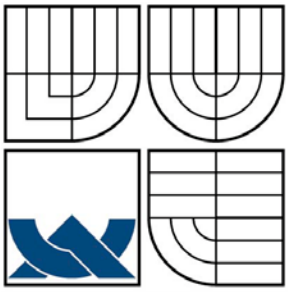
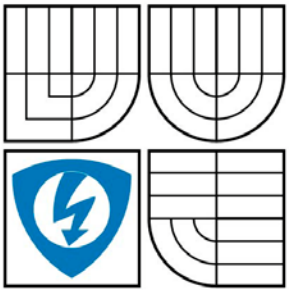


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

TVORBA POKROČILÉHO SIMULAČNÍHO MODELU TECHNOLOGIE DIFFSERV V PROSTŘEDÍ OPNET MODELER

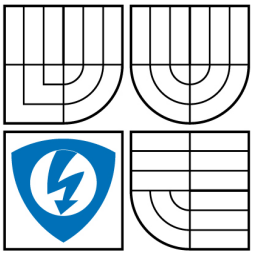
DESIGN OF ADVANCED SIMULATION MODEL OF DIFFSERV TECHNOLOGY IN OPNET MODELER ENVIRONMENT

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE — TOMÁŠ MACHATA
AUTHOR

VEDOUCÍ PRÁCE — ING. JIŘÍ HOŠEK
SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Tomáš Machata	***ID:***	98260
Ročník:	3	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

Tvorba pokročilého simulačního modelu technologie DiffServ v prostředí Opnet Modeler

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte studentské práce zaměřené na problematiku zajištění kvality služeb (QoS) v IP sítích. Zaměřte se zejména na technologii DiffServ a způsob klasifikace síťového provozu na hraničních směrovačích této technologie. Seznamte se simulačním prostředím Opnet Modeler a vytvořte v něm pokročilý simulační model DiffServ domény, který se bude skládat z několika hraničních a několika páteřních směrovačů. Uvnitř DiffServ domény nakonfigurujte směrovací protokol umožňující využívání alternativních linek. Pomocí simulací ověřte vliv výpadku hlavní linky a použití náhradní trasy na zajištění kvality služeb.

DOPORUČENÁ LITERATURA:

[1] PARK, K.: QoS in Packet Networks. Boston: Springer, 2004, ISBN: 0-387-23390-3.
[2] SZIGETI, T., HATTINGH, C.: End-to-end QoS network design. Indianapolis: Cisco Press, 2005, ISBN: 1-58705-176-1.

Termín zadání:	9.2.2009	***Termín odevzdání:***	2.6.2009

Vedoucí práce: Ing. Jiří Hošek

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ANOTACE

Cílem této bakalářské práce bylo seznámit se s problematikou zajištění kvality služeb v IP sítích. Největší pozornost byla věnována technologii DiffServ a způsobu identifikace a klasifikace síťového provozu na hraničních směrovačích.

Byl vytvořen simulační model technologie DiffServ v prostředí OPNET Modeler. Tento model obsahuje několik okrajových a vnitřních směrovačů, přepínačů, serverů a klientských stanic, na kterých je vytvářen síťový provoz (VoIP, HTTP, databáze a FTP). Jednotlivé aplikace byly zařazeny do různých tříd na okrajových směrovačích DiffServ domény a bylo zajištěno různé chování s jednotlivými aplikacemi pomocí Assured Forwarding PHB. Pomocí simulací bylo zjištěno, jaká metoda třídění paketů je pro výslednou kvalitu služeb nejlepší.

Dále byl model DiffServ domény rozšířen o alternativní linku. Zde byl nakonfigurován směrovací protokol OSPF a byl zkoumán vliv výpadku hlavní linky a použití alternativní trasy na kvalitu služeb.

Klíčová slova:

DiffServ, kvalita služeb, OPNET Modeler, OSPF, QoS, simulace, výpadek/nahození linky

ANNOTATION

The aim of this bachelor´s thesis was to get acquainted with the problems of ensuring quality of service in IP networks. The greatest attention was devoted to DiffServ technology and methods for the identification and classification of network traffic at edge routers.

Simulation model of DiffServ technology was created in OPNET Modeler environment. This model contains a number of edge and core routers, switches, servers and client stations on which it is generating a network traffic (VoIP, HTTP, FTP and databases). Individual applications were included in different classes at the edge routers of DiffServ domain and the different behavior of the applications was ensured with usage of an Assured Forwarding PHB. It was found by using simulations which method of classification packets is the best for the quality of service.

Furthermore the model of DiffServ technology was extended about alternative route. OSPF routing protocol was configured here and has been studied the impact of failure of the main line and using of alternative routes to the quality of service.

Keywords:

DiffServ, failure/recovery route, OPNET Modeler, OSPF, QoS, Quality of Service, simulation

Bibliografická citace práce:

MACHATA, T.: Tvorba pokročilého simulačního modelu technologie DiffServ v prostředí OPNET Modeler. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 50 s. Vedoucí bakalářské práce Ing. Jiří Hošek.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma "Tvorba pokročilého simulačního modelu technologie DiffServ v prostředí OPNET Modeler" jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne …….. ……...

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Jiřímu Hoškovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování bakalářské práce.

V Brně dne …….. ……...

(podpis autora)

Obsah

Obsah 1
Seznam obrázků 3
Seznam tabulek 3
1 Kvalita služeb - QoS 4
1.1 Parametry kvality služeb - QoS 4
1.1.1 Zpoždění 4
1.1.2 Kolísání zpoždění 4
1.1.3 Ztrátovost paketů 5
1.1.4 Propustnost 5
1.2 Implementace kvality služeb - QoS 5
1.2.1 Implementace kvality služeb v IP sítích 5
1.3 Technologie kvality služeb – QoS 6
2 Integrované služby – IntServ 7
3 Diferencované služby – DiffSserv 9
3.1 ToS pole 9
3.2 DS pole 9
3.2.1 DSCP 10
3.3 PHB 10
3.4 DiffServ doména 11
3.4.1 Směrovače v DiffServ doméně 12
3.5 Model Diffserv 12
3.5.1 Klasifikátor 13
3.5.2 Měřič 13
3.5.3 Značkovač 13
3.5.4 Tvarovač 14
Token bucket 14
3.5.5 Zahazovač 14
4 Access Control List 15
5 Fronta typu WFQ (Weighted Fair Queuing) 16
6 Směrovací protokol OSPF (Open Shortest Path First) 17
7 OPNET Modeler 19
7.1 Základní prvky prostředí OM 19
7.2 Základní editory prostředí OM 19
8 Tvorba modelu DiffServ v prostředí OM 21
8.1 Vložení a nastavení jednotlivých prvků modelu 21
8.1.1 Objekt Application Config 22
8.1.2 Objekt Profile Config 22
8.1.3 Nastavení serveru 24
8.1.4 Nastavení klienta 24
Nastavení VoIP klienta 25
8.2 Vytvoření dalšího scénáře 26
8.3 Nastavení kvality služeb 26
8.3.1 Scénář QoS s tříděním paketů podle protokolu a portu 27
Nastavení ACL na okrajových směrovačích 27
Nastavení tříd provozu na okrajových směrovačích 27

Nastavení politiky provozu na okrajových směrovačích 28
Nastavení rozhraní na směrovačích....29
Nastavení vnitřních směrovačů....30
Objekt QoS Atribute Config....31
8.3.2 Scénář QoS s tříděním paketů podle IP adresy 31
8.3.3 Scénář QoS s tříděním paketů podle DSCP 32
8.4 Nastavení alternativní trasy a protokolu OSPF 33
8.4.1 Rozšíření modelu – dvě linky....34
8.4.2 Nastavení protokolu OSPF 34
Nastavení metriky....34
Nastavení časovačů v protokolu OSPF....35
Směrovací tabulky 35
8.4.3 Nastavení výpadku linky 36
8.5 Simulace v prostředí OM 37
8.5.1 Výsledky simulace v prostředí OM....38
Výsledky simulace – metody třídění paketů 38
Výsledky simulace – vliv výpadku hlavní linky....39
9 Závěr....42
Seznam zkratek 43
Použitá literatura 44
Přílohy....46
Seznam příloh 46

Seznam obrázků

Obr. 2.1:Blokové schéma Integrované služby....8
Obr. 3.1: Struktura ToS – Type of Service v IPv4....9
Obr. 3.2: DS pole....10
Obr. 3.3: DiffServ doména....12
Obr. 3.4: Model DiffServ - blokové schéma úpravy provozu....13
Obr. 3.5: Token bucket - princip....14
Obr. 5.1: Metoda WFQ....16
Obr. 7.1: Hlavní menu a tlačítka pracovní plochy prostředí OM....19
Obr. 8.1: Výsledný model technologie DiffServ....21
Obr. 8.2: Objekt Application Config - nastavení FTP....23
Obr. 8.3: Nastavení objektu Profile Config....23
Obr. 8.4: Nastaveni serveru - databáze....24
Obr. 8.5: Nastavení klienta – databáze a FTP....25
Obr. 8.6: Nastavení VoIP klienta – VoIP a databáze....26
Obr. 8.7: Nastavení ACL pravidel pro HTTP....28
Obr. 8.8: Nastavení tříd provozu....28
Obr. 8.9: Nastavení politiky provozu na okrajovém směrovači....29
Obr. 8.10: Nastavení rozhraní na okrajovém směrovači....30
Obr. 8.11: Nastaven vnitřních směrovačů....30
Obr. 8.12: Nastavení objektu QoS Atribute Config....31
Obr. 8.13: Přiřazení DSCP značky VoIP aplikaci....32
Obr. 8.14: Nastavení ACL podle DSCP značky....33
Obr. 8.15: Model s protokolem OSPF – metriky a IP adresy rozhranní....33
Obr. 8.16: Nastavení směrovacího protokolu....34
Obr. 8.17: Nastavení metriky linky....35
Obr. 8.18: Nastavení OSPF časovačů....35
Obr. 8.19: Směrovací tabulka směrovače Core_Router_1....36
Obr. 8.20: Objekt Failure-Recovery....36
Obr. 8.21: Nastavení objektu Failure-Recovery....36
Obr. 8.22: Nastavení simulace....37
Obr. 8.23: Graf závislosti jitteru na čase u VoIP aplikace....38
Obr. 8.24: Graf závislosti zpoždění paketů mezi koncovými body na čase u VoIP aplikace....39
Obr. 8.25: Graf závislosti propustnosti mezi dvěma vnitřními směrovači na čase....40
Obr. 8.26: Graf závislosti jitteru na čase u scénáře s výpadkem a bez výpadku....41
Obr. 8.27: Graf závislosti zpoždění paketů mezi koncovými body na čase u VoIP aplikace a scénáře s výpadkem a bez výpadku....41

Seznam tabulek

Tab. 1: Osm tříd priorit podle hodnoty DSCP/CSCP....11
Tab. 2: Tlačítka prostředí OM....20
Tab. 3: Přiřazení značky a váhy aplikacím podle protokolu a portu....27
Tab. 4: Přiřazení IP adres a aplikací jednotlivým klientům....32

1 Kvalita služeb - QoS

V ideálním světě by bylo možné kdykoliv a kdekoliv okamžitě přistupovat ke všem požadovaným informacím, stačilo by pouhé kliknutí myši. Nebylo by nutné čekat na žádné doručování souborů přes celý internet. Bohužel toto v našem reálném světě neplatí. Internetové sítě nemají obrovská a široká datová potrubí, po kterých by se dala okamžitě přenášet jakákoliv data. Vzhledem k problémům s dostupnou šířkou pásma, cenou a rychlostí přenosu, se kterými se moderní sítě potýkají, musíme nějakým způsobem umět poslat nejdříve data, které jsou více důležitá. Z tohoto důvodu vzniklo řešení kvality služeb - Quality of Service (dále jen QoS), které umožňuje přiřadit různým typům síťového provozu různou prioritu. A podle této priority posílat data.

Kvalita služeb je technologie, která se zrodila ve sdružení IETF (Internet Engineering Task Force) [3]. Je to soubor procesů, které aktivně vstupují do řízení šířky pásma s cílem poskytovat zaručené úrovně síťových služeb daným aplikacím nebo uživatelům [13]. Tento mechanismus upřednostňuje určitý síťový provoz před jiným. A již z názvu je patrné, že poskytuje uživatelům určité služby s definovanou kvalitou. O kvalitu služeb QoS se neopírají pouze přenosy obrazového záznamu, hlasového přenosu na internetu (VoIP – Voice over IP) a jiné Real-Time aplikace, ale třeba i obyčejné organizace, pro které je důležité, aby jejich veškerý síťový provoz byl odeslán co nejrychleji.

1.1 Parametry kvality služeb - QoS

Pro spokojenost s vyšší kvalitou služeb je dnes stejně jako hrubá propustnost sítě důležité také správné načasování a koordinace přenosu. Většina multimediálních aplikací je samozřejmě hodně náročná na přenosové kapacity, ale mnohé z nich kladou na činnost IP sítě zcela nové požadavky. Například takový hlasový přenos (VoIP), pokud zde dojde ke zpoždění paketů, budou oba účastnící mluvit jeden přes druhého a plynulost spojení se vytratí. Mezi tyto parametry kvality služeb patří: zpoždění, kolísání, ztrátovost, propustnost [12].

1.1.1 Zpoždění

Zpoždění (delay) je doba potřebná k přenosu dat od odesílatele k adresátovi. Zpoždění dále můžeme dělit na lokální a globální. Lokální zpoždění je čas potřebný na zpracování dat (např. převod z analogového signálu do digitálního a naopak nebo čekání ve frontě na odbavení). Globální zpoždění je celkový čas potřebný k přenosu jednoho paketu. Zpožděním nebo prodlevou rozumíme pomalejší doručení jinak neporušené zprávy [13].

1.1.2 Kolísání zpoždění

Kolísání zpoždění (jitter) zachycuje změnu zpoždění během přenosu jednotlivých paketů od odesílatele k adresátovi. Odesílatel odesílá pakety v určitém časovém rozestupu. V ideálním případě by i v tomto časovém rozestupu byly přijaty

a hodnota jitteru by byla nulová. Ale v mnoha případech dojde ke změně tohoto intervalu a tím některé pakety dorazí dříve nebo mnohem později. Tyto pakety jsou pak vyřazeny, aby uvolnili místo paketům, které následují za nimi. Kolísání se liší od zpoždění tím, že u kolísání dochází k poškození celistvosti zprávy [13].

U hlasového přenosu (technologie VoIP) jsou důsledky kolísání velmi znatelné, z tohoto důvodu může být výsledná konverzace v podstatě nepoužitelná. Tradiční aplikace (např. elektronická pošta, HTTP, přenos souboru FTP a další) nejsou tak citlivé vůči problémům s časováním a kolísání jim nevadí.

1.1.3 Ztrátovost paketů

Ztrátovost (loss) paketů udává kolik procent paketů se během přenosu ztratí. Ke ztrátě paketů může dojít hned z několika důvodů. Nejčastěji je to z důvodu zahlcení či přetížení síťového uzlu, kdy dojde k zahození paketů při přeplnění bufferu [13]. Pro aplikace neprobíhající v reálném čase tato ztráta není kritická. Protože protokol TCP zajistí její opětovné zaslání. Kdežto aplikace pracující pod protokolem UDP a v reálném čase, nemají možnost opětovného zaslání ztraceného paketu.

1.1.4 Propustnost

Propustnost (throughput) je maximální počet datových jednotek přenesených za jeden časový interval (paket/s) [12].

1.2 Implementace kvality služeb - QoS

V linkách rozlehlých sítí se kvalita služeb používá již několik let. Mechanismy kvality služeb závisí na jejich implementaci do různých vrstev referenčního modelu ISO/OSI [13]. Dnes jsou tyto mechanismy nejčastěji zaváděny do asynchronních sítí ATM (Asynchronus Transfer Mode), které pracují na spojové vrstvě, sítích frame relay pracujících na vyšších vrstvách modelu a sítích IP (Internet Protocol), které pracují na síťové vrstvě.

1.2.1 Implementace kvality služeb v IP sítích

Dříve byl IP protokol založen pouze na službě Best-Effort (BE). Tato služba zajišťuje základní konektivitu bez garancí. To znamená, že přenos bude vykonán nejlepším možným způsobem v závislosti na daných podmínkách. Ale s postupným ústupem technologie ATM a slučováním všech typů komunikačních médií (televize, telefon, …) do protokolu IP se pozornost kvality služeb přesunula na Internet Protocol. S tímto přechodem vzniká několik problémů a implementace kvality služeb v sítích IP je obtížná hned z několika důvodů: nespojovaná architektura postavená na nejlepším možném postupu paketů sítí, kdy je rozhodnutí o jeho cestě při doručování do cíle ponecháno pouze protokolu IP. Šířka pásma je konečným prostředkem, a proto každý

bit za sekundu z celkové kapacity, vyhrazený jednomu konkrétnímu spojení, jde na úkor ostatním současně aktivním spojením [13]. Při zajištění kvality služeb je nutné procházet prostředím s mnoha komunikačními protokoly, čímž se zvyšuje složitost celého problému. A aby byl internet i nadále univerzální síť, nesmí být různým, náročným a nepružným aplikacím přiděleno tolik šířky pásma, aby strádaly jiné, časově nezávislé aplikace (HTTP, FTP, …). Z těchto důvodů se kvalita služeb na úrovni IP protokolu stále vyvíjí a zdokonaluje.

Pro správnou činnost musí mít mechanismy kvality služeb v sítích IP tzv. dohodu o úrovni služeb SLA (Service Level Agreement) [3]. Tato dohoda definuje zásady služeb a je nutné ji rozšířit mezi všechna zařízení, která ji mají prosazovat.

1.3 Technologie kvality služeb – QoS

V současnosti existují dvě hlavní technologie implementace kvality služeb - QoS na úrovni IP protokolu [12]:

- Integrované služby (Integrated Services) - IntServ
- Diferencované služby (Differentiated Services) - DiffServ

2 Integrované služby – IntServ

Technologie integrovaných služeb (Integrated Services), ve zkratce IntServ, je první architektura kvality služeb v IP sítí. Pojem integrovaný zde znamená, že jsou všechna zařízení (koncoví hostitelé i mezilehlá zařízení) sjednocena do jednoho útvaru kvality služeb, který je udržován v obou směrech po celou dobu trvání datového toku. U integrovaných služeb aplikace oznámí své požadavky na přenos dat, tedy požaduje určitou kvalitu služeb. Na směrovačích se ověří, zda je k dispozici dostatek prostředků a podle toho se rozhodne. Je-li k dispozici dostatek prostředků, počítačová síť požadavku vyhoví a informuje všechny síťové komponenty (směrovače), aby rezervovaly odpovídající parametry jako je minimální šířka pásma, maximální zpoždění a podobně. Jestliže dostatek prostředků počítačová síť nemá, spojení je zamítnuto a aplikace se může rozhodnout, jestli sníží svoje požadavky na kvalitu služeb, nebo jestli počká a pokusí se stejný požadavek podat znovu o něco později.

