VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

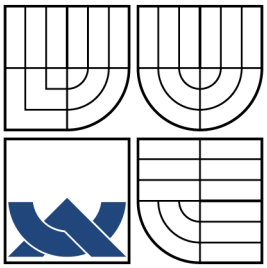
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ZAJIŠTĚNÍ KVALITY SLUŽEB POMOCÍ PROTOKOLU MPLS

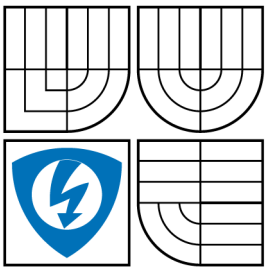
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE BC. VÁCLAV HENZL
AUTHOR

BRNO 2010



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

ZAJIŠTĚNÍ KVALITY SLUŽEB POMOCÍ PROTOKOLU MPLS

QUALITY OF SERVICE ASSURANCE USING PROTOCOL MPLS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — BC. VÁCLAV HENZL
AUTHOR

VEDOUCÍ PRÁCE — ING. JIŘÍ HOŠEK
SUPERVISOR

BRNO 2010

ZADÁNÍ

Podrobně se seznamte s technologií MPLS (Multiprotocol Label Switching) a způsobu jejího využití pro zajištění kvality služeb v datových sítích. Zaměřte se také na problematiku výpadku linky v sítích MPLS a jejího vlivu na vlastní komunikaci. Nedílnou součástí moderních MPLS sítí je oblast Traffic Engineering (TE). V rámci této oblasti se zaměřte zejména na možnosti budování MPLS TE tunelů. V praktické části práce se seznamte se simulačním prostředím OPNET Modeler a vytvořte v něm rozšířený model MPLS sítě, do kterého se pokuste formou několika scénářů implementovat výše uvedené stavy (výpadek linky, TE tunel). Proveďte komplexní simulaci a následné vyhodnocení chování sítě při různých stavech s ohledem na zajištění kvality služeb. Dosažené výsledky přehledně uveďte v závěrečné práci.

ABSTRAKT

Cílem této diplomové práce je vytvoření modelu MPLS sítě v programovém vybavení OPNET Modeler a následné simulování chování této sítě se zaměřením na zajištění kvality služeb. Tato práce obsahuje teoretický úvod do problematiky MPLS sítí a zajištění kvality služeb - dále zkoumá simulaci přenosu dat a hlasových služeb v rozšířeném modelu komunikační sítě. Práce v několika scénářích zkoumá vlastnosti MPLS a jeho vlivu na dosažení kvality služeb s kladením důrazu na problematiku výpadku linky v sítích MPLS a na TE tunely.

KLÍČOVÁ SLOVA

MPLS, QoS, OPNET Modeler, simulace, výpadek linky, TE tunel

ABSTRACT

The aim of this diploma thesis is to create a MPLS network model using OPNET Modeler program and consecutive simulation of this network behaivor with focus on quality of service assurance. This project contains also theoretical introduction to the MPLS network problematic and quality of service assurance - hereafter it is focused on simulation and analysis of data flows in extended communication network model with focus on Voice Over IP parameters. This thesis is discovering attributes of MPLS with respect to network traffic with specific focus on the MPLS data link degradation and TE tunnels.

KEYWORDS

MPLS, QoS, OPNET Modeler environment, simulation, network link disturbtion, TE tunnel

HENZL V. *Zajištění kvality služeb pomocí protokolu MPLS*. Brno: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ, ÚSTAV TELEKOMUNIKACÍ, Rok vydání: 2010. Počet stran 98. Vedoucí diplomové práce Ing. Jiří Hošek.

PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma **Zajištění kvality služeb pomocí protokolu MPLS** jsem vypracoval samostatně pod vedením vedoucího práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

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

V Brně dne 17.05.2010

Bc. Václav Henzl

OBSAH

Úvod **11**

1 Kvalita služeb - QoS **13**
1.1 Definice . . . 13
1.2 Faktory . . . 13
1.3 Třídy QoS . . . 14
1.4 Protokoly a algoritmy QoS . . . 14

2 MPLS protokol **16**
2.1 Základy MPLS . . . 16
2.2 Výhody . . . 16
2.3 Stavební prvky . . . 17
2.4 Forward equivallence class . . . 18
2.5 Značky a jejich přiřazování, MPLS hlavička . . . 18
2.6 Základní principy signalizace . . . 18
2.7 Signalizační protokoly . . . 19
2.7.1 LDP . . . 20
2.7.2 CR-LDP . . . 20
2.7.3 RSVP-TE . . . 21
2.7.4 BGP-4 . . . 22
2.8 Cesta sítí . . . 22
2.9 MPLS a detekce chyb . . . 25
2.9.1 RSVP-TE HELLO . . . 25
2.9.2 RSVP-TE Softstate . . . 26
2.9.3 LSP Ping/ Traceroute . . . 26
2.9.4 Elegantní restart . . . 26
2.10 Mechanismy obnovy u MPLS . . . 27
2.10.1 Globální obnova . . . 28
2.10.2 Lokální obnova . . . 28
2.10.3 Obnova přesměrováním . . . 29
2.10.4 Obnova ochranným přepínáním (Protection Switching) . . . 29
2.10.5 Techniky pro výběr náhradní cesty . . . 30
2.11 MPLS Traffic engineering . . . 31
2.11.1 MPLS Tunely . . . 33
2.11.2 Traffic Engeneering v souvislosti s QoS . . . 34
2.11.3 Omezení šířky pásma - Bandwidth Constriants . . . 35
2.11.4 Maximum Allocation Model - MAM . . . 35

2.11.5 Russian Doll Model - RDM 36

3 Implementace QoS v MPLS **37**

4 Příprava prostředí OPNET **39**
4.1 Síťový model 39
4.2 Vytvoření modelu 39

5 Simulace **51**
5.1 MPLS bez QoS 51
5.1.1 Vytvoření scénáře 51
5.1.2 Výsledky simulace 56
5.2 MPLS s implementací QoS 56
5.2.1 Vytvoření scénáře 57
5.2.2 Výsledky simulace 60
5.3 MPLS a jeho chování při výpadku linky 64
5.3.1 Vytvoření scénáře 64
5.3.2 Výsledky simulace 65
5.4 MPLS s TE tunelem 71
5.4.1 Vytvoření scénáře 71
5.4.2 Výsledky simulace 75
5.5 MPLS s TE tunelem a vliv výpadku páteřní linky 80
5.5.1 Vytvoření scénáře 80
5.5.2 Výsledky simulace 81

6 Závěr **87**

Reference **88**

Seznam symbolů, veličin a zkratek **91**

A Přílohy **93**

SEZNAM OBRÁZKŮ

2.1 Obecný formát MPLS hlavičky 19
2.2 Princip signalizačního mechanismu MPLS 19
2.3 Vytváření LSP a předávání dat v rámci MPLS domény 23
2.4 Globální obnova v síti MPLS 29
2.5 Lokální obnova v síti MPLS 30
2.6 MPLS TE tunel 33
2.7 Model řízení šířky pásma - MAM 35
2.8 Model řízení šířky pásma - RDM 36
4.1 Schéma topologie sítě pro simulaci 41
4.2 Založení nového projektu v prostředí OPNET 41
4.3 Připravená mapa České republiky 42
4.4 Konfigurace LAN parametrů modelu 1000BaseX_LAN 43
4.5 Vyobrazení modelu vzorové subsítě Praha 44
4.6 Vyobrazení hotového modelu sítě vč. linek 45
4.7 Vyobrazení nastavení aplikačních profilů 47
4.8 Nastavení aplikačního profilu v segmentu Cheb_Users1 48
4.9 Nastavení aplikačního profilu v segmentu Cheb_Users2 50
5.1 Definice aplikační sady, aplikačních profilů a profilu MPLS 51
5.2 Definice profilu MPLS a nastavení FEC tříd 52
5.3 Nastavení mapování LSR na daném LER směrovači 53
5.4 Mapa rozšířeného modelu MPLS sítě s nakonfigurovanými LSP 54
5.5 Zobrazení dílčích datových toků sítí MPLS (traffic flows) 56
5.6 Konfigurace mapování provozu MPLS pro uplatnění QoS 59
5.7 Průměrná hodnota parametru **jitter** hlasového přenosu VoIP v závislosti na času simulace 62
5.8 Průměrná hodnota parametru **jitter** hlasového přenosu VoIP v závislosti na času simulace s vyobrazeným průběhem zatížení linky 63
5.9 Průměrná hodnota parametru **jitter** hlasového přenosu VoIP - výpadek linky 68
5.10 Průměrná hodnota parametru **jitter** hlasového přenosu VoIP - výpadek linky - s vyobrazeným průběhem výpadku páteřní linky 69
5.11 Simulace výpadku páteřní linky - vliv na **jitter** a počet skoků (HOP) při průchodu paketu sítí 70
5.12 Konfigurace přiřazení portů jednotlivé linky 72
5.13 Průvodce pro sestavení TE tunelu 73
5.14 Konfigurace žádosti ICMP REQUEST mezi pobočkami 74
5.15 Nastavení atributů pro PING 74

5.16 Ping mezi jednotlivými firemními pobočkami na konci TE tunelu . . 78
5.17 Odezva mezi jednotlivými firemními pobočkami na konci TE tunelu . 79
5.18 Ping při výpadku páteřní linky mezi jednotlivými firemními pobočkami na konci TE tunelu 85
5.19 Odezva mezi jednotlivými firemními pobočkami na konci TE tunelu - simulace při výpadku linky 86
A.1 Propustnost point-to-point spoje Plzeň-Praha v závislosti na času . . 94
A.2 Propustnost point-to-point spoje Plzeň-České Budějovice v závislosti na času 95
A.3 Zatížení point-to-point spoje Plzeň-Praha v závislosti na času 96
A.4 Packet Delay Variation hlasových služeb simulace v závislosti na času 97
A.5 End-to-end zpoždění hlasových služeb simulace v závislosti na času . 98

SEZNAM TABULEK

1.1 Třídy kvality služeb 14
2.1 MPLS protokol - aktivity při přenosu 23
4.1 Seznam měst ČR v rozšířeném modelu sítě 40
5.1 Přiřazení síťových rozhraní směrovačů LER 72
5.2 Trasování paketů TE tunelu na úrovni MPLS sítě 76
5.3 Trasování paketů TE tunelu na úrovni MPLS sítě 83

ÚVOD

MPLS protokol (Multi Protocol Label Switching) je označován - jak je uvedeno v [7] - za víceprotokolární, tj. může být nasazen v kombinaci s jakýmkoliv protokolem síťové vrstvy[1]. Jak zdroje [7] a [8] totožně uvádějí, lze definici MPLS vykládat několika různými správnými způsoby a to v závislosti na výrobci technologie, ke kterému danou problematiku vztahujeme.

Hlavní myšlenkou MPLS je vytvořit rozhraní mezi síťovou a spojovou vrstvou dle modelu ISO/OSI[2]. MPLS se v dnešní době váže na silné optické technologie pomocí myšlenky, že pokud je cesta na konkrétní místo určení nebo skupinu destinací systémem předem navržena, je tato technika mnohem rychlejší a efektivnější než v případě, kdy každý směrovač nebo zařízení podél cesty zkoumá záhlaví přenášených dat a definuje v každém takovém okamžiku cestu dalšího skoku směrování.

Jak jednoznačně vyplývá z [7] a [8], v současné době hraje MPLS hlavní roli zejména u následujících přenosových médií:

- Ethernet
- ATM
- optické sítě

Kromě výše zmíněné základní myšlenky má v sobě MPLS implementováno ještě více funkcí. Za zmínku stojí právě nativní řízení kvality služeb s pomocí metody DiffServ[3], které je přenášeno přímo v EXP bitech MPLS hlavičky. V této práci si představíme princip funkce MPLS do hloubky. Praktickým pokusem - Simulací v programu Opnet Modeler - se poté pokusím dokázat praktické důsledky výhod, které MPLS přináší a to s ohledem na řízení kvality služeb. Diplomová práce je koncipována především formou návodu, tj. čtenář bude mít možnost na základě poznatků zde uvedených zpětně rekonstruovat scénáře v programu OPNET Modeler. Rád bych hned v úvodu této práce upozornil, že základním předpokladem pro případnou rekonstrukci popsaných simulací je, že čtenář má s programovým rozhraním OPNET Modeler již nějakou zkušenost - vzhledem k rozsahu práce není možné čtenáře seznamovat se základním ovládáním a funkcemi programu.

[1] stará se o směrování v síti a adresaci

[2] Referenční model ISO/OSI vypracovala organizace ISO jako hlavní část snahy o standardizaci počítačových sítí nazvané OSI a v roce 1984 ho přijala jako mezinárodní normu ISO 7498.

[3] IP Differentiated Services

V simulačním prostředí vytvořím rozšířený model sítě, který bude provádět přenos dat na základě zkoumaného protokolu MPLS. Pilotní veličinou pro simulaci bude hodnota parametru **jitter**[4] v závislosti na času během hlasového hovoru Voice Over IP sestaveného mezi dílčími prvky modelované sítě. **Jitter** bude podroben zkoumání z pohledu řízení kvality služeb.

Další problematikou, kterou se bude tato diplomová práce zabývat, bude problematika výpadku linky v sítích MPLS a vliv tohoto výpadku na vlastní komunikaci v rámci sítě. Simulace a zkoumání tohoto jevu bude vycházet z předcházejícího scénáře - tedy zkoumání parametru **jitter** během probíhajícího hlasového VoIP hovoru. Jelikož nedílnou součástí moderních MPLS sítí je i oblast Traffic Enegeeringu (TE), zaměří se tato práce rovněž na možnost budování MPLS TE tunelů v rámci rozšířeného modelu sítě. V diplomové práci seznámím čtenáře s vytvořením TE tunelu mezi virtuálními pobočkami firmy, které budou úmyslně rozmístěné v různých geografických lokalitách. Simulace MPLS TE tunelu bude mít za úkol prakticky demonstrovat teoretické předpoklady TE tunelování.

[4]kolísání velikosti zpoždění paketů při průchodu sítí, které vzniká např. na směrovačích jako důsledek změn směrování, chování interních front směrovačů a podobně

1 KVALITA SLUŽEB - QOS

1.1 Definice

Zajištěním kvality služeb (dále jen QoS) se rozumí široká kolekce síťových technologií a technik, jejichž cílem je poskytovat garanci schopnosti sítě dodávat dané síťové službě předem zjistitelné parametry (výsledky) jako jsou např. dostupnost, propustnost, zpoždění, chybovost a podobně. Jak je uvedeno v [4] QoS zavádí do komunikačních sítí priorizaci síťového provozu s jehož pomocí se dosahuje výše uvedeného cíle tohoto mechanismu. Záruky QoS jsou zajímavé zejména v případech, kdy se setkáváme s omezenou nebo dokonce proměnlivou kapacitou sítě, která by plynule umožňovala přenos real-time aplikací, jakými jsou např. VoIP[1], IPTV[2] nebo pokud nelze kapacitu sítě rozumným způsobem rezervovat. Velice často real-time aplikace vyžadují pro svůj přenos konstantní bitovou rychlost (analogie CBR[3] v ATM[4]) a jsou velice citlivé na změnu zpoždění při přenosu. Nasazením mechanismu QoS můžeme výše zmíněné požadavky real-time aplikací velice spolehlivě uspokojit.

1.2 Faktory

Dle [9] výpočetní kapacity neumožňovaly vývojářům v prvopočátcích Internetu zahrnovat řízení kvality do funkcí síťových prvků, směrovačů a podobně. Toto chování postupem času vedlo k samovolnému vzniku třídy zajištění kvality služeb, kterou dnes známe jako **best effort**, tedy poskytování těch nejlepších parametrů sítě, které jsou momentálně k dispozici. V dnešních paketově přepínaných sítích může během přenosu paketů od odesílatele k příjemci dojít k působení mnoha negativních vlivů:

- **Zpoždění**
  Z mnoha důvodů může paket během přenosu získat zpoždění (např. kvůli dlouhým vstupním frontám, náhradním směrovacím trasám apod.)

- **Jitter**
  Zpoždění se bude během přenosu nejrůznějším způsobem měnit (různou čekací dobou paketů ve frontách síťových prvků apod.)

[1] Voice over IP
[2] IP Television
[3] Constant bit rate
[4] Asynchronous transfer mode - standard pro vysokorychlostní síťovou architekturu

- **Chybovost**
  Může vznikat špatným směrováním, poškozením paketů při jejich přenosu sítí, apod.

- **Zahazování paketů**
  Nejčastějším případem zahazování paketů je přeplnění vstupních front směrovačů. Příjemce dat tuto skutečnost rozpozná a požádá odesílatele o nové zaslání zprávy, což ovšem generuje nemalé zpoždění v doručení paketu.

- **Doručení mimo pořadí**
  Sada paketů putující komunikační sítí může být směrována různými cestami, což může mít za následek doručení paketů v jiném pořadí než byly původně odeslány.

1.3 Třídy QoS

Právě v souvislosti s řízením kvality služeb a zavedení metody DiffServ[5] byla dle [10] v roce 2004 zavedena norma IEEE 802.1p, která zavedla vyjádření CoS[6] s pomocí tří bitů (viz tab. 1.1). Stejné vyjádření tříd kvality služeb přejímá dále i MPLS ve svých EXP bitech. Touto problematikou se blíže zabývá kapitola 3 této práce.

Tab. 1.1: Třídy kvality služeb

Třída	**Typ provozu**
0	Best Effort
1	Přenos na pozadí
2	Standardní
3	Významná zátěž (aplikace kritické pro firmy)
4	Kontrolovaná zátěž (streaming multimédií)
5	Video a hlas (zpoždění a jitter pod 100ms)
6	Přenosy rezervované na úrovni L3 (zpoždění a jitter pod 10ms)
7	Přenosy rezervované na úrovni L2 (nejmenší zpoždění a jitter)

1.4 Protokoly a algoritmy QoS

Kvalitu služeb můžeme charakterizovat několika způsoby z několika různých úhlů pohledu. Obecně lze říci, že QoS je schopnost elementu sítě (aplikace, uživatel,

[5]IP Differentiated Services
[6]Class Of Service

směrovač) poskytnout určitou konstantní úroveň a kvalitu datového toku v rámci sítě. Jak uvádí [4] lze zjednodušeně říci, že QoS se dělí na dva základní typy: 1) rezervace síťových prostředků (integrované služby IntServ) 2) priorizace (diferencované služby DiffServ). Dále je ještě možné QoS mechanismy rozdělit na základě toho, zdali jsou uplatňovány pro jednotlivé datové toky, nebo zdali je jejich funkce definována pro agregovanou skupinu více datových toků v rámci sítě. Protože množství jednotlivých mechanismů je velké a předmětem této práce není zkoumání kvality služeb do hloubky, uvádím pouze výčet možných QoS protokolů a algoritmů:

- Resource reservation protocol (RSVP)
- IP Integrated Services (IntServ)
- IP Differentiated Services (DiffServ)
- Resource reservation protocol - Traffic Engeneering (RSVP-TE)
- Multiprotocol label switching (MPLS)
- Asynchronous transfer mode - standard pro vysokorychlostní síťovou architekturu (ATM)
- Frame relay
- X.25
- IEEE 802.1p
- IEEE 802.1q
- IEEE 802.11e
- IEEE 802.11p

2 MPLS PROTOKOL

2.1 Základy MPLS

MPLS je standardizovaná technologie určená ke zrychlování toku síťového provozu, která zároveň umožňuje jeho kvalitnější a jednodušší správu. Jak je uvedeno v [3] a [5], MPLS využívá principu sestavení přenosové trasy pro danou sekvenci přenášených paketů, která je definována záhlavím takovéhoto paketu. Směrovače uvnitř MPLS sítě tak vyhodnocují směrování paketů na základě informací uložených v hlavičkách dat, což umožňuje mnohem rychlejší a efektivnější síťovou komunikaci. Přirovná-li tento princip k referenčnímu modelu OSI[1] dojde ke zpracování většiny paketů na úrovni spojové vrstvy (přepínání) namísto klasického zpracování na úrovni vrstvy síťové (směrování), kterou známe např. z TCP/IP [2]. Kromě rychlejšího zpracování paketů MPLS rovněž obsahuje přirozenou podporu QoS, což umožňuje jednodušší a levnější správu kvality služeb.
MPLS nese označení multiprotocol (tedy víceprotokolární), protože umí spolupracovat s nejrůznějšími technologiemi jako např. TCP/IP, ATM (schopnosti navázání MPLS na ATM je hojně využíváno např. u akademické sítě CESNET), Frame Relay[3] a dalšími.

