



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

VÝKONNOST IP PROVOZU

IP TRAFFIC PERFORMANCE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

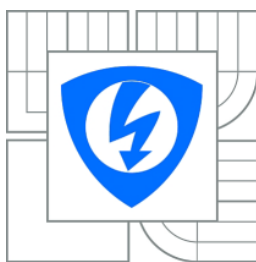
AUTOR PRÁCE
AUTHOR

Bc. FRANTIŠEK BEDNÁŘ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ MÁCHA

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. František Bednář
Ročník: 2

ID: 98666
Akademický rok: 2011/2012

NÁZEV TÉMATU:

Výkonnost IP provozu

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je seznámit se a prostudovat provoz v IP (Internet Protocol) sítích se zaměřením na strukturu protokolu a aplikací. Navrhnout nový model síťové struktury a odsimulovat v Opnet Modeleru. Dále využít dostupné simulátory (NS-3) pro modifikaci, případně tvorbu nových návrhů, které by vylepšily směrování v síti. Zaměření by se týkalo především OSPF protokolu.

DOPORUČENÁ LITERATURA:

[1] GRIMM, CH., SCHLÜCHTERMANN, G. IP-Traffic Theory and Performance, Springer, 1 edition (September 1, 2008), 488 pages, ISBN-13: 978-3540706038.

[2] HUITEMA, CH. Routing in the Internet (2nd Edition). 385 p. 1999. ISBN-13: 978-0130226471.

Termín zadání: 6.2.2012

Termín odevzdání: 24.5.2012

Vedoucí práce: Ing. Tomáš Mácha
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 následku 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 c.40/2009 Sb.

ANOTACE

Diplomová práce „Výkonnost IP provozu“ je zaměřena na testování provozu v IP sítích. V teoretické části je vysvětlena problematika směrování v autonomních systémech. V práci je podrobněji popsán protokol OSPF. Práce se dále zabývá chováním sítě při výpadku. Jsou zde vysvětleny mechanismy, které slouží k detekci výpadku linky a následnému přesměrování provozu.

V další části jsou popsány mechanismy pro zajištění kvality služeb QoS. V diplomové práci jsou vysvětleny mechanismy InterServ a DiffServ. Velká část je věnována mechanismu DiffServ, který zajišťuje rozlišení datových toků a klasifikování jednotlivých paketů do tříd. Směrovače pak mohou jednotlivé datové toky zpracovávat s různými prioritami.

Praktická část obsahuje návrh experimentální sítě a vytvoření simulace v programu Opnet Modeler. Použitím několika scénářů jsou srovnány vlastnosti směrovacích protokolů a vlivu QoS na přenosové vlastnosti sítě. Součástí praktické části je i návrh vylepšení směrování protokolu OSPF zavedením nové metriky a implementace nové metriky do směrovacího softwaru Quagga.

Klíčová slova: QoS, OSPF, směrování, metrika, Opnet Modeler, Quagga

ABSTRACT

The master thesis „IP traffic performance“ is focused on traffic testing in IP networks. Theoretical section explains routing issue in an autonomous system. This work contains a detailed description of OSPF protocol. This work also deals with behavior of a link failure. There are described mechanisms that are used to link failure detection and subsequent traffic rerouting.

The next section describes mechanisms to ensure quality of service. In master thesis are explained InterServ and DiffServ mechanisms. A large part is devoted to DiffServ mechanism that ensures distinction of data flows and classification packets into different classes. The routers than can process the individual data streams with different priorities.

The practical section includes the design of experimental network and creation of simulation in Opnet Modeler. By using several scenarios are compared the characteristics of routing protocols and impact of QoS on the transmission characteristics of the network. Part of practical section is the improvement of OSPF protocol by adding a new metric and implementing a new metric in software suite Quagga.

Keywords: QoS, OSPF, routing, metric, Opnet Modeler, Quagga

BEDNÁŘ, F. *Výkonnost IP provozu: diplomová práce*. Brno: FEKT VUT v Brně, 2012.
71 stran, 1 příloha. Vedoucí práce Ing. T. Mácha.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Výkonnost IP provozu“ 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 uvedeného diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu 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 práce za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne

.....
podpis autora

Výzkum popsaný 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.

OBSAH

Obsah	9
Seznam obrázků	10
Seznam tabulek	11
Úvod.....	12
1. Směrování v IP sítích	13
1.1. Směrovací tabulka	13
1.2. Směrovací protokoly	14
1.3. Směrovací protokol RIP (Router Information Protocol)	15
1.4. Směrovací protokol OSPF (Open Shortest Path Protocol).....	16
2. Spolehlivost provozu v IP sítích	27
2.1. Detekce výpadku u OSPF.....	28
3. QoS - Kvalita služeb	30
3.1. Integrované služby - IntServ	31
3.2. Diferencované služby Diffserv	33
4. Opnet Modeler	39
4.1. Základní prvky prostředí Opnet Modeler	39
4.2. Editory	39
5. Simulace v prostředí Opnet Modeler	40
5.1. Nastavení aplikací	41
5.2. Nastavení profilu aplikací	42
5.3. Nastavení podpory aplikací a profilů ve scénáři	43
5.4. Nastavení scénáře s protokolem RIP a Diffserv	44
5.5. Nastavení scénáře s protokolem OSPF	46
5.6. Nastavení charakteristik a parametrů simulace	47
5.7. Výsledky simulace	48
6. Adaptivní změna metriky u protokolu OSPF	54
6.1. Exponenciálně vážené klouzavé průměry EWMA	55
6.2. Implementace nové metriky do protokolu OSPF	56
7. Závěr	63
8. Citovaná literatura.....	65
Seznam zkratk	68
Seznam příloh.....	70

SEZNAM OBRÁZKŮ

- Obr. 1.1: Oblasti u OSPF
- Obr. 1.2: Výpočet SPF
- Obr. 1.3: Struktura OSPF paketu
- Obr. 1.4: Struktura záhlaví LSA
- Obr. 1.5: Směrovací LSA
- Obr. 1.6: Síťová LSA zpráva
- Obr. 1.7: Sumární LSA zpráva
- Obr. 2.1: Struktura redundantní topologie
- Obr. 3.1: Architektura IntServ
- Obr. 3.2: Komunikace protokolu RSVP
- Obr. 3.3: Struktura pole DS
- Obr. 3.4: Blokové schéma zpracování provozu
- Obr. 3.5: a) Graf neupraveného datového toku
- Obr. 3.5: b) Graf omezeného datového toku
- Obr. 3.5: c) Graf tvarovaného datového toku
- Obr. 5.1: Topologie simulované sítě
- Obr. 5.2: Nastavení aplikace FTP
- Obr. 5.3: Nastavení parametru Page Properties
- Obr. 5.4: Nastavení aplikace VoIP
- Obr. 5.5: Nastavení profilu pro aplikaci FTP
- Obr. 5.6: Nastavení výpadku linky mezi Router1 - Headquarters
- Obr. 5.7: Graf zpoždění aplikace VoIP
- Obr. 5.8: Graf kolísání zpoždění aplikace VoIP
- Obr. 5.9: Graf odezvy FTP serveru při stahování
- Obr. 5.10: Graf odezvy stránek webového serveru
- Obr. 5.11: Graf doby konvergence v síti
- Obr. 5.12: Graf velikosti datového toku protokolů OSPF a RIP
- Obr. 5.13: Propustnost linky mezi směrovači Router1 a Headquarters_Edge_router
- Obr. 5.14: Propustnost linky mezi směrovači Router3 a Router2
- Obr. 6.1: Ukázková topologie
- Obr. 6.2: Graf EWMA
- Obr. 6.3: Architektura Quagga
- Obr. 6.4: Architektura NS3
- Obr. 6.5: Algoritmus adaptivní metriky

SEZNAM TABULEK

Tab. 1.1: Základní metriky OSPF protokolu

Tab. 1.2: Typy LSA zpráv

Tab. 1.3: Typy linek (rozhraní) u směrovacích LSA

Tab. 2.1: Tabulka hodnot rychlosti detekce výpadku a doby obnovení

Tab. 3.1: Značení tříd u zajištěného přenosu AF

Tab. 5.1: Nastavení parametru

Tab. 5.2: Nastavení podpory aplikací a profilů v jednotlivých podsítích

Tab. 5.3: Čísla portů aplikaci

Tab. 5.4: Čísla portů aplikaci

Tab. 6.1: Schéma nastavení nových metrik

ÚVOD

Diplomová práce je zaměřena na výkonnost provozu v IP sítích. Výkonnost provozu v IP sítích je ovlivněna použitým směrovacím protokolem. V úvodu jsou popsány základy směrování. Směrovací protokoly ovlivňují, jak rychle budou směrovače schopny na základě vytvořených směrovacích tabulek rozhodovat, na jaké rozhraní mají příchozí pakety odeslat. Směrovací protokoly se liší použitými algoritmy, které používají k vytváření směrovací tabulky. K výběru nejlepší cesty od zdroje k cíli používají jiné metriky. Důležitou vlastností je, jak rychle dokáže směrovací protokol zjistit, že příslušná linka nebo síť je nedostupná a následně o této skutečnosti informovat ostatní směrovače. Jakmile si směrovače aktualizují záznamy ve směrovacích tabulkách, mohou provoz směrovat jinou cestou. Diplomová práce je zaměřena protokol OSPF.

Podstatná část je také věnována mechanismu QoS, který také výrazně ovlivňuje výkonnost IP sítí. První IP sítě byly navrhovány tak, aby zajistili co nejrychlejší a nejjednodušší přenos dat. Případnou spolehlivost přenosu měli zajišťovat až koncové stanice. Směrovače nedokázali rozeznat, jakým aplikacím procházející pakety patří a tak se všemi zacházeli se stejnou prioritou. V počátcích Internetu se v sítích nejčastěji vyskytovaly pakety aplikací, které nebyly citlivé na zpoždění, potřebovali pouze, aby byla zajištěna spolehlivost přenosu, kterou zajišťovaly koncové stanice. Příchodem aplikací, které pracují v reálném čase jako IP telefonie, videotelefonie se situace změnila. U těchto aplikací je nutné dodržet určité hodnoty zpoždění a kolísání zpoždění. Pokud budou tyto hodnoty překročeny, dochází k degradaci služby. Proto je v souvislosti s těmito službami je nasazována kvalita služeb QoS. QoS je mechanismus, který umožňuje, aby určité aplikace byly zpracovávány přednostně. To může být zajištěno rezervací síťových prostředků, nebo přidělením vyšší priority při zpracování paketů ve směrovači.

Součástí praktické části diplomové práce je vytvoření experimentální sítě, ve které je provedeno srovnání protokolů OSPF a RIP. Součástí je simulace v programu Opnet Modeler, která obsahuje několik výpadků linky, které umožní pozorovat, jak rychle dokážou protokoly detekovat výpadek a jak rychle je provedeno přesměrování provozu. Je provedeno srovnání chování sítě s nastavenou kvalitou služeb a bez nastavení kvality služeb. Součástí praktické části je návrh vylepšení protokolu OSPF, které by umožnilo vylepšení směrování. Vylepšení OSPF protokolu je provedeno implementací nové metriky, která zajistí, že výběr nejlepší cesty bude proveden nejen na základě ceny linky, ale také na základě vytížení linky.

1. SMĚROVÁNÍ V IP SÍTÍCH

Směrování (routing) je proces, který zajišťuje výběr nejlepší cesty mezi dvěma sítěmi. Směrování ve většině případů zajišťuje zařízení, které se nazývá směrovač (router). Směrování může také provádět L3 přepínač, firewall, server, nebo obyčejný počítač. Směrování je nejčastěji protokol IP (Internet Protocol). Další protokoly, které mohou být směrovány, jsou IPX (operační systém Novell) nebo DDP (součást sady protokolů Appletalk).

Směrování je z hlediska ISO/OSI modelu umístěno na třetí síťové vrstvě. Směrování IP protokolu je prováděno na základě IP adres. Při směrování se pracuje pouze z částí IP adresy, která se nazývá adresa sítě. Adresu sítě si směrovač určí vynásobením masky a cílové adresy v binárním tvaru.

Směrovač rozhoduje o tom, na které rozhraní bude paket odeslán, aby byl doručen do cílové sítě. Tento proces se odehrává na každém směrovači podél celé cesty od zdroje paketu k cíli. Směrovač zajišťuje propojení LAN a WAN sítí.

Směrovač je zařízení, které zajišťuje dvě základní funkce:

- Určení optimální cesty – směrovač zajišťuje výběr nejlepší cesty v sítích, kde existuje více cest do cílové sítě. Tuto funkci zajišťují směrovací algoritmy, které jsou implementovány ve směrovačích. Pro usnadnění směrovacího procesu je vytvářena a udržována směrovací tabulka.
- Předávání paketů – směrovač slouží ke zpracování a předávání paketů. Směrovač příchozí paket analyzuje, zjistí adresu cílové sítě a provede srovnání se záznamy ve směrovací tabulce. Pokud je nalezena shoda, je paket odeslán na rozhraní, které je uvedeno jako další skok ve směrovací tabulce.
- Zapouzdření paketu/ rozbalení rámce – směrovač zajišťuje propojení sítí, které na linkové vrstvě využívají jinou technologii (Ethernet, PPP, Token Ring). Směrovač příchozí rámec rozbalí (odstraní záhlaví a zápatí rámce) a zjistí cílovou IP adresu. Pokud najde shodu ve směrovací tabulce, musí zjistit pomocí ARP protokolu (Address Resolution Address) protokolu fyzickou adresu rozhraní směrovače, na který bude paket odeslán. Z přijatého paketu je opět vytvořen rámec odpovídající technologii linkové vrstvy, která je použita v připojené síti.

1.1. SMĚROVACÍ TABULKA

Je to datová struktura, kterou obsahuje každé zařízení, které provádí směrování paketů. Směrovací tabulka obsahuje seznam známých vzdálených sítí. Každý záznam pouze informuje směrovač, jakým rozhraním má paket odeslat, aby byl doručen do cílové sítě. Neobsahuje popis celé cesty od zdroje k cíli. Každý záznam ve směrovací tabulce obsahuje následující informace:

- Adresa cílové sítě – udává IP adresu sítě, do které má být paket odeslán
- Síťová maska – slouží k porovnávání záznamů ve směrovací tabulce a cílové adresy paketu

- Odchozí rozhraní – označuje síťové rozhraní, na které má být paket odeslán
- Administrativní vzdálenost (AD – administrative distance) – určuje prioritu jednotlivých záznamů podle protokolu, kterým byl záznam vytvořen. Administrativní vzdálenost určuje spolehlivost a kvalitu protokolu. Čím je AD nižší, tím roste priorita záznamu.
- Typ protokolu - udává, jakým protokolem byl záznam vytvořen

U počítačů je směrovací tabulka vytvářena na základě konfigurace TCP/IP protokolu ihned po startu systému. U směrovačů je směrovací tabulka vytvářena v paměti RAM na základě konfigurace administrátora nebo pomocí dynamických protokolů. Podle způsobu jak si směrovač vytváří směrovací tabulku, existují dva typy směrování:

- Statické směrování – cesta mezi cílovou a zdrojovou sítí, je předem určena v místě původu paketů. Informace o cestě však není uložena v paketu a proto musí být nakonfigurována na všech uzlech, přes které paket prochází. Konfiguraci směrovačů provádí ručně administrátor. Jedná se o jednoduché a rychlé řešení vhodné pro menší sítě. Administrátor má kontrolu nad obsahem směrovací tabulky. Nevýhodou je, že tento způsob směrování se nedokáže přizpůsobit změnám v síti a správce musí znát topologii sítě. Výhodou je, že síť není zatížena výměnou směrovacích informací a aktualizací. [24]
- Dynamické směrování – cesta mezi cílovou a zdrojovou sítí není určena v místě původu paketu ani v dalším jiném uzlu v síti. Směrovací proces je závislý na směrovacích tabulkách v jednotlivých směrovačích, které se mění v závislosti na změnách topologie. Sestavení směrovací tabulky provádí směrovací protokol, který dokáže reagovat na změny a výpadky v síti. Použití dynamických směrovacích protokolů je výhodnější ve větších sítích. Dynamické protokoly jsou výpočetně náročnější a více zatěžují procesor směrovacího prvku. Dynamické protokoly generují režijní datový tok, který zabírá určitou šířku pásma. Tento datový tok je tvořen aktualizacemi směrovacích tabulek a dalšími informacemi, které si směrovače mezi sebou vyměňují. [24]

1.2. SMĚROVACÍ PROTOKOLY

Směrovací protokoly implementují směrovací algoritmy, které zajišťují, sestavování a udržování směrovací tabulky. Směrovací protokoly mohou reagovat na změny v topologii a upravovat záznamy směrovací tabulky. Další důležitou vlastností směrovacích protokolů je, že každá linka (rozhraní) je ohodnocena určitou metrikou, která zajišťuje zvýhodnění kvalitnějších cest a jejich umístění do směrovací tabulky. Metrika je kritérium nebo soubor kritérií, podle kterého je určena kvalita cesty. Do směrovací tabulky jsou umisťovány záznamy pro rozhraní, které mají nejnižší metriku. Směrovací protokoly jako metriku využívají

počet skoků, zpoždění, šířku pásma, spolehlivost, vytížení linky, cenu.

Dynamické směrovací protokoly můžeme rozdělit na dva typy podle způsobu, jakým vytvářejí směrovací tabulku. Dynamické protokoly dělíme:

- Distance-vektor směrovací protokol
- Link-state směrovací protokol

1.2.1. DISTANCE-VEKTOR SMĚROVACÍ PROTOKOL

Protokoly z této rodiny používají algoritmus, který periodicky odesílá pomocí všesměrového (skupinového) vysílání kompletní směrovací tabulku svým přímo připojeným sousedům. Sousedé si podle přijatých informací upraví svoje směrovací tabulky, které dále distribuují. Vypočet nejlepší cesty je prováděn na základě vzdálenosti (distance) a směru (direction) k cílové síti. Kvalita cesty je ohodnocena metrikou. Metrika u distance vektor protokolů může být počet skoků mezi zdrojovou a cílovou sítí, šířka pásma nebo zpoždění. Distance vektor protokoly neznají celou topologii sítě, znají pouze vzdálenosti do příslušných sítí. Mezi distance vektor protokoly patří RIP (Routing Information Protocol), IGRP (Interior Gateway Protocol). [18][11]

1.2.2. LINK-STATE SMĚROVACÍ PROTOKOL

Link-state protokoly jsou určeny pro použití ve větších a komplexnějších sítích, kde zajistí rychlejší konvergenci než distance-vektor protokoly. Směrovače zaplavují síť informacemi o svých připojených rozhraních a sítích. Tyto zprávy se nazývají LSA (Link State Advertisement) a obsahují metriky a vlastnosti rozhraní. Na základě těchto informací si ostatní směrovače vytváří kompletní topologickou databázi, která umožňuje každému směrovači znát úplnou topologii sítě. Mimo topologické databáze ještě směrovače udržují informace o sousedech a vytvářejí směrovací tabulku. Na základě topologické databáze je pomocí algoritmu SPF (Shortest Path First) vytvořen strom, který znázorňuje topologii sítě. Kořen stromu je tvořen lokálním směrovačem, větve znázorňují dostupné sítě a uzly reprezentují směrovače. [24] Výpočet nejlepší cesty je prováděn na základě Dijkstrova algoritmu. Příkladem link-state protokolu je OSPF (Open Shortest Path First) a IS-IS (Intermediate System to Intermediate System)

1.3. SMĚROVACÍ PROTOKOL RIP (ROUTER INFORMATION PROTOCOL)

Protokol RIP je zástupce distance vektor protokolů. Existují tři verze:

- RIPv1 – pouze třídní (classfull) protokol, popsáno v RFC 1058 [13]
- RIPv2 – vylepšený RIPv1, popsáno v dokumentu RFC 2453 [19]
- RIPng – verze protokolu pro IPv6

Jedná se o jednoduchý protokol, který je vhodný pro použití v menších sítích. Jako metrika je použit počet skoků. Počet skoků udává počet směrovačů na cestě mezi zdrojovou a cílovou sítí. Maximální počet skoků je 15. [25]

Vlastnosti RIPv1:

- třídní (classful) protokol – nepřenáší se maska sítě
- aktualizace jsou odesílány každých 30 sekund pomocí všesměrového vysílání
- neobsahuje žádné bezpečnostní mechanismy

Vlastnosti RIPv2:

- beztřídní (classless) protokol – přenáší se maska sítě
- aktualizace jsou odesílány každých 30 sekund pomocí skupinového vysílání na adresu 224.0.0.9
- autentizace – výměna hesel (hašovací funkce MD5)
- značení cest (route tagging)

1.4. SMĚROVACÍ PROTOKOL OSPF (OPEN SHORTEST PATH PROTOCOL)

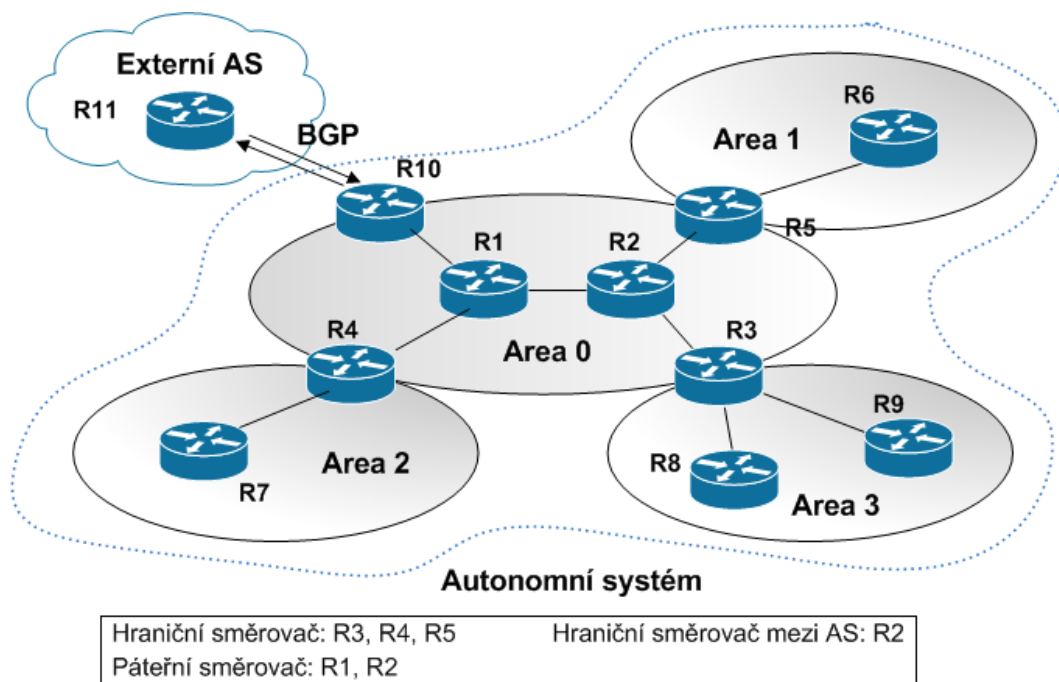
OSPF je otevřený protokol, který se používá pro směrování uvnitř autonomního systému. Autonomní systém (dále AS) je množina sítí a pod společnou technickou správou. Aktuálně se používají dvě verze: OSPFv2 pro protokol IPv4, který je popsán v dokumentu RFC 2328 [7] a OSPFv3 pro IPv6, který se je popsán v dokumentu RFC 2740. [10] OSPF podporuje beztřídní mezidoménové směrování CIDR a také směrování s maskami s proměnou délkou VLSM (Variable-length Subnet Masking). OSPF je nejrozšířenější protokol pro směrování v rámci autonomního systému.

