

Data jsou vysílána souběžně, na porty spínače tedy přijdou v jeden čas. Data se uloží do vstupního bufferu a tam je přezkoumán jejich obsah. Podle typu přenášených dat, která jsou různě náchylná na zpoždění, budou obslouženy jednotlivé linky. Nejprve jsou od sebe odděleny typy dat – data, hlas a multimédia. Postupným zvyšováním priority jsou jednotlivé pakety obsluhovány. Spínač může v jednom čase pracovat vždy jen s jedním kanálem. Postupně jsou obslouženy všechny kanály.

Typ adresáta

Každá stanice má možnost posílat svoje data libovolnému adresátovi. Adresáta volí nastavením mechanických spínačů tak jako u typu dat.



Obr. 33. Výběr adresáta a) stanice 1. b) stanice 2. c) stanice 3. d) stanice 4.

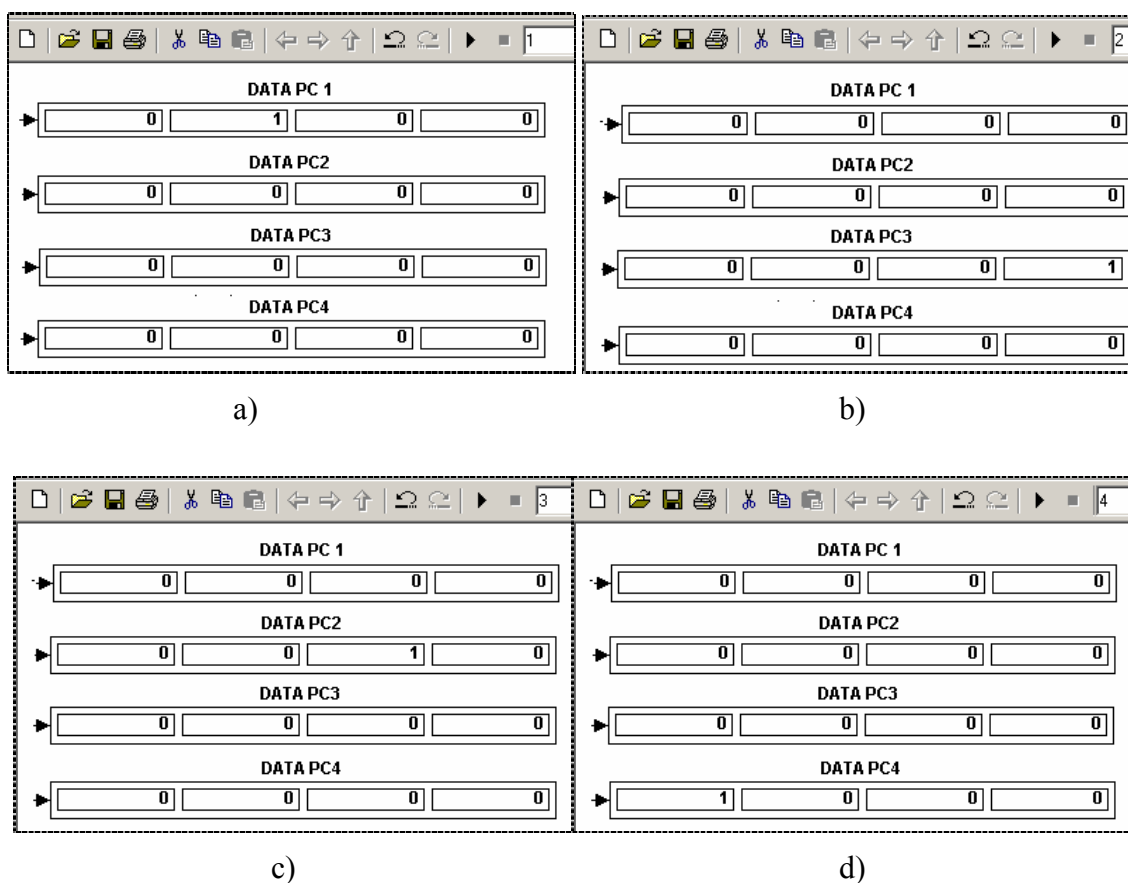
Obsluha probíhá v druhé části spínače a spočívá ve výběru výstupního portu (ke kterému je připojen adresát). Tuto operaci obstarává podsystém *neuronova sit*.

Výsledek

Na vstup každé stanice je napojen také display, který zobrazí přijatá data. Pro vhodnou ilustraci jsou vysílaná data jen zlomkem velikosti paketu, ale v praxi je tomu přesně naopak a velikost hlavičky IP datagramu je zanedbatelná oproti velikosti přenášených dat.