2.2 Výhody

V klasických konvenčních IP sítích je směrování paketů prováděno na každém směrovači v rámci sítě a to na základě zcela samostatného a nezávislého rozhodování. Jakmile síťový paket dorazí na vstupní rozhraní směrovače, provede tento směrovač prohledání směrovacích tabulek, aby našel podle cílové adresy rozhraní, na které bude daný paket dále odeslán. Za účelem tvorby výše zmíněných směrovacích tabulek používá každý ze směrovačů některý ze směrovacích protokolů jako např. BGP[4], OSPF[5] a IS-IS[6]. Každý ze směrovačů provádí během transportu paketu sítí vždy stejné operace, aby zjistil jeho dalšího příjemce. Protože rozhodovací logika u klasického směrování je založena na základě adresy příjemce paketu, dochází k agregaci paketů směřujících ke stejnému cíli. V případě využití MPLS protokolu a

[1] Open system interconnection reference model
[2] Transmission Control Protocol / Internet Protocol
[3] technologie přepínání paketů v počítačových sítích
[4] Border Gateway Protocol
[5] Open Shortest Path First
[6] Intermediate system to intermediate system - protokol využívaný směrovači k určení nejlepší přenosové cesty

jeho rozšíření TE[7] k tomuto jevu nedochází, protože celá přenosová MPLS síť je rozdělena na jednotlivé segmenty mezi směrovači, které se chovají jako samostatné toky dat a tak je možné vhodně řídit případné toky, přenosové kapacity a zatížení linek. Stejně dobře tak můžeme ovlivnit parametry samotných přenášených dat. Jak bude zmíněno v dalším textu, MPLS přiřazuje dle třídy paketu hlavičku pro průchod sítí pouze jednou, a proto je následné směrování mnohem efektivnější, rychlejší a bez zbytečné nadbytečnosti.

2.3 Stavební prvky

- **Label Edge Router**
  Taktéž známý jako MSR[8]
  Jedná se o zařízení, které pracuje na rozhraní přístupové sítě a sítě MPLS. Tento typ zařízení podporuje několik různých síťových rozhraní přístupových sítí (typicky Frame Relay, ATM, Ethernet) a předává dále provoz z/do těchto sítí z/do nitra MPLS sítě. LER hraje klíčovou roli v sítích MPLS, neboť je zodpovědný za přidávání a odstraňování MPLS značek (tedy modifikace hlaviček) jednotlivým paketům/rámcům tak, jak vcházejí nebo opouštějí síť MPLS směrem z/do přístupové sítě. Pokud se jedná o vkládání značky MPLS hovoříme o tzv. ingress LER (nebo také vstupním hraničním směrovači) - v případě odstraňování značky se bavíme o egress LER (nebo také výstupním hraničním směrovači).

- **Label Switch Router**
  Jedná se o vysokorychlostní směrovač uvnitř infrastuktury MPLS sítě. Podílejí se na sestavování cest s pomocí patřičného signalizačního protokolu a následně přepínají data na základě takto sestavených cest.

- **Label Switched Path**
  Jedná se o sekvenci značek, která je dostupná na každém prvku MPLS sítě přes který budou přenášena data od odesílatele směrem k příjemci. Sekvence mohou být vytvořeny na základě odesílání dat sítí nebo na základě datových toků. Značky, které přenášejí specifické identifikátory, jsou distribuovány v síti s pomocí distribučního protokolu LDP[9], RSVP, OSPF[10] či BGP[11]. Každý paket, který je přenášen sítí MPLS, má připojenu MPLS značku, která má pevně

[7] Traffic Engeneering
[8] Multiservice edge směrovač
[9] Label Distribution Protocol
[10] Open Shortest Path First
[11] Border Gateway Protocol

stanovenou délku a je umístěna přímo na začátku každého paketu či buňky. Tyto vlastnosti umožňují prvkům LSR velice rychle značku identifikovat v hlavičkách dat/rámců a přepínat tak data mnohonásobně vyšší rychlostí, než je tomu u běžně známých sítí s přepínáním datových jednotek.

2.4 Forward equivallence class

Je reprezentací skupiny paketů/rámců, které sdílejí stejné požadavky na přenos sítí, tj. všechny pakety v takovéto skupině jsou zpracovávány stejnými mechanismy. V porovnání s klasickým IP směrováním je v případě MPLS prováděna klasifikace paketů do příslušné třídy FEC (Forward Equivallence Class) pouze jednou, a to právě na hranici při vstupu do sítě MPLS (LER). Každý LSR si vytváří svou vlastní tabulku, podle které rozhoduje, jakým způsobem budou data předávána okolním prvkům sítě. Takováto tabulka je nazývána Label Information Base (LIB) a skládá se z provázání FEC s příslušnou značkou.

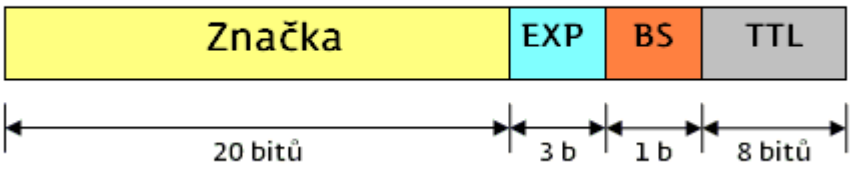
2.5 Značky a jejich přiřazování, MPLS hlavička

Jak uvádí [2], MPLS značka (ve své nejjednodušší formě) označuje síťovou cestu, kterou bude paket směrován. Značka je obsažena v MPLS hlavičce (viz obr. 2.1), která je přidána mezi linkovou a síťovou část standardní části hlavičky přenášených dat. Směrovač, který aktuálně zpracovává příchozí data, prozkoumává značku v MPLS části hlavičky, aby zjistil a určil další směrování dat. Značky mají pouze lokální význam vzhledem k LSR - vztahují se pouze na lokální cesty mezi jednotlivými LSR.
Kromě 20-bitové značky obsahuje MPLS hlavička 3 experimentální bity, které jsou používány pro účely QoS, dále 1 S bit (bottom-of-stack indikátor) a 8 bitů vyjadřujících MPLS TTL[12] indikaci. MPLS TTL je velice podobný hondotě TTL, kterou známe z IP sítí (jakmile dosáhne TTL nulové hodnoty jsou rámce automaticky zahozeny).

2.6 Základní principy signalizace

- **žádost o značku (label request)**
  S použitím tohoto mechanismu LSR požádá o značku svého následujícího souseda tak, aby tuto informaci mohl přiřadit konkrétní třídě FEC. Tento

[12]Time To Live

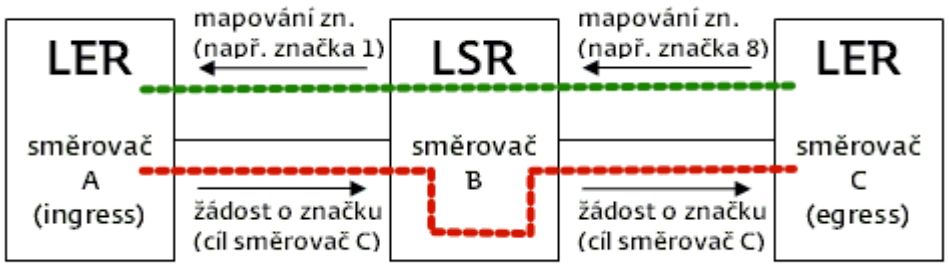


Obr. 2.1: Obecný formát MPLS hlavičky

mechanismus je funkční v celém rozsahu sítě MPLS až po hraniční LER směrovače.

- **mapování značky (label mapping)**
  Následující LSR odešle značku jako odpověď na požadavek, který byl vyvolán předcházejícím LSR partnerem.

Výše uvedený princip signalizačního mechanismu detailně popsaný v [3] je uveden na obr. 2.2.



Obr. 2.2: Princip signalizačního mechanismu MPLS

2.7 Signalizační protokoly

Signalizace je mechanismus s jehož pomocí si jednotlivé směrovače vyměňují informace důležité pro chod sítě. V sítích MPLS závisí typ přenášených informací mezi jednotlivými směrovači na signalizačním protokolu, který se konkrétně využívá. Podmínkou však zůstává, že MPLS značky musí být distribuovány sítí ke všem MPLS směrovačům. Architektura MPLS protokolu neurčuje žádný konkrétní signalizační protokol a proto byly doposud specifikovány následující signalizační protokoly:

- Label Distribution Protocol (LDP)

- Resource Reservation Protocol - rozšíření pro MPLS (RSVP-TE)
- Constrained Routing s LDP (CR-LDP)
- Distribuce značek s pomocí BGP-4

2.7.1 LDP

Protokol LDP byl dle [20] vyvinut pracovní skupinou IETF[13] specificky za účelem distribuce MPLS značek a sestavování cest LSP v rámci MPLS sítě. LDP pracuje na principu směrovacího protokolu IGP[14] a vždy vybírá stejnou fyzickou cestu MPLS sítí - stejně jako klasické IP směrování. LDP protokol neumožňuje rozšíření MPLS s pomocí TE.

2.7.2 CR-LDP

Jak uvádí [21], je protokol CR-LDP rozšířením původního LDP protokolu o podmínky, které musí být na dané lince splněny tak, aby došlo k výběru této linky jako potenciálního nositele cesty LSP. Podmínkou může být například volná šířka přenosového pásma nebo šifrování linky a podobně. V souvislosti s CR-LDP byly definovány čtyři typy LDP zpráv pro výměnu informací a MPLS hlavičky:

Discovery - registruje a udržuje registraci směrovače LSR v rámci MPLS domény. Tato zpráva je odeslána prostřednictvím HELLO zprávy a protokolu UDP[15] na všesměrovou adresu dané MPLS sítě

Session - je určena k sestavení, údržbě a rušení relací mezi jednotlivými sousedy v rámci LDP

Advertisment - vytváří, mění a ruší mapování MPLS značek pro danou třídu FEC

Notification - poskytuje informace o stavu, provádí diagnostiku popř. poskytuje informace o chybách

[13] Internet Engineering Task Force
[14] Interior Gateway Protocol
[15] User Datagram Protocol

2.7.3 RSVP-TE

Protokol RSVP-TE [15] je nadstavbou původního protokolu RSVP [16], který využívá jeho mechanismů k distribuci MPLS značek, k sestavování LSP cest v rámci MPLS sítě a splňuje požadavky rozšíření MPLS TE. Cesty sestavené s pomocí RSVP-TE musí být dle [15] neustále aktualizovány, aby nedošlo ke zrušení rezervace síťových prostředků, což jej odlišuje od předchozích zmíněných signalizačních protokolů. RSVP-TE je takzvaným reverzním rezervačním protokolem, což znamená, že žádost o rezervaci síťových prostředků je odesílána od příjemce paketů směrem k jejich odesílateli. RSVP-TE protokol započne sestavení LSP cesty v síti tím, že odešle PATH zprávu s pomocí navrhované cesty, ve které by dále mělo dojít k sestavení cesty LSP. Každý z páteřních směrovačů, který leží na této cestě, musí provést kontrolu, zdali je z jeho strany možno danou cestu sestavit a v případě, že to možné není, musí ve zpětném kanálu odeslat vstupnímu hraničnímu směrovači chybovou zprávu typu ERROR, která jej o tomto stavu informuje. V opačném případě (tedy, že je možno cestu sestavit) je daným směrovačem PATH zpráva předána dalšímu sousednímu směrovači, který provede rovněž tuto kontrolu tak, až se zpráva dostane k výstupnímu hraničnímu směrovači. Jakmile se tak stane, výstupní hraniční směrovač odešle ve zpětném kanálu zpět vstupnímu hraničnímu směrovači zprávu typu RESV. Zpětný kanál kopíruje původní cestu, která byla sestavena s pomocí zprávy PATH. Součástí zpětného kanálu a zasílání zpráv RESV je rovněž umístění MPLS značek na trasách mezi jednotlivými směrovači. Na jednotlivých směrovačích nedochází k rezervaci síťových prostředků do té doby, dokud není obdržena zpráva typu RESV. Aby bylo možné využívat funkcí Traffic Engeneeringu, byl původní RSVP protokol rozšířen dle [15] o další objekty, což vedlo k přímé definici RSVP-TE.

ERO[16] **objekt** - je součástí zprávy typu PATH. Pokud je definován je sestavení LSP určeno tímto objektem, nikoliv aktuální nejkratší cestou v síti. Tento objekt se sestává z objektů TLV[17] a každý z LSR směrovačů, který je součástí LSP, je zakódován do jednoho z objektů typu TLV

RRO[18] **objekt** - je součástí zpráv typu PATH a RESV. Součástí tohoto objektu jsou záznamy o všech směrovačích LSR a o informaci v jakém pořadí je přes ně sestavena cesta LSP.

[16]Explicit Route Object
[17]Type Length Value
[18]Record Route Object

SA[19] **objekt** - je součástí zprávy typu PATH a obsahuje informace k indetifikaci jednotlivých relací a k popisu jejich charakteristik (např. priorita pro danou cestu LSP).

2.7.4 BGP-4

Protokol BGP může být také použit pro distribuci MPLS značek v rámci sítě. Standardně je BGP protokol využíván k výměně směrovacích informací mezi jednotlivými autonomními sítěmi rozsáhlejších sítí.

2.8 Cesta sítí

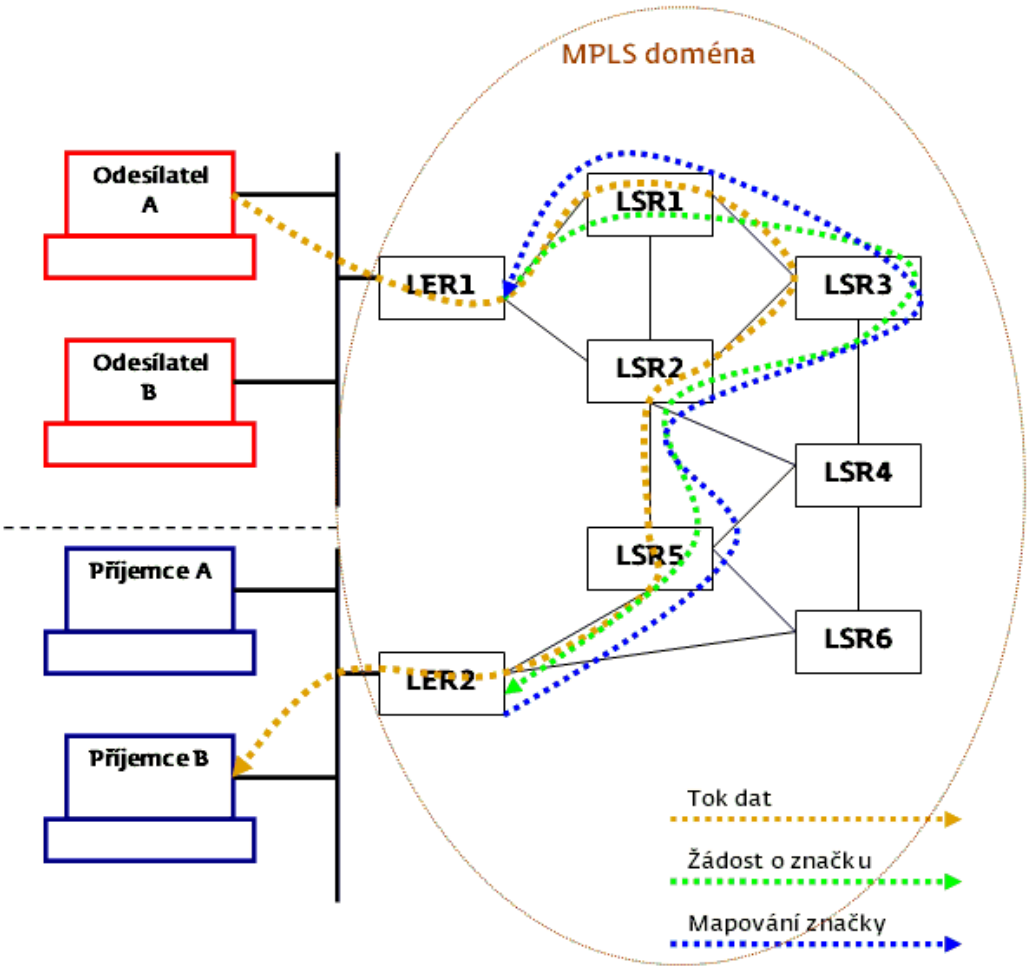
Aby datový paket korektně prošel celou MPLS doménou od odesílatele směrem k příjemci, musí být provedeny následující kroky:

- vytvoření značek a jejich distribuce
- vytvoření tabulky LIB na každém ze směrovačů
- vytvoření přenosových cest - LSP
- vložení značky do MPLS hlavičky (LER)
- předávání paketů sítí
- odstranění MPLS části hlavičky a předání dat příjemci (LER)

V rámci jedné MPLS domény neplatí, že všechna data od stejného odesílatele, která jsou určena pro stejného příjemce, jsou přenášena jednou totožnou přenosovou cestou. V závislosti na charakteristikách daného přenosu mohou být vytvářeny různé přenosové cesty pro pakety s různou CoS [20]. Na obr. 2.3 je vyobrazena ilustrace cesty dat skrz síť MPLS.

Přehled jednotlivých MPLS operací a jejich sledu pro přenos dat v rámci MPLS domény je vyobrazen v tab. 2.1.

[19] Session Attribute
[20] CoS - Class of Service - viz Třídy QoS 1.3



Obr. 2.3: Vytváření LSP a předávání dat v rámci MPLS domény

Tab. 2.1: MPLS protokol - aktivity při přenosu

Aktivita	**Popis**
tvorba a distribuce značek	- Ještě před přenosem samotných dat musí provést hraniční směrovače rozhodnutí, jaký typ značky bude přiřazen příslušnému typu FEC. Na základě tohoto rozhodnutí jsou sestaveny LIB tabulky. - Sousední směrovače započnou distribuci značek a mapování značka/FEC a zároveň dochází k přenosu charakteristik provozu s pomocí protokolu LDP.
vytvoření LIB tabulek	- Jakmile dojde k příjmu mapování značka/FEC je vytvořen záznam v LIB tabulce každého LSR.
Pokračování tabulky na další straně	

Tab. 2.1 – pokračování tabulky z předchozí strany

Aktivita	Popis
sestavení cesty LSP	- Jak vyplývá z obr. 2.3, dochází k sestavení LSP v opačném pořadí, než v jakém byly přidávány položky do LIB.
práce se značkami	- LER směrovač využije LIB tabulku, aby určil následující síťový skok a vyžádá si značku pro daný FEC. - Sousední směrovače použijí značku k nalezení dalšího síťového skoku. - Jakmile se paket dostane k výstupnímu LER směrovači MPLS sítě, MPLS hlavička je z paketu odstraněna a paket je odeslán do cílové přístupové sítě.
průchod paketu sítí	S odkazem na obr. 2.3 budeme blíže zkoumat cestu paketu sítí od odesílatele A k příjemci B: - Směrovač LER1 nemá přiřazenu žádnou značku pro tento paket, protože se jedná o první výskyt takovéhoto požadavku v síti. Další síťový skok je LSR1. - Směrovač LER1 odešle žádost o značku sousednímu směrovači LSR1. - Výše uvedený požadavek bude propagován sítí dále, jak je naznačeno zelenou přerušovanou čarou. - Každý sousední směrovač obdrží značku od svého následujícího sousedního směrovače počínaje hraničním směrovačem LER2 směrem k předchůdcům až po LER1. Sestavení LSP cesty je označen hnědou přerušovanou čarou s pomocí protokolu LDP. - Směrovač LER1 vloží značku do hlavičky MPLS a předá paket směrovači LSR1. - Každý následující LSR (LSR3, LSR2, LSR5) prozkoumá značku v MPLS hlavičce přijatého paketu, nahradí tuto značku odchozí značkou a předá síti k dalšímu zpracování. - Jakmile paket doputuje na výstupní hraniční směrovač LER2, dojde k odstranění MPLS hlavičky protože paket bude opouštět MPLS doménu a k předání paketu dále do přístupové sítě směrem k příjemci.