OSPF umožňuje dělení autonomního systému na menší části, kterým se říká oblasti (Area). Dělení na menší části výrazně přispívá k menšímu zatěžování sítě režijními informacemi. Změny v topologii se šíří pouze v dané oblasti a využívá se sumarizace záznamů pro zmenšení směrovacích tabulek. V každém AS je vždy jedna páteřní oblast (backbone area) ke které jsou připojeny ostatní oblasti. Hierarchie oblastí u protokolu OSPF je vidět na obrázku 1.1. [25]

Každá oblast je popsána 32-bitovým identifikátorem Area ID. Páteřní oblast bývá označena Area ID 0.0.0.0. Směrovače v každé oblasti se podle funkčnosti dělí na několik typů:

- Hraniční směrovač (Area Boundary Router - ABR) – jedná se o směrovač, který je součástí několika oblastí. Pro každou oblast vytváří oddělenou topologickou databázi. Každý ABR směrovač musí mít alespoň jedno rozhraní v páteřní oblasti a jedno rozhraní v normální oblasti. Hraniční směrovač plní hlavně funkci šíření, filtrování a sumarizaci topologických informací, které jsou přeposílány mezi oblastmi. [26]
- Hraniční směrovač mezi AS (Autonomous System Boundary Router - ASBR) – směrovač se nachází v páteřní oblasti a umožňuje komunikaci s dalšími autonomními systémy. Na tomto směrovači běží kromě OSPF také nějaký EGP (Exterior Gateway Protocol) protokol, nejčastěji se používá BGP protokol. ASBR provádí filtrování a sumarizaci informací přicházející z jiného AS. [25] [26]

- Pátevní směrovač – směrovače leží v pátevní oblasti a mají alespoň jedno rozhraní spojené s dalším směrovačem v pátevní oblasti. Směrovací informace získávají stejnými metodami a algoritmy jako interní směrovače. [26]
- Vnitřní směrovač – má všechny rozhraní ve stejné oblasti. Všechny vnitřní směrovače v jedné oblasti mají identickou topologickou databázi. [26]



Obr. 1.1: Oblasti u OSPF

OSPF může fungovat v pěti typech sítí: síť typu bod-bod (point-to-point), všesměrové síť (broadcast), vícebodové síť bez všesměrového vysílání (non-broadcast multipleaccess - NBMA), bod-skupina bodů (point-to-multipoint) a virtuální linky. [25]

V následujícím textu je hlavně popsán nejčastější typ sítě, kterým jsou síť se všesměrovým vysíláním. U těchto typů sítí by záplavové šíření informací mezi všemi směrovači způsobilo velké zatížení linek. Proto je v každém segmentu zvolen tzv. DR (Designated Router) směrovač, který slouží jako centrální bod pro přijímání a šíření aktualizací. DR primárně zajišťuje, aby všechny směrovače v oblasti měly totožné topologické databáze. Všechny směrovače v oblasti komunikují s DR směrovačem. Pokud dojde v síti ke změně topologie, směrovače odesílají informaci o změně pomocí skupinové IP adresy 224.0.0.6, která zajistí doručení pouze DR a BDR směrovačům. DR směrovač tyto získané informace přeposílá na skupinovou IP adresu 224.0.0.5, která zajistí doručení všem směrovačům v dané oblasti. V případě výpadku DR směrovače je v oblasti zvolen i záložní směrovač BDR (Backup Designated Router). [22] [11]

1.4.1. VÝPOČET METRIKY ROZHRAŇÍ

OSPF využívá pro výpočet nejlepší cesty Dijkstrův algoritmus. Algoritmus využívá pro výpočet metriku. Metrika je kritérium, které umožňuje ohodnotit kvalitu linky (rozhraní). Metrika umožňuje směrovacím protokolům, aby do směrovacích tabulek byly vybírány nejkvalitnější a nejvýhodnější cesty.

OSPF jako metriku používá tzv. cenu linky (link cost). Cena linky je dána číslem v rozmezí 1-65535. Čím je menší číslo, tím je lepší metrika a cesta bude mít větší prioritu při výpočtu nejlepší cesty. V základním nastavení Cisco směrovačů určuje OSPF protokol cenu rozhraní na základě šířky pásma rozhraní. V tabulce č. 1.1 jsou uvedeny základní metriky pro nejpoužívanější technologie.

Pro ceny linky u OSPF platí následující vzorec:

$$\text{Cena} = 100 / (\text{šířka pásma v Mbps})$$

Problém tohoto schématu je, že maximální šířka pásma může být pouze 100 Mbps. V případě použití technologie Gigabit Ethernet bude tato linka ohodnocena stejnou metrikou jako Fast Ethernet linka. OSPF proto umožňuje ruční konfiguraci ceny linky.

Tab. 1.1: Základní metriky OSPF protokolu

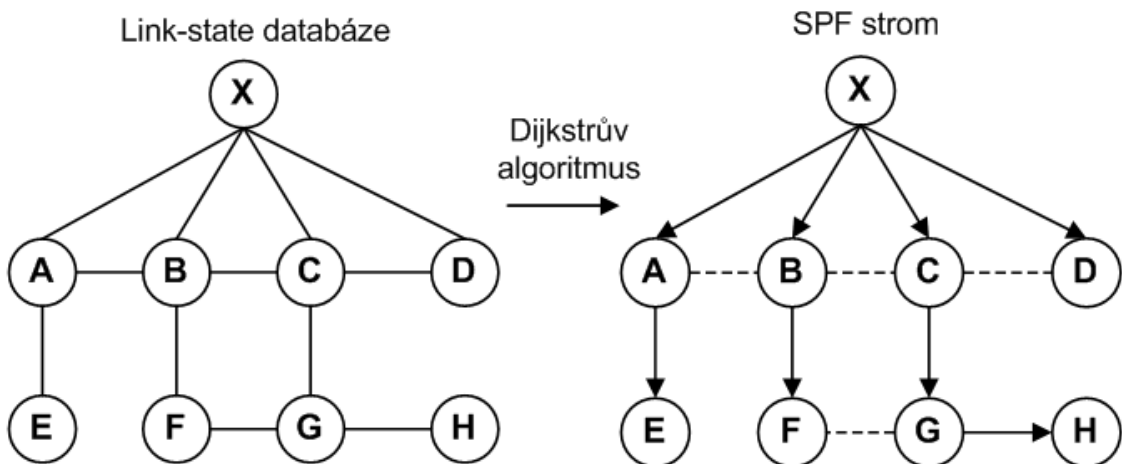
Typ linky	Metrika
56-kbps seriová linka	1785
64-kbps seriová linka	1562
T1 (1,544-Mbps seriová linka)	64
E1 (2,048-Mbps seriová linka)	48
Ethernet	10
FastEthernet	1
FDDI	1
ATM	1

1.4.2. DIJKSTRŮV ALGORITMUS

Dijkstrův algoritmus je matematický algoritmus, který navrhl nizozemský informatik Edsger W. Dijkstra. Dijkstrův algoritmus je obecně využíván pro nalezení nejkratší cesty v grafu. Fungování algoritmu je naznačeno na obrázku 1.2. Algoritmus na základě ceny linky vypočítá nejlepší cestu do požadované sítě.

Výpočet SPF:

- Směrovač H oznamuje pomocí LSA svoji existenci směrovači G. Směrovač G předává tyto LSA zprávy spolu se svými LSA směrovačům C a F. Směrovače C a F přidají ke svým LSA informace o svých rozhraních a předávají je svým sousedům atd.
- Záplavové šíření LSA se řídí pravidlem Split Horizon, které říká, že směrovače nesmí přeposílat LSA zprávy směrem kterým přišly. Pokud směrovač G přijme LSA od směrovače H, posílá je dále na všechny rozhraní kromě rozhraní, kterým je směrovač propojen s H.
- Směrovač označen písmenem X zjistí pomocí LSA zpráv, že má sousedy směrovače A, B, C, D. S těmito směrovači naváže spojení (Adjacency). Od těchto směrovačů se také dozví o ostatních směrovačích v oblasti a vytvoří si link-state databázi. Tato databáze mu umožňuje znát celou topologii sítě. Grafické znázornění získané databáze je naznačeno v levé části obrázku 1.2
- Jakmile je známá topologie, je spuštěn Dijkstrův algoritmus, který na základě metrik určí nejkratší cesty ke všem ostatním směrovačům. Algoritmus vytvoří tzv. SPF strom, který je znázorněn v pravé části obrázku 1.2. Na základě tohoto stromu jsou pak vkládány záznamy do směrovací tabulky.
- Všechny linky v naznačeném jsou Fast Ethernet a počítá se s cenou linky jedna.



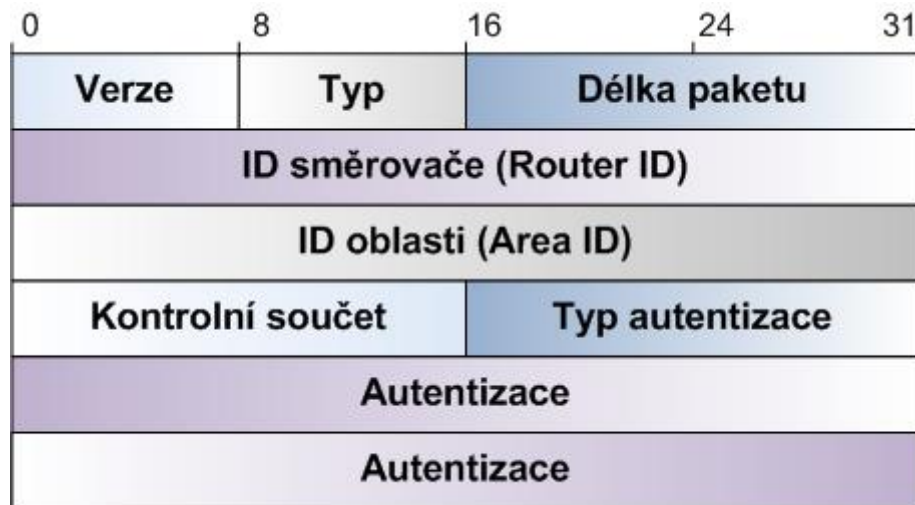
Obr. 1.2: Výpočet SPF

1.4.3. FORMÁT OSPF PAKETU

OSPF pakety jsou přímo zapouzdřeny přímo v IP paketu. OSPF nevyužívá transportních protokolů TCP nebo UDP. Pro zajištění spolehlivosti přenosu paketů má OSPF vlastní mechanismus. Pro přenos je používán protokol č. 89.

OSPF pakety slouží k přenosu LSA zpráv a dalších datových struktur. Všechny OSPF pakety mají stejnou strukturu záhlaví o fixní velikosti 24 bajtů. Další

struktura OSPF paketu se liší podle typu paketu. Strukturu hlavičky je možné vidět na obrázku č. 1.3. [26]



Obr. 1.3: Struktura OSPF paketu

Typ v záhlaví OSPF paketu udává, o jaký druh paketu se jedná. V OSPF protokolu jsou definovány následující typy paketů:

- Hello paket – používá se pro navazování a udržování sousedství mezi směrovači. Hello paket je používán pro volbu DR a BDR směrovače. Slouží také k vyjednávání dalších parametrů OSPF (síťová maska, hello interval, deadRouter interval).
- Paket pro popis topologie (Database Description Packet - DDP) – tyto pakety slouží k synchronizaci topologických databází při inicializaci sousedství. Vyměňovány jsou pouze názvy link-state záznamů (link-state záznam obsahuje vlastnosti rozhraní), není přenášena úplná topologická informace. [24]
- Link-state požadavek (Link-State Request - LSR) – směrovač si pomocí této zprávy vyžádá konkrétní záznam z topologické databáze souseda.
- Link-state aktualizace (Link-State Update - LSU) – zajišťují samotný přenos topologické informace mezi sousedy. LSU je tvořena jedním nebo několika LSA záznamy. [24]
- Link-state potvrzení (Link-State Acknowledgement) – potvrzení úspěšného přijetí LSU. Zajišťuje spolehlivost procesu záplavového šíření link-state záznamů napříč oblastí. [22] [11]

Důležitým součástí OSPF záhlaví je ID směrovače. ID směrovače je čtyř bajtové číslo, které identifikuje odesílající směrovač. ID směrovače je zvoleno jako nejnižší adresa rozhraní směrovače. Bývá zvykem jako ID směrovače volit virtuální rozhraní (loopback), u kterých nemůže dojít k vypnutí rozhraní.

1.4.4. FORMÁT LSA ZPRÁVY

LSA zpráva je základní datová struktura, kterou odesílá každý směrovač v autonomním systému. LSA zprávy slouží k vytváření topologické databáze, na

základě které je pak vytvářena směrovací tabulka. LSA zprávy jsou obsaženy v OSPF paketu LSU (Link State Update). LSA zprávy jsou označeny pomocí identifikátoru Link-State ID. Existuje několik typů LSA zpráv. Směrovací LSA (router-LSA) a síťové LSA (network-LSA) popisují, jak jsou směrovače a jednotlivé sítě propojeny. Sumární LSA zajišťují zjednodušení velikosti směrovacích tabulek. Externí LSA zajišťují zpracování směrovacích záznamů z jiných autonomních systémů. Všechny LSA zprávy obsahují záhlaví o velikosti 20 bajtů. Struktura záhlaví je vidět na obrázku 1.4.



Obr. 1.4: Struktura záhlaví LSA

- Věk LS (LS age) – udává čas, kdy byla LSA zpráva vytvořena. Maximální životnost LSA je 3600 sekund. Věk je průběžně inkrementován a pokud je LSA starší než 3600 sekund, musí být záznam smazán z databáze. [25]
- Možnosti – definuje několik volitelných parametrů, které záleží na vybavenosti směrovače. Nastavení parametrů tohoto pole záleží na možnostech, typu směrovače a typu OSPF oblasti.
- Typ LS – typ LSA zprávy
- Link-state ID – zajišťuje identifikaci LSA zprávy v databázi. Link-State ID obsahuje čtyř bajtové číslo, které je dáno typem LSA.
- Oznamující směrovač – obsahuje ID směrovače, který vygeneroval LSA
- LS sekvenční číslo – obsahuje 32-bitové číslo, které je inkrementováno při každém vygenerování nové LSA. Slouží k detekci starých a duplicitním LSA záznamům. [25]
- LS kontrolní součet – slouží pro kontrolu integrity LSA zprávy. Kontrolní součet je počítaný z celého LSA kromě pole věk LS, které je inkrementováno.

1.4.5. TYPY LSA ZPRÁV

Podle účelu se LSA dělí na několik typů. Jednotlivé typy jsou rozlišeny pomocí identifikátoru typ LS v hlavičce LSA. Typy LSA a jejich identifikátory můžeme vidět v tabulce 1.2. Nejpoužívanější typy LSA jsou označeny identifikátory 1 – 5.

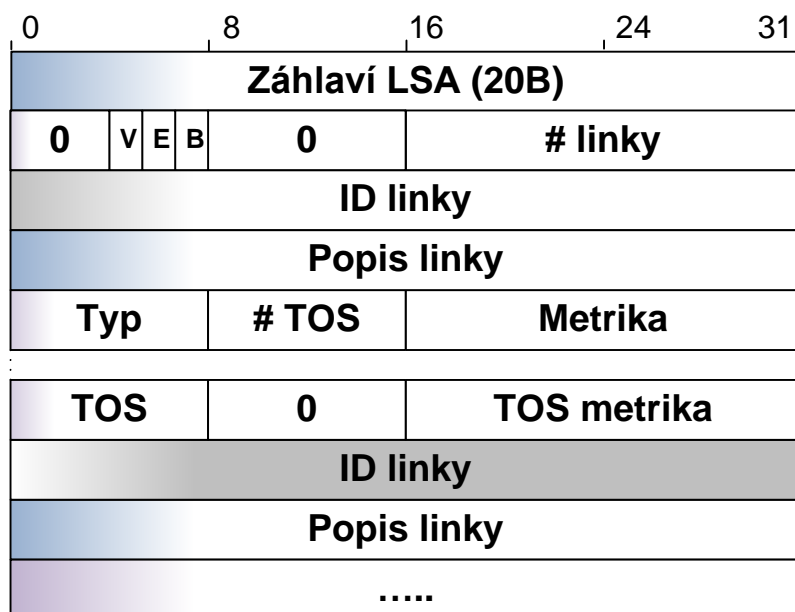
Tab. 1.2: Typy LSA zpráv

Typ LSA	Název LSA
1	Směrovací LSA
2	Síťové LSA
3	Sumární LSA
4	
5	Externí LSA
6	Skupinové LSA
7	LSA pro oblasti typu "no so stubby"
8	LSA pro BGP
9,10,11	rezerva

Směrovací LSA (Router LSA)

Tento typ zprávy generuje každý směrovač v oblasti. Pokud je součástí více oblastí generuje tento typ zprávy pro každou oblast. Směrovací LSA zprávy se mohou šířit pouze v oblasti, ve které byly vytvořeny. V této zprávě směrovač oznamuje svoji existenci v oblasti a uvádí seznam jeho rozhraní, která jsou připojena k okolním směrovačům a sítím v oblasti. Obsahuje popis a stav rozhraní. Formát směrovací LSA zprávy je vidět na obrázku 1.5. Prvních 20 bajtů tvoří záhlaví, které bylo uvedeno na obrázku 1.4. [22][25] [26]

Pomocí bitů E (external) a B (boundary) směrovač informuje, že je hraničním směrovačem v oblasti nebo v autonomním systému. Bit V (virtual) slouží k označení směrovače, který je koncovým bodem u virtuálního spojení dvou oblastí. V jedné LSA zprávě musí být uvedeny všechny rozhraní daného směrovače. Počet rozhraní, které směrovač oznamuje, jsou uvedeny v poli #linky. Linky jsou podle způsobu připojení rozděleny na čtyři typy. Popis jednotlivých typů je vidět v tabulce 1.3. První sloupec udává číslo, kterým je linka v LSA identifikována. Následuje popis linky a nejdůležitější sloupec udává, jaký identifikátor obsahuje pole ID linky. Obsah dalších polí je závislý na typu linky.



Obr. 1.5: Směrovací LSA

Pole ID linky identifikuje objekt, který je připojen k příslušnému rozhraní. Pokud se jedná o objekt, který je také zdrojem LSA zpráv (další směrovač, transitní síť) musí být ID linky shodné s link-state ID sousedního objektu. Toto je důležitá vlastnost, která je důležitá při vyhledávání sousedských LSA v topologické databázi během výpočtu nejlepší cesty.

Tab. 1.3: Typy linek (rozhraní) u směrovacích LSA

Typ linky	Popis	ID linky
1	Připojení k jinému směrovači	ID sousedního směrovače
2	Připojení k tranzitní síti	IP adresa DR smerovače
3	Připojení ke koncové síti	Adresa sítě/podsítě
4	Virtuální linka	ID sousedního směrovače

Obsah popisu linky je závislý na typu rozhraní. Například u koncových sítí je v tomto poli uvedena síťová maska. Pro spojení bod-bod je v tomto poli uvedena hodnota ifIndex pro MIB-II (Management Information Base). [1] U dalších typů linek je zde uvedena IP adresa rozhraní směrovače. Další pole popisují metriku a cenu rozhraní a nastavení typu služby TOS (Type of service). TOS je uváděno pro zpětnou kompatibilitu se staršími verzemi OSPF. Hodnota TOS v OSPF paketu je uvedena v decimálním tvaru, která je mapována na binární hodnotu TOS, která je uvedena v záhlaví IP paketu. [22][25] [26]

Sít'ové LSA zprávy (Network LSA)

Tento typ LSA generují DR směrovače ve všesměrových a NBMA segmentech. Sít'ové LSA obsahuje seznam směrovačů v síti. Tento typ se šíří pouze do oblastí, do kterých daná síť patří. [22][25] [26]

DR směrovače začnou sít'ové LSA šířit, jakmile mají navázáno plné sousedství alespoň s jedním směrovačem. LSA obsahuje ID všech směrovačů (Router-ID), se kterými DR směrovač navázal spojení. V LSA je uvedeno ID směrovače DR. Link-state ID obsaženo v záhlaví sít'ové LSA zprávy udává IP adresu rozhraní DR směrovače. [22][25] [26]



Obr. 1.6: Sít'ová LSA zpráva

Protože tento typ zprávy popisuje pouze přímo připojené směrovače, není třeba uvádět metriku rozhraní, protože vzdálenost mezi nimi je nula. Sít'ová LSA zpráva proto mimo záhlaví obsahuje pouze dvě pole:

- Sít'ová maska – udává sít'ovou masku sítě v hexadecimálním tvaru. Např.: maska 255.255.255.0 bude v LSA zapsána ve tvaru 0xfffff00.
- Připojený směrovač – obsahuje seznam ID směrovačů, které s DR směrovačem navázali plné sousedství. Délka seznamu může být omezena polem délka v záhlaví.

Sumární LSA zprávy (Summary LSA)

Tento typ LSA zpráv generují hraniční směrovače oblastí ABR. Tento typ se používá pro popis cílových sítí uvnitř autonomního systému. LSA typu 3 a 4 jsou opět šířeny pouze v rámci příslušné oblasti. Oba typy mají stejnou strukturu LSA zprávy. Rozdíl je v obsahu pole link-state ID v záhlaví. Pole link-state ID u typu 3 bude obsahovat adresu cílové sítě. Pole link-state ID u typu 4 bude obsahovat ID hraničního směrovače autonomního systému ASBR. Struktura sumární LSA zprávy je vidět na obrázku 1.7. [22][25] [26]

Sumární LSA typu 3 je pro rozhlašování použita, pokud je cílový objekt IP síť. Pokud je cílový objekt hraniční směrovač autonomního systému, je použita sumární zpráva LSA typu 4.

V sumárních LSA zprávách typu 3 jsou rozhlašovány sítě, které se směrovač naučil pomocí směrovacích LSA. LSA typu 3 zajišťuje propagaci sítí za hranice oblasti. ABR směrovač přijme LSA typu 1, na základě které vygeneruje

LSA typu 3, kterou šíří v sousední oblasti. LSA typu 3 se šíří pouze v jedné oblasti, avšak pokud dorazí na hranici s jinou oblastí je obnovena ABR směrovačem a přeposlána do další oblasti.