Data jsou také zasílána tak, aby byl na první pohled rozeznatelný odesílatel tak jak je to popsáno v kapitole výše.

Výsledek je možný demonstrovat po odkrokování jednotlivých časových okamžiků. Na obrázku jsou zobrazeny displeje stanic v jednotlivých časech.



Obr. 34. Displeje a) v čase 1. b) v čase 2. c) v čase 3. d) v čase 4.

V 1. časovém okamžiku přijala paket stanice PC1, a protože přijatá data jsou ve formátu [0,1,0,0] víme, že odesílatelem byla stanice PC2. Víme také, že stanice PC2 odesílala multimédia. V 2. časovém okamžiku přijala paket stanice PC3 od odesílatele PC4 (multimédia). Ve 3. časovém okamžiku přijala paket stanice PC2 od odesílatele PC3 (hlas). A ve 4. časovém okamžiku přijala paket stanice PC4 od odesílatele PC1 (data). Z toho vyplývá správná funkce směrovače, upřednostňuje multimediální tok před hlasovým a datovým tokem.

6. VYUŽITÍ V LABORATORNÍCH CVIČENÍCH

6.1 Teoretický úvod

Switch (česky přepínač) je aktivní síťový prvek, propojující jednotlivé segmenty sítě. Switch obsahuje větší či menší množství portů (až několik stovek), na něž se připojují síťová zařízení nebo části sítě. Pojem switch se používá pro různá zařízení v celé řadě síťových technologií. Obecnou vlastností switchů je, že analyzují procházející pakety a podle informací v nich obsažených (adres, identifikátorů apod.) rozhodují, kam paket předat dál.

Ethernet switch

Nejčastěji switch potkáte jako aktivní prvek v síti Ethernet realizované kroucenou dvojlinkou. Switch nahradil dříve používané huby (rozbočovače), které signál jednoduše kopírovaly do všech ostatních rozhraní. Pracuje zde na 2. vrstvě OSI modelu. Vedle vyššího výkonu znamená přínos i pro bezpečnost sítě, protože médium již není sdíleno a data se vysílají jen do rozhraní, jímž je připojen jejich adresát.

Adresy stanic se switch učí automaticky z procházejícího provozu, konkrétně z adres odesílatelů uvedených v rámcích, které do switchu přicházejí. Používá se algoritmus Backward Learning Algorithm. Z těchto údajů si switch automaticky plní tabulku identifikující cílová rozhraní pro jednotlivé adresy. Pokud switch dostane k doručení rámec směřující na jemu dosud neznámou adresu, chová se jako hub a rozešle rámec do všech ostatních rozhraní. Lze očekávat, že oslovená stanice pravděpodobně odpoví a switch se tak vzápětí dozví, kde se nachází.

Ethernetové switchy mají problém s cykly v síti, vytvářenými za účelem redundance. Pokud síť obsahuje cyklus, mohou pakety od stejného odesílatele přicházet chaoticky z různých rozhraní a dokonce tentýž paket může do switchu dorazit několikrát. Switch není v takovém prostředí schopen rozpoznat, kde se kdo nachází. Tento problém řeší switchy mechanismem zvaným Spanning Tree protokol, kterým se dohodnou na nepoužívání některých tras tak, aby ze sítě zmizely cykly. Vytvoří se minimální kostra sítě dosahující do všech jejích míst. Když dojde ke změně v topologii (např. rozpojení některé linky), bude aktivována některá z dosud odstavených tras tak, aby nový strom nadále pokud možno pokrýval celou síť. Tyto změny se ovšem nedějí okamžitě, je zde jisté zpoždění.

Switche mohou nabízet i některé pokročilejší funkce, jako například:

- Mmanagement - možnost upravovat nastavení switche pomocí Telnetu nebo webového rozhraní (*http*).
- VLAN - podpora virtuálních sítí.
- SNMP - vzdálená správa zařízení, hlášení určitých stavů a situací apod.

Ethernet přepínače (switche) pracují na fyzické a spojové vrstvě. Ethernet přepínače můžeme považovat za nejrozšířenější prvek v lokálních sítích. Jsou určeny pro připojení koncových stanic, kterými kromě uživatelských počítačů mohou být IP telefony, Wi-Fi přístupové body, servery atd.