2.9 MPLS a detekce chyb

Detailní terminologie v souvislosti s metodami zotavení po pádu páteřních tras v rámci MPLS domény je detailně popsána v RFC 3469 [6]. V této části diplomové práce chci zdůraznit, že MPLS představuje novou vrstvu v síťovém modelu, která je naprosto nezávislá na dalších vrstvách, které jsou technologicky využity při výstavbě rozšířených sítí. Vzhledem k tomuto faktu se musela MPLS technologie potýkat s problémem vytvoření vlastních mechanismů určených pro detektci chyb, protože vzhledem k rozdílnosti protokolů dílčích nižších vrstev nebylo možné tyto detekční mechanismy využívat.

2.9.1 RSVP-TE HELLO

Rozšíření RSVP-TE HELLO (nebo také žádost o pozdrav) umožňuje dle [15] jednotlivým směrovačům LSR detekovat situace, kdy jejich sousední směrovače nejsou dostupné. Toto rozšíření se skládá z jednotlivých dílčích fází žádosti o pozdrav (HELLO REQUEST) a povrzení žádosti o pozdrav (HELLO ACK). Zpracování jednotlivých zpráv mezi sousedními LSR směrovači umožňuje administrátorovi sítě nastavení detekčního intervalu, který bude reprezentovat časovou prodlevu (hysterezi) po jejímž uplynutí bude tento mechanismus spuštěn. Typická standardní hodnota bývá nastavena na 5 ms. Detekce chyb v tomto mechanismu je dále doprovázena ukládáním tzv. instancí sousedních LSR směrovačů. Pokud je detekována změna v hodnotě HELLO instance, sousední směrovač je považován za restartovaný. Každý LSR směrovač pravidelně generuje žádosti o pozdrav obsahující HELLO REQUEST, který zároveň obsahuje i číslo instance a dále číslo instance, které bylo v předchozí HELLO REQUEST žádosti přijato od sousedního směrovače. Jakmile sousední směrovač přijme tuto HELLO REQUEST zprávu, porovná hodnotu instance odesílatele s poslední přijatou hodnotou instance od tohoto LSR směrovače. Zároveň dochází ke kontrole, že odesílatel vrací poslední číslo instance, které bylo použito přijímačem. Pokud některé z uvedených hodnot nesouhlasí (jsou rozdílné), pak je sousední směrovač považován za restartovaný. V případě, že jsou přijaté hodnoty v pořádku, příjemce vygeneruje jako odpověď zprávu HELLO ACK, která obsahuje hodnoty instancí odesílatele a příjemce - v opačném případě odesílatel zprávy HELLO ACK nastaví hodnotu instance příjemce na nulu, čímž dojde zpětně k signalizaci, že komunikace mezi sousedními dvěma směrovači byla přerušena. V případě, že nedojde v administrátorem konfigurovaném počtu HELLO intervalů k výměně žádosti o pozdrav, pak daný směrovač bude považovat sousední směrovač za nedostupný. HELLO protokol je navržený pro detekci nedostupnosti popř. restartu sousedního LSR směrovače - tedy dochází ke

zjištění, zdali je sousední LSR aktivní a spojení k němu jdoucí je funkční - neslouží však k ověřování funkčnosti plné přenosové cesty.

2.9.2 RSVP-TE Softstate

Tento mechanismus detekce chyb vychází dle [15] ze znalosti, že RSVP-TE je stavový protokol, který neustále odesílá PATH a RESV zprávy v rámci dané LSP. Pokud PATH nebo RESV zprávy selžou v souvislosti s aktualizací přenosové cesty, dojde z místa, kde došlo k selhání rezervace směrem ke vstupnímu směrovači LSP, k odeslání PathErr nebo ResvErr zprávy. Jako každý ze signalizačních mechanismů má i tento svoji nevýhodu - a to že je interval, ve kterém se stavové zprávy opakují, nastaven často na vyšší hodnotu, aby na přenosových trasách došlo k minimalizaci kontrolního provozu sloužícího k signalizaci. Standardní hodnota bývá často nastavena na 30 sekund, což je v souvislosti se zajištěním kvality služeb v MPLS sítích hodnota často naprosto nemyslitelná, takže takovýto mechanismus detekce chyb je pro rychlé přesměrování provozu nepoužitelný.

2.9.3 LSP Ping/ Traceroute

S využitím tohoto modelu detekce chyb [24] může být ověřeno, zda pakety, které náleží do specifické třídy FEC, skutečně končí svou přenosovou trasu LSR na správném výstupním směrovačí LSR. V módu PING je paket zapouzdřen do UDP paketu, který je dále směrován v rámci dané LSP jako jakýkoliv jiný paket náležející do dané FEC třídy. Tento paket se nazývá ECHO REQUEST (žádost o odpověď). Jakmile dorazí na výstupní směrovač LSR, je konfrontován s řídící vrstvou LSR a dojde k ověření, zdali tento LSR směrovač je skutečně výstupním směrovačem v rámci LSP pro danou třídu FEC. Jakmile dojde k pozitivnímu ověření, je tímto směrovačem vygenerován paket ECHO REPLY (odpověď) a je zaslán zpět výchozímu směrovači společně s výsledkem kontroly. V módu TRACEROUTE je paket porovnáván s řídící vrstvou každého směrovače LSR v rámci LSP, aby došlo k ověření, zdali je směrovač skutečně tranzitním směrovačem pro danou třídu FEC. TRACEROUTE mód je tedy vhodným mechanismem pro zjištění místa poruchy a často tak bývá efektivně kombinován s PING mechanismem. Jako každá metoda, má i tato nevýhodu v podobě potřeby vysokého výpočetního výkonu na směrovačích LSR a LER.

2.9.4 Elegantní restart

Dle [22] funguje tento princip na základě principu protokolu BGP definovaném v [23]. MPLS směrovače mají dvě základní vrstvy - řídicí vrstvu a datovou vrstvu. Každá z

těchto vrstev je řízena svým specifickým procesorem. Řídicí jednotka zpracovává na základě informací od okolních prvků sítě směrovací tabulky a poté je předává datové vrstvě, která je zodpovědná za jejich distribuci. Toto rozdělení funkcí umožňuje MPLS směrovačům předávání dat v rámci jejich právoplatných směrovacích cest, i když je řídicí vrstva směrovače momentálně nedostupná. Jakmile ovšem dojde k ukončení činnosti řídicí vrstvy, dojde zároveň k resetu řídicího kanálu mezi směrovačem a jeho okolními LSR, což způsobí ukončení všech cest LSP vedoucích skrze tento páteřní směrovač. Abychom tomuto 'neduhu' předešli, implementuje MPLS techniku nazvanou elegantní restart, který zavádí do HELLO zpráv RSVP-TE novou stavovou zprávu nazvanou RESTART_CAP. Zpráva obsahuje dvě přenášené hodnoty RESTART TIME (odpovídá časovému údaji, který reprezentuje dobu směrovače potřebnou pro restart řídicí vrstvy do doby schopnosti zasílat HELLO zprávy) a RECOVERY TIME (odpovídá časovému údaji, která reprezentuje dobu směrovače potřebnou pro re-synchronizaci RSVP a MPLS datové vrstvy, která následuje po úspěšném navázání HELLO synchronizace). Jakmile směrovač detekuje ztrátu komunikace s některým ze svých partnerů, tento směrovač vyčká po dobu stanovenou v RESTART TIME a až poté vyvolá procedury pro detekci chyb.

2.10 Mechanismy obnovy u MPLS

Stejně jako mechanismy obnovy u ostatních vrstev, mechanismus obnovy u MPLS funguje na základě principu směrování provozu novou směrovací cestou, která má za úkol obejít část cesty, která je momentálně nefunkční. Způsob, jakým je tato 'obchozí' cesta určena, kdy je kalkulována a jak je sestavena, záleží na konkrétním modelu obnovy, který je v daném okamžiku použit.

Pokud není definován žádný z mechanismů obnovy MPLS sítě, obnova je provedena signalizačním protokolem, který je určený k vytvoření a udržování cest LSP. Tento mechanismus obnovy je nazýván obnova metodou best-effort. V tomto případě může být porucha detekována zprávami HELLO stavovým mechanismem RSVP-TE. V případě, že klademe vyšší požadavky na obnovu MPLS sítě (rychlejší obnovení), musí být nasazeny další přídavné mechanismy obnovy, které mají za úkol redukovat čas potřebný k jejich funkci.

Dle [6] je jedním z klíčových parametrů obnovy v sítích MPLS schopnost sítě zajistit obnovení systému z poruchového stavu linky nebo dílčího směrovače s minimálním vlivem na data procházející touto sítí. Často zmiňovaným akceptovatelným limitem pro provedení funkce obnovy je 50 ms vzhledem k nejdelšímu akceptovatelnému

zpoždění, které ještě nebude mít vliv na hlasové přenosy a je zároveň použit jako limit pro obnovu v sítítch SONET[21] / SDH[22]. V případě, že použijeme níže zmiňovaný mechanismus obnovy přesměrováním, bude čas obnovy záležet na času objevení chyby, času notifikace poruchy, času výpočtu záložní cesty a na času sestavení nové záložní trasy. Všechny tyto operace jsou pomalé a v součtu mohou trvat až několik jednotek sekund, což je pro řadu interaktivních aplikací nepřijatelným parametrem. Pokud je obnova provedena ochranným přepínáním, bude čas obnovy rapidně snížen, protože kalkulace pro výpočet záložních tras nebudou zapotřebí a nebude zapotřebí ani signalizace pro sestavení záložní cesty (cesta předpřipravena). Doba obnovy tak bude záležet pouze na detekci chyby a na době přenosu této informace. Technika s předsestavenou záložní cestou, která používá mechanismu lokální obnovy je často nazývána jako FAST REROUTE (rychlé přesměrování). Protože je tato metoda velice vstřícná vůči přenosu hlasu, který je předmětem simulace této diplomové práce, budeme se v simulacích zabývat užitím tohoto mechanismu a případného vlivu na dobu trvání obnovy a probíhající hlasové hovory.

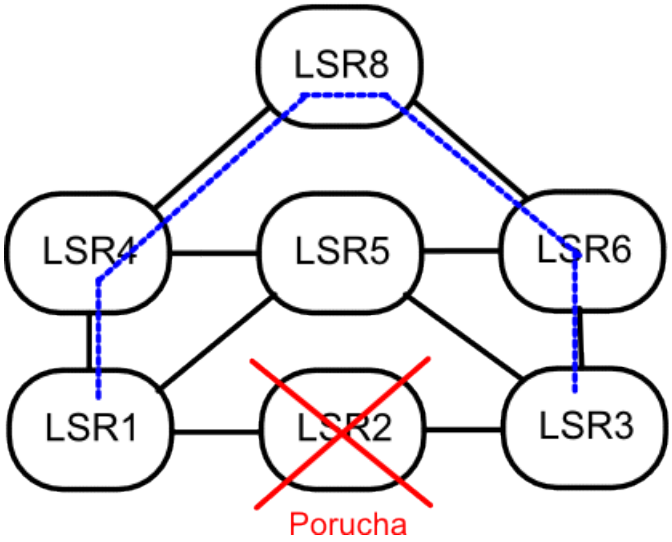
2.10.1 Globální obnova

Cílem mechanismu globální obnovy - viz obr. 2.4 - je ochrana vůči jakékoliv poruše linky či směrovače na cestě, popř. na její dílčí části - s vyjímkou chyb, které se mohou objevit na příchozím, popř. odchozím LSR směrovači. Záložní cesta v sítí MPLS s využitím mechanismu globální obnovy je až na zmíněné vyjímky naprosto odlišná od cesty původní - využívá tak jiné LSR a linky. Nevýhoda této metody je, že záložní cesta musí být propagována sítí až směrem ke vstupnímu LSR směrovači a to ještě před tím, nežli může mechanismus obnovy započít.

2.10.2 Lokální obnova

Oproti mechanismu globální obnovy je cílem lokální obnovy - viz obr. 2.5 - ochránit MPLS síť vůči výpadku linky popř. sousedního směrovače a tím minimalizovat potřebný čas pro propagaci této změny v rámci sítě. Mechanismus lokální obnovy se dělí na metody obnovy linky (nalezení nejkratší náhradní cesty) a obnovy směrovače (nalezení nejkratší obchozí cesty k překonání nefunkčního směrovače).

[21] Synchronous Optical Networking
[22] Synchronous Digital Hierarchy



Obr. 2.4: Globální obnova v síti MPLS

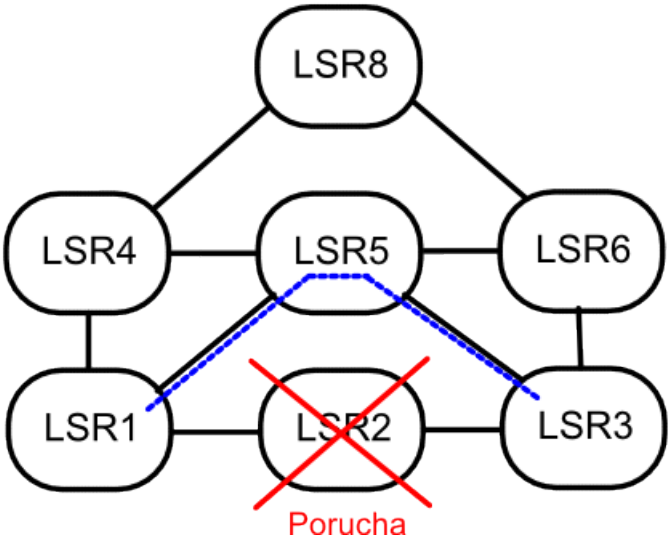
2.10.3 Obnova přesměrováním

V případě obnovy přesměrováním je záložní cesta sestavena na základě přítomnosti poruchy v síti. Jedná se o reaktivní chování sítě na základě detekce poruchy. Sestavení náhradní cesty může být zpracováno na základě detekce poruchy, politik směrování sítě a informací o topologii dané sítě. V okamžiku, kdy je sítí detekována porucha se nejbližší přecházející LER směrovač pokusí provést obnovu sítě tak, aby došlo k přemostění místa poruchy a další pokračování cesty LSP již bylo totožné s cestou původní. Náhradní cesta může být vypočítána v okamžiku poruchy nebo může být předpočítána a použita až v případě, kdy to podmínky v síti budou vyžadovat.

Mechanismus obnovy přesměrováním je dle [6] výhodný zejména v tom, že dokud nenastane porucha, nevyužívá žádné záložní prostředky sítě a tím nevzniká ani žádná nadbytečnost tohoto mechanismu. Nevýhodou tohoto mechanismu je skutečnost, že síťové prostředky vždy nemusí být v okamžiku poruchy sítě k dispozici.

2.10.4 Obnova ochranným přepínáním (Protection Switching)

V případě obnovy ochranným přepínáním je dle [6] záložní cesta předpočítána a předpřipravena (šířka pásma, fronty, zpracování) ještě před okamžikem, kdy porucha v síti fyzicky nastává. Vzhledem k tomu, že je již záložní cesta sestavená,



Obr. 2.5: Lokální obnova v síti MPLS

není třeba pro sestavení záložní cesty žádné signalizace a tak je tento mechanismus rychlejší než u obnovy přesměrováním. V souvislosti s metodou obnovy ochranným přepínáním rozeznáváme dva typy:

1+1 - síťové prostředky v rámci záložní cesty jsou plně rezervovány sítí a předpřipraveny pro toto využití. Tento mechanismus je velice rychlý, ovšem představuje vícenáklady v podobě rezervovaných, přídavných, síťových prostředků.

1:1 - síťové prostředky v rámci záložní cesty mohou být v případě neporuchového stavu používány sítí k běžným přenosům pro data s nízkou prioritou, která mohou být v kterémkoliv okamžiku zahozena. Mechanismus reaguje rychleji nežli 1+1, nicméně tato metoda je finančně velice efektnivní, protože do okamžiku poruchy nepředstavuje žádné vícenáklady či redudantní technologie.

2.10.5 Techniky pro výběr náhradní cesty

MPLS protokol umožňuje využití dvou základních technik pro sestavení náhradní cesty v rámci sítě - spojování a stohování. V obou případech je sestavení záložní cesty prováděno směrem od směrovače PSL[23] ke směrovači PML[24], ovšem s rozdílem zpracování MPLS značek mezi těmito směrovači.

[23]Path Switch LSR - směrovač LSR v topologii za poruchou, který je zodpovědný za přepínání nebo přesměrování provozu z pracovní cesty na cestu záložní

[24]Path Merge LSR - směrovač LSR zopodvědný za propojení nové záložní cesty s částí původní funkční přenosové trasy

V případě **spojování** provede PSL změnu směrovací tabulky na základě vytvoření záložní cesty tak, aby došlo k předávání paketů směrem do záložního směru namísto původního nefunkčního směru. PSL směrovač v tomto případě využije nové odchozí síťové rozhraní, pro které přiřadí novou MPLS značku. Jakmile jsou pakety přijaty PML směrovačem, dojde ke spojení dat z této záložní cesty s původní funkční cestou LSP a data putují směrem k příjemci.

V případě **stohování** PSL směrovač rovněž provede úpravu svých směrovacích tabulek na základě vzniku záložní cesty. Stejně jako u předchozí metody dojde k využití nového síťového rozhraní PSL směrovačem a k nezbytnému přiřazení nové MPLS značky. V tomto případě, ale dojde k přiřazení nové značky nad značku původní, která je zde ponechána a data jsou přenášena pomocí náhradní cesty směrem k PML směrovači, kdy na předposledním LSR směrovači v rámci náhradní cesty dojde k odstranění nové značky, takže data již na PML dorazí se značkou původní. Jakmile PML směrovač přijme takovýto paket, není si vědom, že došlo k výpadku v síti, protože hodnota značky je totožná se značkou původní cesty. Pro správnou funkci metody stohování musí být splněny dvě následující podmínky:

1. Směrovač PML musí využívat registr globálních značek (nedojde k asociaci příchozí značky s konkrétním síťovým rozhraním - pakety jdoucí ze záložní cesty přijdou se stejnou značkou, ale z jiného rozhraní)

2. Směrovač PSL musí znát značku, která by byla využita pro přenos dat standardní cestou směrem k PML.

Metoda stohování může být použita k obnově jedné nebo i více cest LSP.

2.11 MPLS Traffic engineering

MPLS Traffic engineering (MPLS TE) je dynamickou realizací řízení kvality služeb v dnešních sítích poskytovatelů služeb. Poptávka po integraci MPLS TE do sítí poskytovatelů datových sítí je způsobena právě výbornými vlastnostmi MPLS TE, které umožňují přebírat a dále rozšiřovat vlastnosti klasických spojových protokolů. MPLS používá informace o dostupnosti sítě, které jsou poskytnuté protokoly síťové vrstvy.

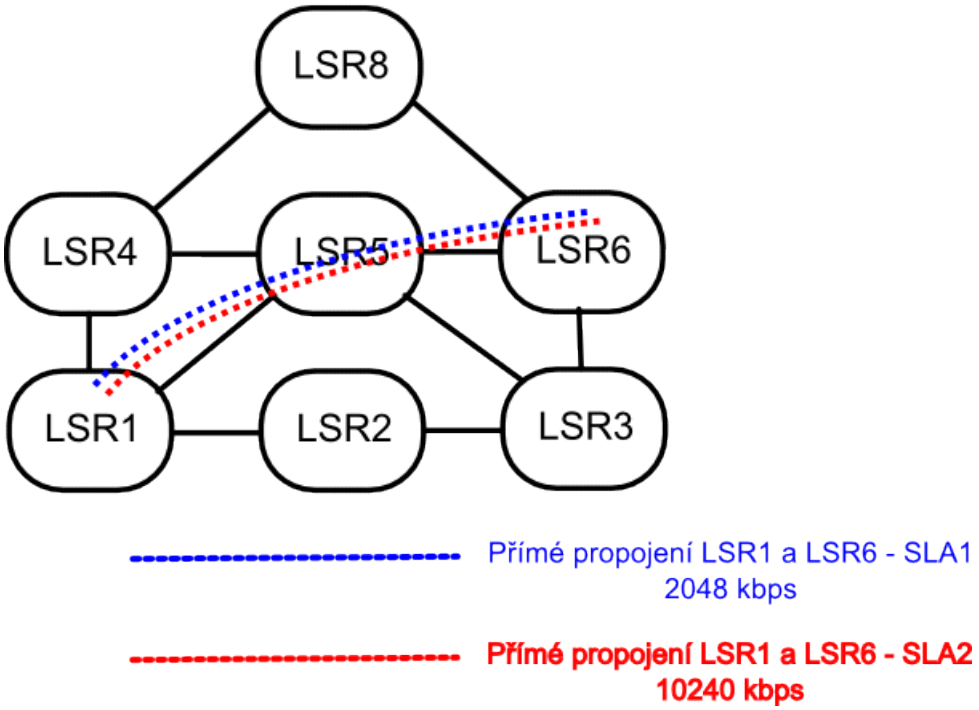
Traffic engineering jako takový - bez návaznosti na MPLS - je viz [12] způsob ovládání parametrů datové sítě tak, aby plně vyhovovala přenášenému provozu. Nezávisle na tom jak pečlivý administrátor sítě bude, vždy se budou očekávání o

datových tocích v rámci sítě odlišovat od reality. V reálném životě nastávají zcela neočekávané situace, jako např. nepřiměřené požadavky na toky dat nebo výpadky páteřních linek. V závislosti na těchto jevech může docházet ke stavům, kdy jsou některé linky sítí naprosto přetížené zatímco ve stejném čase mohou být jiné linky v rámci stejné sítě naprosto nevyužité. Traffic engineering je umění řízení datových toků takovým způsobem, že část provozu z přetížených linek přesměrujeme na linky nevyužité a vyřešíme tak mnohdy problematickou situaci.

