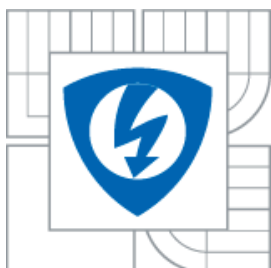




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

OPTIMALIZACE PŘEPÍNAČE V KONVERGOVANÉ SÍTI

OPTIMIZATION OF SWITCH IN CONVERGED NETWORK

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE

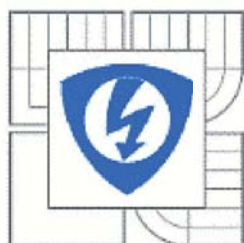
Mgr. Adam Švec, Ph.D.

AUTHOR

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. Ing. Vladislav Škorpil, Csc.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Mgr. Adam Švec, Ph.D.

ID: 115134

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TĚMATU:

Optimalizace přepínače v konvergované síti

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte problematiku přepínačů v konvergované síti s důrazem na řízení síťových prvků v moderních konvergovaných sítích. Porovnejte možné alternativy řízení. Navrhněte architekturu přepínače, která bude podporovat prioritní zpracování různých typů spojení se zaměřením na VoIP a kvalitu služby QoS. Navrhněte algoritmus přepojování vstupních požadavků na výstupy s tím, že kvalita služby QoS bude zachována. Modelování a simulaci celého síťového prvku realizujte v prostředí MATLAB-SIMULINK. Výsledky aplikujte při návrhu zadání a vzorového protokolu minimálně jedné laboratorní úlohy předmětu Služby telekomunikačních sítí.

DOPORUČENÁ LITERATURA:

- [1] PUŽMANOVÁ, R. Moderní komunikační sítě A-Z. Computer Press, Brno 2007
- [2] ODOM, W., HEALY, R., MEHTA, N. Směrování a přepínání sítí. Computer Press, Brno 2009
- [3] WENDELL, O, RUS, H., MEHTA, N. Směrování a přepínání sítí. Computer Press, Brno 2009

Termín zadání: 11.2.2013

Termín odevzdání: 29.5.2013

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

Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ANOTACE

Práce Optimalizace přepínače v konvergované síti pojednává o principech a činnosti o roli přepínače - ethernetového switche v síti. Popisuje rozdíly mezi přepínačem a rozbočovačem v síti a dopady na velikost kolizní domény.

V konvergované síti je rovněž řešeno prioritní přepínání dat podle jejich původu a tudíž i kvalitativního hlediska - jinak se budou chovat e-maily, jinak hlasové služby a jinak multimédia.

Zmíněn je reálný prvek - přepínač s podporou kvality služeb - přepínač L3. Je popsán stručně nástroj Matlab - Simulink a vytvořen funkční zjednodušený model ethernetového přepínače. Dále bylo téma použito jako námět pro laboratorní úlohu v předmětu Služby telekomunikačních sítí. Jako příloha práce je model pro Simulink a vzor laboratorního protokolu k vytvořené úloze.

Klíčová slova:

switch, přepínač, switch L3, přepínač L3, opakovač, rozbočovač, hub, koncentrátor, MAC adresa, Matlab, Simulink, simulace, model, knihovna, prioritá, síť, LAN, ISO/OSI

ABSTRACT

Switch Optimization in a Converged Network thesis discusses the role of the Ethernet switch in the network. It describes differences between the switch and the hub in the network and the impacts on the collision domain size.

In the converged network, priority data switching according to their origin and thus the qualitative point of view is also addressed - e-mails, voice services and multimedia will each behave differently.

A real element, switch L3 - a switch with services quality support - is mentioned. The tool Matlab - Simulink is briefly described and a functional simplified model of the ethernet switch was created. Furthermore, the topic was used as a theme for a laboratory task in the subject Services of Telecommunication Networks. A model for Simulink and an example of the laboratory protocol for the created task are attached to the thesis.

Keywords:

switch, hub, repeater, concentrator, MAC address, Matlab, Simulink, simulation, model, library, priority, network, LAN, ISO/OSI

ŠVEC, A. *Optimalizace přepínače v konvergované síti*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 91 s. Vedoucí diplomové práce doc. Ing. Vladislav Škorpil, CSc.

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Optimalizace přepínače v konvergované síti.“ 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/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 29. 5. 2013

.....
podpis autora

Poděkování

Děkuji vedoucímu práce panu Doc. Ing. Vladislavu Škorpilovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce. Děkuji rovněž člověku, který mě při jejím psaní podporoval – mé milované ženě Hance.

V Brně dne 29. 5. 2013

.....
podpis autora

Cíle, kterých má být dosaženo:

Prostudujte problematiku přepínačů v konvergované síti s důrazem na řízení síťových prvků v moderních konvergovaných sítích. Porovnejte možné alternativy řízení. Navrhněte architekturu přepínače, která bude podporovat prioritní zpracování různých typů spojení se zaměřením na VoIP a kvalitu služby QoS. Navrhněte algoritmus přepojování vstupních požadavků na výstupy s tím, že kvalita služby QoS bude zachována. Modelování a simulaci celého síťového prvku realizujte v prostředí MATLAB-SIMULINK. Výsledky aplikujte při návrhu zadání a vzorového protokolu minimálně jedné laboratorní úlohy předmětu Služby telekomunikačních sítí.

Obsah

1 Úvod.....	7
2 Popis konvergované sítě	9
2.1 Referenční model ISO/OSI a začlenění aktivních prvků sítě do něj.....	9
2.2 Topologie sítě.....	17
3 Aktivní prvky v síti.....	19
3.1 Rozbočovač – hub.....	19
3.2 Přepínač – switch.....	21
4 Kvalita služeb.....	23
5 Reálný přepínač pro Ethernet	24
5.1 Přístupová metoda CSMA/CD.....	24
5.1.1 Funkce a činnosti reálného přepínače	26
5.1.2 Bufferování přepínače.....	27
6 Simulace přepínače v prostředí Matlab Simulink.....	29
6.1 Založení nového modelu a vytvoření typického zdroje dat	31
6.1.1 Zdroj dat	31
6.1.2 Zásady vytváření modelu přepínače.....	34
6.2 Komponenty modelu přepínače	35
6.2.1 Popis zapojení.....	36
6.2.2 Popis modelu.....	36
6.2.3 Nastavení simulace.....	46
6.2.4 Výsledky simulace.....	47
6.2.5 Východiska pro laboratorní úlohu.....	60
7 Laboratorní úloha.....	61
8 Závěr.....	84
9 Seznam obrázků a tabulek.....	86
10 Seznam literatury.....	88

1 Úvod

V soudobých počítačových sítích se vedle kvalitních kabeláží neobejdeme bez spolehlivých aktivních prvků. Propustnost sítě je dána nejenom jejími elektronickými či optoelektronickými vlastnostmi, ale při stále se zvětšujícím objemu transferovaných dat i způsobem, jak tato data proudí vlastní síťovou infrastrukturou. V počátcích počítačových sítích byl základním médiem 50 Ω koaxiální kabel, zakončovaný terminátory s charakteristickou impedancí vedení a BNC konektory ve formě odboček k počítačům. Později se toto řešení ukázalo jako nevýhodné nejenom z hlediska fyzikálních možností takto koncipované sítě, ale i z hlediska stochastických vlastností celého takto pojatého Ethernetu – v současné době dominujícímu typu datových i konvergovaných sítí.

Vzhledem k přístupové metodě CSMA/CD se totiž s nárůstem počtu stanic zvyšuje pravděpodobnost kolize sítě a řešení těchto kolizí není již možno nechat jen na schopnostech síťového rozhraní koncových uzlů – serverů, stanic, popřípadě směrovačů. Taková síť by se při konverzi provozovaných služeb stala relativně brzy zcela zahlcenou, a tudíž nepoužitelnou.

Nasazení strukturované, elektricky symetrické kabeláže a aktivních prvků sice podstatně snížila při výpadku některé koncové komponenty riziko havárie celé sítě, nicméně hlavní problém zůstal: rozbočovače (huby) nebyly schopny zmenšit kolizní doménu („ucpaný přenosový kanál“) nijak ji odstínit od ostatních částí sítě. Hub je v podstatě elektronická rozbočka – opakovač, či digitální zesilovač signálu, jehož jedinou inteligencí je, že nepustí zopakovaný signál do *vlastního portu* (tvořeného standardizovaným rozhraním RJ45. Pokud navíc v síti existoval zdroj šumu či brumu, pak se toto rušení s užitečným signálem rozšířilo prakticky na celou síť. Toto řešení, založené na fyzické vrstvě ISO/OSI je tedy vhodné jen na relativně malém prostoru, čítajícím několik pracovních stanic, jeden server, na kterém ale nesmí běžet mnoho rozdílných služeb. Kvalitativně i proudící data musí být podobná, takže princip QoS (kvalita služeb) je v tomto případě diskutabilní. Toto omezení, a tím zvýšení propustnosti sítě, odstranily až aktivní prvky, které rozdělují síťový provoz podle sofistikovanějších pravidel. Tyto prvky pracují na linkové vrstvě ISO/OSI a nazývají se souhrnně *přepínače*.

Přepínač (switch, mikroswitch...) je zařízení, které v Ethernetu přebírá na sebe zodpovědnost za provoz na lince a data tak do jisté míry filtruje provoz tak, aby na sebe „viděly“ jen ty části sítě, které tak potřebují učinit a tím nebrzdily provoz celé sítě či segmentu. Samozřejmě – jiná je situace při vysílání bod proti bodu a jiná při vysílání bod proti multibodu (multicast), ale základní myšlenka je stejná – komunikují jen ti, kteří komunikovat chtějí. Z tohoto důvodu stejně robustní síť přenesla větší objem dat, neboť kolizní doména se sníží – vytvoří se více dvojic či skupin komunikujících

koncových bodů, které se navzájem neruší a nebrzdí další provoz.

Základní údaj, se kterým na linkové vrstvě přepínač pracuje, je fyzická adresa (MAC). Ta dovolí přepínacímu algoritmu zmapovat situaci a podle potřeby přepnout provoz do aktuálních portů. Dalším rysem přepínačů je to, že v konvergované síti některé typy dovedou pracovat mezi linkovou a síťovou vrstvou, a tím již (třebaže v omezené míře) implementovat kvalitu služeb.

Diplomová práce se zabývá popisem činnosti přepínače v konvergované síti a ukázkou prostředí Matlab Simulink firmy MathWorks Inc. V prostředí Matlab Simulink bude navržen zjednodušený model přepínače – ethernetového switchu a popsán vhodný algoritmus přepínání. Výsledky simulace poté poslouží jako podklady pro laboratorní (počítačové) cvičení ze Služeb telekomunikačních sítí.

2 Popis konvergované sítě

Často bývá budována jako podniková, v poslední době rovněž domácí, neboť si takové sítě začínají budovat i lidé ve svých domech či bytech. Zahrnuje koncová zařízení, kabeláž a aktivní prvky, které se starají o propojení koncových zařízení, distribuci signálu po budově, domě, bytě či kampusu a jednoduché či vícenásobné propojení do metropolitní sítě.

Typickým rysem konvergované sítě je *mix uživatelských služeb*, které jsou nabízeny „technologii jednoho drátu“. Může to být kabelová televize, VoIP telefon a připojení k Internetu, popřípadě další služby, jako například zabezpečení objektů – napojení alarmů na pult centrální ostrahy, sledování kamerovým systémem, apod.

V takto pojaté síti je nezbytné respektovat prioritu zpracování dat. V síti Ethernet, která v současnosti dominuje ve světě sítí, je tento požadavek zajištěn metodou kvality služeb QoS, která rozhoduje o prioritě síťového provozu. Rovněž protokolová sada TCP/IP je dominantní. Všimněme si proto, jak je kvalita služeb implementována s ohledem na vrstvý model ISO/OSI a jak tento model souvisí s implementací konkrétní protokolové sady TCP/IP.

2.1 Referenční model ISO/OSI a začlenění aktivních prvků sítě do něj

Referenční model ISO/OSI vypracovala *Mezinárodní organizace pro normalizaci (ISO)* jako hlavní část snahy o standardizaci počítačových sítí nazvané OSI (**Open Systems Interconnection**) a v roce 1984 ho přijala jako normu **ISO 7498**. Kompletní text normy přijala také CCITT jako **doporučení X.200**. Referenční model ISO/OSI se používá jako názorný příklad řešení komunikace v počítačových sítích pomocí vrstvého modelu, kde jsou jednotlivé vrstvy *nezávislé a snadno nahraditelné*.

Posláním tohoto modelu je poskytnout praktický návod a metodiku vypracování systémů propojování. Model je tedy standardem, od kterého by se principy propojování měly odvíjet. Je modelem abstraktním a do jisté míry idealizovaným – v podstatě jeho doslovnou interpretaci žádný reálný systém plně neimplementuje. Jsou uvedeny pouze obecné principy, které je nutno při návrhu komunikačních systémů dodržet, a to proto, aby tyto systémy, byť mnohdy odlišně pojaté, mohly v reálném světě spolupracovat. Z toho důvodu zde také není zmínka o konkrétní protokolové sadě; to je již věcí autorů této sady.

Jednotlivé vrstvy implementují vlastní algoritmy a mezi sebou komunikují přes rozhraní

API. Je zde tedy použit jakýsi objektový princip zapouzdření – jedna vrstva se nestará, jak konkrétně je implementována činnost jiné vrstvy. Využívá pouze služeb této – sousední – vrstvy, a to právě prostřednictvím API. To dovoluje jednotlivé části modelu programovat odděleně a rovněž standardizovat komunikaci jednotlivých vrstev. Například při výměně síťového rozhraní, routeru, switche (přepínače), apod. je možno ponechat stávající komponenty beze změny, neboť je zaručeno, že se opět „domluví“.

Tyto vrstvy se nazývají takto:

- *Fyzická*
- *Linková*
- *Síťová*
- *Transportní*
- *Relační*
- *Prezentační*
- *Aplikační*

Podívejme se na tyto vrstvy (počítáno zezdola) poněkud podrobněji:

Fyzická vrstva (physical layer)

Je to první a nejnižší vrstva referenčního modelu. Stará se o to, aby datový tok byl převeden na elektrický nebo optický signál a byl správně předán fyzickému médiu (koaxiální kabel, kroucená dvojlinka, světlovod...). U komutovaných spojů – přepínání okruhů – rovněž fyzicky spíná a rozpojuje tyto okruhy. Fyzické médium podle ISO/OSI do této vrstvy nepatří, je však s ní spojeno přinejmenším těsně; mechanické provedení konektorů a kabelů a jejich elektrické či optické parametry jsou nezáměnné a proto s touto vrstvou přímo souvisejí.

Vrstva pracuje se *signálem*, její realizace je tedy výhradně hardwarová. Samozřejmě – tento hardware má své ovládání (firmware). Protokoly a normy fyzické vrstvy stanovují přesné parametry těchto signálů.

Mezi **hardware této vrstvy** typicky patří:

- Rozbočovač (hub)
- Opakovač (digitální zesilovač a tvarovač signálu)
- Modem
- Síťová karta
- Různé kmitočtové filtry

Linková vrstva (data link layer)

Linková vrstva (též někdy zvaná spojovací vrstva) je druhá vrstva ISO/OSI. Poskytuje spojení mezi dvěma nebo více systémy zapojenými na téže lince (například datová propojka dvěma přepínači nebo přepínačem a koncovým zařízením). Tato linka může být dvoubodová (sériová linka) nebo mnohobodová (Ethernet). Data uspořádává do rámců (frames), detekuje a opravuje kódové chyby a stará se o přenos těchto rámců.

Mezi typické **hardwarové komponenty** této vrstvy je v konvergované síti řazen **přepínač (switch)** a do jisté míry i síťové rozhraní, neboť v praktické implementaci bývají fyzická i linková vrstva těsně spojeny.

Linková vrstva rovněž řeší topologii sítě, směr provozu (simplexní, poloduplexní nebo duplexní) a také typ spojení (například bod proti bodu nebo bod proti vícenásobnému bodu).

Zavádí se pojem **fyzická adresa** (MAC adresa – podle Media Access Control podvrstvy – integrální a v podstatě hlavní části této vrstvy). Fyzickou adresu v tomto pojetí má jakékoli síťové rozhraní. Je to jedinečný a unifikovaný kód, 48 bitové hexadecimální číslo. Jak je známo, toto číslo jde klonovat – tedy vnutit zařízení námi definovanou hodnotu. Dá se to ovšem přirovnat ke „hře s ohněm“, neboť poslání tohoto údaje je podobné rodnému číslu člověka. MAC adresa je multiplatformní – multiprotokolární záležitost. Hraje rovněž bezpečnostní roli – síť má právo odmítnout zařízení s některými hodnotami MAC a naopak vyžadovat pouze předem vyjmenované hodnoty MAC. Jestliže tento princip jakkoli zpochybníme, nemůžeme pak bezpečnostní algoritmy založené na tomto principu použít.

V této práci bude podrobně diskutována činnost přepínače, proto si všimněme této vrstvy a MAC poněkud podrobněji:

MAC podvrstva tedy zajišťuje:

- fyzické adresování,
- řízení přístupu k médiu
- je hardwarově závislá

Co se týče přístupu k médiu, rozeznáváme **deterministický přístup** – lze určit maximální časový interval, ve kterém se pracovní stanice dostane k médiu. Zde je definován token (pešek – známka), který uděluje „slovo“ - řídí komunikaci, moderuje ji. Jestliže se token ztratí, popřípadě by se tokenů objevilo více, vypuká chaos, proto zde existují mechanismy, jak token předávat. Monitoring tokenu je obvykle zvolen dohodou koncových bodů (Token passing). Rovněž lze token předávat řízeně,

pak ovšem musí existovat řídicí systém (Token polling). Selhání tohoto systému je rovněž pro komunikaci fatální, proto je považován za nejslabší článek tohoto principu.

Naproti tomu **stochastický – nedeterministický přístup** je vůči takovému selhání imunní. Zde žádný token není, ovšem koncové body musí čekat na uvolnění přenosového kanálu – média. *Sem patří i Ethernet a WiFi, nejčastější používané komunikační technologie současnosti.* Nelze zaručit konkrétní časový interval a v podstatě ani jeho šíři, takže koncové body se dostávají „ke slovu“ pseudonáhodně. Aby nedošlo k zahlcení sítě tohoto typu, která je vůči hustotě provozu velice citlivá, používají se přístupové metody, které přístup k médiu zajistí poněkud uspořádanějším přístupem.

Aloha je nejjednodušší metoda, jejíž název byl odvozen z havajského pozdravu (*Aloha! Nazdar!*). Tento princip byl poprvé použit v roce 1971 na Havaji k řízení radarové sítě. Každý koncový bod, pokud chce vysílat, tak také učiní. Důležitou podmínkou je, že prodleva mezi následnými pokusy o vysílání je náhodná. Metoda je neobyčejně prostá, ovšem z hlediska zahlcení sítě si svojí podstatou o ni přímo říká. Přenosové médium se využije tak na 15%, jednotlivé rámce se často – až 4x musí opakovat. Zátěž sítě prudce narůstá a její propustnost rychle klesá.

Těžká zahlcení i běžným provozem byla důvodem, že se informatici snažili tuto metodu vylepšit a optimalizovat. Koncové body tedy vysílají jen v předem definovaných intervalech – *slotech*. Nazývá se to *slotová aloha*. Dosáhlo se sice zkrácení délky kolizního slotu, avšak hlavní problém zůstal – síť se i bez kolizí při jisté zátěži dostávala do nepropustného stavu, ze kterého se bez vnějšího zásahu (restartu) nebyla schopna dostat. Dalším „vylepšením“ bylo, že po jisté době, když se zjistilo, že rámec byl špatně odvysílán, se koncové zařízení pokusilo v náhodném časovém okně rámec vyslat znovu. Pro další zvyšování síťového provozu však metoda dosáhla svých limitů, a proto se v konvergovaných sítích používají metody sofistikovanější.

CSMA/CD (Carrier Sense Multiple Access/Collision Detection) je metodou, která se používá v síti Ethernet. Bude rovněž vzhledem k tématu práce rozebrána podrobněji i dále. Její podstatou je, že pokud koncový bod, který chce vysílat, naslouchá na síti a když žádný jiný zdroj dat nevysílá, teprve začne vysílat. Zahájí-li vysílání v témže okamžiku i jiný zdroj, dojde samozřejmě ke kolizi. Při detekci kolize stanice vyšle kolizní signál *jam* („zmuchlání“). Ostatní zdroje dat po obdržení tohoto signálu obdržený rámec zahodí. Vysílající zdroj přestane na náhodnou dobu (aby nedošlo k další kolizi) vysílat. Aby kolizi zaznamenala všechna zařízení na lince, je stanovena *minimální délka rámce*. *Pro klasický Ethernet je minimální délka rámce 64 oktetů (k tomu je potřeba započítat 7 oktetů preamble a 1 synchronizační oktet).* Protože rychlost šíření signálu je konečná, požadavek, aby kolizi zaznamenala všechna zařízení na lince, omezuje fyzické rozměry linky – definuje se kolizní doména. Typickým příkladem je část sítě typu Ethernet, ve které se používají

pouze rozbočovače). Tyto kolize jsou normální funkční záležitostí této přístupové metody. Při velkém počtu stanic jejich četnost stoupá, takže průchodnost linky se zmenšuje. Pokud dojde ke kolizi později než po minimální délce rámce, jedná se o tak zvanou pozdní kolizi (late collision), která buď znamená, že velikost kolizní domény překračuje povolenou mez, nebo že některé zařízení nedodržuje „pravidla“, nebo je to příznakem hardwarové chyby, typicky nepřizpůsobené vedení.

CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) – metoda předcházení kolizím. Tato metoda je použita u technologie WiFi. a bezdrátových sítí vůbec. Zde detekce kolize totiž není technicky možná. U WiFi je postup následující:

Je-li médium volné po určenou dobu, může zdroj zahájit vysílání. Pokud je vysílání neúspěšné (druhá strana nepotvrdí příjem), zahájí se *exponenciální čekání*. Exponenciální čekání znamená odložený pokus o vysílání. Zdroj si náhodně vybere dobu z intervalu, jehož velikost se během opakovaných pokusů *zdvojnásobuje*. Tuto náhodnou dobu čeká a zároveň sleduje, zda nezačal vysílat někdo jiný. Pokud ano, exponenciálně zvětšuje dobu čekání. Zůstalo-li médium volné, odvysílá rámeček. Jestliže vysílání neuspěje, opakuje se exponenciální čekání. Náhodný výběr čekací doby z rychle rostoucího intervalu má za cíl snížit pravděpodobnost, že kolidující stanice spolu budou při dalším pokusu kolidovat znovu.

Toto všechno musí linková vrstva zajistit. Další její úkol je opravovat a detekovat chyby, což se děje částečně už v metodách přístupu.

Síťová vrstva (network layer)

Tato vrstva se stará o směrování v síti a síťové adresování. Poskytuje spojení mezi systémy, které spolu přímo *nesousedí*. Obsahuje funkce, které umožňují *překlenout rozdílné vlastnosti technologií v konvergovaných sítích*. Ve světě TCP/IP se zde pracuje s protokolem (sadou protokolů) IP. Definiuje *IP adresu*, a to jak ve verzi 4 (32 bitové dekadické číslo oddělené tečkou po oktetech), tak i ve verzi 6 (128 bitové hexadecimální číslo oddělené dvojtečkou po dvojicích oktětů). Tato vrstva je již softwarová – abstrahuje od implementovaného hardwaru a definuje vlastní „pravidla“, závislá od použité protokolové sady. Je-li protokolová sada směrovatelná (a to v konvergované síti by měla být vždy), pak se zajistí toto směrování. Typickým zařízením pracujícím na této vrstvě je **směrovač (router)**. Tato vrstva již *kvalitativně hodnotí data* dle jejich typu a původu a zajišťuje *kvalitu služeb QoS*, a to zejména v routerech. Je samostatnou vrstvou nejenom v teoretickém modelu, ale i v konkrétních protokolových sadách – zejména v TCP/IP. Aby toto hodnocení mohlo probíhat smysluplným způsobem, jsou data oklasifikována příznakem ToS (v TCP/IP).

Transportní vrstva (transport layer)

Tato vrstva se stará o přenos dat. V sadě TCP/IP je opět vrstvou samostatnou. Vrstva umožňuje adresovat přímo aplikace a pracuje s adresou zvanou **port**. S touto vrstvou jsou spojené **firewally**, založené na technologii *paketového filtru*. Vrstva poskytuje transparentní, spolehlivý přenos dat s požadovanou kvalitou. Vyrovnává různé vlastnosti a kvalitu přenosových sítí. Provádí převod transportních adres (portů) na síťové (IP), ale *nestará se o směrování* – to je úkolem vrstvy síťové.

V TCP/IP jsou definovány dvě velké skupiny protokolů, a to TCP (Transmission Control Protocol) a UDP (User Datagram Protocol). TCP je spolehlivý (potvrzovaný) transportní protokol, který je použit všude tam, kde se nesmí z dat nic ztratit. Je sice pomalejší (o potvrzení), ale systém má vždy zprávu, v jakém stavu a zda vůbec data k cíli dorazila. Vyžaduje se u služeb jako e-mail, WWW, FTP, apod. *Hlavním kritériem je spolehlivost přenosu, rychlost je druhořadá*. Naproti tomu UDP (User Datagram Protocol) je nespolehlivý (nepotvrzovaný) transportní protokol, který je implementován tam, kde je *prvořadá rychlost přenosu*. Jde o veškerý přenos v reálném čase, zejména telefonní hovory, multimediální datové proudy, ale i takové „servisní“ záležitosti jako DNS.

Relační vrstva (session layer)

Úkolem této vrstvy je organizovat a synchronizovat dialog mezi spolupracujícími relačními vrstvami obou systémů a řídit výměnu dat mezi nimi. *V TCP/IP není vrstvou samostatnou, je nejnižším článkem tří posledních vrstev a TCP/IP ji přisuzuje aplikacím*. Vrstva (v teoreticky pojaté samostatné entitě) umožňuje vytvoření a ukončení *relačního spojení - session*, synchronizaci a obnovení spojení, oznamování výjimečných stavů, apod. Zde bychom se mohli ilustrativně zmínit o **Cookies**, známých ze světa WWW. Relace a její stav umožňuje nastavit například odlišné prostředí pro různé uživatele sdílené aplikace. Často relace využívají aplikace založené na PHP technologii a databázi MySQL (například e-learningový systém Moodle vždy při přihlášení uživatele vytvoří relaci (session), kterou si ukládá do pomocného adresáře na serveru ve formě souboru).

Prezentační vrstva (presentation layer)

Posláním této, předposlední vrstvy modelu ISO/OSI je *transformovat data do tvaru, který používají aplikace*. Toto musíme vysvětlit poněkud podrobněji. Každá aplikace vyžaduje smysluplná a ucelená data, která pak svými algoritmy zpracovává. Aplikaci nezajímá, jak probíhal

přenos, tedy „jaká byla cesta“. K aplikaci v současnosti přicházejí často šifrovaná data. A právě **šifrování/dešifrování a vlastně celé zdrojové kódování a úprava dat je úkolem této vrstvy**. Zde pracují algoritmy šifrování, šifrovací a dešifrovací klíče, konverze, komprese, expanze, atd. Vrstva data přizpůsobí rovněž pro přenos – provede přizpůsobení pro přijetí nižšími vrstvami, zejména vrstvou transportní (ale i relační, protože ta leží přímo pod ní). V TCP/IP je úkol této vrstvy opět svěřen aplikacím, takže není vrstvou samostatnou.

Aplikační vrstva (application layer)

Jde o nejvyšší, uživatelsky „viditelnou“ vrstvu modelu ISO/OSI. Zajímavé je její pojetí v ISO/OSI a její skutečná implementace v TCP/IP. Zatímco v teoretickém modelu je úkolem vrstvy jenom poskytnout aplikacím přístup ke komunikačnímu systému a umožnit tak jejich spolupráci, **v TCP/IP jde o robustní část aplikace, sdružující funkce tří vrstev do jedné**. Proto označujeme za aplikační protokoly v TCP/IP i takové, které vlastně ryze aplikačními nejsou – suplují tak funkci sloučených tří vrstev dohromady. Tyto protokoly jsou všeobecně známy; často jejich názvy figurují v URL, kterým voláme službu. Uvedme si namátkový výčet:)

- AIM (AOL Instant Messenger Protocol)
- DHCP (Dynamic Host Configuration Protocol)
- FTP (File Transfer Protocol)
- HTTP (Hyper Text Transfer Protocol)
- IMAP (Internet Message Access Protocol)
- IRC (Internet Relay Chat)
- SILC (Secure Internet Live Conferencing)
- LDAP (Lightweight Directory Access Protocol)
- NNTP (Network News Transfer Protocol)
- SSH (Secure Shell)
- SIP (Session Initiation Protocol)
- SMTP (Simple Mail Transfer Protocol)
- SNMP (Simple network management protocol)

ISO/OSI		TCP/IP
APLIKAČNÍ VRSTVA		APLIKAČNÍ VRSTVA
PREZENTAČNÍ VRSTVA		
RELAČNÍ VRSTVA		
TRANSPORTNÍ VRSTVA		TRANSPORTNÍ VRSTVA
SÍŤOVÁ VRSTVA		SÍŤOVÁ VRSTVA
LINKOVÁ VRSTVA		FYZICKÁ A LINKOVÁ VRSTVA
FYZICKÁ VRSTVA		

Obr.1: Srovnání modelů ISO/OSI a TCP/IP – odpovídající vrstvy

2.2 Topologie sítě

Sebelépe optimalizovaná a navržená konvergovaná síť nemůže garantovat kvalitní služby, pokud jsou zdroje dat nevhodně připojeny. Mnohdy je to limitováno fyzickými podmínkami, kde se síť nachází – například těžko lze požadovat připojení malého korálového ostrůvku pevnou optickou linkou, která by musela být speciálně tažena podmořským kabelem, navíc pokud je na tomto území stěží zajištěna výroba a distribuce potřebné silové elektřiny. Tento případ se optimálně vyřeší rádiovým nebo satelitním spojením, neboť průmyslové rušení zde bude minimální. Naopak v mnohamilionovém velkoměstě s masivním elektrickým smogem na pozadí budou bezdrátové technologie náhodně chybovat a tyto chyby navíc budeme velice těžko hledat.

Lokální sítě mají různé topologie – jejich přehled lze najít v každé lepší příručce o sítích. Ukažme si proto základní fragment soudobé sítě založené na technologii Fast Ethernet a jeho fyzickou i logickou topologii:

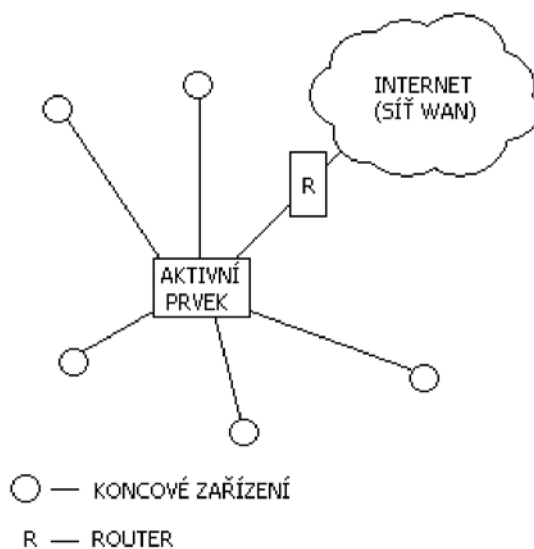
O *fyzické topologii* se dá hovořit zejména na nejnižší úrovni zasíťování, tedy zpravidla v sítích LAN, z hlediska telekomunikačního v případě sítí přístupových. Ale i v sítích transportních a páteřních, respektive v oblasti MAN a WAN lze vysledovat fyzické zapojení až do konkrétních komponent, které síť tvoří. *Fyzická topologie je skutečný obraz toho, jak je síť zapojena.*

Základní fyzickou topologií v dnešní době je **hvězda**, která je propojena dále **uzlinově**; jednotlivé hvězdy jsou propojovány mezi sebou různými médii a technologiemi. Propojení nemusí být pouze jedno. Pomocí směrovačů a bran lze propojení zdvojit či znásobit tak, aby při výpadku jedné cesty bylo možno jít „náhradní cestou“. Na konkrétní technologii pak záleží, zda jsou aktivní prvky sítě propojeny rovnocenně (peer-to-peer), jako je tomu například u paketových sítí Fast Ethernet či s nějakou hierarchií (kupř. stromově), jako například u sítí 100VG-AnyLAN.

U sítí Fast Ethernet a Gigabit Ethernet je na nejnižší úrovni před aktivními prvky či koncovými zařízeními základním propojovacím médiem v topologii hvězdy kroucená dvojlinka v nestíněném nebo stíněném provedení, opatřena konektory RJ-45. Je to osmižilový symetrický kabel, kde jsou 4 páry celistvých vodičů nebo lanek vodičů vzájemně zkrouceny a svojí polohou zajišťují vysokou odolnost před průmyslovým rušením. Na tomto typu vedení, byť je velmi jednoduché, dosahujeme velikých přenosových rychlostí – testují se desítky Gbps; běžně na kabelu kategorie 6 (na menších vzdálenostech i kategorie 5e) se dosahuje přenosu řádů jednotek Gbps. Pro univerzálnost použití je *kroucená dvojlinka* nejrozšířenějším typem kabelu ve slaboproudých

rozvedech a pokud pomineme dálkové metalické telekomunikační kabely, které se mnohdy nahrazují optickými vlákny, dá se s ní realizovat téměř jakýkoli slaboproudý rozvod po budově či kampusu.