Hlavním úkolem přepínačů je přepínání rámců mezi porty. Kritéria pro rozhodnutí, na který port bude příchozí rámec odeslán, jsou uložena v přepínací tabulce. V ní jsou uloženy informace vypovídající, ke kterému portu je zařízení připojeno. Každé zařízení je identifikováno unikátní hardwarovou adresou. Bývá též nazývána fyzická adresa či MAC adresa. Adresy společně s číslem příslušného portu jsou do tabulky ukládány automaticky. Přepínač při příjmu rámce vyhledá v tabulce záznam s příslušnou cílovou fyzickou adresou a odešle rámec na příslušný port. Pouze v případě, že cílová adresa není nalezena v tabulce, odešle se rámec na všechny porty.

Nejzákladnějšími přepínači jsou přepínače bez managementu s metalickými porty, které jsou vhodné pro propojení uživatelských počítačů, pro vytvoření malé lokální sítě. K dispozici jsou přepínače nejčastěji s 5, 8, 12, 16 nebo 24 porty, s podporou rychlostí 10/100 Mb/s (Fast Ethernet) nebo 10/100/1000 Mb/s (Gigabit Ethernet). Jednou z funkcí bývá Auto-MDI/MDI-X pro automatickou detekci a správné nastavení přijímacích a vysílacích pinů na metalickém portu. Lze tak použít přímý i křížený kabel.

U přepínačů s rozšířenou funkcionalitou je pro konfiguraci nutný management. Konfigurace pomocí příkazů v příkazovém řádku je možná lokálně přes sériovou konzoli (RS-232) nebo vzdáleně přes Telnet či SSH (Secure Shell, zabezpečený přístup). Poměrně snadnou konfiguraci poskytuje webové rozhraní, pro vzdálenou správu je vhodný zabezpečený webový přístup přes *https*. Pro správu rozsáhlejších sítí je vhodná podpora SNMP (Simple Network Management Protocol), která umožňuje správu jednotek z centrální SW aplikace (NMS, Network Management System), kterou nabízí přímo výrobce přepínačů. Z hlediska bezpečnosti je nejvýhodnější šifrovaná komunikace dle SNMP verze 3.

Při implementaci přepínačů do dohledových aplikací je výhodou, pokud výrobce poskytuje tzv. MIB tabulku (Management Information Base table).

Porty, sloty, PoE

Přepínače mohou být kromě metalických portů s rychlostmi 10/100 Mb/s nebo 10/100/1000 Mb/s pro připojení uživatelů osazeny také porty pro uplink, tedy pro propojení přepínačů nebo pro připojení k centrálnímu prvku. Uplink porty mohou být metalické nebo optické. Kromě pevných optických portů mohou být přepínače vybaveny sloty pro osazení moduly GBIC (Gigabit Interface Converter) nebo SFP (Small Form Pluggable, též MiniGBIC), případně 10G moduly XENPAK nebo XFP (eXtended small Form Pluggable), ke kterým se připojují optická vlákna přímo přes konektory SC nebo LC. Odpadá nutnost použití případných media-konvertorů, což je další prvek v síti, u kterého může nastat případná porucha. Media-konvertor vyžaduje také samostatné napájení a svůj prostor v rozvaděči.

L2 funkce

Pro správnou funkci přepínače je nutná stromová nebo sběrníková topologie. Kruhová topologie sítě je možná pouze v případě použití přepínačů s podporu Spanning Tree protokolu (STP, IEEE 802.1d), který umožňuje vytváření záložních spojů – smyček. Doba zotavení v případě výpadku linky nebo některého z přepínačů v kruhu činí 30 až 40 sekund. Proto je vhodnější nasazení verze Rapid Spanning Tree Protokol (RSTP, IEEE 802.1w), kde doba zotavení je 2 až 3 vteřiny.

Kvalita služeb

Pro zajištění kvality služeb jsou určeny funkce pro klasifikaci a prioritizaci dat. Klasifikace dat je možná na úrovni L2 až L4 vrstvách referenčního modelu OSI. Tyto přepínače bývají označovány jako L2/L4 nebo L2+ přepínače. Na základě klasifikace dat je možná jejich prioritizace pomocí HW front (standard IEEE 802.1p). Pořadí odesílání rámců z jednotlivých front je možné nastavit v určitém poměru nebo lze vybranou frontu striktně upřednostnit. Tyto funkce jsou vhodné především pro provoz konvergovaných služeb, tedy pro přenos dat, hlasu a videa.