Hlavní výhodou implementace MPLS TE je kombinace vlastností ATM TE společně s kategorizací CoS (Class of Service), využívanou u IP protokolu. V klasických IP sítích dochází ke směrování IP paketů na tzv. per-hop principu (skokově mezi jednotlivými uzly), přičemž vyhledávání cesty je prováděno na každém z jednotlivých průchozích směrovačů v rámci cesty paketu. Tento princip přenosu dat vede k neefektivnímu využití přenosových tras mezi jednotlivými směrovači sítě (linky s lepšími váhami jsou plně využité, jiné naprsoto nevyužité). K tomuto účelu je primáně nasazován MPLS TE, který si klade za cíl rozložení přenášeného datového toku v závislosti na zatížení jednotlivých linek, typu provozu a podobně.

2.11.1 MPLS Tunely

Nedílnou součástí TE je možnost sestavování TE tunelů [11] - viz obr. 2.6 - za účelem efektivního řízení šířky pásma, kvality služeb, či poskytování separátní VPN konektivity na bázi SLA v rámci páteřní sítě pro sub-odběratele. TE tunely jsou konfigurovány vždy jako jednosměrné elementy, tedy pokud potřebujeme propojit dva směrovače TE tunelem, bude vždy zapotřebí vytvořit separátní linku pro každý z uvažovaných směrů.



Obr. 2.6: MPLS TE tunel

MPLS TE může být rovněž používáno pro alokaci tříd provozu (CoS) oproti cíli datového toku, ke kterému pakety směřují. TE tunely si můžeme představit jako datové toky, které spojují konkrétní zdroje dat s jejich cíli a tyto tunely mohou mít parametry nebo vlastnosti, které jsou s těmito objekty přímo spojeny. Atributy, které je možné asociovat s TE tunely mohou obsahovat požadavky na šířku přenosového pásma a třídu CoS, která bude použita pro přenos dat v rámci takovéhoto tunelu. Data jsou předávána v rámci cesty definované jako TE tunel s použitím MPLS protokolu. TE tunely nejsou nikdy svázány se specifickou cestou v rámci domény. TE tunely mohou přesměrovat pakety jakoukoliv z cest, která je využita v rámci MPLS LSP.

2.11.2 Traffic Engeneering v souvislosti s QoS

MPLS DS-TE[25] je implementací, která umožňuje detailní kontrolu datových toků za účelem snížení přetížení přenosových tras, snížení zpoždění a celkovému zlepšení výkonnosti MPLS sítě. DS-TE ve svém jádru zachovává již dříve popisované vlastnosti MPLS TE (distribuce informací o linkách, signalizace, výpočet cest), ale zároveň přidává rozšíření, které umožňuje řízení a rozhodování na základě rozdělení provozu do několika tříd. Rošíření DS-TE protokolu je blíže specifikováno v [17] a v této práci si jej blíže popíšeme.

Mechanismy DiffServ a DS-TE pracují na základě alokace šířky pásma jednotlivých přenosových tras dle rozdělení a klasifikace do QoS tříd. Zatímco protokol DS-TE fuguje v případě MPLS TE jako řídicí, DiffServ funguje jako směrovací vrstva. Z výše uvedeného poznatku vyplývá, že konfigurace řídicí a směrovací vrstvy se tak budou vzájemně ovlivňovat a budou spolu úzce souviset. Toto ovšem neznamená, že konfigurace obou vrstev musí být totožné. Jednotlivé vrstvy mohou využívat rozdílný počet tříd a rozdílné alokace šířky pásma.

DS-TE využívá koncepce klasifikace provozu do jednotlivých tříd CoS (v DS-TE terminilogii nazývané **CT**) a to v rozsahu celkem osmi tříd a zavádí nový CLASSTYPE RSVP objekt, který je nositelem informace, která obsahuje asociaci mezi CoS a cestami TE LSP. Jednotlivé směrovače DS-TE musí podporovat tento objekt a musí ho vkládat do PATH zpráv protokolu RSVP-TE s vyjímkou CoS o hodnotě 0, která je vyhrazena pro MPLS sítě nepodporující DS-TE rozšíření.

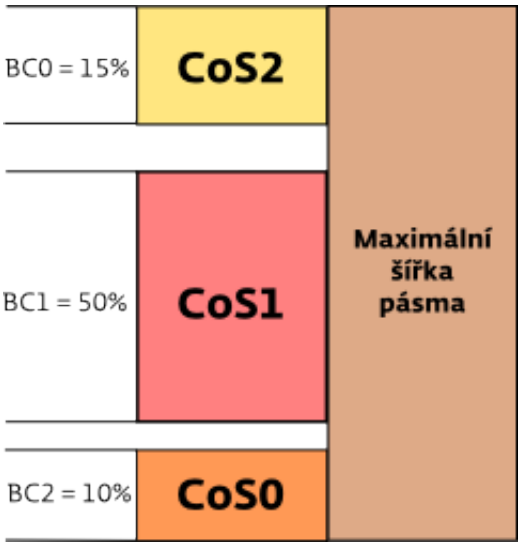
[25] DiffServ-Aware Traffic Engeneering

2.11.3 Omezení šířky pásma - Bandwidth Constriants

Sada podmínek pro omezení šířky pásma definuje pravidla, která LSR směrovače používají při alokaci šířky pásma na jednotlivých přenosových linkách na základě hodnoty CoS. Každá linka v rámci DS-TE sítě má definovanou sadu podmínek pro omezení šířky pásma, která odpovídá CoS, které se v daném místě v rámci sítě používají. Tato sada může obsahovat až 8 podmínek pro omezení šířky pásma. DS-TE může používat různé modely pro omezení šířky pásma, přičemž zde uvádím dva nejpoužívanější - Maximum Allocation Model a Russian Doll Model.

2.11.4 Maximum Allocation Model - MAM

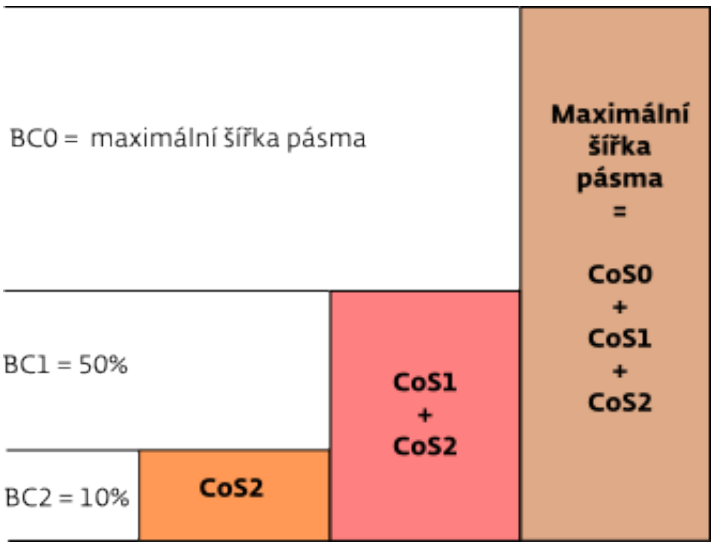
MAM definuje asociaci 1:1 mezi omezením šířky pásma a CoS. MAM nabízí omezenou šířku pásma (určenou maximální šířkou přenosového kanálu), která může být sdílena mezi jednotlivými třídami CoS. Jednotlivé třídy CoS nemohou v tomto případě využívat volnou šířku pásma nevyužitou ostatními třídami CoS. Grafická reprezentace modelu MAM je vyobrazena na obr. 2.7. Více informací o tomto modelu je k nalezení v [18].



Obr. 2.7: Model řízení šířky pásma - MAM

2.11.5 Russian Doll Model - RDM

RDM definuje souhrnnou sadu podmínek, které spoluutvářejí jednotlivé třídy CoS. Třída BC0 vždy definuje maximální šířku přenosového kanálu přes veškeré třídy CoS. Jednotlivé třídy CoS mohou v tomto případě využívat volnou šířku pásma nevyužitou ostatními třídami CoS. Grafická reprezentace modelu RDM je vyobrazena na obr. 2.8. Více informací o tomto modelu je k nalezení v [19].



Obr. 2.8: Model řízení šířky pásma - RDM

3 IMPLEMENTACE QOS V MPLS

Všechny MPLS QoS implementace vycházejí z již dříve zmiňovaných mechanismů DiffServ a DS-TE. V souvislosti s mechanismem DiffServ využívají páteřní směrovače tří bitů, které jsou nazývány z historických důvodů bity experimentálními a jsou umístěny v MPLS hlavičce každého z paketů procházejících MPLS sítí. Tato vlastnost nám umožní využívat až osm samostatných tříd kvality služeb (CoS). Pokud budeme brát v úvahu, že CoS 0 vyjadřuje standardní kvalitu dodávaných služeb, pak nám tedy zbývá 7 dalších tříd s různými priorizacemi kvality. Pokud bude naším cílem nabízet nasmlouvanou/nenasmlouvanou kvalitu služeb (podobně jako je tomu u Frame Relay sítí pomocí bitu DE nebo u ATM sítí pomocí bitu CLP (Cell Loss Priority)), pak jeden bit vyhradíme pro tento mechanismus a stávající dva bity nám umožní využívat pouze čtyř tříd CoS, což může být ale v řadě případů dostačující.

Klasifikace CoS bývá zpravidla prováděna na hraničních směrovačích (LER), které třídí provoz do MPLS tříd. Nejjednodušší algoritmus, který je pro tuto operaci určen a který je nazýván **uniformní** jednoduše modifikuje pole DSCP[1] v IP paketu a mapuje jeho hodnotu do hlavičky MPLS paketu. V případě, že naše aplikace či zákazník vyžadují zachování DSCP značek a nelze tedy použít uniformní režim, bude muset náš LER směrovač přímo vyhodnocovat objem a druh provozu a podle naměřených hodnot nastavovat CoS pro MPLS přímo. Informace o CoS v MPLS jsou pak přenášeny pouze mezi vstupními a výstupními hraničními směrovači v rámci MPLS domény. Tento mechanismus je označován jako mechanismus **krátkého potrubí** (short pipe). Mechanismus **dlouhého potrubí** (long pipe) je vyhrazen pro případy, kdy si zákazník chce stanovovat bit pro nasmlouvanou/nenasmlouvanou kvalitu služeb. V tomto případě je LSP přímo navázána na vstupní hraniční LER směrovač zákazníka.

Pro další využití v této diplomové práci využiji pro dektekci chyb v rámci MPLS domény mechanismu elegantního restartu a dále pro obnovu využiji mechanismu lokální obnovy s přesměrováním. Mechanismus detekce chyb jsem zvolil vzhledem k současným moderním trendům, které v praxi tento mechanismus využívají a zároveň čerpají z jeho nesporných výhod, jejichž dominantním zástupcem je radikální zkrácení času detekce chyby a snížení přenosu redudantních informací na přenosových trasách. Mechanismus obnovy byl zvolen s ohledem na zkoumání kvality služeb v souvislosti s přenosem hlasového hovoru. Lokální obnova s přesměrováním je procesorově nenáročnou implementací, nicméně se

[1] DiffServ Code Point

jedná o mechanismus s největším časem zpracování, který je potřebný pro propočet nových přenosových tras a sestavení záložní cesty LSP. Záměrně jsem tak vybral tento scénář, protože ostatní mechanismy obnovy mají při své funkci lepší časové parametry vzhledem ke kvalitě služeb.

4 PŘÍPRAVA PROSTŘEDÍ OPNET

4.1 Síťový model

Jak jsem již uvedl v úvodu, cílem této diplomové práce je vytvoření modelu MPLS sítě v programovém prostředí OPNET Modeler a simulace chování modelu této sítě se zaměřením na kvalitu služeb. Cílem této práce není seznámení čtenáře se základním ovládáním a funkcemi prostředí OPNET Modeler, které je velice rozsáhlé. V další části této práce se budu výhradně zabývat již jen funkcemi, které přímo souvisí s danou tématikou.

Pro účely simulace bude důležitý vhodný výběr rozšířeného síťového modelu. Protože vliv protokolu MPLS bude více zřejmý u rozsáhlejších datových sítí, vytvořil jsem model s přihlédnutím k těmto kritériím. Abych se vhodným způsobem mohl alespoň částečně dotknout reálného světa kolem nás, zvolil jsem jako prvek vhodný pro modelování mapu České Republiky ve které jsem propojil páteřními spoji vybraná města s hustším osídlením. Do jednotlivých měst síťového modelu jsem umístil dílčí segmenty síťového modelu, do kterých jsem umístil lokální počítačové sítě LAN za účelem simulace rozsáhlejších datových toků a směrování - viz tabulka 4.1.

Pro propojení páteřních směrovačů byly použity přenosové linky E3 s přenosovou rychlostí 34,368 Mbps FULL DUPLEX. Schéma topologie vytvořeného rozšířeného modelu sítě je uveden na obr. 4.1.

4.2 Vytvoření modelu

—TATO ČÁST DIPLOMOVÉ PRÁCE JE NAPSÁNA FORMOU NÁVODU—

1. Spustíme OPNET Modeler.
2. V menu programu zvolíme **File** → **New...**.
3. Z vyobrazené nabídky vybereme **Project** a potvrdíme **OK**.

Tab. 4.1: Seznam měst ČR v rozšířeném modelu sítě

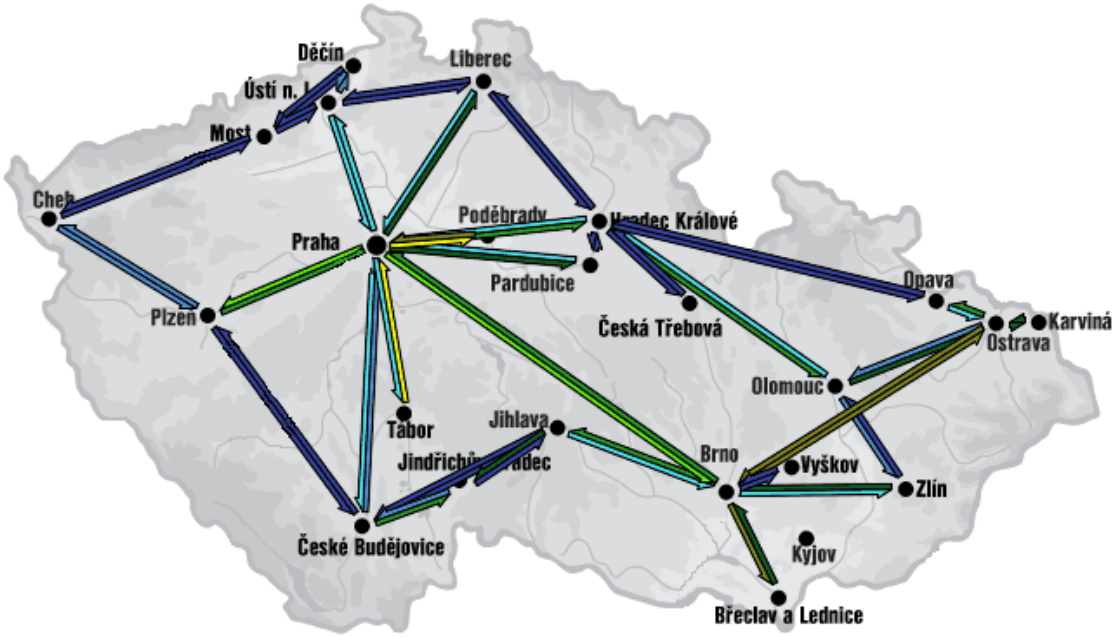
Název města	**Počet subsítí**
Cheb	2
Plzeň	3
Most	2
Ústí nad Labem	3
Děčín	2
Praha	5
České Budějovice	3
Tábor	2
Jindřichův Hradec	2
Poděbrady	3
Liberec	3
Jihlava	3
Pardubice	3
Hradec Králové	3
Česká Třebová	3
Brno	5
Břeclav	3
Vyškov	3
Olomouc	4
Zlín	3
Opava	3
Ostrava	3
Karviná	2

4. Zadáme jméno projektu **např. MPLS_SIM** a jméno prvního scénáře, **např. MPLS_NOQOS**, položku **Use Startup Wizard when creating new scenarios** ponecháme zaškrtnutou jak popisuje obr. 4.2 a potvrdíme **OK**.

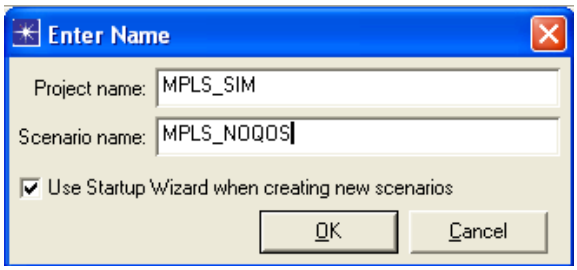
5. V nově otevřeném průvodci vytvořením nového projektu vybereme položku **Create empty scenario** a klikneme na tlačítko **Next**.

6. V další obrazovce budeme vybírat rozsah naší síťové topologie - v tomto případě zvolíme **Choose from maps** se zapnutou volbou **Use metric units**

7. Jako hraniční mapu zvolíme **Border map: europe** a z dostupných map vybereme **MapInfo maps: europe** a klikneme na tlačítko **Next**.



Obr. 4.1: Schéma topologie sítě pro simulaci



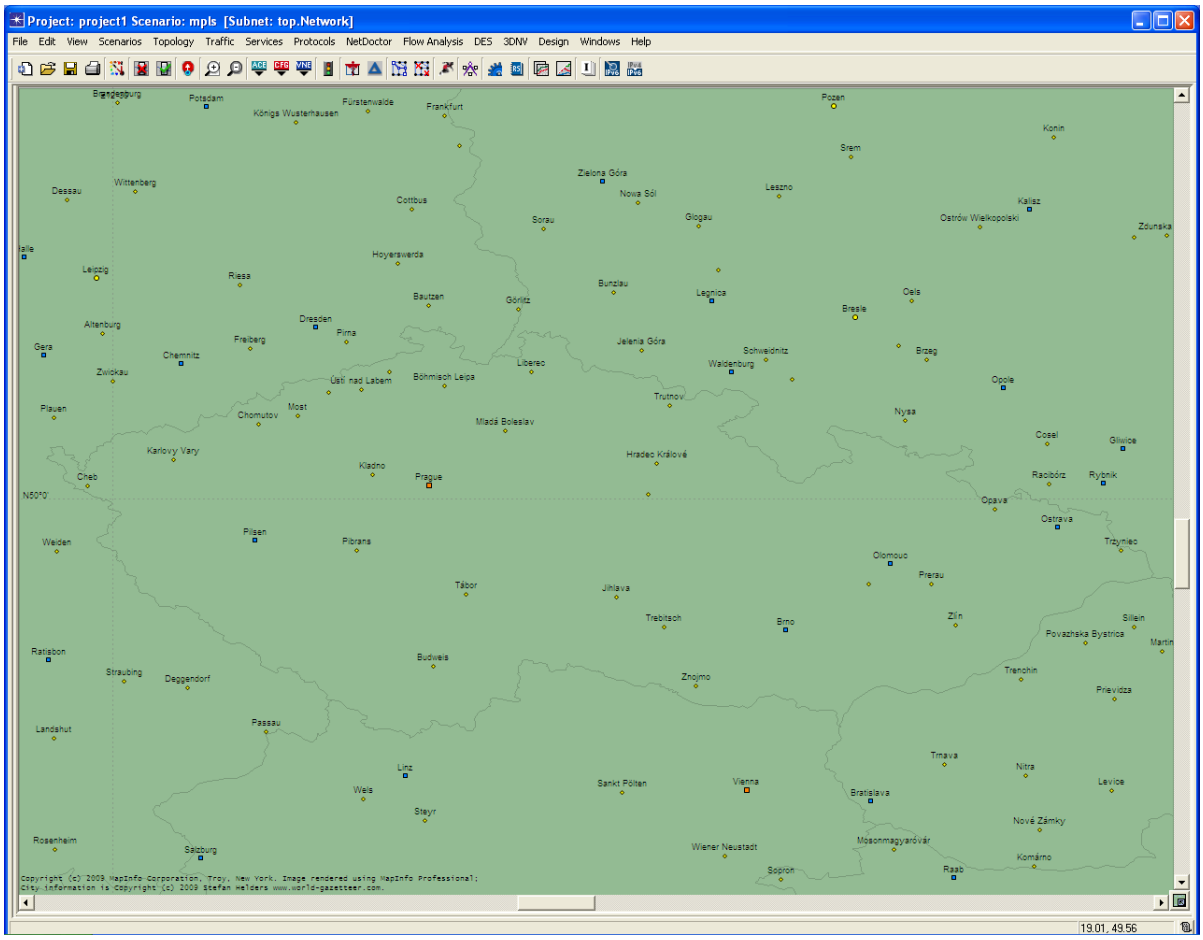
Obr. 4.2: Založení nového projektu v prostředí OPNET

8. Dalším naším úkolem je volba síťových technologií, které budeme v daném modelu využívat. Tyto technologie se nám nahrají do naší standardní palety a práce se síťovými prvky tak bude rychlá. Označíme položky **ethernet** a **MPLS** a klikneme na tlačítko **Next**. Dokončíme průvodce potvrzením tlačítka **Finish**.