Logická topologie je schéma, jak sítí proudí data. V principu se od fyzické topologie může lišit, kupř. síť Token Ring je fyzicky hvězda, avšak aktivní prvek – koncentrátor z ní tvoří logický kruh. Sítě typu Ethernet mají fyzickou i logickou topologii prakticky shodnou. Vzhledem k nedeterministickým vlastnostem přístupových metod této sítě (CSMA/CD, popř. CSMA/CA u WiFi) se jeví všechny komponenty zapojeny paralelně. Topologie hvězda je otevřená a odolná proti výpadku jednotlivých větví. Aktivní prvky (switche, routery, brány) řídí provoz tak, aby nedocházelo ke kolizím či zahlcování sítě a rovněž při poruše logicky odpojí poškozenou či nefunkční část sítě.



Obr.2: Postavení aktivního prvku v segmentu sítě. Aktivním prvkem je v současnosti nejčastěji přepínač – switch. Buď transparentní nebo s podporou kvality služeb (L3 switch)

3 Aktivní prvky v síti

Každá síť – to jsou především vysokorychlostní datové spoje. Avšak každý drát vede odněkud někam, navíc signál ve formě elektrického proudu trpí útlumem, fázovým posuvem, odrazy, rušením... V síti proto existují i zařízení, která se starají o správný chod a řízení přenosové části tak, aby data správně dorazila od zdroje k příjemci. Dá se říci, že tato zařízení vlastně síť definují. Souhrnně jim říkáme *aktivní prvky*.

Na obrázku 2 máme zachycen fragment sítě s hvězdovou topologií. Aktivním prvkem, který propojuje koncová zařízení, je buď *rozbočovač (hub)* nebo *přepínač (switch)*. Podle toho, který z těchto prvků použijeme, budou v daném segmentu proudit data. Každý z těchto prvků se totiž chová jinak a ovlivňuje nejenom přítomnost dat na médiu, ale i do jisté míry propustnost síťového segmentu a tím i kvalitu a dostupnost používaných služeb.

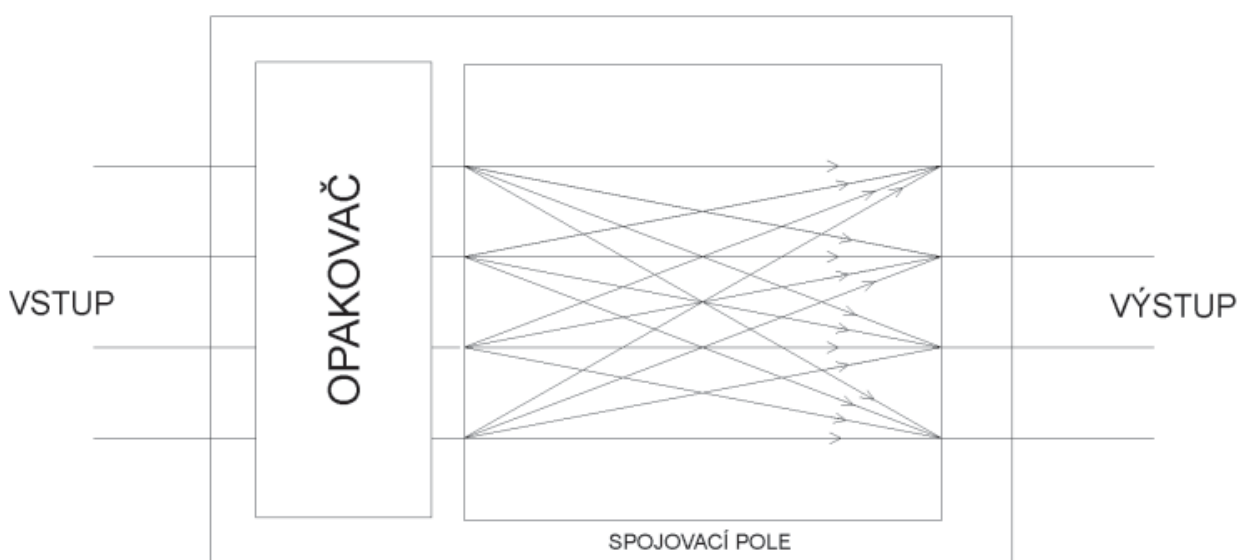
Každý aktivní prvek má působnost na určitou vrstvu modelu ISO/OSI. Kupříkladu směrovač pracuje se síťovou vrstvou, opakovač a rozbočovač s fyzickou vrstvou a přepínač s linkovou vrstvou. Dále si uvedeme princip přepínače, jednotlivé jeho části popíšeme a srovnáme jeho činnost s rozbočovačem.

3.1 Rozbočovač – hub

Rozbočovač je základním aktivním prvkem v síti s hvězdovou topologií. Pracuje na fyzické vrstvě ISO/OSI a z hlediska tekoucích dat je proto transparentní. Teoreticky vzato – kdybychom na libovolný port rozbočovače poslali *jakýkoli* digitální signál, měl by jej pouze zkontrolovat tvarově a zopakovat na ostatní porty. Jeho jedinou „inteligencí“ je to, že na příchozí port signál nedistribuuje. Dalším jeho posláním je impedanční přizpůsobení vysokofrekvenčního vedení a tím zabránění odrazům na tomto vedení.

Rozbočovač je velmi jednoduché zařízení. Nijak neřídí provoz, který skrz něj prochází. Signál, který do něj vstoupí, je obnoven a vyslán všemi ostatními porty. Zpoždění je proto pouze 1 bit, takže na rozdíl od switche způsobuje hub nižší latenci. Na hubu jsou typicky signalizační LED diody, podle kterých se dá snadno zjistit vadné spojení.

Síť opatřená rozbočovači je velice snadno zahlitelná, protože všechna koncová zařízení jsou vlastně spojena paralelně. Z jednoho portu vede propojka na směrovač do další sítě - ven. Proto, aby mohl Ethernet detekovat kolize, je počet rozbočovačů v síti omezen. Pro síť 10 Mbps je počet segmentů omezen na 5, tedy 4 rozbočovače mezi dvěma koncovými stanicemi. Pro Fast Ethernet (100 Mbps) je limit snižen na 3 segmenty, tedy 2 rozbočovače. Gigabitový Ethernet rozbočovače nepoužívá vůbec. Dalším podstatným rysem použití rozbočovače je to, že celý síťový segment opatřený rozbočovači může pracovat jen s jednou přenosovou rychlostí. To znamená, že pokud do čistě 100 Mbps sítě připojíme byť jediné koncové zařízení pracující s rychlostí 10 Mbps, pak všechna ostatní síťová rozhraní se této situaci přizpůsobí – přepnou se automaticky do režimu přenosu 10 Mbps. To se může stát brzdou v rozsáhlejší podnikové síti, kdy skupina mnohportových rozbočovačů tuto situaci vnutí celé síti. Rovněž, pokud některé nekvalitní síťové rozhraní šumí nebo je průmyslově rušeno, pak se přes rozbočovač zašumí celý segment sítě. Stejná situace principiálně nastane i při zkratu datového kabelu nebo v zásuvce či v domovním datovém rozvodu, i když modernější a dražší rozbočovače tuto situaci dokázaly samy vyřešit softwarovým vypnutím provozu na poškozeném segmentu. V minulosti byly podobné závady příčinou těžkých poruch celých sítí, navíc jsou obtížně vyhledatelné. Uživatel šumícího zařízení se do sítě může připojovat náhodně a administrátoři pak marně a dlouhodobě závadu hledali postupným odpojováním částí sítě a četnými restarty rozbočovačů. Z těchto a mnoha dalších důvodů se rozbočovače dnes ani nevyrábějí, protože jsou nahrazeny inteligentnějšími prepínači pracujícími na linkové vrstvě ISO/OSI.



Obr.3: Blokové schéma rozbočovače - hubu

3.2 Přepínač – switch

Pracuje na linkové vrstvě ISO/OSI. Jeho posláním je v podstatě totéž, co dělá rozbočovač – zopakovat signál a rozdělit jej na výstupní porty, ale již poněkud jiným způsobem. Linková vrstva umožňuje kvalitativně rozlišit tekoucí data – rámce, a to podle fyzické adresy (MAC). Tato adresa je, jak jsme uvedli, jedinečná pro každé síťové rozhraní. Podle toho, kde se požadované cílové zařízení nachází, přepínač propojí vstup s požadovaným výstupem – tedy bod proti bodu. V ostatních zásuvkách se tato (nepotřebná) data neobjeví. To má za následek výrazné snížení provozu v přímo nekomunikujících segmentech sítě a tím vlastně zvyšujeme determiničnost sítě (vzdor tomu, že celkově Ethernet je sítí stochastickou).

Multiplexer (rozdělovač datového toku) v přepínači je řízen algoritmem pracujícím s MAC adresami. Každé koncové zařízení připojené k přepínači se identifikuje svojí MAC adresou a tudíž přepínač okamžitě ví, kam datový provoz má poslat.

V podstatě existují tři základní typy těchto algoritmů:

- Store and forward (Ulož a poté přepni)
- On the fly (Přepínej za letu)
- Adaptive switching (Načítej a průběžně přepínej dle situace)

Nyní k jednotlivým přepínacím postupům podrobněji:

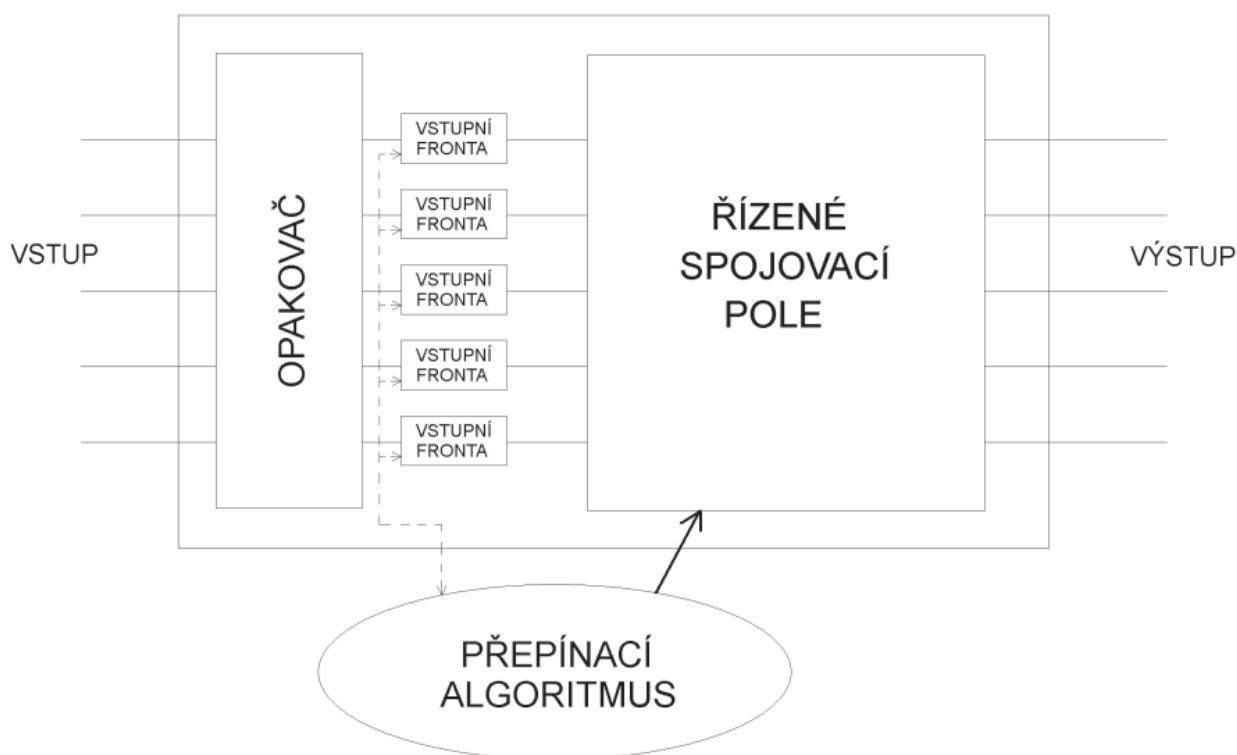
Store and Forward (Ulož a poté přepni) je klasický přepínací algoritmus, typický pro realizaci první generace přepínačů. Tento postup je použitelný nejenom v Ethernetu a spočívá v postupu, kdy přepínač načte do vstupní fronty celý rámec (paket) a provede kontrolu jeho konzistence. Pokud není konzistentní, dojde k zahození a znovunačtení. Pokud je vše v pořádku, pak dojde k porovnání cílové MAC rámce s MAC v tabulce přepínání, kterou přepínač neustále aktualizuje a podle toho, co má skutečně připojeno, pak to také propojí. Vzhledem k rychlosti, se kterou pracují dnešní přepínače, nemusí to být nutně na závadu, i když – obecně vzato – je tento algoritmus nejpomalejší, avšak také nejspolehlivější.

On the fly (Přepínej za letu). Tato metoda je naopak nejrychlejší, neboť odpadá kontrola konzistence rámce. Po identifikaci cíle je zdroj s cílem propojen – takřka v reálném čase. Tím se minimalizuje zpoždění, zvyšuje se propustnost sítě, avšak kontrola je v režii koncového (cílového) bodu. Hodí se proto pro sítě, které příliš nechybují nebo kde jde o přenos s vyššími nároky na rychlost, tedy ve finále pro data s vyššími hodnotami QoS.

Adaptive switching (Načítej a průběžně přepínej dle situace). Soudobé přepínače jsou schopny

zběžně zkontrolovat rámec a pokud nějakým způsobem zjistí, že jde o prioritu, která „nesnese odklad“, pak přepnou „za letu“, jinak přepínají po rámcové kontrole. Rovněž se přepnou do režimu s kontrolou tehdy, jestliže zjistí zvýšenou chybovost dat (zvýšené požadavky na opakované přenosy). Je to kombinace předchozích postupů. Tento algoritmus je nejsostifikovanější a dokáže se adaptovat na situaci v síti podle chybovosti dat.

Přepínací algoritmy v celkové implementaci tvoří firmware přepínače a je know-how jednotlivých výrobců – v podstatě lze říci, že čím dražší přepínač, tím více optimalizačních rutin v přepínacích algoritmech poskytuje.



Obr.4: Zjednodušené blokové schéma přepínače - switche

Podrobněji o přepínacích algoritmech se zmiňují autoři např. v [2], [6], [7], [8].

4 Kvalita služeb

Úvodem jedno menší upozornění: I když switch pracuje na linkové vrstvě a kvalita implementovaných služeb je záležitostí síťové vrstvy – a odpovídajícího zařízení, tedy routeru, zmiňme se o důležité vlastnosti sítě – tedy schopnosti kvalitativně rozlišovat typ provozu a tím ovlivňovat i rychlost proudění různých typů dat sítě. Switch sám nepracuje přímo s QoS, avšak existují i zařízení, která částečně integrují i směrování – routing, tudíž již nejsou pro druh protokolů plně transparentní. Pak je i na místě hodnotit data podle tohoto kritéria – tedy QoS.

O QoS se můžeme ve stručnosti dočíst i např. v [2], [4], [5].

Jak již bylo uvedeno, některé přepínače zpravidla implementují v praxi optimalizaci dle kvalitativní analýzy dat přímo v přepínacím algoritmu.

5 Reálný přepínač pro Ethernet

5.1 Přístupová metoda CSMA/CD

Je pro Ethernet jakéhokoli druhu typická. Ethernet je síť stochastická, nedeterministická. To je za určitých okolností limitující faktor, neboť při rostoucí zátěži klesá její propustnost – klesá efektivní přenosová rychlost. Přístupová metoda CSMA/CD sice odstraňuje možnost totální kolize dat na přenosovém médiu, ovšem s rostoucím provozem má každý uzel stále menší pravděpodobnost úspěšného vysílání. Patří do třídy CSMA, tedy metod s vícenásobným kolizním přístupem a nasloucháním nosné. Tato třída se hojně využívá i u WiFi (Wireless Fidelity), a to ve variantě CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Zastavme se u CSMA/CD - osvědčené metody, která v praxi bohatě prokázala svoji životaschopnost, blíže.

Pokud chce stanice odeslat data, postupuje se následovně:

- Nejprve poslouchá, zda je přenosový kanál volný. Pokud ne, pak čeká (tím je vysvětleno i snižování propustnosti za provozu v závislosti na zatížení sítě)
- Při uvolnění kanálu se zahájí vysílání. Zároveň detekuje, zda nevysílá jiná stanice zapojená do téhož segmentu sítě. Pokud ano, pak dochází ke kolizi, vysílající stanice vysílání ukončí a vyšle ostatním kolizní signál *jam*. Poté vybere náhodné číslo z intervalu 0 až 2^k-1 , kde k je pořadové číslo pokusu o vysílání. Toto číslo nabývá hodnot max. do 10. Hodnota tohoto čísla je tedy nejvýše 1023. je to délka čekací doby, po jejímž uplynutí stanice opakuje pokus o vysílání (od začátku). Maximální počet pokusů je 16, poté je pokus o odesílání považován za neúspěšný.

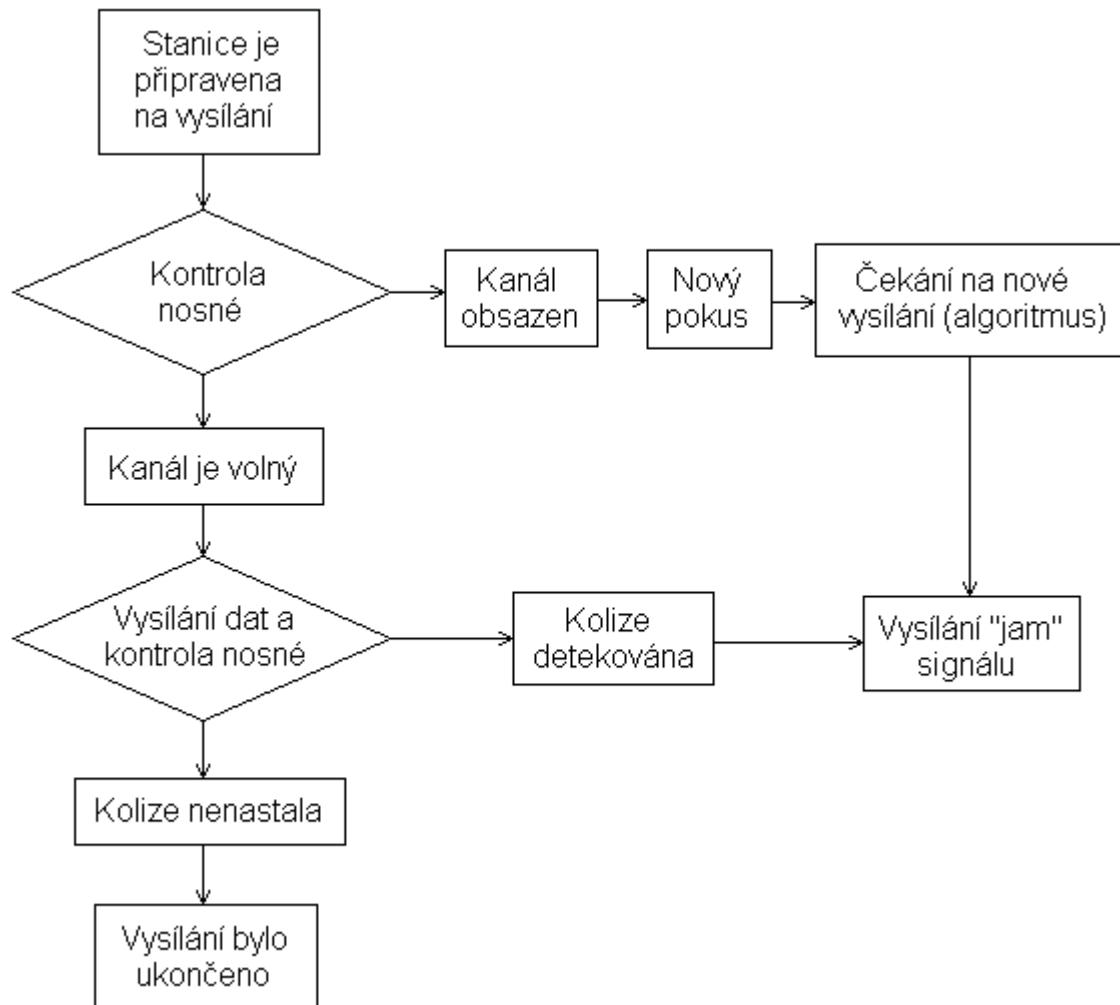
Důležité je zastavit se rovněž u pojmu **kolizní okno**. V algoritmu CSMA/CD má totiž přímé dopady na nejmenší datovou část Ethernetového rámce. Podíváme se na situaci poněkud podrobněji:

Ke kolizi může dojít, jen pokud dvě (či více) stanice zahájí vysílání současně nebo v krátkém časovém úseku za sebou. Jakmile první vysílající stanice „obsadí“ médium, ostatní již čekají. Doba, která uplyne od okamžiku zahájení vysílání stanice do okamžiku, kdy se její signál rozšíří do celého přenosového média, se nazývá kolizní okno. Je jasné, že pouze během této doby může dojít ke kolizi. Tento signál se šíří rychlostí elektromagnetického záření, resp. rychlostí světla (max. hranice této rychlosti je cca $3 \cdot 10^8$ m/s), závisí tudíž na délce a kvalitě média a na zpoždění v aktivních prvcích. Aby byl algoritmus CSMA/CD úspěšný, musí být kolizní okno menší než doba vysílání nejmenšího rámce. Dá se odvodit, že datová část ethernetového rámce má minimální délku 46 B,

což společně s hlavičkami představuje minimální rámec velikosti 64 B.

Další dopady existence kolizního okna:

- Minimální velikost celého rámce 64B
- Maximální délka média a počet opakovačů na trase jsou omezeny.
- Komplikuje se zvýšení přenosové rychlosti. Vede ke zkrácení doby vysílání minimálního rámce, která vynucuje příslušné zkrácení kolizního okna.



Obr.5: Princip algoritmu CSMA/CD

CSMA/CD rozebírá podrobně [1].

5.1.1 Funkce a činnosti reálného přepínače

Vlastní přepínač je, podobně jako fyzické zařízení, pojat jako „krabička“, v Simulinku bychom mohli říci *subsystem*. Ten má v podstatě 4 vstupy *serverů* a 4 vstupy *klientů*, kteří poptávají data, produkovaná těmito servery. Úkolem přepínače je správně a především ve správném pořadí distribuovat data jednotlivým cílům, a to právě a jenom jim – žádnému jinému. Pro zjednodušení uvažujeme simplexní provoz, protože duplex jsou v podstatě dva antiparalelně zapojené simplex.

Základní přepínač, který se hodí kupř. pro domácí síť, zpravidla neobsahuje podporu priority dat, tedy QoS. Proto pracuje pouze s adresami MAC a data pustí prostě tam, kde najde cílovou MAC adresu. Nenajde-li ji, pak ji patrně nemá ve své paměti a pak signál distribuuje do všech výstupů – protože jde ve skutečnosti o plně duplexní provoz, pak prakticky do všech zásuvek kromě své vlastní – pouze tam má totiž stoprocentní jistotu, že tam neznámý cíl nemůže být připojen.

Další problematika souvisí s *ethernetovým automatickým dohodnutím (Ethernet Auto-negotiation)*. Přepínače se takto dohodnou na *plném* či *polovičním duplexu* a hlavně na rychlosti provozu. Každý reálný přepínač umí rozpoznat přenosovou rychlost rozhraní, kabelem připojeného na svůj port. S tímto rozhraním pak komunikuje jeho – nižší – přenosovou rychlostí, než svojí vlastní. Například síťová karta 10 Mbps je připojena na switch schopný provozu 100Mbps. Pak se toto rychlejší zařízení přizpůsobí pomalejšímu. Je to sice brzda síťového provozu, avšak přepínač je schopen komunikovat pomalu pouze s tímto zařízením, čímž nerozšiřuje nevhodnou kolizní doménu do dalších částí sítě.

Zařízení může ovšem použít plný duplex pouze tam, pokud na lince není možná kolize. V praxi to znamená, že *nepoužíváme mix přepínačů a rozbočovačů*, který by tuto kolizní doménu neúměrně (a ve větší síti i nekontrolovatelně) rozšiřoval. Lze-li to, pak rozbočovače ze síťové infrastruktury vyřadíme úplně. Není-li to z jakéhokoli důvodu možné, pak je umístíme *až na úplný konec sítě, co nejbliže stanicím, nikdy do blízkostí serverů či na páteřní propojky!*

5.1.2 Bufferování přepínače

Rozbočovač se chová jako multiportový opakovač, přijatý signál jednoduše zopakuje do všech ostatních zásuvek. Přepínač však musí počítat se situací, kdy přijme několik ethernetových rámců současně z různých vstupů a musí je distribuovat na jeden výstup. Pokud používá algoritmus typu Store and Forward, pak postupuje zpravidla dvojím způsobem:

1. Nemá podporu kvality služeb (jde o přepínač pracující čistě na linkové vrstvě) – načte do vyrovnávací paměti ve formě fronty FIFO (First In, First Out) všechna data, která je potřeba přepnout do konkrétního výstupu, a to v pořadí, ve kterém přišla (sekvenčně). Ve stejném pořadí tato data pošle do výstupního portu.
2. Má podporu kvality služeb – pracuje tedy na rozhraní linkové a síťové vrstvy (L3 switch). Pak, podle parametru určujícího prioritu rámce, provede nejprve seřazení do fronty a poté její přeskládání vzestupně tak, aby nejprve byl obslužen rámeček s nejvyšší prioritou. Algoritmů řazení je více, stejně tak může firmware pracovat s více buffery.

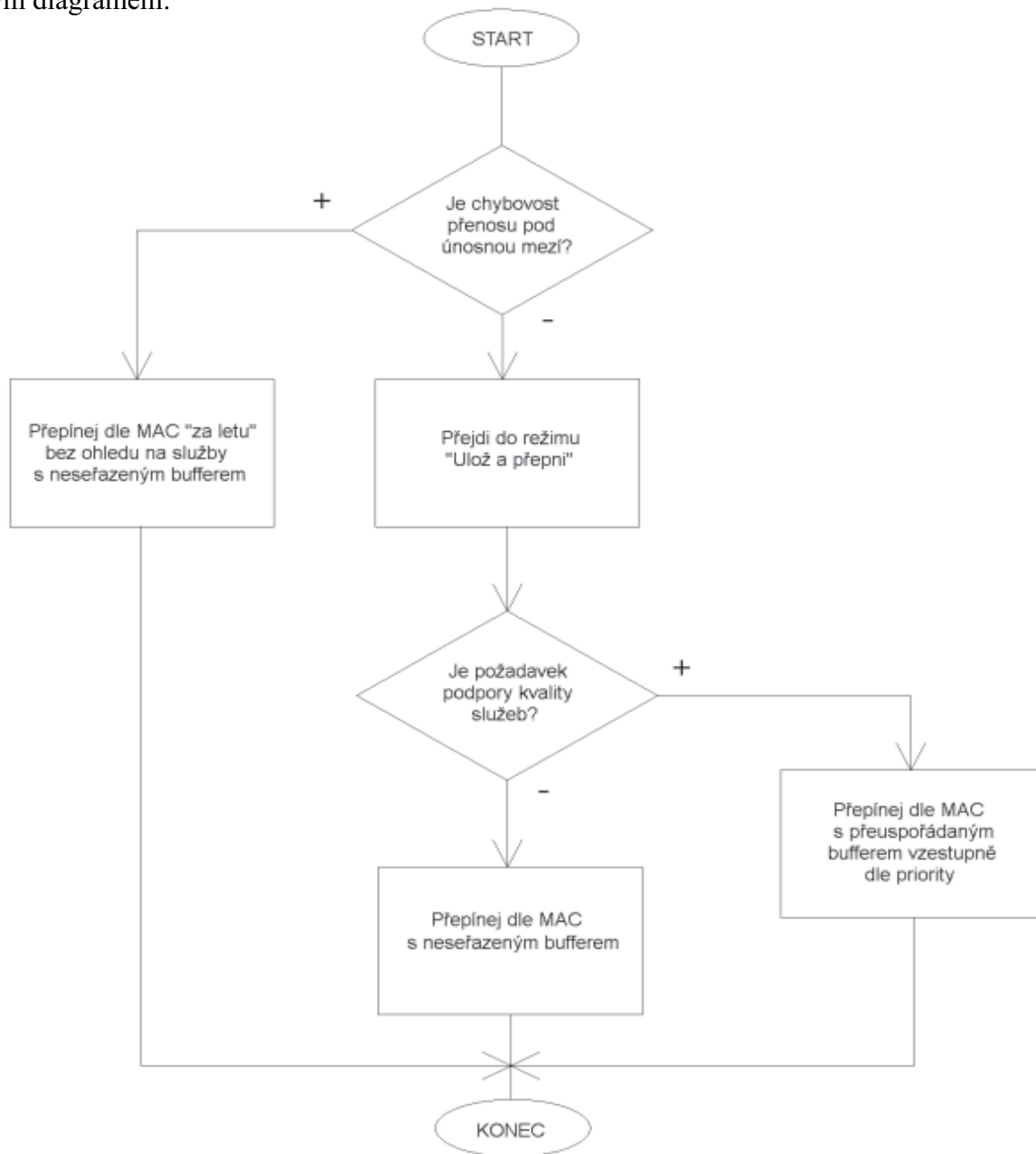
Vlastní řazení má však svoji výpočetní složitost, tudíž časovou režii. Nesmí proto neúměrně zdržovat odesílání rámců – je nutno pracovat kompromisně v určitých daných časových úsecích a frontu po uplynutí tohoto úseku nekompromisně vyprázdnit do náležitého výstupu.

U přepínací metody využívající algoritmus On the Fly je *požadovaná doba průchodu přepínačem minimální*. Pokud nastane provozní situace, kdy rámce proudí sítí plynule a nedochází k vyžádání opakovaného vysílání (sít' nechybuje), může i přepínač bez podpory kvality služeb dosahovat výborných výsledků. Časová režie je minimální a navíc takový přepínač je protokolárně transparentní. I když převážná většina síťového provozu se odehrává pod TCP/IP, mohou být takto lehce odbavovány i rámce nesoucí protokoly IPX/SPX či jinou protokolovou sadu. Rovněž u přepínače transparentního k datovému obsahu rámců nemusíme řešit nekompatibilitu verzí TCP/IP (4 vs. 6). Kvalita služeb pak může být plnohodnotně řešena až na směrovačích, kam tato problematika standardně patří.

Soudobé přepínače podporující kvalitu služeb, zejména dražší produkty (např. Cisco) však implementují *adaptivní přepínání*. Tím se přepínač přizpůsobuje aktuální situaci v síti a volí mezi typem přepínání tak, aby vlastní přepnutí proběhlo v co nejkratším čase.

Samozřejmě – čím sofistikovanější přepínací algoritmus, tím je i vyšší cena vlastního přepínače. Profesionální – konfigurovatelné přepínače mají rovněž druhotný náklad – práci administrátora, který by měl optimálně a netriviálně nastavit jejich vlastnosti.