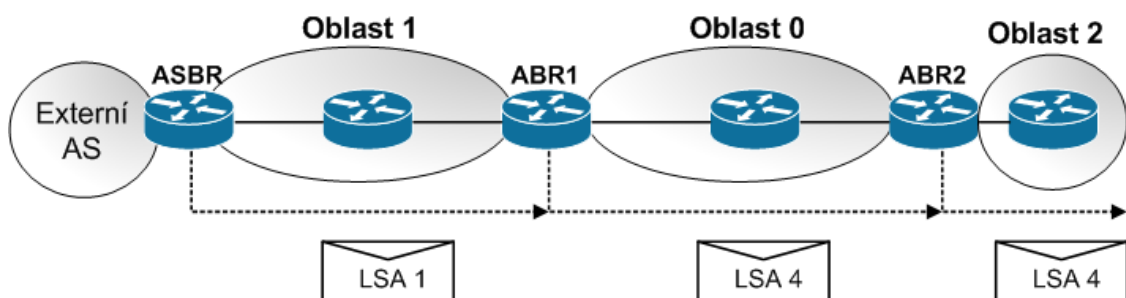
V základním nastavení LSA typu 3 zajišťuje propagaci všech sítí a podsítí do páteřní oblasti. Avšak OSPF nezajišťuje sumarizaci sítí a podsítí. Proto je vhodné, aby byla sumarizace nastavena manuálně na každém ABR směrovači. Oblast bude zaplavována menšími toky dat a dojde ke snížení velikosti směrovací tabulky. [22][25] [26]



Obr. 1.7: Sumární LSA zpráva

LSA typu 4 je generována pouze pokud se v oblasti vyskytuje ASBR. LSA typu 4 identifikuje ASBR směrovač a informuje o cestě k tomuto směrovači. Veškerý provoz směřující mimo autonomní systém potřebuje vědět cestu k ASBR směrovači, který je původcem externích záznamů ve směrovací tabulce.

Způsob šíření LSA typu 4 je vidět na obrázku 1.8. ASBR směrovač posílá do oblasti LSA typu 1 s nastaveným bitem E, který informuje o existenci ASBR směrovače v oblasti. Směrovač ABR1 přijme LSA typ 1 a protože uvidí nastavený bit E, vygeneruje LSA typu 4 a šíří ji páteřní oblastí. Směrovač ABR2 zajistí obnovení LSA zprávy a šíření v další oblasti. [22][25] [26]



Obr. 1.8: Šíření LSA zprávy typu 4

Externí LSA zprávy

Externí LSA zprávy jsou určeny k popisu cest mimo autonomní systém. LSA typu 5 jsou generovány ASBR směrovačem do všech oblastí autonomního systému. Link-state ID u LSA typu 5 obsahuje číslo externí sítě. Opět jako u LSA typu 3 a 4 je důležité nakonfigurovat na ASBR sumarizaci, aby se snížil počet LSA typu 5 v autonomním systému. [22][25] [26]

1.4.6. POPIS FUNGOVÁNÍ OSPF

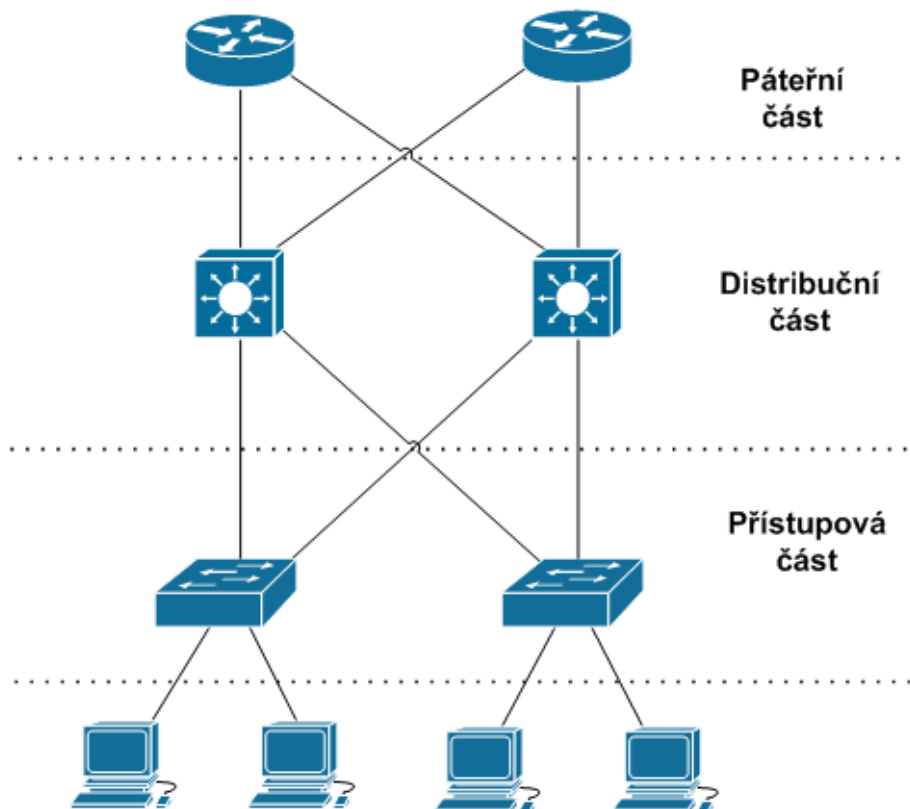
V předchozí části byly popsány hlavní pojmy a součásti protokolu OSPF. Nyní bude popsána činnost protokolu od spuštění po plnou konvergenci. Pro jednoduchost bude vysvětlen princip fungování ve vícebodových sítích, kde je volen DR a BDR směrovač.

1. Nalezení sousedů - po připojení směrovače do sítě a spuštění instance protokolu OSPF se směrovač snaží navázat spojení s ostatními směrovači. Hello pakety jsou vysílány na skupinovou adresu 224.0.0.5. Aby se směrovače staly sousedy, musí souhlasit síťová adresa a maska, číslo oblasti a její typ, hello interval a deadInterval.
2. Volba DR/BDR – každému směrovači je možné nakonfigurovat prioritu od 0 – 255. Směrovač s nejvyšší prioritou se stává DR směrovačem. V případě, že není možná volba pomocí priorit, je volba provedena na základě identifikátoru směrovače (Router ID).
3. Synchronizace topologických databází – při synchronizaci musí být jeden z dvojice směrovačů DR/BDR. Směrovače si vymění obsahy svých databází a zjistí které informace má soused aktuálnější. O tyto záznamy požádá vysláním paketu LSR. Soused informaci pošle pomocí paketu LSU. Synchronizace je dokončena pokud mají oba směrovače stejné topologické databáze.
4. Výpočet nejlepší cesty – na základě topologické databáze jsou pomocí Dijkstrova algoritmu vypočteny nejkratší cesty do všech sítích o kterých směrovač ví.

2. SPOLEHLIVOST PROVOZU V IP SÍTÍCH

Při návrhu počítačové sítě je nutné zajistit dostatečnou spolehlivost. Při návrhu je důležité myslet na možné výpadky v síti. Může se jednat o výpadky linek nebo dokonce jednotlivých zařízení. Síť musí být navržena tak, aby byla těmito událostmi minimálně ovlivněna. Pokud dojde k výpadku, je důležitá jeho rychlá detekce a poté zajištění příslušné akce, která opět zajistí záložní provoz.

První podmínkou pro spolehlivou síť je, aby síť byla dostatečně redundantní. Je třeba zajistit záložní linky a redundantní zařízení, které budou sloužit pro přenos v případě výpadku primární linky. Např. v LAN sítích by měli být přístupové přepínače (access switch) připojovány na dva různé distribuční přepínače (distribution switch). Pokud bude splněna tato podmínka, výpadek distribučního přepínače nezpůsobí nedostupnost připojených podsítí. V případě propojování autonomních systémů je důležité mít vytvořené záložní linky. Na tyto linky je poté přesměrován provoz v případě výpadku. V některých případech mají záložní linky nižší kapacitu než linky primární a proto může docházet ke zpoždování a zahazování paketů.



Obr. 2.1: Struktura redundantní topologie

Zajištění přesměrování provozu mají na starosti dynamické směrovací protokoly. S přesměrováním souvisí pojem konvergence. Konvergence je stav sítě, kdy všechny uzly sítě znají topologii sítě, mají vytvořeny směrovací tabulky a jsou

schopny směrovat provoz. Doba, za jakou síť konverguje, je velmi důležitý kvalitativní parametr, který určuje, jak je síť schopna reagovat na výpadky. Rychlost konvergence ovlivňují tři základní věci:

- Detekce výpadku – rychlost za jakou je zařízení schopno detekovat výpadek.
- Šíření informace – rychlost jakou se ostatní uzly v síti o výpadku dozvědí
- Náprava výpadku – rychlost za jakou budou zařízení, které byly o výpadku informovány, schopna posílat data alternativní cestou. [12]

K detekci může být použit hardwarový detekční mechanismus. Jedná se o rychlé řešení, ale nevýhodou je, že některé hardwarové detekční mechanismy nemohou komunikovat s vyššími vrstvami ISO/OSI modelu a proto musí být problém detekce výpadku řešen vlastními směrovacími protokoly.

Detekční mechanismy směrovacích protokolů jsou založeny na jednoduchém principu, kdy přílehlající směrovače spolu navazují sousedství. Směrovače si v pravidelných intervalech posílají zprávy, kterými toto sousedství obnovují. Pokud paket v určitém časovém intervalu nedorazí, směrovač považuje sousedský směrovač za nedostupný. Rychlost detekce je možné ovlivnit nastavením intervalů pro odesílání zpráv pro aktualizaci sousedství. Pokud nastavíme rychlejší odesílání paketů, směrovač bude moci rychleji detekovat výpadek, avšak dojde k větší spotřebě pásma linky režijními informacemi. Volba délky intervalu pro posílání těchto paketů proto musí být kompromis mezi výkonem a rychlostí detekce.

2.1. DETEKCE VÝPADKU U OSPF

K detekci výpadku u OSPF slouží hello protokol. Hello protokol slouží k objevování sousedských směrovačů a poté k navázání spojení mezi těmito směrovači. Po navázání spojení mezi sousedy jsou v určitých intervalech odesílány hello pakety, které slouží pro udržování spojení mezi směrovači. Standardní nastavení hello intervalu je 10 sekund. Pokud směrovač neobdrží hello paket v intervalu, který má název routerDeadInterval (dále pouze RDI), považuje linku za nedostupnou. RDI je standardně nastaven na 40 sekund (čtyřnásobek hello intervalu). Pokud dojde ke změně stavu linky, směrovač vygeneruje LSA zprávu, kterou informuje o změně ostatní směrovače v oblasti. Vygenerovaná LSA zpráva postupně zaplavuje celou oblast. Všechny směrovače, které obdrželi LSA znovu spouští SPF algoritmus a vypočítávají nové cesty a upravují záznamy ve směrovací tabulce. Jakmile směrovač přijme aktualizaci LSA, naplánuje spuštění algoritmu SPF. Jedná se o náročnou operaci, která velmi zatěžuje procesor směrovače, proto směrovač čeká, jestli nepříjde další LSA. Doba, po kterou směrovač čeká, se nazývá spfDelay a má standardně hodnotu 5 sekund. Tento interval zamezuje zbytečnému spouštění SPF např. v případě planého poplachu, kdy k výpadku skutečně nedošlo. OSPF také omezuje frekvenci, jakou budou výpočty SPF spouštěny. Další SPF výpočet může být spuštěn po uplynutí intervalu spfHoldTime. Hello paket je odesílán v intervalu 10 sekund a interval

RDI je 40 sekund. Z toho vyplývá, že k detekci výpadku může dojít v rozmezí 30 – 40 sekund. Pokud k této hodnotě připočítáme ještě dobu šíření LSA zpráv a výpočet nových cest, jedná se o poměrně vysokou hodnotu. [12] [9]

Hodnoty hello a RDI intervalu je možné měnit. Doba detekce výpadku je možné snížit, zkrácením intervalu pro odesílání hello paketu. Častější odesílání hello paketu způsobí snížení doby detekce výpadku. Pokud by byla nastavena příliš malá hodnota, může dojít k zahlcení linky hello pakety a následně může docházet k ukládání paketů do front a případně i k zahazování paketů. Hello pakety se tak nemusejí v intervalu RDI dostat k sousednímu směrovači. Směrovač považuje linku za nedostupnou, i když k žádnému výpadku linky nedošlo. Takový falešný výpadek způsobí zatěžování sítě zbytečnými LSA a přepočítávání nových cest zbytečně zabírají procesorový čas. Při nastavování hello intervalu je potřeba zvolit správnou hodnotu, která nebude příliš zatěžovat síť a nebude vyvolávat falešné výpadky. V tabulce 2.1 jsou uvedeny hodnoty naměřené v simulátoru NS2 publikované v článku [12]. DV značí dobu detekce výpadku a CDO je celková doba obnovení. [12]

Tab. 2.1: Tabulka hodnot rychlosti detekce výpadku a doby obnovení

Hello interval	DV	CDO
10s	32,08	36,6
2s	7,82	11,68
1s	3,81	9,02

3. QoS - KVALITA SLUŽEB

Starší počítačové sítě byly navrhovány pro zajištění rychlého a nespolehlivého přenosu dat. Tento způsob byl nazván jako přenos s maximálním úsilím (best-effort). Nevýhodou tohoto přenosu je, že uzly v síti neumí rozpoznat procházející datové toky a proto se všemi datovými jednotkami zacházejí stejně. Přenos best-effort také nedokáže zajistit, aby nedošlo k zabránění pásma jednou aplikací.

V současných moderních sítích se pohybují data různých služeb a aplikací, které sdílí společné přenosové pásmo. Avšak každá služba či aplikace má jiné nároky na přenosové parametry. Některé služby jsou citlivé na zpoždění, ale tolerují určitou ztrátu informace na přenosové cestě. Takovým příkladem jsou hlasové služby v reálném čase. Naopak například datové služby jako přenos dat pomocí FTP protokolu mohou pracovat s větším zpožděním, ale je nutné zajistit, aby byla nulová ztrátovost. Proto bylo třeba zajistit, aby bylo možné rozlišovat datové toky jednotlivých aplikací a udělovat jednotlivým tokům různé priority. K tomuto účelu byl navrhnut systém kvality služeb QoS. Kvalita služeb začala být nepostradatelnou při větším rozšíření služeb náchylných na zpoždění a kolísání zpoždění.

QoS můžeme definovat jako soubor mechanismů, protokolů a nastavení, které zajišťují dodržení potřebných přenosových vlastností datových jednotek v počítačových sítích. QoS je nasazováno hlavně v souvislosti se službami poskytujícími přenos dat v reálném čase. Jedná se zejména o přenos hlasu, videokonference, streamování videa atd. Účelem QoS je rozpoznat a identifikovat jednotlivé toky dat a podle daných parametrů tyto toky klasifikovat do tříd a označovat. Směrovače a další síťové prvky se podle značek snaží zajistit požadovanou kvalitu služeb. S datovými jednotkami stejné třídy je zacházeno podle stejných pravidel. Kvalita služeb je určena parametry, které specifikují potřeby jednotlivých služeb.

Hlavní parametry používané v QoS:

- Zpoždění (delay) – čas, který uplyne od odeslání paketu zdrojovým uzlem až po jeho přijetí cílovým uzlem. Zpoždění je soumou dílčích zpoždění, která zahrnují zpoždění způsobené šířením přenosovou cestou a zpoždění způsobené zpracováním paketů v mezilehlých uzlech (směrovače, prepínače), koncovými zařízeními atd. [25]
- Variace zpoždění (jitter) – rozdíl mezi zpožděním paketu na přenosové cestě a referenčním zpožděním. Referenční zpoždění je průměrné zpoždění množiny paketů.
- Ztrátovost paketů – parametr udává počet ztracených paketů mezi zdrojem a cílem.
- Šířka pásma (bandwidth) – počet datových jednotek za jednotku času.

QoS implementuje 3 základní mechanismy:

- Služby s maximálním úsilím (best-effort) – datové jednotky jsou doručovány k cíli nejrychlejším způsobem. Se všemi datovými jednotkami je zacházeno stejným způsobem. Pokud nejsou dostupné síťové prostředky, dochází k zahazování datových jednotek.
- Prioritní mechanismus – datové jednotky jsou identifikovány a klasifikovány do tříd podle nastavených pravidel. S každou třídou je zacházeno odlišně. Na principu prioritních mechanismů pracují systémy DiffServ (Differentiated services), SBM (Subnet Bandwidth Management).
- Rezervace uzlů – je založena na rezervaci síťových prostředků danou službou. Služba má rezervované síťové prostředky po dobu přenosu dat. Po ukončení spojení jsou prostředky uvolněny a ostatní služby je mohou využívat. Tento mechanismus používá systém IntServ (Integrated Services). IntServ používá pro rezervaci prostředků protokol RSVP (ReSource reservation Protokol).

3.1. INTEGROVANÉ SLUŽBY - INTSERV

IntServ je založen na rezervaci přenosového pásma. IntServ zajišťuje kvalitu služeb na principu konec-konec (end-to-end), tedy od zdrojového uzlu až po uzel cílový. IntServ proto musí být podporován aplikací i všemi uzly na přenosové cestě. Před samotným přenosem dat je nutné vyjednat potřebné síťové prostředky a provést rezervaci pásma. Aplikace nejdříve požádá o rezervaci určitých síťových prostředků, pokud jsou prostředky volné, je provedena rezervace na všech uzlech. Tento proces se nazývá správa řízení (Policy Control). K tomuto účelu slouží rezervační protokoly. U IntServ se používá protokol RSVP.

Integrované služby definují 2 typy služeb:

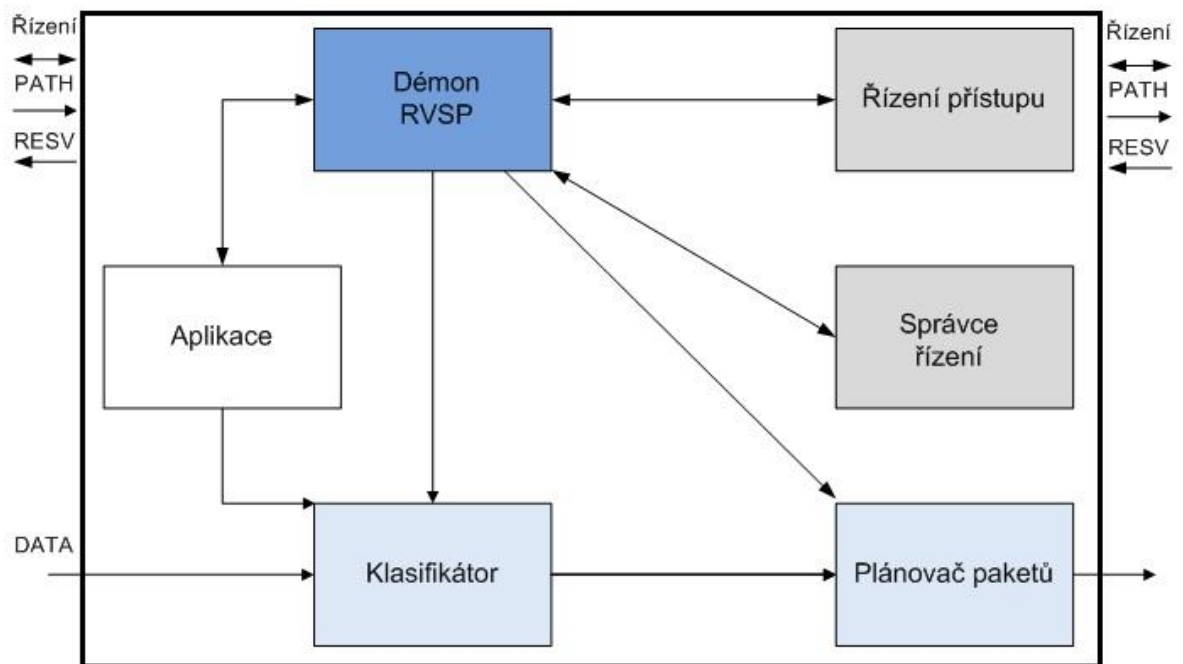
- Služba s garantovanými parametry (Guaranteed) – tento typ je určen pro služby pro přenos v reálném čase. Garantuje dodržení zpoždění a přenosového pásma.
- Služba s řízenou zátěží (Controlled Load) – pakety využívající tuto službu, budou zahazovány jako poslední.

IntServ se skládá z několika základních částí:

- Plánovač paketů (Packet Scheduler)
- Klasifikátor (Packet Classifier)
- Správa řízení (Policy Control)
- Řízení přístupu (Admission Control)
- Démon RSVP

Démon RSVP je nejdůležitější částí IntServ. Řídí celý proces rezervace. Komunikace RSVP démona s ostatními částmi IntServ je vidět na obrázku 3.1. RSVP démon komunikuje s bloky správa řízení a řízení přístupu. Správce řízení

zjišťuje, zda má uzel k dispozici síťové prostředky, aby zajistil potřebnou kvalitu služeb. Jednotka řízení přístupu zjišťuje, zda má příslušná služba oprávnění rezervaci provést. Pokud uzel nemůže uvolnit požadované prostředky, protokol RSVP odešle žadateli o rezervaci chybovou zprávu. Pokud je dostupné požadované pásmo a aplikace má oprávnění pro rezervaci RSVP démon provede nastavení parametrů klasifikátoru a plánovače paketů. Démon RVSP poté provede nastavení parametru paketového klasifikátoru a plánovače paketů tak, aby byla zajištěna potřebná kvalita služeb. [25]



Obr. 3.1: Architektura IntServ

3.1.1. RSVP PROTOKOL

RSVP je signalizační protokol, který je využíván pro rezervaci přenosového pásma. RSVP provádí rezervaci toků v jednom směru. Každý tok je definován adresou a portem cílového uzlu a identifikátorem síťového protokolu (IntServ podporuje více síťových protokolů, nejčastěji se jedná o IP protokol). RSVP protokol lze využít pro jednosměrné (unicast) nebo i pro skupinové vysílání (multicast). Je možné ho provozovat jak na IPv4 tak i na IPv6 sítích.

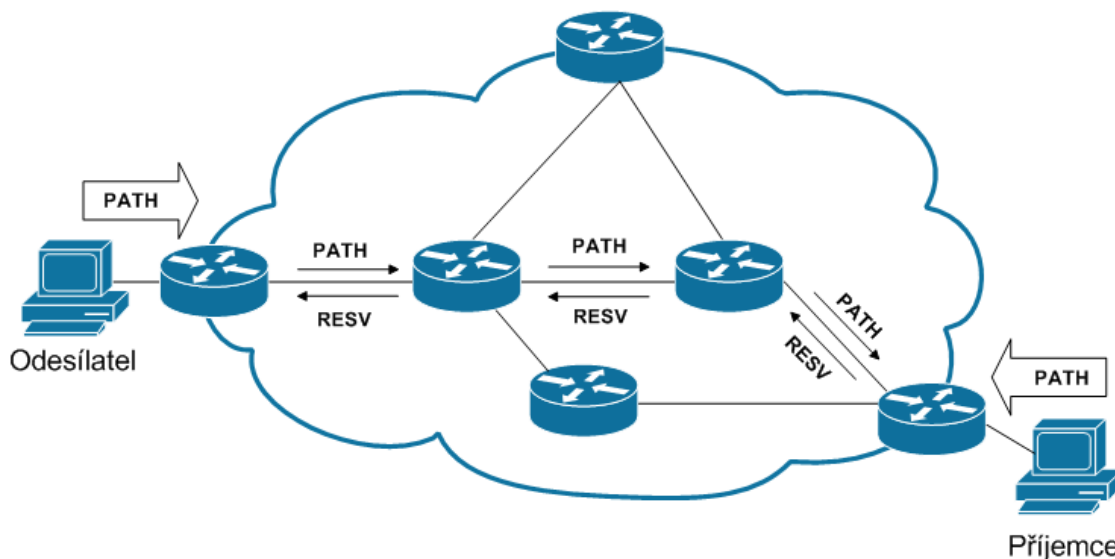
3.1.2. PRŮBĚH KOMUNIKACE RSVP

Nejdůležitějšími dvěma typy, které RVSP používá, jsou zprávy PATH a RESV. Zpráva PATH je vysílána od odesílatele k příjemci a zpráva RESV od příjemce k odesílateli. Komunikace je vidět na obrázku 3.2. Odesílatel je zde myšlen jako stanice, který poskytuje službu. Zpráva PATH obsahuje informace, které popisují vlastnosti datového toku TSpec (Traffic Specification), který generuje odesílatel.