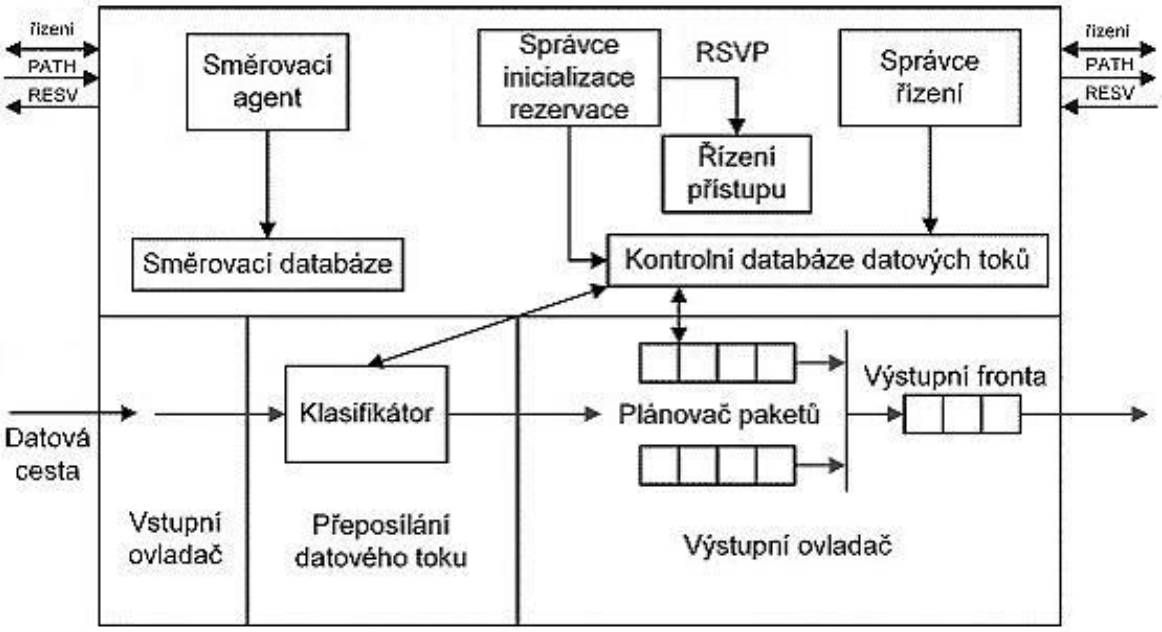
Základní částí modelu IntServ (viz obr. 2.1) jsou: plánovač paketů (Packet scheduler), kontrola přístupu (Admission control), klasifikátor (Classifier) a rezervační protokol (Reservation protocol) [5].

Plánovač paketů řídí zasílaní různých proudů paketů a musí být implementován v místě, kde jsou pakety řazeny do front.

Kontrola přístupu rozhoduje, zda novému toku může být přidělena rezervace, aniž by došlo k narušení předešlých záruk. Tato kontrola se spouští v každém uzlu.

Klasifikátor paketů identifikuje v hostitelích a směrovačích třídy paketů. S pakety, které byly zařazeny do stejné třídy, bude v plánovači paketů zacházeno stejně.

K rezervaci určitého množství prostředků šířky pásma po celé cestě slouží signalizační protokol rezervace prostředků RSVP (Resource Reservation Protocol). RSVP je nutný k vytvoření a udržování stavů v koncových zařízeních a ve směrovačích podél cesty. Protokol RSVP je velice komplikovaný. Pro každý datový tok je v něm definován odesílací a přijímací hostitel. Odesílatel zašle po síti zprávu PATH se seznamem zařízení po cestě. Po přijetí zprávy PATH odešle příjemce zpět odesílateli RSVP po stejné cestě zprávu s požadavkem RESV. Tato zpráva definuje parametry s požadovanými charakteristikami (šířka pásma, zpoždění, ztrátovost a další). Jakmile se na podpoře dané úrovně shodnou všechna zařízení, může komunikace začít. Po ukončení spojení nastane demontáž, při které se uvolní prostředky na všech rezervovaných zařízení. Proces rezervace se periodicky opakuje. Protokol RSVP není pro většinu prostředí tím nejvhodnějším řešením kvality služeb. Je totiž založen na vytvoření cesty přes celou síť, čímž klade velké nároky na směrovače [5].



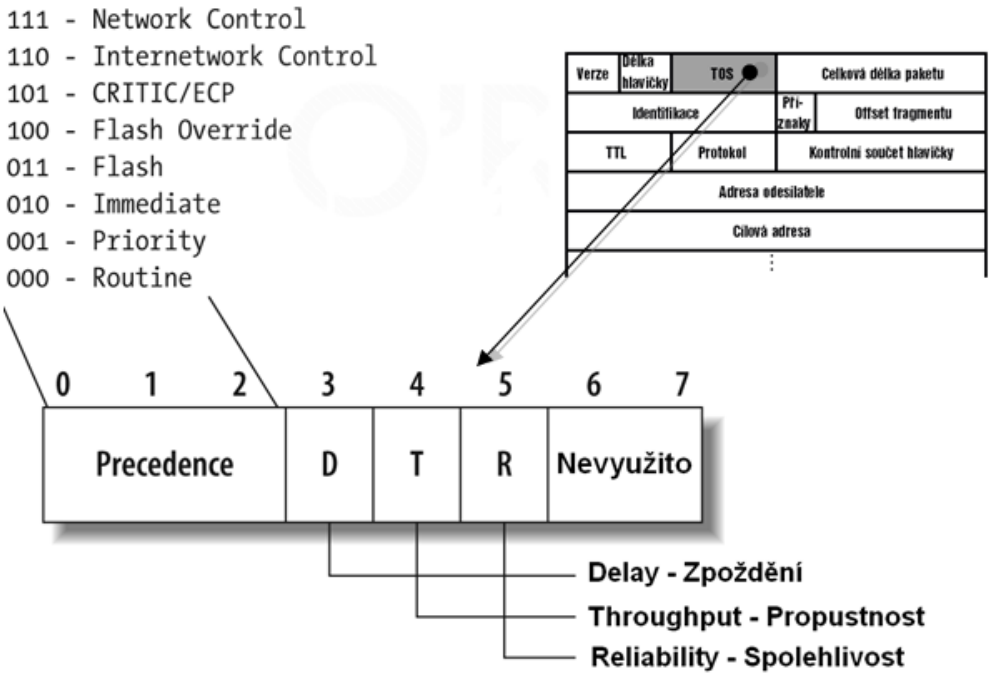
Obr. 2.1: Blokové schéma Integrované služby

3 Diferencované služby – DiffSserv

Technologie diferencovaných služeb (Differentiated Services), zkráceně DiffServ, je založen na prioritě šířky pásma. Tato metoda by se dala označit jako výhodnější. Protože na rozdíl od metody integrovaných služeb – IntServ, které využívají vyhrazení určité šířky pásma po celé cestě, zde nemusíme vymezovat určitou cestu přes celou síť, ale důležité pakety urychlíme a doručíme do cíle jako první. Tímto způsobem nedochází ke zbytečnému plýtvání šířky pásma. Kvalita služeb založená na prioritě je orientovaná na pakety. Způsob jakým bude zacházeno s paketem je definován informacemi uvnitř samotného paketu [13].

3.1 ToS pole

Pro roztřídění paketů do jednotlivých skupin se využívá 8-bitové pole typu služby ToS (Type of Service) [7], které je obsaženo v hlavičce protokolu IPv4. ToS (viz obr. 3.1) je tvořen třemi bity priority (IP Precedence). Tyto bity přiřazují paketu určitou úroveň přednosti v síti. Důležitější pakety se upřednostní před pakety s nižší prioritou. Další tři bity označují zpoždění, propustnost a spolehlivost přenosu [7].

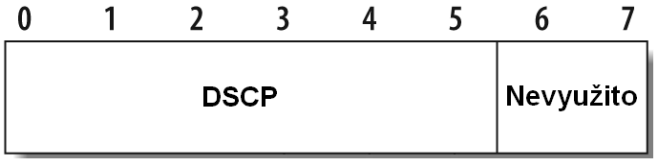


Obr. 3.1: Struktura ToS – Type of Service v IPv4

3.2 DS pole

Architektura DiffServ mění význam původního pole typu služby ToS podle svého vlastního schématu. Toto nové pole se označuje DS (Differentiated Services)

(obr. 3.2). Prvních šest bitů se používá pro označení paketů (DSCP - Differentiated Service CodePoint). Další dva zůstávají nevyužity (CU - Currently Unused). V budoucnu by mohli být využívány pro oznámení zahlcení. Nové schéma implementuje v šestibitovém prostoru pro označení tzv. kódové body [3]. Pakety se označí jednotlivými třídami DiffServ v okamžiku vstupu do sítě (okrajový směrovač) s technologií kvality služeb založené na DiffServu.



Obr. 3.2: DS pole

3.2.1 DSCP

Pro zápis DSCP se využívá šest bitů, kde nultý bit je nejvýznamnější a poslední pátý bit je nejméně významný. Těchto šest bitů v poli DSCP poskytuje 2^6=64 kombinací. Tyto kombinace byly rozděleny do tří skupin [7].

1. skupina: Poslední bit pole DSCP (tedy pátý bit) má hodnotu „0“. V ostatních pěti bitech může hodnota nabývat „0“ nebo „1“. Tomuto stavu odpovídá 32 kombinací. Ty jsou využívány pro standardní činnosti [7].

2. skupina: Poslední dva bity pole DSCP mají pevně danou hodnotu „11“. U zbývajících čtyř bitů může být hodnota „0“ nebo „1“. Do této skupiny patří 16 kombinací. Jsou využívány pro experimentální a lokální účely. Mají pouze místní význam, mimo intranet by nebyly rozpoznány [7].

3. skupina: Poslední dva bity pole DSCP mají pevně danou hodnotu „01“. U zbývajících čtyř bitů může být hodnota „0“ nebo „1“. Této skupině odpovídá také 16 kombinací. Podobně jako u 2. skupiny jsou tyto kombinace využívány pro experimentální a lokální účely. Rozdíl je v tom, že můžou být využity i pro standardní činnosti, tedy pokud je to nutné [7].

Je potřeba, aby DS uzly byly zpětně kompatibilní i s tradičními směrovači, které běží na protokolu IPv4 a mohou podporovat prioritu (IP Precedence) pole ToS. Tyto tři bity umožňují přiřadit osmi různým typům dopravy přednost v sítích. Osm DSCP kombinací z 1. skupiny se používá pro tento účel a je označováno jako Class Selector Code Points (CSCP). Poslední tři bity CSCP jsou nastaveny na „000“. Z tohoto důvodu mají CSCP formu „XXX000“, kde X je „0“ nebo „1“ [7].

3.3 PHB

Technologie DiffServ se snaží řídit pouze chování tzv. síťových přechodů. Po nadefinování příslušných síťových zásad v celé síti se již vše odehrává pouze uvnitř

zařízení. DiffServ využívá BA (Behavior Aggregate), který vybírá pakety pouze podle hodnoty DSCP, a MF (Multi-Field) klasifikátor, který vybírá pakety podle jedné nebo více hodnot obsažených v hlavičce paketu. Podle těchto hodnot je provoz namapován do PHB (Per-Hop Behaviors). PHB identifikuje prioritu zpracování paketů ve směrovačích, označuje tzv. chování uzlu. Implicitně definované úrovně služeb neboli třídy chování uzlu PHB jsou Standardní a Class Selector, EF a AF [7].

1. Standardní a Class Selector označuje osm tříd priorit, jsou uvedeny v tabulce (tab. 1), určených prvními třemi bity v ToS poli, tzv. CSCP [5].

Tab. 1: Osm tříd priorit podle hodnoty DSCP/CSCP

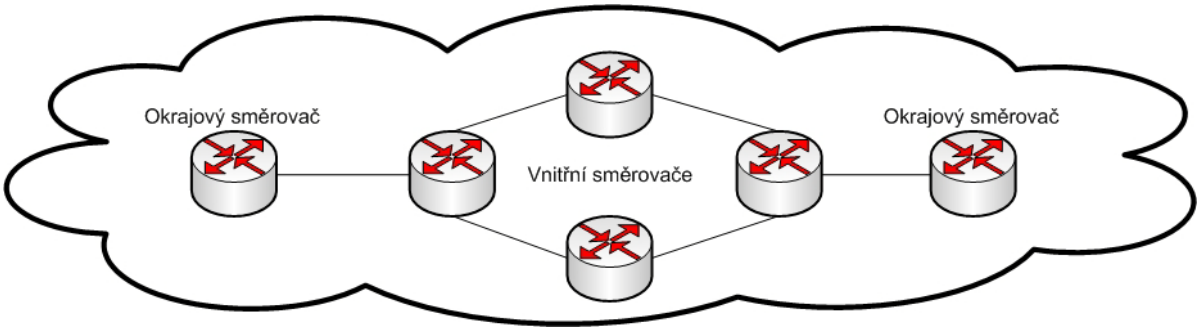
hodnota DSCP/CSCP	priorita
000000	Best-Effort
001000	nejnižší priorita
010000	6. nejvyšší priorita
011000	5. nejvyšší priorita
100000	4. nejvyšší priorita
101000	3. nejvyšší priorita
110000	2. nejvyšší priorita
111000	nejvyšší priorita

2. Urychlené doručení (Expedited Forwarding, EF) minimalizuje zpoždění a kolísání přenosu. Jestliže síťový provoz převyšuje maximální hodnotu zatížení, začnou se zahazovat pakety [3].

3. Zaručené doručení (Assured Forwarding, AF) definuje čtyři podtřídy. Každá podtřída obsahuje tři priority. Celkem tak vzniká 12 bodů. Jestliže síťový provoz převyšuje úroveň stanovenou v lokální zásadě, pak se nadbytečné pakety, které jsou označeny třídou AF, nedoručí se stanovenou prioritou, ale převedou se do nižší priority a poté se doručí. Nedochází k jejich zahození [3].

3.4 DiffServ doména

Obecně lze říci, že doména v IP sítích odkazuje na zeměpisné oblasti s hranicemi, na které se implementují určité schopnosti či postupy řízení provozu. IP doména je IP síť, která je pod kontrolou jedné řídící instance. IP domény mohou být tvořeny z několika sítí, které jsou geograficky rozptýleny, ale pod stejnou instancí. IP doména, která je schopná podporovat DiffServ, se označuje DiffServ doména (zkráceně DS doména) [7]. DS doména je tedy část nějaké rozsáhlé sítě s vlastní správou DiffServ. Na obrázku 3.3 je znázorněn základní model DiffServ domény a jeho klíčové prvky.



Obr. 3.3: DiffServ doména

3.4.1 Směrovače v DiffServ doméně

DS doména rozlišuje tři typy směrovačů: okrajový směrovač, vnitřní směrovač a hraniční směrovač [12].

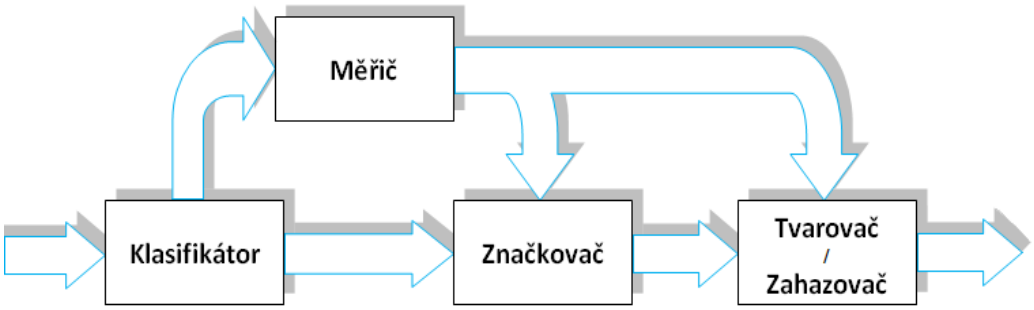
1. Okrajový směrovač (edge router) leží na hranici DS domény a sítě, která neoznačuje jednotlivé pakety (nepodporuje DiffServ). Okrajový směrovač se dále dělí na vstupní (ingress) a výstupní (egress). Tento směrovač je nejdůležitější, protože se zde zpracovávají a klasifikují pakety. Každému paketu se předá jeho značka [10].

2. Vnitřní směrovač (core router) neprovádí žádnou klasifikaci paketů. Pouze pakety předá dál s garancí služby podle jeho značky [10].

3. Hraniční směrovač leží na rozhraní dvou DS domén. Tyto DS domény mohou mít jiná klasifikační pravidla. Způsob jakým bude naloženo s pakety přicházející z první DS domény záleží na dohodě mezi těmito doménami. Většinou se provede rekvalifikace paketů [10].

3.5 Model Diffserv

Referenční model technologie DiffServ (viz obr. 3.4) se stará o úpravu provozu. Prvek, který upravuje provoz, musí být součástí DiffServ. Úprava provozu určuje jak má být zacházeno s dopravou tak, aby data vstupující do DS domény byla přizpůsobena podle pravidel specifikovaných v TCA (Traffic Conditioning Agreement). TCA je dohoda o úpravě provozu, která specifikuje parametry pro každou úroveň služeb. Tato úprava provozu se provádí na okrajových směrovačích DS domény. Referenční model DiffServ se skládá z těchto částí: klasifikátor, měřič, značkovač a tvarovač/zahazovač [5].



Obr. 3.4: Model DiffServ - blokové schéma úpravy provozu

3.5.1 Klasifikátor

Klasifikátor (Traffic Classifier) vybírá pakety podle obsahu určité části hlavičky a zařazuje je do jednotlivých tříd PHB. Klasifikátor musí být nastaven podle příslušného TCA. Existují dva základní druhy klasifikátorů [5].

1. BA (Behavior Aggregate) klasifikátor je nejjednodušší klasifikátor. Vybírá pakety pouze podle hodnoty DSCP v DS poli. Hodnota DSCP musí být již danému paketu přiřazena, nejčastěji tak bývá u předešlého směrovače. V případě, že paket má hodnotu DSCP, která není nastavena v příslušné TCA, je přiřazen ke standardní agregaci (tj. Best-Effort) [7].

2. MF (Multi-Field) klasifikátor vybírá pakety podle jedné nebo více hodnot obsažených v hlavičce paketu. Jsou to hodnoty zdrojové adresy, cílové adresy, DS pole, ID protokolu, čísla zdrojového a cílového portu a další [7].

3.5.2 Měřič

Měřič parametrů datového toku (Traffic Profile Meter) změří dočasné vlastnosti proudu paketů, která byla vybrána klasifikátorem, vůči stanovenému profilu v TCA. Měřič poté pošle informaci o stavu datového toku k značkovači a tvarovači. Pokud datový tok vyhovuje stanovenému profilu, může být vpuštěn do sítě. Ale pokud datový tok přesahuje stanovený limit, nesmí vstoupit do sítě a musí se dále upravit. Mezi tyto úpravy patří přeznačení, tvarování a zahození [5].

3.5.3 Značkovač

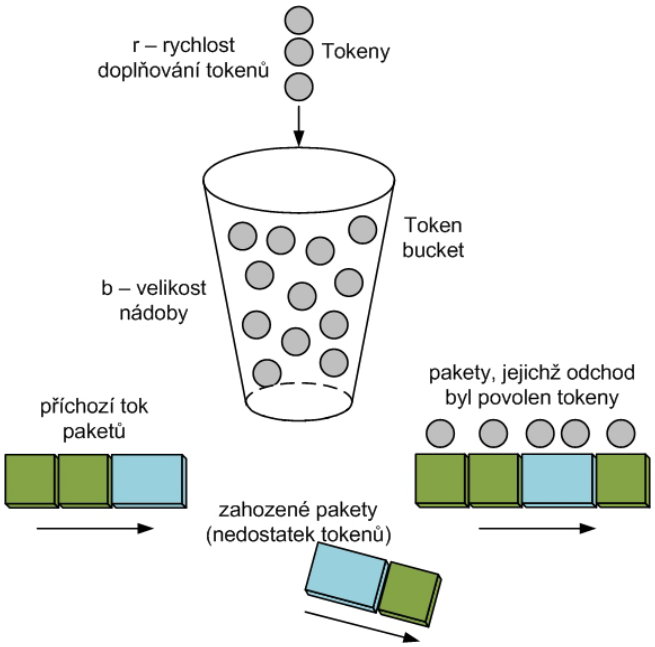
Značkovač (Marker) přidělí paketu určitou hodnotu DSCP v poli DS a tím přiřadí paket k určité třídě použitého PHB. Dále zde může probíhat přeznačení paketů. Při přechodu z jedné DS domény do druhé může být hodnota DS pole změněna na jinou značku se stejným významem (když odlišná DS doména, používá odlišné značky pro stejné zpracování paketů) nebo na jinou značku s jiným významem (když následující DS doména nemůže zaručit zacházení s pakety použité v předešlé DS doméně) [5].

3.5.4 Tvarovač

Tvarovač (Shaper) dokáže řídit charakteristiky provozu vpouštěného do sítě. Řídí objem paketů, které do sítě vstupují, a rychlost jejich vysílání. Výsledný síťový provoz, pak odpovídá požadovanému. Pro tvarování síťového provozu se používá mechanismus nádoby tokenů [5].