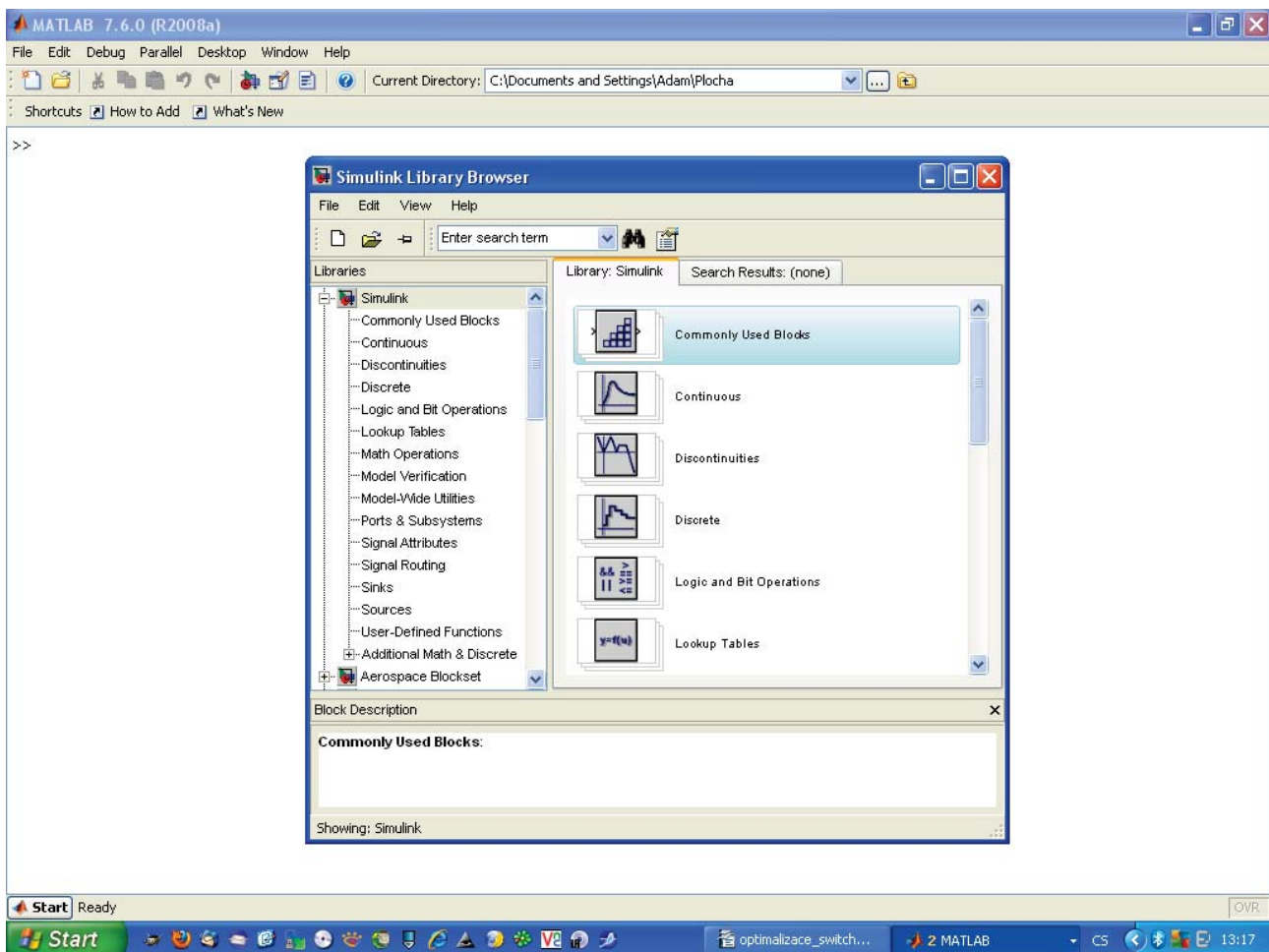
Algoritmus adaptivního přepínání (Adaptive Switching) můžeme popsat následujícím vývojovým diagramem:



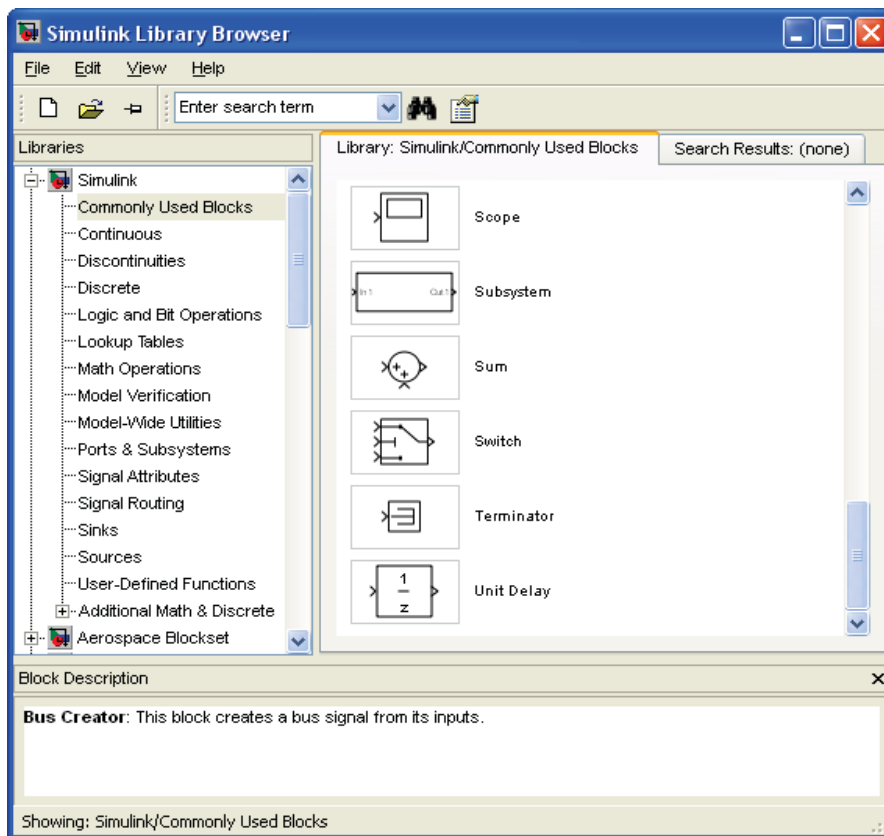
Obr.6: Algoritmus zjednodušeného adaptivního přepínání

6 Simulace přepínače v prostředí Matlab Simulink

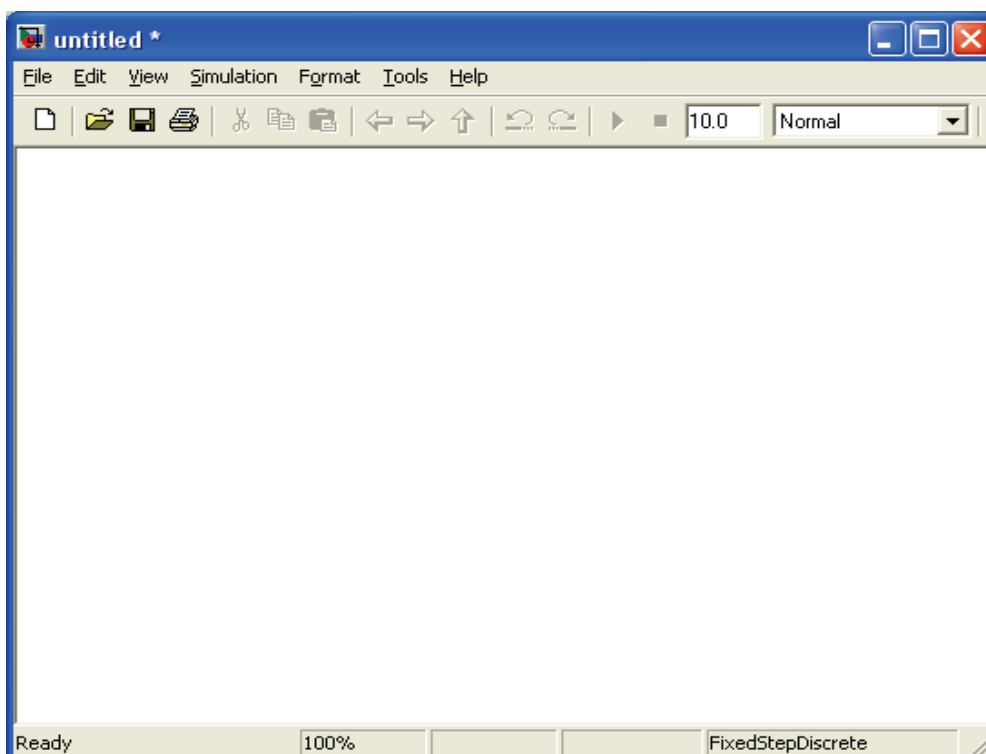
Firma MathWorks, Inc. dala technikům neobyčejně silný a účinný nástroj v produktu Matlab. Je to zkratka *Matrix Laboratory*, tedy doslova *Maticová laboratoř*. Skutečně – většina dat, se kterými v Matlabu pracujeme, má charakter vícerozměrných vektorů – matic. Avšak to není všechno. V posledních verzích se z tohoto čistě praktického nástroje vyvinulo kvalitní integrované vývojové prostředí pro vědeckotechnické výpočty všeho druhu. Popisem tohoto programu se nebudeme zabývat – zabral by totiž několik set stran. Jeho podstatným nástrojem je ale doplněk *Simulink*, ve kterém bude realizována simulace jednoduchého přepínače (switche) v konvergované síti. Navrhne model, který generuje paketová data různého typu, které v této – zjednodušené úvaze – oklasifikujeme metodami podobnými jako v QoS a posléze je zavedeme do modelu přepínače, který s nimi bude nakládat tak, aby propojil příslušné vstupy s příslušnými výstupy. Nesmí být přitom neúměrně zvětšována kolizní doména.



Obr.7: Prostředí Matlab se spuštěným Simulinkem



Obr.8: Hlavní okno Simulinku s výběrem simulovaných komponent



Obr.9: Okno nového modelu, ve kterém bude simulace prováděna

6.1 Založení nového modelu a vytvoření typického zdroje dat

Nový model založíme např. volbou **File – New – Model**. Vytvoří se soubor s příponou `.mdl` a otevře se k editaci. Z hlavního okna postupně vybereme komponenty, které budeme přidávat do okna modelu tažením myši, tyto komponenty navzájem propojíme a přiřadíme jim požadované vlastnosti. Komponenty modelu můžeme do systému přidávat ve formě diskretních bloků, častěji se to ovšem realizuje formou subsystémů, které pak dále specifikujeme v nižších úrovních. Celkově se tak vzhled modelu zpřehlední a model pak může být pojat jako subsystém složitějšího modelu. Z toho pohledu se jako nejdůležitější komponenta jeví *subsystem*, který najdeme ve skupině *Commonly Used Blocks*. Primární pojmenování skupin i komponent je anglické, každou komponentu je však vhodné pojmenovat ilustrativním českým názvem. Pojmenování ponecháme však z důvodu nastavení Simulinku bez české diakritiky. Detailní popis Simulinku je v [10].

6.1.1 Zdroj dat

V této práci vytvoříme jako ilustrativní příklad zdroj dat (rámců), který bude reprezentovat počítač připojený k přepínači. Počítač má výstup na síťovém rozhraní, vstup z přepínače a rovněž číselný ukazatel, který zobrazí požadovaná data, aby na modelu bylo vidět, jak data proudí přepínačem.

Vlastní data jsou pojata jako rámec, který má několik částí, podstatnou je pro činnost hodnota MAC, která reprezentuje MAC adresu a parametr Q, který klasifikuje typ dat, podobně, jako tomu je u QoS. MAC je adresa fyzická, to znamená, že stojí a padá se síťovým rozhraním, je pro něj unikátní a zde nám označuje zdroj a cíl – tak, jak s těmito údaji skutečný switch pracuje. V každém rámci je nutno označit zdrojovou a cílovou MAC. Zatímco v našem modelu je zdrojová MAC konstanta, cílová MAC je proměnná, kterou volíme skupinou mechanických přepínačů podle toho, jaký má být adresát. Dalším údajem je již zmíněný parametr Q, který označuje *typ provozu*. Je to opět proměnná, která figuruje jako faktor priority, a tím i *optimalizace provozu*. Čím je tato hodnota vyšší, tím je případná priorita vyšší. I tuto proměnnou testovacímu rámci zadáme kombinací mechanických přepínačů. Vzestupná hodnota tohoto parametru respektuje i filozofii QoS, kde se rovněž parametry IPP a DSCP s rostoucí prioritou zvětšují.

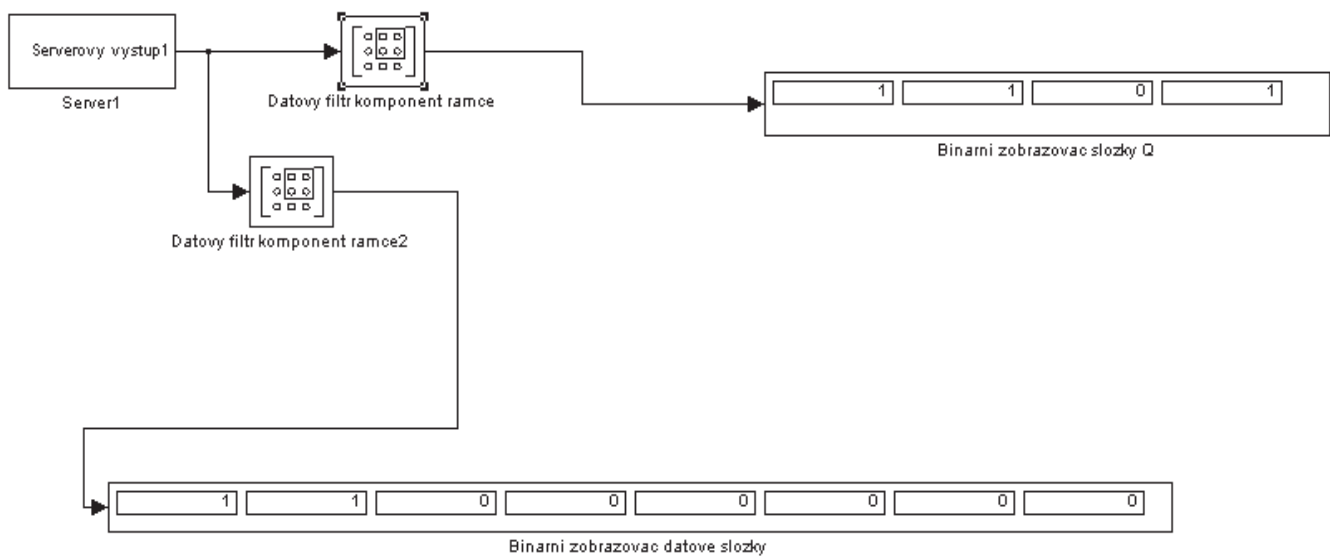
Typ dat	Hodnota parametru Q (binárně)	Hodnota parametru Q (dekadicky)
Televize (multimédia)	1101	13
Digitální telefonie VoIP	1001	9
Řízení báze dat	0101	5
Přenos souborů	0000	0

Tab.1: Hodnoty parametru Q podle typu dat

Základní schéma zdroje datových rámců pro simulaci přepínače je uvedeno níže. Skládá se z jediného subsystému *Serverový výstup*, který sestaví datový rámec pro vlastní zpracování přepínačem. Místo přepínače zde demonstrujeme jednu velice důležitou funkci – *datový filtr komponent rámce*. Je to maticový selektor, komponenta *Submatrix*, který podle zadané pozice vybere z posloupnosti nul a jedniček tu část, kterou právě potřebujeme – tedy je to jakýsi digitální filtr. S jeho pomocí si můžeme rámec „rozřezat“ na libovolné komponenty, ty analyzovat a podle výsledků této analýzy s rámcem i naložit. Takto budeme v přepínači filtrovat MAC adresy a parametr Q, prioritizující provoz přepínače.

Stanice	MAC adresa binárně	MAC adresa dekadicky
Televize (multimédia)	0001	1
Digitální telefonie VoIP	0011	3
Řízení báze dat	0111	7
Přenos souborů	1111	15

Tab.2: MAC adresy stanic. S těmi model bude pracovat

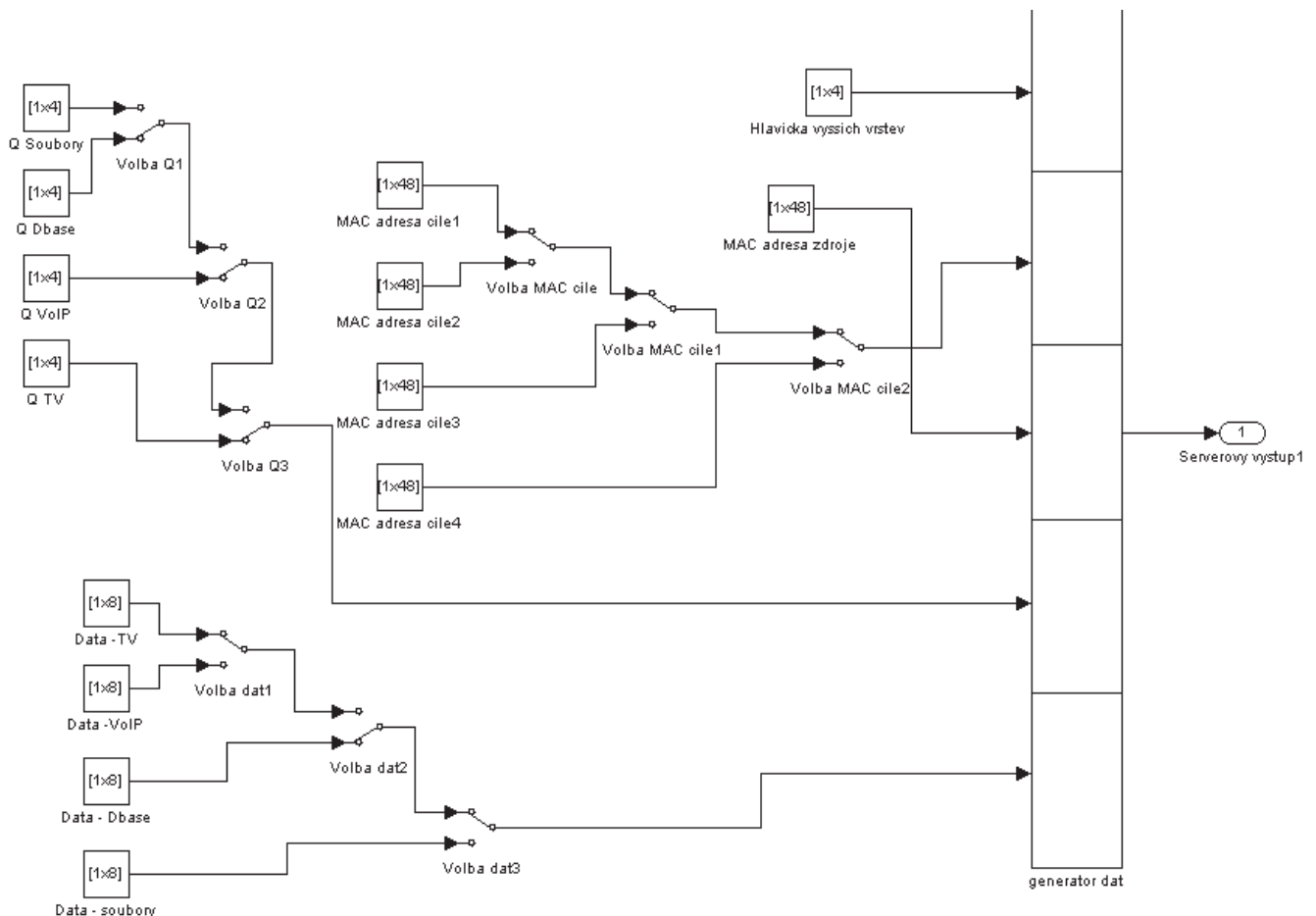


Obr.10: Ze serveru odchází data a podle filtrace parametru Q ihned poznáváme, že jde o data digitální televize – binární hodnota Q je 1101

Nyní se podívejme stručně na subsystém *Serverový výstup 1*. V celkovém modelu bude takovýchto konfigurovatelných zdrojů dat tolik, kolik je vstupů přepínače – zvolili jsme 4, protože uvažujeme o 4 kvalitativních typech dat. Každý zdroj je možno přepnout na vysílání jiných typů dat, je možno mu vnútit MAC zdroje i cíle a rovněž parametr Q. Jedinými konstantními hodnotami zůstává zdrojová MAC a *hlavička vyšších vrstev*, kterou „přibalily“ jiné vrstvy ISO/OSI. Pro činnost přepínače je tato hlavička obsahově transparentní.

Většinu parametrů v laboratorním modelu zvolíme skupinou sériových střídavých přepínačů. *Vzhledem k potřebě vícepólových přepínačů a jejich absenci v knihovně komponent Simulinku jsme zvolili jejich sériově kaskádní zapojení.*

Podle potřeby je možno měnit hodnotu konstant přímo v masce nebo nevyhovující komponentu zaměnit v subsystému za jinou, kterou vybereme z knihovny komponent Simulinku.



Obr.11: Subsystem Serverový výstup 1 (zdroj dat). Z polohy přepínačů je vidět, že odtud vysíláme televizní data a parametr Q činí 1101. Tuto hodnotu jsme odchytili za pomoci filtru – viz předchozí obrázek

6.1.2 Zásady vytváření modelu přepínače

Vlastní model přepínače je *zjednodušením reálné situace*. Dá se říci, že je to didaktická pomůcka pro výuku předmětů, které pojednávají o principu přepínání dat v počítačových sítích. Proto je tento model zjednodušením, které samozřejmě nepostihuje všechny situace, které mohou v přepínání rámců nastat.

Zejména – pro názornost – je provoz pojat jako simplexní a bod proti bodu (unicast). Toto zjednodušení není na závadu, pokud pojmem duplexní provoz jako dva antiparalelní simplexní provozy.

Dále jsme využili velice užitečné možnosti Simulinku vytvářet subsystemy – menší složené prvky. To značně zpřehlední situaci, neboť bez této možnosti by vzniklo rozsáhlé, zdánlivě chaoticky pospojované schema, přestože i v tomto systému by každá propojka i komponenta byly smysluplné.

6.2 Komponenty modelu přepínače

Model sestává z několika komponent. Mezi jeho hlavní části patří:

- 4 serverové systémy jako zdroje dat
- 4 stanice jako datové spotřebiče
- Vlastní přepínač

Zdroje dat jsou nastaveny tak, aby produkovaly zjednodušené rámce v ethernetovském pojetí, spotřebiče dat tyto rámce vyhodnocují a jejich vybrané – vyfiltrované – části zobrazují, a to buď na číselný displej nebo na osciloskopickou obrazovku, která dovoluje zobrazit průběh v jednotlivých časových úsecích simulace.

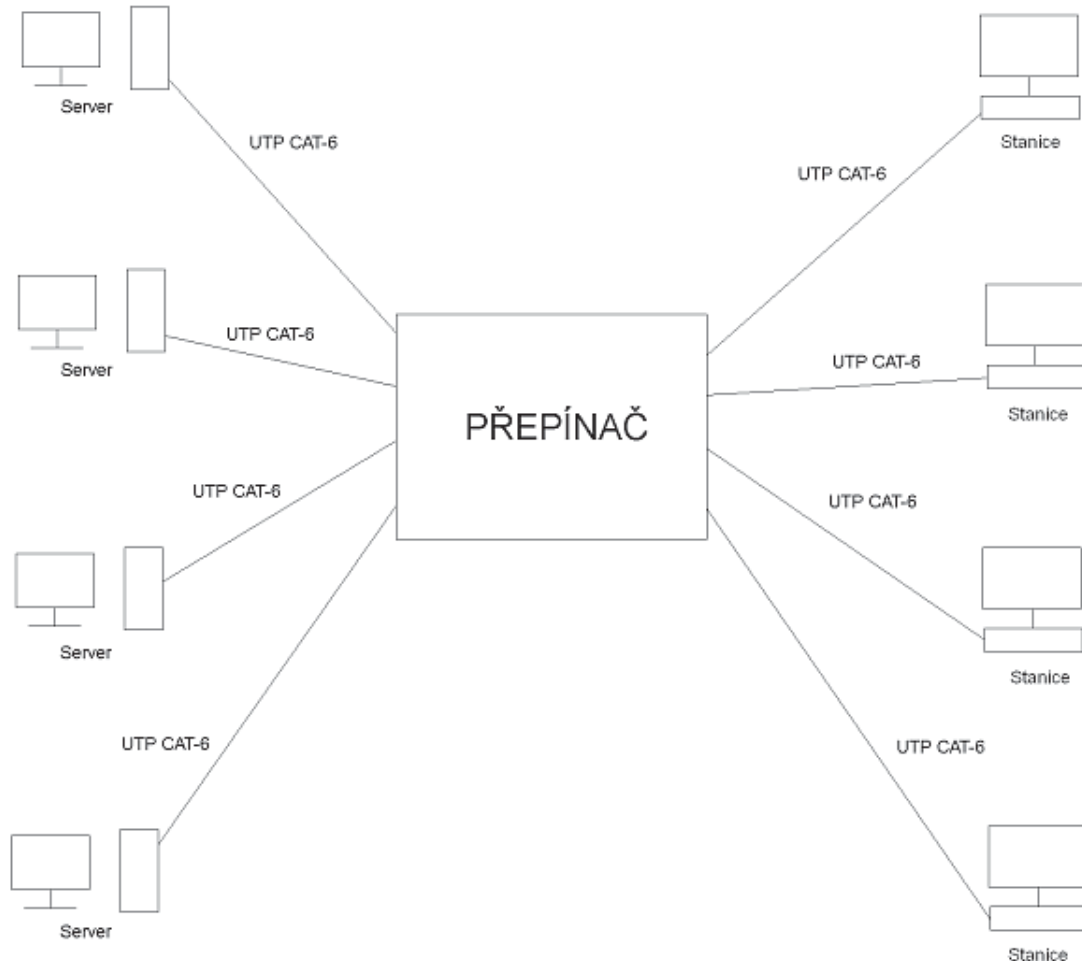
Obecně platí, že přepínač pracuje s MAC adresami, kontroluje formální správnost a integritu rámce, ovšem vůči tekoucím datům je transparentní. Tato zásada je zde poněkud porušena, neboť se počítá i s dobou doručení, tj. s kvalitou služby. Hodnotu QoS, nesoucí tuto informaci jsme nahradili čtyřbitovým parametrem Q.

Rovněž jsme zjednodušili MAC adresy. Ty jsou ve skutečnosti 48 bitové. To je zachováno, nicméně pouze pro ilustraci šířky adresy. Používáme poslední, nejnižší 4 bity adresy, které srovnáme v tabulce přepnutí a tím určíme správný výstup přepínače. Takto postupuje i skutečný přepínač. Zde je přepnutí deterministické, tedy model *vždy* najde cíl. Ve skutečnosti tomu tak být nemusí. Skutečný přepínač se při *nenalezení* cílové MAC adresy chová jako rozbočovač – zopakuje signál na všechny výstupy.

Data jsou v našem modelu 8 bitová. Chápeme je bez hodnoty Q, se kterou pracujeme. Můžeme si je opět zobrazit na displeji nebo osciloskopu. Protože však nemají pro naši ilustraci hlubší význam, tento zobrazovač primárně v modelu není. Zobrazujeme cílovou MAC adresu a hodnotu parametru Q, abychom viděli, *kam a jaká data posíláme*. Ve skutečném rámci jsou data nejdelší a mají šíři řádově stovek až tisíce bitů.

6.2.1 Popis zapojení

Celkově je situace zapojena dle obrázku. Je to logická topologie, která respektuje tok dat od serverů ke klientům:



Obr.12: Základní logické zapojení modelu

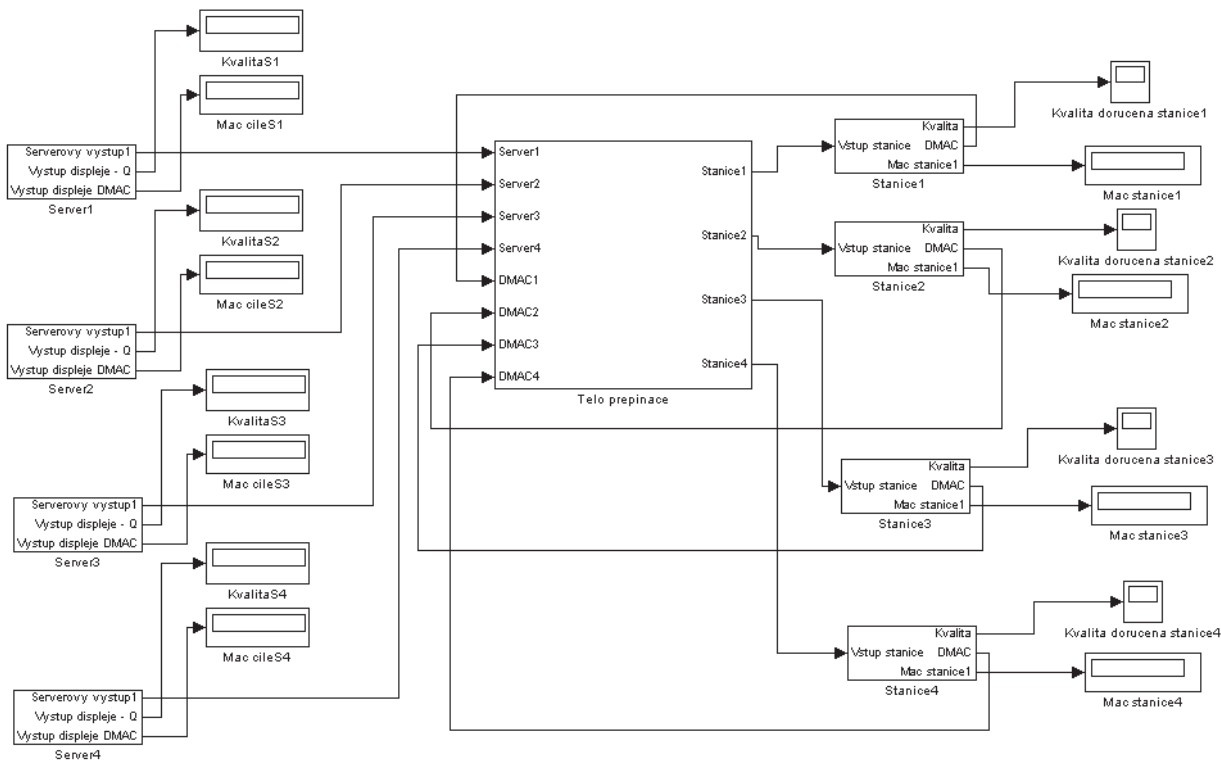
6.2.2 Popis modelu

Uvedeme pouze stručný popis jednotlivých komponent modelu. Zájemce má tento model na doprovodném CD-ROM, může si jej spustit a prohlédnout podrobněji – až do hodnot nastavení jednotlivých bloků. Zde jde spíše o takový popis, aby čtenář získal představu o činnosti modelu podle jeho dílčích částí.

Spustíme na naší pracovní stanici program Matlab. Po náběhu základní konzoly zadáme příkaz `simulink` a vyčkáme na start tohoto modulu. Překopírujeme ze sítě nebo z příloženého CD

-ROM soubor `prepinac13.mdl` do pracovního adresáře – může být libovolný, avšak takový, kde máme právo číst, zapisovat a spouštět soubory. Simulink si totiž v průběhu práce vytváří pomocné soubory a při nemožnosti jejich zápisu by mohl havarovat. Z prostředí Simulinku pak soubor s modelem otevřeme.

Hlavní vrstva modelu



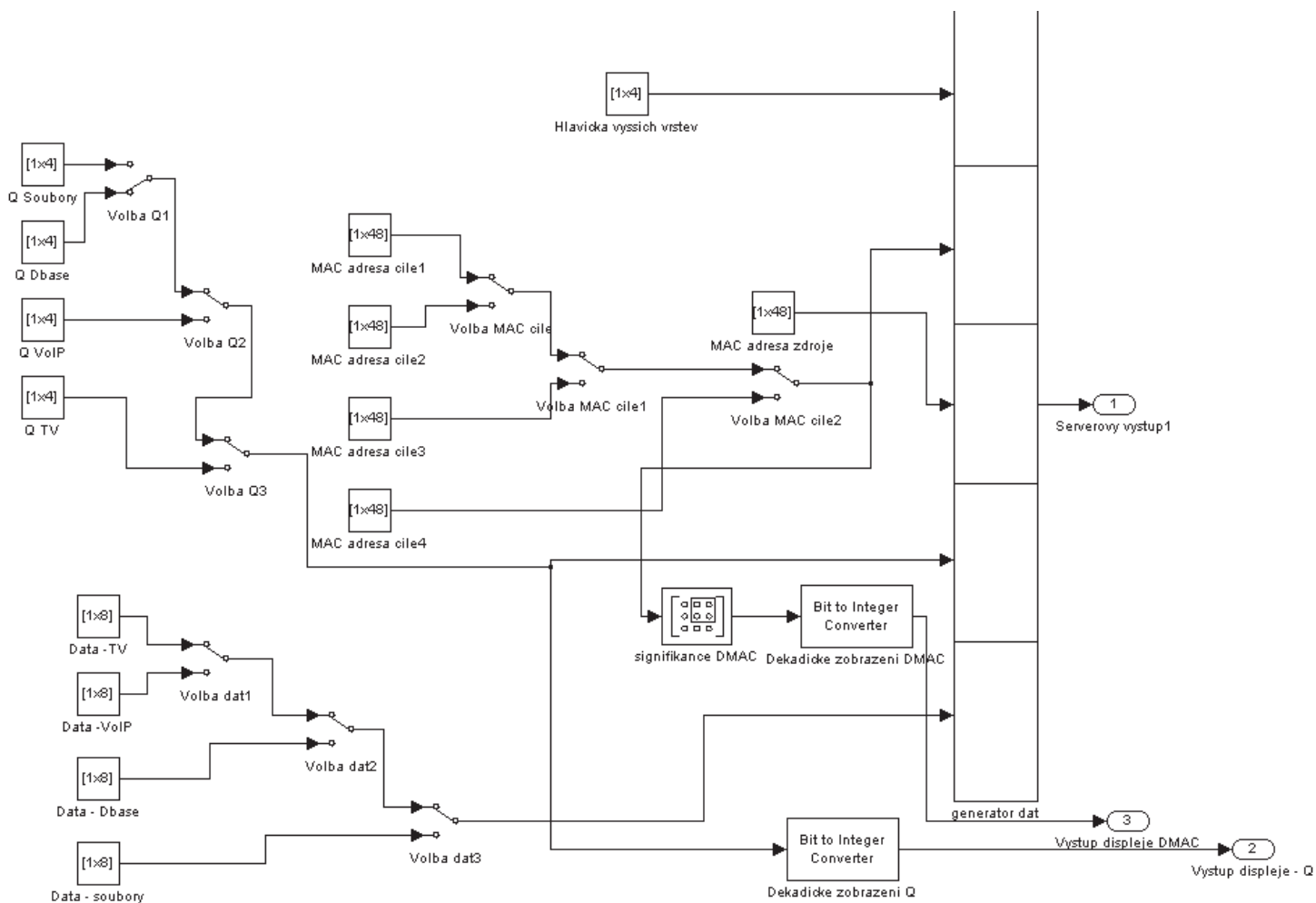
Obr.13: Základní pohled na model. Sestává z přepínače , 4 serverů jako zdrojů dat a 4 stanic jako cílových uzlů

Po otevření souboru s modelem se objeví schema podle výše uvedeného obrázku. Model sestává ze 4 serverů, které jsou schopny doručovat data s příslušnou kvalitou k odpovídajícím stanicím, dále z vlastního přepínače a 4 stanic, které jsou cílovými uzly. Tvoří logickou topologii hvězda, bez dalších návazností, celek je tedy *uzavřeným systémem*. Servery i stanice mají displeje, zobrazující dekadické údaje signifikantních částí MAC adres a stanice navíc osciloskopy, které zobrazí příchozí údaj Q, tedy kvalitu služby – typ dat. Podle tohoto údaje můžeme lehce identifikovat propojení bod proti bodu, což je pro přepínač typické. Přepínač je řízen deterministicky MAC adresami cílového objektu. V praxi se k MAC adrese dospěje např. použitím protokolu ARP, zde je cílová Mac (označeno DMAC) nastavena apriori. Uvnitř přepínače se

emuluje přepínací tabulka a řídí selektivní přepnutí na příslušný výstup.