V TSpec je vymezeno potřebné pásmo, zpoždění a kolísání zpoždění. Tato informace se periodicky vysílá pomocí jednosměrného nebo skupinového vysílání. Tímto způsobem jsou všichni uživatelé informováni o parametrech datového toku dané služby. Při průchodu zprávy PATH síť si každý uzel zaznamená zdrojovou adresu odesílatele PATH. Tato informace je důležitá pro směrování zprávy RESV po stejné cestě jako PATH, ale v opačném směru.

Pokud bude chtít příjemce provést rezervaci, vygeneruje pomocí TSpec parametrů ze zprávy PATH žádost o rezervaci RESV. RESV obsahuje položky RSpec (Reservation specification) a filtrovací informace. RSpec udává typ integrované služby (garantovaná nebo s řízenou zátěží). Filtrovací informace udávají, které pakety mohou uskutečnit rezervaci. Filtrovací informace a RSpec dohromady tvoří identifikátor datového toku, který používají směrovače k identifikaci rezervací.

Příjemce odešle RESV cestou, kterou přijal zprávu PATH. První směrovač přijme zprávu a pokusí se provést rezervaci a kontrolu přístupu. Pokud jsou obě podmínky splněny, odešle RESV zprávu dalšímu směrovači. Pokud dojde k úspěšné registraci na všech směrovačích, je příjemci odeslána potvrzovací zpráva. Rezervace přenosového pásma je ukončena zprávou Reservation Teardown. Aby byla zpráva směrovačem přijata, musí být uvedeny platné informace o datovém toku, jinak bude požadavek zamítnut.



Obr. 3.2: Komunikace protokolu RSVP

3.2. DIFERENCOVANÉ SLUŽBY DIFFSERV

Na rozdíl od IntServ služba DiffServ neprovádí rezervaci přenosového pásma pro daný datový tok, ale pakety jednotlivých datových toků jsou klasifikovány do tříd a podle příslušnosti do tříd je s pakety zacházeno, aby byla zajištěna potřebná kvalita služeb. Pakety jsou označovány při vstupu do sítě. Během průchodu paketů uvnitř sítě směrovače čtou značky a podle nich je s pakety zacházeno.

DiffServ rozděluje síť do menších organizačních jednotek, které jsou nazývány Diffserv domény (dále DS). Každá doména obsahuje směrovače, které

mají společné nastavení kvality služeb a mají společné nastavení požadovaného způsobu zacházení PHB (Per Hop Behaviour). PBH bude popsána v další kapitole. [25]

V každé DS jsou dva druhy směrovačů:

- Hraniční – na okraji domény, umožňují propojení s jinými doménami
 - Příchozí (Ingress) – značkování paketů z jiných domén
 - Odchozí (Egress) – značuje odchozí pakety z domény
- Vnitřní – zajišťují zpracování a přeposlání paketů na základě identifikátoru, při dodržení požadované kvality služby.

Hraniční uzly také zajišťují prosazování dohody o zpracování přenosu TCA (Traffic Conditioning Agreement). TCA popisuje parametry pro jednotlivé úrovně služby. [20] TCA popisuje následující parametry:

- Popis parametrů přenosové sítě
 - propustnost
 - zpoždění
 - kolísání zpoždění
 - ztrátovost paketů
- Profily provozu (parametry měřičů)
 - průměrná rychlost
 - maximální okamžitá rychlost
 - maximální velikost tolerovaného shluku
- Způsob zpracování paketů překračující sjednané parametry [20] [5]

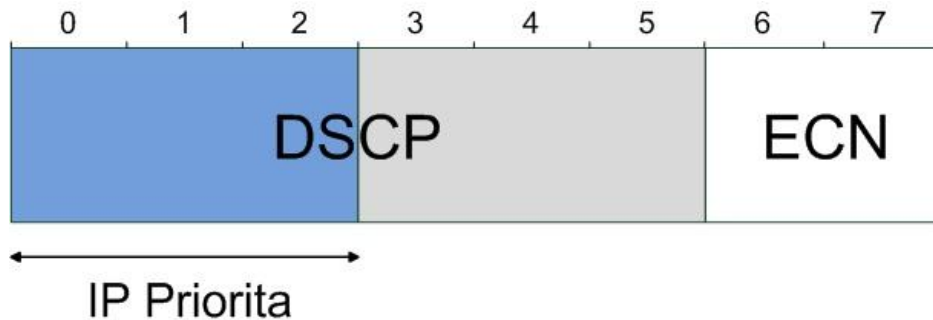
TCA je součástí dohody o úrovni služby SLA (Service Level Agreement). SLA je dohoda mezi poskytovatelem služby a klientem. V SLA je specifikováno, které parametry služby jsou garantovány poskytovatelem. Jsou zde také popsány situace, pokud dojde k nedodržení těchto parametrů. TCA obsahuje všechna pravidla pro klasifikaci, značkování, měření, tvarování, zahazování, která jsou aplikována na příchozí datové toky. V TCA jsou také popsány datové profily. [2]

3.2.1. KLASIFIKACE A ZNAČKOVÁNÍ PAKETŮ

Klasifikaci a značkování příchozích paketů provádí hraniční směrovače. Klasifikace spočívá ve sdružování paketů do skupin na základě informací z hlavičky IP paketu. [20] Existují dva druhy klasifikace:

- Souhrnné chování (Behaviour Aggregate - BA) – pracuje pouze s identifikátory DSCP. Používá se v případě, že paket procházející směrovačem je již označen. Klasifikátor sdružuje pakety se stejným DSCP, které pocházejí z určitého zdroje do jednoho toku. Zajišťuje uplatnění stejného PHB na pakety se stejným DSCP.
- Klasifikace na základě více polí MF (Multi-field)- pracuje s informacemi ze záhlaví paketu, jako zdrojová nebo cílová adresa, čísla portů atd. [2]

Ke značkování se používá šestibitový identifikátor DSCP (Differentiated Services Code Point). U paketu protokolu IPv4 je využit oktet s názvem Typ služby TOS (Type of Service). U IPv6 je to oktet pole Třída provozu (Traffic Class). Struktura pole DS je vidět na obr. 3.3.



Obr. 3.3: Struktura pole DS

Součástí DSCP je 3 bitový identifikátor IP priorita IPP (IP Precedence), který je zaveden kvůli zpětné kompatibilitě. IPP bylo použito v prvotní implementaci IPv4 protokolu pro rozdělení do základních tříd. Dva bity ECN se používají pro oznámení přetížení linky.

3.2.2. ZPŮSOB ZACHÁZENÍ S PAKETY (PER HOP BEHAVIOR - PHB)

Samotná klasifikace a označení paketů ještě nezajistí požadovanou kvalitu služeb. Je potřeba definovat, jak budou směrovače jednotlivé pakety zpracovávat a přeposílat. Je třeba zajistit upřednostnění určitých tříd před ostatními a také efektivní přidělování síťových prostředků. U mechanismu DiffServ je toto zajištěno metodou, která se nazývá způsob zacházení s pakety. PHB popisuje, jak se bude ve směrovačích s datovými jednotkami zacházeno, jaké jim bude přiděleno pásmo, jakou budou mít prioritu při zpracování ve frontách atd. PHB souvisí s plánováním odesílání paketů, s tvarováním a zahazováním provozu a s měřením provozu. PHB není žádný obecný standard, jedná se o doporučení podle kterých je možné požadovaných způsobů zacházení s pakety dosáhnout. Proto si implementaci PHB do směrovačů výrobci zajišťují sami.

Aplikace potřebného PHB se je řízeno na základě identifikátoru DSCP. Každý směrovač obsahuje tabulku, kde jsou jednotlivé PHB mapovány na identifikátory DSCP. [2]

Současně se používají čtyři základní mechanismy: Základní způsob zacházení, způsob zacházení s výběrem třídy, urychlený přenos a zajištěný přenos.

Základní způsob zacházení (default PHB)

Využívá se pro pakety, které jsou označeny identifikátorem DSCP „000000“ nebo pro pakety které nebyly přiřazeny do žádné třídy provozu. Směrovač se bude snažit, doručit tyto pakety s největším úsilím “best-effort“ a nebude garantovat žádné parametry pro zpracování. Tento mechanismus musí umět všechny směrovače v DiffServ doméně. [7]

Způsob zacházení s výběrem třídy (Class-Selector PHBs)

Tento druh PHB je definován pro zachování zpětné kompatibility se schématem IPP.

Urychlený přenos (Expedited Forwarding – EF)

Jedná se o nejvyšší třídu přenosu, která je vhodná pro aplikace s menším, ale konstantním datovým tokem, s požadavkem na nízké zpoždění a kolísání zpoždění. Pro splnění těchto požadavků EF rezervuje určitou velikost přenosového pásma výstupního rozhraní směrovače. Dále je nutné mít pro tento typ vyhrazenou prioritní frontu, která zajistí přednostní odbavování paketů. V této třídě by měli být pouze služby, které jsou velmi náročné na zpoždění a kolísání zpoždění, aby bylo možné zajistit požadovanou kvalitu. Pokud by bylo v této třídě více služeb, mohlo by dojít k degradaci účelu této třídy, tím že by směrovač nebyl schopný prioritně obsloužit všechny služby. Urychlený přenos je vhodný pro videotelefonii, IP telefonii nebo emulaci telefonních okruhů. [7] [16]

Zajištěný přenos (Assured Forwarding – AF)

Je to ekvivalent služby s řízenou zátěží u IntServ. Zajištěný přenos AF je určen pro služby, pro které je důležitější spolehlivost přenosu než zpoždění. PHB AF definuje 4 třídy, které umožňují nastavení chování směrovače při zpracování paketu pro různé typy služeb. Každá třída má přidělenou určitou kvótu vyrovnávací paměti a šířku pásma. V každé třídě je možné definovat priority zahození, podle kterých se bude směrovač řídit, pokud dojde k zahlcení linky a bude třeba nějaký paket zahodit. Určování této priority se děje na základě měření provozu. Pokud paket (datový tok) splňuje rychlostní limit, je umístěn do třídy s nejnižší pravděpodobností zahození. Naopak pokud paket (datový tok) překračuje i krátkodobé rychlostní limity je paket zařazen do třídy s největší pravděpodobností zahození. Třídy jsou označeny řetězcem „AFxy“, kde „x“ udává číslo třídy a „y“ udává podtřídou priority zahození. Kompletní tabulku tříd a priorit zahození je vidět v tabulce 3.1. [14] [20] [7]

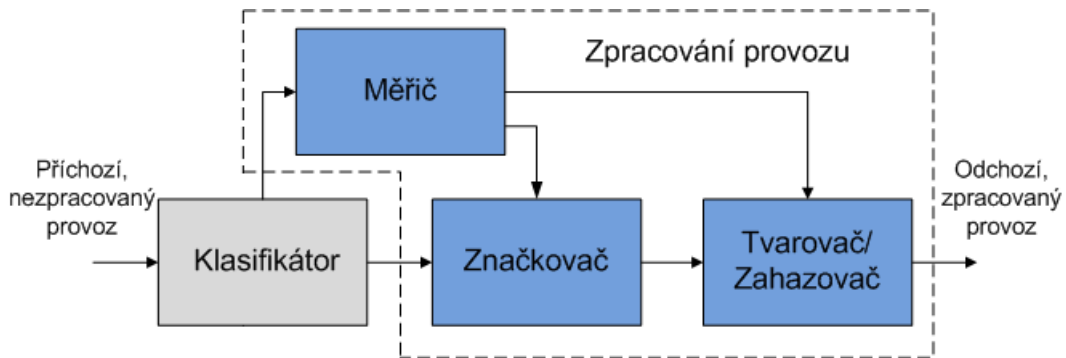
Tab. 3.1: Značení tříd u zajištěného přenosu AF

Priorita zahození	Třída 1	Třída 2	Třída 3	Třída 4
Nízká priorita zahození	AF11	AF21	AF31	AF41
	001010	010010	011010	100010
Střední priorita zahození	AF12	AF22	AF32	AF42
	001100	010100	011100	100100
Vysoká priorita zahození	AF13	AF23	AF33	AF43
	001110	010110	011110	100110

3.2.3. ZPRACOVÁNÍ PROVOZU (TRAFFIC CONDITIONING)

Zpracování provozu je téměř nejdůležitější součástí DiffServ. Zpracování provozu zahrnuje měření, značení, tvarování a zahazování provozu. V této funkční části je možné provádět přeznačování paketů, tvarovat provoz tak, aby splnil nastavený profil provozu. Blokované schéma je vidět na obrázku 3.4. Zpracování provozu se

obvykle provádí na hraničních směrovačích. Někdy může být i ve vnitřním směrovači.



Obr. 3.4: Blokové schéma zpracování provozu

Pravidla, kterými se zpracování provozu řídí, jsou uvedena v TCA. Zpracování provozu zajišťuje, aby se datové toky vstupující do DiffServ domény přizpůsobili pravidlům v TCA a byli v souladu s politikami o poskytnutí služby (Service Provisioning Policy). Tyto politiky definují, jak je zpracování provozu nastaveno na hraničních směrovačích. [15]

Zpracování provozu je závislé na profilu provozu. Profil provozu udává dočasné vlastnosti datového toku, která zjištěna měřičem. Paket je následně zpracován podle toho, zda profil splňuje (in-profile) anebo nesplňuje (out-of-profile). Pokud paket profil splňuje, může vstoupit do domény bez dalšího úprav. Pokud doména používá jiné identifikátory DSCP a jiné PHB může dojít ke změně DSCP. Pokud pakety profil nesplňují, jsou ukládány do front, po dobu než budou profil splňovat. Případně může dojít k zahazení, nebo ke změně DSCP. [2]

Funkce zpracování provozu jsou zajištěny čtyřmi základními součástmi:

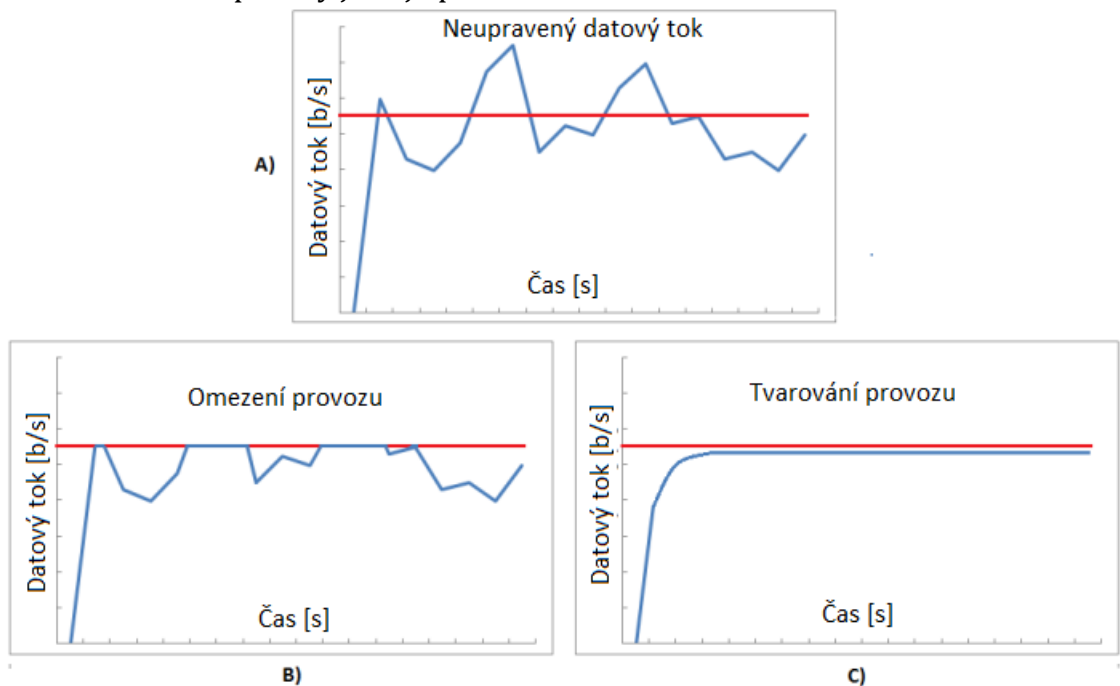
- Měřič (Meter) – měřič provozu přijímá pakety od klasifikátoru a měří dočasné hodnoty příchozího datového toku a srovnává je s příslušným profilem provozu. Měřič vyhodnocuje, zda datový tok nastavený profil splňuje či ne. Tuto informaci přikládá k paketu a tím informuje další součásti, jak s paketem zacházet.
- Značkovač (Marker) – značkovač zajišťuje označení paketu identifikátorem DSCP.
- Tvarovač (Shaper) – zajišťuje zpoždování některých nebo všech paketů, případně jejich zahazování, aby byl splněn příslušný profil provozu. Tvarovač ukládá pakety do vyrovnávací paměti s konečnou velikostí. Pokud dojde k obsazení kapacity vyrovnávací paměti, pakety jsou zahazovány.
- Zahazovač (Dropper)- speciální typ tvarovače, který nemůže ukládat pakety do fronty.

3.2.4. TVAROVÁNÍ PROVOZU (SHAPING, POLICING)

Tvarování provozu slouží k omezení přenosového pásma pro určitý datový tok. Zajišťuje lepší využití přenosového pásma linky. K tvarování provozu jsou využívány dvě metody, které plní stejnou funkci, ale každá používá jiný postup.

- Omezení na základě tříd (Class-Based Policing) – zajišťuje zahození paketů, které překročí nastavenou hranici přenosového pásma.
- Tvarování na základě tříd (Class-Based Shaping) – u této metody nedochází k zahazování paketů, ale pakety jsou zpoždovány ukládáním do front.

Rozdíl mezi oběma způsoby je nejlépe vidět na obrázku 3.5.



Obr. 3.5: a) Graf neupraveného datového toku b) Graf omezeného datového toku
c) Graf tvarovaného datového toku

3.2.5. OMEZENÍ NA ZÁKLADĚ TŘÍD (CLASS-BASED POLICING)

Omezení na základě tříd řídí přenosovou rychlost na rozhraní směrovače. Pokud dojde k překročení nastaveného pásma, omezovač provozu začne zahazovat pakety. K omezování se využívá algoritmus Token Bucket. (dále pouze TB).

TB pracuje s tzv. „kbelíkem“, který se plní tokeny. Kbelík má určitou velikost, která je dána parametrem Burst Size. Kbelík je neustále doplňován rychlostí, kterou určuje parametr Burst size nebo Average Traffic Rate. Tokeny určují, kolik bajtů paketu je možné odeslat. Většinou odpovídá jeden token jednomu bajtu. Pokud dojde k naplnění kbelíku, jsou další tokeny zahazovány. Když dorazí paket, TB algoritmus zkontroluje, zda je k dispozici dostatečný počet tokenů. Pokud je k dispozici potřebný počet tokenů, paket je předán k dalšímu zpracování. Pokud je v kbelíku málo tokenů, paket může být zahozen nebo uložen do fronty dokud nebude v kbelíku dostatečný počet tokenů.

4. OPNET MODELER

Program Opnet Modeler (dále OM) je simulační prostředí, které bylo vyvinuto firmou Opnet Technologies Inc. OM slouží pro návrh, simulaci a analýzu síťových technologií a mechanismů. [21] Pomocí tohoto nástroje jsme schopni simulovat množství topologií, standardů, protokolů a aplikací. Nejsme omezeni pouze modely, které jsou po instalaci k dispozici, ale můžeme instalovat nově vytvořené modely pro OM.

OM se ovládá pomocí interaktivního grafického prostředí, kdy pomocí myši na plochu vkládáme objekty, které pomocí modelů simulující přenosová media propojujeme. Důležitou funkcí je duplikace scénářů, pomocí které se snadno srovnávají výsledky simulací s odlišnými prvky nebo nastaveními. Výstupem z OM je vykreslení charakteristik, směrovací tabulky atd. Naměřená data můžeme exportovat např. do aplikace Microsoft Excel. Použití OM je výhodné pro simulaci sítí, které by bylo finančně náročné fyzicky sestavit. Nebo naopak, můžeme využít teoretických výsledků z OM při výstavbě reálné sítě. Můžeme simulovat extrémní situace, jako jsou přetížení serverů, nebo jeho výpadky a těmito situacím předcházet.