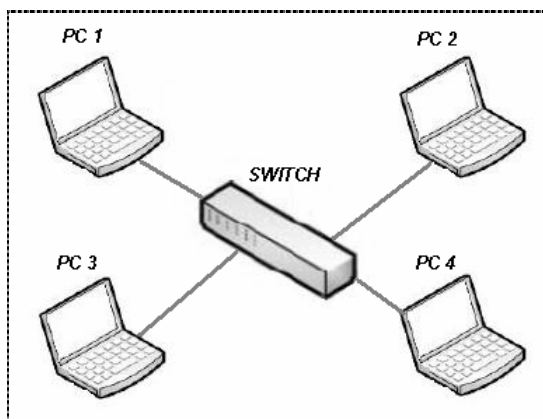
6.2 Zadání

- Vyzkoušejte si funkce přepínače.
- Navrhněte úpravu přepínače tak, aby po příchodu dat k adresátovi byl zobrazen odesílatel.
- Navrhněte úpravu přepínače tak, aby po příchodu dat k adresátovi byl zobrazen typ přenášených dat.
- Navrhněte zefektivnění kódu programu pro zvyšování priority.
- Navrhněte změnu priority určitého toku dat.

K řešení této laboratorní úlohy je předpokladem základní znalost simulačního prostředí Simulink.

6.3 Schéma zapojení

Jedná se o jednoduché zapojení čtyř počítačových stanic propojených pomocí přepínače. Stanice jsou pracovně označeny PC1, PC2, PC3 a PC4.



Obr. 35. Schéma zapojení laboratorní úlohy

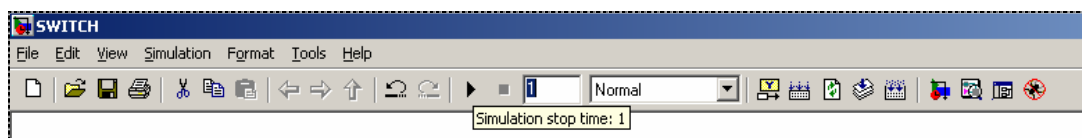
6.4 Pracovní postup

Spustíte program Matlab a v něm potom simulační prostředí Simulink pomocí ikonky . Načtete vytvořený model přepínače pomocí *File – Open....* Model je uložen na pracovní ploše počítače *SWITCH.mdl*.

Načtený model se skládá z jednoduché sítě několika počítačových stanic vzájemně propojených pomocí síťového prvku. Síťový prvek je přepínač řízený neuronovou sítí. Tato neuronová síť reaguje na vstupní data a podle typu adresáta ovládá přepínač a doručuje data do správné stanice. Pro implementaci kvality služeb přepínač pracuje jednoduše tak, že zjistí jaký typ dat je vysílán a následně upřednostní jeden datový tok před druhým. Přepínač dynamicky reaguje na změnu přenášených dat nebo změnu adresáta pomocí mechanických přepínačů.

Add1

Vyzkoušejte si funkčnost přepínače tak, že budete ve vývojovém prostředí simulaci krokovat pomocí délky simulace (*Simulation stop time*) v režimu *Normal*.

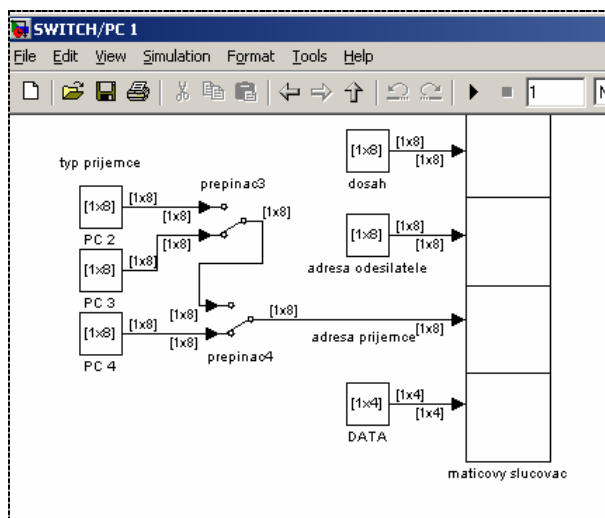


Obr. 36. Nastavení délky simulace

Zjistěte, jestli se vysílaná data periodicky opakují popř. s jakou periodou. Nastavujte délku simulace postupně od 1 až X.

Zapojení stanice *PCI-4* je prakticky totožné kromě adresy odesílatelů. Prostudujte, z čeho se skládá vysílaný paket odesílaný ze stanice PC1.