Token bucket

Nádoba tokenů (Token bucket) (obr. 3.5) řídí rychlost přenosu podle přítomnosti tzv. tokenů v nádobě. Token je abstraktní jednotka, měřená v bajtech, která musí být v libovolném okamžiku k dispozici pro další paket, který má z fronty FIFO (klasická fronta se zachováním pořadí) opustit síťové rozhraní. Tokeny v nádobě musí obsahovat nejméně tolik bajtů, kolik jich obsahuje přenášený paket. Tokeny jsou do nádoby doplňovány stálou rychlostí, dokud se nádoba nenaplní. Token bucket lze popsat dvěma parametry: rychlostí doplňování paketů a velikostí nádoby [12].



Obr. 3.5: Token bucket - princip

3.5.5 Zahazovač

Zahazovač (Dropper) zahazuje pakety (paket je směrovačem ignorován, není poslán dále a o jeho zahození není odeslána žádná zpráva) při překročení dohodnutých podmínek. Mezi tyto podmínky může patřit např. překročení objemu přicházejících dat nebo vyčerpání kapacity fronty. Tento proces se také nazývá policing (hlídání) [5].

4 Access Control List

Při tvorbě modelu DiffServ byl použit rozšířený model ACL (Access Control List, dále jen ACL) k identifikaci a klasifikaci provozu podle různých kriterií (zdrojový/cílový protokol, port, IP adresa, hodnota DCSP a další).

ACL je seznam jednoduchých pravidel, která řídí přístup k nějakému objektu. ACL se hlavně používá k filtraci paketů, tedy k řízení síťového provozu. Dále slouží k blokování nebo povolení provozu, kontrole šířky pásma, vynucení síťové politiky, identifikaci a klasifikaci provozu.

ACL, ve své nejjednodušší podobě, testuje pakety podle souboru pravidel. Když paket splňuje pravidla je zamítnut nebo povolen (záleží na nastavení pravidla: permit = povolen; deny = zamítnut). Pokud paket nesplňuje podmínky pravidla je testován, dokud nebude nalezena shoda. Na konci každého ACL je implicitní pravidlo – deny all. To znamená, že paket, který projde až na konec, je automaticky zahozen.

ACL se nejčastěji dělí na dva typy. První typ je standardní ACL (Standard ACL). A druhým typem je rozšířené ACL (Extended ACL).

Standardní ACL je starším pravidlem. Umožňuje méně konfigurací. Bývá označováno čísly od 1 do 99 a v rozšířeném módu od 1300 do 1399. Filtruje pouze podle zdrojové adresy a masky. Je jednoduší na konfiguraci.

Rozšířené ACL umožňuje podrobnější nastavení. Filtruje řadu položek v hlavičce vrstvy. Mezi tyto položky patří protokol, zdrojová/cílová adresa, maska a port. Čísla rozšířeného ACL jsou v rozsahu 100-199 a 2000-2699 v rozšířeném módu [9].

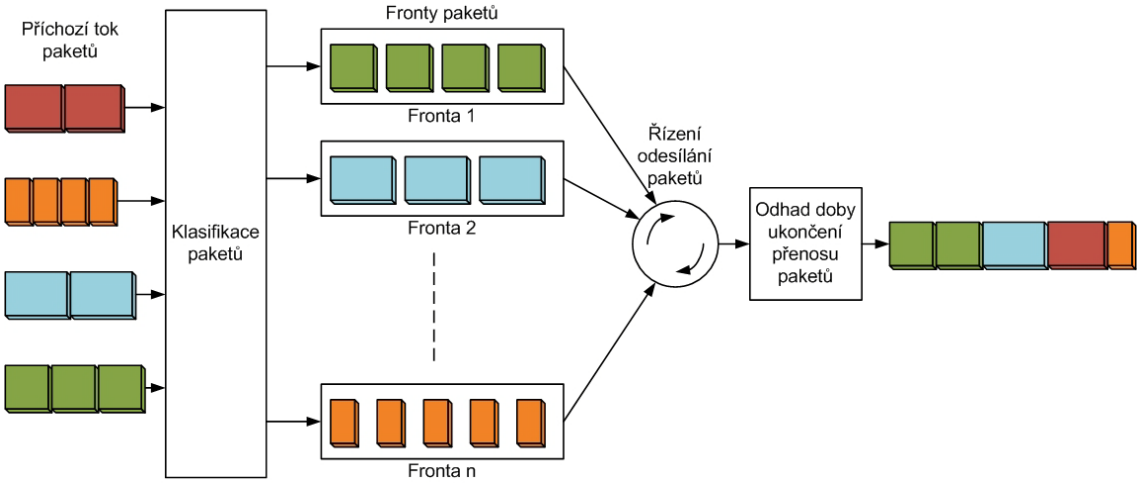
5 Fronta typu WFQ (Weighted Fair Queuing)

V modelu DiffServ byla fronta typu WFQ (Weighted Fair Queuing, dále jen WFQ) využita k řazení paketů do front na okrajových a vnitřních směrovačích. Z několika typů front (FIFO, PQ, …) byla tato metoda vybrána dle teoretických znalostí jako nejlepší varianta pro zajištění kvality služeb.

Metoda váženého řazení front (WFQ) zajišťuje paketům síťovou propustnost podle jejich priority neboli jejich váhy. Při zahlcení sítě se rozhoduje o tom, který paket bude zahozen a který dostane přednost. Toto se rozhoduje podle bitů v poli typu služby. Pokud dojde k zahlcení směrovače a pokud se další pakety, které již není možné odeslat, dále snaží vstoupit do směrovače, směrovač je odstřihne (zahodí). Metoda WFQ zajišťuje, že směrovač zahodí pakety s vyšší prioritou s menší pravděpodobností než pakety s nižší prioritou. WFQ užívá pro svoji činnost statický algoritmus s lineárním výsledkem. Je definováno osm priorit paketů (0 až 7) a tedy pravděpodobnost zahození paketu, který má prioritu 4, je čtyřikrát menší než u paketu s prioritou 0.

Metoda WFQ (obr. 5.1) přiřazuje každému datovému toku váhu. Váha určuje pořadí vysílání paketů. Jako první bude odeslán paket s největší vahou.

Rozšířenější verzí metody WFQ je metoda řazení front podle tříd (Class-Based Weighted Fair Queuing, CBWFQ). Při použití této metody máme možnost přesnější kontroly síťového provozu pomocí priorit [7].



Obr. 5.1: Metoda WFQ

6 Směrovací protokol OSPF (Open Shortest Path First)

V druhé části práce zaměřené na simulace výpadku hlavní linky a využití alternativní linky byl zvolen a nastaven směrovací protokol OSPF (Open Shortest Path First, dále jen OSPF).

OSPF byl vytvořen organizací IETF. Specifikace je dostupná v RFC. První verze OSPF v RFC 1131, druhá verze OSPFv2 v RFC 2328 a třetí verze OSPFv3 založená na IPv6 v RFC 5340 [8].

Protokol OSPF je dynamický směrovací protokol zaměřený na stav linky, označovaný též algoritmy nejkratší cesty (shortest path first, SPF). Tyto protokoly sledují v síti stav linek, ze kterých se skládají jednotlivé cesty. Směrovače vysílají každým rozhraním Hello pakety protokolu Handshake a snaží se navázat spojení s nejbližšími směrovači, sousedy (neighbour). Po výměně Hello paketů a dohodnutí společných parametrů se směrovače označují jako sousedící (adjacent). Sousedící směrovače si mezi sebou vyměňují oznámení o stavu linky (link-state advertisement, LSA). LSA jsou šířeny všemi směry (tzv. záplavové šíření). Každý směrovač si uloží přijaté LSA do své topologické databáze a přepošle ho dál. Takto se postupně rozšíří mezi všechny směrovače v síti a vznikne stejná topologická databáze ve všech směrovačích. Pokud směrovač naplní svou databázi, spustí algoritmus stavu linek, vypočítá nejkratší cestu do cíle a odstraní smyčky v topologii sítě. K výpočtu stavu linek se využívá Dijkstrův (SPF) algoritmus. Trasy vypočítané algoritmem SPF neobsahují uzavřené smyčky. Algoritmus stavu linek zatěžuje o hodně více směrovače než protokoly pracující s vektorem vzdálenosti, kde stačí pouhé přičtení lokální vzdálenosti ke kumulované hodnotě (RIP). Nově vypočtené cesty se zapíší do směrovací tabulky. Při jakékoli změně stavu některé linky si směrovače mezi sebou vymění oznámení o stavu linky. Tato informace se rozšíří do celé sítě a každý směrovač upraví svoji topologickou databázi a vykoná nový výpočet pomocí Dijkstrova (SPF) algoritmu [13].

Zprávy protokolu OSPF, které si směrovače mezi sebou posílají, jsou přenášena přímo v protokolu IP. Za IP hlavičkou následuje hlavička OSPF, která určuje verzi a typ OSPF paketu. Poté následují informace specifické pro jednotlivé typy OSPF paketů [4]:

- Hello pakety – slouží k vytváření a udržování sousednosti mezi směrovači.
- Database Description pakety – popisují topologickou databázi odesílajícího směrovače a tak soused ví, jestli má odpovídající položky ve své databázi nebo si je musí vyžádat.
- Link State Request paket – vysílá se tehdy, pokud některé LSA chybí nebo je již zastaralé.
- Link State Update paket – používá se pro šíření LSA a jako odpověď na Link State Request paket.
- Link State Acknowledgement – slouží k potvrzení LSA. Každé LSA musí být potvrzeno.

K přenosu se využívají linky s nejmenší hodnotou celkové metriky, označované také jako cena (cost). Hodnota metriky linky se pohybuje v rozmezí 1 až 65535 a celková hodnota metriky cesty přenosu není nijak omezena. Jednotlivé hodnoty

metrik lze odvodit od šířky pásma, zpoždění linky, vzdálenosti, zabezpečení a dalších. V praxi se nejčastěji metrika odvozuje od šířky pásma (bandwidth) podle vztahu

$$cena\ (cost) = \frac{100000000}{\text{šířka pásma (bandwidth)[bps]}}. \quad (6.1)$$

Protokol OSPF podporuje alternativní trasy. To znamená, že se při výpadku hlavní linky snaží najít alternativní trasu využitím náhradních linek. Součástí alternativních tras je i podpora rozložení zátěže, kdy se zátěž rozloží mezi více alternativních linek se stejnou metrikou.

Protokol OSPF obsahuje čtyři základní časovače. Jsou to HelloInterval, RouterDeadInterval, Transmission Delay a Retransmission Interval. Hello pakety jsou vysílány periodicky v rozmezí HelloIntervalu. Tento interval je v LAN sítích nastaven na 10 sekund. Pokud směrovač neobdrží od sousedního směrovače paket Hello po dobu RouterDeadIntervalu, stanoví spojení se sousedním směrovačem za nefunkční. Délka RouterDeadIntervalu je v LAN sítích 4·HelloInterval, tedy 40 sekund. Transmission Delay slouží ke zvětšení doby LSA pro pomalé přenosy a šíření. Velikost Transmission Delay je přednastavena na 1 sekundu. Retransmission Interval určuje dobu mezi opětovným zasláním žádosti k sousednímu směrovači. OSPF posílá žádosti na sousední směrovač a očekává od něho potvrzení. Pokud není potvrzení přijato, směrovač opakovaně zašle žádost na sousední směrovač. Velikost Retransmission Intervalu je standardně 5 sekund [4].

Každý směrovač, který generuje LSA zprávy, musí opakovaně vysílat tuto zprávu minimálně jednou během 30 minut svým sousedům.

OSPF umožňuje rozdělení rozsáhlé sítě na oblasti (area). Jednotlivé oblasti jsou označeny identifikátorem v rozsahu 0 až 65535. Páteřní síť tvoří oblast 0. Tato oblast propojuje všechny ostatní oblasti. Topologie oblasti a její změny se udržují pouze uvnitř oblasti a ostatní mimo oblast je nevidí. To vede ke snížení objemu provozu LSA a snížení výpočetní náročnosti Dijkstrova algoritmu [8].

Velkou předností směrování podle stavu linky je, že se LSA vysílá jen tehdy, kdy je to opravdu potřeba a nese informace pouze o změněných linkách, a proto tolik nezatěžuje síť, cestuje rychleji a je zde menší riziko smyček ve směrování. Tyto protokoly jsou řízené událostmi, a proto nemusí proces konvergence (dosažení shody-vybudování směrovací tabulky) čekat na vypršení určitých časovačů a síť je připravena vysílat skoro okamžitě.

Nevýhodou protokolu OSPF je jeho velká složitost a výpočetní složitost. Proto tyto protokoly potřebují velké množství pamětí a výkonnější hardware.

7 OPNET Modeler

Program OPNET Modeler (dále jen OM) je simulační prostředí, které slouží pro návrh, simulaci a analýzu odlišných síťových technologií. Tento program byl vyvinut firmou OPNET Technologies Inc.

OM je hierarchicky a objektově orientován. Grafické prostředí odráží reálné rozmístnění jednotlivých síťových prvků a na nejnižší úrovni je chování jednotlivých komponent zapsáno v jazyce C/C++. Dále obsahuje široké možnosti v oblasti simulace a analýzy výsledků. Této vlastnosti se využívá hlavně při ověření chování reálných objektů v různých podmínkách.

Výsledky simulací můžeme vygenerovat do souborů ve formátech XML, HTML nebo data uložit jako tabulku. OM obsahuje také prohlížeč animací, díky kterému je možné sledovat průběh simulace. Simulace v programu OM jsou oproti reálným hodnotám zrychlené. Takže je možné nasimulovat několika hodinové chování sítě během pár minut.

Velkou výhodou OM je, že všechny dostupné knihovny (např. síťových prvků, definic paketů, linkových vrstev apod.) mají dostupný zdrojový kód, takže ho lze upravovat, což je velice výhodné je-li potřeba např. vyzkoušet nové možnosti nějakého standardního prvku [6].

7.1 Základní prvky prostředí OM

Podsíť (subnet) se skládá ze stanic, aktivních síťových prvků, komunikačních linek apod.

Model uzlu (node model) tvořený ze základních funkčních bloků (např. zdroj, procesor a další).

Model procesu (process model), ve kterém jsou definovány jednotlivé procesy modelu uzlu (např. stavy procesu, události, přechody apod.) [6].

7.2 Základní editory prostředí OM

Editor projektu (Project Editor) je grafický editor, který modeluje topologii a komunikaci v síti. Síť obsahuje uzly a odkazy na konfigurovatelné objekty. Objekty můžeme vybrat z knihovny OM nebo si nadefinovat nové uzly a modely. Editor projektu bývá označován také jako pracovní plocha. Důležité oblasti pro sestavení a simulaci na pracovní ploše jsou hlavní menu, tlačítka, pracovní plocha, textová oblast a tipy [6]. Na obrázku 7.1 je zobrazeno hlavní menu a rozmístění tlačítek. V tabulce 2 jsou uvedena funkce tlačítek.

Obr. 7.1: Hlavní menu a tlačítka pracovní plochy prostředí OM

Tab. 2: Tlačítka prostředí OM

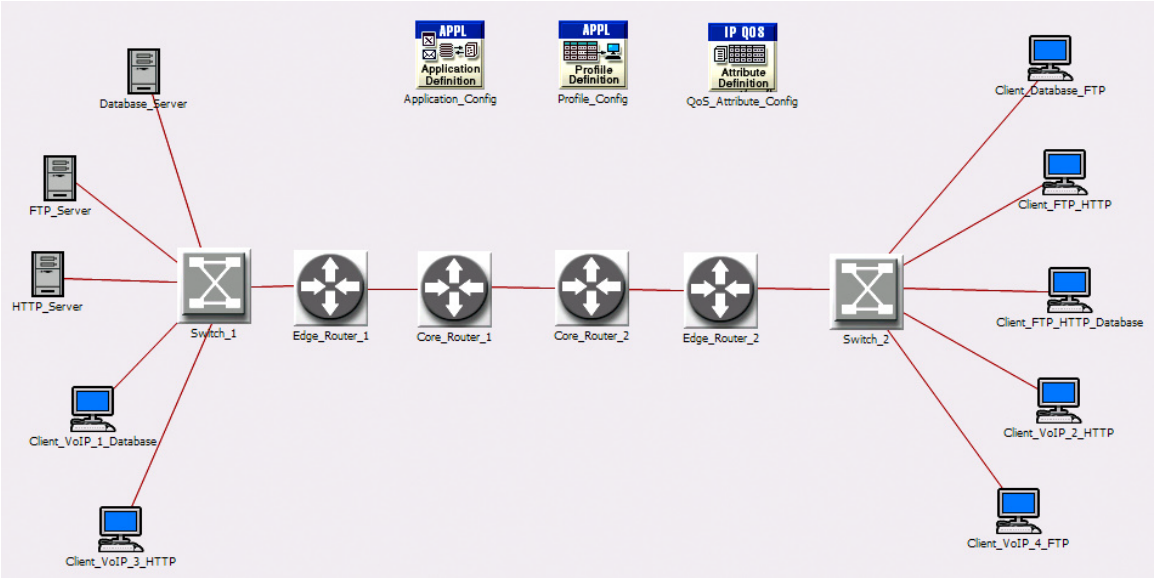
Číslo	Funkce tlačítka
1.	Nový (New)
2.	Otevřít (Open)
3.	Uložit (Save)
4.	Tisk (Print)
5.	Paleta objektů (Object Palette)
6.	Chybně vybraný objekt
7.	Ověřit shody odkazu
8.	O úroveň výš
9.	Zvětšit
10.	Zmenšit
11.	Vložení topologie z VNE serverů
12.	Otevřít okno pro sledování datových toků
13.	Generování zprávy a rozdílu scénáře
14.	Nastavení a spuštění simulace
15.	Ukázat výsledek simulace
16.	Skrýt / zobrazit panely grafů
17.	Seznam zařízení

Editor uzlu (Node Editor) je rozhraní nižší úrovně. Zobrazuje vnitřní strukturu síťového zařízení nebo systému a vzájemné vztahy mezi funkčními moduly a volanými funkcemi. Uzel je složen z modulů, které představují aplikace, protokolové vrstvy, algoritmy a fyzické prostředky.

Editor procesu (Process Editor) pracuje na nejnižší úrovni. Procesní editor je konečný stavový automat (FSM – Finite State Machines) [6].

8 Tvorba modelu DiffServ v prostředí OM

V této části práce bude popisována činnost při návrhu pokročilého simulačního modelu technologie DiffServ v prostředí OM. Model technologie DiffServ (viz obr. 8.1) se bude skládat ze třech serverů (server), sedmi klientů, dvou přepínačů (switch), dvou okrajových směrovačů (edge router) a dvou vnitřních směrovačů (core router). Pro směrování bude použit protokol pracující s vektorem vzdálenosti, tedy RIP. Budou zde použity tyto aplikace: internetová telefonie VoIP (Voice over Internet Protocol), www stránky pomocí protokolu HTTP (Hypertext Transfer Protocol), přístup do databáze - Database a přenos souborů pomocí protokolu FTP (File Transfer Protocol). Na okrajových směrovačích bude probíhat identifikace a klasifikace paketů podle použité aplikace. K identifikaci bude použit rozšířený typ ACL. Podle klasifikace bude paketům přiřazena DSCP značka a podle této značky je s pakety zacházeno v celém modelu. Dále bude model rozšířen o alternativní trasu, která bude obsahovat dva vnitřní směrovače. A uvnitř tohoto modelu bude nastaven směrovací protokol OSPF, který umožní využít alternativní linku při výpadku hlavní linky.



Obr. 8.1: Výsledný model technologie DiffServ

8.1 Vložení a nastavení jednotlivých prvků modelu

Před sestavením samotného modelu DiffServ byl nejdříve vytvořen nový projekt. Byla vybrána položka **File – New…** a zvolen nový projekt, scénář a objekty.