Servery

Servery jsou tvořeny aditivním blokem, který sestavuje zjednodušený rámec podle podstatných a využitelných komponent. Vynechali jsme nepodstatné části, protože by entita rámce narostla do neúměrných binárních hodnot – přes 1500 bitů. Náš rámec má celkem 112 bitů pevných, i když samozřejmě skutečný rámec má svoji hodnotu násobně delší. Pro konfiguraci serverů slouží mechanické přepínače, které nastavují hodnotu parametru Q, cílovou MAC adresu a (formálně) i vlastní data. Ta lze dodatečně vyfiltrovat a rovněž zobrazit na displeji či osciloskopu. Vnitřní struktura serveru je následující:

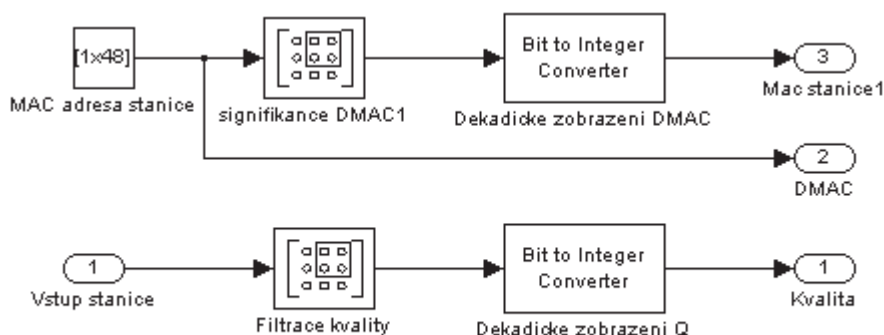


Obr.14: Vnitřní struktura serveru. Generátor dat spojí jednotlivé dílčí konstanty a proměnné do celistvého rámce, dlouhého 112 bitů

Poklepáním na mechanické přepínače lze servery konfigurovat a nastavovat tak různá vstupní data pro zpracování přepínačem.

Stanice

Stanice jsou tvořeny maticovými filtry, které z rámce filtrují signifikantní hodnoty pro zobrazení. Toto zobrazení jde ze staničního subsystému jednak na displej (Cílová MAC – zobrazeny poslední 4 bity. Zbylou sekvenci vyšších bitů nepoužíváme – je rovna nulám. V praxi tomu tak samozřejmě není; MAC adresa je 48 bitové číslo zobrazené hexadecimálně, avšak zde si můžeme situaci zjednodušit a zobrazit tento údaj dekadicky.



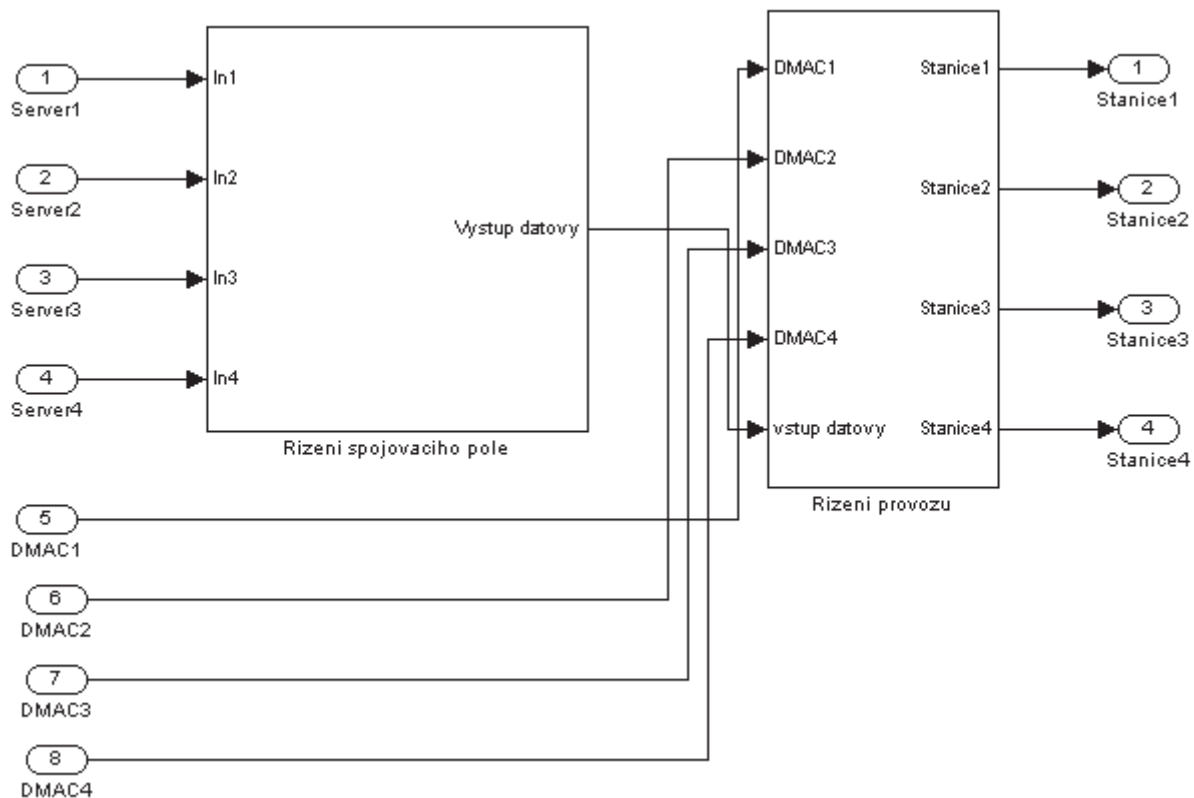
Obr.15: Vnitřní zapojení subsystému Stanice.

Vnitřní struktura přepínače

Přepínač se skládá z několika subsystémů, které jsou vzájemně propojeny. Na hlavní vrstvě jsou to tyto komponenty:

- *Řízení spojovacího pole* – pro kvalitativní hodnocení proudících dat. Sem data prvotně vstupují. Subsystém má za úkol provést *analýzu vstupujících dat a rozhodnout o pořadí přepnutí*. Zde se simuluje skládání dat do fronty typu *FIFO (First In First Out)*, klasifikace podle priority – parametru *Q*, tedy v podstatě *třídění* a dále určuje pořadí přepínání jednotlivých rámců.
- *Řízení provozu*, což je vlastní sekvence přepínání, emulace přepínací tabulky. Tento subsystém pracuje s MAC adresami stanic. Srovnává tento údaj v příchozích datech se skutečností a podle toho, kde nalezne shodu, pošle rámec do příslušné stanice.

Situaci názorně popisuje následující obrázek:

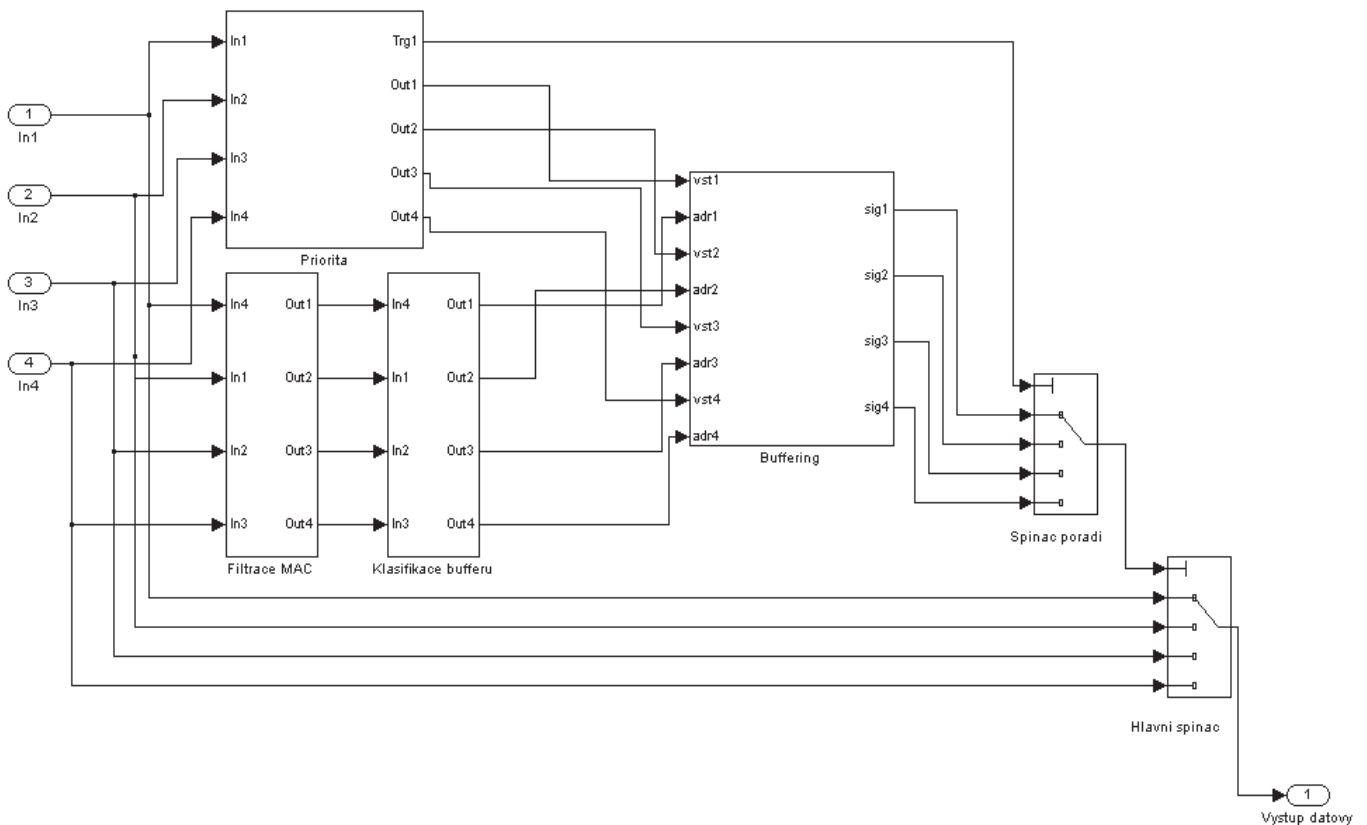


Obr.16: Komponenty Řízení spojovacího pole a Řízení provozu

Řízení spojovacího pole

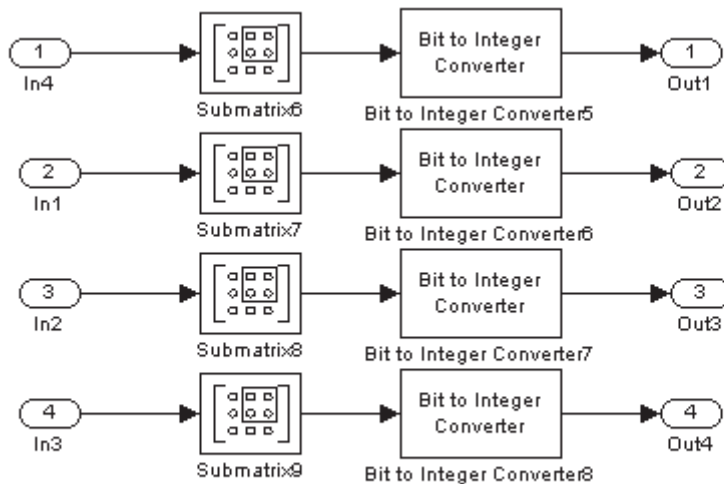
Je to subsystém, který má jako hlavní součásti klasifikaci rámce podle MAC adresy a podle parametru Q. MAC adresa rozhoduje, kam data budou poslána, parametr kvality Q potom kdy se tak stane, tedy v jakém pořadí. Zde je proto nezbytnou komponentou jednoduchý časovač, respektive spouštěč (trigger), který řídí pomocný přepínač. Tento přepínač vysílá do *spínače pořadí* posloupnost řídicích pulsů podle kvalitativního vyhodnocení dat. Pokud bychom zařadili do *hlavního spínače* přímo signál triggeru, pak by přepínač pracoval bez podpory kvality služeb – prostým kruhovým plánováním (round robin). Spínače jsou zapojeny kaskádně a data přenášena sériově. Vzhledem k nespojitosti spouštění musíme jednak nastavit vhodnou dobu a typ simulace (viz dále), jednak jsou zachycená data rovněž „vzorkována“ s frekvencí spouštění. Proto se jeví vhodnější je odchyťovat osciloskopem namísto displeje.

Situaci subsystému Řízení spojovacího pole zachycuje následující obrázek:



Obr.17: Řízení spojovacího pole sestává ze 4 subsystémů. Řeší se zde priorita a bufferování přepínače, což je v podstatě algoritmus třídění

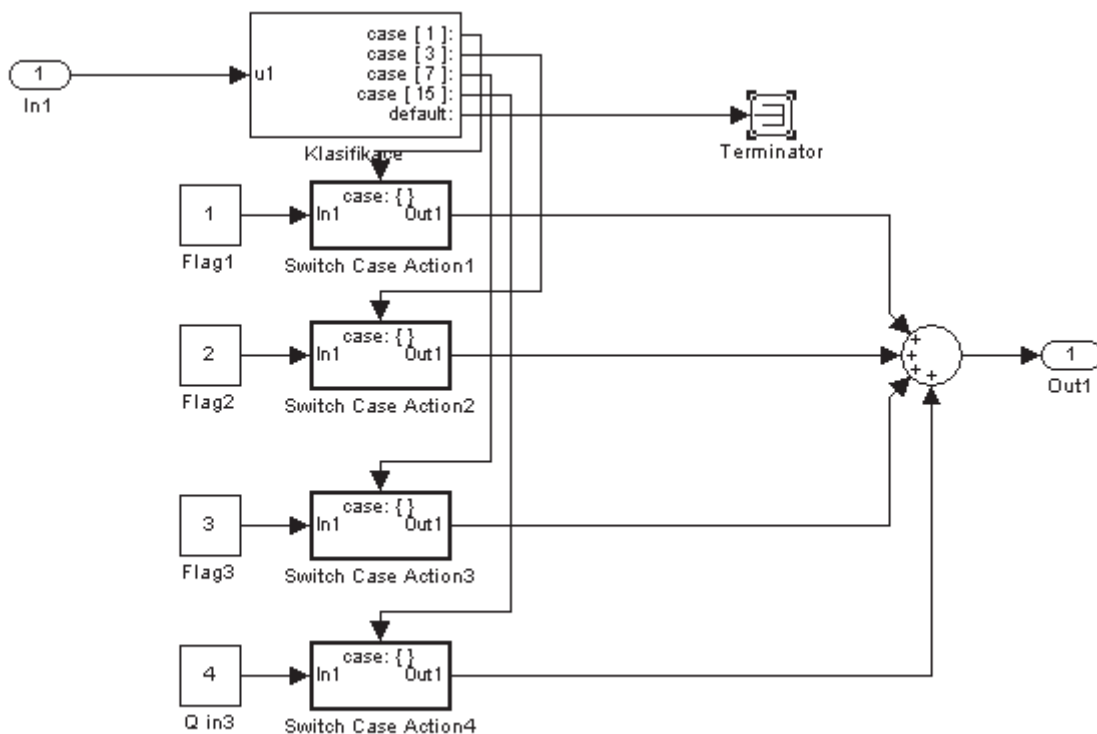
V Řízení spojovacího pole se data dělí a filtruje se cílová MAC adresa z rámce v subsystému Filtrace MAC, aby s ní bylo možno pracovat. Také se zde původní 48 bitové číslo ořízne na signifikantní nejnižší 4 bity, které jsme použili. Poté je toto číslo převedeno do dekadické podoby. Z tohoto subsystému jde tento údaj do Klasifikace bufferu, který vygeneruje signál pro hlavní spínač. Ten má hodnotu 1 až 4 dekadicky a bude ovládat v seřazeném pořadí hlavní spínač. Uveďme si pro přehled zapojení právě popsaných subsystémů:



Obr.18: Filtrace MAC adresy

Filtrace MAC adresy je velmi jednoduchá. Podobně, jako na mnohých jiných místech tohoto modelu, je použita komponenta Submatrix, která ze sériově běžících dat vyfiltruje požadovanou sekvenci. Ta je poté převedena na desítkové číslo komponentou Bit to Integer Converter. Poté již desítkové číslo vstupuje do subsystému Klasifikace bufferu, který přiřadí jednotlivým MAC adresám hodnoty čísel v rozsahu 1 až 4, abychom mohli jimi ovládat hlavní spínač.

Vlastní klasifikace je implementována takto:

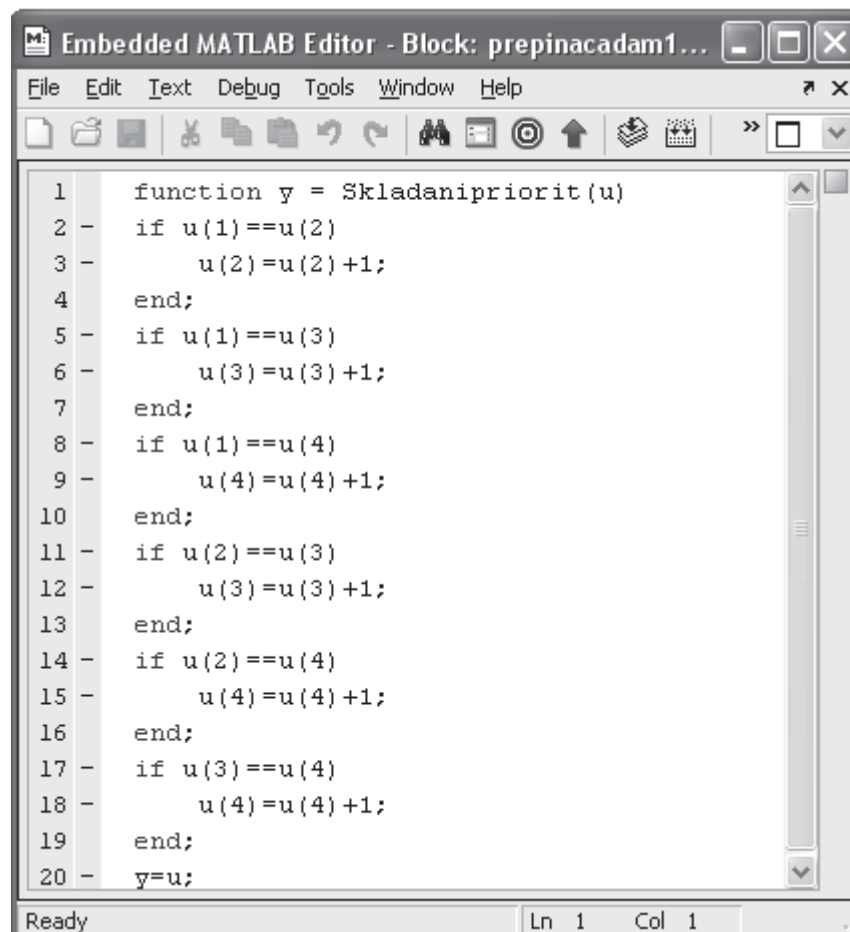


Obr.19: Jeden element subsystému Klasifikace bufferu. V subsystému jsou celkem 4.

Nejprve se zjistí dekadická hodnota MAC. Poté se jí přiřadí čísla z intervalu 1 až 4. To lze v podstatě libovolně, protože k přetřídění dojde až při bufferování. Aby však nevznikal chaos, zvolili jsme vzestupné hodnocení. Komponenta Constant, nazvaná Flag (známka, praporek...) nám tuto hodnotu pro každý vstup a výstup přiřadí a akční blok přepínače typu *Case* ji propojí do výstupu.

Komponenta Priorita a v ní obsažený subsystém Přeskládání FIFO řeší situaci, kdy na jeden výstup přijde *více dat*. Například dva servery poskytují službu jedné stanici. Tato situace v praxi nastává poměrně často, proto ji model musí ošetřit. Popišme si jeden vstup (jsou celkem 4) tohoto subsystému. Při detekci určitého stupně kvality dat (parametru Q se seřadí data do pomocné fronty. Poté se v případě, že přišla data stejného typu, *překlasifikuje* Q tak, že jej zvýšíme o 1 – tím zajistíme *sekvenčnost* dat. Data tedy půjdou na výstup subsystému v takovém pořadí, jak byla generována. Protože pokud jsou stejného typu, pak doručujeme podle pravidla FIFO, tedy první dovnitř, první ven.

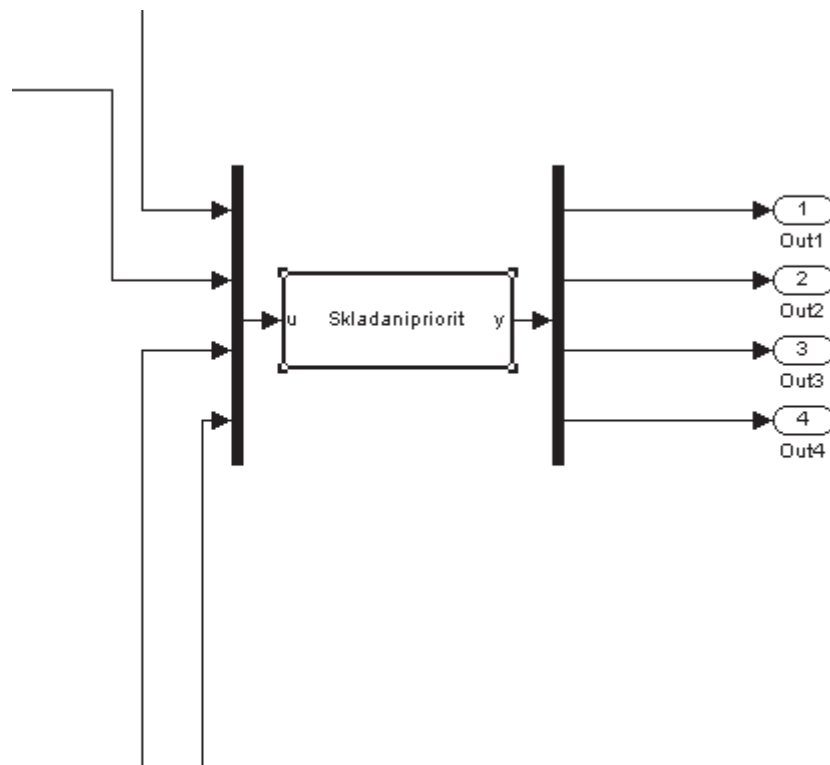
Algoritmus Skládání priorit je následující a zajistí ho komponenta Embedded MATLAB Function:



```
1 function y = Skladanipriorit(u)
2 - if u(1)==u(2)
3 -     u(2)=u(2)+1;
4 - end;
5 - if u(1)==u(3)
6 -     u(3)=u(3)+1;
7 - end;
8 - if u(1)==u(4)
9 -     u(4)=u(4)+1;
10 - end;
11 - if u(2)==u(3)
12 -     u(3)=u(3)+1;
13 - end;
14 - if u(2)==u(4)
15 -     u(4)=u(4)+1;
16 - end;
17 - if u(3)==u(4)
18 -     u(4)=u(4)+1;
19 - end;
20 - y=u;
```

Obr.20: Skládání kvalitativně stejných rámců

Do tohoto bloku vedeme data přes komponentu Multiplexer a poté ji seřazenou demultiplexerem opět rozdělíme do 4 výstupů:



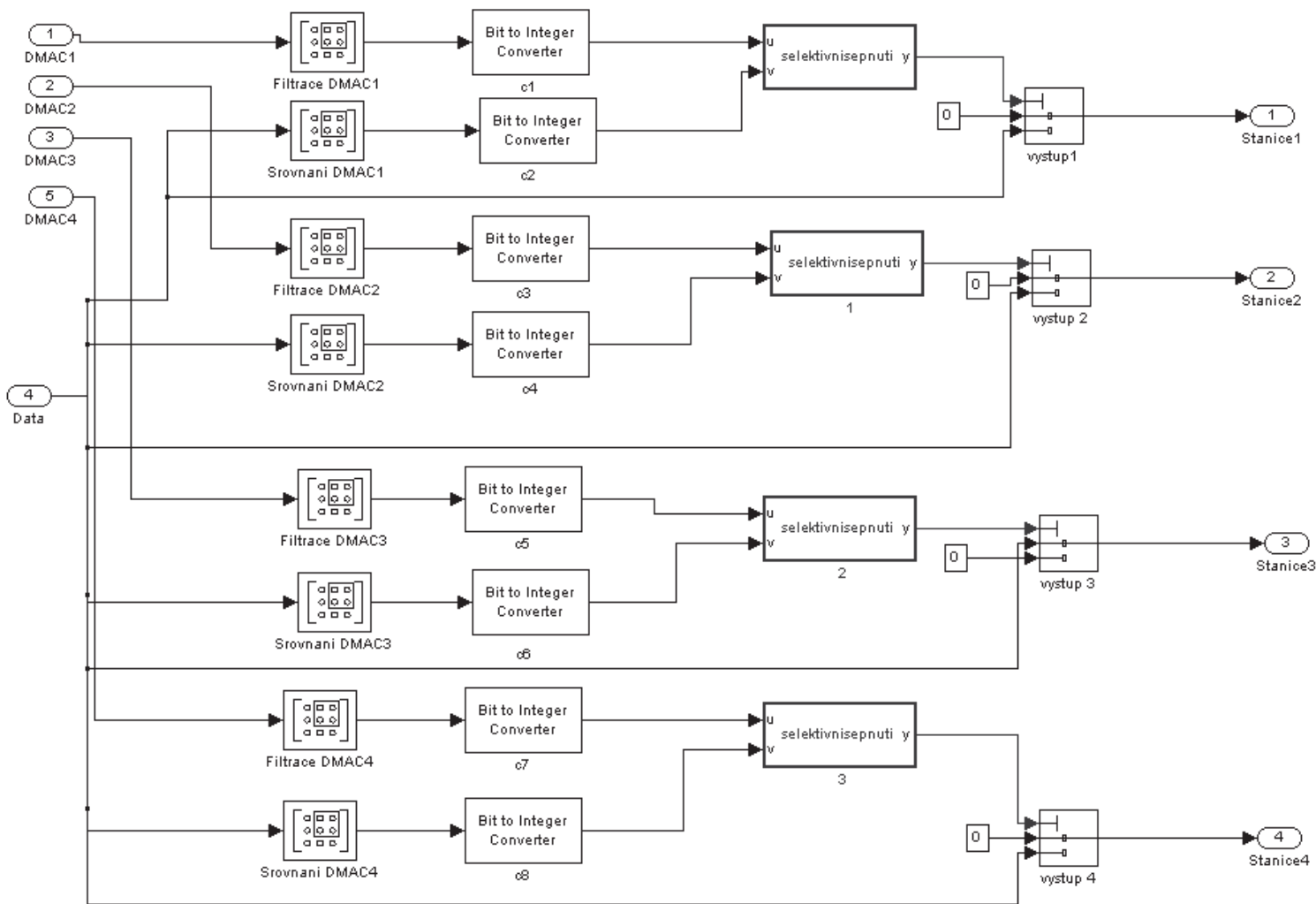
Obr.21: Implementace výše uvedené překlasifikace Q

Výstupy subsystémů Priorita a Klasifikace bufferu vedou do bloku Buffering, kde probíhá jednak finální setřídění dat bublinovým tříděním (možno použít i jiné metody), jednak vygenerování spouštěcích povelů pro hlavní spínač v intervalu 1 až 4 v příslušném pořadí. Tím *hlavní spínač sepne v určitém pořadí hlavní datový tok* do subsystému Řízení provozu, který již pracuje srovnávací metodou s MAC adresami – emuluje tedy přepínací tabulku.

Řízení provozu

Tento subsystém zajistí doručení dat příslušné stanici. Funguje na principu srovnání skutečné MAC adresy stanice (její signifikantní části) s MAC adresou označenou jako cílová, jež nese rámeček. Srovnání se provádí pomocí Embedded MATLAB Function, lze ho však provést i komponentou If.

Jeho vnitřní strukturu označuje následující obrázek:

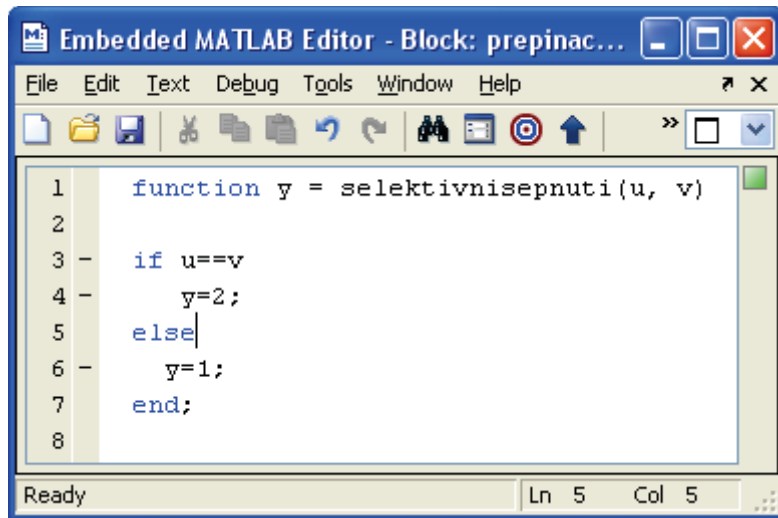


Obr.22: Řízení provozu přepínací části

Činnost tohoto subsystému je zřejmá ze zapojení: Jakmile dojde ve frontě seřazený rámec, v bloku Matlabu nazvaném *selektivnisepnuti* dojde ke srovnání MAC adres. Protože ty v modelu mohou být pouze 4, propojí se odpovídající adresa bod proti bodu tím způsobem, že portový spínač dostane povel k přepnutí na datovou část. Jinak propojí nulu.

Pokud bychom chtěli simulovat přepnutí na neznámou MAC adresu, doplnili bychom tento subsystém tak, aby propojil tyto adresně neznámé rámce do všech výstupů. Tím bychom softwarově suspendovali přepínač do role rozbočovače.

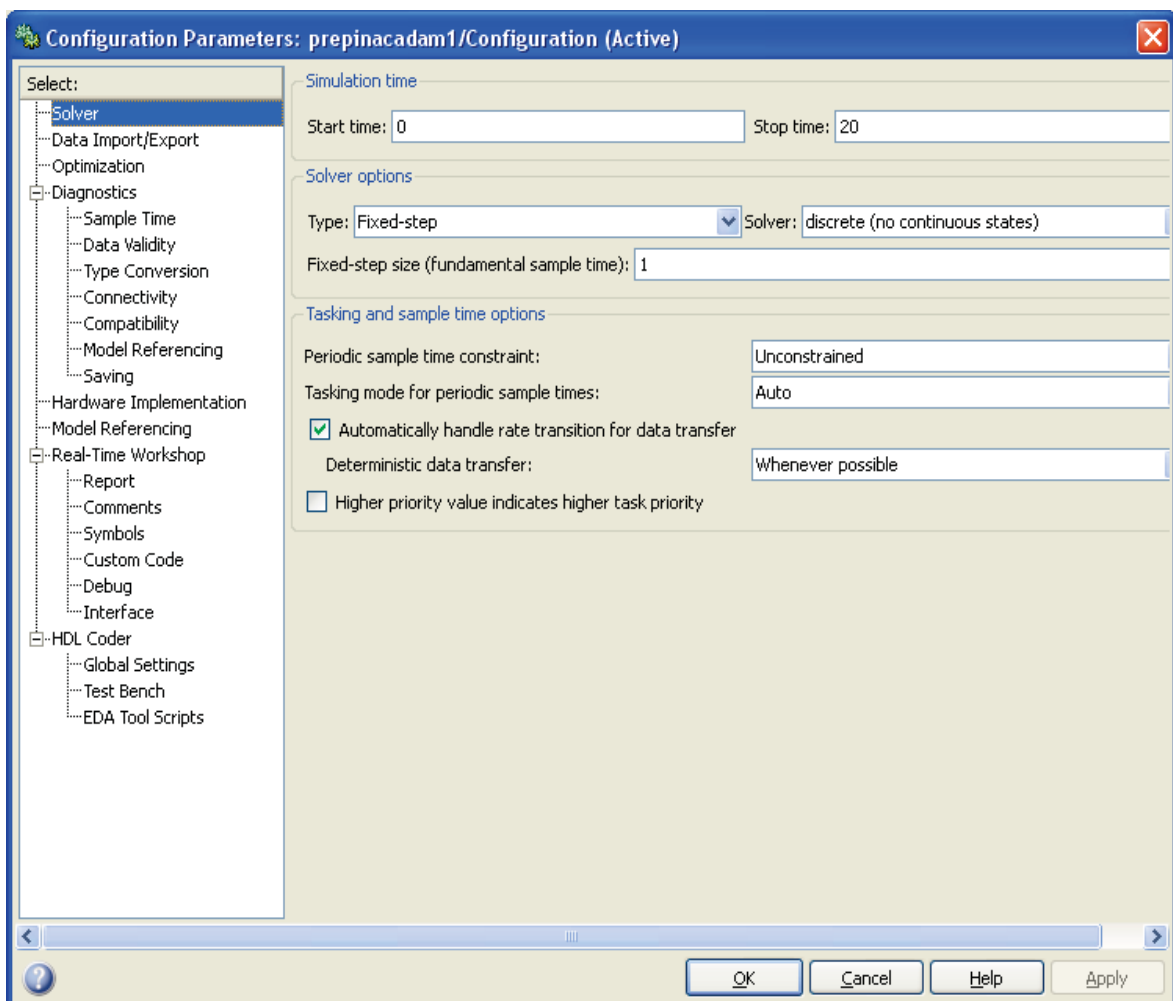
Někteří autoři [6], [7], [8] algoritmus propojení do výstupního portu řešili (modelově) pomocí *neuronové sítě*.