Změnou adresáta ve stanici PC 1 vypořizujte, v jakém tvaru data přicházejí. Ke změně adresáta použijte mechanických spínačů *prepinac3* resp. *prepinac4*.



Obr. 37. Výběr adresáta ve stanici PC1

Změnu proveďte u tolika stanic, dokud vám nebude jasné, v jaké tvaru jsou vysílaná data z každé stanice. Je vidno, že přijatá data jsou totožná od všech odesílatelů.

Add2

Upravte vysílaná data tak, aby na první pohled bylo jasné, od kterého odesílatele byly data vysílány. Tvar vysílaných dat zapište do tabulky.

Stanice	Tvar vysílaných dat
PC 1	
PC 2	
PC 3	
PC 4	

Tab. 07. Tabulka tvaru vysílaných dat

Výstupní signál ze spínače, který přichází do jednotlivých stanic je pouze část vysílaného paketu. Jedná se pouze o data, která jsou připojena na konec paketu.

Add3

Upravte zapojení tak, aby se do jednotlivých stanic dostal celý vysílaný paket. Na prvním displeji by se měly zobrazit přijatá data, tak jako tomu bylo na začátku. Vytvořte druhý displej a na ten přiveďte typ přenášených dat. Můžete k tomu využít prvek *Submatrix*, tak jako je tomu v případě přijatých dat. Je nutné zjistit, na jaké pozici resp. v které části vysílaného paketu se nachází informace o typu přenášených dat.

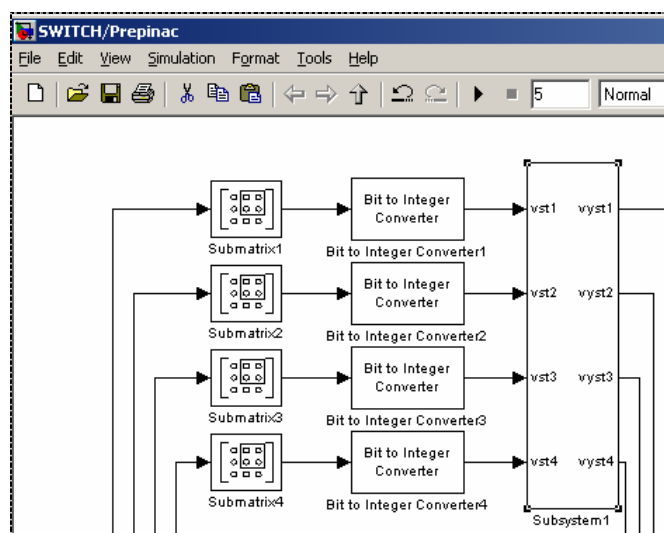
Tvar různých typů přenášených dat запиšte do tabulky.

Typ přenášených dat	Tvar typu přenášených dat (bin)	Tvar typu přenášených dat (dec)	Priorita
Data			
Hlas			
Multimédia			