Objekty byly vkládány pomocí tlačítka **Paleta objektů** (tlačítko číslo 5 v prostředí OM (tab. 2)). Zde byla vybrána knihovna **internet_toolbox**. Z této knihovny byly vybrány tyto prvky:

- ethernet_server – model serveru,
- ethernet4_slip8_gtwy – model vnitřních a hraničních směrovačů,
- ethernet16_switch – model přepínačů,
- ethernet_wkstn – model pracovních stanic – klientů,

- 100BaseT - síťová technologie s rychlostí 100 megabitů za sekundu,
- 10BaseT - síťová technologie s rychlostí 10 megabitů za sekundu.

Prvky byly vhodně rozmístěny a pojmenovány. Klienti s přepínači a přepínače se směrovači byli propojeni síťovou technologií **100BaseT**. Mezi hraničními a vnitřními směrovači byla použita síťová technologie **10BaseT**.

Dále byly vloženy objekty, které definují globální nastavení. Jsou to tyto tři objekty:

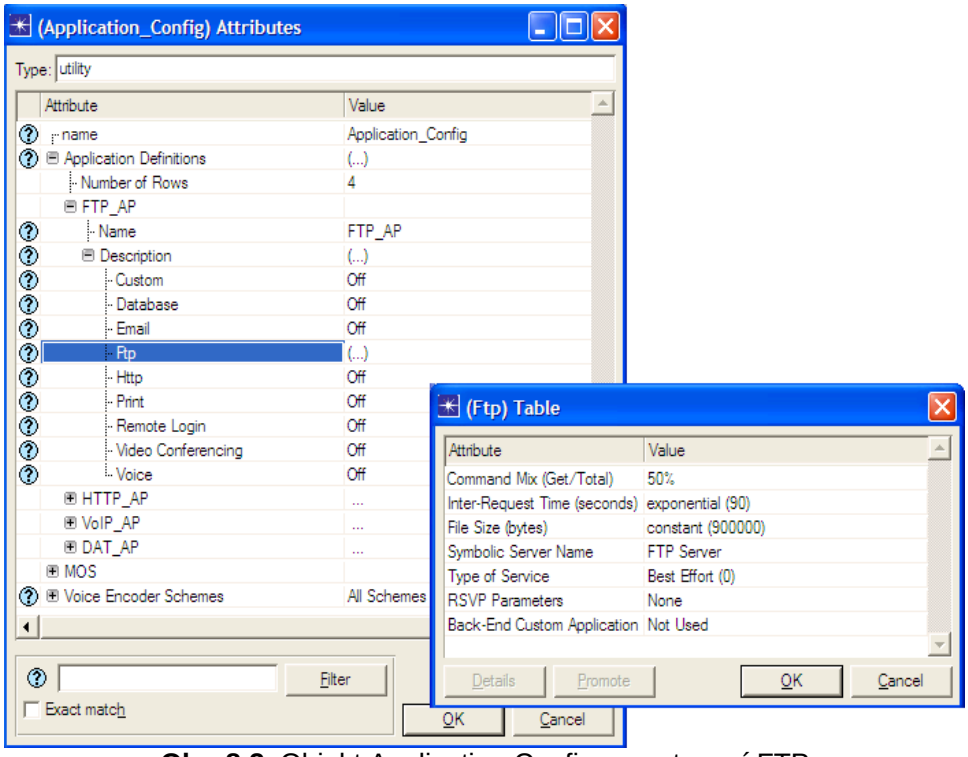
- Application Config – objekt pro nastavení aplikací, které se budou v modelu vyskytovat,
- Profile Config – objekt, který jednotlivým aplikacím přiřadí profily,
- QoS Atribute Config - objekt pro nastavení kvality služeb.

8.1.1 Objekt Application Config

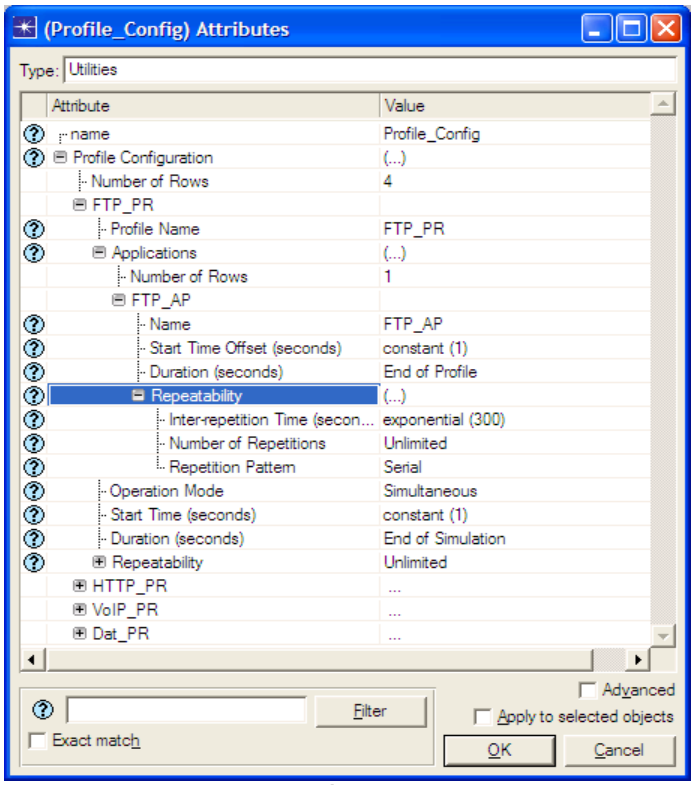
Pro nastavení jednotlivých aplikací byl použit objekt Application Config. Na tento objekt bylo kliknuto pravým tlačítkem a zvolena položka **Edit Attributes**. V záložce **Application Definitions** byl nastaven počet aplikací, hodnota **Rows**. U každé aplikace bylo zadáno jméno (**Name**) a v záložce **Decription** byla vybrána určitá aplikace a nastaveny vlastnosti s co největšími hodnotami přenosu. V prostředí OM mohou být vybrány tyto aplikace: Custom, Database, Email, FTP, HTTP, Print, Remote login, Video Conferencing a Voice. Nastavení FTP aplikace je zobrazeno na obrázku 8.2.

8.1.2 Objekt Profile Config

V objektu Profile Config byly přiřazeny nadefinovaným aplikacím jejich profily. Pro vytvoření profilu bylo opět kliknuto pravým tlačítkem a zvolena položka **Edit Attributes**. A podobně jako u objektu Application Config byla v záložce **Profile Configuration** zadána hodnota **Rows**, počet profilů. Každému profilu bylo zadáno jméno (**Name**) a v záložce **Applications** byl profil přiřazen k aplikaci vytvořené v Application Config. Dále by bylo možné nastavit dobu trvání aplikace, dobu spuštění aplikace či profilu, počet opakování a další. Položka **Edit Attributes** objektu Profile Config je na obrázku 8.3.



Obr. 8.2: Objekt Application Config - nastavení FTP

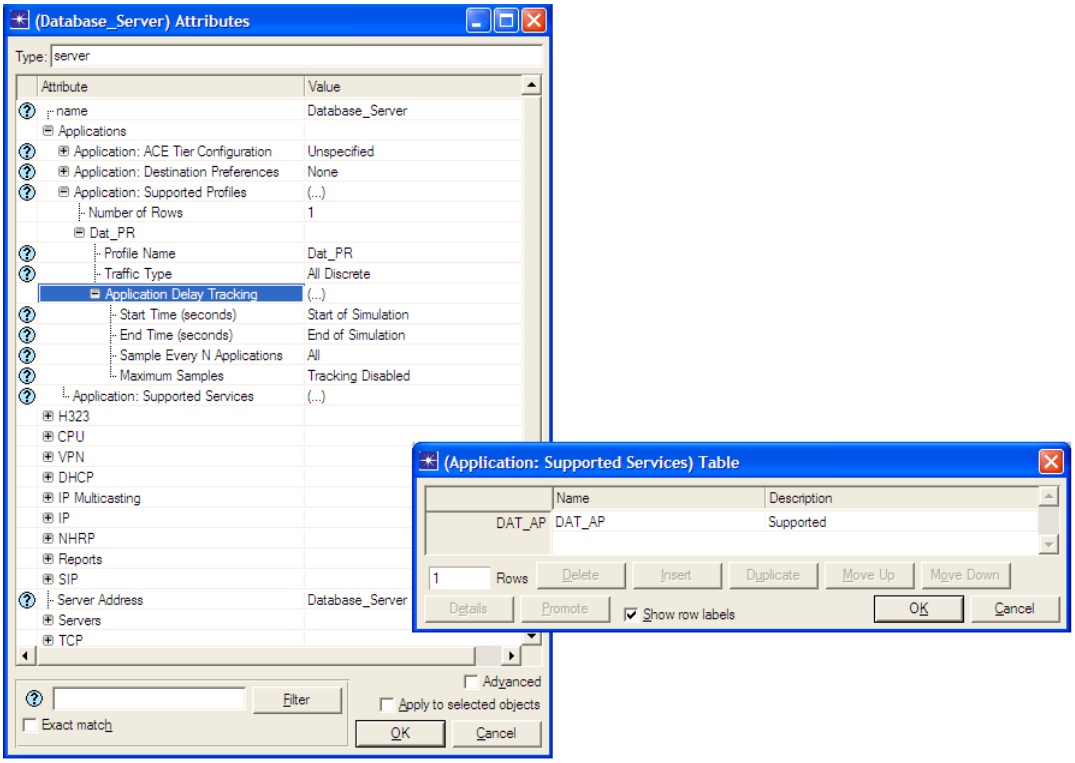


Obr. 8.3: Nastavení objektu Profile Config

8.1.3 Nastavení serveru

Serverům byly nastaveny jejich síťové adresy a aplikace. Na serveru byla vybrána položka **Edit Attributes**. Síťová adresa serveru (název serveru) byla zadána v položce **Server Address**. Tato adresa slouží k jednoznačné identifikaci serveru v síti.

Dále byla nastavena aplikace, kterou bude server poskytovat. To bylo provedeno v záložce **Applications**. Zde bylo zvoleno **Application: Supported Services** – **Edit** – **Rows**, počet aplikací. V položce **Name** byla vybrána aplikace, která již byla nadefinována v objektu Application Config. A v položce **Description** byla ponechána hodnota Supported. Dále byly serveru nastaveny profily, které bude server podporovat. Toto bylo uděláno podobným způsobem jako u aplikace. V záložce **Applications** bylo vybráno **Application: Supported Profiles** – **Edit** – **Rows**, počet profilů. A v položce **Traffic Name** byl vybrán příslušný profil. Nastavení serveru jako databázového serveru je uvedeno na obrázku 8.4.



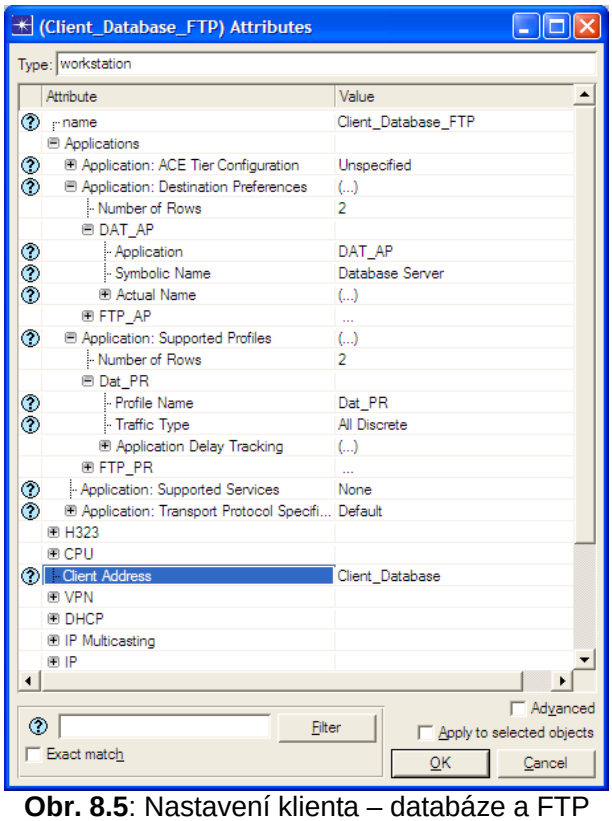
Obr. 8.4: Nastaveni serveru - databáze

8.1.4 Nastavení klienta

Klientům byly nastaveny profily, které budou podporovat, a cílové servery, se kterými budou komunikovat. Nejdříve byly nastaveny profily aplikací, které klient bude používat. Byl vybrán klient a na něm **Edit Attributes**. V záložce **Application** bylo vybráno **Application: Supported Profiles** – **Edit** – **Rows**, počet profilů. V **Profile Name** byl zvolen určitý profil.

Cílový server byl nastaven v záložce **Application** – **Application: Destination Preferences** – **Edit** – **Rows**, počet podporovaných aplikací. V poli **Application** byla

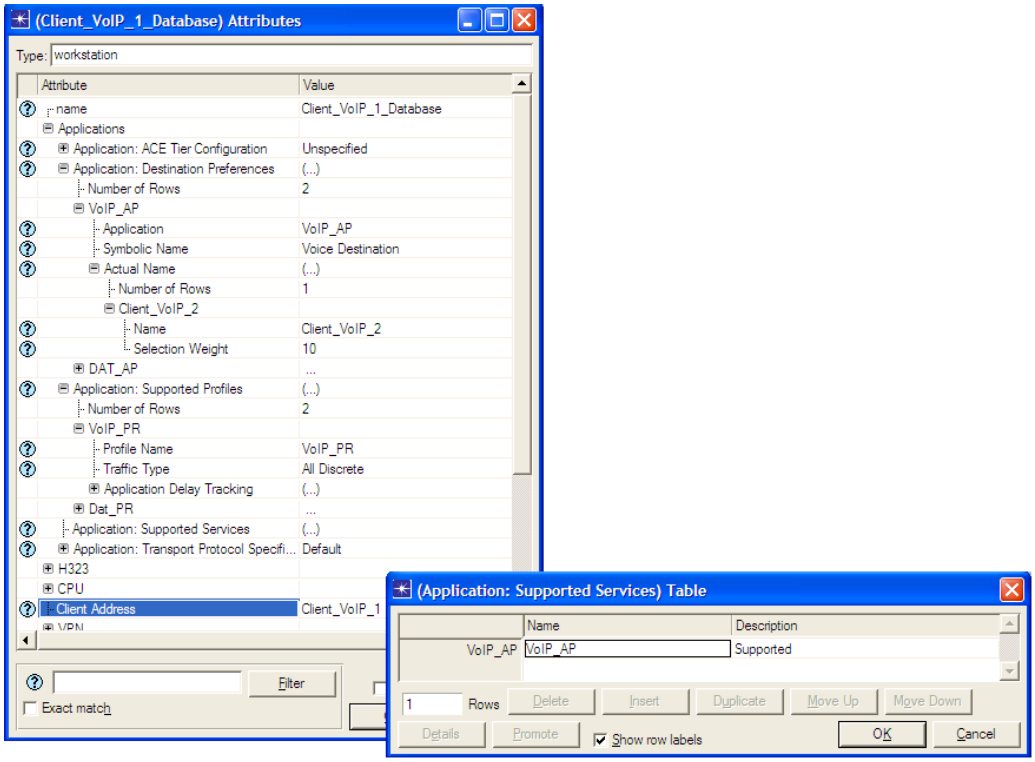
zvolena aplikace a položce **Symbolic Name** byl přiřazen název cílového serveru. Nastavení klienta je ukázáno na obrázku 8.5.



Obr. 8.5: Nastavení klienta – databáze a FTP

Nastavení VoIP klienta

Aplikace VoIP se liší od ostatních aplikací tím, že je provozována mezi dvěma klienty. Ostatní aplikace se provozují mezi klientem a serverem. Z tohoto důvodu bylo potřeba nastavit VoIP klienta, aby se choval jako klient i server zároveň. VoIP klient byl nastaven podobně jako jiní klienti (viz kapitola 8.1.4). Jen u cílového serveru pro VoIP v záložce **Application** – **Application: Destination Preferences** – **Edit** – **Rows** – **Symbolic Name** byl uveden místo serveru klient, se kterým bude příslušný klient komunikovat. Proto bylo nutné nastavit síťovou adresu VoIP klienta v položce **Client Address**. Navíc byla nastavena v záložce **Applications** – **Application: Supported Services** – **Edit** – **Rows**, hodnota 1, a v položce **Name** vybrána aplikace VoIP. VoIP klient je zobrazen na obrázku 8.6.



Obr. 8.6: Nastavení VoIP klienta – VoIP a databáze

8.2 Vytvoření dalšího scénáře

Pro srovnání více metod třídění provozu bylo vytvořeno více scénářů. Bylo vybráno tlačítko **Scenarios** hlavního menu (viz obr. 7.1) a položka **Duplicate Scenario…**. Nový scénář byl pojmenován a upraven podle požadavků. Scénáře je možné dále spravovat v položce **Scenarios – Manage Scenarios…**.

8.3 Nastavení kvality služeb

Než bylo možné začít nastavovat kvalitu služeb, bylo potřeba vyřešit otázku, jak bude probíhat třídění provozu na okrajových směrovačích podle aplikací, aby výsledná kvalita služeb byla co nejlepší. Po bližším seznámení s prostředí OM bylo zjištěno, že nejlepší bude použít metodu ACL. Tato metoda umí třídit provoz podle několika kritérií (viz kapitola 4). V našem případě, kdy každý klient používá několik aplikací, by bylo nejlepší, kdyby každá aplikace obsahovala již svoji DCSP značku a na okrajových směrovačích by probíhala pouze klasifikace paketů. To bohužel v reálných podmínkách často nenastane. Dalším způsobem je nastavit ACL podle protokolu a portu, na kterém je jednotlivá aplikace posílána. V prostředí OM bylo zjištěno na jakých protokolech a portech (zdrojové i cílové) se jednotlivé aplikace posílají (viz tab. 3).

V dřívějších pracích zaměřených na obdobné téma bylo třídění paketů prováděno podle zdrojové a cílové IP adresy. Tento způsob třídění paketů může být výhodný tehdy, využívá-li klient pouze jednu aplikaci. Tedy v našem případě by

výsledná kvalita služeb byla velice nepřesná, protože by musela být vybrána jedna aplikace u každého klienta a ostatní aplikace by se museli zanedbat.

Z důvodu porovnání více metod třídění provozu byly vytvořeny, a poté i odsimulovány, všechny tři způsoby identifikace paketů (podle protokolu a portu, IP adresy a DSCP).

Dále bylo nutné vytvořit frontu provozu podle DSCP. K tomuto byla použita fronta WFQ (viz kapitola 5). Každé aplikaci byla přiřazena váha (viz tab. 3), která určuje pořadí vysílání paketů.

Tab. 3: Přiřazení značky a váhy aplikacím podle protokolu a portu

Aplikace	Protokol	Port	DSCP značka	Váha
VoIP	UDP	-	AF31	25
HTTP	TCP	80	AF21	15
Database	TCP	101	AF11	10
FTP	TCP	20, 21	BE	5

8.3.1 Scénář QoS s tříděním paketů podle protokolu a portu

V tomto scénáři probíhá klasifikace a identifikace paketů na okrajových směrovačích. Podle toho o jakou aplikaci se jedná (příslušný protokol a port) je jednotlivým paketům přiřazena hodnota DSCP (viz tab. 3). Podle této hodnoty je s paketem zacházeno v celé DS doméně.