Obr.23: Řízení sepnutí portového spínače

6.2.3 Nastavení simulace

Simulink dává několik možností, jak ovlivňovat běžící simulaci. Blíže to upřesní následující okno:



Obr.24: Nastavení parametrů simulace

Jako nejdůležitější parametr je *řešitel (solver)*, který umožňuje nastavit přesnost numerických řešení, použitých v Simulinku pro jednotlivé komponenty systému. Implicitně je nastaven řešitel `ode45` (Dormand-Prince) což je vhodné pro řešení diferenciálních rovnic, například modelování přechodných jevů v elektrotechnice, odezvy systémů na spojité či kvazispojité signály, apod. Pro naši potřebu se hodí lépe simulace s *diskrétním časem*, která je vhodná pro simulaci číslicových systémů, kam přepínač rovněž patří. Nastavení se jmenuje `discrete (no continuous states)`. Uvedeme proto doporučené (nikoliv však striktní) parametry simulace, které použijeme i v laboratorní úloze. Řešíme pouze oddíl Solver:

Nejsou-li hodnoty takto nastaveny, nastavíme je podle obrázku. Ostatní nastavení ponecháme implicitní.

6.2.4 Výsledky simulace

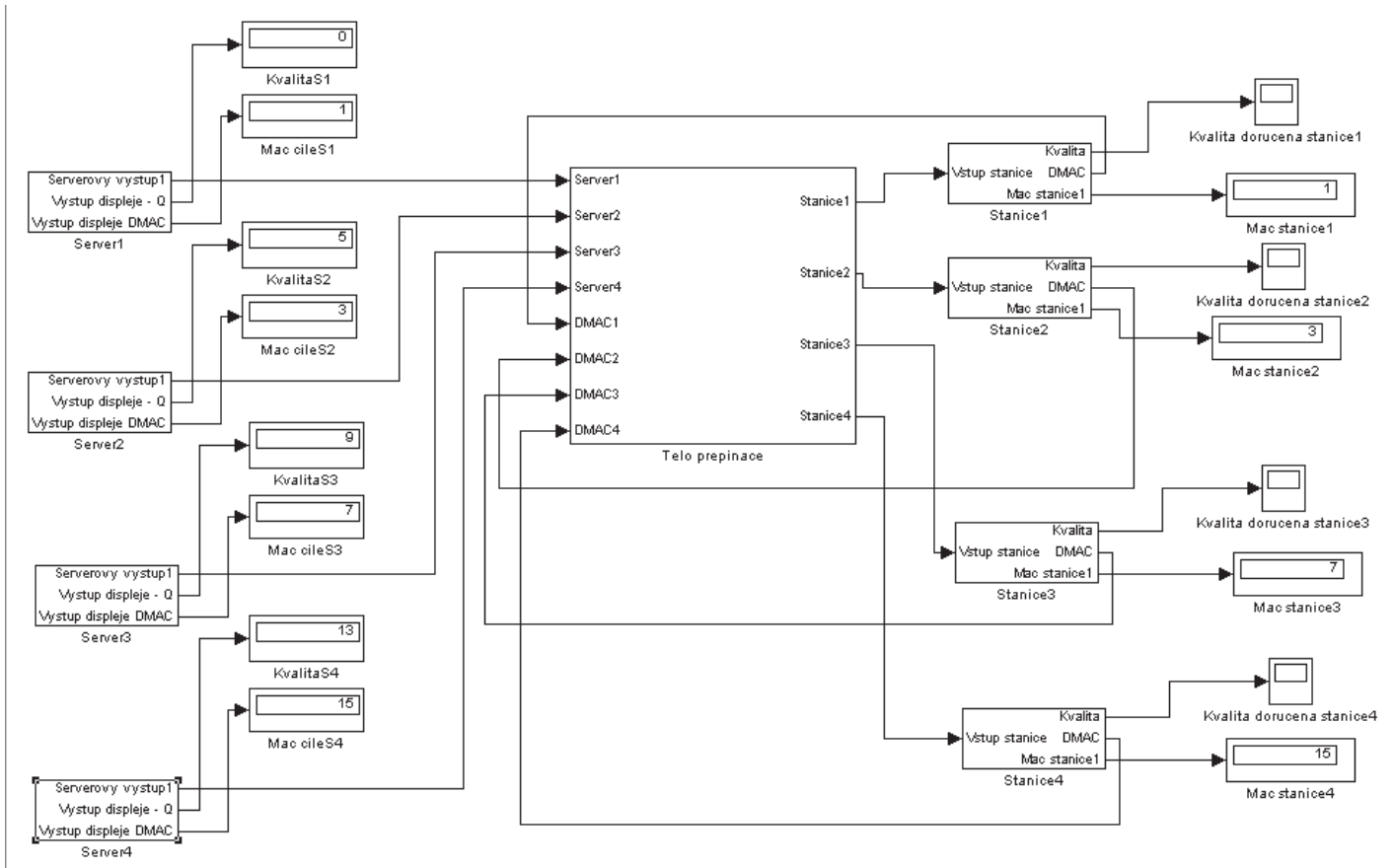
Nyní již můžeme spustit rovnou simulaci přenosu dat. Nejprve však nastavíme požadovaná data podle scénářů, které si dopředu připravíme. Zde uvedeme pouze některé z nich, protože vzájemné kombinace doručování dat ze serverů na stanice si můžeme stanovit v podstatě libovolně.

Scénář 1 – bod proti bodu, 4 druhy dat, 4 příjemci

Pořadí serveru	Cílová MAC (dekadicky)	Kvalita (dekadicky)	Stav doručení rámce
1.	1	0 – data	Náležitě
2.	3	5 - databáze	Náležitě
3.	7	9 – VoIP	Náležitě
4.	15	13 – IPTV	Náležitě

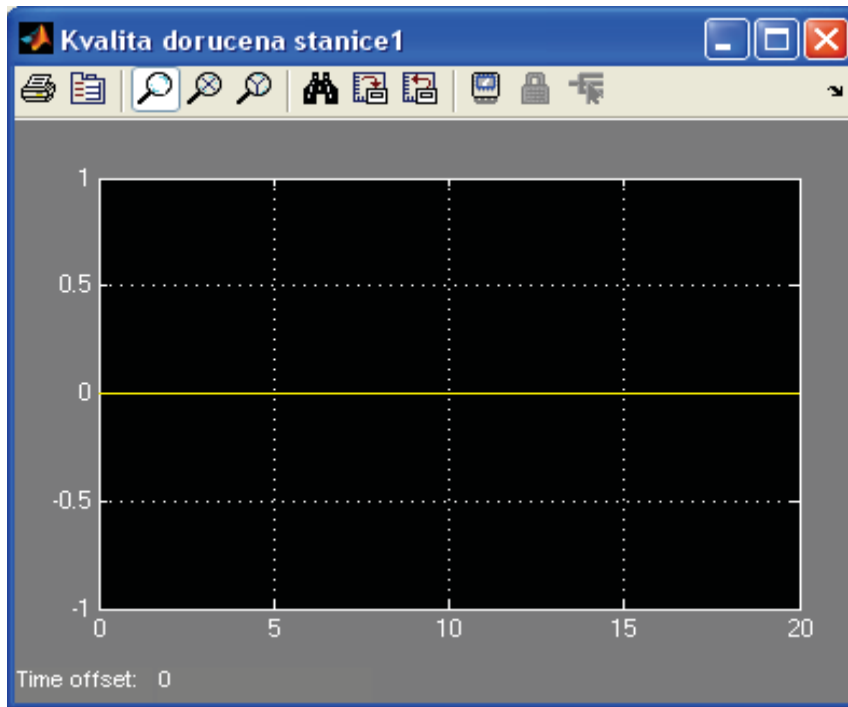
Tab.3: Scénář 1

Stav ukazatelů modelu po simulaci:

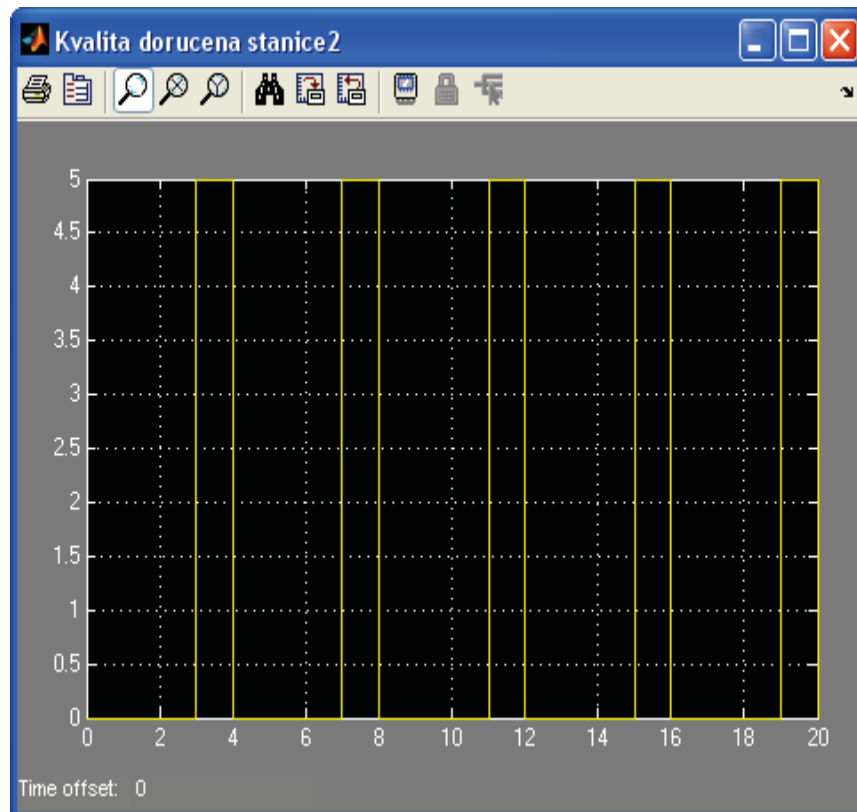


Obr.25: Model po simulaci prvního scénáře

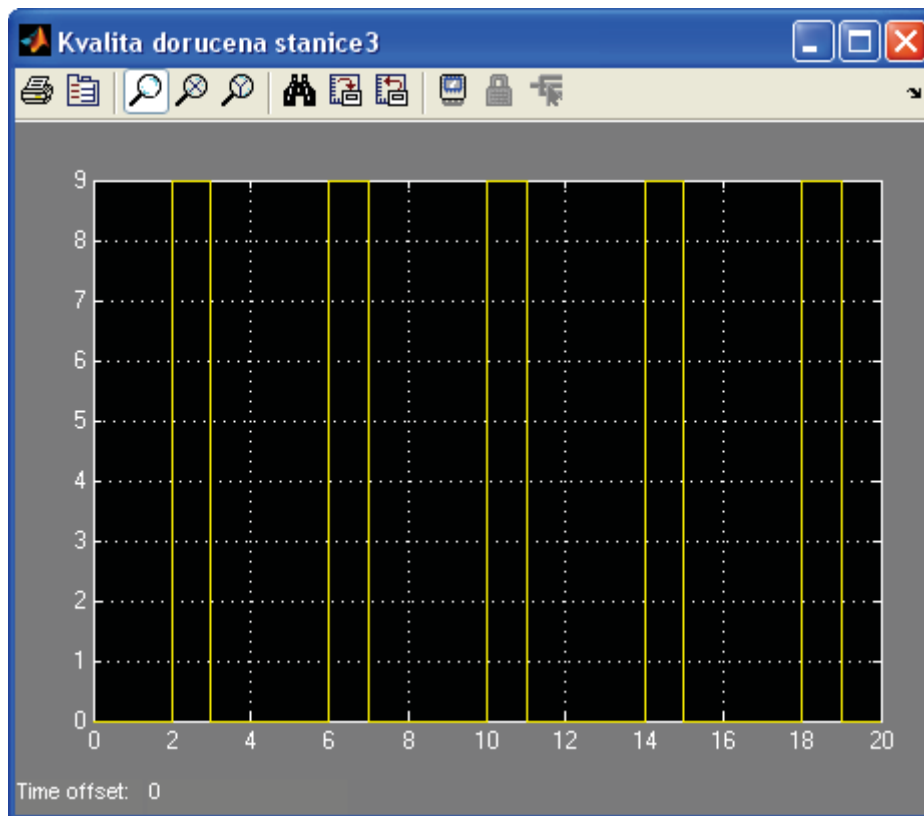
Doručená data po simulaci:



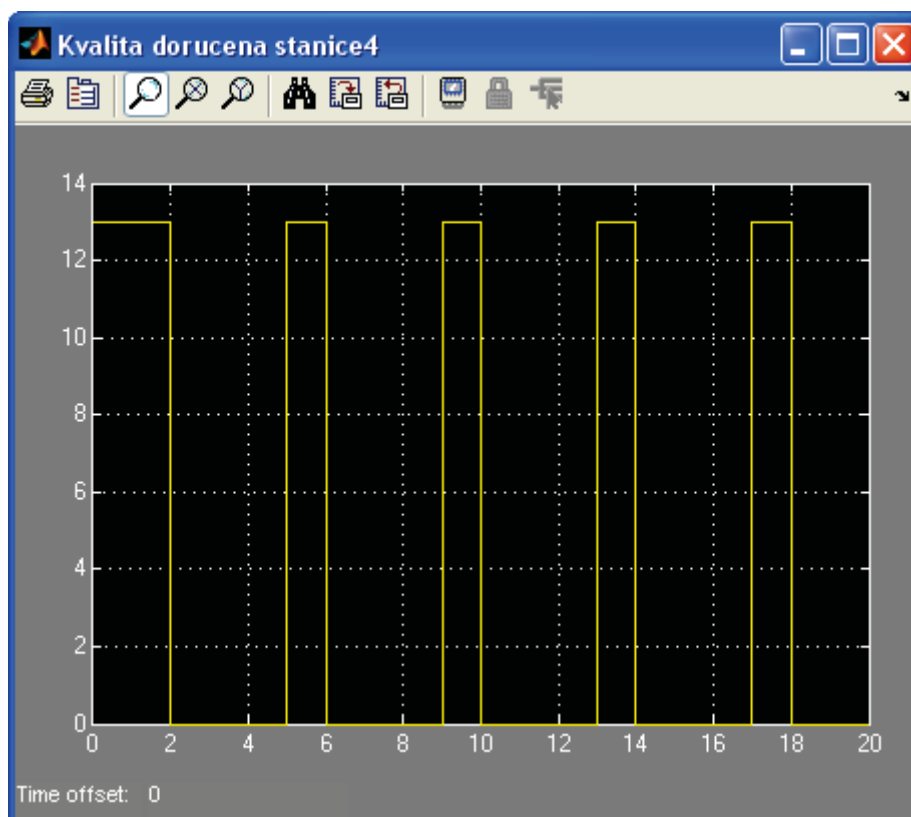
Obr.26: Parametr Q doručených dat na stanici 1 v závislosti na čase. $Q=0$ (Data)



Obr.27: Parametr Q doručených dat na stanici 2 v závislosti na čase. $Q=5$ (Databáze)



Obr.28: Parametr Q doručených dat na stanici 3 v závislosti na čase. $Q=9$ (VoIP)



Obr.29: Parametr Q doručených dat na stanici 4 v závislosti na čase. $Q=13$ (IPTV)

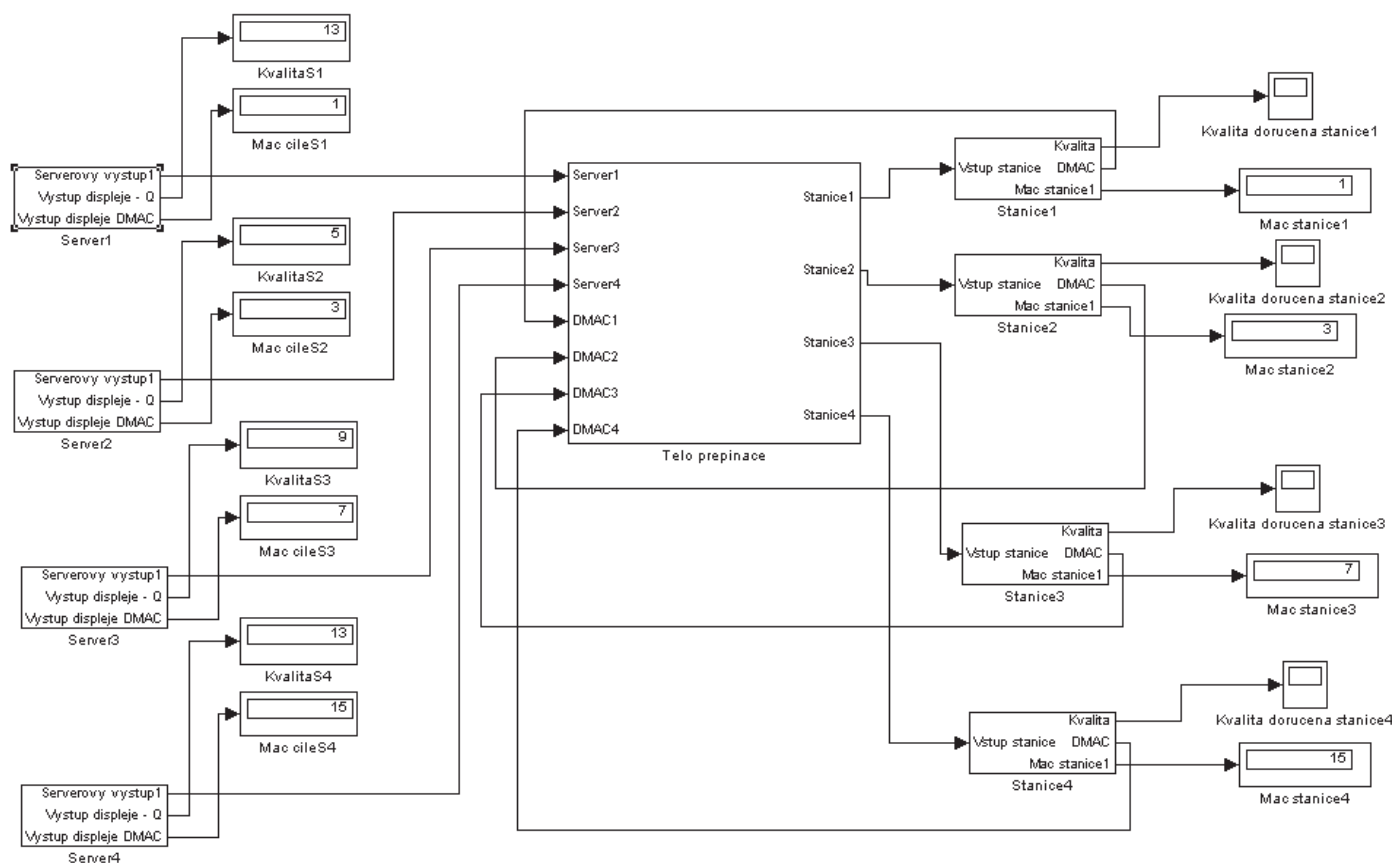
Zhodnocení výsledků:

Přepínač správně identifikoval data a doručil je na cílové porty podle MAC adres. Protože každý typ dat šel do jiného výstupu, doručení proběhlo bez konfliktů i kolizí.

Scénář 2 – bod proti bodu, 3 druhy dat, 4 příjemci

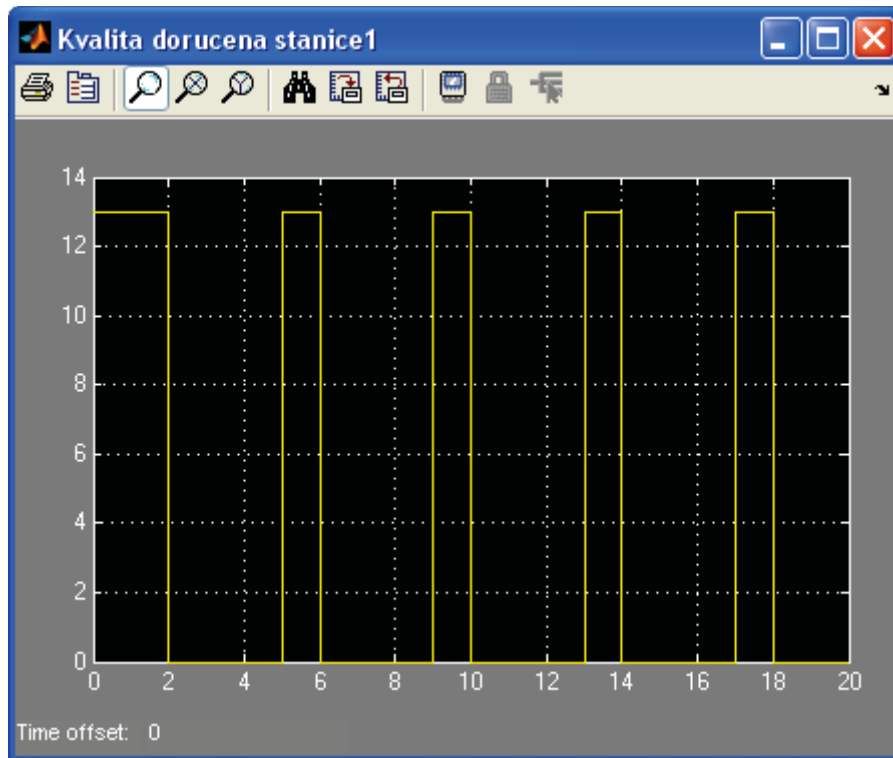
Pořadí serveru	Cílová MAC (dekadicky)	Kvalita (dekadicky)	Stav doručení rámce
1.	1	13 – IPTV	Náležitě
2.	3	5 - databáze	Náležitě
3.	7	9 – VoIP	Náležitě
4.	15	13 – IPTV	Náležitě

Tab.4: Scénář 2

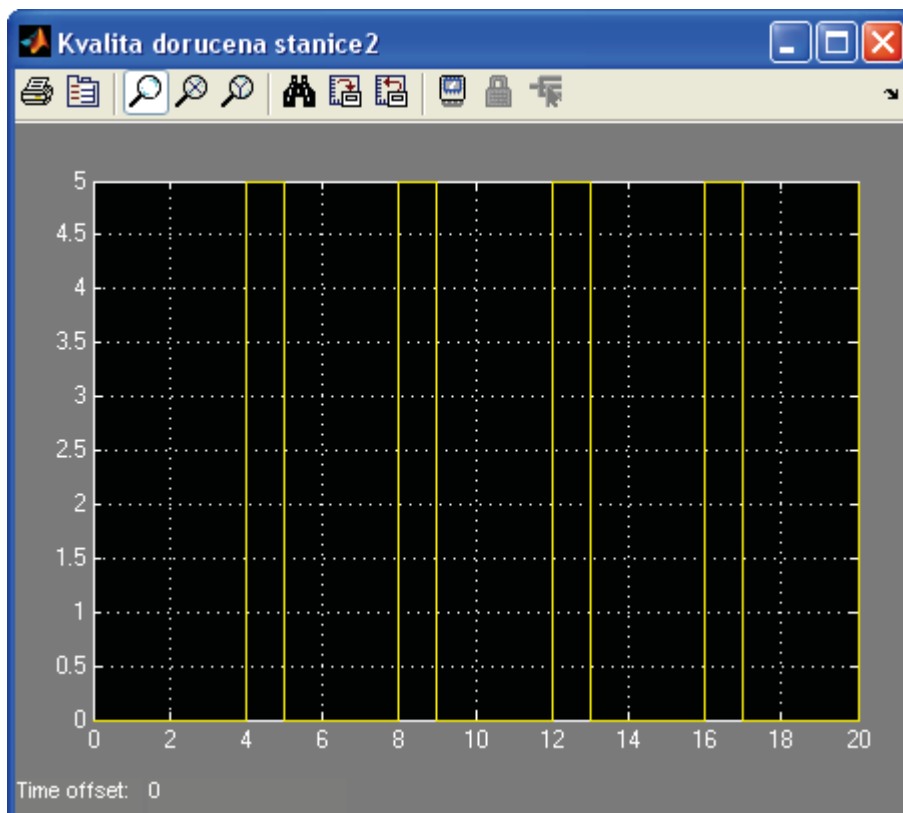


Obr.30: Model po simulaci druhého scénáře

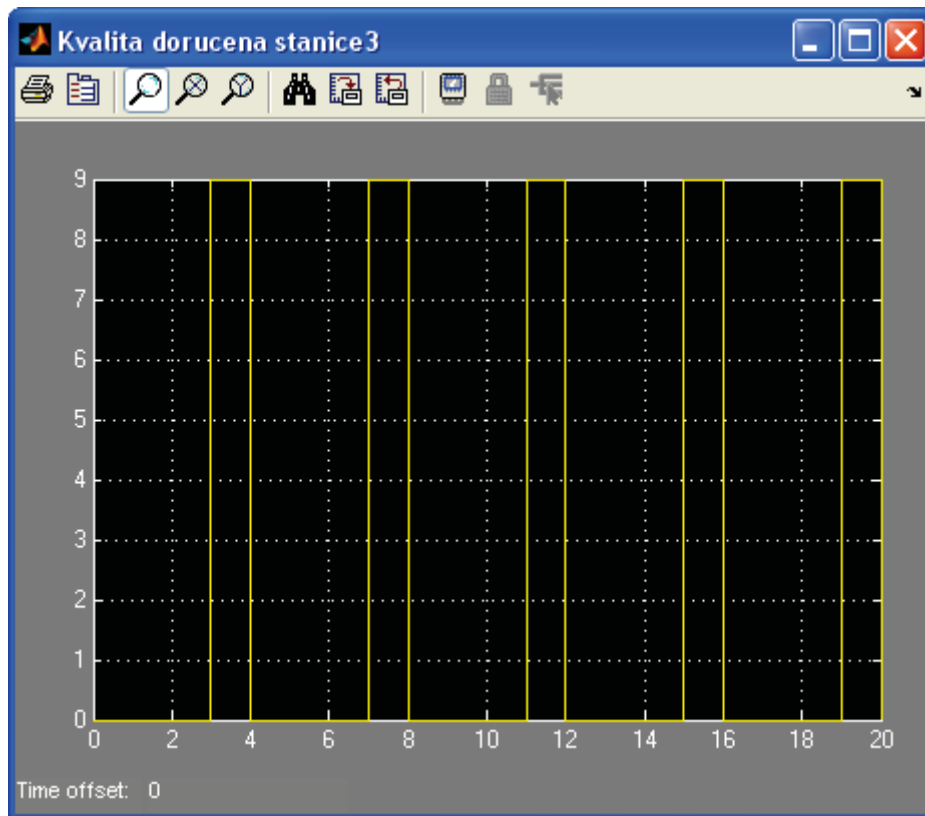
Doručená data po simulaci:



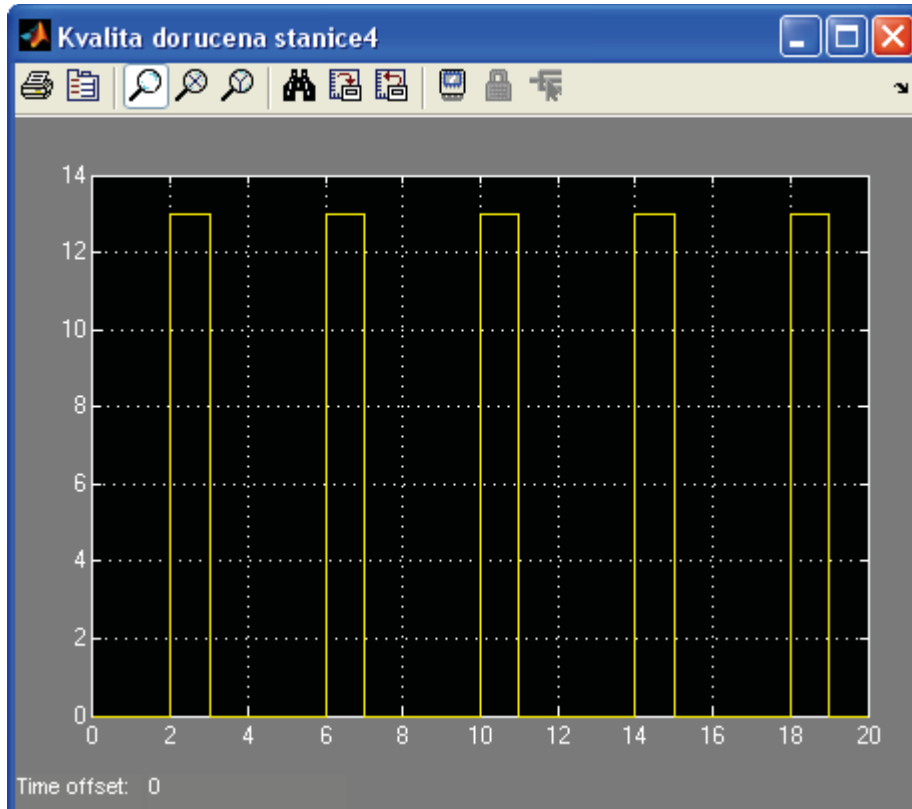
Obr.31: Parametr Q doručených dat na stanici 1 v závislosti na čase. Q=13 (IPTV)



Obr.32: Parametr Q doručených dat na stanici 2 v závislosti na čase. Q=5 (Databáze)



Obr.33: Parametr Q doručených dat na stanici 3 v závislosti na čase. $Q=9$ (VoIP)



Obr.34: Parametr Q doručených dat na stanici 4 v závislosti na čase. $Q=13$ (IPTV)

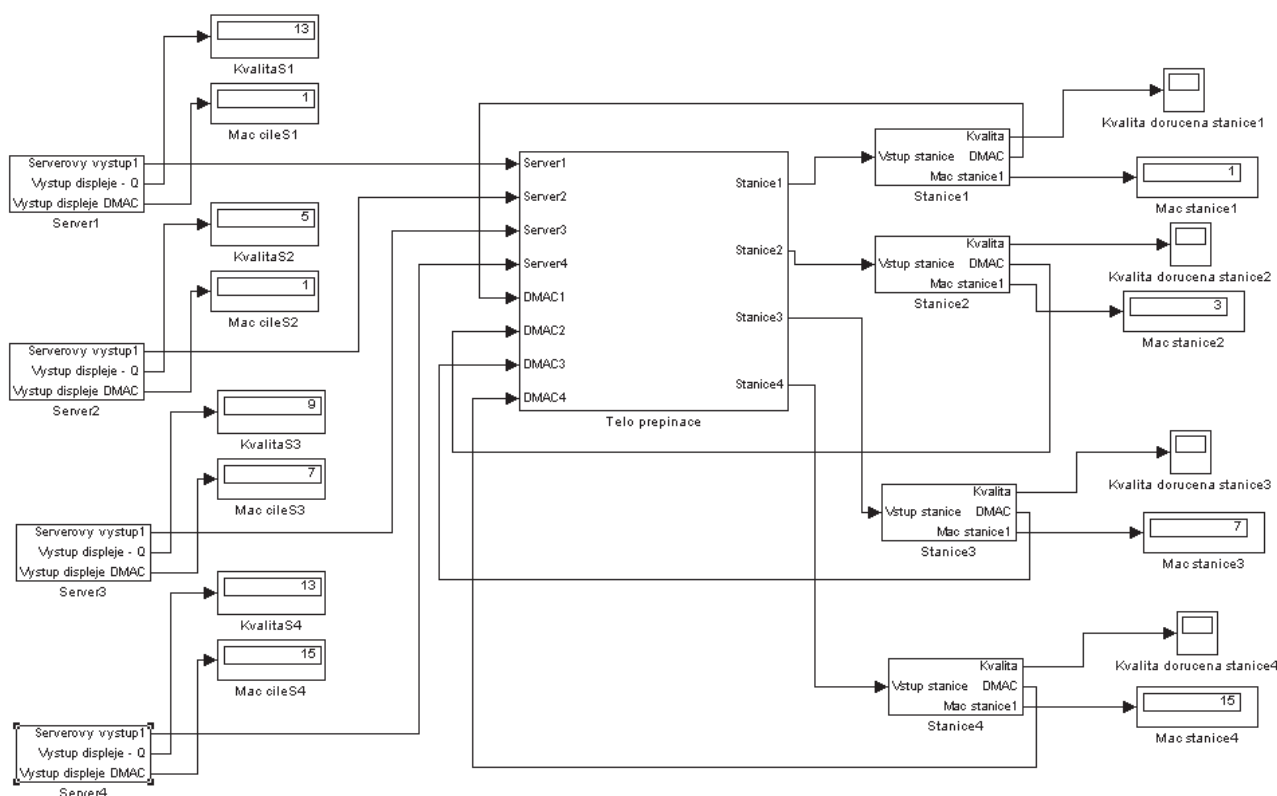
Zhodnocení výsledků:

Přepínač správně identifikoval data a doručil je na cílové porty podle MAC adres. Ke kolizi nedošlo, stanici, které nepatřila žádná data, neposlal nic.