Tab. 08. Tabulka tvaru typu přenášených dat

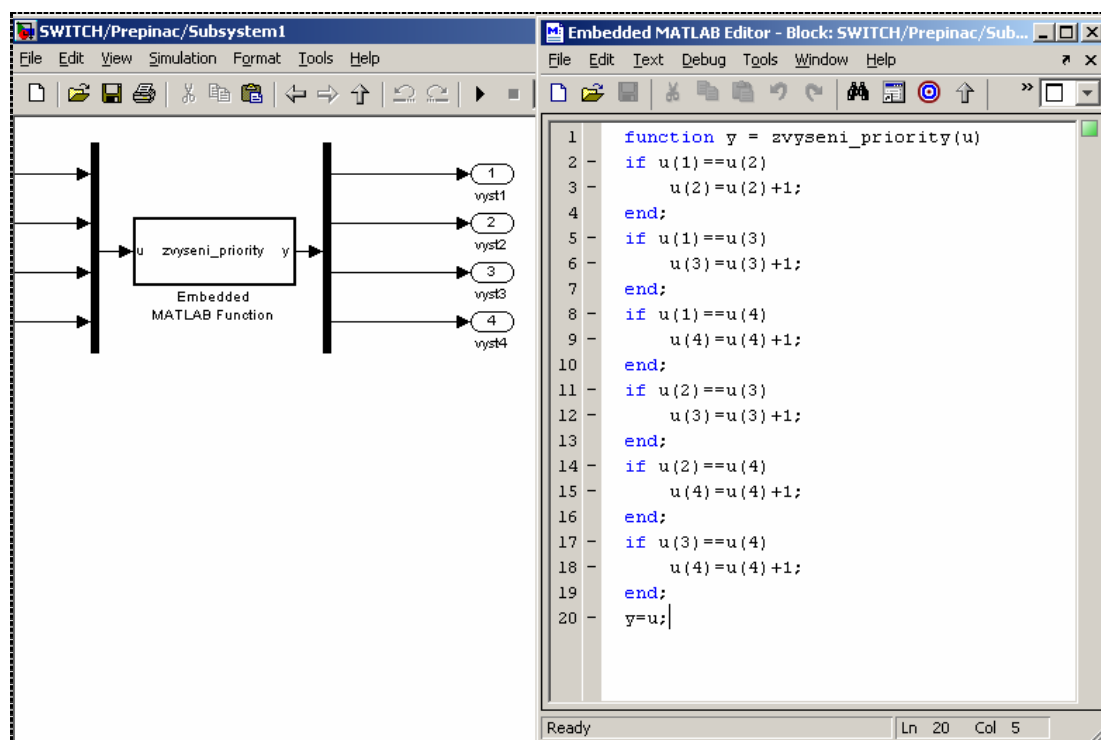
Add4

Další úkol bude spočívat ve zefektivnění kódu programu pro zvyšování priority. Uvnitř přepínače je podsystém *Subsystem1*.



Obr. 38. Subsystem1

Uvnitř podsystému se nachází prvek *Embedded MATLAB Function* je to v podstatě funkce napsaná a uložená v M-File souboru.



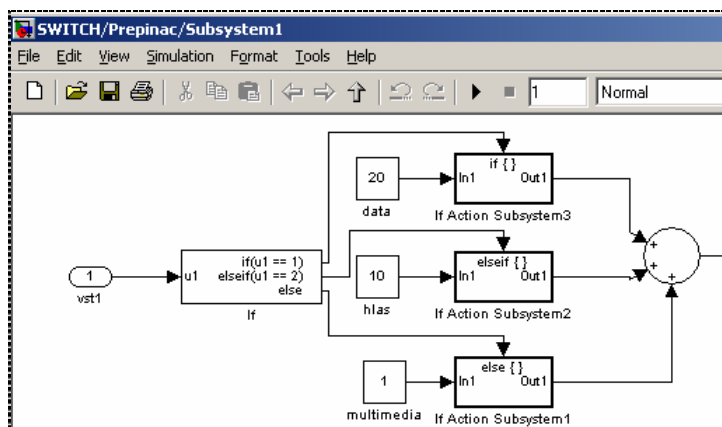
Obr. 39. Kód pro zvýšení priority

Funkce *zvyseni_priority* ve stručnosti obstarává – pokud bude mít stejnou hodnotu nějaký prvek vstupního vektoru *u* stejnou hodnotu jako jiný prvek, pak druhá hodnota bude zvýšena o hodnotu 1.

Zefektivněte kód například pomocí cyklu *for*.

Jak jste již zjistili, spínač je navržen tak, že při zahlcení ve vstupním bufferu upřednostní určitý typ dat před druhým. K prioritizaci je použito jednoduché pravidlo. Mezi stanicemi jsou vysílány tři typy dat. Jedná se o prostý datový přenos, přenos hlasu a nebo multimediální přenos. Aplikace je navržena tak, že upřednostní nejprve toho, který chce přenášet multimedia. Jako další se dostane na řadu stanice, která přenáší hlas. A jako poslední bude obsloužen ten, který přenáší data. Pokud více stanic posílá stejný typ dat, která jsou uložena v bufferu přepínače za sebou, přepínač postupně zvyšuje prioritu těm, které zaslali data nejdříve.

V podsystému *Subsystem1* jsou detekovány hodnoty *DSCP* v záhlaví paketu, kde jsou uloženy informace o typu dat v přenášeném paketu. Při detekci určitého typu datového přenosu signálu je přidělena hodnota priority. Přičemž čím nižší je hodnota tím vyšší priorita pro obsluhu.



Obr. 40. Detekce typu dat v paketu

Add5

Vaším úkolem bude změnit upřednostnění hlasového a datového toku před pakety obsahující multimédia (video). Změnu priority si ověřte krokováním.

Upravený model uložte a funkcionality demonstруйте vyučujícímu.

6.5 Závěr laboratorní úlohy

K řešení této laboratorní úlohy je předpokladem základní znalost simulačního prostředí Simulink. Původní model je odevzdán jako *SWITCH_zadani.mdl* a výsledný model se jmenuje *SWITCH_reseni.mdl*.