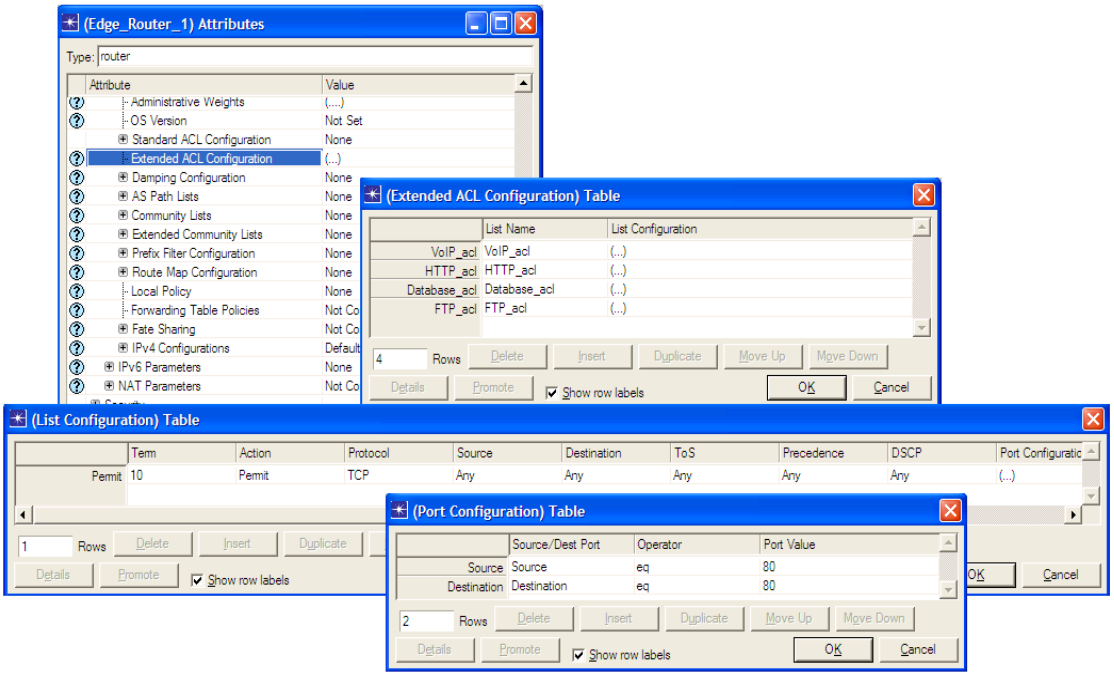
Na okrajových i vnitřních směrovačích bylo nastaveno řazení do front a odesílání paketů podle jejich hodnoty DSCP.

Nastavení ACL na okrajových směrovačích

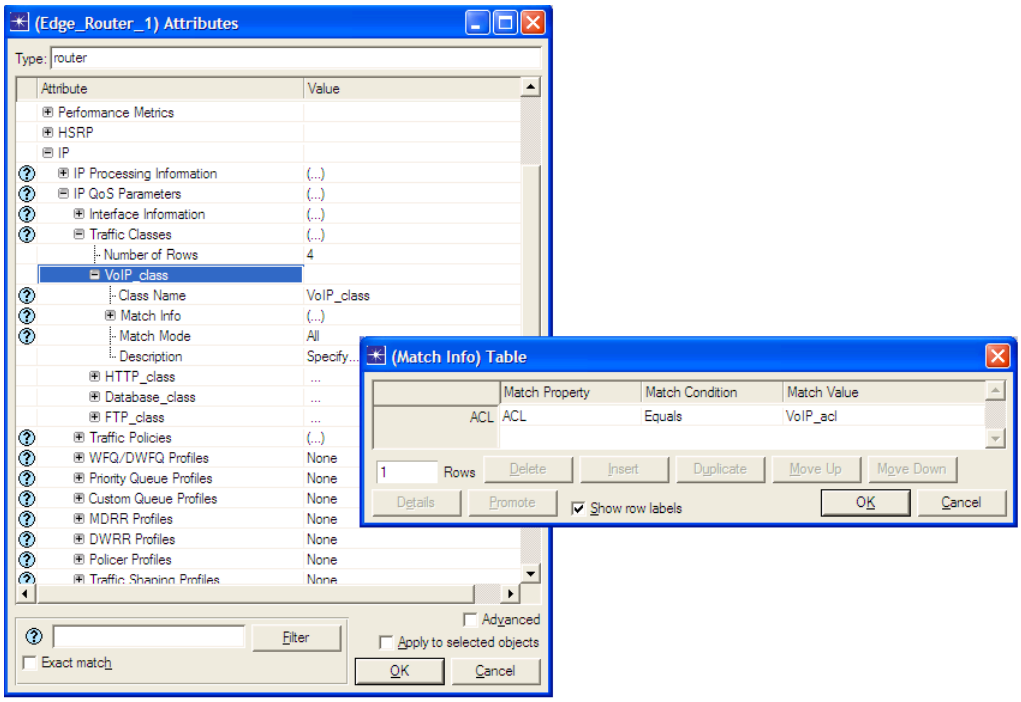
Na okrajový směrovač (edge router) bylo kliknuto pravým tlačítkem a byla vybrána položka **Edit Attributes**. Zde byla zvolena záložka **IP – IP Routing Parameters – Extended ACL Configuration – Edit – Rows**, počet ACL pravidel odpovídá počtu aplikací. Každé pravidlo bylo pojmenováno (**List Name**) podle aplikace a v položce **List Configuration** byly dále nastaveny hodnoty: **Action** – Permit, **Protocol** – příslušný protokol a **Port Configuration** – zdrojový (source) a cílový (destination) port aplikace. Pravidla ACL byla nastavena na obou okrajových směrovačích. Nastavení ACL pravidel na okrajovém směrovači (edge_router_1) je na obrázku 8.7.

Nastavení tříd provozu na okrajových směrovačích

V záložce **IP – IP QoS Parameters – Traffic Classes – Edit – Rows**, počet tříd provozu, byly vytvořeny třídy provozu, do kterých budou pakety zařazeny na základě ACL pravidel. Jednotlivé třídy byly pojmenovány a v položce **Match Info** byly nastaveny hodnoty: **Match Property** – ACL, **Match Condition** – Equals a **Match Value** – ACL pravidlo dané třídy. Nastavení tříd provozu je zobrazeno na obrázku 8.8.



Obr. 8.7: Nastavení ACL pravidel pro HTTP

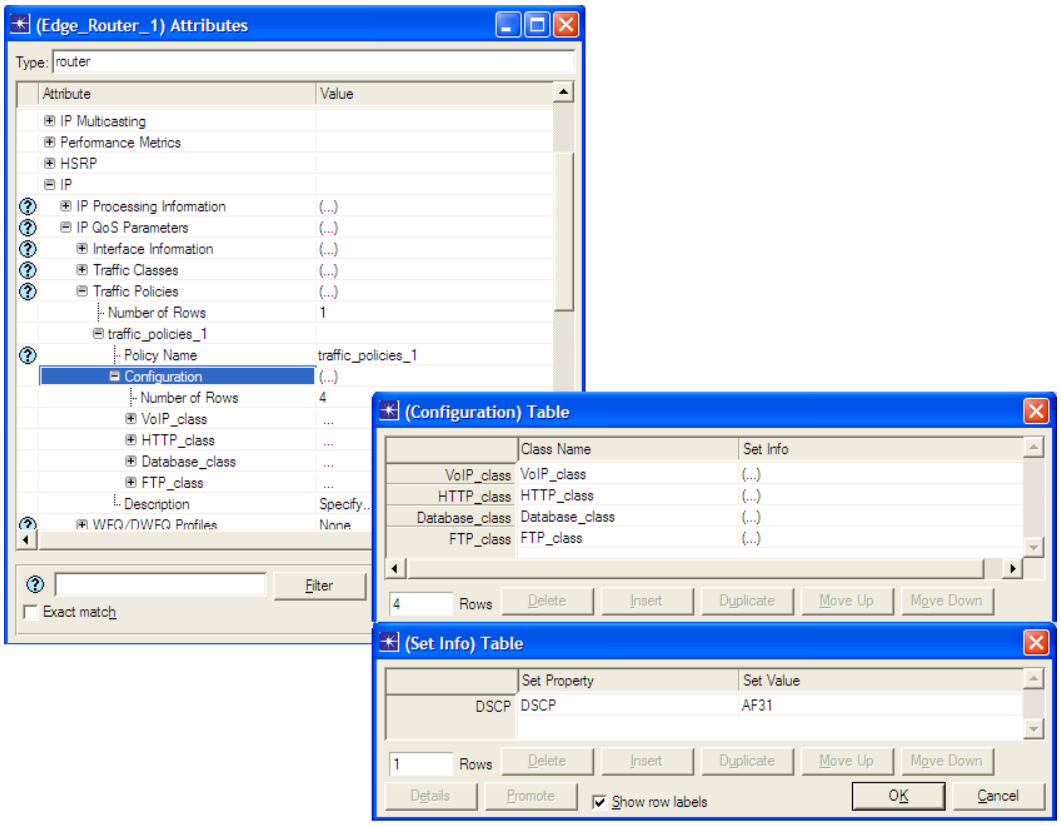


Obr. 8.8: Nastavení tříd provozu

Nastavení politiky provozu na okrajových směrovačích

V položce **Traffic Policies** byla přiřazena vytvořeným třídám provozu hodnota DSCP. Nastavení bylo provedeno opět na obou okrajových směrovačích. Byla vybrána záložka **IP – IP QoS Parameters – Traffic Policies – Edit – Rows**, hodnota 1.

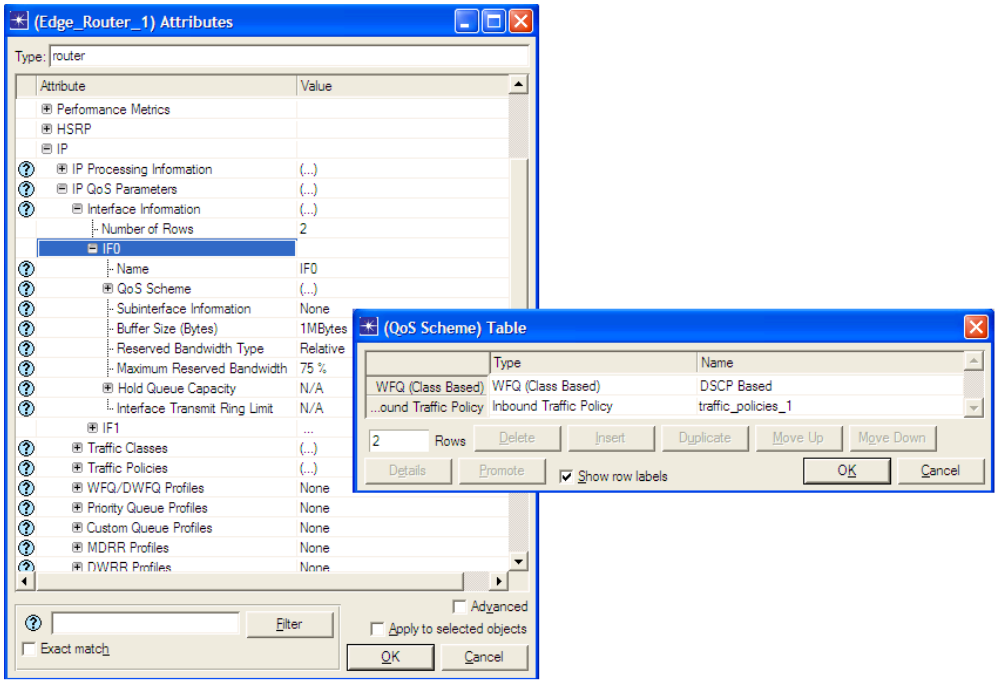
V položce **Policy Name** bylo zadáno jméno příslušné politiky. V záložce **Configuration – Edit – Rows**, počet tříd provozu, byly přiřazeny hodnoty DSCP pro danou třídu provozu. V záložce **Set Info** byly každé třídě provozu zadány hodnoty: položka **Set Property** – DSCP a **Set Value** – hodnota DSCP pro určitou aplikaci (viz tab. 3). Příklad nastavení položky Traffic Policies pro VoIP je uvedeno na obrázku 8.9.



Obr. 8.9: Nastavení politiky provozu na okrajovém směrovači

Nastavení rozhraní na směrovačích

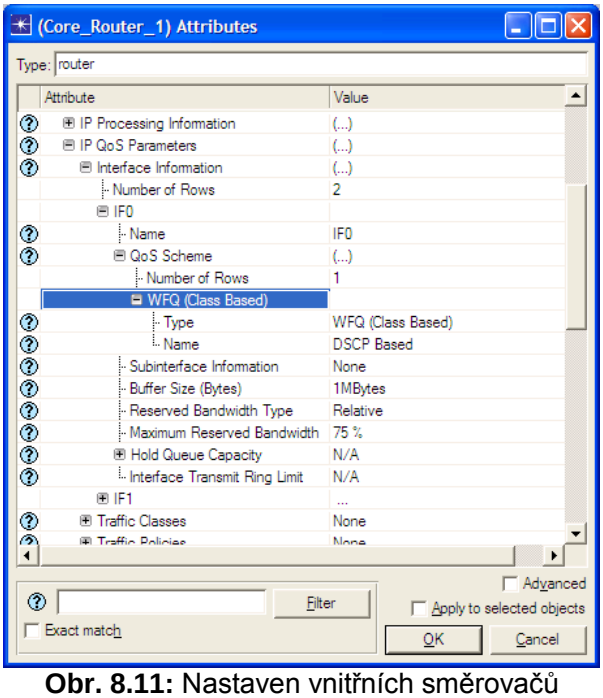
Jednotlivým rozhraním na směrovačích byly přiřazeny pravidla kvality služeb. Nejdříve bylo potřeba zjistit na jakých rozhraních je směrovač propojen s okolím. To se zjistí tak, že se klikne pravým tlačítkem na linku, která jde od směrovače, a vybere se **Edit Ports**. Nastavení rozhraní bylo provedeno v záložce **IP – IP QoS Parameters – Interface Information – Edit – Rows**, počet rozhraní, tedy 2. Pro každé rozhraní byla nastavena kvalita služeb v položce **QoS Scheme – Edit – Rows**. V poli **Type** byl vybrán typ profilu QoS - WFQ (Class Based) a v poli **Name** byla zvolena hodnota DSCP Based. Dále bylo potřeba rozhraní přiřadit politiku provozu. V poli **Type** byl nastaven typ politiky – Inbound Traffic Policy a v položce **Name** byla zvolena již nadefinovaná politika provozu. Na obrázku 8.10 je ukázáno nastavení rozhraní směrovače edge_router_1.



Obr. 8.10: Nastavení rozhraní na okrajovém směrovači

Nastavení vnitřních směrovačů

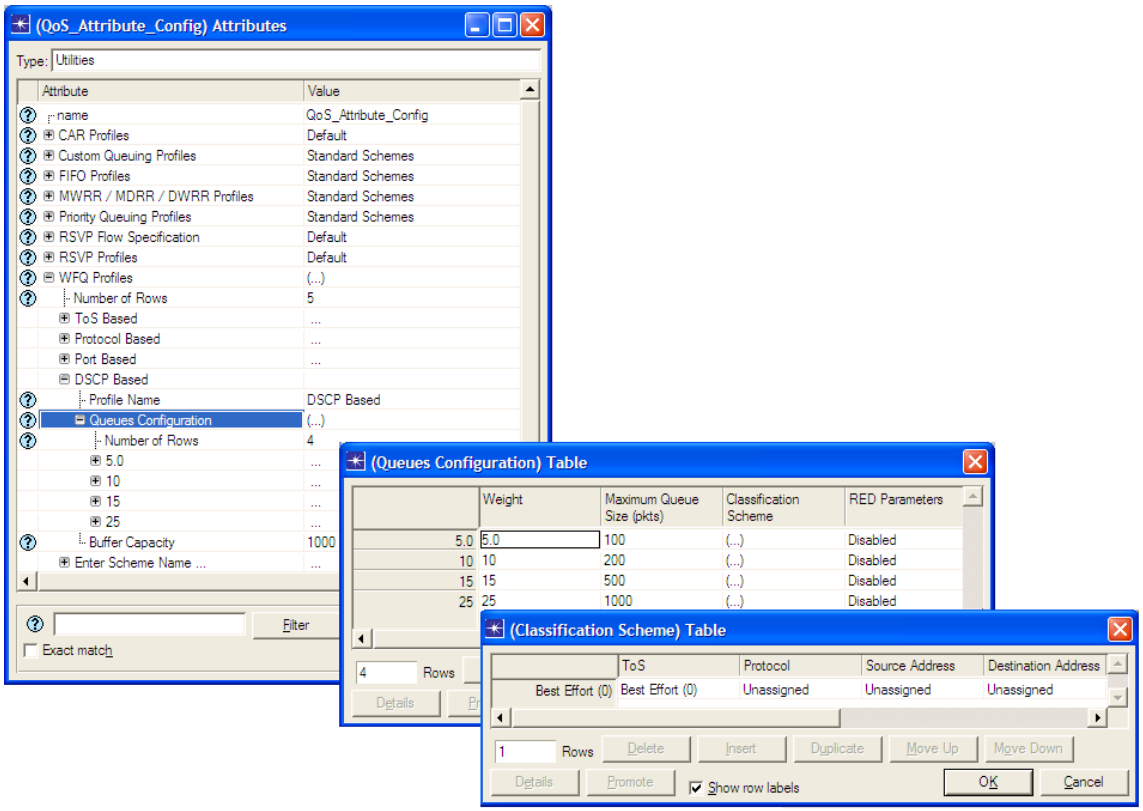
Na vnitřních směrovačích bylo pouze nastaveno řazení paketů do front na jednotlivých rozhraních (viz obr. 8.11). Rozhraní bylo nastaveno stejně jako na okrajových směrovačích. V položce **Edit Attributes** na vnitřním směrovači bylo vybráno **IP – IP QoS Parameters – Interface Information – Edit – Rows**, hodnota 2. V záložce **QoS Scheme – Edit – Rows**, hodnota 1, byly nastaveny hodnoty položek **Type** – WFQ (Class Based) a **Name** – DSCP Based.



Obr. 8.11: Nastaven vnitřních směrovačů

Objekt QoS Atribute Config

Pro zajištění kvality služeb bylo nutné nastavit objekt QoS Atribute Config. V objektu je možné nastavit kvalitu služeb podle několika možností. V našem případě byla kvalita služeb definována v záložce **WFQ Profiles – DSCP Based – Queues Configuration**, ke které se dostane přes položku **Edit Attributes**. Byla vybrána položka **Edit** a nastavena hodnota **Rows**, počet datových toků. Dále byly každému datovému toku nastaveny hodnoty: **Weight** – váha fronty (viz tab. 3), **Maximum Queue Size** – max. počet paketů ve frontě a **Classification Scheme** – způsob jakým bude paket přiřazen k určitému datovému toku. K přiřazení k datovému toku bylo použito pole **ToS** v položce **Classification Scheme**. Hodnota ToS byla vybrána podle tabulky 3. Nastavení objektu QoS Atribute Config pro FTP aplikaci je zobrazeno na obrázku 8.12.



Obr. 8.12: Nastavení objektu QoS Atribute Config

8.3.2 Scénář QoS s tříděním paketů podle IP adresy

V nově vytvořeném scénáři bylo nastaveno třídění paketů podle IP adres cílových klientů. Nastavení bylo provedeno podle Diplomových prácí kolegů Bartoše [1] a Šibíka [10]. Jednotlivým klientům byly přiřazeny IP adresy (viz tab. 4). Pouze dvěma klientům, Client_VoIP1_Database a Client_VoIP3_HTTP, nemuseli být IP adresy přiřazeny, protože jejich provoz neprochází přes žádný okrajový směrovač.

Dále bylo potřeba vyřešit, k jaké aplikaci bude paket po roztřídění přiřazen. Protože paket může být přiřazen pouze k jedné aplikaci, bylo potřeba aplikace jednotlivých klientů shrnout do jedné aplikace a ostatní zanedbat. Z tohoto důvodu dojde ke zkreslení kvality služeb. Kdyby se aplikace jednotlivých klientů neshrnuly do jedné aplikace, nastal by problém při roztřídění. Okrajový směrovač by na základě zdrojové IP adresy nevěděl k jaké aplikaci daný paket přiřadit. Aplikace, do kterých byly aplikace klienta shrnuty (viz tab. 4), byly zvoleny libovolně. Jediným požadavkem byla podpora aplikace ze strany klienta.