9. Po dokončení průvodce je zobrazena mapa prázdného projektu a je otevřena paleta standardních objektů, kterou pro další práci zavřeme.

10. Máme zobrazenu mapu celé Evropy. Pro další práci provedeme ZOOM na lokaci České republiky jak je uvedeno na obr. 4.3. V takovéto podobě máme program připraven pro vytvoření síťové topologie.

11. Otevřeme si paletu objektů a vybereme položku **Subnet**. Vytvoříme první z



Obr. 4.3: Připravená mapa České republiky

podsítí **Praha**. Kliknutím do mapy umístíme subsít' do příslušné lokality a s použitím **Set name** změníme jméno dané subsítě na hodnotu **Praha**.

12. Dvojklikem na subsít' Praha otevřeme její editaci.

13. Z palety objektů do subsítě přidáme následující položky:
    SUBSÍŤ: **Node Models** → **1000BaseX_LAN** a to celkem 5x
    PŘEPÍNAČ: **Node Models** → **ethernet16_switch**
    LER: **Node Models** → **ethernet2_slip8_ler**
    LSR: **Node Models** → **ethernet2_slip8_lsr**

14. Model LER směrovače přejmenujeme na **Praha LER**

15. Model LSR směrovače přejmenujeme na **Praha LSR**

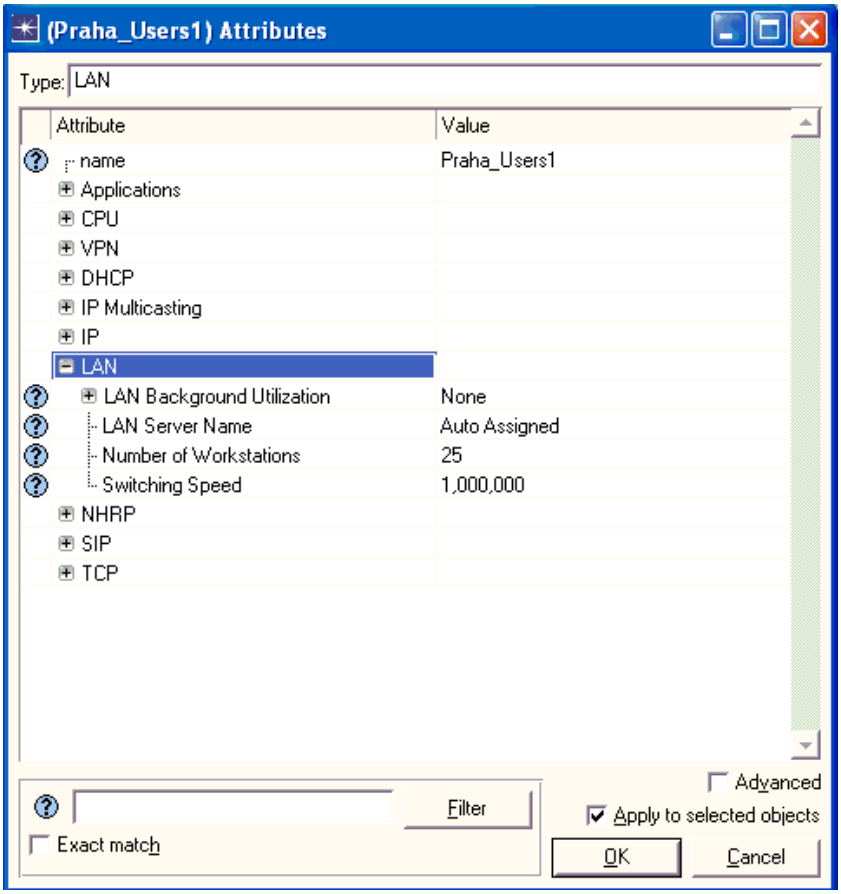
16. Protože model směrovače LSR je opatřen pouze dvěma porty (pro naší simulaci nedostatečné) bude nezbytné, abychom do modelu přidali další porty.

To provedeme s pomocí volby objektu **Add Ports...**. S pomocí intuitivního průvodce přidáme do modelu směrovače dalších **10 portů typu Serial - PPP**. To již bude dostatečná kapacita pro vytvoření našeho síťového modelu.

17. Model přepínače přejmenujeme na **Praha_switch_central**

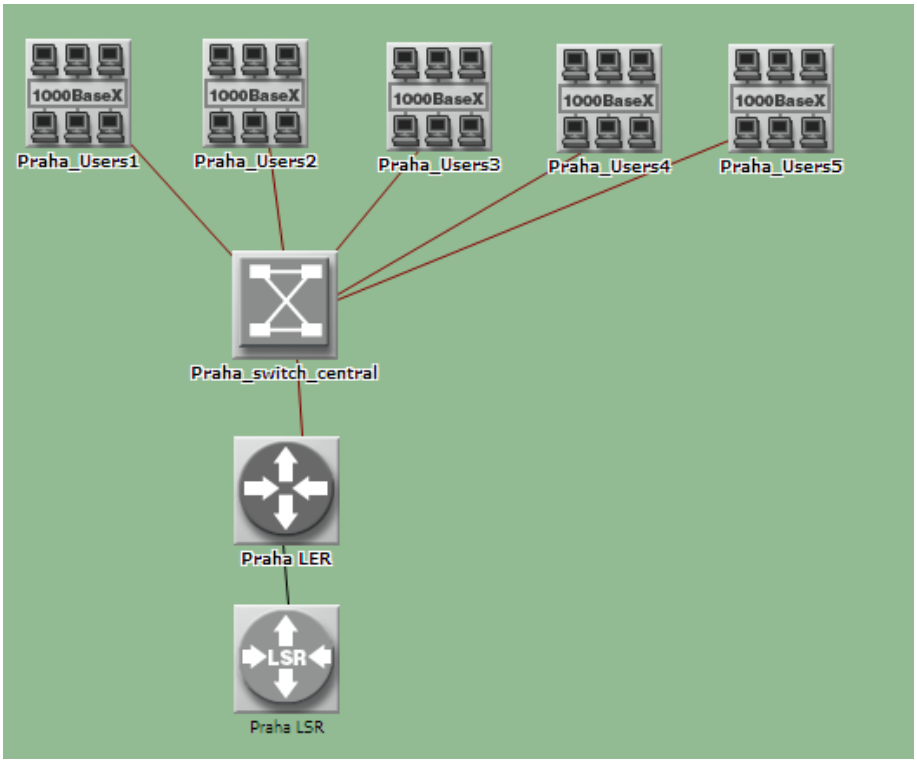
18. Modely všech pěti subsítí s počítači přejmenujeme postupně na **Praha_Users1**, **Praha_Users2**, **Praha_Users3**, **Praha_Users4** a **Praha_Users5**.

19. Všechny modely **Praha_Users*** označíme a pomocí **Edit attributes** nastavíme konfiguraci lokální sítě **LAN** tak, abychom měli v subsíti vždy **25 stanic** s přepínací rychlostí switche **1 mil. paketů/s**. V tomto kroku editujeme parametr více prvků v modelu, proto nezapomeneme zaškrtnout volbu **Apply to selected objects** viz. obr. 4.4 a potvrdíme tlačítkem **OK**.



Obr. 4.4: Konfigurace LAN parametrů modelu 1000BaseX_LAN

20. Z palety objektů vybereme **Link Models** → **10Gbps_ethernet** a s použitím tohoto média propojíme všechny lokální subsítě do přepínače, který následně spojíme s LER směrovačem, který nám vytvoří rozhraní sítě MPLS. Z palety objektů dále vybereme linku typu E3 [1]. Linku E3 přidáme z palety objektů pomocí kategorie **Link Models** → **PPP_E3**, kterou propojíme směrovače LER a LSR v dané subsíti. Výsledek by měl být totožný s obr. 4.5.



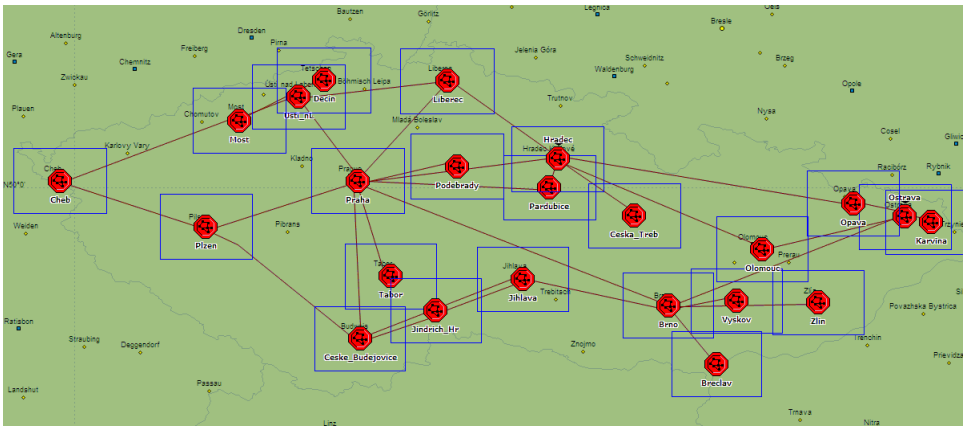
Obr. 4.5: Vyobrazení modelu vzorové subsítě Praha

21. Tímto máme vytvořený model jedné subsítě. Nyní se přesuneme o úroveň výš a subsít' Praha vykopírujeme do všech patřičných lokalit, dle obr. 4.2 a tabulky 4.1. Všechny subsítě řádně pojmenujeme a v lokalitách, kde budeme očekávat nižší frekvenci provozu můžeme v subsíti snížit počet dílčích podsítí s pracovními stanicemi.

22. Jakmile máme v modelu naší sítě vloženy všechny dílčí subsítě, vybereme z palety objektů sériovou linku typu E3 **Link Models** → **PPP_E3**. **Směrovače** typu LSR jednotlivých segmentů sítě vzájemně propojíme. ***!POZOR! Důležité je propojit vzájemně právě LSR SMĚROVAČE.***

[1] linka typu E3 nám poskytne propojovací rychlost 34,368 Mbps FULL DUPLEX, která bude pro simulaci dostačující

23. Dílčí výsledek modelu sítě by měl být totožný s vyobrazením na obr. 4.6.

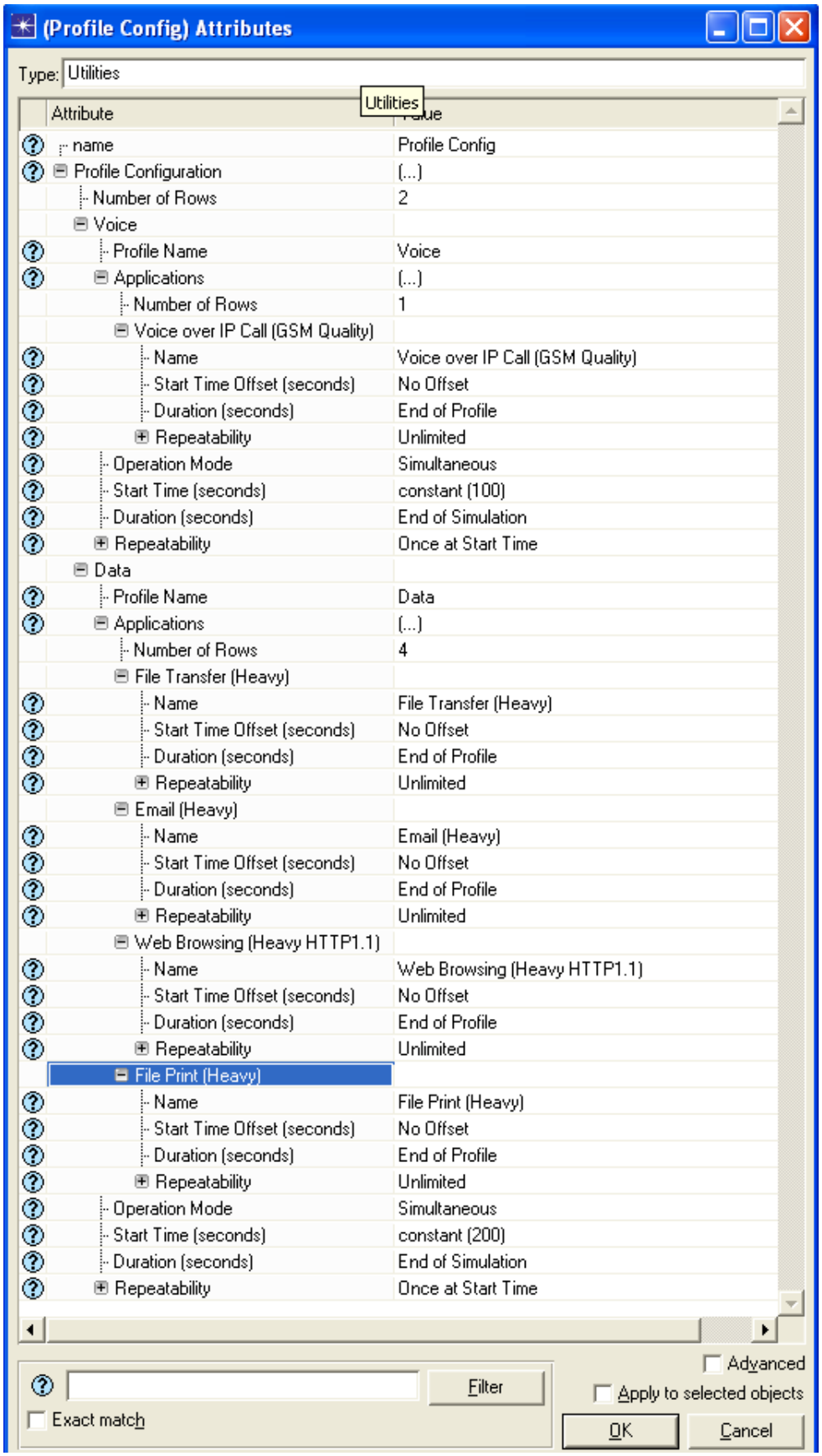


Obr. 4.6: Vyobrazení hotového modelu sítě vč. linek

24. Korpus rozšířeného modelu sítě máme nyní hotový. Pro správnou činnost simulace bude nutné nastavit aplikace, které budeme v rámci sítě provozovat a zkoumat jejich vliv. Z palety objektů vložíme do mapy České republiky položku **Node Models** → **Application config** a řádně pojmenujeme. Pomocí menu **Edit Attributes** nastavíme položku **Application Definitions** na hodnotu **Default** a uložíme. Tímto zpřístupníme naší simulaci standardní předdefinovanou sadu aplikací jako jsou např. FTP, HTTP, VoIP, databázové přenosy a další.

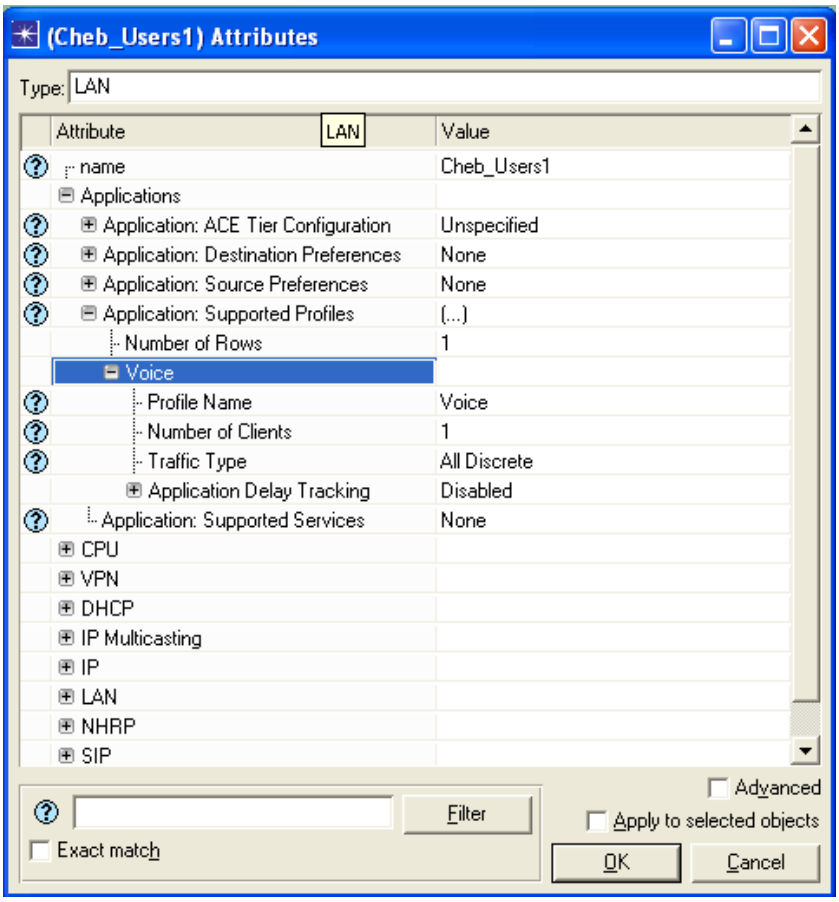
25. Simulace by nefungovala správně, pokud by rozšířený model sítě postrádal definici aplikačních profilů, tj. konkrétní nastavení používaných aplikací vč. definice rozsahu a dalších rozšířených parametrů. Z palety objektů vložíme do mapy České republiky položku **Node Models** → **Profile config** a řádně pojmenujeme. Pomocí menu **Edit Attibutes** nastavíme položku **Profile Configuration** na hodnotu **Edit...**. Do otevřeného kontextového okna vložíme první řádek s názvem profilu **Profile Name** např. **Voice** s aplikací **Applications** **Voice over IP Call** (GSM Quality) [No Offset, s trváním **Duration** do konce profilu **End of Profile** a s nekonečným opakováním **Repeatibility** → **Unlimited**], se souběžným operačním módem **Operation Mode** → **Simultaneous**, s časem startu 100 sekund od počátku simulace **Start Time** → **constant(100)**, s délkou trvání do konce simulace **Duration** → **End of Simulation** a bez opakování

Repeatability → **Once at Start Time**. Další řádek s názvem profilu **Data** vytvoříme obdobným postupem s následujícími parametry: aplikace File Transfer(Heavy), Email(Heavy), Web Brosing(Heavy), File Print(Heavy) [vše No Offset, End of Profile, Unlimited], se souběžným operačním módem se souběžným operačním módem **Operation Mode** → **Simultaneous**, s časem startu 200 sekund od počátku simulace **Start Time** → **constant(200)**, s délkou trvání do konce simulace **Duration** → **End of Simulation** a bez opakování **Repeatability** → **Once at Start Time**. Jakmile máme vytvořeny oba aplikační profily **Voice** a **Data**, uložíme nastavení dvojnásobným stisknutím tlačítka **OK**. Referenční ukázka nastavení aplikačních profilů je vyobrazena na obr. 4.7.