7. ZÁVĚR

Cílem této práce bylo seznámit se s problematikou kvality služby v konvergovaných sítích především s užitím IP protokolu verze 6. Konvergované sítě jsou schopné efektivně přenášet různé typy dat – hlasový, datový nebo multimediální tok. Tyto aplikace kladou na síť rozdílné požadavky. Datový přenos je náročný na šířku pásma a spolehlivost spojení. Hlasové a multimediální přenosy požadují konstantní pásmo a garantovanou dobu doručení, přičemž částečnou ztrátu informace lze kompenzovat opravnými metodami. Kvalita služby pro komunikační síť znamená, schopnost rozlišit typ přenášených dat a plnit jejich požadavky na zpoždění, ztrátovost nebo jitter. Těmto nárokům se musí přizpůsobit celý provoz sítě. Internet Protokol verze 6 je nástupce předcházející verze IPv4 a řeší některé jeho nedostatky. Podstatou zdokonalení IPv6 je mimo jiné rozšíření IP prostoru adres z 32 bitů na 128 bitů, což umožňuje několikanásobné zvýšení počtu jedinečných IP adres.

Podle zadání diplomové práce jsem měl za úkol seznámit se s existujícím modelem síťového prvku s neuronovou sítí a analyzovat možnosti zvýraznění QoS. Tento prvek byl dříve ve fázi návrhu a nyní je konečně realizován, avšak jedná se o velmi složitou konstrukci a z tohoto důvodu mi bylo doporučeno vytvořit vlastní model.

Návrh aktivního síťového prvku jsem realizoval v prostředí Simulink, které je součástí programu Matlab. Navrhnutý model se skládá z jednoduché sítě několika počítačových stanic vzájemně propojených pomocí síťového prvku – prepínače. Model prepínače simuluje reálný provoz počítačových stanic, které odesílají data vzdáleným uživatelům. Odesílané pakety se od sebe liší různým obsahem dat. Stanice vysílají pakety neustále za sebou a ve spínači se data uloží do vstupní paměti – bufferu. Pakety jsou následně řazeny podle typu datového toku. Informace o typu dat nese hodnota DSCP uložená v hlavičce každého paketu. S pakety je zacházeno tak, aby byla dodržena požadovaná kvalita služeb. Přednost mají datové toky, které potřebují být doručeny bez většího zpoždění, jako jsou hlasový nebo multimediální přenos. Pokud více stanic posílá stejný typ dat, která se uloží se do bufferu, jsou data obsloužena postupným zvyšováním priority.

Neuronové sítě se začínají využívat v mnoha odvětvích. Jsou výhodné při zpracování dat z důvodu rychlosti rozhodování a schopnosti učení. Ve své práci jsem využil neuronovou síť pro řízení adresace prepínače. Prepínač si adresy zapisuje do interní tabulky k dalšímu využití. Neuronová síť reaguje na vstupní data a podle typu adresáta ovládá prepínač. Data jsou

doručována přes výstupní port do cílové stanice. Přepínač dynamicky reaguje na změnu přenášených dat nebo adresáta pomocí mechanických spínačů.

Model přepínače lze využít v laboratorních cvičeních. Navrhnutá úloha simuluje možný provoz mezi stanicemi v malé lokální síti. K řešení této laboratorní úlohy je předpokladem základní znalost vývojového prostředí Simulink. K vypracování úlohy jsou zadány následující úkoly: úprava přepínače, aby po příchodu dat k adresátovi byl zobrazen na displeji odesílatel a typ přenášených dat; zefektivnění kódu pro zvyšování priority; změna upřednostnění určitého toku dat. Pracovní postup je koncipován tak, aby se úloha dala vypracovat během jednoho laboratorního cvičení.

Další zefektivnění funkce přepínače by mohlo spočívat ve zpětném potvrzení doručeného paketu, rozšíření počtu portů pro uživatele nebo síťové propojení přes více aktivních prvků.