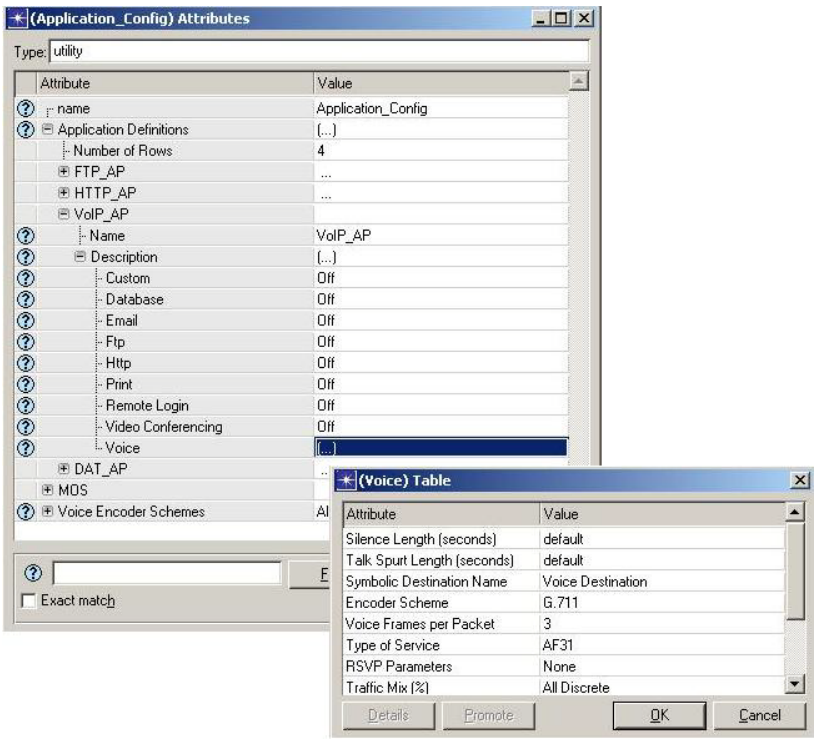
Tab. 4: Přiřazení IP adres a aplikací jednotlivým klientům

Klient	IP adresa	Aplikace
Client_Database_FTP	147.229.149.101	Databáze
Client_FTP_HTTP	147.229.149.102	FTP
Client_FTP_HTTP_Database	147.229.149.103	HTTP
Client_VoIP2_HTTP	147.229.149.104	VoIP
Client_VoIP4_FTP	147.229.149.105	VoIP

8.3.3 Scénář QoS s tříděním paketů podle DSCP

Dále byl vytvořen scénář, kde je jednotlivým aplikacím přiřazena DSCP značka ještě před vstupem do DS domény. Na okrajových směrovačích probíhá pouze klasifikace paketů.

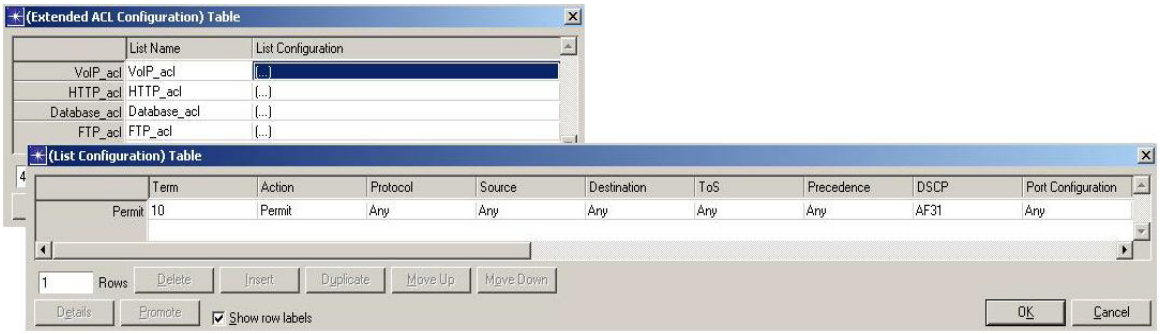
Nejdříve bylo potřeba nastavit jednotlivým aplikacím DSCP značku. To bylo provedeno v objektu **Application Config** v položce **Application Definitions**, kde byla vybrána aplikace a v záložce **Type of Service** se nastavila příslušná hodnota DSCP (viz obr. 8.13).



Obr. 8.13: Přiřazení DSCP značky VoIP aplikaci

Dále byly nastaveny okrajové směrovače. To bylo provedeno podobně jako ve scénáři QoS s tříděním paketů podle protokolu a portu (kapitola 8.3.1). Pouze se upravily hodnoty ACL v záložce **Extended ACL Configuration – List Configuration** na **Action** – Permit a **ToS** – hodnota DSCP (viz obr. 8.14).

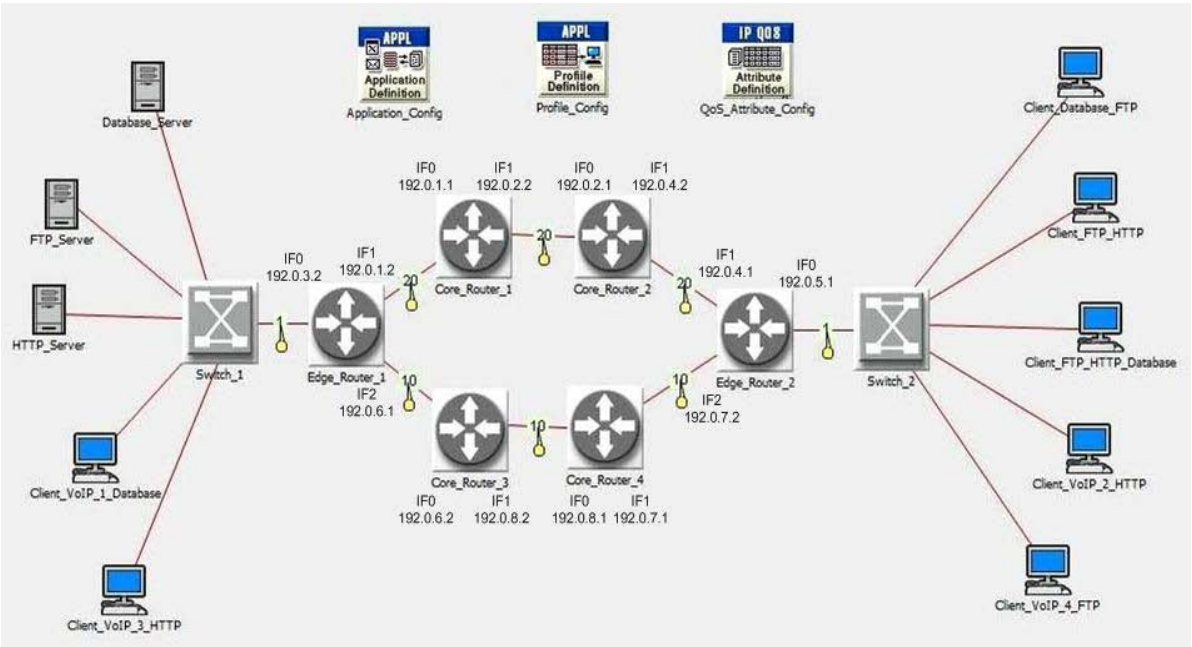
Nastavení vnitřních směrovačů zůstalo stejné jako v kapitole 8.3.1.



Obr. 8.14: Nastavení ACL podle DSCP značky

8.4 Nastavení alternativní trasy a protokolu OSPF

U scénářů s alternativní trasou byla nastavena identifikace paketů podle DSCP značky. Model DiffServ byl rozšířen o alternativní trasu, která obsahuje další dva vnitřní směrovače (obr. 8.15). V předchozích scénářích byl použit směrovací protokol RIP využívající vektor vzdálenosti, který vyhledává cestu k cíli podle počtu skoků. Tento protokol není vhodný pro simulace výpadku hlavní linky, protože nepodporuje alternativní trasy. A proto byl uvnitř tohoto modelu nastaven směrovací protokol OSPF, který alternativní trasy podporuje (viz kapitola 6).



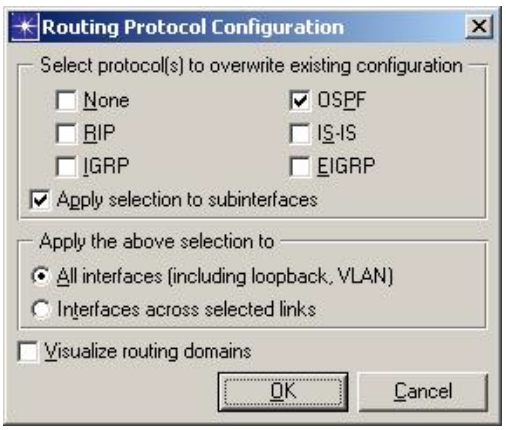
Obr. 8.15: Model s protokolem OSPF – metriky a IP adresy rozhranní

8.4.1 Rozšíření modelu – dvě linky

Po vytvoření druhé linky se dvěma vnitřními směrovači, bylo nutné nastavit na okrajových směrovačích další rozhraní (IF2). To bylo provedeno obdobně jako v kapitole **Nastavení rozhraní na směrovačích**.

8.4.2 Nastavení protokolu OSPF

V rozšířeném modelu byl nastaven směrovací protokol OSPF přes menu **Protocols – IP – Routing – Configure Routing Protocols…** v okně **Routing Protocol Configuration** byla zrušena položka **RIP** a volba **Visualize Routing Domains** a byla vybrána položka **OSPF** (obr. 8.16).

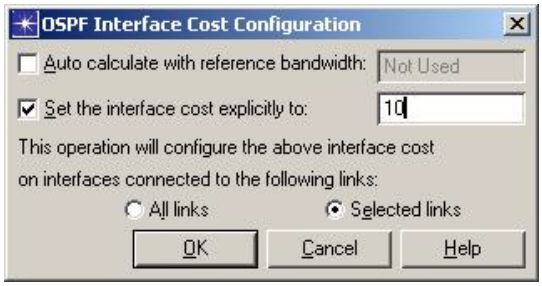


Obr. 8.16: Nastavení směrovacího protokolu

Nastavení metriky

K rozlišení hlavní a náhradní linky nám slouží metriky (váhy) jednotlivých linek. Protokol OSPF udržuje pouze nejlepší cestu. Tedy cestu, která má nejmenší hodnotu metriky. Proto pokud je potřeba nastavit spodní linku jako hlavní, nastaví se jí menší hodnota metriky.

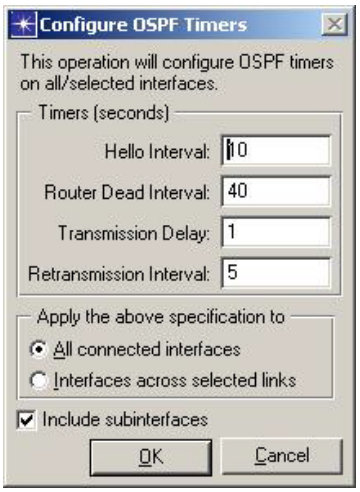
Nastavení jednotlivých metrik bylo vykonáno po dvou krocích. Nejdříve byla nastavena metrika pro spodní linku (hlavní linka) a poté pro horní linku (náhradní linka). Pomocí klávesy **Ctrl** byly označeny všechny spoje se stejnou metrikou a v menu byla vybrána položka **Protocols – OSPF – Configure Interface Cost**. V okně **OSPF Interface Cost Configuration** byly vybrány položky **Selected links** a **Set the interface cost explicity to**, kde byla zadána hodnota metriky (obr. 8.17). V našem případě byla zadána hodnota 10 pro hlavní linku a hodnota 20 pro náhradní linku (viz obr. 8.15).



Obr. 8.17: Nastavení metriky linky

Nastavení časovačů v protokolu OSPF

Výše byly popsány jednotlivé časovače protokolu OSPF i s doporučenými hodnotami (viz kapitola 6). V prostředí OM je možné tyto doporučené hodnoty časovačů změnit pomocí menu **Protocols – OSPF – Configure Interface Timers…**. V našem případě byly tyto hodnoty zachovány (viz obr. 8.18).



Obr. 8.18: Nastavení OSPF časovačů

Směrovací tabulky

Pro zobrazení všech směrovacích tabulek bylo nutné zvolit v menu položku **Protocols – IP – Routing – Export Routing Tables…** a v okně **Export Routing Tables** zaškrtnout **All nodes**.

Aby byly směrovací tabulky kompletní, bylo potřeba nastavit IP adresy jednotlivým rozhraní směrovačů (viz obr. 8.15). IP adresy mohly být nastaveny ručně na každém směrovači zvlášť, nebo pomocí automatického přiřazení položkou **Protocols – IP – Addressing – Auto-Assign IPv4 Addresses…**.

Ukázka směrovací tabulky v prostředí OM směrovače Core_Router_1 je zobrazena na obrázku 8.19.

Destination	Source Protocol	Route Preference	Metric	Next Hop Address	Next Hop Node	Outgoing Interface	Outgoing LSP	Insertion Time (secs)
192.0.1.0/24	Direct	0	0	192.0.1.1	Logical Network.C...	IF0	N/A	0.000
192.0.2.0/24	Direct	0	0	192.0.2.2	Logical Network.C...	IF1	N/A	0.000
192.0.3.0/24	OSPF 1	110	21	192.0.1.2	Logical Network.E...	IF0	N/A	61.982
192.0.4.0/24	OSPF 1	110	40	192.0.2.1	Logical Network.C...	IF1	N/A	61.982
192.0.5.0/24	OSPF 1	110	41	192.0.2.1	Logical Network.C...	IF1	N/A	61.982
192.0.6.0/24	OSPF 1	110	30	192.0.1.2	Logical Network.E...	IF0	N/A	61.982
192.0.7.0/24	OSPF 1	110	50	192.0.2.1	Logical Network.C...	IF1	N/A	61.982
	OSPF 1	110	50	192.0.1.2	Logical Network.E...	IF0	N/A	61.982
192.0.8.0/24	OSPF 1	110	40	192.0.1.2	Logical Network.E...	IF0	N/A	61.982
Gateway of last re...	not set							

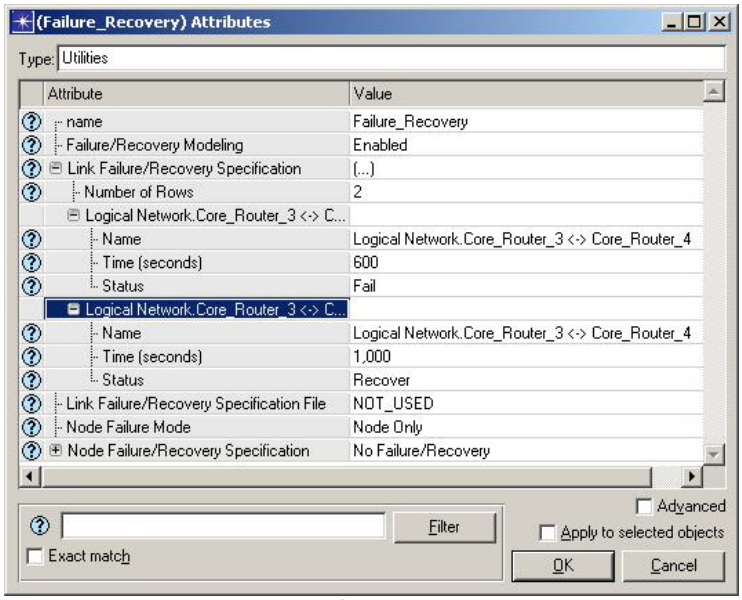
Obr. 8.19: Směrovací tabulka směrovače Core_Router_1

8.4.3 Nastavení výpadku linky

Objekt Failure-Recovery (viz obr. 8.20) slouží k výpadku a obnovení linky. Tento objekt se nachází v sadě objektů **Shared Object Palttes – Utilities**.

Obr. 8.20: Objekt Failure-Recovery

Objekt byl vložen na plochu a pojmenován. Pro nastavení parametrů bylo na objekt kliknuto pravým tlačítkem a byla vybrána položka **Edit Attribures – Link Failure/Recovery Specification** – **Edit** – **Rows**, počet výpadků/obnovení. V záložce **Name** byla vybrána linka, na které dojde k výpadku/obnovení, a v záložce **Time** byl nastaven čas, po kterém dojde k výpadku/obnovení. V položce **Status** byla vybrána možnost Fail = výpadek nebo Recover = obnovení. Nastavení objetu Failure-Recovery je uvedeno na obrázku 8.21.



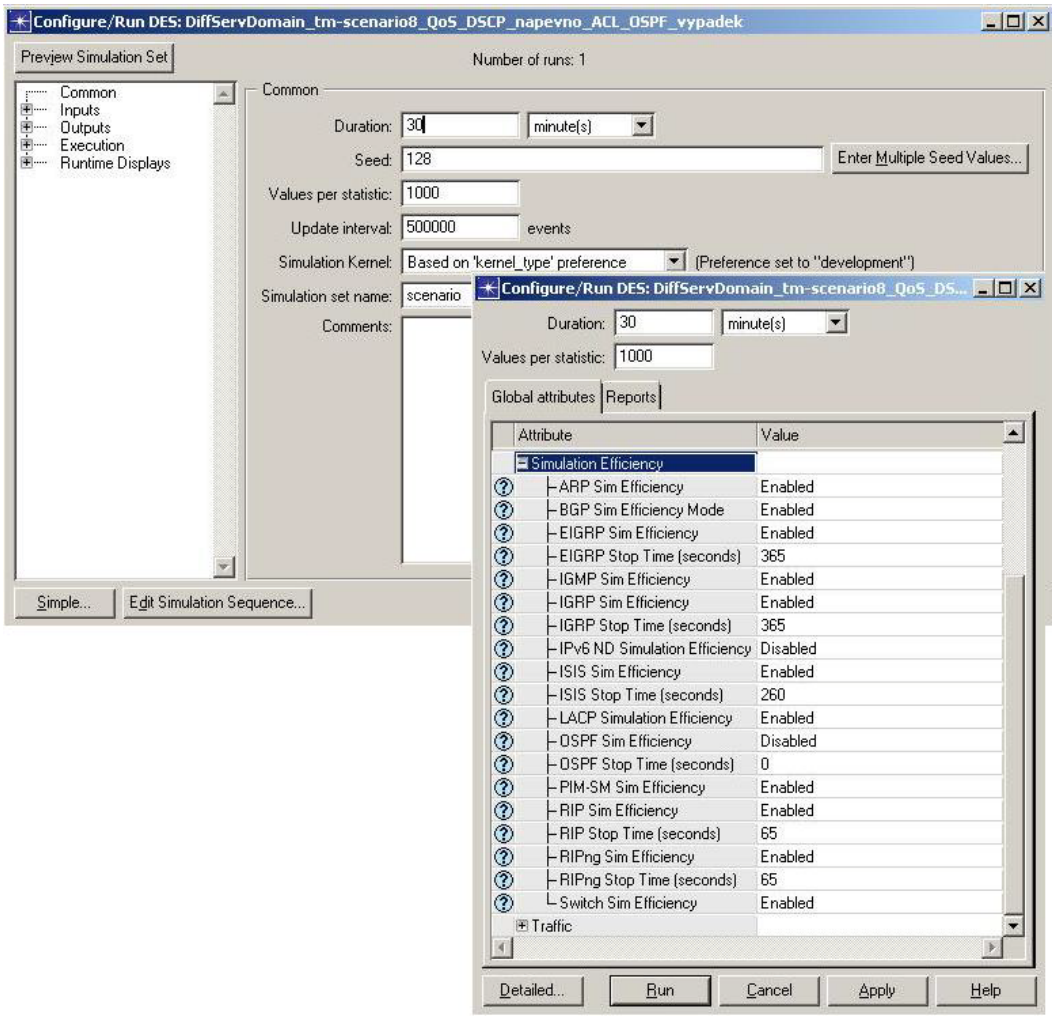
Obr. 8.21: Nastavení objektu Failure-Recovery

8.5 Simulace v prostředí OM

Po nastavení všech směrovačů, klientů, serverů a objektů podle předešlých postupů, byly nastaveny hodnoty, které se budou simulovat. Toto nastavení se provede kliknutím pravým tlačítkem na pracovní plochu prostředí OM a výběrem položky **Choose Individual DES Statistics**.