Scénář 3 – Dva servery doručují jedné stanici, ostatní bod proti bodu, 4 druhy dat, 3 příjemci

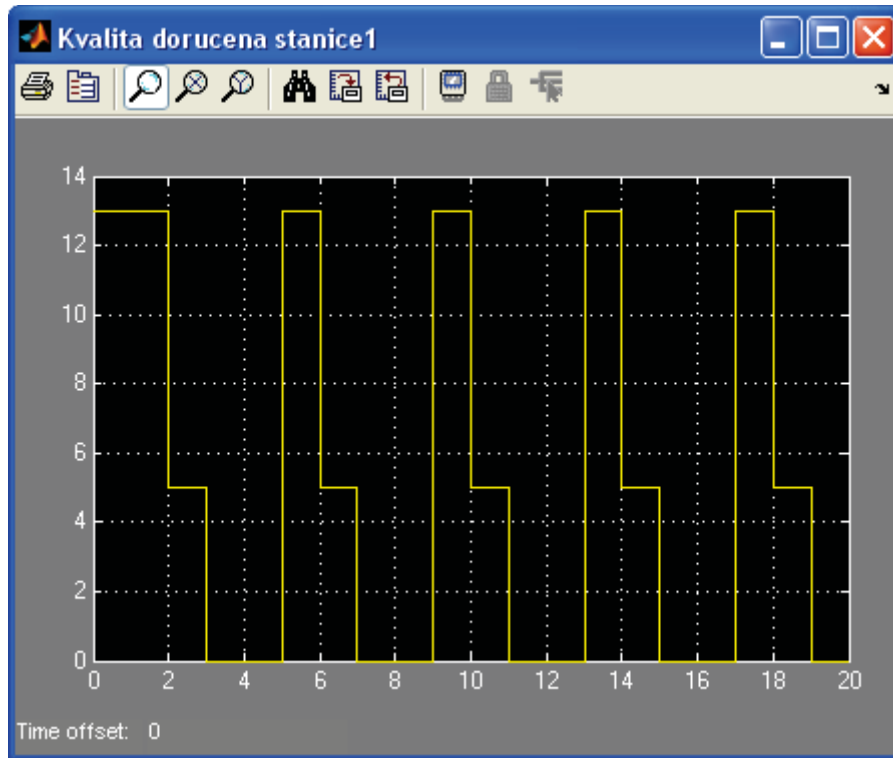
Pořadí serveru	Cílová MAC (dekadicky)	Kvalita (dekadicky)	Stav doručení rámce
1.	1	13 – IPTV	Náležitě
2.	1	5 - databáze	Náležitě
3.	7	9 – VoIP	Náležitě
4.	15	13 – IPTV	Náležitě

Tab.5: Scénář 3

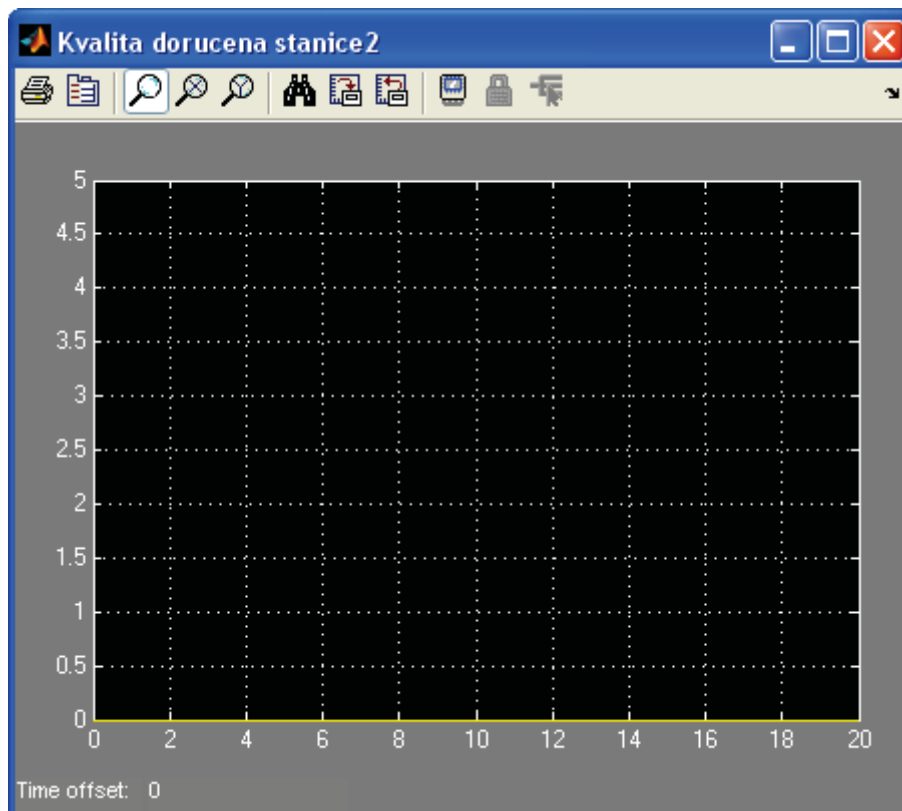


Obr.35: Model po simulaci třetího scénáře

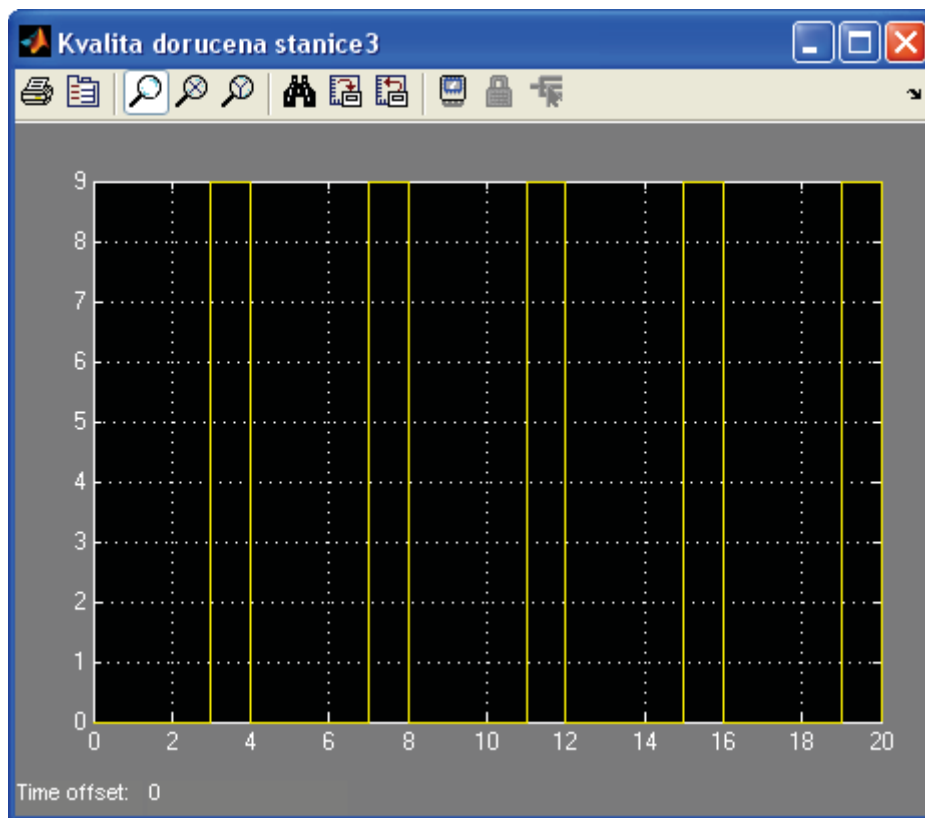
Doručená data po simulaci:



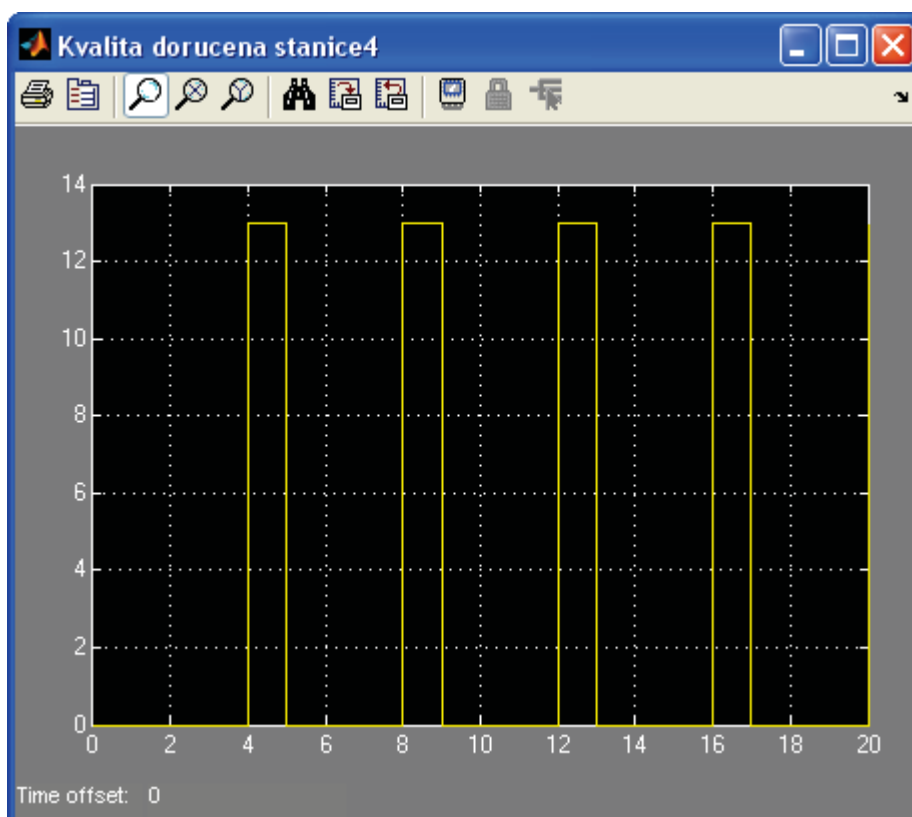
Obr.36: Parametr Q doručených dat na stanici 1 v závislosti na čase. Q=13 (IPTV) a Q=5 (Databáze)



Obr.37: Stanici 2 nebyla doručena žádná data – náležité



Obr.38: Parametr Q doručených dat na stanici 3 v závislosti na čase. $Q=9$ (VoIP)



Obr.39: Parametr Q doručených dat na stanici 4 v závislosti na čase. $Q=13$ (IPTV)

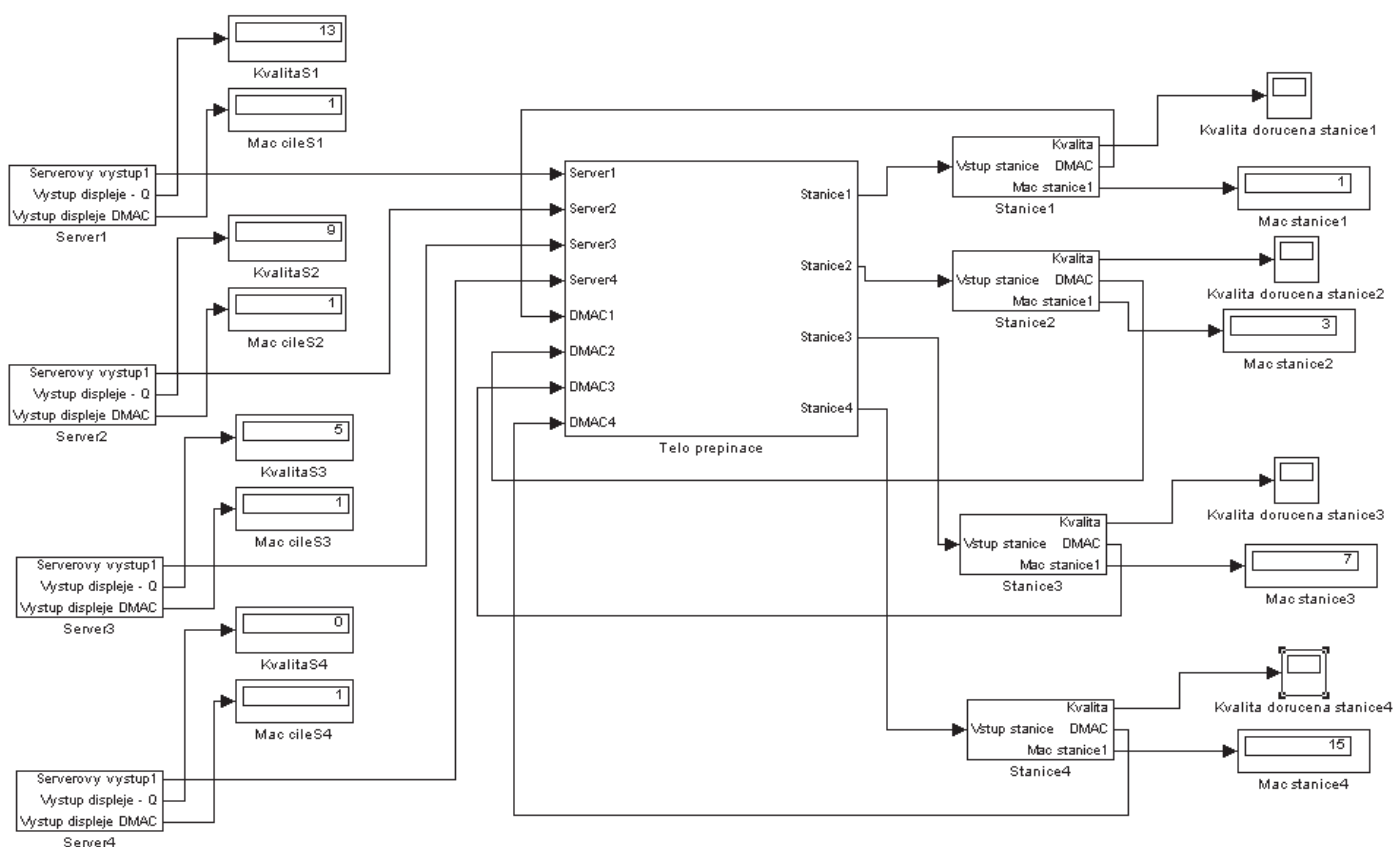
Zhodnocení výsledků:

Přepínač správně identifikoval data a doručil je na cílové porty podle MAC adres. Při doručování stanici 1 respektoval kvalitu služeb a doručoval sestupně dle parametru Q. Ke kolizi nedošlo, stanici, které nepatřila žádná data, neposlal nic.

Scénář 4 – Všechny servery doručují jedné stanici 4 druhy dat

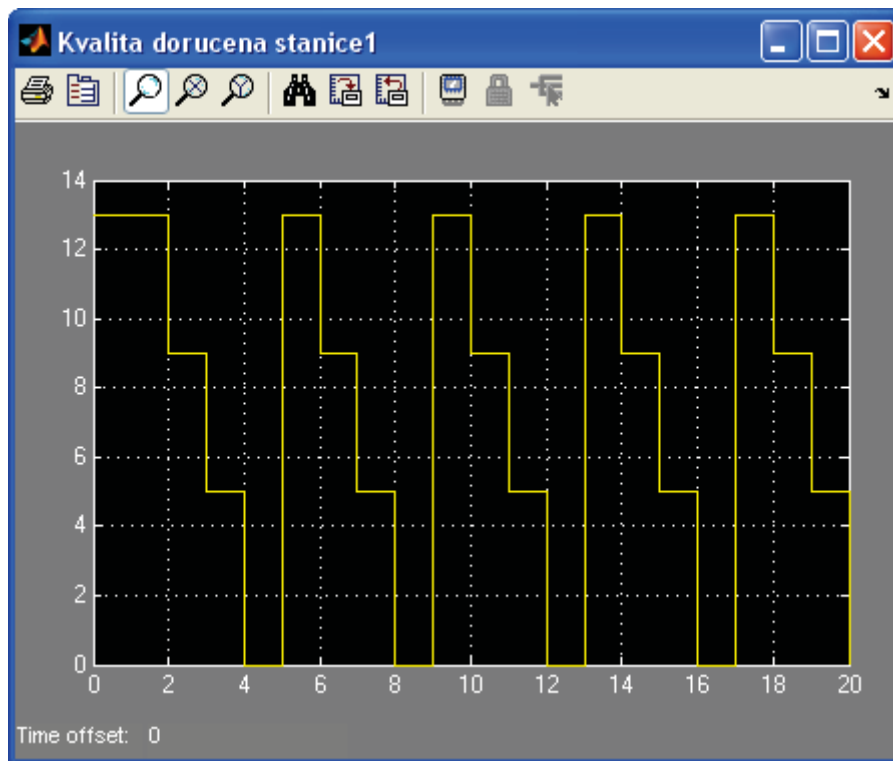
Pořadí serveru	Cílová MAC (dekadicky)	Kvalita (dekadicky)	Stav doručení rámce
1.	1	13 – IPTV	Náležitě
2.	1	9 - VoIP	Náležitě
3.	1	5 – Databáze	Náležitě
4.	1	0 – Data	Náležitě

Tab.6: Scénář 4

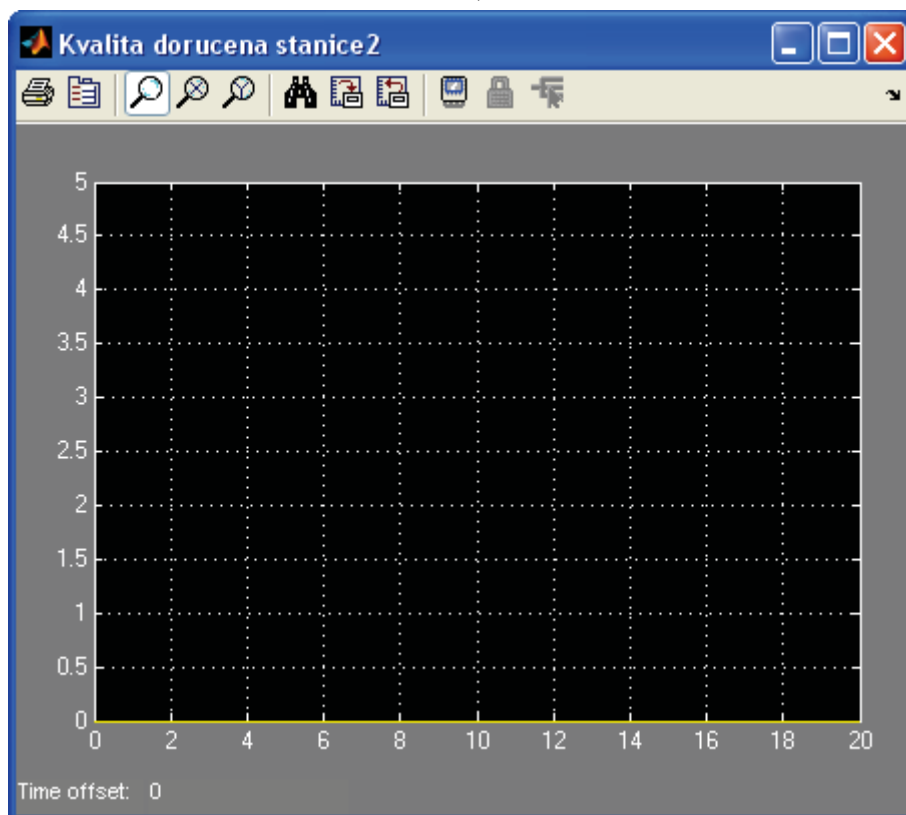


Obr.40: Model po simulaci čtvrtého scénáře

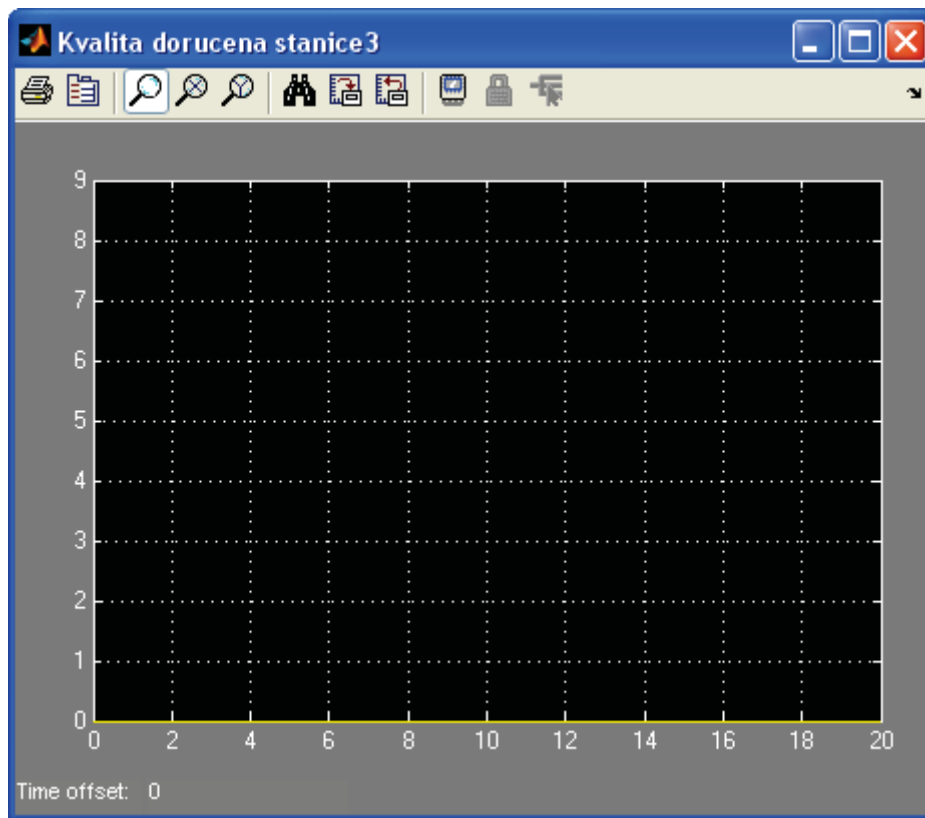
Doručená data po simulaci:



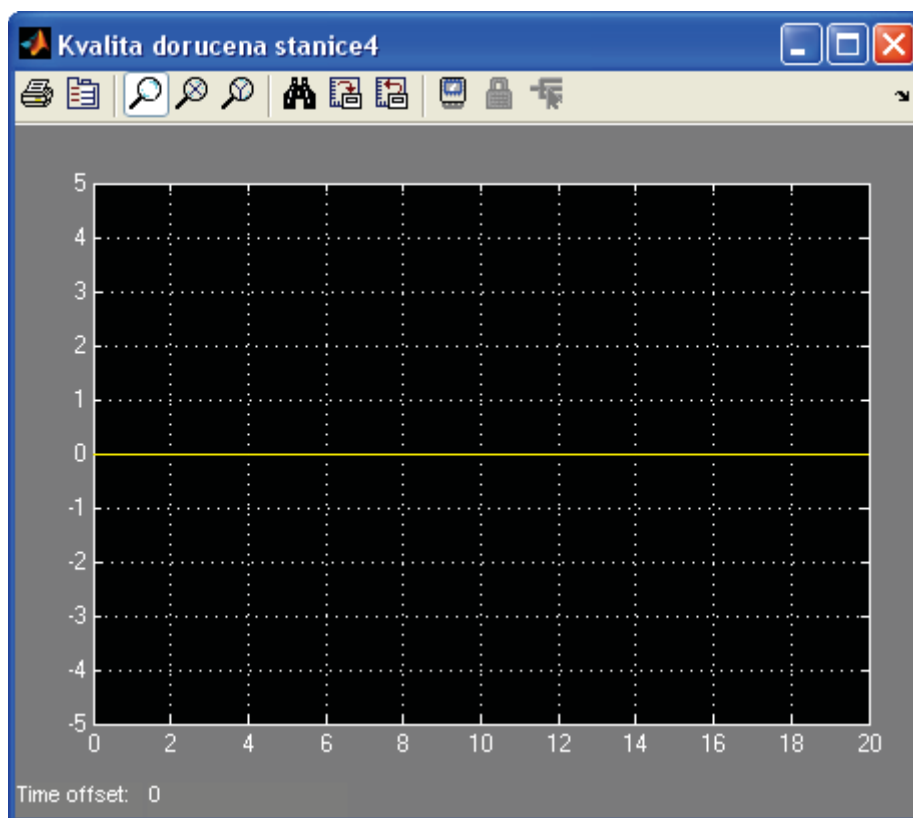
Obr.41: Parametr Q doručených dat na stanici 1 v závislosti na čase. $Q=0, 5, 9, 13$, (Všechny typy dat)



Obr.42: Stanici 2 nebyla doručena žádná data – náležité



Obr.43: Stanici 3 nebyla doručena žádná data – náležité



Obr.44: Stanici 4 nebyla doručena žádná data – náležité

Zhodnocení výsledků:

Přepínač správně identifikoval data a doručil je na jediný cílový port (stanice s MAC adresou 1) podle MAC adres. Při doručování respektoval kvalitu služeb a doručoval sestupně dle parametru Q. Ke kolizi nedošlo, stanicím, kterým nepatřila žádná data, neposlal nic.

Závěrečné zhodnocení simulace:

Simulace proběhly podle očekávaných předpokladů. Dalšími možnostmi je optimalizace algoritmů uvnitř přepínače. Rovněž použité komponenty z knihoven Simulinku můžeme dále nahrazovat a optimalizovat; činíme to však tak, aby výsledek byl co nejnázornější. Model nemá v žádném případě suplovat skutečný přepínač, nýbrž nastínit problematiku přepínání rámců v síti Ethernet a nastínit vhodnost optimalizovaného přepínače, podporujícího kvalitu služeb. A ještě na závěr zdůrazňujeme: Rozhodnutí, jaký typ přepínače se do konkrétní sítě použije, přísluší síťovému administrátorovi, popřípadě koordinátorovi ICT strategie firmy či instituce. To by měli být erudovaní odborníci, kteří se nejen v problematice dobře orientují, ale i dokážou zvolit nejvhodnější řešení pro konkrétní situaci

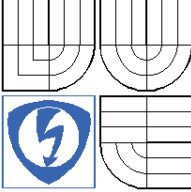
6.2.5 Východiska pro laboratorní úlohu

Tato zmínka v diplomové práci má za cíl diskutovat několik zásad pro realizaci laboratorního, popřípadě počítačového cvičení v předmětu Služby telekomunikačních sítí. V tomto předmětu jsou probírány jednotlivé služby, které se v konvergovaných sítích vyskytují. Navržený model musí být tedy vybaven *srozumitelným a didakticky přínosným návodem*. Je vhodné tento návod spojit i s protokolem, pokud je tento protokol vyučujícím požadován.

Model musí být upraven nebo pojat tak, aby student za 2x45 minut (zkušenost z kombinovaného studia) mohl splnit všechny body zadání. Do protokolu buď zapíše výsledky na předem připravené místo nebo přiloží scan (otisk) obrazovky, aktivního okna, simulace, apod. Dobré je na konec protokolu, možno až za vlastní závěr, položit několik otevřených otázek či zadat krátký, několikaotázkový test s volbou odpovědí. I když úloha může být celkově jednoduchá, měla by názorně přiblížit činnost přepínače v soudobé konvergované síti.

7 Laboratorní úloha

Vzor desek protokolu:

 VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ	Předmět:	Služby telekomunikačních sítí		
	Jméno:			
	Ročník:		Studijní skupina:	
	Spolupracoval:		Měřeno dne:	
Kontroloval:		Hodnocení:	Hodnoceno dne:	
Číslo úlohy:	Název úlohy: Optimalizace přepínače v konvergované síti			

Zadání:

Prostudujte činnost přepínače (switche) v síti Ethernet a srovnajte jej s rozbočovačem (hubem). Zařaďte oba prvky do správné vrstvy ISO/OSI a diskutujte, jak s těmito prvky prochází protokolová sada TCP/IP. Kdy je zajištěna transparentnost prvků a kdy ne v souvislosti s kvalitou služeb?

Na přiloženém modelu v Simulinku předved'te přenos rámce ze serveru na cílovou stanici v závislosti na MAC adrese a v závislosti na kvalitativním obsahu rámce. Odpovězte na dílčí otázky, vypracujte dílčí úkoly a případně zodpovězte dotazy vyučujícího. V závěru diskutujte vhodnost použití různých přepínačů v domácí a podnikové konvergované síti.

Teoretická příprava:

Rozbočovač je základním aktivním prvkem v síti s hvězdivou topologií. Pracuje na fyzické vrstvě ISO/OSI a z hlediska tekoucích dat je proto transparentní. Teoreticky vzato – kdybychom na libovolný port rozbočovače poslali *jakýkoli* digitální signál, měl by jej pouze zkontrolovat tvarově a zopakovat na ostatní porty. Jeho jedinou „inteligencí“ je to, že na příchozí port signál nedistribuuje. Dalším jeho posláním je impedanční přizpůsobení vysokofrekvenčního vedení a tím zabránění odrazům na tomto vedení.

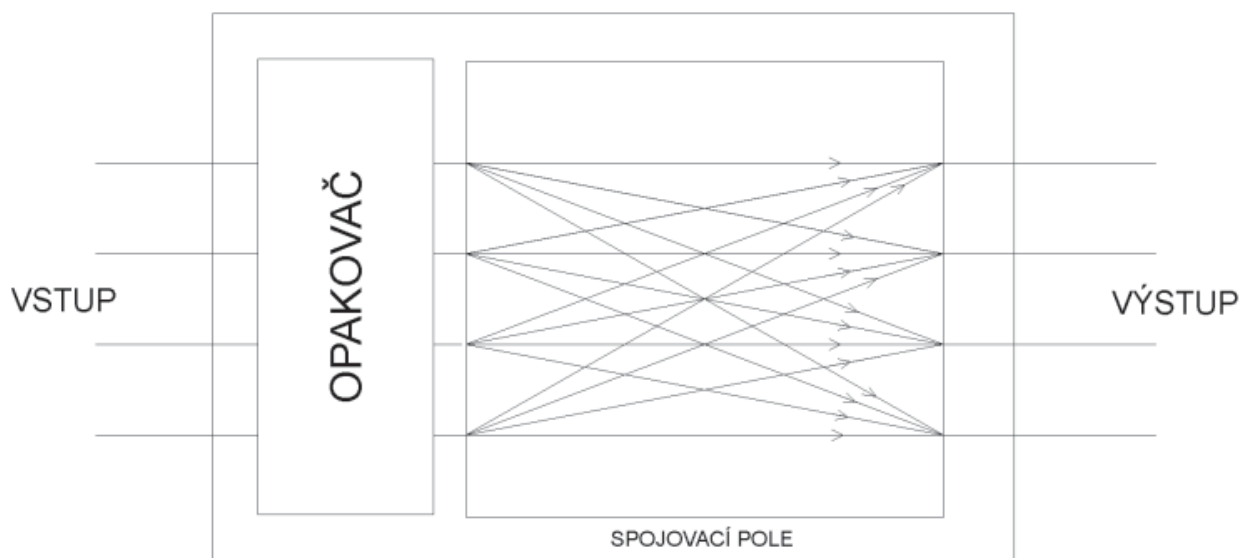
Stručný popis rozbočovače (hubu)

Rozbočovač je velmi jednoduché zařízení. Nijak neřídí provoz, který skrz něj prochází. Signál, který do něj vstoupí, je obnoven a vyslán všemi ostatními porty. Zpoždění je proto pouze 1 bit, takže na rozdíl od switche způsobuje hub nižší latenci. Na hubu jsou typicky signalizační LED diody, podle kterých se dá snadno zjistit vadné spojení.

Síť opatřená rozbočovači je velice snadno zahlitelná, protože všechna koncová zařízení jsou vlastně spojena paralelně. Z jednoho portu vede propojka na směrovač do další sítě - ven. Proto, aby mohl Ethernet detekovat kolize, je počet rozbočovačů v síti omezen. Pro síť 10 Mbps je počet segmentů omezen na 5, tedy 4 rozbočovače mezi dvěma koncovými stanicemi. Pro Fast Ethernet (100 Mbps) je limit snižen na 3 segmenty, tedy 2 rozbočovače. Gigabitový Ethernet rozbočovače nepoužívá vůbec. Dalším podstatným rysem použití rozbočovače je to, že celý síťový segment opatřený rozbočovači může pracovat jen s jednou přenosovou rychlostí. To znamená, že pokud do čistě 100 Mbps sítě připojíme byť jediné koncové zařízení pracující s rychlostí 10 Mbps, pak všechna ostatní síťová rozhraní se této situaci přizpůsobí – přepnou se automaticky do režimu přenosu 10 Mbps. To se může stát brzdou v rozsáhlejší podnikové síti, kdy skupina

mnohportových rozbočovačů tuto situaci vnutí celé síti. Rovněž, pokud některé nekvalitní síťové rozhraní šumí nebo je průmyslově rušeno, pak se přes rozbočovač zašumí celý segment sítě. Stejná situace principiálně nastane i při zkratu datového kabelu nebo v zásuvce či v domovním datovém rozvodu, i když modernější a dražší rozbočovače tuto situaci dokázaly samy vyřešit virtuálním vypnutím provozu na poškozeném segmentu. V minulosti byly podobné závady příčinou těžkých poruch celých sítí, navíc jsou obtížně vyhledatelné. Uživatel šumícího zařízení se do sítě může připojovat náhodně a administrátoři pak marně a dlouhodobě závadu hledali postupným odpojováním částí sítě a četnými restarty rozbočovačů. Z těchto a mnoha dalších důvodů se rozbočovače dnes ani nevyrábějí, protože jsou nahrazeny inteligentnějšími přepínači pracujícími na linkové vrstvě ISO/OSI.

Činnost rozbočovače je schematicky znázorněna zde:



Obr.L1: Schematické zapojení ethernetového rozbočovače – hubu

Stručný popis přepínače (switche)

Pracuje na linkové vrstvě ISO/OSI. Jeho posláním je v podstatě totéž, co dělá rozbočovač – zopakovat signál a rozdělit jej na výstupní porty, ale již poněkud jiným způsobem. Linková vrstva umožňuje kvalitativně rozlišit tekoucí data – rámce, a to podle fyzické adresy (MAC). Tato adresa je, jak jsme uvedli, jedinečná pro každé síťové rozhraní. Podle toho, kde se požadované cílové zařízení nachází, přepínač propojí vstup s požadovaným výstupem – tedy bod proti bodu. V ostatních zásuvkách se tato (nepotřebná) data neobjeví. To má za následek výrazné snížení provozu v přímo nekomunikujících segmentech sítě a tím vlastně zvyšujeme determiničnost sítě (vzdor tomu, že celkově Ethernet je sítí stochastickou).

Multiplexer (rozdělovač datového toku) v přepínači je řízen algoritmem pracujícím s MAC adresami. Každé koncové zařízení připojené k přepínači se identifikuje svojí MAC adresou a tudíž přepínač okamžitě ví, kam datový provoz má poslat.