4.1. ZÁKLADNÍ PRVKY PROSTŘEDÍ OPNET MODELER

- **Podsít' (Subnet)** - obsahuje stanice, směrovače, firewally, přenosová media a další síťové komponenty
- **Model uzlu (Node model)** - složený ze základních funkčních bloků jako zdroj, paměti, procesor [21]
- **Model procesu (Process model)** - zde jsou popsány procesy modelu uzlu, jako např. stavy procesu, události, přechody [21]

4.2. EDITORY

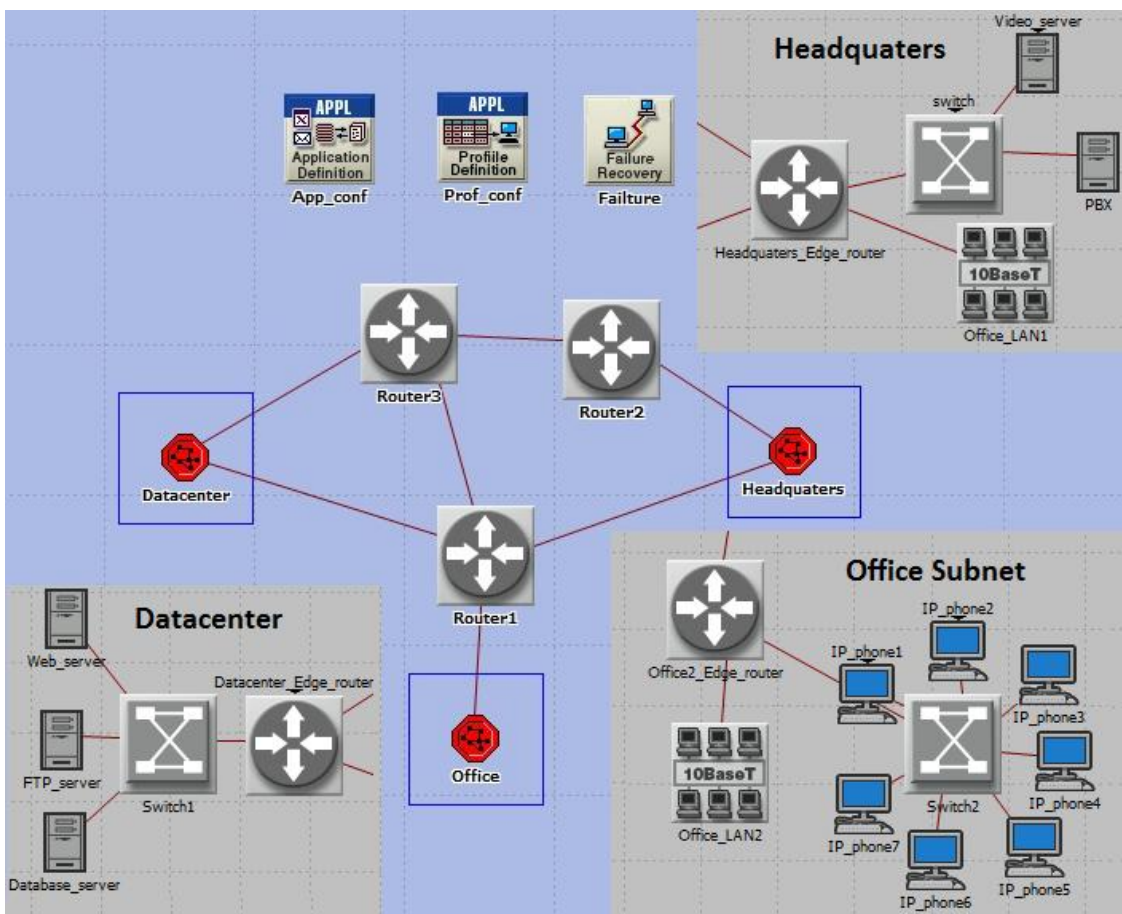
Struktura OM je rozložena do tří vrstev. Těmito vrstvám se říká editory:

- **Editor projektu (Project Editor)**- hlavní grafický editor umožňující vytváření topologií, pomocí definovaných modelů. Můžeme zde generovat různé druhy zátěží, pomocí základních aplikací nebo vytvářet vlastní.
- **Editor uzlu (Node Editor)** - umožňuje pohled na vnitřní uspořádání síťového zařízení nebo systému a naznačuje vazby mezi jednotlivými procesy a funkcemi.
- **Editor procesu (Process Editor)** - každý stav a proces modelu obsahuje kód v C/C++ podporovaný rozsáhlou knihovnou s funkcemi vytvořenými pro protokolové programování. [21]

5. SIMULACE V PROSTŘEDÍ OPNET MODELER

V této části diplomové práce je vytvořena experimentální síť. Vytvořená topologie je použita k simulaci několika scénářů, které umožnily srovnání vlastností směrovacích protokolů a zjištění vlivu QoS na vlastnosti provozu v síti.

Experimentální síť, kterou můžeme vidět na obrázku 5.1, napodobuje firemní síť. Na obrázku jsou v rozích zobrazeny jednotlivé podsítě. Topologii tvoří tři podsítě, které jsou spojeny páteří sítí. Podsít' Headquarters je ústředí firmy, která je pomocí technologie 10BaseT spojena s další pobočkou (Office) a také s datacentrem (Datacenter).



Obr 5.1: Topologie simulované sítě

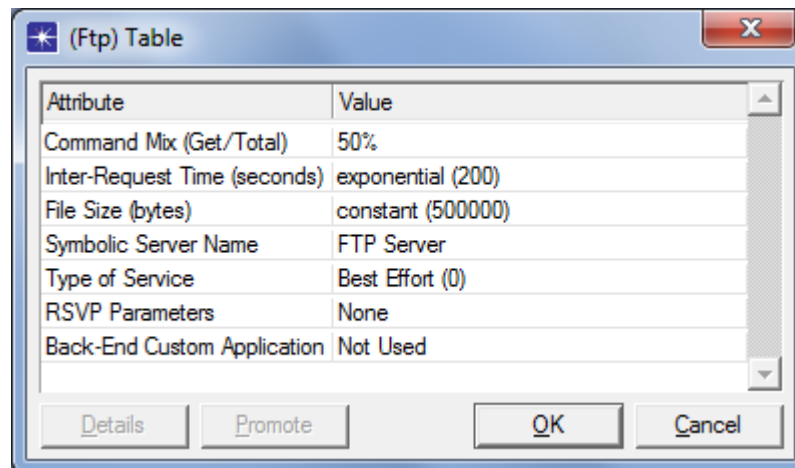
V podsíti Datacenter je umístěn webový server, FTP server a databázový server. V podsíti Headquarters je umístěna síť LAN (Office_LAN1) o 30 stanicích a pobočková ústředna PBX. Podsít' Headquarters a Office komunikují se servery v datacentru. Provoz je tvořen aplikacemi HTTP, FTP a komunikací s databází. Uživatelé v LAN sítích v obou podsítích (Headquarters a Office) mezi sebou komunikují pomocí IP telefonie.

5.1. NASTAVENÍ APLIKACÍ

V projektu je vložen objekt **Application Config**, který je pojmenovaný **App_conf**. Tento model slouží k nastavení používaných aplikací. Je možné použít defaultně nastavené aplikace nebo vytvořit vlastní. V simulaci budou využity čtyři aplikace: HTTP, FTP, databáze a VoIP.

5.1.1. NASTAVENÍ APLIKACE FTP

Aplikace FTP je v modelu **App_conf** pojmenována **FTP**. Nastavení aplikace je vidět na obrázku 5.2. Nastavení je provedeno editací atributu ftp v položce **Description**.



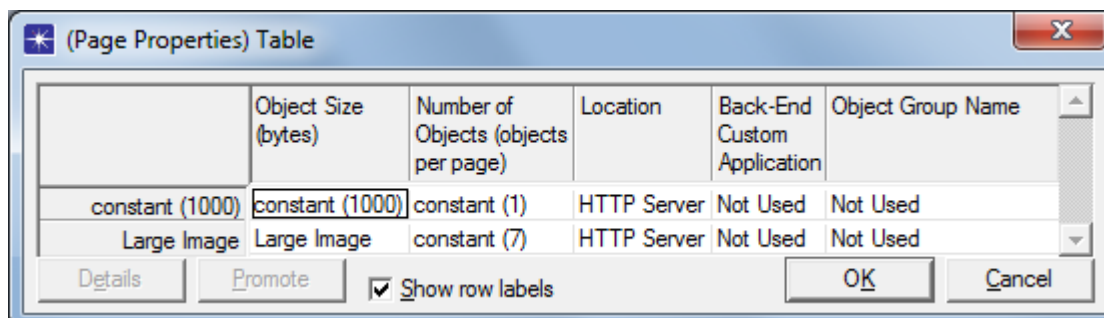
Attribute	Value
Command Mix (Get/Total)	50%
Inter-Request Time (seconds)	exponential (200)
File Size (bytes)	constant (500000)
Symbolic Server Name	FTP Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

Obr. 5.2: Nastavení aplikace FTP

- **Command mix (Get/total)** – parametr udává poměr stažených a poslaných souborů.
- **Inter-Request Time** – parametr udává časový mezi jednotlivými požadavky
- **File size** – parametr udává velikost stahovaného/posílaného souboru

5.1.2. NASTAVENÍ APLIKACE HTTP

Postup je stejný jako u **FTP** pouze v položce **Description** je editován řádek http. U **HTTP** jsou nastaveny **Page Properties** a **Server Selection**. Nastavení parametru **Page Properties** je na obrázku 5.3.

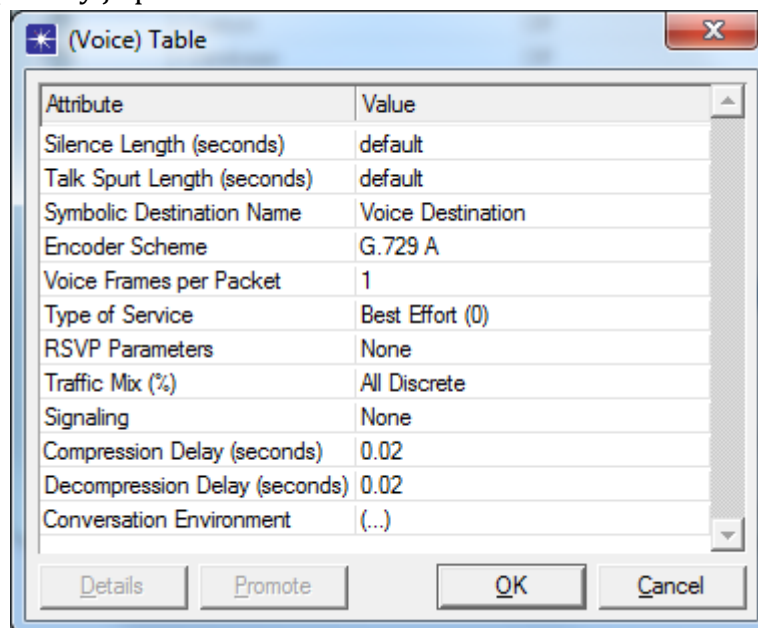


Obr. 5.3: Nastavení parametru Page Properties

- **Page Interval Time** – doba mezi jednotlivými požadavky na webový server
- **HTTP Version** – verze HTTP protokolu

5.1.3. NASTAVENÍ APLIKACE VOIP

Nastavení aplikace **VoIP** je vidět na obrázku 5.4. Pro kódování hlasového signálu do digitální podoby je použito kódovací schéma G.729A.



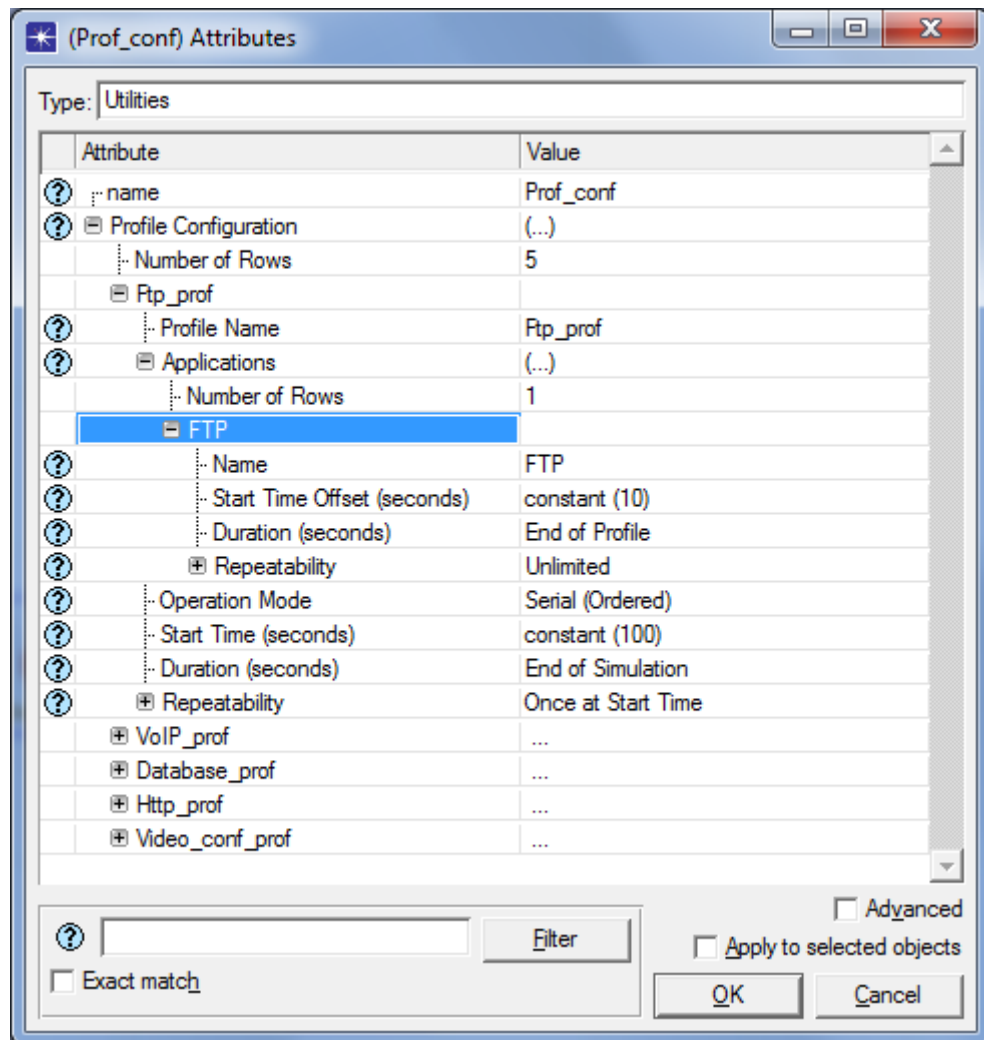
Obr. 5.4: Nastavení aplikace VoIP

5.2. NASTAVENÍ PROFILU APLIKACÍ

Pomocí tohoto modelu je nastaveno, kdy se jaká aplikace bude spouštět. Model umožňuje nastavení parametrů, které ovlivní, jak dlouho je aplikace spuštěna nebo v jakém pořadí jsou aplikace spuštěny. V této simulaci je pro každou aplikaci použit jeden profil. Model **Profile Config** je pojmenován jako **Prof_conf**. Na obrázku 5.5 je vidět nastavení profilu pro aplikaci **FTP**. Nejdůležitější je parametr **Start Time**, který udává v jakém čase po začátku simulace je profil spuštěn. Všechny spuštěné profily běží až do konce simulace. Tato vlastnost je ovlivněna parametrem **Duration**.

Tab. 5.1: Nastavení parametru

Profil	FTP_prof	HTTP_prof	Database_prof	VoIP_prof
Start Time [s]	100	160	180	120



Obr. 5.5: Nastavení profilu pro aplikaci FTP

5.3. NASTAVENÍ PODPORY APLIKACÍ A PROFILŮ VE SCÉNÁŘI

Při nastavování podpory aplikací a profilů jsou důležité tyto parametry:

- **Application: Destination Preferences** - provádí mapování symbolických jmen cíle, které jsou definovány v **Application Config**, se jmény klientů (Client name) nebo adresami (Server Address)
- **Application: Supported Profiles** - seznam profilů, které budou na tomto objektu spuštěny
- **Application: Supported Services** - seznam aplikací, které budou tímto objektem podporovány

Objekt, který je zdrojem dat (zpravidla server), musí mít v parametru **Application: Supported Services** nastavenou službu kterou bude podporovat.

Objekt, který bude příjemcem dat (zpravidla stanice) musí mít v parametru **Application: Supported Profiles** nastaven podporovaný profil. Pokud je v síti více zdrojů stejné aplikace parametrem **Application: Destination Preferences** nastavíme konkrétní zdroj. Nastavení serverů a stanic ve scénáři je popsáno v tabulce 5.2.

Tab. 5.2: Nastavení podpory aplikací a profilů v jednotlivých podsítích

Podsít' Datacenter			
Parametr/Objekt	Web_Server	FTP_server	Database_server
Supported Services	HTTP	FTP	Database
Supported Profiles	-	-	-
Destination Preferences	-	-	-

Podsít' Office		
Parametr/Objekt	Office_LAN2	IP_phone1-7
Supported Services	-	-
Supported Profiles	FTP_prof, HTTP_prof, Database_prof	VoIP_prof
Destination Preferences	-	PBX

Podsít' Headquarters		
Parametr/Objekt	Office_LAN2	PBX
Supported Services	-	VoIP
Supported Profiles	FTP_prof, HTTP_prof, Database_prof	-
Destination Preferences	-	-

5.4. NASTAVENÍ SCÉNÁŘE S PROTOKOLEM RIP A DIFFSERV

V tomto scénáři je v síti nastaveno zajištění kvality služeb QoS. QoS je realizováno pomocí mechanismu Diffserv. Celá topologie tvoří jednu Diffserv doménu, kdy okrajové směrovače v jednotlivých podsítích tvoří příchozí (Ingress) směrovače. Na těchto směrovačích probíhá identifikace, klasifikace a značkování datových toků. Pakety jsou značeny pomocí identifikátorů DSCP. Většina nastavení Diffserv je provedena na okrajových směrovačích. Konkrétně na směrovačích **Office2_Edge_router**, **Headquarters_Edge_Router**, **Datacenter_Edge_Router**. Směrovače, které jsou v simulaci použity, mají defaultně nastavený RIP protokol.

5.4.1. NASTAVENÍ IDENTIFIKACE PAKETŮ NA ZÁKLADĚ ACL

Identifikace datových toků probíhá na základě seznamu řízení přístupu (ACL – Access Control List). V simulaci jsou využity rozšířené ACL, které umožňují filtrování na základě protokolu a portu. Na základě těchto seznamů je možné rozlišit pakety jednotlivých aplikací. Jakmile směrovač určí aplikaci, jejíž data jsou v paketu obsažena, může paketu přiřadit příslušné identifikátor DSCP.

Nastavení ACL je provedeno kliknutím na **Edit Attributes - IP - IP Routing Params - Extended ACL Configuration - Edit**. Každé pravidlo musí mít

v položce **Action** nastaveno **Permit**. Posledním nastavením je nastavení zdrojových a cílových portů u každého pravidla. Porty pro jednotlivé aplikace jsou vidět v tabulce 5.3. Pravidlo pro **VoIP** je filtrováno pouze na základě UDP protokolu.

Tab. 5.3: Čísla portů aplikací

Aplikace	HTTP	FTP	Databáze	VoIP
Port	80	20,21	101	-

5.4.2. NASTAVENÍ TŘÍD PROVOZU

Pakety příchozí do DiffServ domény jsou na základě ACL listů klasifikovány do tříd provozu. Pro každou aplikaci je vytvořena jedna třída. Ke každé třídě přiřazeno ACL pravidlo, které pokud bude splněno, paket bude do této třídy klasifikován.

Nastavení je provedeno v **IP – IP QoS Parametrů – Traffic Class**. Nastaveny jsou čtyři třídy. U každé třídy je nutné v **Match Info**, nastavit pravidlo podle kterého bude výběr třídy proveden. Nastavením **Match Property** je docíleno, že klasifikace bude probíhat pomocí ACL. V **Match Value** vybereme ACL pravidlo pro příslušnou aplikaci.

5.4.3. NASTAVENÍ POLITIK PŘÍCHOZÍHO PROVOZU

V této části je vytvořeným třídám provozu přidělen identifikátor DSCP, který odpovídá příslušnému způsobu zacházení s pakety. Pakety třídy pro VoIP jsou označeny identifikátorem EF, který reprezentuje urychlený přenos. Pakety aplikace FTP budou označeny identifikátorem BE, který zajistí, že s pakety bude zacházeno mechanismem best-effort. Nastavení je provedeno na všech okrajových směrovačích **Office2_Edge_router**, **Headquarters_Edge_Router**, **Datacenter_Edge_Router**.

Nastavení je provedeno v **IP – IP QoS Parametrů – Traffic Policies**. V simulaci je vytvořena jedna politika **Ingress_Traffic_policy**. V nastavení politiky jsou vytvořeny čtyři záznamy, které slouží pro přiřazení DSCP třídám vytvořeným v minulé části. V tabulce 5.4 jsou vidět identifikátory DSCP pro třídy provozu. Pro každou třídu je v položce **Set Info** nastaveno **Rows** na jedna a **Set Property** na hodnotu DSCP. Hodnota **Set Value** je nastavena podle tabulky 5.4.

Tab. 5.4: Čísla portů aplikací

Třída provozu	VoIP_class	HTTP_class	FTP_class	Database_class
DSCP	EF	AF21	BE	AF21

Nově vytvořenou politiku **Ingress_Traffic_policy** je třeba nastavit všem rozhraním na okrajových směrovačích. Nastavení rozhraní provedeme v **IP – IP QoS Parametrů – Interface Information**. Pro každé rozhraní **IFx** nastavíme vlastnosti QoS. Nastavení je provedeno vytvořením dvou položek v **QoS scheme**. První položka udává použitý QoS profil. V simulaci je použit systém front

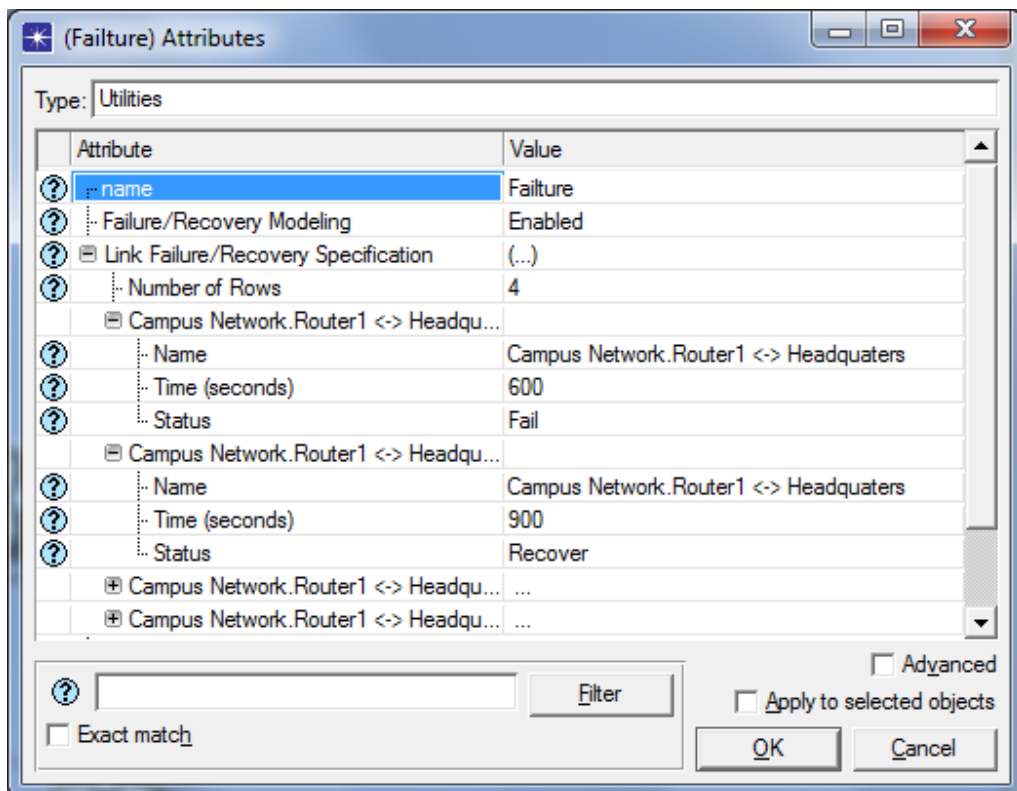
s váženou spravedlivou obsluhou (WQF – Weighted Fair Queuing). WQF zajišťuje dělení příchozího provozu do front, které mají jinou váhovou hodnotu. Tato hodnota, pak určuje šířku pásma datového toku, kterou fronta může obsloužit. Pakety jsou do front řazeny podle DSCP. [20]

První položka obsahuje: **Type – WFQ (Class Based)** a **Name – DSCP Based**. Druhá položka obsahuje: **Type – Inbound Traffic Policy** a **Name – Ingress_Traffic_policy**. Nastavení politik vnitřních směrovačů je provedeno podobně jako u vstupních, avšak obsahuje pouze nastavení QoS profilu na použití systému front s váženou spravedlivou obsluhou WQF na základě DSCP.