Samotná simulace byla otevřena z hlavního menu přes položku **DES – Configure/Run Discrete Event Simulation…** nebo přes tlačítko pracovní plochy číslo 14 (viz tab. 2). V otevřeném okně simulace bylo možné nastavit hodnoty položek: **Duration** – doba simulace, **Seed** – počáteční hodnota generátoru náhodného čísla, **Values per statistic** – počet hodnot, který slouží pro vykreslení statistiky, a **Update Interval** – interval, jak často se bude měnit křivka počtu událostí. Dále OM umožňuje vypnutí směrovacího protokolu po určité době. Toto nastavení se nachází pod tlačítkem **Simple…** v záložce **Simulation Efficiency – OSPF Sim Efficiency** a **OSPF Stop Times**. Z důvodu výpadku a nahození linky bylo potřeba, aby směrovací protokol byl aktivní až do konce simulace. A proto byla možnost vypnutí směrovacího protokolu zrušena výběrem položky **Disabled** (viz obr. 8.22).

Simulace byla spuštěna tlačítkem **Run**.



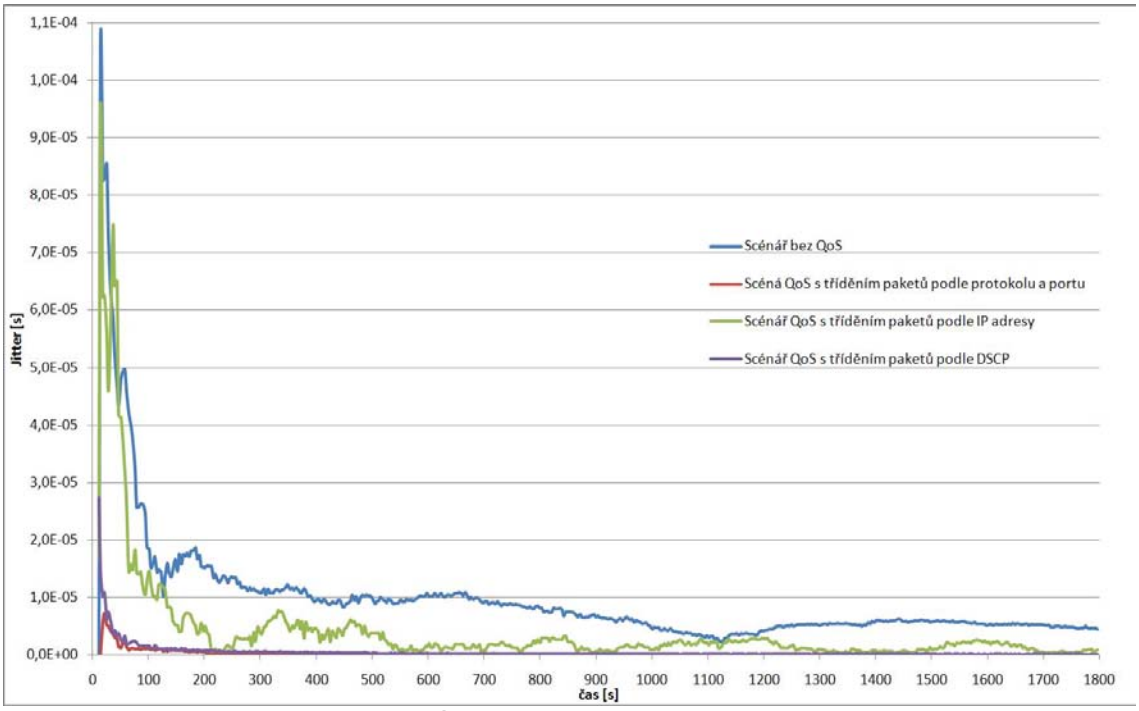
Obr. 8.22: Nastavení simulace

8.5.1 Výsledky simulace v prostředí OM

Po skončení simulace byly zobrazeny výsledky simulace kliknutím na pracovní plochu pravým tlačítkem a zvolením položky **View Results**.

Výsledky simulace – metody třídění paketů

V prvním grafu je zobrazena závislost jitteru (kolísání zpoždění) na čase u VoIP aplikace (viz obr. 8.23). A ve duhém grafu je uvedena závislost zpoždění paketů mezi koncovými body na čase u VoIP aplikace (obr. 8.24). Na těchto grafech je vidět, jak velký vliv má nastavená kvalita služeb na provoz VoIP aplikace, která má nejvyšší prioritu (AF31). Hodnoty jitteru i zpoždění paketů klesly téměř na nulu jak u scénáře QoS s tříděním paketů podle protokolu a portu, tak i u scénáře QoS s tříděním paketů podle přiřazené DSCP značky. U scénáře QoS s tříděním paketů podle IP adresy dochází ke zkreslení z důvodů sloučení několika aplikací do jedné třídy a proto zde není dosažena tak dobrá kvalita služeb jako u předchozích scénářů. U scénáře bez kvality služeb jsou tyto hodnoty největší, protože v tomto scénáři je se všemi aplikacemi zacházeno stejně (BE) a není zde žádná aplikace upřednostňována.



Obr. 8.23: Graf závislosti jitteru na čase u VoIP aplikace

Obr. 8.24: Graf závislosti zpoždění paketů mezi koncovými body na čase u VoIP aplikace

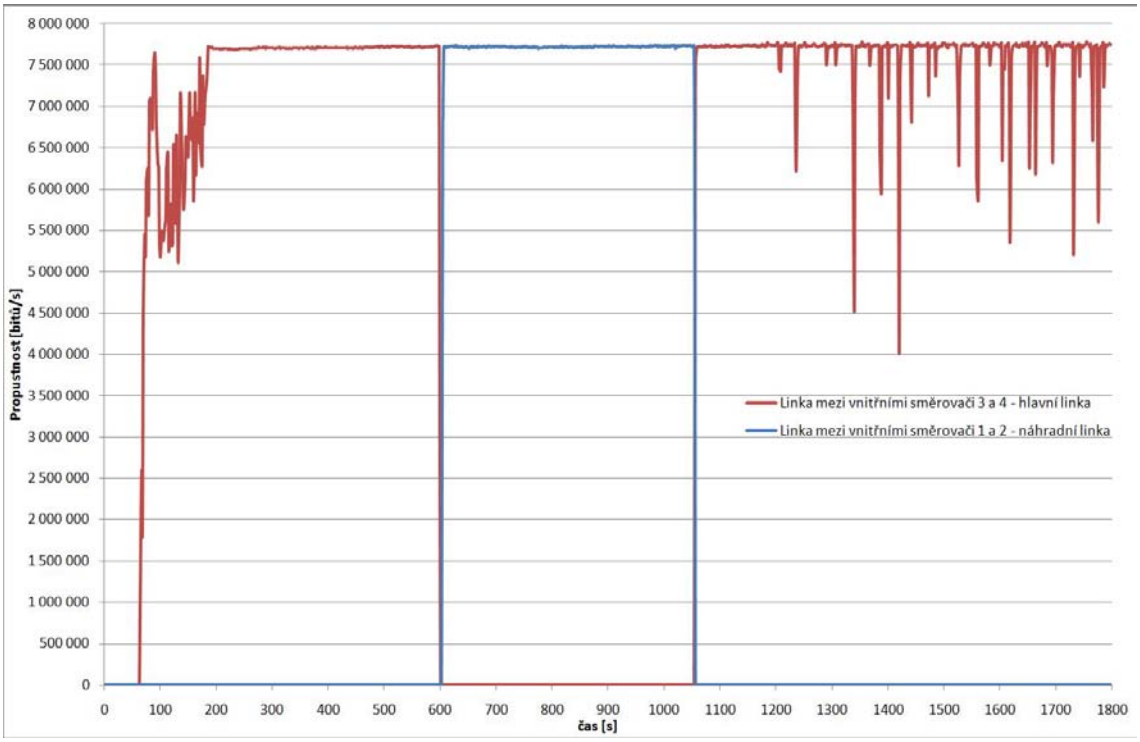
V dalším grafu (viz příloha 1) je zobrazena závislost odezvy HTTP stránek na čase pro jednotlivé scénáře. Zde je patrné kolísání odezvy u scénářů s kvalitou služeb. Ze začátku odezva HTTP aplikace (AF21) při použití kvality služeb stoupne podobně jako u provozu bez nastavené kvality služeb. To je způsobeno využitím šířky pásma pro odesílání aplikací, které mají větší prioritu než HTTP aplikace (tedy VoIP aplikace). Po určité době začne odezva HTTP aplikace klesat a ke konci je již tato hodnota menší než u provozu bez kvality služeb.

Na dalším grafu (viz příloha 2) je závislost odezvy příjmu FTP aplikace na čase. Na tomto grafu je vidět větší přístupová doba FTP aplikace při použití kvality služeb. To je způsobeno zařazením FTP aplikace do třídy BE s nejnižší váhou přenosu. Pouze u scénáře QoS s tříděním paketů podle IP je přístupová doba podobná jako u scénáře bez kvality služeb.

Výsledky simulace – vliv výpadku hlavní linky

V prvním grafu simulace výpadku hlavní linky je zobrazena závislost propustnosti mezi dvěma vnitřními směrovači na čase (viz obr. 8.25). Na tomto grafu je vidět, že průměrná propustnost linky je přibližně 7,7 Mbitů/s. Přenos probíhá pouze po hlavní lince, která prochází přes vnitřní směrovač 3 a 4, dokud nedojde k výpadku této linky. Výpadek hlavní linky byl nastaven na 600 sekund. Jak je vidět z grafu, protokol OSPF zareagoval na tento výpadek okamžitě a veškerý přenos přesměroval na náhradní linku, která prochází přes vnitřní směrovač 1 a 2. K opětovnému nahození hlavní linky dojde po uplynutí 1000 sekund. Ale z grafu je zřejmé, že k opětovnému přesměrování přenosu na hlavní linku došlo až po 1050 sekundách. To je způsobeno

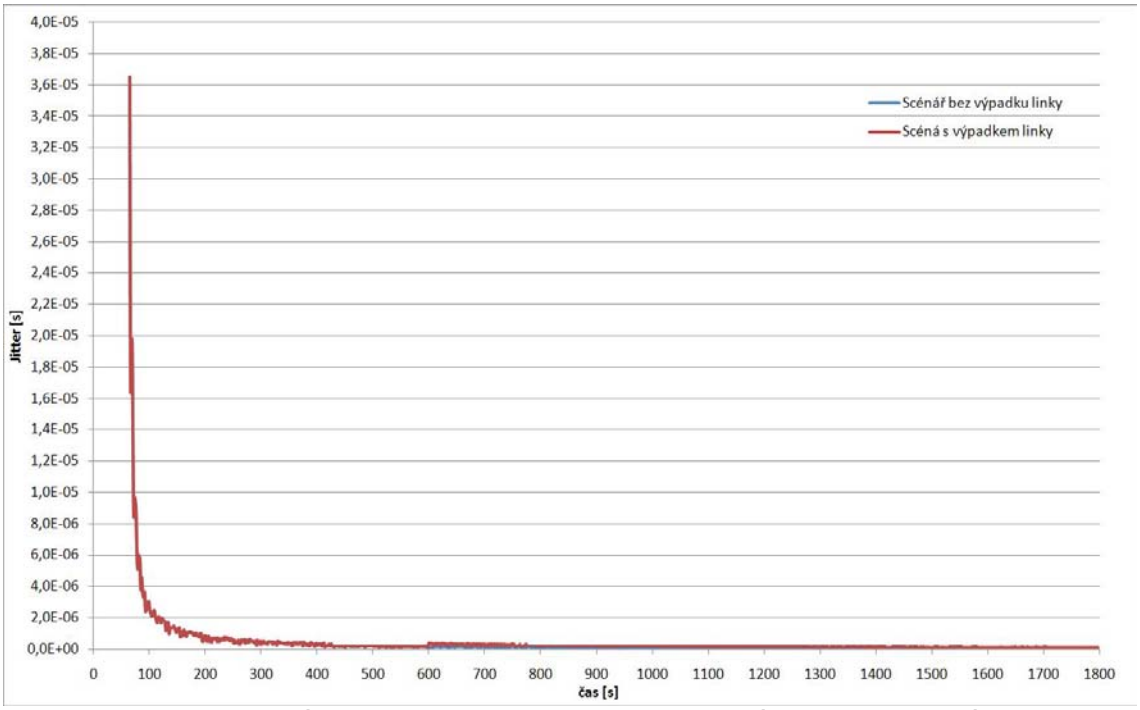
nastavením RouterDeadIntervalu (doba, po kterém je spojení se sousedním směrovačem prohlášeno za nefunkční) na 40 sekund a HelloIntervalu (interval, po kterém jsou vysílány periodicky Hello pakety) na 10 sekund.



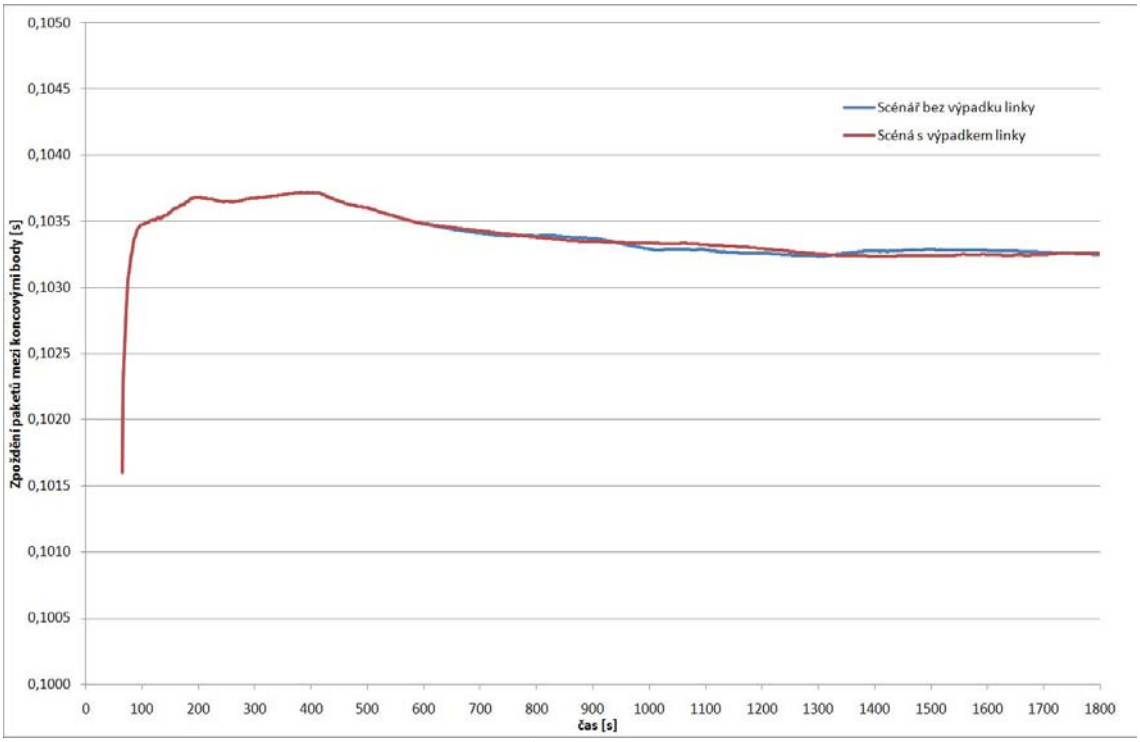
Obr. 8.25: Graf závislosti propustnosti mezi dvěma vnitřními směrovači na čase

Na dalších grafech je uvedena závislost jitteru na čase (obr. 8.26) a závislost zpoždění paketů mezi koncovými body na čase u VoIP aplikace (obr. 8.27). Z těchto grafů je patrné, že u VoIP aplikace nedochází k téměř žádnému zkreslení kvality služeb v důsledku výpadku hlavní linky. Jen v grafu závislosti jitteru na čase dojde v době výpadku (600 sekund) k nepatrnému zakolísání.

V dalším grafu (viz příloha 3) je zobrazena závislost odezvy HTTP stránek na čase pro jednotlivé scénáře. A v posledním grafu (viz příloha 4) je uvedena závislost odezvy příjmu FTP aplikace na čase. V době výpadku a opětovného nahození hlavní linky dojde u všech aplikací k malému zakolísání. V takto malé síti nemá výpadek hlavní linky žádný větší vliv na výslednou kvalitu služeb.



Obr. 8.26: Graf závislosti jitteru na čase u scénáře s výpadkem a bez výpadku



Obr. 8.27: Graf závislosti zpoždění paketů mezi koncovými body na čase u VoIP aplikace a scénáře s výpadkem a bez výpadku

9 Závěr

Úkolem bakalářské práce bylo prostudovat problematiku zajištění kvality služeb v IP sítích. Největší pozornost byla věnována technologii DiffServ a způsobu klasifikace síťového provozu na hraničních směrovačích.

Dále bylo třeba seznámit se se simulačním prostředím OPNET Modeler. Po prostudování studentských prací zaměřených na problematiku zajištění kvality služeb, byl vytvořen v prostředí OM model DiffServ domény. Tento model se skládal ze třech serverů, sedmi klientů, dvou přepínačů, dvou okrajových směrovačů a dvou vnitřních směrovačů. Byly v něm použity tyto aplikace: VoIP, HTTP, databáze a FTP. Z tohoto modelu byly vytvořeny další tři scénáře. V prvním scénáři byla nastavena všem aplikacím třída přenosu Best-Effort a kvalita služeb zde nebyla nastavena. Ve druhém scénáři byla již kvalita služeb nastavena a aplikace byly roztříděny do různých tříd provozu. Identifikace a klasifikace paketů podle protokolu a portu probíhala na okrajových směrovačích. K identifikaci byl použit rozšířený typ ACL filtrující přenos podle protokolu a portu. Ve třetím scénáři byla také nastavena kvalita služeb. Identifikace paketů zde nebyla nastavena podle aplikací, ale podle IP adresy klienta. A ve čtvrtém scénáři byla kvalita služeb nastavena podle DSCP značky, která byla jednotlivým aplikacím přiřazena ještě před vstupem do DS domény. Na okrajových směrovačích probíhala pouze klasifikace paketů pomocí ACL.

Nakonec byl model DiffServ domény rozšířen o alternativní trasu, která obsahovala další dva vnitřní směrovače. Uvnitř tohoto modelu byl nastaven směrovací protokol OSPF. Z rozšířeného modelu byl vytvořen další scénář, kde byl nastaven výpadek hlavní linky.

Všechny scénáře byly odsimulovány. Z výsledných grafů je patrné, že nejlepší kvalita služeb byla dosažena u scénáře číslo 4, kde bylo třídění paketů nastaveno podle DSCP. Podobná kvalita služeb byla dosažena i u druhého scénáře, kde třídění paketů bylo podle protokolu a portu. Pouze u některých aplikací (HTTP, FTP a databáze) došlo k časovému posunu. U scénáře číslo 3 byla kvalita služeb zkreslená dalšími aplikacemi, které klient používá. Z důvodu třídění paketů podle IP adresy musely být tyto aplikace zanedbány. Proto je výhodnější použít u klientů s více aplikacemi najednou třídění paketů podle jednotlivých aplikací.

Ze simulací výpadku hlavní linky je zřejmé, že výpadek hlavní linky a použití náhradní linky má v takto malé síti, která je složena ze dvou okrajových směrovačů a čtyř vnitřních směrovačů, na výslednou kvalitu služeb jen nepatrný vliv.