Obr. 4.7: Vyobrazení nastavení aplikačních profilů

26. Pro vytváření simulace přenosů dat po síti MPLS bude zapotřebí vytvořit síťový provoz, který jsme již nadefinovali v aplikačních profilech. Pro odesílání dat budeme využívat server, který bude umožňovat jednotlivým klientům přenášet svá data. Vzhledem k tomu, že budeme budoucí síťový provoz vztahovat k uzlu Praha, otevřeme si tuto subsíť a z palety objektů do této pozice vložíme objekt **Node Models** → **ethernet_server** a pojmenujeme jej např. jako **server**. Nezapomeneme propojit linkou typu Gigabit ethernet s objektem **Praha_switch_central**. Pomocí menu **Edit Attributes** nastavíme položku **Applications** → **Application: Supported Services** na hodnotu **All** a uložíme. Tímto jsme nastavili vlastnost serveru, obsluhovat všechny příchozí požadavky podporovaných aplikací.



Obr. 4.8: Nastavení aplikačního profilu v segmentu Cheb_Users1

27. Aby bylo možné provádět patřičnou simulaci, musíme mimo serveru do našeho modelu sítě dále nastavit i klientské stanice tak, aby se k tomuto serveru připojily a sestavily určitý druh síťového provozu. Za tímto účelem si vybereme v našem rozšířeném modelu sítě subnet s názvem Cheb, který si otevřeme.

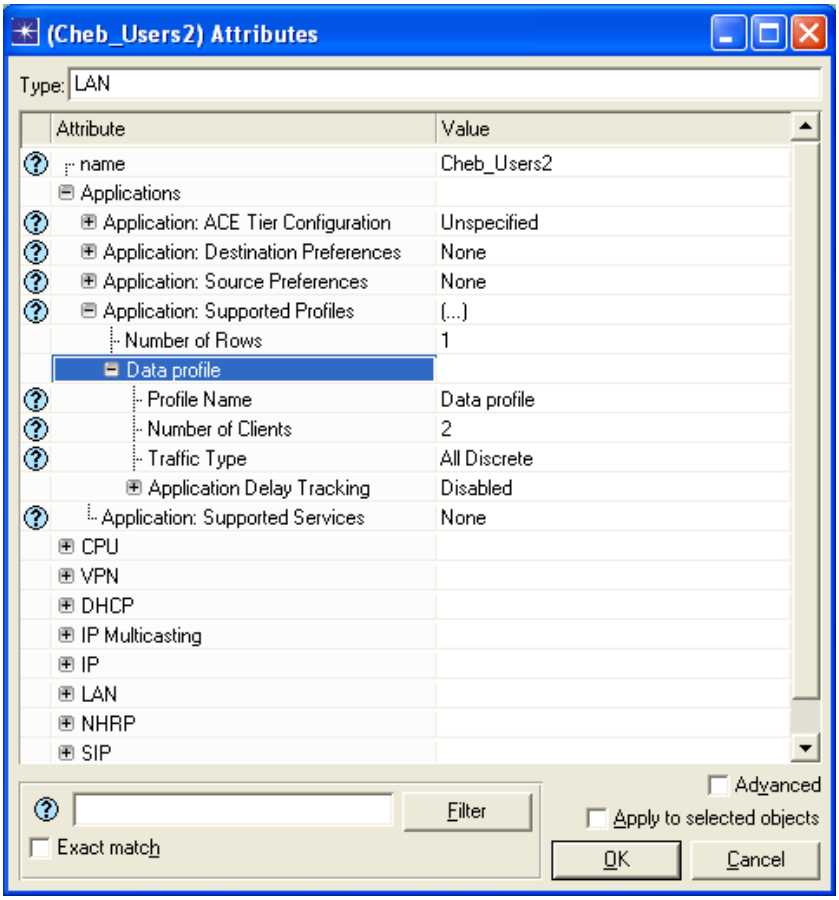
Označíme klientskou sadu **Cheb_Users1** a s pomocí volby **Edit Attributes** nastavíme položku **Applications** → **Application: Supported Profiles** na hodnotu **Edit...**. V otevřené kontextové tabulce vložíme nový řádek, kde ze seznamu v poli **Profile Name** vybereme položku **Voice**[2], položku **Number of Clients** nastavíme na hodnotu **1**, **Traffic Type** **All Discrete** a položku **Application Delay Tracking** nastavíme na hodnotu **Disabled** jak je vyobrazeno na obr. 4.8.

28. Stejným postupem, jako je uvedeno v bodě 27 této sekce, vytvoříme aplikační profil pro klientskou sadu **Cheb_Users2** v subnetu Cheb. Předmětem nastavení bude profil s názvem **Data profile**[3], položku **Number of Clients** nastavíme na hodnotu **2**, **Traffic Type** **All Discrete** a položku **Application Delay Tracking** nastavíme na hodnotu **Disabled** jak je vyobrazeno na obr. 4.9.

Rozšířený model sítě máme dokončením tohoto postupu vytvořen. V této kapitole jsme vytvořili vlastní model sítě, který je pro další simulaci sítě MPLS dostatečně rozsáhlý. Do jednotlivých měst jsme umístili subnety vč. klientských stanic, přepínačů a směrovačů typu MPLS (LER i LSR). Jednotlivé subsítě jsme propojili sériovými linkami Point-To-Point typu E3, jenž nám poskytne maximální datovou propustnost 34,368 Mbps v dopředném i zpětném směru zároveň, což je dostatečná kapacita k provedení simulace a zkoumání protokolu MPLS.

[2]toto nastavení provede asociaci s dříve definovaným aplikačním profilem - klientská stanice tedy sestaví Voice Over IP hovor se serverem

[3]toto nastavení provede asociaci s dříve definovaným datovým profilem - klientská stanice provede přenosy FTP, WWW, Databáze, Tisk



Obr. 4.9: Nastavení aplikačního profilu v segmentu Cheb_Users2

5 SIMULACE

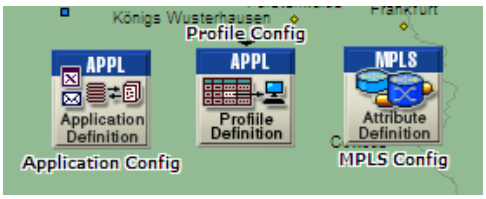
5.1 MPLS bez QoS

V tomto scénáři naváži na rozšířený model sítě, který jsem vytvořil v kapitole 4.2 této diplomové práce. Cílem této kapitoly je sestavení funkčního modelu sítě s použitím protokolu MPLS. Předmětem simulace bude sestavení hlasového hovoru[1] v kvalitě GSM mezi uzlem Cheb_Users1 a serverem umístěného v uzlu Praha bez použití priorizace služeb pomocí mechanismu QoS na úrovni protokolu MPLS. Všechny datové přenosy tak budou typu BEST EFFORT. Nad rámec provozu na pozadí linek je uvažováno sestavení datového spojení[2] mezi uzlem Cheb_Users2 a serverem taktéž umístěným v uzlu Praha. Datový tok tohoto provozu tak bude směřovat stejným směrem a v tomto konkrétním scénáři i přes stejné linky transportní sítě.

5.1.1 Vytvoření scénáře

—TATO ČÁST DIPLOMOVÉ PRÁCE JE NAPSÁNA FORMOU NÁVODU—

1. Otevřeme paletu objektů a z nabídky vybereme v kategorii **MPLS** volbu **Node Models** → **mpls_config_object** a přetáhneme jej do mapy České republiky vedle aplikačních profilů a definice aplikací. Objekt pojmenujeme jako **MPLS Config** jak je ilustrováno na obr. 5.1.

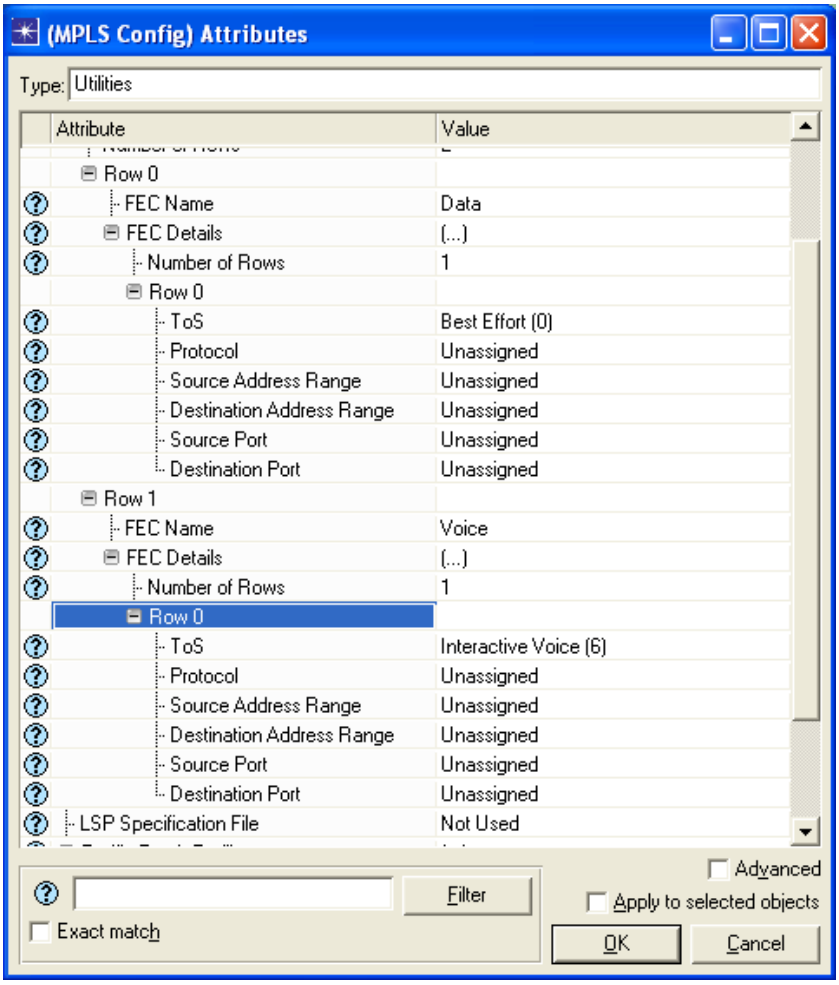


Obr. 5.1: Definice aplikační sady, aplikačních profilů a profilu MPLS

2. Označíme nově vytvořený profil MPLS a s pomocí položky **Edit Attributes** budeme dále konfigurovat nastavení profilu. Budeme editovat položku **FEC Specifications** → **Edit**. V otevřeném kontextovém okně vložíme dva nové řádky - např. **Data** a **Voice**. Pro FEC třídu Data dále nastavíme **ToS**

[1] aplikační profil Voice
[2] aplikační profil Data profile

na hodnotu **Best Effort(0)** a ostatní položky necháme jako **Unassigned**. Pro FEC třídu Voice nastavíme **ToS** na hodnotu **Interactive Voice(6)** a ostatní položky necháme jako **Unassigned** jak je vyobrazeno na obr. 5.2.

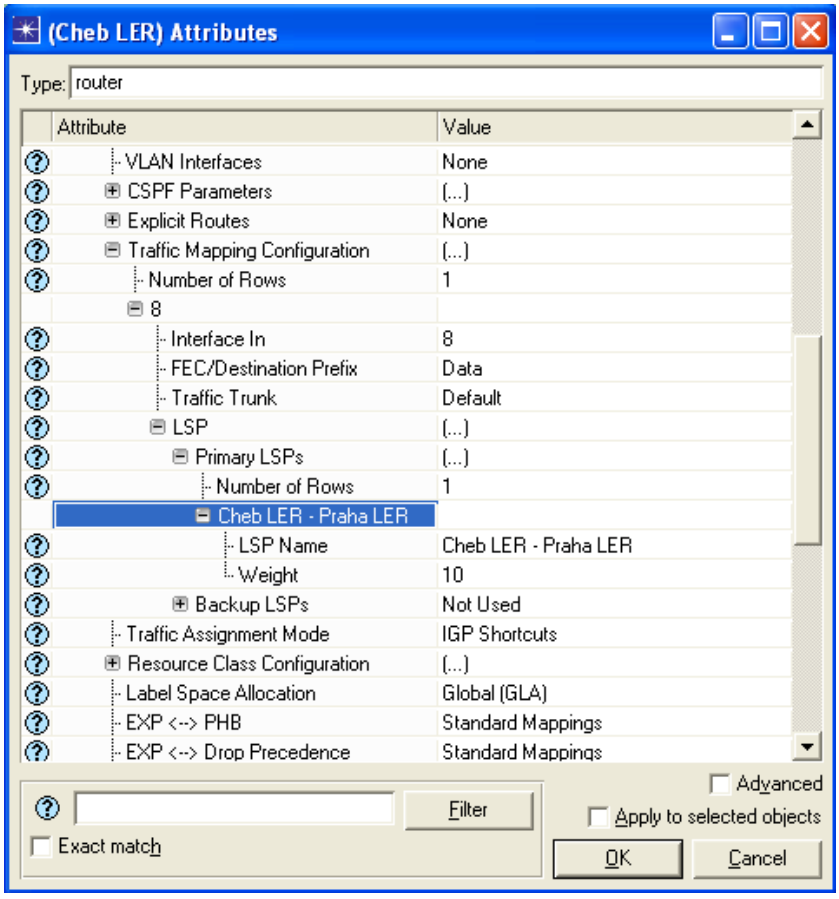


Obr. 5.2: Definice profilu MPLS a nastavení FEC tříd

3. Jakmile máme nadefinovány FEC třídy můžeme přistoupit k samotnému vytvoření dynamických cest MPLS LSP. V paletě objektů vybereme položku **Path models** → **MPLS_E-LSP-DYNAMIC** a touto dynamickou cestou spojíme jako první prvky **Cheb LER** a **Praha LER**. Při vytváření LER musíme být opatrní, nesmíme propojit směrovače LSR, které k tomuto nejsou určeny. Dále je velice důležité na pořadí, tj. povedeme linku směrem Cheb→ Praha. Po vytvoření LSP tuto linku OPNET zobrazí barevně a zakončí šipkou ve směru definované cesty.

4. Otevřeme směrovač LER v příslušném bodě, ze kterého LSP vychází (v prvním případě Cheb LER) a volbou položky **Edit Attributes** zobrazíme kontextovou

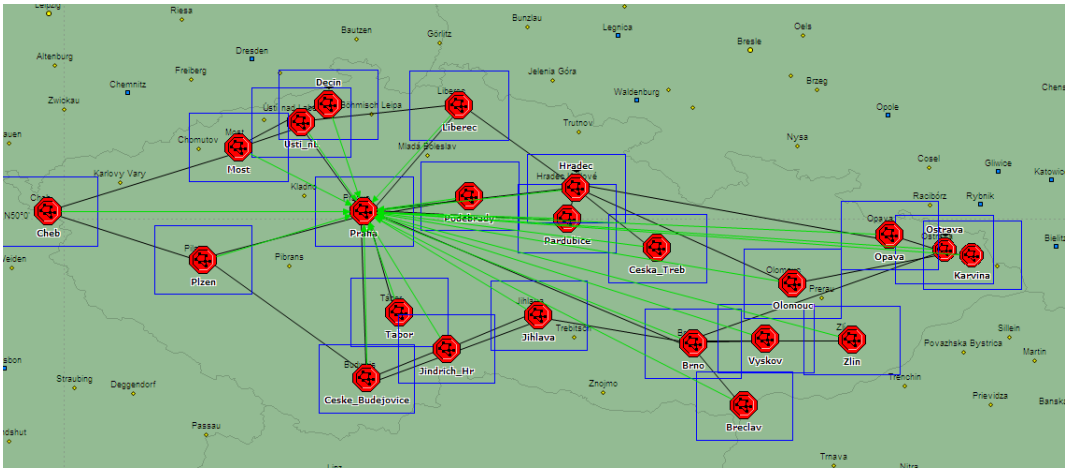
nabídku, ve které postupujeme následovně: **MPLS** → **MPLS Parameters** → **Traffic Mapping Configuration** a vybereme z rolovací nabídky volbu **Edit...**. V následující nabídce bude důležité nastavit **Interface In** tak, že kliknutím rozbalíme nabídku rozhraní směrovače a u příslušného rozhraní nastavíme volbu **Apply Binding** na hodnotu **Yes**[3] . V našem případě se jedná o rozhraní **Cheb.Cheb_switch**. Velice důležitou položkou pro dané mapovaní je **FEC**, kdy jednoduše kliknutím vybereme některý z námi definovaných FEC tříd - v tomto případě **Data**. **Traffic Trunk** ponecháme na hodnotě **Default** a pokliknutím na položku **LSP** a dále **Primary LSPs** → **Edit...** zvolíme LSP náležející tomuto směrovači - v našem případě **Cheb LER - Praha LER**. Váhu **Weight** necháme nastavenu na standardní hodnotu, protože při volbě jedné LSP nemá tato hodnota na naši simulaci žádný vliv. Vzorové vyobrazení LSP mapování je uvedeno na obr. 5.3.



Obr. 5.3: Nastavení mapování LSR na daném LER směrovači

[3]tento mechanismus interpretuje funkci LER tak, že všechny příchozí datové prvky na daném rozhraní interpretuje do dané LSP s danou třídou FEC

5. Stejným způsobem jak je uvedeno v předchozích bodech 3 a 4 této kapitoly, provedeme vytvoření LSP cest a LSP mapování mezi všemi koncovými LER a směrovačem LER Praha. Výsledek by měl odpovídat obr. 5.4.



Obr. 5.4: Mapa rozšířeného modelu MPLS sítě s nakonfigurovanými LSP

6. Abychom provedli zatížení přenosové sítě, budeme se zabývat vytvořením datových toků na pozadí. Datové toky vytvoříme s použitím hlavního menu aplikace OPNET[4] **Traffic** → **Create Traffic Flows** → **IP Unicast...**. V nově otevřeném okně **Create IP Unicast Traffic Flows** zvolíme položku **From all other nodes to:** a vybereme prvek sítě **Praha.Server**. Musíme dále nadefinovat intenzitu provozu a to v mezích **Packets/sec: 100**, **Bits/sec: 12000000**, **Duration (secs): 2880**[5], **Type of service: Best Effort(0), Protocol: IP**. Datové toky vytvoříme kliknutím na tlačítko **Create**. Vytvořené datové na pozadí pak můžeme blíže zkoumat s použitím **Traffic** → **Open Traffic Center**.

7. Abychom byli schopni s pomocí simulace analyzovat chování námi vytvořené sítě, je nezbytné kromě standardních statistik nastavit individuální statistiky, tedy jinými slovy vlastnosti sítě, jejichž hodnoty chceme zaznamenávat v závislosti na čase. V našem případě budeme zkoumat **jitter**[6] . V subnetu Praha označíme prvek sítě Server a zvolíme volbu **Choose Individual DES Statistics**.