5.4.4. NASTAVENÍ VÝPADKU LINKY

Ve všech scénářích je nasimulován výpadek primární linky mezi směrovačem **Router1** a podsítí **Headquaters**. Provoz linky je následně přesměrován na linku záložní. Na záložní lince je nakonfigurován provoz v pozadí. V simulaci jsou dva výpadky. První výpadek nastane v šesté minutě a trvá pět minut. Druhý výpadek nastane ve 40. minutě a trvá tři minuty.

Do scénáře je z palety objektu z **Utilities** vložen objekt **Failure Recovery**. V nastavení výpadku je provedeno v položce **Link Failure/Recovery Specification**. Nastavení výpadku hlavní linky je na obrázku 5.6.



Obr. 5.6: Nastavení výpadku linky mezi Router1 - Headquaters

5.5. NASTAVENÍ SCÉNÁŘE S PROTOKOLEM OSPF

Pro simulaci protokolu OSPF jsou vytvořeny dva scénáře, duplikací scénáře RIP s použitím best-effort a RIP s nastavením Diffserv. Všechny směrovače v obou

scénářích nastavíme na používání OSPF protokolu. Všechny směrovače jsou nakonfigurovány v jedné oblasti, která má ID 0.0.0.0. Metriky jednotlivých linek jsou nakonfigurovány manuálně.

Nastavení je provedeno vybráním všech směrovačů, a kliknutím na menu **Protocols - IP - Routing - Configure Routing Protocols**. Zde je vybrán OSPF protokol. Nastavení metrik linek je provedeno v menu **Protocols - OSPF - Configure Interface Cost**. Primární linky mezi podsítěmi, konkrétně mezi směrovači **Datacenter_Edge_router, Router1** a **Headquarters_Edge_router** mají nastavenou metriku na 10. Záložní linka bude mít vyšší metriku, aby přes ni data tekly pouze v případě výpadku.

5.6. NASTAVENÍ CHARAKTERISTIK A PARAMETRŮ SIMULACE

Poslední krokem je nastavení parametrů simulace a toho jaké statistiky se budou vykreslovat. Nastavení charakteristik se provádí kliknutím kamkoli do projektu pravým tlačítkem a vybráním **Choose Individual DES Statistic**.

Sledované statistiky:

- Globální
 - DB Query – Response Time
 - FTP – Download Response Time
 - FTP – Upload Response Time
 - HTTP – Object Response Time
 - HTTP – Page Response Time
 - IP – Network Convergence Duration
 - IP - Traffic Dropped
 - Voice – Packet Delay Variation
 - Voice – Packet End-to-End Delay
 - OSPF – Traffic Send
 - RIP – Traffic Send
- Uzlové
 - Router1 – Datacenter - Throughput
 - Router2 – Router3 - Throughput

Nastavení parametrů simulace je provedeno po kliknutí na **Configure/Run Discrete Event Simulation**. Délka trvání simulace (Duration) je jedna hodina. Hodnota **Kernel Simulation** je nastavena na hodnotu **Optimized**.

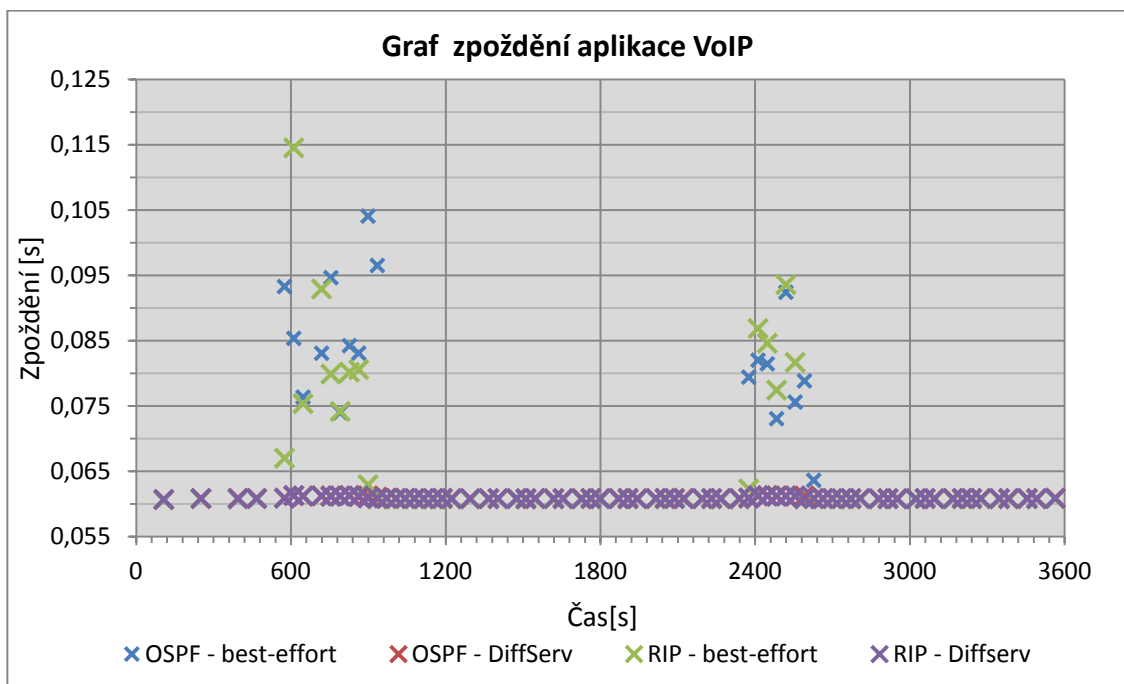
V položce **Input – Global Attributes - IP - IP Dynamic Routing Protocols** je třeba nastavit, aby byl použit směrovací protokol, který je ve scénáři nastaven. V položce **Input – Global Attributes - IP - Simulation Efficiency** je třeba vypnout parametr **SIM efficiency** pro příslušný směrovací protokol. Tento parametr slouží k zvýšení rychlosti simulace, tím že po nastavené době vypne směrovací protokol.

5.7. VÝSLEDKY SIMULACE

Všechny naměřené statistiky je možné vyvolat kliknutím do projektu a vybráním **View Results**. V simulaci jsou vytvořeny čtyři scénáře. První dva scénáře směřují pakety pomocí RIP a OSPF protokolu a používají mechanismus best-effort. V dalších dvou scénářích je v síti nakonfigurována kvalita služeb pomocí mechanismu DiffServ. Ve všech scénářích jsou nastaveny výpadky primární linky, které umožní pozorovat, jak rychle dokáže směrovací protokol reagovat na výpadky a jaký vliv bude mít nastavení kvality služeb na přenosové parametry. V následujících podkapitolách budou popsány naměřené grafy jednotlivých aplikací.

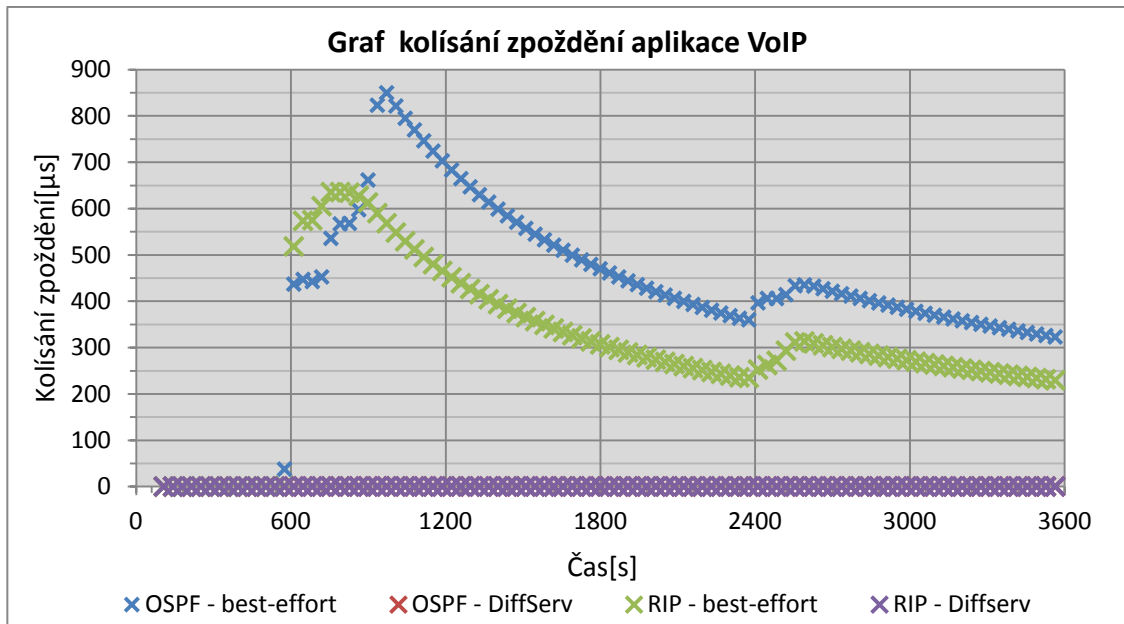
5.7.1. VÝSLEDKY SIMULACE - APLIKACE VOIP

IP telefonie je služba, která je velice citlivá na zpoždění a na kolísání zpoždění. Zpoždění by nemělo přesahovat hodnotu 300ms. VoIP je služba, která není náročná na přenosové pásmo, proto pro přenos nejsou nutné linky s vysokou přenosovou rychlostí. Problém nastane, pokud VoIP aplikace sdílí pásmo s jinými aplikacemi. Potom může zpoždění narůstat, což je nežádoucí a je třeba nastavit QoS. Na obrázku 5.7 je graf zpoždění VoIP paketů během simulace. V grafu jsou špatně vidět hodnoty pro scénář OSPF - DiffServ, které splývají s RIP - DiffServ.



k nárůstu zpoždění. U scénářů s QoS jsou VoIP pakety pomocí nastavení DiffServ upřednostňovány před ostatními a proto je zpoždění stále konstantní.

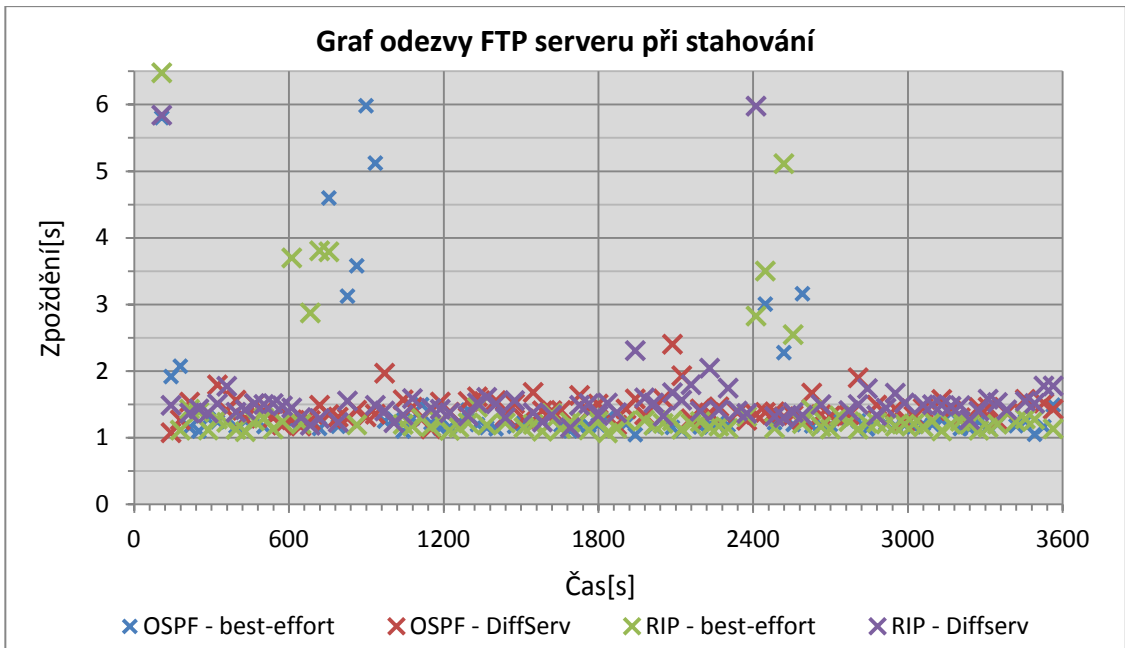
Dalším důležitým parametrem u VoIP je kolísání zpoždění (jitter). Kolísání zpoždění by nemělo přesáhnout hodnotu 50ms. Tento parametr není tak kritický jako zpoždění, protože je možné ho částečně kompenzovat vyrovnávací pamětí. Naměřenou závislost je vidět na obrázku 5.8. U scénářů s QoS je kolísání téměř nulové. Hodnota jitter u scénářů bez QoS nepřesáhla 1 ms.



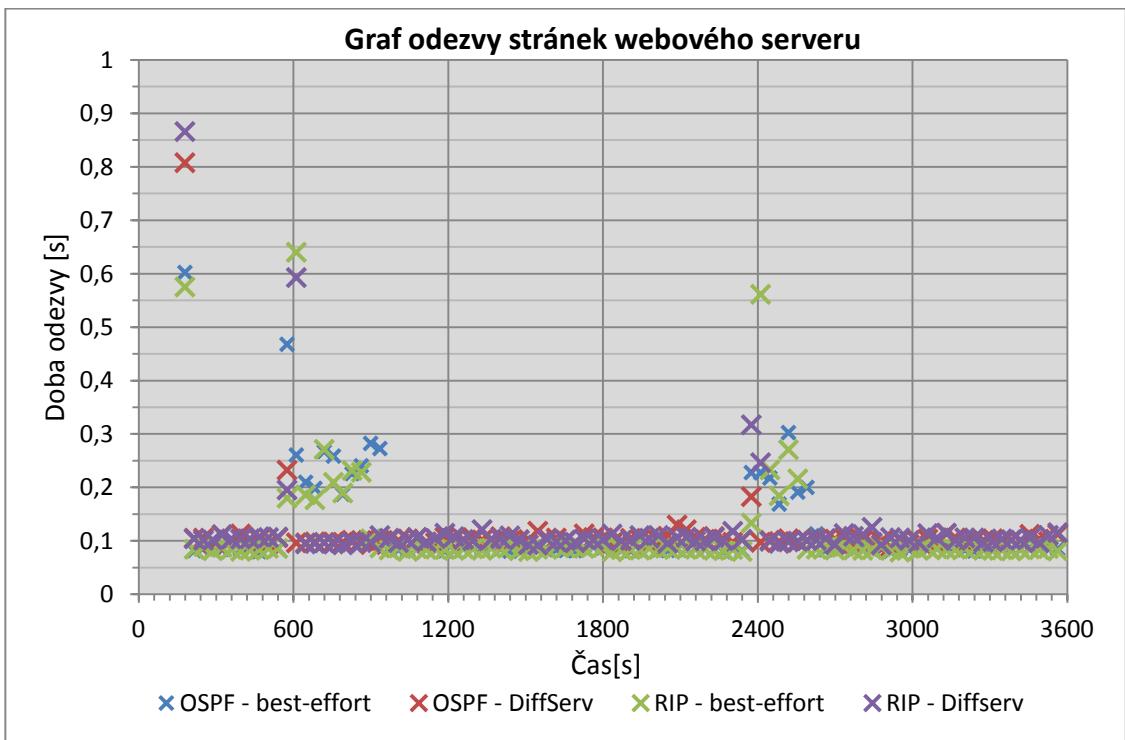
Obr. 5.8: Graf kolísání zpoždění aplikace VoIP

5.7.2. VÝSLEDKY SIMULACE – APLIKACE FTP A HTTP

Aplikace FTP slouží pro přenos data pomocí spolehlivého protokolu TCP. FTP není protokol citlivý na zpoždění nebo na kolísání zpoždění. Pro správné fungování je důležité, aby byla zaručena spolehlivost přenosu. Musí být zajištěna nulová ztrátovost na úkor vyššího zpoždění. Proto je při nastavování třídy provozu u DiffServ nastaveno zacházení způsobem best-effort. Z grafu na obrázku 5.9 je vidět, že odezva FTP serveru při stahování se pohybuje mezi jednou a dvěma sekundami. Během výpadku maximální hodnoty odezvy FTP serveru dosáhnou až šesti sekund. Z grafu je vidět, že ve scénářích s QoS je s FTP pakety zacházeno jinak než s pakety VoIP. FTP pakety mají díky odlišnému nastavení způsobu zpracování paketů nižší prioritu, a proto dochází ke kolísání zpoždění. Odezva FTP serveru je během normálního provozu nižší u scénářů bez QoS. Což odpovídá tomu, že aplikace FTP má ve scénářích s QoS nejnižší prioritu a přednostně jsou zpracovávány pakety aplikací VoIP, HTTP a databáze. Na obrázku 5.10 je vidět graf odezvy stránek webového serveru.



Obr. 5.9: Graf odezvy FTP serveru při stahování

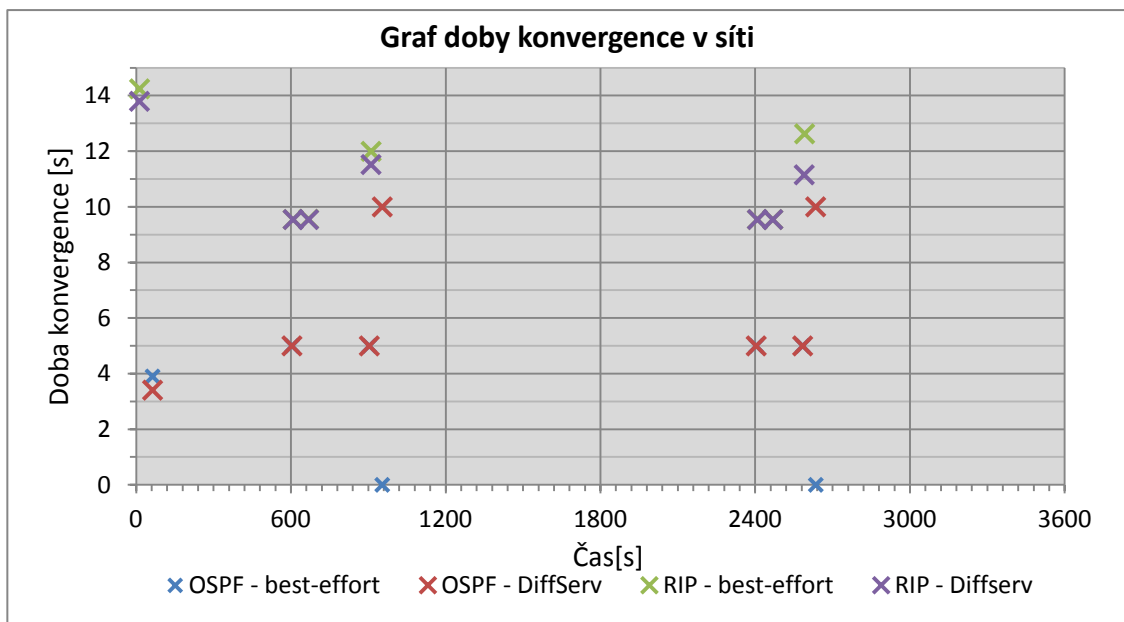


Obr. 5.10: Graf odezvy stránek webového serveru

5.7.3. VÝSLEDKY SIMULACE – SMĚROVACÍ PROTOKOLY

V projektu byly srovnány směrovací protokoly OSPF a RIP. Na obrázku 5.11 je vidět graf porovnávající rychlost konvergence v síti. Z grafu je vidět, že lepších výsledků podle předpokladů dosáhl protokol OSPF. Je to dáno tím, že OSPF používá vlastní mechanismus pro zjištění výpadku. Hello pakety jsou odesílány každých deset sekund. RIP neobsahuje podobný protokol, který by umožnil detekci výpadku linky. U protokolu RIP jsou pouze periodicky odesílány směrovací tabulky každých 30 sekund. Aktualizace může být spuštěna i dříve, pokud směrovač obdrží informaci o změně od jiného směrovače.

Konvergence při první inicializaci protokolu na začátku simulace je u OSPF asi čtyři sekundy. Konvergence je stav sítě, kdy všechny směrovače mají vytvořené kompletní směrovací tabulky a jsou schopny směrovat provoz. U protokolu RIP byla naměřena doba konvergence 14 sekund. Během výpadku již není rozdíl v době konvergence tak znatelný. Ale i při výpadku dosahuje OSPF lepších hodnot.

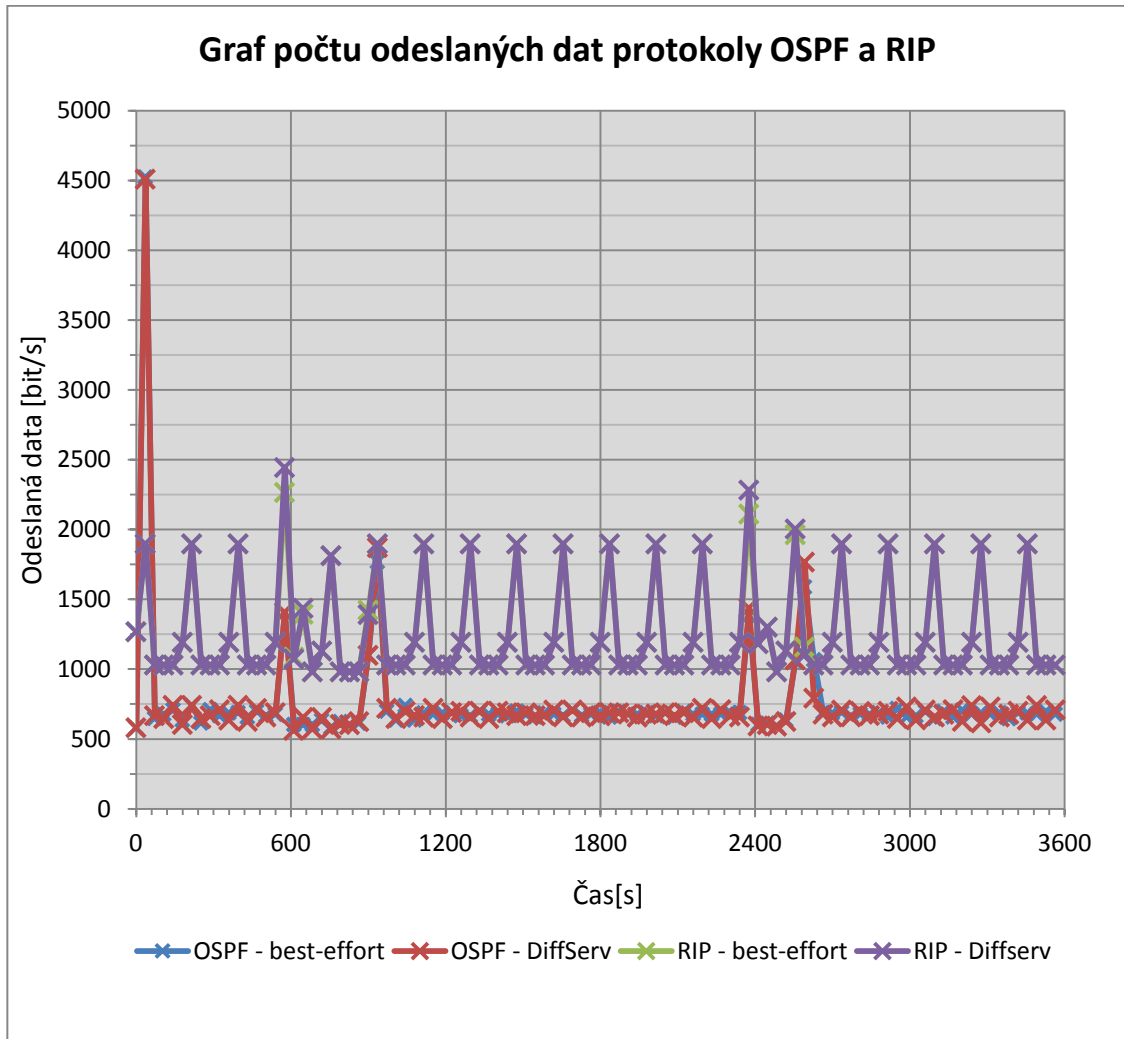


Obr. 5.11: Graf doby konvergence v síti

Na dalším grafu, který je vidět na obrázku 5.12 je porovnání velikosti generovaného datového toku. RIP posílá více dat, protože přenáší kompletní směrovací tabulky. Periodické odesílání směrovacích tabulek je vidět z grafu protokolu RIP. V případě této simulace není rozdíl mezi datovými toky příliš velký. Pokud bychom ale RIP nasadili do prostředí, kde by se nacházely desítky či stovky směrovačů, směrovací tabulky by obsahovaly tisíce záznamů. Datový tok by byl při přenosu tak velkých tabulek znatelně větší.