V podstatě existují tři základní typy těchto algoritmů:

- Store and forward (Ulož a poté přepni)
- On the fly (Přepínej za letu)
- Adaptive switching (Načítej a průběžně přepínej dle situace)

Nyní k jednotlivým přepínacím postupům podrobněji:

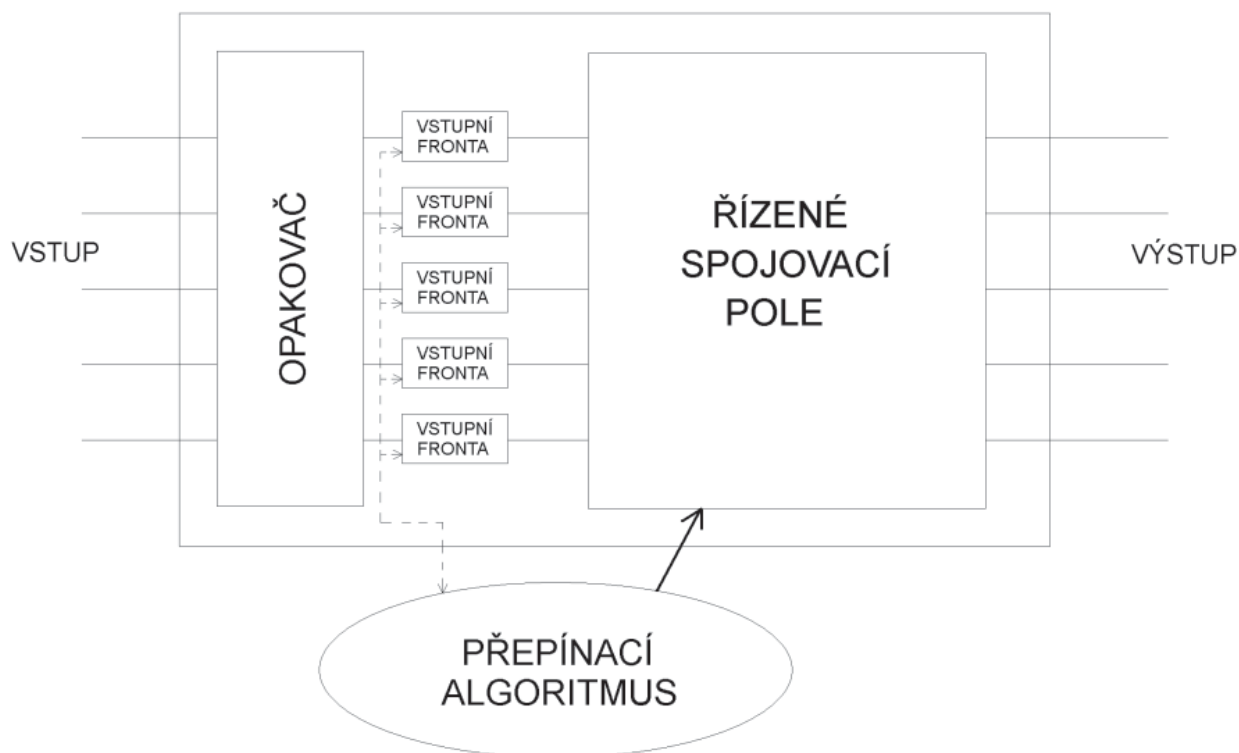
Store and Forward (Ulož a poté přepni) je klasický přepínací algoritmus, typický pro realizaci první generace přepínačů. Tento postup je použitelný nejenom v Ethernetu a spočívá v postupu, kdy přepínač načte do vstupní fronty celý rámec (paket) a provede kontrolu jeho konzistence. Pokud není konzistentní, dojde k zahození a znovunačtení. Pokud je vše v pořádku, pak dojde k porovnání cílové MAC rámce s MAC v tabulce přepínání, kterou přepínač neustále aktualizuje a podle toho, co má skutečně připojeno, pak to také propojí. Vzhledem k rychlosti, se kterou pracují dnešní přepínače, nemusí to být nutně na závadu, i když – obecně vzato – je tento algoritmus nejpomalejší, avšak také nejspolehlivější.

On the fly (Přepínej za letu). Tato metoda je naopak nejrychlejší, neboť odpadá kontrola konzistence rámce. Po identifikaci cíle je zdroj s cílem propojen – takřka v reálném čase. Tím se minimalizuje zpoždění, zvyšuje se propustnost sítě, avšak kontrola je v režii koncového (cílového) bodu. Hodí se proto pro sítě, které příliš nechybují nebo kde jde o přenos s vyššími nároky na rychlost, tedy ve finále pro data s vyššími hodnotami QoS.

Adaptive switching (Načítej a průběžně přepínej dle situace). Soudobé přepínače jsou schopny zběžně zkontrolovat rámec a pokud nějakým způsobem zjistí, že jde o prioritu, která „nesnese odklad“, pak přepnou „za letu“, jinak přepínají po rámcové kontrole. Rovněž se přepnou do režimu s kontrolou tehdy, jestliže zjistí zvýšenou chybovost dat (zvýšené požadavky na opakované přenosy). Je to kombinace předchozích postupů. Tento algoritmus je nejsostifikovanější a dokáže se adaptovat na situaci v síti podle chybovosti dat.

Přepínací algoritmy v celkové implementaci tvoří firmware přepínače a je know-how jednotlivých výrobců – v podstatě lze říci, že čím dražší přepínač, tím více optimalizačních rutin v přepínacích algoritmech poskytuje.

Přepínač má obecné zjednodušené schema následující:



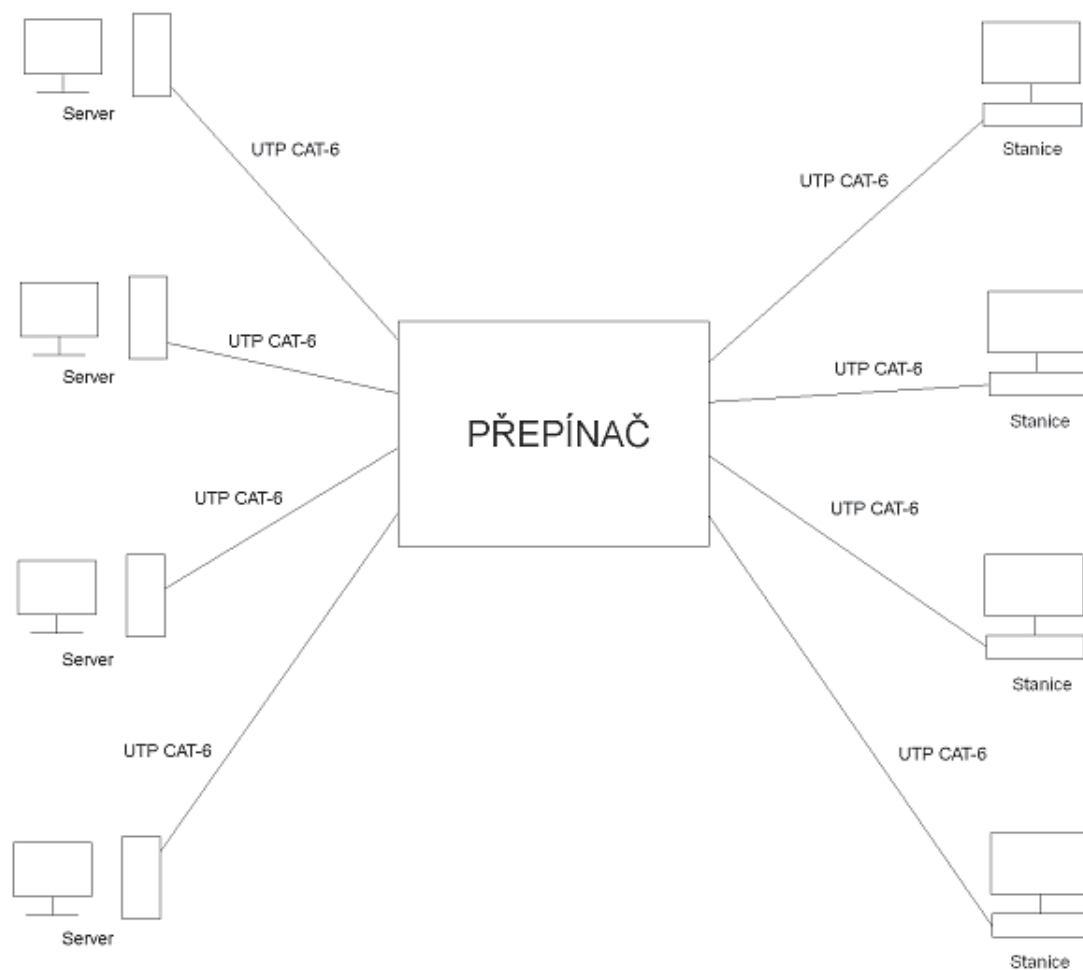
Obr.L2: Schematické zapojení ethernetového přepínače – switche

Existují i přepínače L3. Ty pracují někde mezi linkovou a síťovou vrstvou. Zohledňují kvalitativně proudící data podle priority, se kterou mají být přepnuta. Tímto typem se budeme v další části zabývat.

Příprava pracoviště:

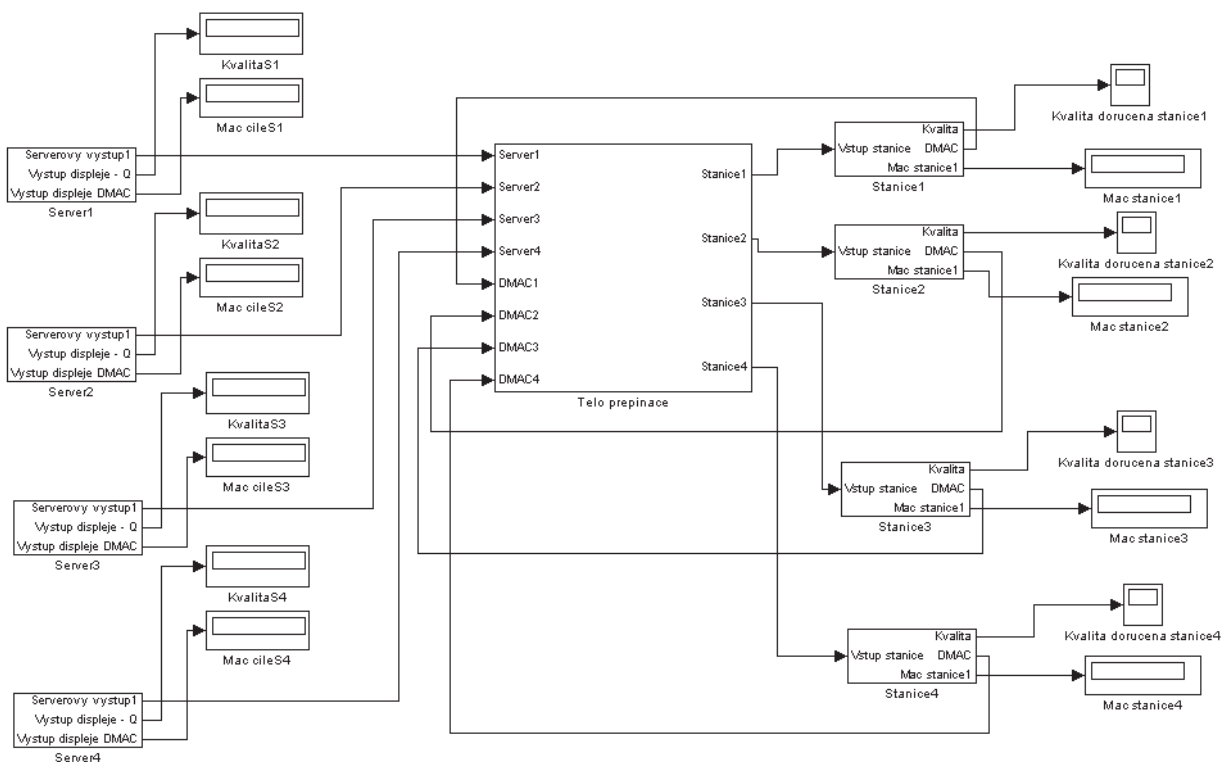
Spustíte na Vaší pracovní stanici program Matlab. Po náběhu základní konzoly zadejte příkaz `simulink` a vyčkejte na start zásuvného modulu. Překopírujte ze sítě nebo z příloženého CD-ROM soubor `prepinacl3.mdl` do pracovního adresáře – může být libovolný, avšak takový, kde máte právo číst a spouštět soubory. Z prostředí Simulinku tento soubor otevřete.

Základní schema zapojení pracoviště je následující:



Obr.L3: Logická topologie úlohy

Zapojení modelu je dle obrázku:

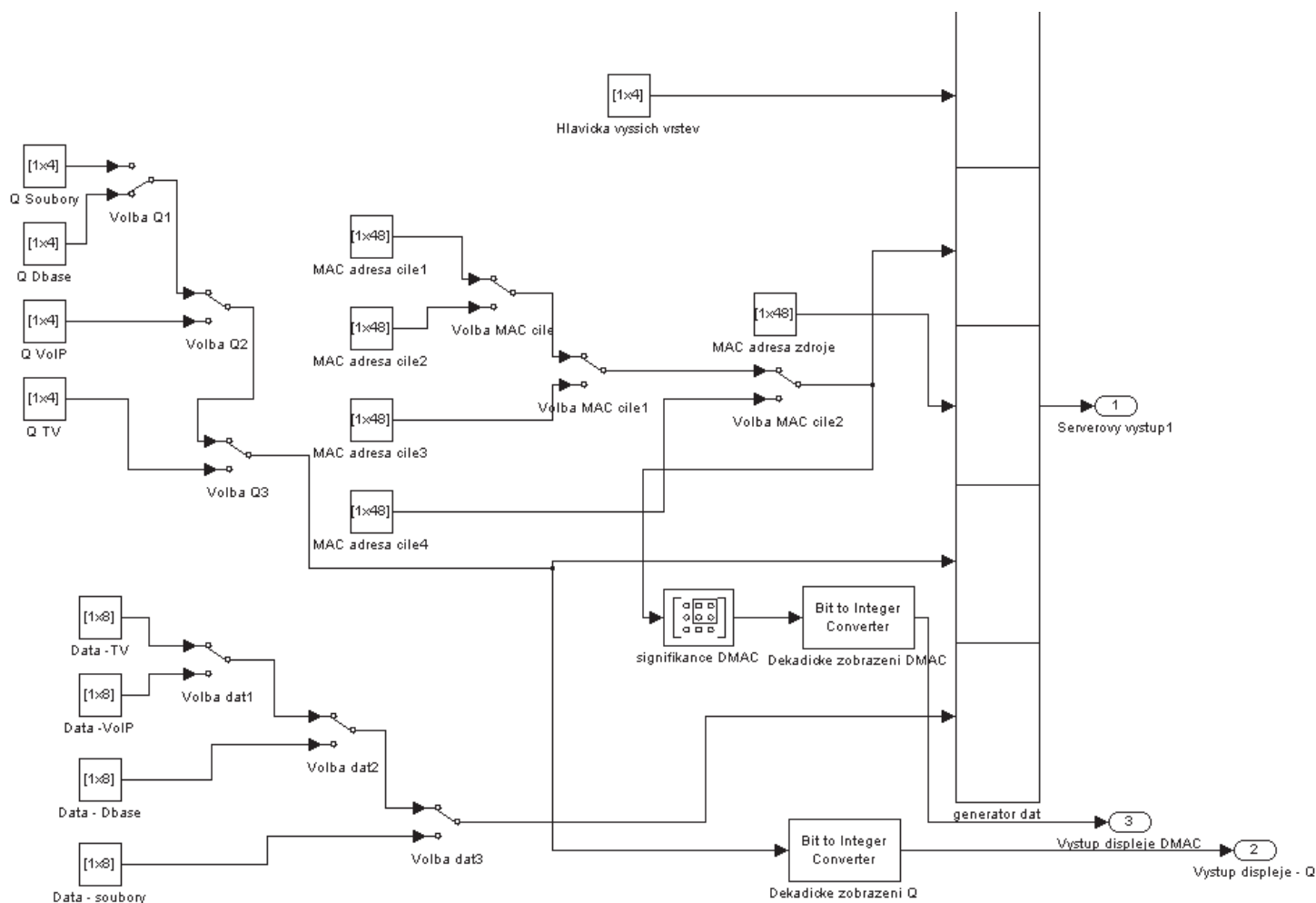


Obr.L4: Zapojení modelu přepínače

Po otevření souboru s modelem se objeví schema podle výše uvedeného obrázku. Model sestává ze 4 serverů, které jsou schopny doručovat data s příslušnou kvalitou k odpovídajícím stanicím, dále z vlastního přepínače a 4 stanic, které jsou cílovými uzly. Tvoří logickou topologii hvězda, bez dalších návazností, celek je tedy *uzavřeným systémem*. Servery i stanice mají displeje, zobrazující dekadické údaje signifikantních částí MAC adres a stanice navíc osciloskopy, které zobrazí příchozí údaj Q, tedy kvalitu služby – typ dat. Podle tohoto údaje můžeme lehce identifikovat propojení bod proti bodu, což je pro přepínač typické. Přepínač je řízen deterministicky MAC adresami cílového objektu. V praxi se k MAC adrese dospěje např. použitím protokolu ARP, zde je cílová Mac (označeno DMAC) nastavena apriori. Uvnitř přepínače se emuluje přepínací tabulka a řídí selektivní přepnutí na příslušný výstup.

Servery

Servery jsou tvořeny aditivním blokem, který sestavuje zjednodušený rámec podle podstatných a využitelných komponent. Vynechali jsme nepodstatné části, protože by entita rámce narostla do neúměrných binárních hodnot – přes 1500 bitů. Náš rámec má celkem 112 bitů pevných, i když samozřejmě skutečný rámec má svoji hodnotu násobně delší. Pro konfiguraci serverů slouží mechanické přepínače, které nastavují hodnotu parametru Q, cílovou MAC adresu a (formálně) i vlastní data. Ta lze dodatečně vyfiltrovat a rovněž zobrazit na displeji či osciloskopu. Vnitřní struktura serveru je následující:



Obr.L5: Vnitřní struktura serveru. Generátor dat spojí jednotlivé dílčí konstanty a proměnné do celistvého rámce, dlouhého 112 bitů

Poklepáním na mechanické přepínače lze servery konfigurovat a nastavovat tak různá vstupní data pro zpracování přepínačem.

Stanice

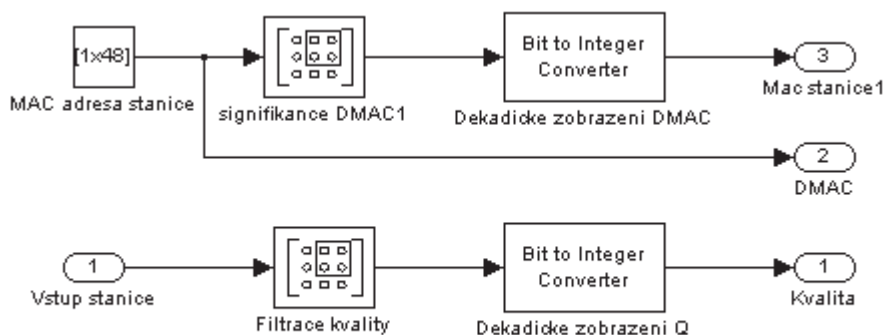
Stanice jsou tvořeny maticovými filtry, které z rámce filtrují signifikantní hodnoty pro zobrazení. Toto zobrazení jde ze staničního subsystému jednak na displej (Cílová MAC – zobrazeny poslední 4 bity. Zbylou sekvenci vyšších bitů nepoužíváme – je rovna nulám. V praxi tomu tak samozřejmě není; MAC adresa je 48 bitové číslo zobrazené hexadecimálně, avšak zde si můžeme situaci zjednodušit a zobrazit tento údaj dekadicky. Hodnoty Q a cílové MAC nabývají hodnot:

Typ dat	Hodnota parametru Q (binárně)	Hodnota parametru Q (dekadicky)
Televize (multimédia)	1101	13
Digitální telefonie VoIP	1001	9
Řízení báze dat	0101	5
Přenos souborů	0000	0

Hodnoty parametru Q podle typu dat

Stanice	MAC adresa binárně	MAC adresa dekadicky
Televize (multimédia)	0001	1
Digitální telefonie VoIP	0011	3
Řízení báze dat	0111	7
Přenos souborů	1111	15

Hodnoty MAC adres stanic



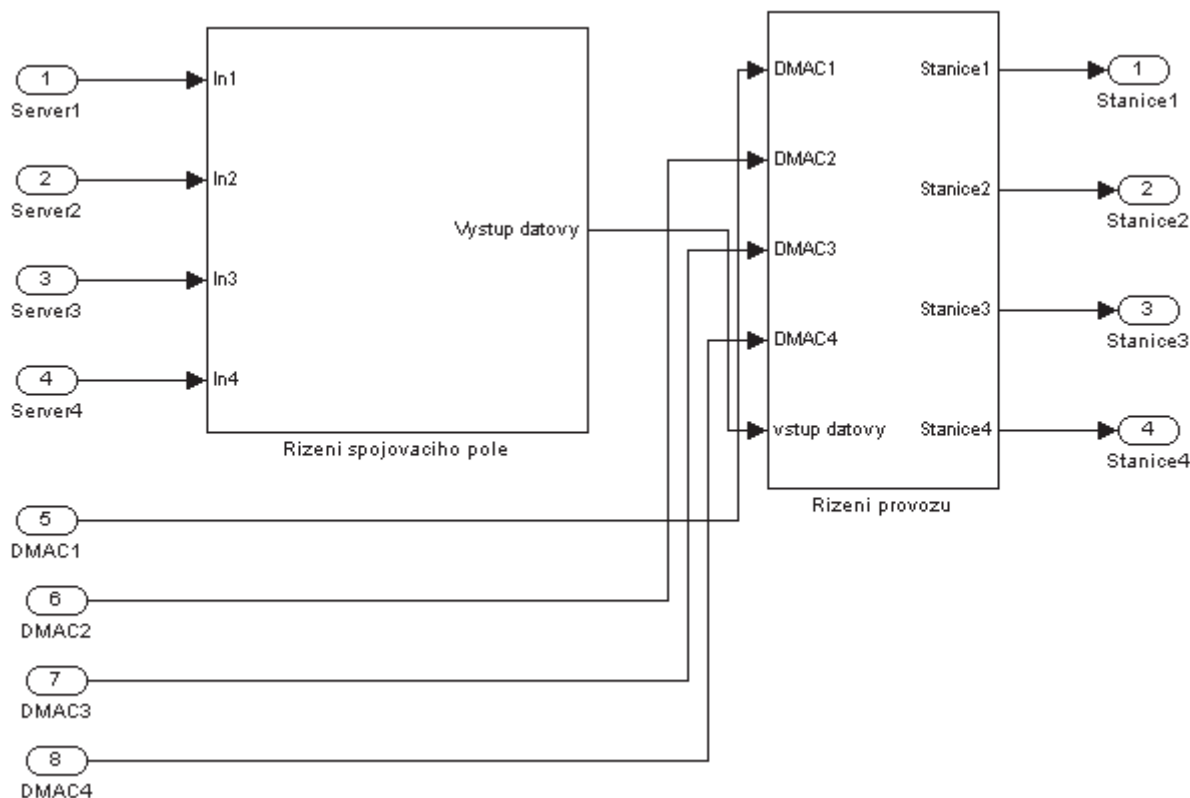
Obr.L6: Vnitřní zapojení subsystému Stanice.

Vnitřní struktura přepínače

Přepínač se skládá z několika subsystémů, které jsou vzájemně propojeny. Na hlavní vrstvě jsou to tyto komponenty:

- *Řízení spojovacího pole* – pro kvalitativní hodnocení proudících dat. Sem data prvotně vstupují. Subsystém má za úkol provést *analýzu vstupujících dat a rozhodnout o pořadí přepnutí*. Zde se simuluje skládání dat do fronty typu *FIFO (First In First Out)*, klasifikace podle priority – parametru Q , tedy v podstatě *třídění* a dále určuje pořadí přepínání jednotlivých rámců.
- *Řízení provozu*, což je vlastní sekvence přepínání, emulace přepínací tabulky. Tento subsystém pracuje s MAC adresami stanic. Srovnává tento údaj v příchozích datech se skutečností a podle toho, kde nalezne shodu, pošle rámec do příslušné stanice.

Situaci názorně popisuje následující obrázek:

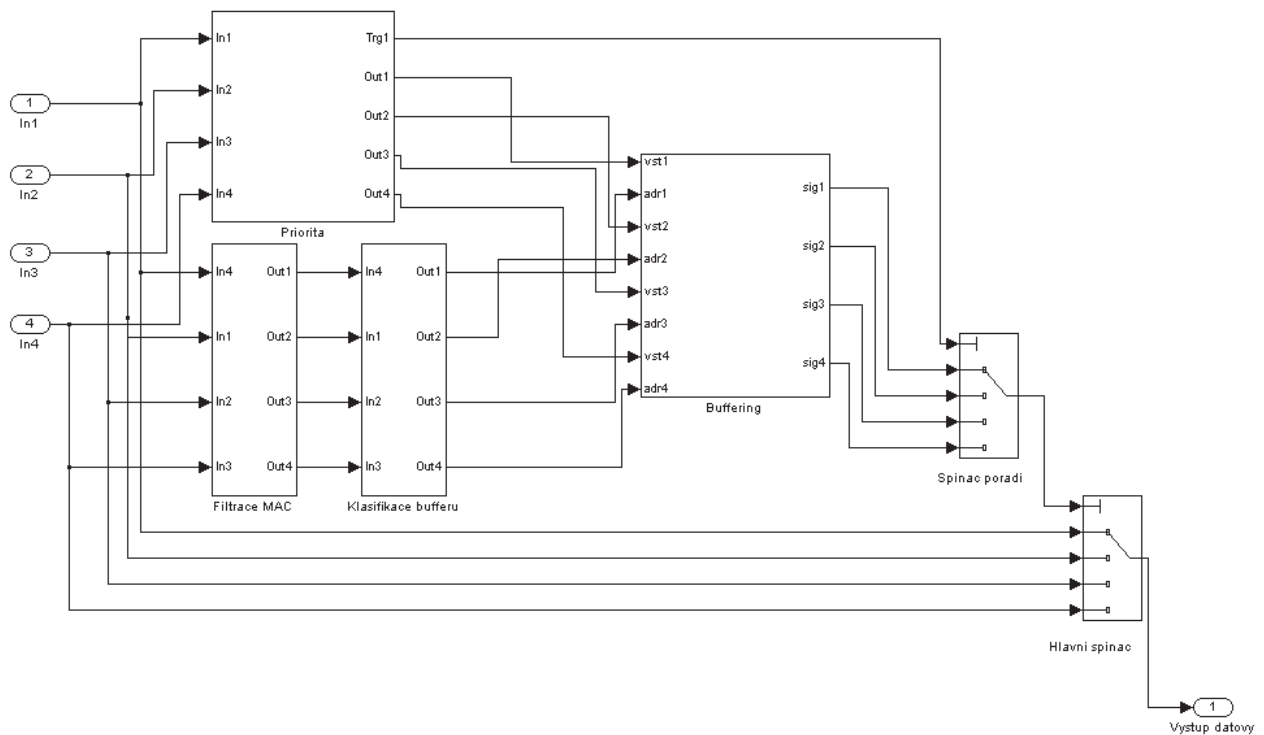


Obr.L7: Komponenty Řízení spojovacího pole a Řízení provozu

Řízení spojovacího pole

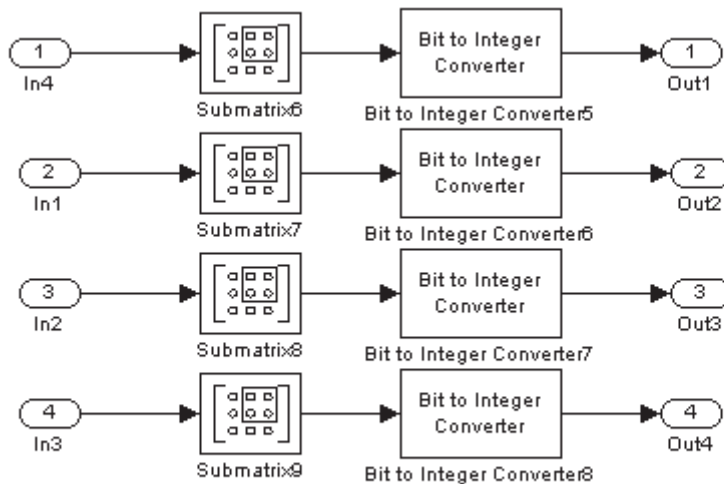
Je to subsystém, který má jako hlavní součásti klasifikaci rámce podle MAC adresy a podle parametru Q. MAC adresa rozhoduje, kam data budou poslána, parametr kvality Q potom kdy se tak stane, tedy v jakém pořadí. Zde je proto nezbytnou komponentou jednoduchý časovač, respektive spouštěč (trigger), který řídí pomocný přepínač. Tento přepínač vysílá do *spínače pořadí* posloupnost řídicích pulsů podle kvalitativního vyhodnocení dat. Pokud bychom zařadili do *hlavního spínače* přímo signál triggeru, pak by přepínač pracoval bez podpory kvality služeb – prostým kruhovým plánováním (round robin). Spínače jsou zapojeny kaskádně a data přenášena sériově. Vzhledem k nespojitosti spouštění musíme jednak nastavit vhodnou dobu a typ simulace (viz dále), jednak jsou zachycená data rovněž „vzorkována“ s frekvencí spouštění. Proto se jeví vhodnější je odchyťovat osciloskopem namísto displeje.

Situaci subsystému Řízení spojovacího pole zachycuje následující obrázek:



Obr.L8: Řízení spojovacího pole sestává ze 4 subsystémů. Řeší se zde priorita a bufferování přepínače, což je v podstatě algoritmus třídění

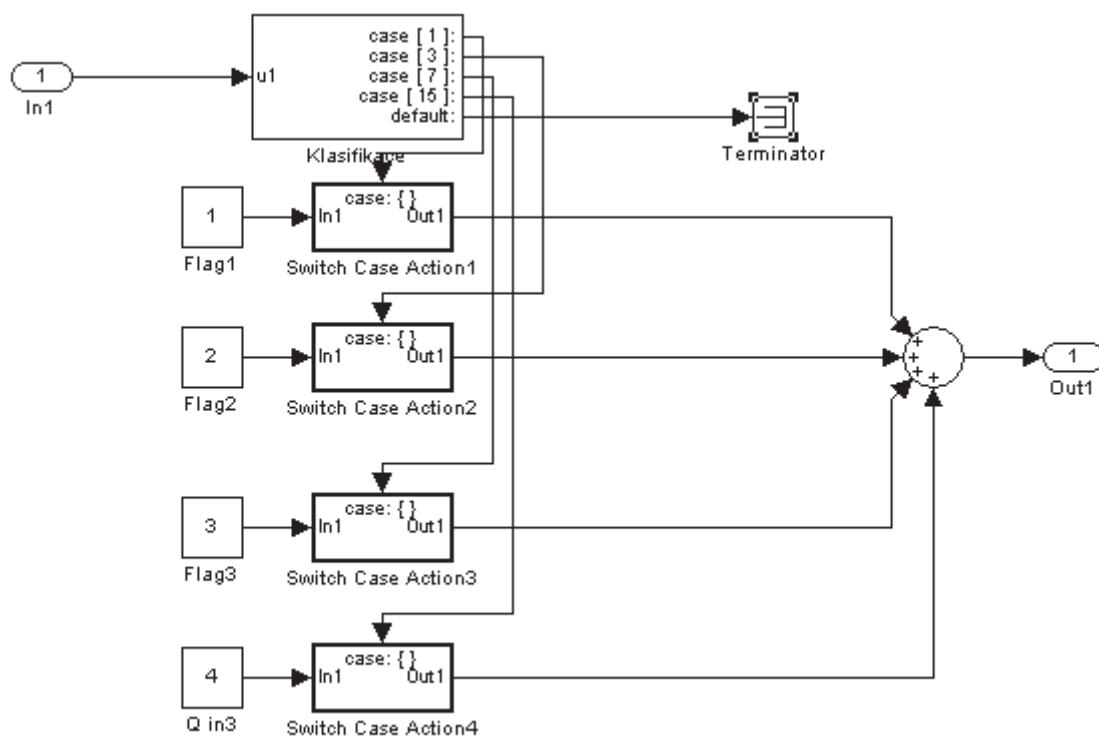
V Řízení spojovacího pole se data dělí a filtruje se cílová MAC adresa z rámce v subsystému Filtrace MAC, aby s ní bylo možno pracovat. Také se zde původní 48 bitové číslo ořízne na signifikantní nejnižší 4 bity, které jsme použili. Poté je toto číslo převedeno do dekadické podoby. Z tohoto subsystému jde tento údaj do Klasifikace bufferu, který vygeneruje signál pro hlavní spinae. Ten má hodnotu 1 až 4 dekadicky a bude ovládat v seřazeném pořadí hlavní spinae. Uvedme si pro přehled zapojení právě popsaných subsystémů:



Obr.L9: Filtrace MAC adresy

Filtrace MAC adresy je velmi jednoduchá. Podobně, jako na mnohých jiných místech tohoto modelu, je použita komponenta Submatrix, která ze sériově běžících dat vyfiltruje požadovanou sekvenci. Ta je poté převedena na desítkové číslo komponentou Bit to Integer Converter. Poté již desítkové číslo vstupuje do subsystému Klasifikace bufferu, který přiřadí jednotlivým MAC adresám hodnoty čísel v rozsahu 1 až 4, abychom mohli jimi ovládat hlavní spínač.