SEZNAM OBRÁZKŮ

OBR. 01.	TRASA RSVP MEZI ODESÍLATELEM (SERVEREM) A PŘÍJEMCEM (KLIENTEM)	14
OBR. 02.	FAKTORY PROVOZU QoS.....	20
OBR. 03.	FORMÁT TEXTU ADRES IPv6	21
OBR. 04.	BIOLOGICKÝ NEURON.....	26
OBR. 05.	UMĚLÝ NEURON	27
OBR. 06.	MODEL PERCEPTRONU.....	28
OBR. 07.	ARCHITEKTURA DVOUVRSTVÉ PERCEPTRONOVÉ SÍTĚ	30
OBR. 08.	BIOLOGICKÁ NEURONOVÁ SÍŤ	33
OBR. 09.	UMĚLÁ NEURONOVÁ SÍŤ.....	34
OBR. 10.	VÍCEVRSTEVNÝ PERCEPTRON	35
OBR. 11.	KOHONENOVA SAMOORGANIZAČNÍ SÍŤ	36
OBR. 12.	NEURONOVÁ SÍŤ S PŘEPÍNACÍMI JEDNOTKAMI.....	37
OBR. 13.	STRUKTURA HOPFIELDOVY SÍTĚ	38
OBR. 14.	AKCE PŘI ZPRACOVÁNÍ PAKETU	44
OBR. 15.	ZAPOJENÍ TESTOVACÍ SÍTĚ.....	45
OBR. 16.	VYSÍLANÝ PAKET ZE STANICE	46
OBR. 17.	ZÁKLADNÍ HLAVIČKA IPv6	47
OBR. 18.	ZAPOJENÍ PŘEPÍNAČE.....	48
OBR. 19.	PRVEK SUBSYSTEM1	50
OBR. 20.	PRVEK SUBSYSTEM2	52
OBR. 21.	VÝSTUP SIGNÁLU Z BLOKU ADD	53
OBR. 22.	FORMÁT ADRESY IPv6	54
OBR. 23.	ZKRÁCENÝ FORMÁT ADRESY.....	54
OBR. 24.	DVOUVRSTVÁ NEURONOVÁ SÍŤ (JEDNA VRSTVA SKRYTÁ).....	55
OBR. 25.	BLOKY VRSTVY (LAYER 1).....	55
OBR. 26.	PRVEK POLE SPINACU.....	57
OBR. 27.	PŘIJATÁ DATA OD PC1	58
OBR. 28.	PŘIJATÁ DATA OD PC2	58
OBR. 29.	PŘIJATÁ DATA OD PC3	58
OBR. 30.	PŘIJATÁ DATA OD PC4	58
OBR. 31.	PŘÍCHOZÍ DATA V PC1	58

OBR. 32.	VÝBĚR VYSÍLANÝCH DAT A)STANICE 1. B)STANICE 2. C)STANICE 3. D)STANICE 4.	59
OBR. 33.	VÝBĚR ADRESÁTA A)STANICE 1. B)STANICE 2. C)STANICE 3. D)STANICE 4.....	59
OBR. 34.	DISPLEJE A) V ČASE 1. B) V ČASE 2. C) V ČASE 3. D) V ČASE 4.	60
OBR. 35.	SCHÉMA ZAPOJENÍ LABORATORNÍ ÚLOHY	64
OBR. 36.	NASTAVENÍ DÉLKY SIMULACE.....	65
OBR. 37.	VÝBĚR ADRESÁTA VE STANICI PC1	66
OBR. 38.	SUBSYSTEM1	67
OBR. 39.	KÓD PRO ZVÝŠENÍ PRIORITY	68
OBR. 40.	DETEKCE TYPU DAT V PAKETU	69

SEZNAM LITERATURY

- [1] PUŽMANOVÁ, R. Moderní komunikační sítě A-Z. Computer Press, Brno 2007.
- [2] BLUNÁR, K. a DIVIŠ, Z. Telekomunikačné siete. ŽU, Žilina 2000.
- [3] PUŽMANOVÁ, R. Širokopásmový Internet. Computer Press, Brno 2004.
- [4] SATRAPA, P. IPv6. Neocortex spol. s r.o. Jihlava 2002.
- [5] MOLNÁR, K. Aplikace umělých neuronových sítí ve vysokorychlostních aktivních síťových prvcích. Edice PhD Thesis, sv. 170, Brno 2003.
- [6] HUCABY, D. a MCQUERRY, S. Konfigurace směrovačů Cisco. Computer Press, Brno 2004.
- [7] FRANEKOVÁ, M. Modelovanie komunikačných systémov v prostredí MATLAB, SIMULINK A COMMUNICATION TOOLBOX. ŽU, Žilina 2003
- [8] NOVÁK, M a kol. Umělé neuronové sítě: teorie a aplikace. C.H.Beck, Praha 1998.
- [9] TUČKOVÁ, J. Úvod do teorie a aplikací umělých neuronových sítí. FEL ČVUT, Praha 2003.
- [10] ŠNOREK, M. Neuronové sítě a neuropočítače. FEL ČVUT, Praha 2004.