Další věc, které si můžeme všimnout, je že při první konvergenci sítě OSPF se datový tok poměrně velký. To je způsobeno záplavových šířením LSA zpráv a vyměňováním kompletních topologických databází po celé síti. Jakmile všechny směrovače znají kompletní topologii, není třeba stále tyto informace přenášet.

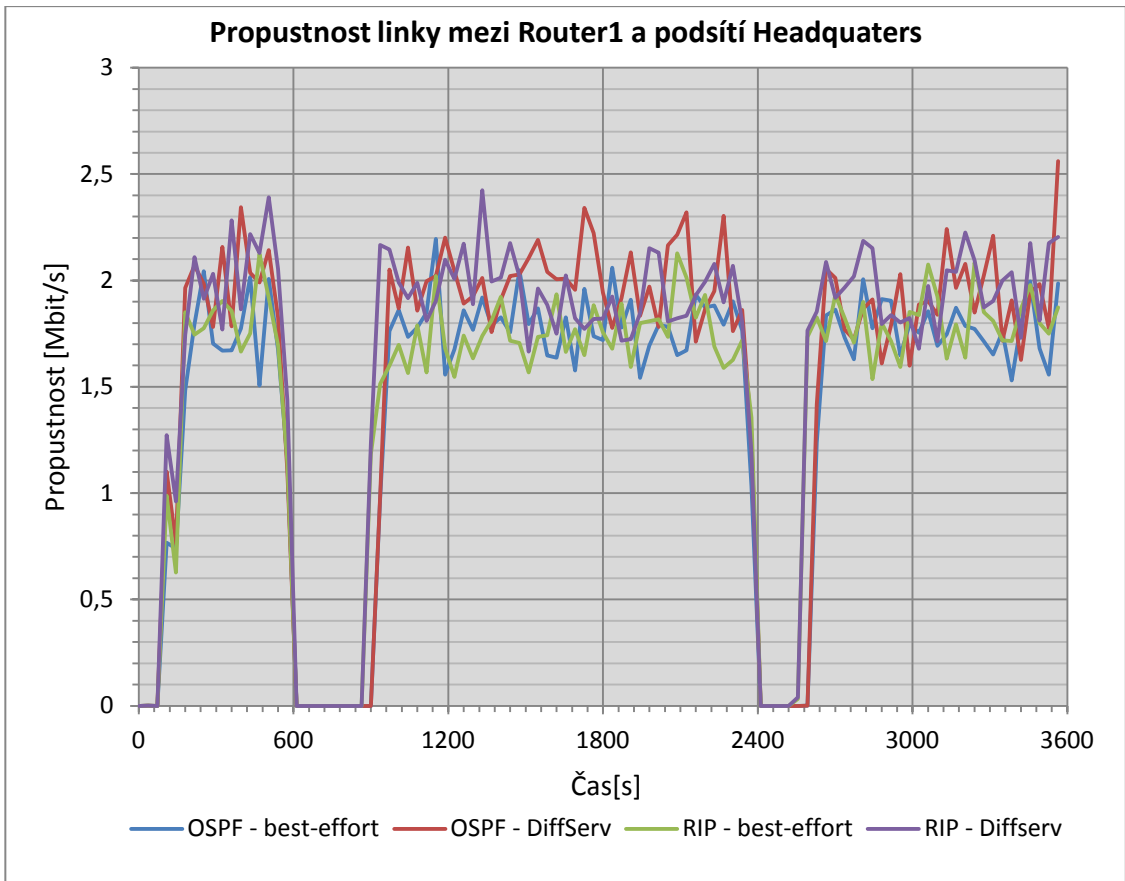
Informace o topologii nebo směrovací tabulky jsou přenášeny, pouze pokud dojde v síti ke změně. Tento jev je vidět v grafu 5.12 v šesté a jedenácté minutě a druhého výpadku v 40. a 43. minutě. V těchto časech jsou v síti rozesílány LSA zprávy, které oznamují, že linka mezi směrovači **Router1** a **Headquaters_Edge_router** je nedostupná. Případně, že linka byla opět provozována. Nastavení QoS velikost datových toků protokolů neovlivní.



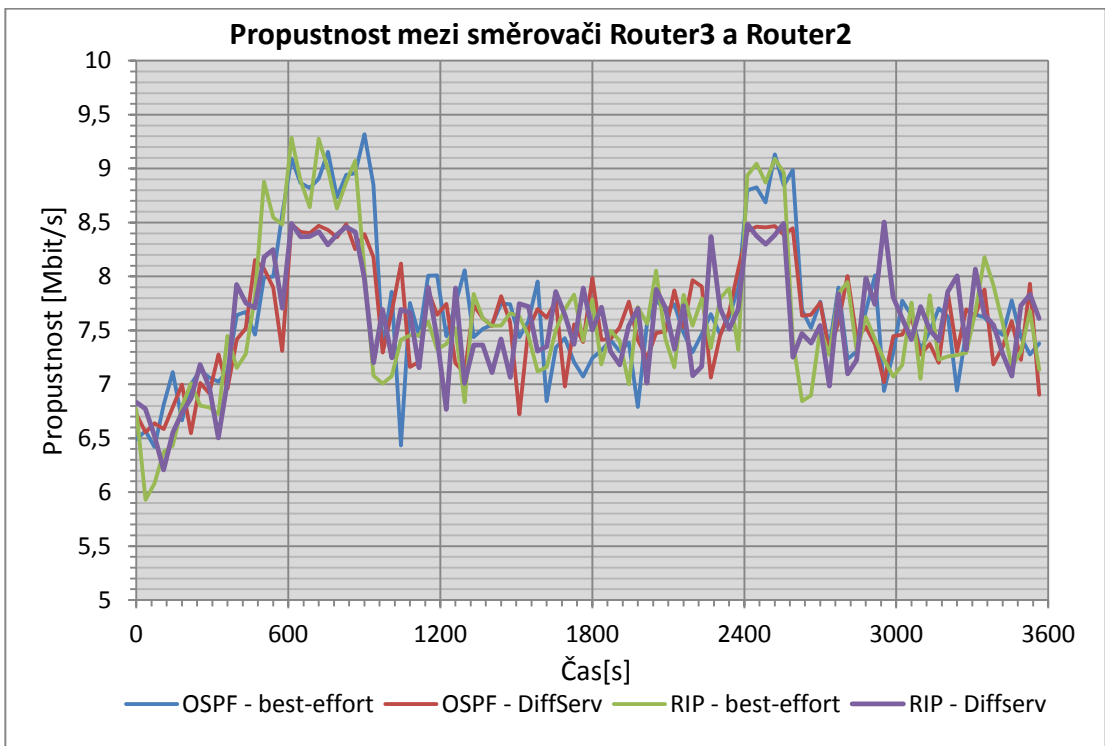
Obr. 5.12: Graf velikosti datového toku protokolů OSPF a RIP

Na obrázku 5.13 je vidět propustnost linky mezi směrovači **Router1** a **Headquaters_Edge_router**. V grafu je vidět simulovaný výpadek v časech 600 sekund a 2400 sekund. A je zde taky vidět doba, po kterou jsou linky nefunkční.

Na dalším obrázku 5.14 je vidět zvýšený provoz v době výpadku. Je to dáno tím, že provoz procházel přes primární linku (přes směrovač Router1) je nyní přeměřován přes směrovače **Router2** a **Router3**. Na těchto linkách byl nastaven provoz v pozadí proto se využití linky mezi směrovači router2 a Router3 blížil 100 procentům. Jakmile je primární linka obnovena, provoz je opět směrován původní cestou.



Obr. 5.13: Propustnost linky mezi směrovači Router1 a Headquarters_Edge_router



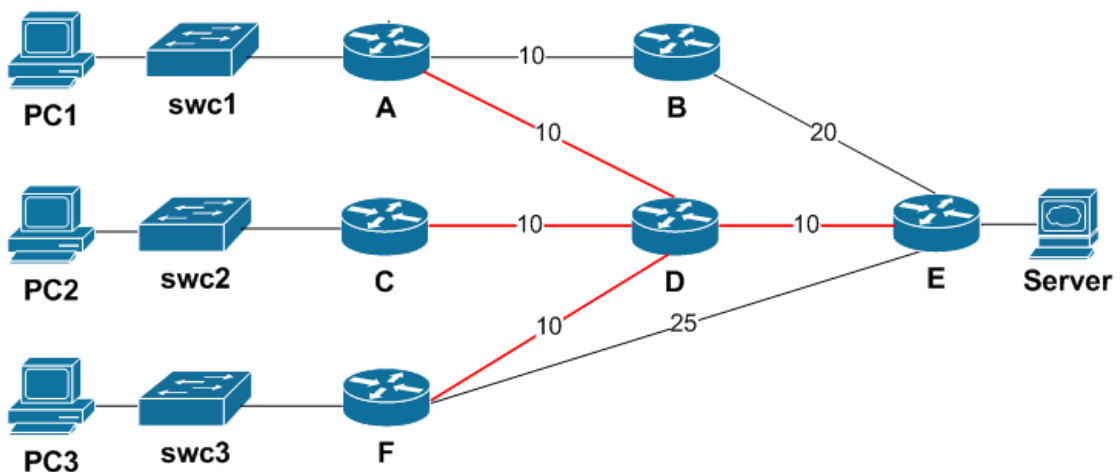
Obr. 5.14: Propustnost linky mezi směrovači Router3 a Router2

6. ADAPTIVNÍ ZMĚNA METRIKY U PROTOKOLU OSPF

Jak již bylo popsáno v kapitole jedna, nejlepší cesta od zdroje k cíli je stanovena na základě metriky. OSPF používá jako metriku cenu linky. Cenu každé linky může automaticky nastavit směrovač při inicializaci protokolu OSPF. V tomto případě je cena linky odvozena od šířky pásma přenosového media. Druhá možnost je, že nastavení ceny linky provede administrátor manuálně. Nastavení ceny linky se po dobu běhu nemění, pokud změnu neprovede ručně administrátor. Tato vlastnost může být v některých případech nedostatkem.

OSPF nedokáže měnit cenu linky na základě vytížení linky. Pokud tedy existují dvě linky se stejnou propustností a jedna bude vytížena z pěti procent a druhá bude vytížena z devadesáti procent, budou ohodnoceny stejnou metrikou.

Tato situace je naznačena na obrázku 6.1 kde stanice PC1 až PC3 komunikují se serverem. Cena linek je nastavena tak, že všechny datové pakety procházejí mezi směrovači D a E. Na této lince může dojít k zahlcování linky. Přitom jsou tu další dvě linky, které zůstávají nevyužité. Řešením tohoto problému je zavedení nové metriky, která by zajišťovala ohodnocení linky z hlediska vytížení. V případě, že by došlo k překročení určitého procentuálního vytížení sledované linky, došlo by ke zvýšení ceny. Zvýšení ceny linky způsobí, že část datového provozu bude směrována jinou cestou. Pro zajištění této funkčnosti, je nutné implementovat do OSPF protokolu algoritmus, který bude sledovat vytížení linky a na základě míry vytížení linky adaptivně přizpůsobovat cenu linky. Pro sledování vytížení linky budou využity regulační diagramy EWMA (Exponentially Weighted Moving Average). EWMA je nazývána jako metoda exponenciálně vážených klouzavých průměrů.



Obr. 6.1: Ukázková topologie

6.1. EXPONENCIÁLNĚ VÁŽENÉ KLOUZAVÉ PRŮMĚRY EWMA

EWMA je metoda, která slouží k vykreslování kontrolních statistik. Vykreslovány jsou zprůměrované hodnoty veličiny sledované v čase. Jednotlivým vzorkům sledované veličiny jsou přiřazovány váhy. U EWMA metody klesá váha u starších vzorků. Výpočet EWMA tedy více ovlivňují aktuální vzorky než starší vzorky. Metoda EWMA je vhodná pro sledování veličin, u kterých nedokážeme dopředu posoudit, jestli bude docházet k plynulým nebo skokovým změnám veličiny. [17] EWMA je počítáno podle následující rovnice:

$$EWMA_{t+1} = EWMA_t + \lambda e_t = EWMA_t + \lambda(y_t - EWMA_t) \quad (6.1)$$

Rovnice 6.1 může být přepsána do tvaru:

$$EWMA_{t+1} = \lambda Y_t + (1 - \lambda)EWMA_{t-1} \quad t = 1, 2, \dots, n \quad (6.2)$$

Kde:

- EWMA_{t+1} - předpovídaný vážený průměr
- EWMA_t - aktuální vážený průměr
- Y_t - sledovaná veličina v čase t
- t - počet pozorování sledované veličiny
- λ - konstanta určující váhu 0 < λ < 1

Konstanta λ určuje hloubku paměti EWMA procesu. Určuje, jak staré hodnoty se uplatní při výpočtu EWMA. Nastavením konstanty lambda je také možné ovlivnit, jak bude průběh EWMA citlivý na skokové či pozvolné změny sledované veličiny. Konstanta λ se nastavuje v rozmezí od nuly do jedné. Pokud bude lambda nastavena na hodnotu jedna, výpočet EWMA bude ovlivněn pouze těmi nejaktuálnějšími hodnotami. Pokud se bude lambda blížit k nule, bude výpočet ovlivněn více staršími hodnotami. Obecně se doporučuje používat konstantu lambda v rozmezí od 0,2 - 0,3. [4]

Rozptyl EWMA statistiky je možné určit podle následujícího rovnice:

$$\sigma_{EWMA}^2 = \frac{\lambda}{2-\lambda} \sigma^2 \quad (6.3)$$

Kde σ je základní odchylka, která je vypočítaná z historické analýzy veličiny sledované veličiny. Odchylka σ je určena podle následující rovnice:

$$\overline{\sigma^2} = \sum_{t=1}^T e_t^2 / (T - 1) = \sum_{t=1}^T (y_t - EWMA_t)^2 / (T - 1) \quad (6.4)$$

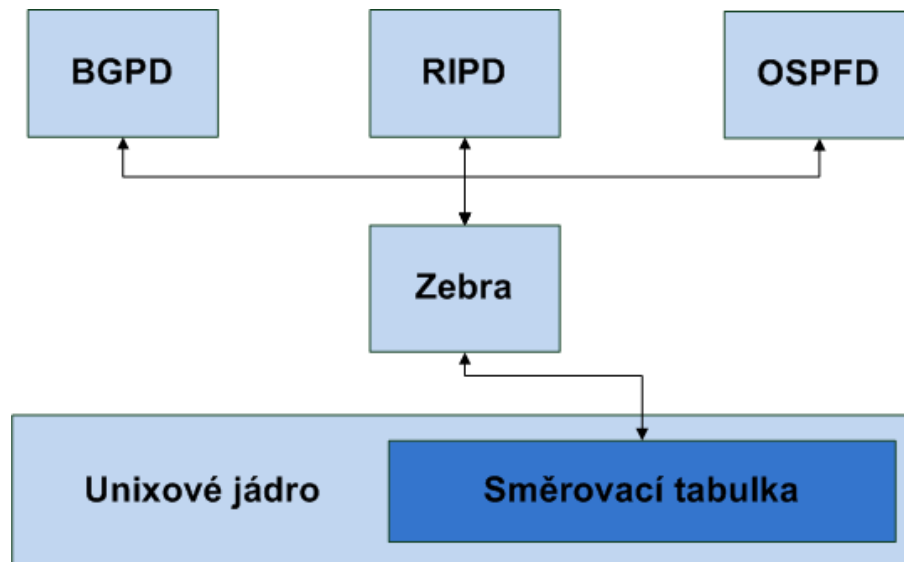
EWMA je ve výrobních procesech používána jako kontrolní mechanismus, který slouží k pozorování veličiny a pokud dojde k překročení určité hranice, dojde k přerušení procesu. Pro omezení EWMA statistik jsou určeny hranice UCL a LCL. Hranice je možné určit podle následujícího vzorce:

$$UCL = EWMA_0 + k\sigma \sqrt{\frac{\lambda}{2-\lambda}} = EWMA_0 + k\sigma_{EWMA} \quad (6.5)$$

$$CL = EWMA_0 \quad (6.6)$$

$$LCL = EWMA_0 - k\sigma \sqrt{\frac{\lambda}{2-\lambda}} = EWMA_0 - k\sigma_{EWMA} \quad (6.7)$$

Quagga je založena na systému GNU Zebra, který navrhl Kunihiro Ishiguro. Architektura systému Quagga je tvořena několika démony. Architektura je naznačena na obrázku 6.3.



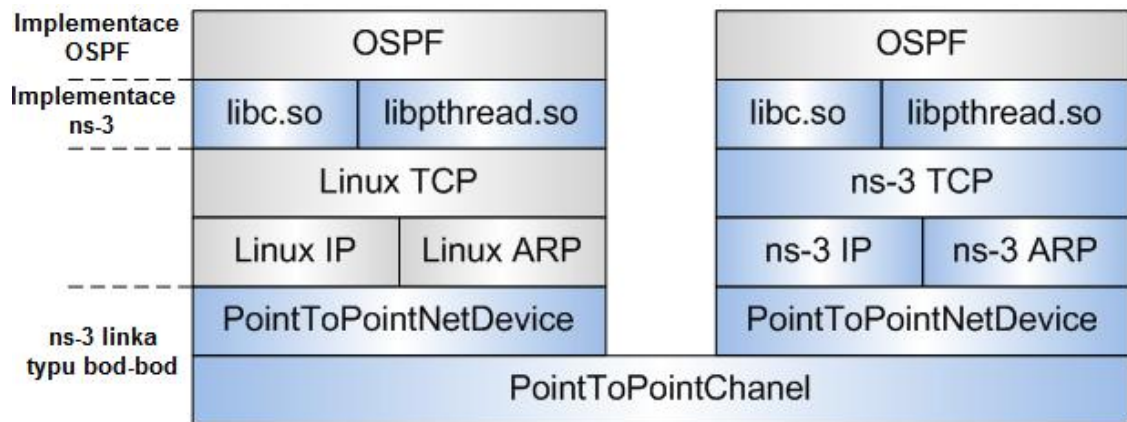
Obr. 6.3: Architektura Quagga

Pro obsluhu jednotlivých směrovacích protokolů jsou použity démony ospfd, ospfd, bgpd, ripd, ripng. Démon Zebra zajišťuje změnu směrovací tabulky v jádru systému, redistribuci cest mezi protokoly. Tento démon bývá nazýván jako správce směrovací tabulky jádra. Tato architektura umožňuje snadnou implementaci dalších směrovacích protokolů. Pro implementaci dalšího protokolu stačí nainstalovat příslušný démon. Výhodou této architektury je snadná rozšiřitelnost, modularita a snadná správa.

6.2.2. SIMULÁTOR NS3-DCE

Pro simulaci upravené metriky je použit simulátor ns3-DCE. Ns3-DCE umožňuje využít síťové prostředky a protokoly systému ve kterém je nainstalovaný. V diplomové práci je ns3-DCE a balík Quagga nainstalován na linuxovou distribuci Slackware 13.17. Začlenění ns-3 do systému Linux je naznačeno na obrázku 6.4. Obrázek znázorňuje součásti, které jsou použity při simulaci. Světle modrou barvou jsou zobrazeny součásti, které jsou součástí simulátoru ns-3. Světle šedou barvou jsou vyznačeny součásti, jejichž kód je spuštěn během simulace, ale tento kód není implementován samotným simulátorem ns-3. Obrázek 6.4 znázorňuje dva uzly, na kterých běží OSPF protokol, které jsou spojeny linkou typu bod-bod, která je vytvořena ns-3 simulátorem. Levý uzel používá TCP/IP zásobník poskytovaný linuxovým systémem. Pravý uzel využívá TCP/IP zásobník implementovaný samotným simulátorem ns-3. Ns-3 zajišťuje vizualizaci protokolů a procesů, tak aby mohlo být vytvářeno více uzlů nad TCP/IP

zásobníkem operačního systému. Pro simulaci nové metriky v diplomové práci je použit TCP/IP zásobník systému Slackware, který poskytne reálnější výsledky.