Vlastní klasifikace je implementována takto:

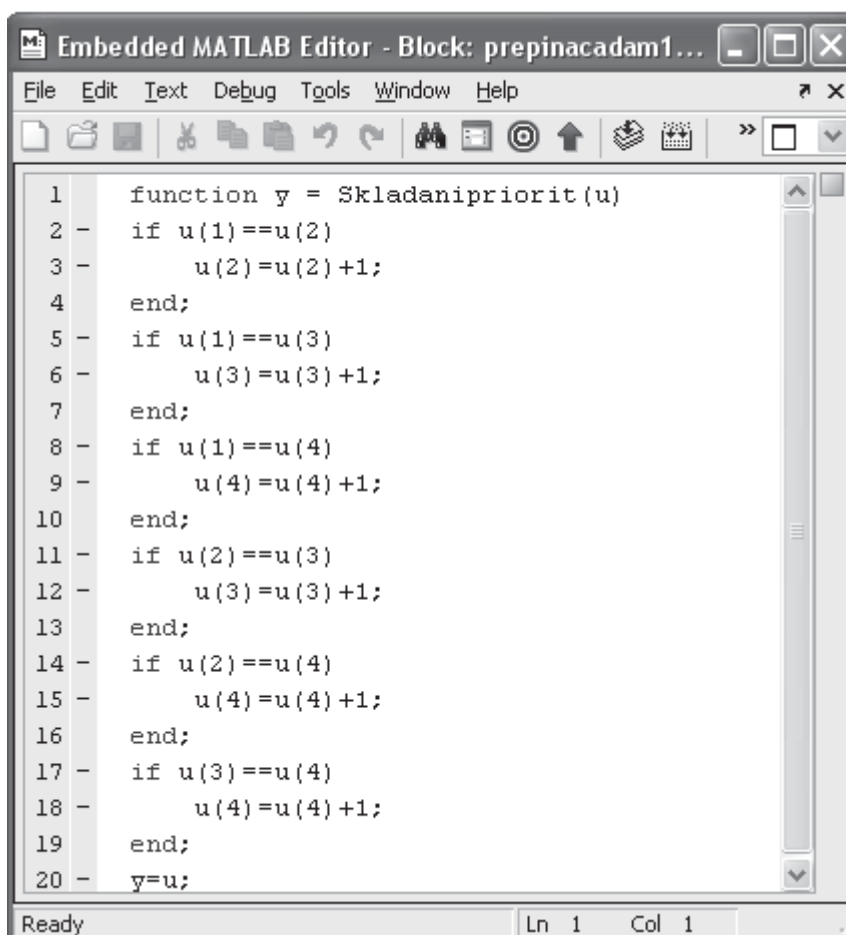


Obr.L10: Jeden element subsystému Klasifikace bufferu. V subsystému jsou celkem 4.

Nejprve se zjistí dekadická hodnota MAC. Poté se jí přiřadí čísla z intervalu 1 až 4. To lze v podstatě libovolně, protože k přetřídění dojde až při bufferování. Aby však nevznikal chaos, zvolili jsme vzestupné hodnocení. Komponenta Constant, nazvaná Flag (známka, praporek...) nám tuto hodnotu pro každý vstup a výstup přiřadí a akční blok přepínače typu *Case* ji propojí do výstupu.

Komponenta Priorita a v ní obsažený subsystém Přeskládání FIFO řeší situaci, kdy na jeden výstup přijde *více dat*. Například dva servery poskytují službu jedné stanici. Tato situace v praxi nastává poměrně často, proto ji model musí ošetřit. Popišme si jeden vstup (jsou celkem 4) tohoto subsystému. Při detekci určitého stupně kvality dat (parametru Q se seřadí data do pomocné fronty. Poté se v případě, že přišla data stejného typu, *překlasifikuje* Q tak, že jej zvýšíme o 1 – tím zajistíme *sekvenčnost* dat. Data tedy půjdou na výstup subsystému v takovém pořadí, jak byla generována. Protože pokud jsou stejného typu, pak doručujeme podle pravidla FIFO, tedy první dovnitř, první ven.

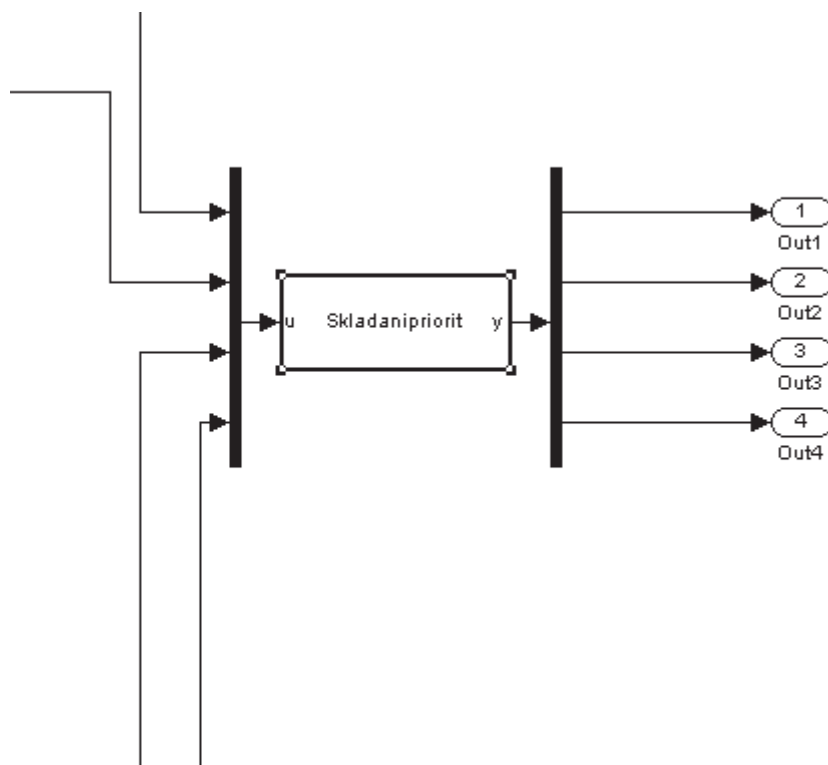
Algoritmus Skládání priorit je následující a zajistí ho komponenta Embedded MATLAB Function:



```
1 function y = Skladanipriorit(u)
2 - if u(1)==u(2)
3 -     u(2)=u(2)+1;
4 - end;
5 - if u(1)==u(3)
6 -     u(3)=u(3)+1;
7 - end;
8 - if u(1)==u(4)
9 -     u(4)=u(4)+1;
10 - end;
11 - if u(2)==u(3)
12 -     u(3)=u(3)+1;
13 - end;
14 - if u(2)==u(4)
15 -     u(4)=u(4)+1;
16 - end;
17 - if u(3)==u(4)
18 -     u(4)=u(4)+1;
19 - end;
20 - y=u;
```

Obr.L11: Skládání kvalitativně stejných rámců

Do tohoto bloku vedeme data přes komponentu Multiplexer a poté ji seřazenou demultiplexerem opět rozdělíme do 4 výstupů:



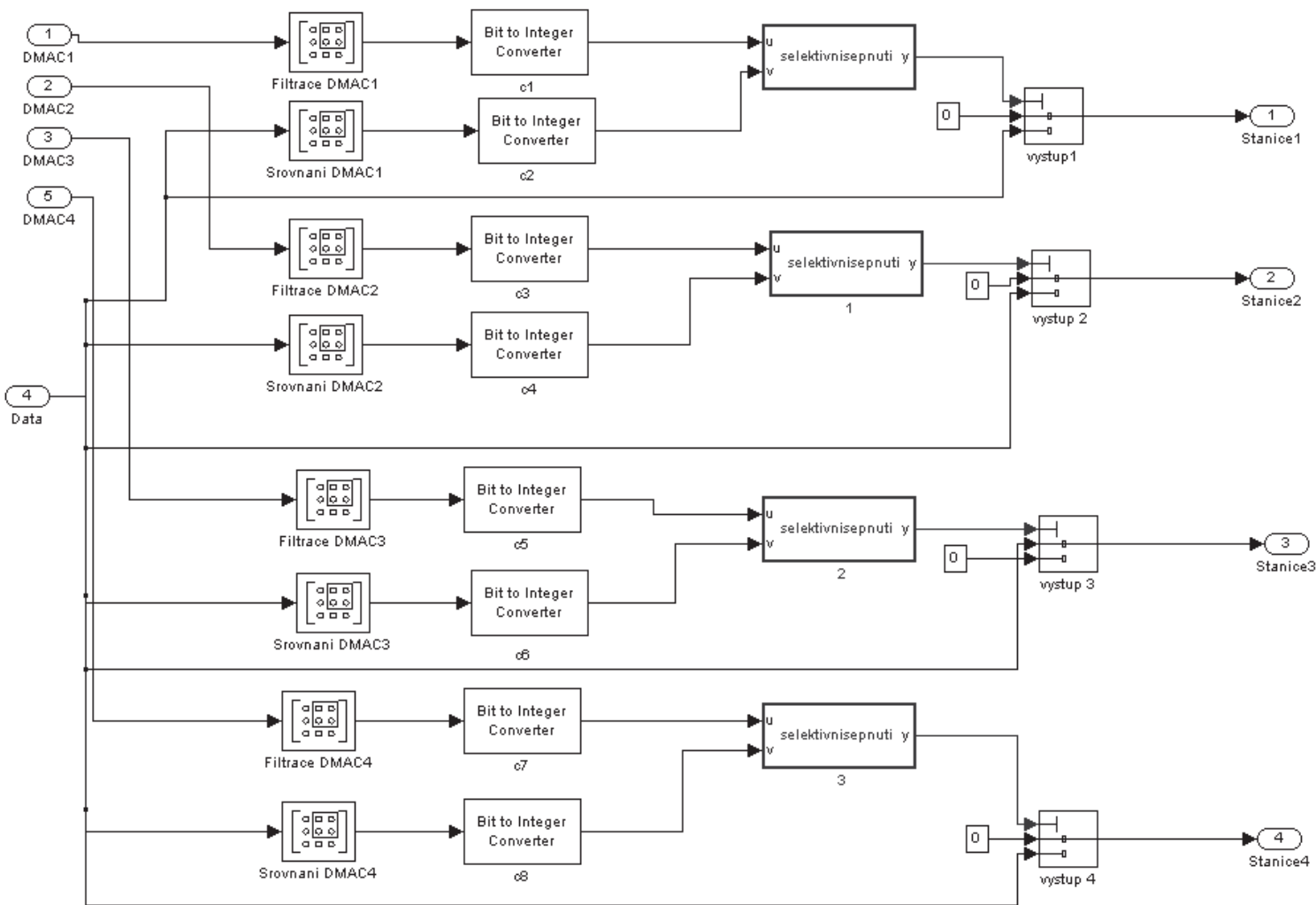
Obr.L12: Implementace výše uvedené překlasifikace Q

Výstupy subsystémů Priorita a Klasifikace bufferu vedou do bloku Buffering, kde probíhá jednak finální setřídění dat bublinovým tříděním (možno použít i jiné metody), jednak vygenerování spouštěcích povelů pro hlavní spínač v intervalu 1 až 4 v příslušném pořadí. Tím *hlavní spínač sepne v určitém pořadí hlavní datový tok* do subsystému Řízení provozu, který již pracuje srovnávací metodou s MAC adresami – emuluje tedy přepínací tabulku.

Řízení provozu

Tento subsystém zajistí doručení dat příslušné stanici. Funguje na principu srovnání skutečné MAC adresy stanice (její signifikantní části) s MAC adresou označenou jako cílová, jež nese rámeček. Srovnání se provádí pomocí Embedded MATLAB Function, lze ho však provést i komponentou If.

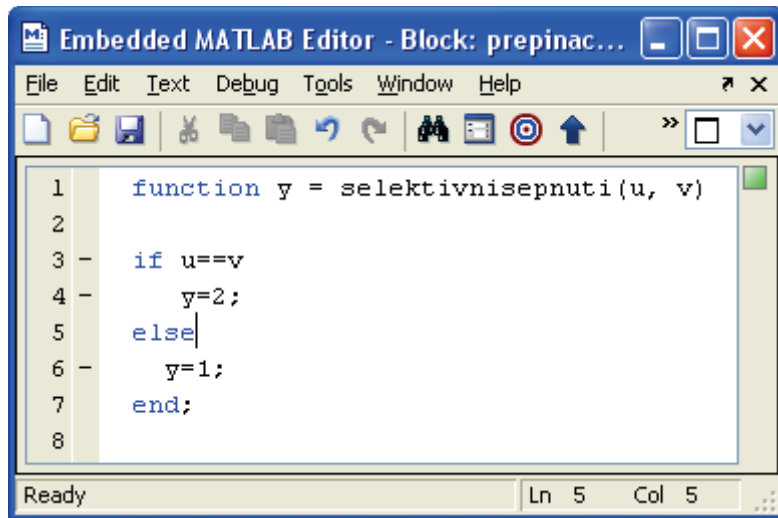
Jeho vnitřní strukturu označuje následující obrázek:



Obr.L13: Řízení provozu přepínací části

Činnost tohoto subsystému je zřejmá ze zapojení: Jakmile dojde ve frontě seřazený rámec, v bloku Matlabu nazvaném *selektivnisepnuti* dojde ke srovnání MAC adres. Protože ty v modelu mohou být pouze 4, propojí se odpovídající adresa bod proti bodu tím způsobem, že portový spínač dostane povel k přepnutí na datovou část. Jinak propojí nulu.

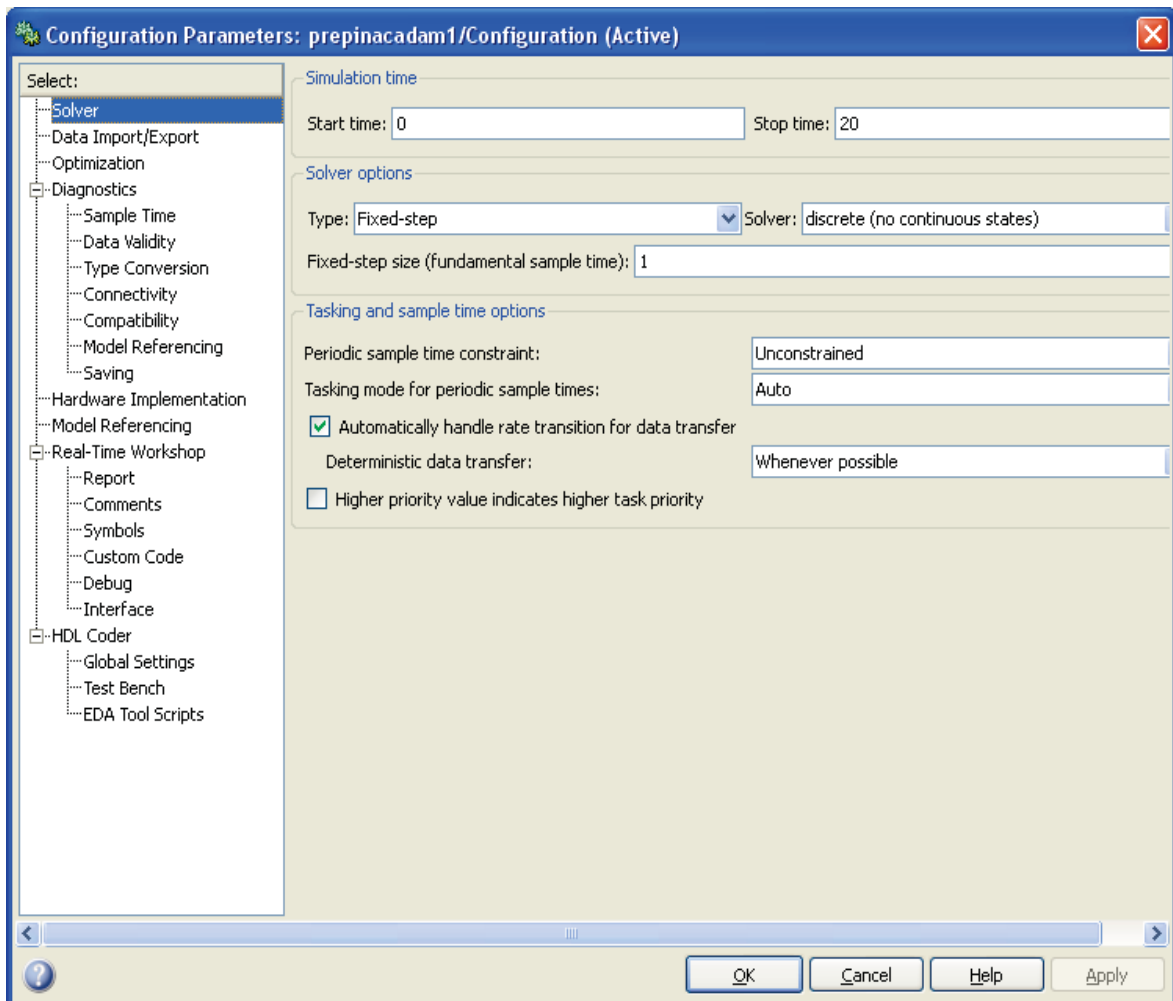
Pokud bychom chtěli simulovat přepnutí na neznámou MAC adresu, doplnili bychom tento subsystém tak, aby propojil tyto adresně neznámé rámce do všech výstupů. Tím bychom softwarově suspendovali přepínač do role rozbočovače.



Obr.L14: Řízení sepnutí portového spínače

Nastavení simulace

Simulink dává několik možností, jak ovlivňovat běžící simulaci. Blíže to upřesní následující okno:



Obr.L15: Nastavení parametrů simulace

Jako nejdůležitější parametr je řešitel (solver), který umožňuje nastavit přesnost

numerických řešení, použitých v Simulinku pro jednotlivé komponenty systému. Implicitně je nastaven řešitel ode45 (Dormand-Prince) což je vhodné pro řešení diferenciálních rovnic, například modelování přechodných jevů v elektrotechnice, odezvy systémů na spojitě či kvazispojitě signály, apod. Pro naši potřebu se hodí lépe simulace s *diskrétním časem*, která je vhodná pro simulaci číslicových systémů, kam přepínač rovněž patří. Nastavení se jmenuje *discrete (no continuous states)*. Uvedeme proto doporučené (nikoliv však striktní) parametry simulace, které použijeme i v laboratorní úloze. Řešíme pouze oddíl Solver:

Nejsou-li hodnoty takto nastaveny, nastavíme je podle obrázku. Ostatní nastavení ponecháme implicitní.

Úloha 1 – simulace s optimalizací priority

Nyní již můžeme spustit rovnou simulaci přenosu dat. Nejprve však nastavíme požadovaná data podle scénářů, které si dopředu připravíme. Zde uvedeme pouze některé z nich, protože vzájemné kombinace doručování dat ze serverů na stanice si můžeme stanovit v podstatě libovolně.

Scénář 1 – bod proti bodu, 4 druhy dat, 4 příjemci

Pořadí serveru	Cílová MAC (dekadicky)	Kvalita (dekadicky)	Stav doručení rámce
1.			
2.			
3.			
4.			

Doručená data po simulaci:

(sem dejte výstupy z osciloskopů)

Zhodnocení výsledků:

Scénář 2 – bod proti bodu, 3 druhy dat, 4 příjemci

Pořadí serveru	Cílová MAC (dekadicky)	Kvalita (dekadicky)	Stav doručení rámce
1.			
2.			
3.			
4.			

Doručená data po simulaci:

(sem dejte výstupy z osciloskopů)

Zhodnocení výsledků:

Scénář 3 – Dva servery doručují jedné stanici, ostatní bod proti bodu, 4 druhy dat, 3 příjemci

Pořadí serveru	Cílová MAC (dekadicky)	Kvalita (dekadicky)	Stav doručení rámce
1.			
2.			
3.			
4.			

Doručená data po simulaci:

(sem dejte výstupy z osciloskopů)

Zhodnocení výsledků:

Scénář 4 – Všechny servery doručují jedné stanici 4 druhy dat

Pořadí serveru	Cílová MAC (dekadicky)	Kvalita (dekadicky)	Stav doručení rámce
1.			
2.			
3.			
4.			

Doručená data po simulaci:

(sem dejte výstupy z osciloskopů)

Zhodnocení výsledků:

Závěrečné zhodnocení simulace:

Úloha 2 – Přepínač bez optimalizace priority

Pokuste se v subsystému Řízení spojovacího pole vyřadit subsystémy implementující prioritizaci podle parametru Q.

Návod: Vyřadte výstup *spínače pořadí* tak, že jej zakončíte terminátorem a na vstup hlavního spínače zavedete rovnou spouštěcí signál *Trg1*. Výstup spínače pořadí zakončete *terminátorem*. Najdete jej v knihovně Commonly Used Blocks v Simulink Library Browser. Pak poklepejte myší na hlavní spínač a zaškněte volbu *Use zero-based indexing*. Při uvádění modelu do původního stavu nezapomeňte tuto volbu znovu odškrtnout.

Zaznamenejte výsledky osciloskopů opět pro různé scénáře.

Jaké změny pozorujete? Srovnajte s předchozími výsledky. Uložte model jako další variantu a předvedte vyučujícímu.

Úloha 3 – Optimalizace Řízení provozu

Pokuste se nahradit matlabovský skript funkce `selektivnisepnuti` čistě simulinkovými komponentami. Náповěda: Potřebujete komponentu pro podmínky a zavedení konstanty. Vyzkoušejte funkčnost modelu. Uložte model jako další variantu a předved'te vyučujícímu.

Úloha 4 – Didaktický test na téma přepínače a Ethernet

Zaškrtněte správnou odpověď. Správná je jen jedna:

1. Přepínač bez podpory kvality služeb pracuje na vrstvě ISO/OSI:

- a) fyzické
- b) linkové
- c) síťové
- d) transportní

2. Přepínač s podporou kvality služeb pracuje mezi vrstvami ISO/OSI:

- a) fyzickou a linkovou
- b) prezentační a aplikační
- c) linkovou a síťovou
- d) transportní a relační

3. Funkce AutoCross zajišťuje:

- a) propojení dvou přepínačů přímým ethernetovým kabelem namísto kříženého
- b) automatické přizpůsobení vodorovné nebo svislé pracovní poloze
- c) automatické chlazení přepínače
- d) nepřerušitelné napájení při výpadku dodávky elektřiny

4. Funkce Auto Negotiation přizpůsobí rozdílné:

- a) zpoždění
- b) jitter
- c) protokoly
- d) rychlosti

5. Mechanické datové zásuvky ethernetového přepínače jsou typu

- a) RJ-11
- b) RJ-45
- c) RJ-40
- d) RJ-12

6. V síti Ethernet je protikolizní opatření zajištěno algoritmem:

- a) CSMA/CD
- b) CSMA/CA
- c) Aloha
- d) Slotted Aloha

7. Kroucená dvojlinka UTP Cat.6 pro Ethernet má:

- a) Jeden pár měděných zkroucených vodičů
- b) Dva páry měděných zkroucených vodičů
- c) Tři páry měděných zkroucených vodičů
- d) Čtyři páry měděných zkroucených vodičů

8. V síti Fast Ethernet (100Mbps) je zpravidla tok dat

- a) simplexní
- b) poloduplexní
- c) duplexní
- d) žádný

9. Prostý rozbočovač (hub)

- a) je kvalitnější než přepínač
- b) zmenšuje kolizní doménu
- c) zvětšuje kolizní doménu
- d) je nezbytně nutné ho zapojit před server

10. Metalická kabeláž v síti Fast Ethernet (100Mbps) je:

- a) souosá
- b) souměrná
- c) silová
- d) neizolovaná

Správné odpovědi: 1b, 2c, 3a, 4d, 5b, 6a, 7d,8c, 9c, 10b

Závěr: (sem napište podrobný závěr a zhodnocení Vašich výsledků v této úloze)

8 Závěr

Kvalita datového toku v konvergovaných sítích se stává aktuální otázkou pro návrháře moderních technologických řešení, kde je nutno rychle, spolehlivě a kvalitně sdílet data a komunikovat. Proto musíme kromě vlastní metalické či optoelektronické kabeláže věnovat pozornost aktivním prvkům v sítích. Správci jednotlivých lokálních sítí stojí před náročnými úkoly, které musí v průběhu každodenní praxe řešit.

Nejobávanějšími závadami, které se běžně v sítích vyskytují, jsou pseudonáhodné výpadky či zpomalení přenosové rychlosti, která se projeví zpravidla v tu nejméně vhodnou dobu. Tyto závady jsou navíc zcela nahodilé, a proto se velice obtížně lokalizují i odstraňují. Je to dáno vlastnostmi Ethernetu, který – v současnosti navíc kombinovaný s WiFi – je nejčastější implementací konvergovaných sítí v domácím i firemním prostředí. Mohou být způsobeny nekvalitní administrací či nevhodným uživatelským chováním, avšak rovněž různými rušivými vlivy, se kterými se v návrhu sítě nepočítalo. Ne vždy lze síť optimalizovat hardwarově – tedy vytvořit zvláštní cesty pro hlasové a multimediální služby a zvláštní cesty pro „běžná“ data. Právě na multimediálních datových proudech je vidět (i slyšet) každá nedokonalost, která se v síti vyskytne.

Existují samozřejmě postupy, jak tyto problémy minimalizovat – i když žádné řešení nemusí být plně univerzální. Jednou z metod je však *kvalifikovaný výběr a nastavení reálných aktivních prvků, respektování zásad kvality služeb a vhodného rozčlenění sítě tak, aby byly kolizní domény co nejmenší*. I když úzkým hrdlem soudobých sítí jsou přípojné body do sítě transportní, vhodnou hardwarovou optimalizací aktivních prvků na místní úrovni lze mnohým problémům předejít.

Přepínač (switch) převzal nejenom funkci rozbočování digitálního datového signálu, ale rovněž sejmul z koncových zařízení (jejich síťových rozhraní) povinnost filtrovat data podle cíle, stal se jakýmsi dispečerem v síti. Vyčerpávající popis tohoto hardware můžeme najít mimo jiné i v [9].

Cílem práce bylo v základech popsat rozdíly mezi klasickým rozbočovačem a přepínačem, jeho umístěním v modelu ISO/OSI a navrhnout postup, jak simulovat jeho činnost v systému Matlab Simulink. Model, který byl vytvořen, je východiskem pro další studium vnitřní architektury přepínačů, včetně přepínače L3, který optimalizuje provoz podle kvalitativního obsahu dat.

Podarilo se osvětlit pojem *kvalita služby* a nastínit, jak by měli návrháři sítí LAN postupovat, aby jejich síť byla nejenom racionálně a přiměřeně navržená, ale aby za provozu dosahovala těch nejlepších výsledků, které lze z dané technologie vytěžit. Byly probrány výhody i jistá rizika nasazení L3 přepínačů do provozu. Tento problém se dá částečně vyřešit vhodnou

teoretickou přípravou návrhu sítě. Různý návrhový software pro síťové architektury v tom může významně pomoci. Namátkou lze jmenovat například Opnet Modeler či Opnet IT Guru Academic Edition firmy Opnet.

Avšak kvalitně navrženou síť prověří důkladně až běžný provoz. Proto by měl každý síťový architekt koncipovat své dílo jako stavebnici, jejíž prvky lze dodatečně modifikovat a rozšiřovat. Standardy, které svět sítí a síťových technologií vytvořil, mu v tom mohou významně pomoci.

Úplně nakonec je třeba se zmínit, že v souvislosti s nekompatibilitou IPv4 a IPv6 je nutné problematiku QoS řešit v podstatě pro obě verze protokolů. Z tohoto důvodu je třeba zvážit, zda navržené reálné komponenty podporují obě verze. Není-li tomu tak, nebo nelze-li tuto kompatibilitu z jakéhokoli důvodu zajistit, je třeba se i přes výhody, které přepínače L3 s podporou kvality služeb, poskytují, jejich použití raději vyhnout. Pokud totiž rychlý, kvalitní a nechybující klasický přepínač odbaví jakoukoli verzi protokolů, pro niž je transparentní, pomůže to ke snižování latence i jitteru sítě, což jsou nejhorší nepřátelé multimediálního a hlasového provozu. Se zvyšujícím se stupněm konvergence telekomunikačních a informačních technologií je to hlavní úkol, který budou příští generace telekomunikačních expertů, elektroinženýrů, síťových architektů i softwarových tvůrců řešit.

9 Seznam obrázků a tabulek

Seznam obrázků:

1. Obr.1: Srovnání modelů ISO/OSI a TCP/IP – odpovídající vrstvy
2. Obr.2: Postavení aktivního prvku v segmentu sítě. Aktivním prvkem je v současnosti nejčastěji přepínač – switch. Buď transparentní nebo s podporou kvality služeb (L3 switch)
3. Obr.3: Blokové schéma rozbočovače – hubu
4. Obr.4: Zjednodušené blokové schéma přepínače – switche
5. Obr.5: Princip algoritmu CSMA/CD
6. Obr.6: Algoritmus zjednodušeného adaptivního přepínání
7. Obr.7: Prostředí Matlab se spuštěným Simulinkem
8. Obr.8: Hlavní okno Simulinku s výběrem simulovaných komponent
9. Obr.9: Okno nového modelu, ve kterém bude simulace prováděna
10. Obr.10: Ze serveru odchází data a podle filtrace parametru Q ihned poznáváme, že jde o data digitální televize – binární hodnota Q je 1101
11. Obr.11: Subsystem Serverový výstup 1 (zdroj dat). Z polohy přepínačů je vidět, že odtud vysíláme televizní data a parametr Q činí 1101. Tuto hodnotu jsme odchytili za pomoci filtru – viz předchozí obrázek
12. Obr.12: Základní logické zapojení modelu
13. Obr.13: Základní pohled na model. Sestává z přepínače, 4 serverů jako zdrojů dat a 4 stanic jako cílových uzlů
14. Obr.14: Vnitřní struktura serveru. Generátor dat spojí jednotlivé dílčí konstanty a proměnné do celistvého rámce, dlouhého 112 bitů
15. Obr.15: Vnitřní zapojení subsystému Stanice.
16. Obr.16: Komponenty Řízení spojovacího pole a Řízení provozu
17. Obr.17: Řízení spojovacího pole sestává ze 4 subsystémů. Řeší se zde priorita a bufferování přepínače, což je v podstatě algoritmus třídění
18. Obr.18: Filtrace MAC adresy
19. Obr.19: Jeden element subsystému Klasifikace bufferu. V subsystému jsou celkem 4.
20. Obr.20: Skládání kvalitativně stejných rámců
21. Obr.21: Implementace výše uvedené překlasifikace Q
22. Obr.22: Řízení provozu přepínací části
23. Obr.23: Řízení sepnutí portového spínače
24. Obr.24: Nastavení parametrů simulace
25. Obr.25: Model po simulaci prvního scénáře
26. Obr.26: Parametr Q doručených dat na stanici 1 v závislosti na čase. Q=0 (Data)
27. Obr.27: Parametr Q doručených dat na stanici 2 v závislosti na čase. Q=5 (Databáze)
28. Obr.28: Parametr Q doručených dat na stanici 3 v závislosti na čase. Q=9 (VoIP)
29. Obr.29: Parametr Q doručených dat na stanici 4 v závislosti na čase. Q=13 (IPTV)
30. Obr.30: Model po simulaci druhého scénáře
31. Obr.31: Parametr Q doručených dat na stanici 1 v závislosti na čase. Q=13 (IPTV)
32. Obr.32: Parametr Q doručených dat na stanici 2 v závislosti na čase. Q=5 (Databáze)
33. Obr.33: Parametr Q doručených dat na stanici 3 v závislosti na čase. Q=9 (VoIP)
34. Obr.34: Parametr Q doručených dat na stanici 4 v závislosti na čase. Q=13 (IPTV)
35. Obr.35: Model po simulaci třetího scénáře
36. Obr.36: Parametr Q doručených dat na stanici 1 v závislosti na čase. Q=13 (IPTV) a Q=5 (Databáze)

37. Obr.37: Stanici 2 nebyla doručena žádná data – náležitě
38. Obr.38: Parametr Q doručených dat na stanici 3 v závislosti na čase. Q=9 (VoIP)
39. Obr.39: Parametr Q doručených dat na stanici 4 v závislosti na čase. Q=13 (IPTV)
40. Obr.40: Model po simulaci čtvrtého scénáře
41. Obr.41: Parametr Q doručených dat na stanici 1 v závislosti na čase. Q=0, 5, 9, 13, (Všechny typy dat)
42. Obr.42: Stanici 2 nebyla doručena žádná data – náležitě
43. Obr.43: Stanici 3 nebyla doručena žádná data – náležitě
44. Obr.44: Stanici 4 nebyla doručena žádná data – náležitě

Seznam tabulek:

1. Tab.1: Hodnoty parametru Q podle typu dat
2. Tab.2: MAC adresy stanic. S těmi model bude pracovat
3. Tab.3: Scénář 1
4. Tab.4: Scénář 2
5. Tab.5: Scénář 3
6. Tab.6: Scénář 4

Laboratorní úloha má vlastní číslování obrázků.

10 Seznam literatury

- [1] PUŽMANOVÁ, R. *Moderní komunikační sítě od A do Z*. Computer Press, Brno 2006. 430 s.
- [2] WENDELL, O, RUS, H., MEHTA, N. *Směrování a přepínání sítí*. Computer Press, Brno 2009
- [3] HORÁK, J. KERŠLÁGER, M. *Počítačové sítě pro začínající správce*. Computer Press Brno 2006
- [4] SPURNÁ, I. *Počítačové sítě*. Brno, Computer Press 2010. 180s.
- [5] ŠVEC, A. *Optimalizace nastavení transportní sítě*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. Vedoucí bakalářské práce doc. Ing. Vladislav Škorpil, CSc.
- [6] ZEDNÍČEK, P. *Síťový prvek s pokročilým řízením*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. Vedoucí diplomové práce doc. Ing. Vladislav Škorpil, CSc.
- [7] PŘECECHTĚL, R. *Optimalizace řízení aktivního síťového prvku*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. Vedoucí diplomové práce doc. Ing. Vladislav Škorpil, CSc.
- [8] BOHÁČ, M. *Implementace kvality služby do řízení síťového prvku*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. Vedoucí diplomové práce doc. Ing. Vladislav Škorpil, Csc.
- [9] MOLNÁR, K., ZEMAN, O., SKOŘEPA, M. *Moderní síťové technologie*. Brno, VUT v Brně, Fakulta elektrotechniky a komunikačních technologií 2008.101 s. Dostupné online: <http://www.zotto.cz/>
- [10] Výukové materiály pro Simulink na <http://www.mathworks.com/>

Na přípravě této diplomové práce byl použit následující aplikační software:



a Matlab Simulink R2008a (ver.7.6.0) firmy MathWorks, Inc.