Seznam zkratek

ACL - Access Control List
AF - Assured Forwarding
ATM - Asynchronous Transfer Mode
BA - Behavior Aggregate
BE - Best-Effort
CBWFQ - Class-Based Weighted Fair Queuing
CSCP - Class Selector Code Points
CU - Currently Unused
DS - Differentiated Services
DSCP - Differentiated Service CodePoint
EF - Expedited Forwarding
FIFO - First In, First Out
FSM - Finite State Machines
FTP - File Transfer Protocol
HTML - HyperText Markup Language
HTTP - Hypertext Transfer Protocol
IETF - Internet Engineering Task Force
IP - Internet Protocol
IPv4 - Internet Protocol version 4
ISO/OSI - International Organization for Standardization/Open Systems Interconnection
LSA - Link-State Advertisement
MF - Multi-Field
OM - OPNET Modeler
OSPF - Open Shortest Path First
PHB - Per-Hop Behaviors
QoS - Quality of Service
RIP - Routing Information Protocol
RSVP - Resource Reservation Protocol
SLA - Service Level Agreement
SPF - Shortest Path First
TCA - Traffic Conditioning Agreement
TCP - Transmission Control Protocol
ToS - Type of Service
UDP - User Datagram Protocol
VoIP - Voice over Internet Protocol
WFQ - Weighted Fair Queuing
WWW - World Wide Web
XML - eXtensible Markup Language

Použitá literatura

[1] BARTOŠ, P.: Detailní analýza způsobu zacházení Assured Forwarding v simulačním prostředí OPNET Modeler, Diplomová práce. Brno: VUT Fakulta elektrotechniky a komunikačních technologií, 2007, 68 s.

[2] BEZCHLEBA, J.: Modelování řízení provozu v hraničních směrovačích DiffServ domény, Bakalářská práce, Brno: VUT Fakulta elektrotechniky a komunikačních technologií, 2006, 45 s.

[3] FLANNAGAN, Michal E., DURAND, B., SOMMERVILLE, J., BUCHMANN, M., FULLER, R.: Administering CISCO QoS in IP Networks, USA, Syngress, 2001, 535 s.

[4] GRYGÁREK, J.: Směrovací protokol OSPF [online], 2002, [cit. 2009-05-01]. Dostupné z WWW:
http://www.cs.vsb.cz/grygarek/SPS/lect/OSPF/ospf.html

[5] KACÁLEK, J.: Modely pro zajištění kvality služeb v IP sítích [online], 2006, [cit. 2008-12-06]. Dostupné z WWW:
http://amarok.cesketelekomunikace.cz/xkacal00/index.php?action=intro

[6] MOLNÁR, K.: Moderní síťové technologie – Laboratorní cvičení, Brno: VUT Fakulta elektrotechniky a komunikačních technologií, 2008, 90 s.

[7] PARK, Kun I.: QoS in Packet Network, USA-Boston, Springer, 2004, 243 s.

[8] SOUMAR, M.: BHWS – Laboratorní cvičení číslo 5: Konfigurace směrovacího protokolu OSPF v prostředí Network Visualizer 4 [online], 2005, 6 s. Dostupné z WWW:
http://www.utko.feec.vutbr.cz/~molnar/bhws/5-OSPF_new.pdf

[9] SOUMAR, M.: BHWS – Laboratorní cvičení číslo 9: Řízení přístupu pomocí ACL v Cisco IOS [online], 2004, 5 s. Dostupné z WWW:
http://www.utko.feec.vutbr.cz/~molnar/bhws/9-ACL.pdf

[10] ŠIBÍK, Š.: Implementace parametrických modelů na okamžitých vlastnostech síťového provozu v simulačním prostředí OPNET Modeler, Diplomová práce, Brno: VUT Fakulta elektrotechniky a komunikačních technologií, 2008, 57 s.

[11] TIRINDA, V.: Klasifikace a značkování v hraničních směrovačích DiffServ domény, Bakalářská práce, Brno: VUT Fakulta elektrotechniky a komunikačních technologií, 2006, 46 s.

[12] UBIK, S.: Technická zpráva TEN-155 CZ číslo 6/2000: QoS a Diffserv – Úvod do problematiky [online], 2000, 12 s. Dostupné z WWW:
http://www.cesnet.cz/doc/techzpravy/2000-6/diffserv.pdf

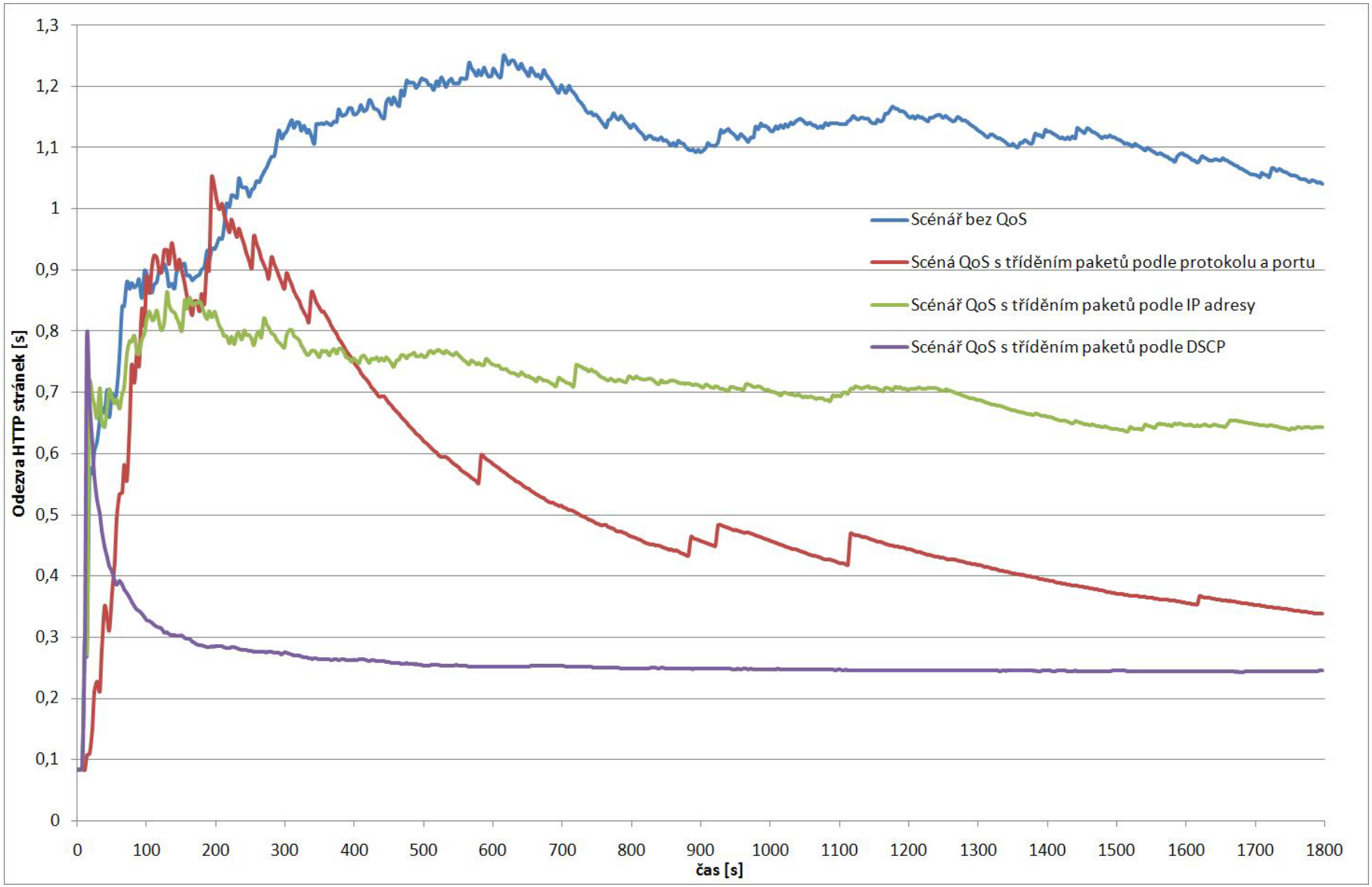
[13] VELTE, Toby J., VELTE, Anthony T.: A Beginner´s Guide, USA, McGraw-Hill Companies, 2001, 759 s.

[14] ZEMAN, O.: Modelování chování páteřních směrovačů DiffServ domény, Bakalářská práce, Brno: VUT Fakulta elektrotechniky a komunikačních technologií, 2006, 70 s.

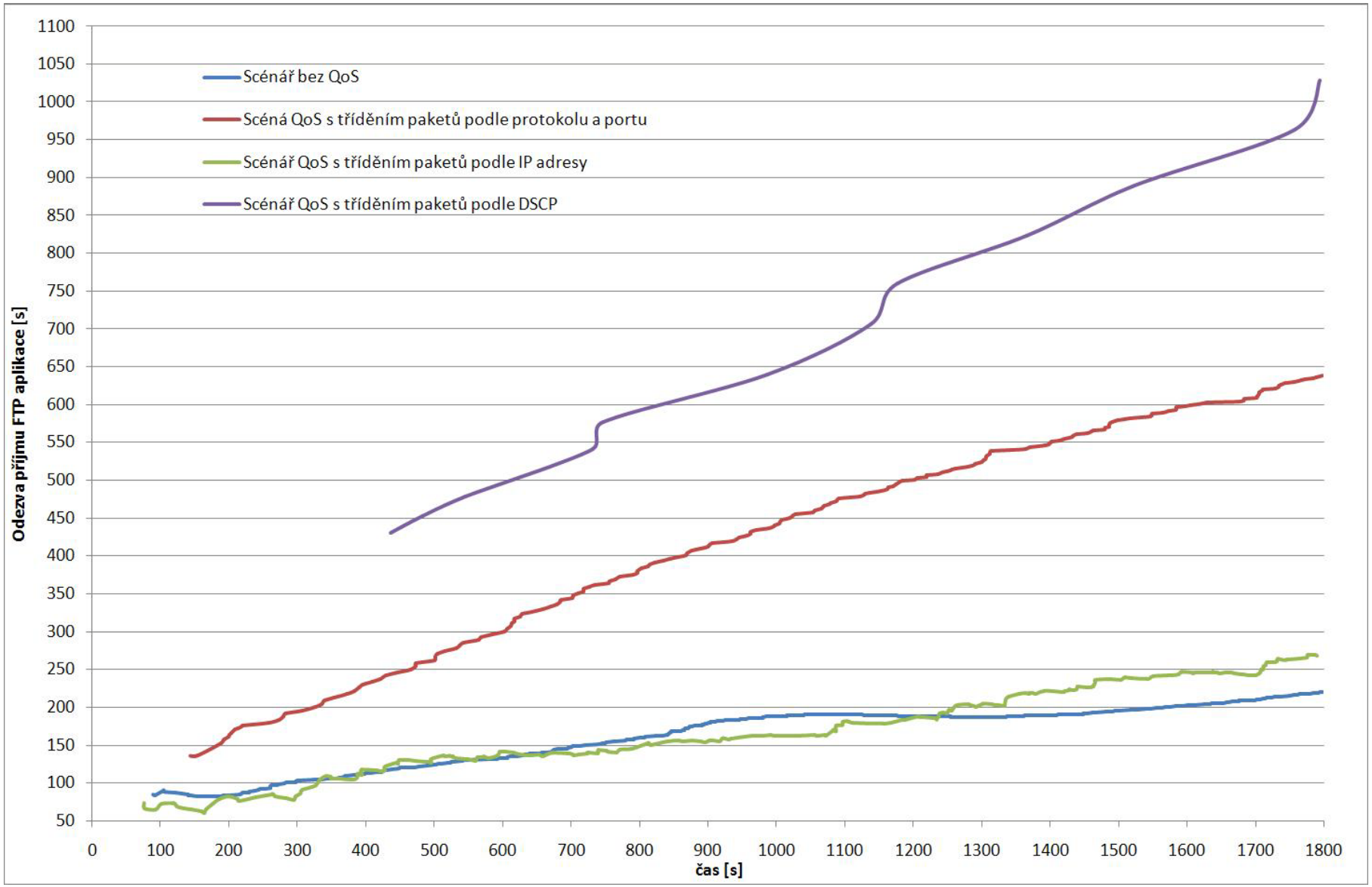
Přílohy

Seznam příloh

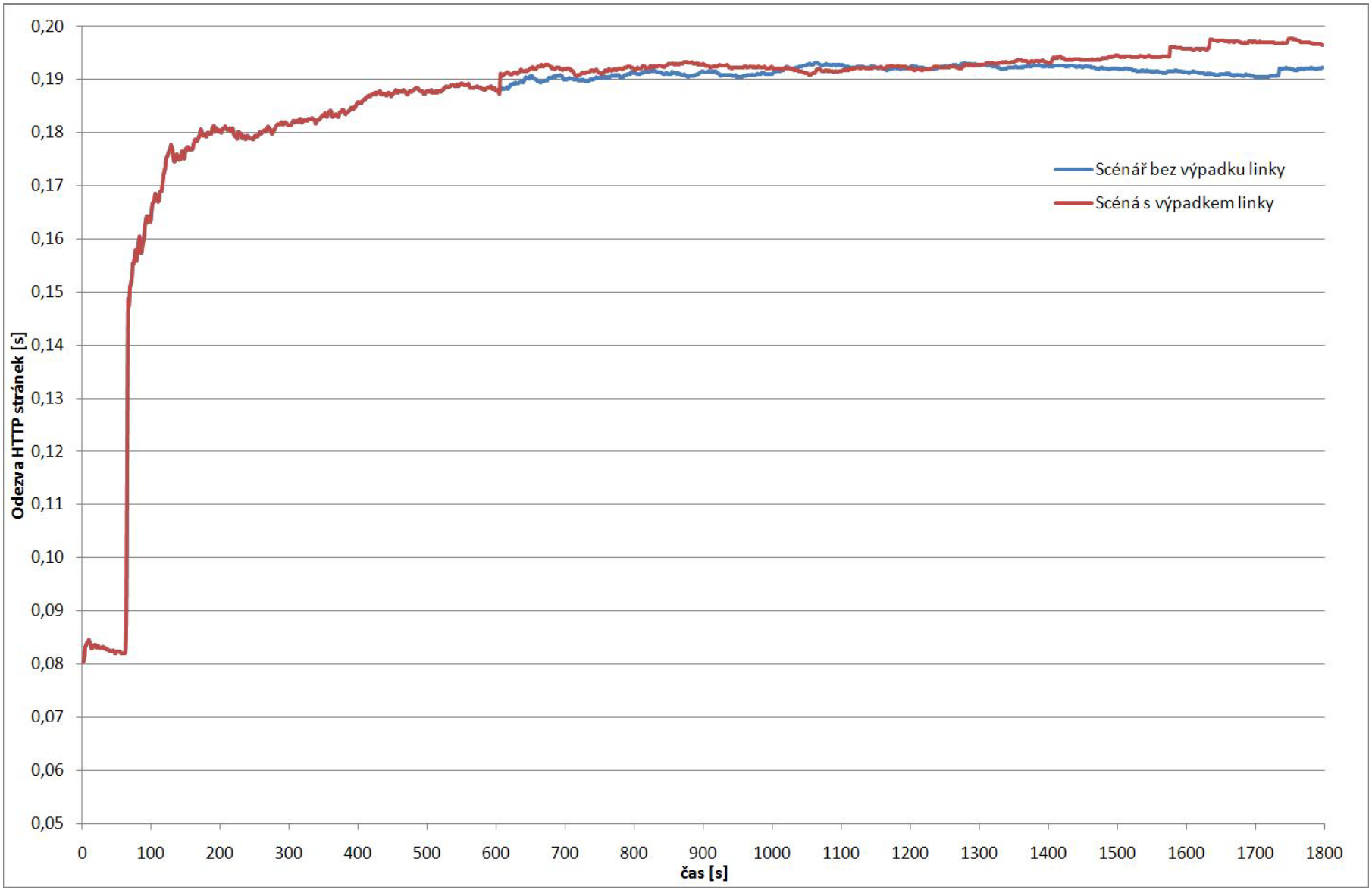
Příloha 1: Graf závislosti odezvy HTTP stránek na čase 47
Příloha 2: Graf závislosti odezvy příjmu FTP aplikace na čase.....48
Příloha 3: Graf závislosti odezvy HTTP stránek na čase u scénáře s výpadkem a bez výpadku.....49
Příloha 4: Graf závislosti odezvy příjmu FTP aplikace na čase u scénáře s výpadkem a bez výpadku 50



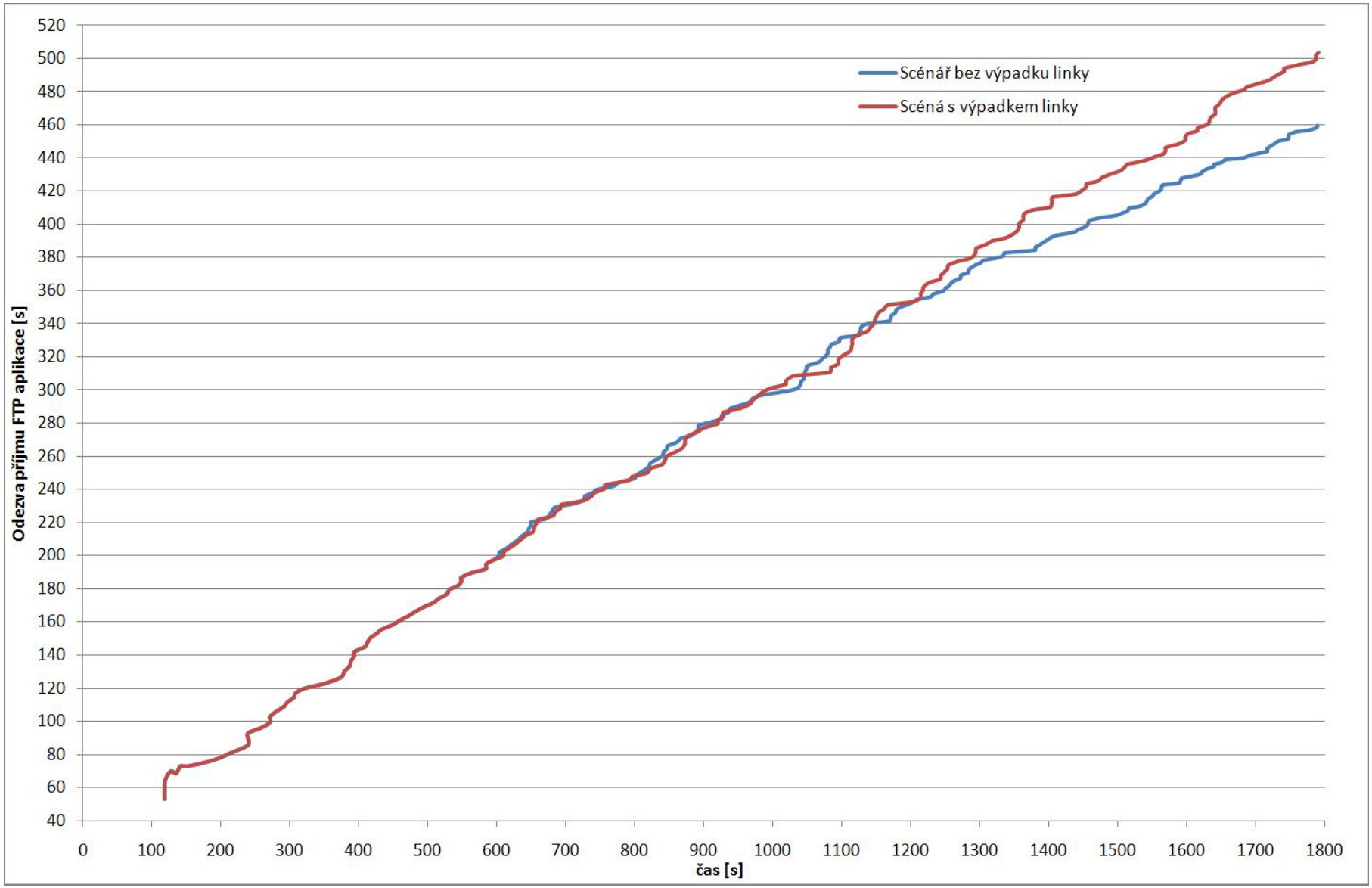
Příloha 1: Graf závislosti odezvy HTTP stránek na čase



Příloha 2: Graf závislosti odezvy příjmu FTP aplikace na čase



Příloha 3: Graf závislosti odezvy HTTP stránek na čase u scénáře s výpadkem a bez výpadku



Příloha 4: Graf závislosti odezvy příjmu FTP aplikace na čase u scénáře s výpadkem a bez výpadku