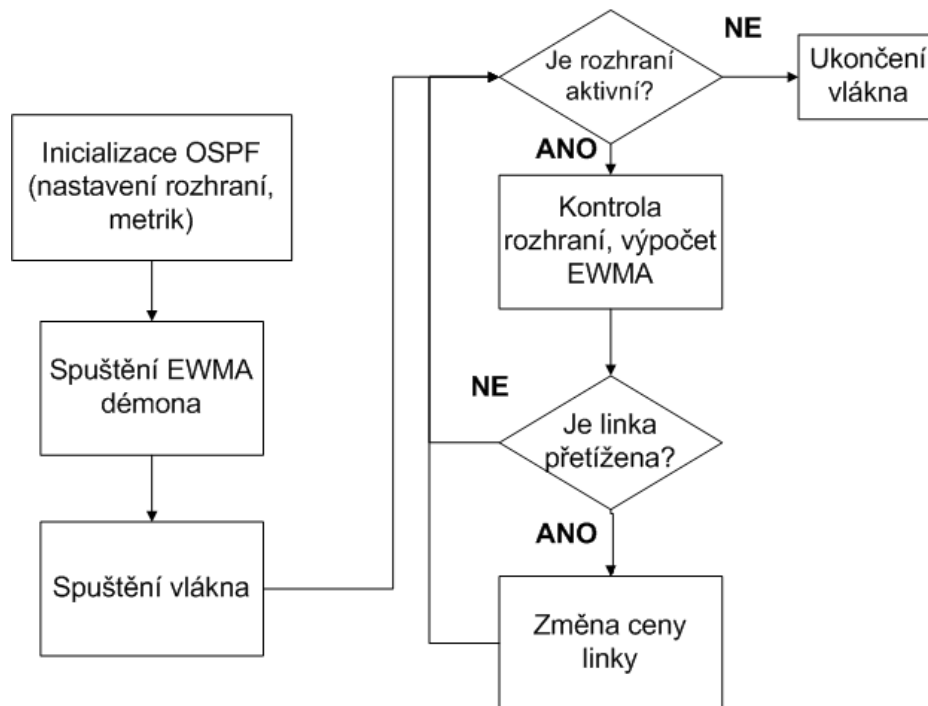
Obr. 6.4: Začlenění ns3-DCE do systému

6.2.3. IMPLEMENTACE NOVÉ METRIKY DO ZDROJOVÉHO KÓDU QUAGGA

V následující části je popsán postup při implementaci nové metriky do zdrojového kódu Quagga. Pro úpravu zdrojových kódů bylo použito vývojové prostředí Anjuta. Upravené zdrojové kódy jsou zkompileovány pomocí sady překladačů GCC (GNU Compiler Collection), která byla vytvořena v rámci projektu GNU. Zdrojové kódy jsou napsány v programovacím jazyce C. Zdrojové kódy démona ospfd jsou umístěny ve složce ./ospfd. Při implementaci nové metriky byly upravovány následující soubory:

- ospf_interface.c – soubor obsahující funkce vztahující se k rozhraním. Vytvoření, odebrání a nastavení rozhraní.
- ospf_interface.h – hlavičkový soubor obsahující deklaraci funkcí a proměných
- ospf_zebra.c – soubor obsahující funkce poskytující rozhraní mezi OSPF démonem a hlavním démonem Zebra
- ospf_main.c – soubor obsahující hlavní metodu, která spouští inicializaci OSPF protokolu
- snmp.c – podpora SNMP protokolu v OSPF
- ewma.c - vytvořený soubor obsahující funkce pro výpočet EWMA

Na obrázku 6.5 je vidět diagram fungování algoritmu přidané funkcionality nové metriky. Nejdříve dojde ke spuštění OSPF démona, inicializace nastavení rozhraní. Jakmile je přidáno nové rozhraní, nebo dojde k aktivaci vypnutého rozhraní, je spuštěno nové vlákno, které v periodických intervalech kontroluje vytížení linky a počítá EWMA. Jednotlivá vlákna spravuje EWMA démon.



Obr. 6.5: Algoritmus adaptivní metriky

Pro správnou funkci EWMA je nutná analýza provozu, procházejícím kontrolovaným rozhraním. Analyzované hodnoty jsou pak využity při prvním výpočtu EWMA. U každého aktivního rozhraní je v pravidelném intervalu vypočítávána hodnota EWMA. Pro zjištění aktuálního vytížení linky je využit SNMP protokol. Vypočítaná hodnota EWMA je poté kontrolována s nastavenou hranicí. Pokud EWMA překročí nastavenou hranici, dojde k navýšení metriky rozhraní. Změna metriky rozhraní probíhá podle schématu, který je uveden v tabulce 6.1. Jakmile hodnota EWMA překročí osmdesát procent, dochází k navýšení metriky rozhraní. Toto navýšení by mělo způsobit, že některé pakety mohou být předávány jinou cestou a dojde ke snížení využitého pásma linky.

Tab. 6.1: Schéma nastavení nových metrik

Vytížení linky	Nová metrika
75%	$(3/2) \cdot \text{aktuální_metrika}$
85%	$2 \cdot \text{aktuální_metrika}$
95%	$3 \cdot \text{aktuální_metrika}$

Kontrola rozhraní a výpočet EWMA

Pro vytvoření nového rozhraní je použita metoda `ospf_if_new` v knihovně `ospf_interface.c`. Po vytvoření a nastavení rozhraní je spuštěna metoda `calculate_ewma`. Tato metoda spouští nové vlákno, které zajistí periodickou kontrolu vytížení linky a výpočet metriky. Vytvoření nového vlákna je vidět ve výpisu kódu 6.1.

Výpis kódu 6.1: Vytvoření nového vlákna

```
pthread_create(&thread, NULL, ewma_check_if, &oi);
```

Při spuštění vlákna jsou načteny a spočítány údaje, které jsou nutné pro výpočet EWMA. Je načtena šířka pásma rozhraní. Z této hodnoty jsou přepočteny skutečné prahové hodnoty propustnosti linky, podle kterých je určeno, jak se bude měnit metrika. Procentuální prahy propustnosti jsou vedeny v tabulce 6.1. Funkcí `ewma_init()` je inicializována hodnota $EWMA_0$, která je nutná pro výpočet první hodnoty EWMA. Funkce `ewma_init()` naměří 20 hodnot aktuální propustnosti rozhraní. Propustnost je měřena v intervalu 10 sekund. Z naměřených hodnot je vypočten průměr, který je použit pro výpočet EWMA v prvním cyklu. Po inicializaci všech hodnot pro výpočet EWMA je spuštěn cyklus, který v intervalu pěti minut počítá EWMA podle rovnice 6.1. V každém cyklu je načtena aktuální vytížení rozhraní pomocí funkce `get_actual_link_utilization()`. Vypočítaná hodnota EWMA je poté srovnávána s prahovými hodnotami, které byly určeny při nastavování hodnot pro výpočet EWMA. Pokud dojde k překročení prahové hodnoty, je volána metoda `ewma_if_recalculate_cost()`, která linku znevýhodní zvýšením její ceny. Cena linky je měněna podle tabulky 6.1. Při překročení prahu je nastaven parametr, který slouží k identifikaci míry vytížení rozhraní. Na základě tohoto parametru se poté změní cena rozhraní. Nakonec je v každém cyklu uložena hodnota EWMA, která je využita pro výpočet v další iteraci.

Metoda `ewma_if_recalculate_cost` zajistí změnu linky. Při překročení nejnižšího prahu je uložena aktuálně nastavená cena linky. Tato hodnota bude použita jako referenční při výpočtu nové ceny rozhraní. Tato cena je opět nastavena pokud vytížení rozhraní opět klesne pod 75%. Po nastavení nové ceny rozhraní je spuštěna metoda `ospf_router_lsa_update_area()`, která odešle LSA zprávu, která informuje ostatní směrovače v oblasti o změně metriky rozhraní.

Kontrola stavu linky

Pro kontrolu vytížení linky je použit SNMP (Simple Network Management Protocol) protokol. SNMP protokol slouží k nastavování a získávání hodnot různých síťových zařízení. Každá hodnota v SNMP protokolu je identifikována pomocí identifikátoru OID (Object Identifier). OID je řetězec čísel oddělených tečkami. Čísla označují objekty v MIB databázi. MIB databáze obsahuje jména a popisy objektů. Objekty MIB databáze jsou popsány v dokumentu RFC 1213. [8]

V práci jsou využity následující objekty:

- ifInOctets – počet celkově přijatých oktetů na rozhraní, včetně záhlaví
 - OID identifikátor - .1.3.6.1.2.1.2.2.1.10
 - textový popis - { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 10 }
- ifOutOctets – celkový počet odeslaných oktetů na rozhraní, včetně záhlaví
 - OID identifikátor - .1.3.6.1.2.1.2.2.1.16
 - textový popis - { ISO(1) org(3) DOD(6) Internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 16 }
- ifSpeed – aktuální šířka pásma rozhraní
 - OID identifikátor - .1.3.6.1.2.1.2.2.1.5
 - textový popis - { ISO(1) org(3) DOD(6) Internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 5 }

Objekty ifInOctets, ifOutOctets a ifSpeed jsou součástí MIB-II databáze. Proměnné u MIB-II jsou ukládány jako čítače a proto je nutné určit rozdíl proměnné mezi dvěma cykly. Výpočet vstupní a výstupní propustnosti podle těchto SNMP objektů je popsáno rovnicemi 6.8 a 6.9. [8]

$$Propustnost_{vst} = \frac{\Delta ifInOctets * 8 * 100}{(\text{počet sekund při } \Delta) * ifSpeed} [\%] \quad (6.8)$$

$$Propustnost_{výst} = \frac{\Delta ifOutOctets * 8 * 100}{(\text{počet sekund při } \Delta) * ifSpeed} [\%] \quad (6.9)$$

Získávání parametrů pomocí SNMP protokolu je uskutečněno komunikací typu server-klient. Server posílá klientovi požadavky, označené identifikátorem OID a klient posílá na tyto požadavky odpovědi. Směrovače základní výbavě podporují funkci SNMP klienta. Pro získání aktuálního vytížení linky je nutné do OSPF protokolu implementovat funkci, která bude posílat požadavky na OID objekty na jednotlivá rozhraní. K implementaci SNMP protokolu jsem použil API které je součástí balíku NET-SNMP, které je možné stáhnout na url <http://www.net-snmp.org>.

Kód umožňující zjištění vytížení linky je umístěn v souboru snmpd.c. Hlavní metoda pro zjištění stavu linky má název get_snmp_object. Vstupním parametrem je rozhraní u kterého je zjišťováno vytížení linky. Nejdříve je inicializována SNMP relace metodou snmp_sess_init(). Základní nastavení struktury relace je možné změnit. Inicializace a nastavení SNMP relace je vidět ve výpisu kódu 6.2. Důležité je uvést verzi SNMP protokolu a také definovat adresu rozhraní.

Výpis kódu 6.2: Definice SNMP relace

```
snmp_sess_init( &session );
session.version = SNMP_VERSION_2;
session.community = "public";
session.community_len = strlen(session.community);
session.peername = oi->address;
```

Po inicializaci a nastavení relace dochází k otevření spojení pomocí metody `snmp_open()`. Po otevření spojení je vytvořena datová jednotka PDU (Primary Data Unit), ve kterých jsou přenášeny požadavky na OID objekty. V jednom PDU může být obsaženo více požadavků na OID objekty. PDU jednotka je zabalena do SNMP paketu. Pro zjišťování parametru jsou použity žádosti pouze pro čtení. Žádosti pro zápis se používají pro vzdálené nastavení parametru pomocí SNMP protokolu. V jednom PDU mohou být pouze žádosti stejného typu. Přidání OID do PDU je dvou fázový proces. Nejdříve je OID načteno z MIB databáze (metoda `read_objid`) a poté je OID zapsáno do PDU (metoda `snmp_add_null_var`). Vytvoření žádosti pro získání objektu přijatých oktetů na rozhraní je vidět na výpisu kódu 6.3.

Výpis kódu 6.3: Vytvoření požadavku v PDU

```
read_objid("IF-MIB::ifInOctets.0", input_oct_oid, &input_oct_len);
snmp_add_null_var(pdu, input_oct_oid, input_oct_len);
```

Následně je pomocí metody `snmp synch response()` odeslána datová jednotka PDU a je očekávána odpověď. Odpověď je uložena do prázdné struktury, která je přístupná pomocí ukazatele `response`. Odpověď obsahuje identifikátory OID a hodnoty požadovaných parametrů. PDU jsou smazána metodou `snmp_free_pdu` a SNMP relace je ukončena metodou `snmp_close`.

7. ZÁVĚR

V teoretické části jsou popsány základní principy směrování v IP sítích. Jsou zde popsány rozdíly mezi link-state a distance-vektor směrovacími protokoly. Podrobněji jsou popsány vlastnosti a součásti protokolu OSPF. V práci je popsána struktura zpráv, které OSPF protokol používá při komunikaci mezi směrovači a při budování topologických databází a směrovacích tabulek. V práci je popsán mechanismus, který umožňuje detekovat výpadky linek.

V další části jsou vysvětleny mechanismy pro zajištění kvality služeb. V práci jsou vysvětleny mechanismy IntServ a DiffServ. Podstatná část je zaměřena na mechanismus DiffServ. Kapitola o DiffServ popisuje, jak probíhá klasifikace a značkování jednotlivých paketů. Jsou zde popsány způsoby zacházení s pakety PHB, které se aplikují na označené pakety.

V praktické části je vytvořena simulace v prostředí Opnet Modeler. V simulaci jsou vytvořeny čtyři scénáře pro srovnání protokolů OSPF a RIP. V těchto scénářích byl také zjištěn vliv QoS na přenosové vlastnosti sítě. Simulací bylo zjištěno, že protokol OSPF rychleji konvergoval při výpadku. Je to dáno tím, že OSPF protokol umožňuje každému směrovači znát úplnou topologii sítě. Každý směrovač díky topologické databázi ví o všech směrovačích v síti. Další vlastností, která napomáhá k rychlejší konvergenci je hello protokol, který zajišťuje udržení spojení mezi sousedními směrovači. Další výhodou OSPF protokolu je nižší datový tok, který generuje. Pokud dojde v síti ke změně, budou u OSPF protokolu přenášeny pouze záznamy týkající se těchto změn, nedochází jako u RIP protokolu k přenosu kompletních směrovacích tabulek. Horších výsledků OSPF dosáhl pouze při měření kolísání zpoždění u aplikace VoIP, který je na obrázku 5.8. Tento jev může být způsoben špatným nastavením.

V simulaci jsou dva výpadky, které způsobí přesměrování provozu přes záložní linku. Propustnost linky mezi směrovači Router1 a Headquarters_Edge_router je vidět na obrázcích 5.12 a 5.13. Přesměrování provozu přes záložní linku způsobí u scénářů bez QoS narůstání zpoždění a kolísání zpoždění. Zpoždění je způsobeno tím, že data procházejí cestou, která je o jeden skok delší a tím, že na záložní lince je nastaven provoz v pozadí. Směrovače Router2 a Router3 tak zpracovávají mnohem větší provoz, což může způsobovat ukládání paketů do front, nebo zahazování paketů.

Při porovnání scénářů s nastavením QoS a bez QoS lze usoudit, že pokud je v síti provozována služba citlivá na zpoždění či kolísání zpoždění, je vhodné použít některý QoS mechanismus. Při nastavování QoS je důležité správně přiřadit priority aplikacím. Při nesprávném nastavení může být funkce QoS degradována, což může způsobit vznik velkých zpoždění nebo k velké ztrátovosti paketů. Z grafů je také vidět, že u scénářů s QoS výpadek téměř neovlivnil zpoždění a kolísání zpoždění u aplikace VoIP, protože tato aplikace měla nejvyšší prioritu.

V závěrečné části bylo navrženo vylepšení směrování u OSPF protokolu. Vylepšení směrování je provedeno přidáním nové metriky. Nová metrika ohodnocuje rozhraní podle jeho vytížení. Každé rozhraní je v intervalu pěti minut kontrolováno a je určeno jeho vytížení. Pomocí metody EWMA je vypočítán klouzavý průměr, který je porovnáván s nastavenými prahovými hodnotami. Pokud dojde k překročení těchto prahů, dochází k navýšení ceny linky. Tento postup je v periodických intervalech opakován a cena rozhraní je měněna v závislosti na jeho vytížení. Pro implementaci nové metriky do protokolu OSPF byl využit balík Quagga. Bohužel se mi z důvodů rozsáhlosti a nedostatečné dokumentace zdrojových kódů nepodařilo plně tuto funkčnost do OSPF implementovat.

8. CITOVANÁ LITERATURA

- [1] ALMQUIST, Philip. Type of Service in the Internet Protocol Suite. [online]. July 1992 [cit. 2012-03-16]. Dostupné z: <http://www.ietf.org/rfc/rfc1349.txt>
- [2] BLAKE, S., BLACK, D. a CARLSON, M. *RFC 2475: An Architecture for Differentiated Services*. Internet Engineering Task Force. [online] December 1998 [cit. 2012-03-16]. Dostupné z: <http://www.ietf.org/rfc/rfc2475.txt>
- [3] BRADEN, R., CLARK, D. a SHENKER, S. *RFC 2212: Specification of Guaranteed Quality of Service*. Internet Engineering Task Force. [online] September 1997 [cit. 2012-03-18]. Dostupné z: <http://rsync.tools.ietf.org/rfc/rfc2212.txt>
- [4] CISAR P., BOSNIAK S., CISAR S. M., *EWMA Algorithm in Network Practice*, International Journal of Computers Communications & Control, ISSN 1841-9836, 5(2):160-170, 2010.
- [5] CISCO SYSTEMS, Inc. *Cisco IOS IP Service Level Agreements*. USA: Cisco Systems, Inc. [online]. 2005 [cit. 2012-03-16]. Dostupné z: http://www.cisco.com/en/US/technologies/tk648/tk362/tk920/technologies_white_paper0900aecd8017f8c9.pdf
- [6] CISCO SYSTEMS, Inc. *Classifying VoIP Signaling and Media with DSCP for QoS*. USA: Cisco System, Inc. [online] 2005 [cit. 2012-03-16] Dostupné z: http://www.cisco.com/en/US/docs/ios/12_2t/12_2t2/feature/guide/ft_dscp.pdf
- [7] CISCO SYSTEMS, Inc. *DiffServ - The Scalable End-to-End QoS Model*. USA: Cisco Systems, Inc. [online] 2005 [cit. 2012-03-16] Dostupné z: http://www.cisco.com/en/US/technologies/tk543/tk766/technologies_white_paper09186a00800a3e2f.pdf
- [8] CISCO SYSTEMS, Inc. *How To Calculate Bandwidth Utilization Using SNMP*. USA: Cisco Systems, Inc., [online] 2005 [cit. 2012-03-16] Dostupné z: http://www.cisco.com/application/pdf/paws/8141/calculate_bandwidth_snmp.pdf
- [9] CISCO SYSTEMS, Inc. *OSPF Support for Fast Hellos*. USA: Cisco Systems, Inc., [online] 2005 [cit. 2012-03-16]. Dostupné z: http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/fasthelo.pdf
- [10] COLTUN, R., FERGUSON D., MOY J. RFC 2740: OSPF for IPv6. [online]. 1999 [cit. 2012-05-13]. Dostupné z: <http://www.ietf.org/rfc/rfc2740.txt>
- [11] DOYLE, J. *Routing TCP/IP Volume I: CCIE Professional Development*. USA: Cisco Press, 1998. ISBN 1578700418.

- [12] GOYAL, M., RAMAKRISHNAN, K. K., FENG, W. *Achieving Faster Failure Detection in OSPF Networks* [online]. USA, May 2003 [cit. 2012-05-13]. Communications, 2003. ICC '03. IEEE International Conference on. ISBN 0-7803-7802-4.
- [13] HEDRICK, C. Routing Information Protocol. [online]. 1988 [cit. 2012-05-12]. Dostupné z: <http://tools.ietf.org/html/rfc1058>
- [14] HEINANEN, J., F. BAKER a W. WEISS. RFC 2597: Assured Forwarding PHB Group. [online]. June 1999 [cit. 2012-05-13]. Dostupné z: <http://tools.ietf.org/html/rfc2597>
- [15] HELDON, Tom. *McGraw Hill's Encyclopedia of Networking and Telecommunications*. USA: Osborne/McGraw-Hill, January 2001. ISBN 0072120053.
- [16] JACOBSON, V., NICHOLS, K. A PODURI, K. *RFC 2598: An Expedited Forwarding PHB*. [online]. June 1999 [cit. 2012-05-13]. Dostupné z: <http://www.ietf.org/rfc/rfc2598.txt>
- [17] KLÁŠTERECKÝ, P. *Některé problémy statistické kontroly jakosti*. Praha, 2003. Diplomová práce. Univerzita Karlova v Praze.
- [18] LAMMLE, T., ODOM S., WALLACE, K. *CCNP Routing Study Guide*. Alameda CA: Sybex Inc, 2001. ISBN 0-7821-2712-6.
- [19] MALKIN, G. RIP Version 2. [online]. 1988 [cit. 2012-05-12]. Dostupné z: <http://tools.ietf.org/html/rfc2453>
- [20] MOLNÁR, K. *Řízení kvality služeb*. Brno: UTKO FEKT VUT. Dostupné z: <https://utko.utko.feec.vutbr.cz/~molnar/mmos/QoS.pdf>
- [21] MOLNÁR, K., SKOŘEPA, M. A ZEMAN, O. *Moderní síťové technologie – Laboratorní cvičení*. [Online] 2008. Dostupné z: http://www.utko.feec.vutbr.cz/~molnar/mmos/MMOS_lab.pdf.
- [22] MOY, J. RFC 2328: OSPF Version 2. [online]. April 1998 [cit. 2012-05-13]. Dostupné z: <http://www.ietf.org/rfc/rfc2328.txt>
- [23] PARK, K. I. *QoS in Packet Networks: The Springer International Series in Engineering and Computer Science*. Boston USA: Springer, October 2004. ISBN 038723389X.
- [24] PUŽMANOVÁ, Rita. *Routing and Switching: TIME OF CONVERGENCE?*. Great Britain: Addison-Wesley Professional, 2001. ISBN 0-201-39861-3.
- [25] RAMASAMY, K. a D. MEHDI. *Network Routing: Algorithms, Protocols, and Architectures: The Morgan Kaufmann Series in Networking*. 1 edition. San Francisco, USA: Morgan Kaufmann, April 12, 2007. ISBN 0-12-088588-3.

[26] TEARE, Diane. *Implementing Cisco IP Routing (ROUTE): Foundation Learning Guide*. Indianapolis, IN 46240 USA: Cisco Press, 2010. ISBN 1587058820.

SEZNAM ZKRATEK:

ABR - Area Boundary Router
ACL - Access Control List
AF - Assured Forwarding
ARP - Address Resolution Protocol
AS - Autonomous System
ASBR - Autonomous System Boundary Router
BA - Behaviour Aggregate
BDR - Backup Designated Router
DDP - Datagram Delivery Protocol
DDP - Database Description Packet
DiffServ - Differentiated services
DR - Designated Router
DSCP - Differentiated Services Code Point
EF - Expedited Forwarding
EGP - Exterior Gateway Protocol
EWMA - Exponentially Weighted Moving Average
FTP - File Transfer Protocol
GCC (GNU Compiler Collection)
GPL - General Public License
HTTP - Hypertext Transfer Protocol
IGRP - Interior Gateway Protocol
IntServ - Integrated Services
IP - Internet Protocol
IPP - IP Precedence
IPX - Internetwork Packet Exchange
IS-IS - Intermediate System to Intermediate System
ISO - International Organization for Standardization
OSI - Open Systems Interconnection
LAN - Local Area Network
LSA - Link State Advertisement
LSR - Link-State Request
LSU - Link-State Update
MD5 - Message-Digest Algorithm
MIB-II - Management Information Base
NBMA - Non-broadcast Multiple Access
OSPF - Open Shortest Path First
PBX - Private branch exchange
PHB - Per Hop Behaviour
PPP - Point-to-Point Protocol
QoS - Quality of Service
RIP - Routing Information Protocol

RSVP - ReSource reserVation Protokol
SBM - Subnet Bandwith Management
SNMP (Simple Network Management Protocol)
SLA - Service Level Agreement
SPF - Shortest Path First
TCA - Traffic Conditioning Agreement
TCP/IP - Transmission Control Protocol/Internet Protocol
TOS - Type of Service
TSpec - Traffic Specification
UDP - User Datagram Protocol
VLSM - Variable-length Subnet Masking
VoIP - Voice over Internet Protocol
WAN - Wide Area Network

SEZNAM PŘÍLOH

A. OBSAH PŘILOŽENÉHO CD

A. OBSAH PŘILOŽENÉHO MÉDIA

- Elektronická verze diplomové práce ve formátu PDF
- Archiv ospfd.zip s upravenými zdrojovými kódy balíku Quagga
- Archiv DP_IP_Performance.project.zip obsahující soubory simulace v OM