[4]budeme simulovat spojení jednotlivých klientských stanic se serverem, proto zvolíme typ Unicast

[5]doba simulace toků na pozadí sítě je 48 minut

[6]absolutní výkyvy v hodnotách odezvy při šíření dat sítí - menší hodnota **jitter** znamená vyšší kvalitu a vyšší stabilitu sítě

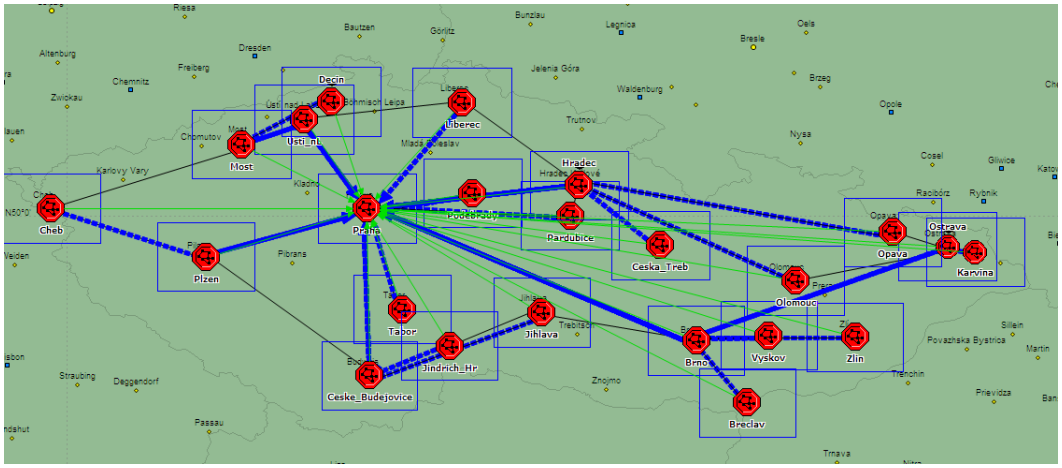
Zaškrtneme volbu **Voice Called Party** (automaticky se nám označí položka Jitter (sec)).
Standardní nastavení OPNETu je při simulaci získávat určitý počet vzorků hodnot v závislosti na čase (sample). Při simulacích se toto dá ovlivnit nastavením zachytávání všech hodnot (all values), což je ale poté náročné na zpracování (přes 40000 hodnot v grafu). Protože Jitter je velice dynamická hodnota, provedeme v aplikaci úpravu způsobu shromažďování dat. Při volbě individuálních DES statistik se u dané položky dá nastavit volba zachytávání hodnot, kdy změníme nastavení z původního SAMPLE na BUCKET, 2s, MAX VALUE. To znamená, že OPNET nebude registrovat všechny hodnoty v daném časovém období, ale jako výstup poskytne maximální simulovanou hodnotu v rámci nastaveného časového intervalu - tedy dvou vteřin. Tímto dochází ke snížení počtu vzorků (v podstatě k podvzorkování), ale v souvislosti s Jitterem a sledováním maximální hodnoty toto není na úkor kvality, protože maximální špičky nás v simulaci právě zajímají a jsou simulací zachyceny.

Abychom nastavili výše zmíněný efekt označíme položku **Jitter** a u této položky změníme způsob shromažďování dat kliknutím na tlačítko **Modify** v sekci **Collection mode**. Provedeme zaškrtnutí volby **advanced** a dále provedeme nastavení **Capture mode: bucket**, **Every 2.0 seconds**, **Bucked mode: max value**, zaškrtneme položku **Reset**. Tímto jsme s nastavením shromažďování statisitk pro naši simulaci hotoví.

8. Nyní máme již vše připraveno pro vlastní simulaci. Tuto simulaci spustíme s pomocí menu **DES** → **Configure/Run Discrete Event Simulation...**. Nastavíme délku simulace **Duration** na hodnotu **80 minutes** a spustíme simulaci s pomocí tlačítka **Run**. Program OPNET provede kompilaci modulů, což může trvat v závislosti na hardwarovém vybavení počítače, a poté následně provede vlastní simulaci, o čemž nás bude informovat s použitím aktualizace stavu včetně ukazatele odhadovaného času do konce simulace.

9. Abychom se přesvědčili, že jsme síťový model sestavili správně, bude naším cílem zobrazit jednotlivé dílčí datové toky mezi klientskými stanicemi a serverem Praha. Zobrazení provedeme skrze menu **Protocols** → **IP** → **Demands** → **Display Route for configured demands**. Zobrazí se nabídka zobrazující všechny zdroje datových toků. Barva vedle názvu zdroje indikuje stav datového toku (zelená indikuje známou síťovou cestu sítí). Po kliknutí na tlačítko **Show All Routes** můžeme dále kliknout na tlačítko **Close** a v

mapě naší sítě se standardně zobrazí síťová cesta jednotlivých datových toků sítí jak je zobrazeno na obr. 5.5.



Obr. 5.5: Zobrazení dílčích datových toků sítí MPLS (traffic flows)

10. Jestliže jsme doposud pracovali správně a datové toky v síti jsou v pořádku, můžeme přejít k samotnému vyhodnocení výsledků simulace během VoIP hovoru mezi uživatelem v uzlu Cheb a serverem v uzlu Praha. Kliknutím pravého tlačítka myši v mapě naší sítě a provedením volby **View results** se dostaneme do prohlížeče výsledků DES simulace. Zde nás zajímají výsledky uzlu Praha.Server, proto postupujeme v prohlížeči tímto směrem **Object Statistics** → **MPLS_SIM** → **Praha** → **Server** → **Voice Called Party**. Zaškrtneme volbu **Jitter(sec)**. U zobrazeného grafu zvolíme **Presentation** typu **average**.

5.1.2 Výsledky simulace

Výsledky této simulace jsou uvedeny v následující kapitole 5.2.2 společně s vyhodnocením a porovnáním simulace **MPLS s impelentací QoS**.

5.2 MPLS s implementací QoS

V tomto scénáři naváži na simulaci MPLS bez QoS, kterou jsem vytvořil v kapitole 5.1 této diplomové práce. Cílem této kapitoly bude sestavení funkčního modelu sítě s použitím protokolu MPLS včetně implementace řízení kvality služeb. Předmětem simulace bude (stejně jako v předcházející kapitole) sestavení hlasového hovoru[7]

[7] aplikační profil Voice

v kvalitě GSM mezi uzlem Cheb_Users1 a serverem umístěného v uzlu Praha s použitím priorizace služeb pomocí mechanismu QoS na úrovni protokolu MPLS. Datové přenosy budu definovat jako službu typu BEST EFFORT a hlasové přenosy (Voice profile) budou mít prioritu INTERACTIVE VOICE. Nad rámec provozu na pozadí linek jsem uvažoval sestavení datového spojení[8] mezi uzlem Cheb_Users2 a serverem taktéž umístěným v uzlu Praha. Datový tok tohoto provozu tak bude směřovat stejným směrem a v tomto konkrétním scénáři i přes stejné linky transportní sítě, stejně jako tomu bylo v předcházejícímu modelu.

5.2.1 Vytvoření scénáře

—TATO ČÁST DIPLOMOVÉ PRÁCE JE NAPSÁNA FORMOU NÁVODU—

1. Nejprve bude zapotřebí zkopírovat scénář vytvořený v kapitole 5.1 této práce. To provedeme s pomocí menu **Scenarios** → **Duplicate Scenario...**. Zadáme jméno nového scénáře na **MPLS_QOS** a potvrdíme. Po provedení této akce se zároveň do daného scénáře přepneme.

2. Síťová topologie a nastavení MPLS je již provedeno z předchozího scénáře, je však nyní nutné provést změnu nastavení tak, aby se při přenosu uplatnilo řízení kvality služeb.

3. V uzlu Cheb přepojíme linku vedoucí mezi **Cheb_users1** a **Cheb_central_switch** tak, aby spojovala **Cheb_users1** přímo se směrovačem **Cheb LER**.

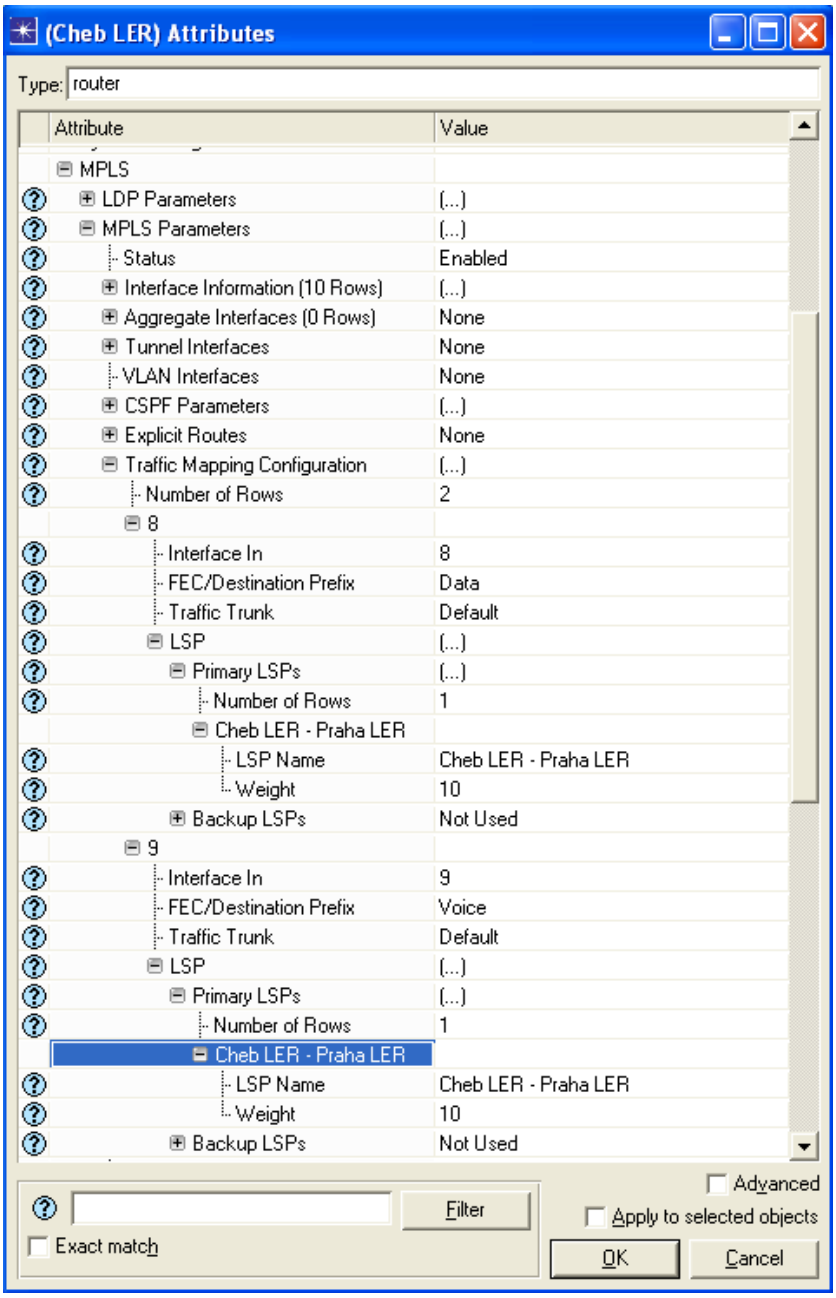
4. Otevřeme směrovač LER v subnetu Cheb a volbou položky **Edit Attributes** zobrazíme kontextovou nabídku, ve které postupujeme následovně: **MPLS** → **MPLS Parameters** → **Traffic Mapping Configuration** a vybereme z rolovací nabídky volbu **Edit...**. V následující nabídce bude důležité nastavit další, nový **Interface In** tak, že kliknutím rozbalíme nabídku rozhraní směrovače a u příslušného rozhraní nastavíme volbu **Apply Binding** na hodnotu **Yes**. V našem případě se jedná o rozhraní **Cheb.Cheb_Users1**. Velice důležitou položkou pro dané mapovaní je **FEC**, kdy jednoduše kliknutím vybereme některou z námi definovaných FEC tříd - v tomto případě **Voice**. **Traffic Trunk** ponecháme na hodnotě **Default** a pokliknutím na položku

[8] aplikační profil Data profile

LSP a dále **Primary LSPs** → **Edit...** zvolíme LSP náležející tomuto směrovači - v našem případě **Cheb LER - Praha LER**. Váhu **Weight** necháme nastavenu na standardní hodnotu, protože při volbě jedné LSP nemá tato hodnota na naši simulaci žádný vliv. Vzorové vyobrazení LSP mapování je uvedeno na obr. 5.6.

5. Statistiky máme nastavené již z předchozí simulace. Spustíme simulaci s pomocí menu **DES** → **Configure/Run Discrete Event Simulation...**. Nastavíme délku simulace **Duration** na hodnotu **80 minutes** a spustíme simulaci s pomocí tlačítka **Run**. Program OPNET provede simulaci.

6. Pro zobrazení výsledků simulace otevřeme kliknutím pravého tlačítka myši a výběrem volby **View results**, kde dále pokračujeme po linii **Object Statistics** → **MPLS_SIM** → **Praha** → **Server** → **Voice Called Party**. Zaškrtneme volbu **Jitter(sec)**. U zobrazeného grafu zvolíme **Presentation** typu **average**. Abychom mohli porovnat výsledek se simulací bez QoS, v sekci **Results for:** zvolíme položku **current project** a dále zaškrtneme název předchozího scénáře tj. **MPLS_NOQOS**. V dolní sekci **Presentation** poté vybereme volbu **Overlaid statistics**.



Obr. 5.6: Konfigurace mapování provozu MPLS pro uplatnění QoS

5.2.2 Výsledky simulace

Pro vyhodnocení našich předchozích dvou simulací bude důležité pro začátek shrnout několik důležitých hodnot, které jsme v simulaci nastavili:

- 100 sekund od začátku simulace - start VoIP hovoru
- 200 sekund od začátku simulace - start toků na pozadí sítě
- 2880 sekund od začátku simulace - konec toků na pozadí sítě

Podíváme-li se blíže na obr. 5.7, je již na první pohled zřejmé, že data přenášená s použitím příznaku TOS INTERACTIVE VOICE (tedy náš hlasový hovor reprezentovaný červeným grafem pro simulaci s implementovaným QoS) vykazují mnohem nižších průměrovaných hodnot **jitter** (maximálně 8 ms) nežli je tomu u hlasového hovoru v režimu BEST EFFORT (reprezentace modrou křivkou pro simulaci bez implementovaného QoS - maximálně 17 ms) a to při zatížení přenosové linky na její plnou možnou kapacitu, jak je uvedeno v příloze této práce na obr. A.4. Vzájemné porovnání parametru **jitter** a jeho průběhu v závislosti na zatížení linky je vyobrazeno na obr. 5.8 (měřítko grafu zatížení linky bylo upraveno vzhledem k parametru **jitter**).

Nástup hodnoty **jitter** z nuly směrem do vyšších hodnot nastává ve výsledcích simulace cca 164 sekund od začátku simulace. Počátek hlasového hovoru začíná 100 sekund v nastavení aplikačního profilu simulace - je tedy zřejmé, že dochází k jistému zpoždění, které je třeba na sestavení hovoru a nežli dojde k saturaci jednotlivých přenosových prvků v síti. Tento jev je nejlépe viditelný na obr. 5.8.

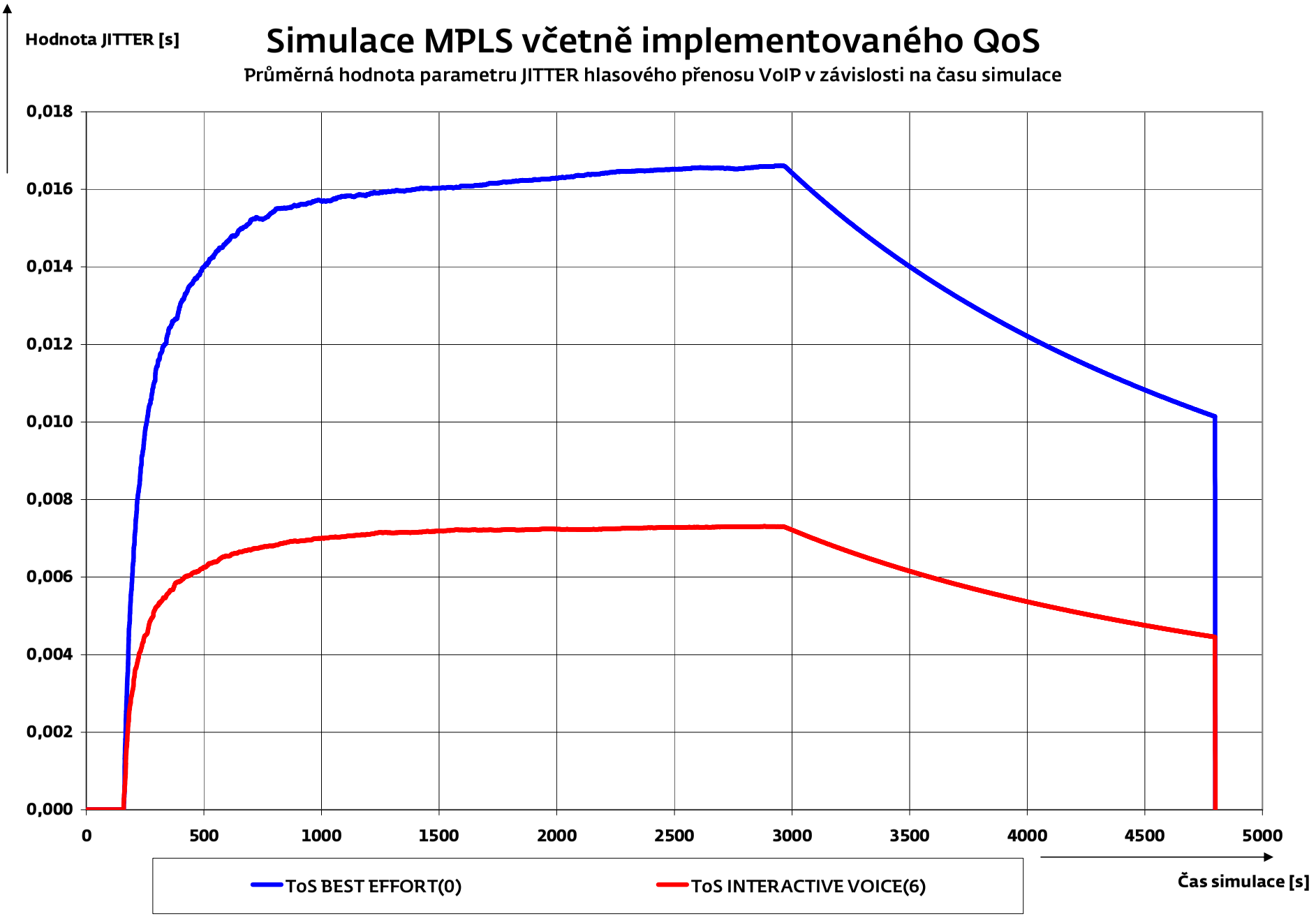
Jak je vidět z obr. 5.7 je v případě obou simulací nástup parametru **jitter** velice strmý, s dobou simulace poté dochází k ustálení na konečné hodnotě (prvky sítě jsou již plně saturovány). Náhlý pád hodnoty **jitter** u obou křivek v čase kolem 3000 sekund od počátku simulace nastává z toho důvodu, že toky na pozadí sítě byly námi nastaveny na dobu 2880 s. Dochází k uvolnění front a přenosových linek, ale samozřejmě toto chování nenastává okamžitě, ale opět s určitým zpožděním (prakticky v simulaci dochází k poklesu v čase 2987 sekund od počátku simulace). Protože došlo k přerušení datových toků na pozadí sítě, dochází k uvolnění jednotlivých cest a průměrná hodnota **jitter** se exponenciálně ustaluje směrem k nulové hodnotě. V čase 4800 s dochází k ukončení simulace (vertikální pád dolů). Exponenciální průběh parametru **jitter** je způsoben exponenciálním průměrováním, které je obsaženo přímo v definici **jitter**u dle [13].

V příloze této diplomové práce je na obr. A.1 vyobrazena datová propustnost point-to-point spoje Plzeň → Praha v závislosti na času simulace. Daný průběh dosahuje při zatížení na 100% přenosové kapacity rychlost 34,368 Mbps, což plně odpovídá maximální přenosové rychlosti linky typu E3. Všechny linky vyobrazené na obr. 5.5 modrými čarami[9] budou vykazovat shodné charakteristiky, tj. zatížení linky dle obr. A.3 a propustnost linky dle obr. A.1, které jsou vyobrazeny v příloze této práce, kde zároveň najdeme i průběhy datové propustnosti referenční přenosové trasy Plzeň → České Budějovice v závislosti na čase simulace jako obr. A.2. Tato linka reprezentuje záložní linky dané topologie, které bychom v obr. 5.5 našli zobrazeny jednoduchými černými čarami[10].

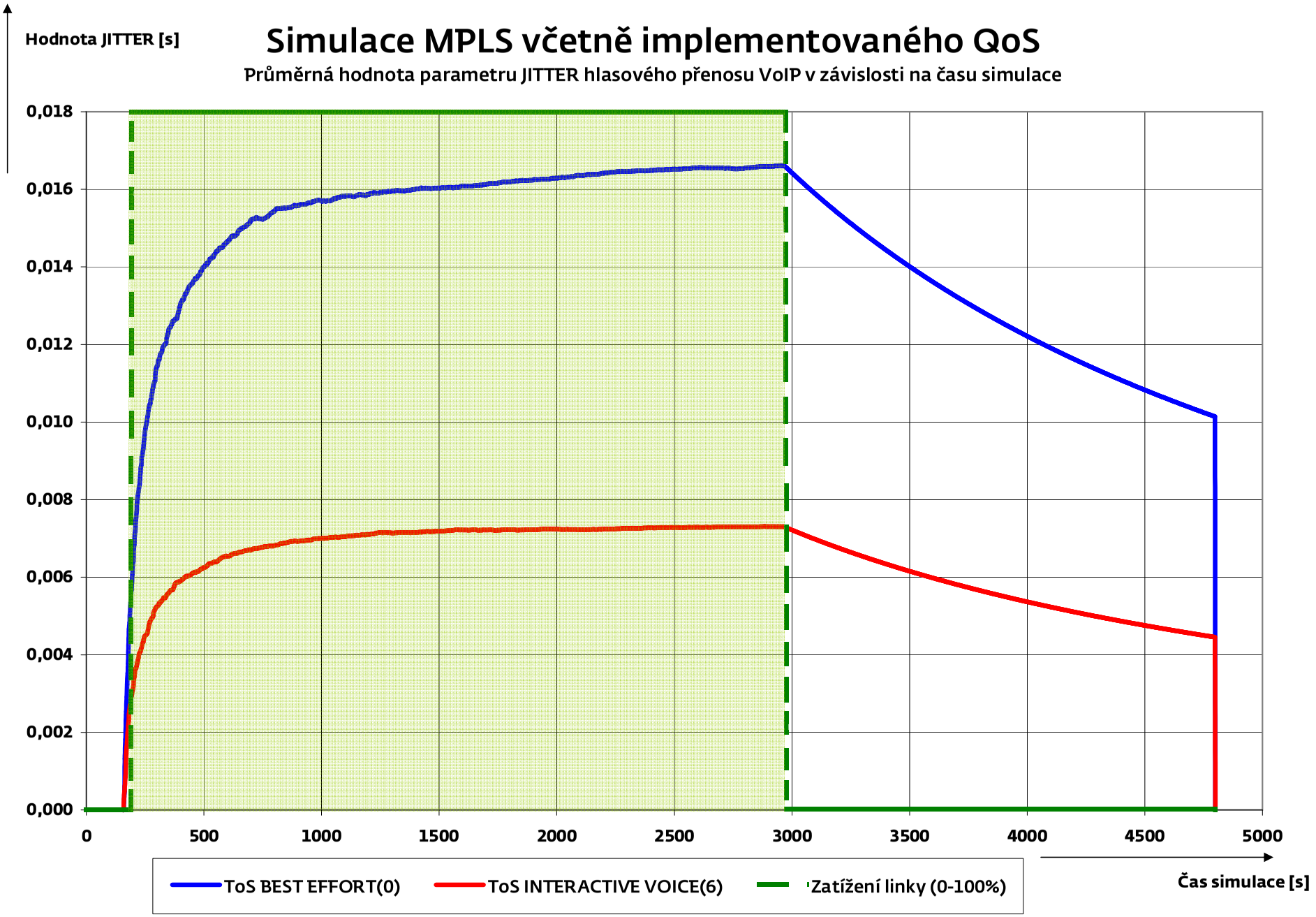
Pro vyhodnocení dalších výsledků simulace je nutno objasnit nový pojem PDV (Packet Delay Variation). Jak uvádí zdroje [13] a [14], PDV je rozdíl mezi end-to-end zpožděním mezi dvěma následujícími pakety v rámci jednoho datového toku. End-to-end zpoždění si můžeme dle [14] vyložit jako zpoždění, které získáme jako rozdíl mezi časem odeslání datového paketu vysílačem na straně jedné a časem přijetí datového paketu přijímačem na straně druhé. Takovéto zpoždění má několik dílčích složek, kterými jsou zpoždění vysíláním, zpoždění zpracováním a zpoždění šířením médiem. Specifický výsledek end-to-end zpoždění hlasového hovoru Voice Over IP naší simulace je vyobrazen v příloze na obr. A.5, PDV je pak na obr. A.4. Na obou zmíněných obrázcích je vidět, že hodnoty zpoždění jsou pro hlasový hovor přenášený přes MPLS síť s implementovanou technikou řízení kvality služeb mnohem nižší. Zpoždění v řádu sekund prakticky odpovídá situaci, kdy pakety procházejí skrz síť z uzlu Cheb do uzlu Praha přes dva plně saturované point-to-point spoje - v případě výpadku páteřního spoje a jeho obnovení je při plné saturaci zpoždění ještě vyšší. Takovéto chování bych z hlediska svých praktických zkušeností u reálné sítě očekával.

Z výše uvedeného vyhodnocení a z obr. 5.7 a obr. 5.8 je zřetelné, že změna parametru TOS má přímý vliv na kvalitu služeb v rámci MPLS sítě. Pakety s vyšší prioritou jsou sítí odbavovány v porovnání s běžným provozem přednostně a proto dochází k celkovému snížení zpoždění těchto paketů při průchodu sítí.

[9] přes dané linky procházejí aktivní toky - linka je protokolem aktivně využívána k přenosu dat
[10] přes dané linky neprocházejí žádné toky



Obr. 5.7: Průměrná hodnota parametru **jitter** hlasového přenosu VoIP v závislosti na času simulace



Obr. 5.8: Průměrná hodnota parametru **jitter** hlasového přenosu VoIP v závislosti na času simulace s vyobrazeným průběhem zatížení linky

5.3 MPLS a jeho chování při výpadku linky

V tomto scénáři naváži na simulaci MPLS s QoS, kterou jsem vytvořil v kapitole 5.2 této diplomové práce. Cílem této kapitoly bude zkoumání vlivu výpadku páteřní linky na chování MPLS sítě - zejména se zaměřením na kvalitu a parametry probíhajícího hlasového hovoru sestaveného mezi uzly Cheb a Praha. Předmětem simulace bude porucha páteřní linky Plzeň-Praha v čase 1000 s od počátku simulace a její opětovné obnovené v čase 2000 s od počátku simulace. Při simulaci využijeme v MPLS síti pro detekci chyb mechanismus elegantního restartu (tj. GRACEFUL RESTART) a jako mechanismus obnovy lokální obnovu s přesměrováním (tj. FAST REROUTE).

5.3.1 Vytvoření scénáře

—TATO ČÁST DIPLOMOVÉ PRÁCE JE NAPSÁNA FORMOU NÁVODU—

1. Nejprve bude zapotřebí zkopírovat scénář vytvořený v kapitole 5.2 této práce. To provedeme s pomocí menu **Scenarios** → **Duplicate Scenario...**. Zadáme jméno nového scénáře na **MPLS_QOS_FAILOVER** a potvrdíme. Po provedení této akce se zároveň do daného scénáře přepneme.

2. Dalším naším úkolem bude nadefinovat v naší simulaci výpadek linky a její znovuobnovení. K tomuto účelu půjdeme do palety objektů a vyhledáme objekt s názvem **Failure Recovery**, který umístíme vedle naší hlavní topologie sítě.

3. Označíme si právě vložený objekt Recovery config typu Failure Recovery a zvolíme možnost **Edit Attributes**.

4. Změníme položku **Failure/Recovery Modeling** na hodnotu **Enabled**, čímž simulačnímu prostředí dáme na jevo, že následující konfigurace bude aktivní, a že při simulaci má program kalkulovat s případnými výpadky či obnoveními linek - dle konfigurace.

5. U atributu **Failure/Recovery Specification** klikneme na položku **Edit...** a do nově vzniklé tabulky vložíme dva nové řádky s následujícími záznamy:
   $MPLS_SIM.Praha <-> Plzen$, Time=1000 s, Status=fail
   $MPLS_SIM.Praha <-> Plzen$, Time=2000 s, Status=recovery
   Těmito konfiguračními direktivami dáváme simulačnímu prostředí signál k vytvoření výpadku linky Praha-Plzeň v čase 1000 s od počátku simulace a k jejímu opětovnému obnovení v čase 2000 s.

6. U atributu **Node Failure Mode** musíme zajistit, aby byl tento atribut nastaven na hodnotu **Node Only** a tím máme konfiguraci výpadku páteřního spoje v rámci rozšířeného modelu sítě hotovou.

7. Dalším krokem před spuštěním simulace je definice parametrů jednotlivých LER směrovačů v souvislosti s MPLS a LDP. Označíme proto všechny LER směrovače a v menu vybereme **Edit attributes**. Nastavíme atribut **MPLS** → **LDP Parametres** → **Graceful Restart** na následující hodnoty: Status-enabled, Reconnect Timeout-120 s, Forwarding State Holdtime-180 s.

8. Před aplikací změny parametrů se ujistíme, že je zaškrtnuta volba **Apply to selected objects**, abychom zajistili, že provedené změny budou aktivní na všech LER směrovačích v rámci našeho modelu sítě.

9. Statistiky máme nastavené již z předchozí simulace. Spustíme simulaci s pomocí menu **DES** → **Configure/Run Discrete Event Simulation...**. Nastavíme délku simulace **Duration** na hodnotu **80 minutes** a spustíme simulaci s pomocí tlačítka **Run**. Program OPNET provede simulaci.

10. Pro zobrazení výsledků simulace otevřeme kliknutím pravého tlačítka myši a výběrem volby **View results**, kde dále pokračujeme po linii **Object Statistics** → **MPLS_SIM** → **Praha** → **Server** → **Voice Called Party**. Zaškrtneme volbu **Jitter(sec)**. U zobrazeného grafu zvolíme **Presentation** typu **average**. Abychom mohli porovnat výsledek se simulací bez QoS, v sekci **Results for:** zvolíme položku **current project** a dále zaškrtneme název předchozích dvou scénářů tj. **MPLS_NOQOS** a **MPLS_QOS**. V dolní sekci **Presentation** poté vybereme volbu **Overlaid statistics**.

5.3.2 Výsledky simulace

Pro vyhodnocení této simulace bude vhodné pro začátek shrnout několik důležitých hodnot, které jsme v simulaci nastavili:

- 100 sekund od začátku simulace - start VoIP hovoru
- 200 sekund od začátku simulace - start toků na pozadí sítě
- 1000 sekund od začátku simulace - výpadek linky Plzeň-Praha
- 2000 sekund od začátku simulace - obnovení linky Plzeň-Praha

- 2880 sekund od začátku simulace - konec toků na pozadí sítě

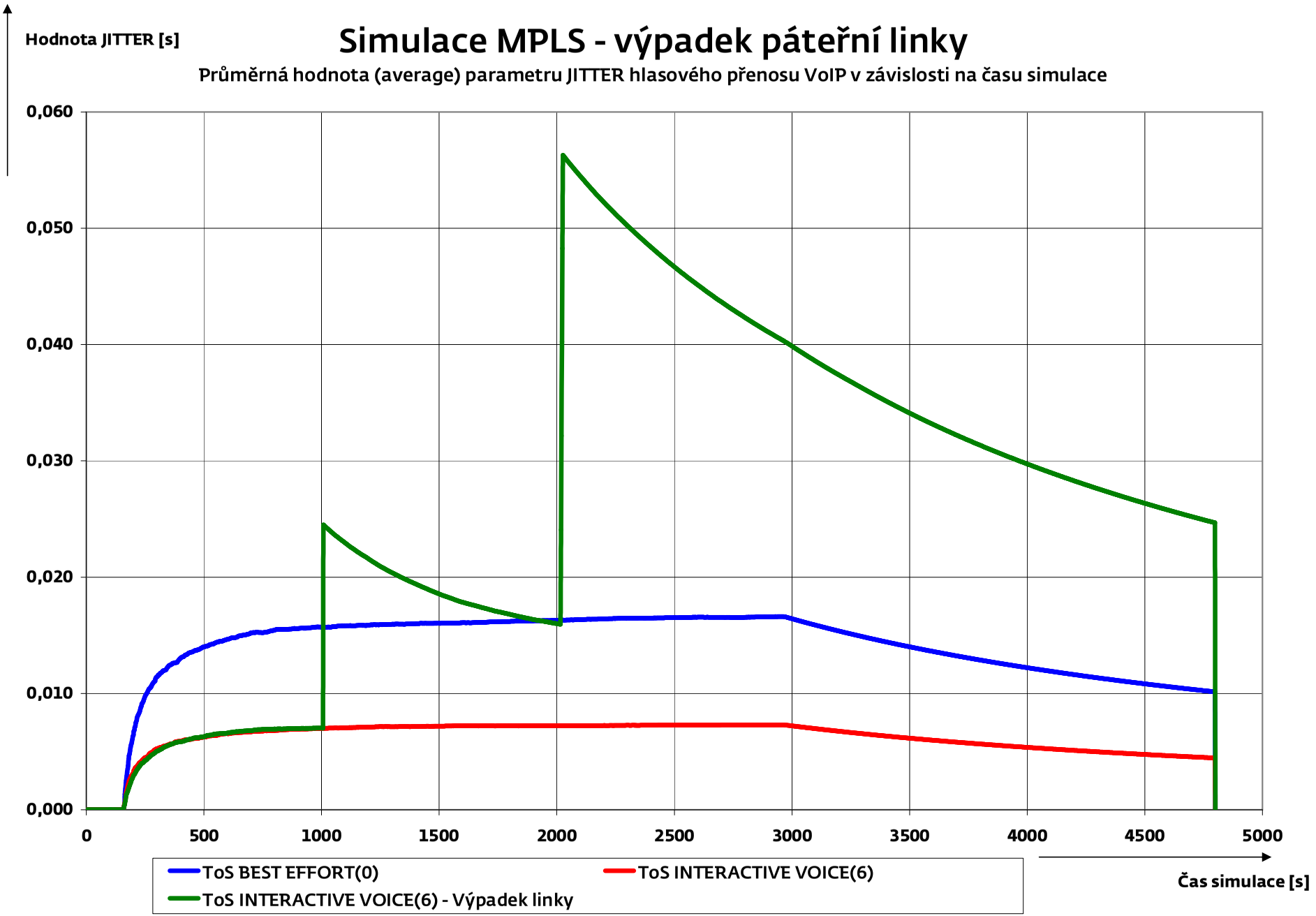
Výsledek této simulace ve srovnání s výsledky předchozích simulací je vyobrazen na obr. 5.9 a obr. 5.10 (zelený graf). Tento graf reprezentuje průměrnou hodnotu parametru **jitter** v závislosti na čase a reprezentuje simulaci, kdy dojde k výpadku páteřní linky Plzeň-Praha.

Z obr. 5.9 vyplývá, že při fyzickém výpadku páteřní trasy dojde ke strmému růstu hodnoty **jitter**, která reprezentuje dobu potřebnou pro přesměrování přenosových tras (14 ms). Po ustálení situace tento parametr exponenciálně klesá až do doby opětovného obnovení přenosové trasy. Doba trvání obnovy a rekalkulace směrování (40 ms) způsobí další strmý nárůst **jitter**u. Po znovuobnovení páteřní linky **jitter** postupně klesá, až do ukončení samotné simulace. Na obr. 5.10 je demonstrována stejná charakteristika s překryvem informace o samotném výpadku páteřní přenosové trasy. Ze simulace je zřejmé, že strmý nárůst hodnoty parametru **jitter** je přímo způsoben samotným výpadkem linky a jejím obnovením, kdy dochází v MPLS doméně k přepočítání přenosových tras.

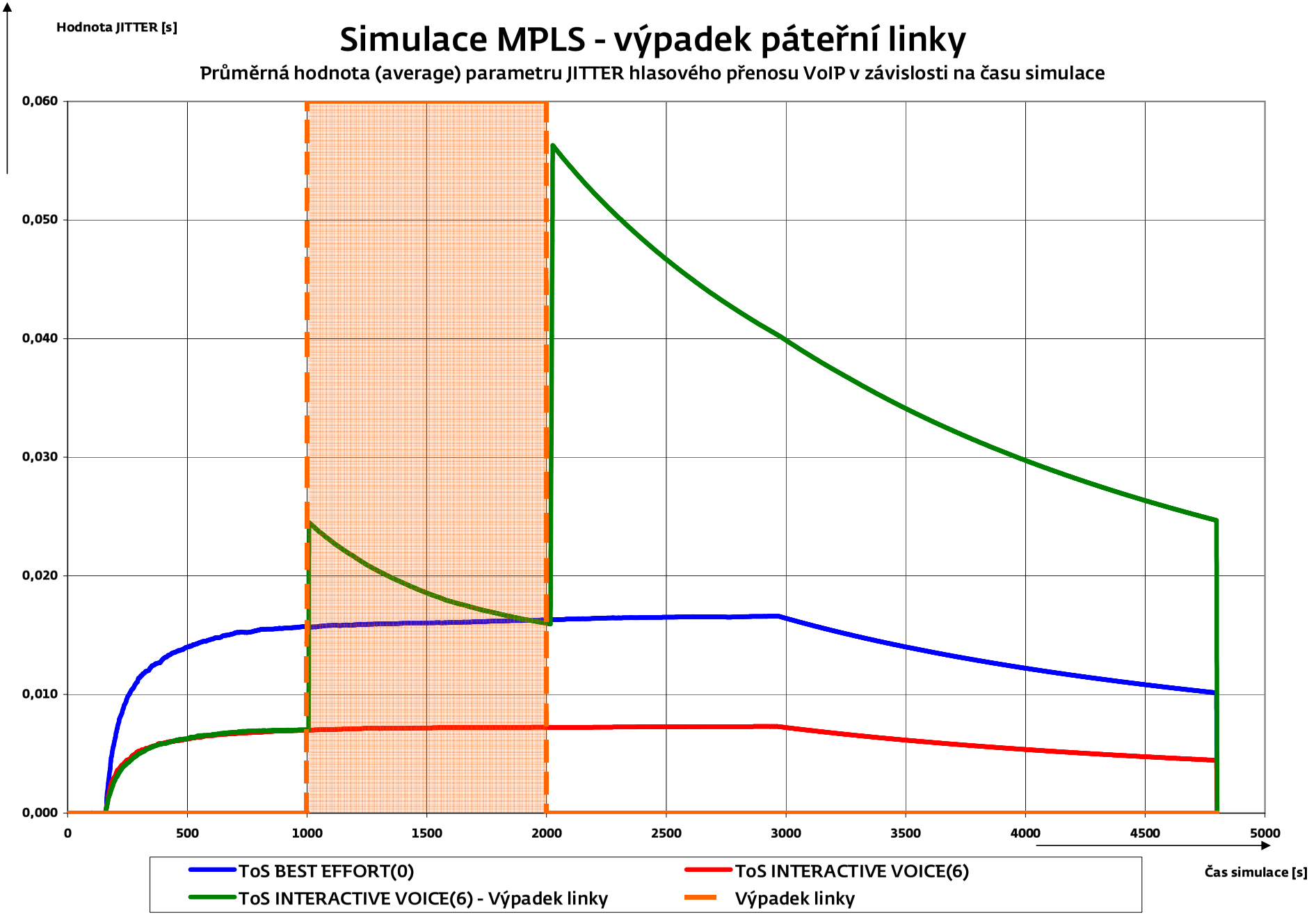
Obrázek 5.11 reprezentuje vliv výpadku linky v MPLS síti na hodnotu parametru **jitter** a na změnu celkového počtu skoků (tzv. hopů), které musí pakety při průchodu sítí překonat. Z obrázku vyplývá, že v případě výpadku linky se reálná hodnota **jitter**u přiblíží až hodnotě 8 sekund, což je hodnota reálná vzhledem k plnému zatížení přenosových tras a saturaci sítě. Toto zatížení zkoumané linky Plzeň-Praha je reprezentováno v obrázku zeleným podbarvením. Zde je důležité si uvědomit, že toto je přirozená hodnota parametru **jitter** - na rozdíl od hodnot průměrovaných funkcí **AVERAGE**, které jsou použity v obr. 5.9 a obr. 5.10. Přímé porovnávání hodnot tak nedává smysl, neboť se jedná o rozdílné metriky. Na obr. 5.10 je dále zřejmé, že se v okamžiku výpadku páteřní linky a přesměrování provozu na náhradní trasy zvýší pro náš hlasový hovor celkový počet HOPů - tedy síťových přeskoků, které musí paket v síti překonat - z původní hodnoty **6** na hodnotu **7** a naopak.

Z výše uvedeného vyhodnocení a z obr. 5.9 a obr. 5.10 je zřetelné, že výpadek páteřní linky sítě MPLS má přímý vliv na hodnotu parametru jitter v rámci simulované sítě, což má návaznost na dostupnou kvalitu přenosu dat a hlasového hovoru. Obr. 5.11 demonstruje vliv výpadku sítě na změnu parametru jitter a počet skoků, které musí dané hlasové pakety urazit. Ze simulace je zřejmé, že takovéto chování v reálné telekomunikační síti by bylo pro hlasové pakety destrukční a že by došlo

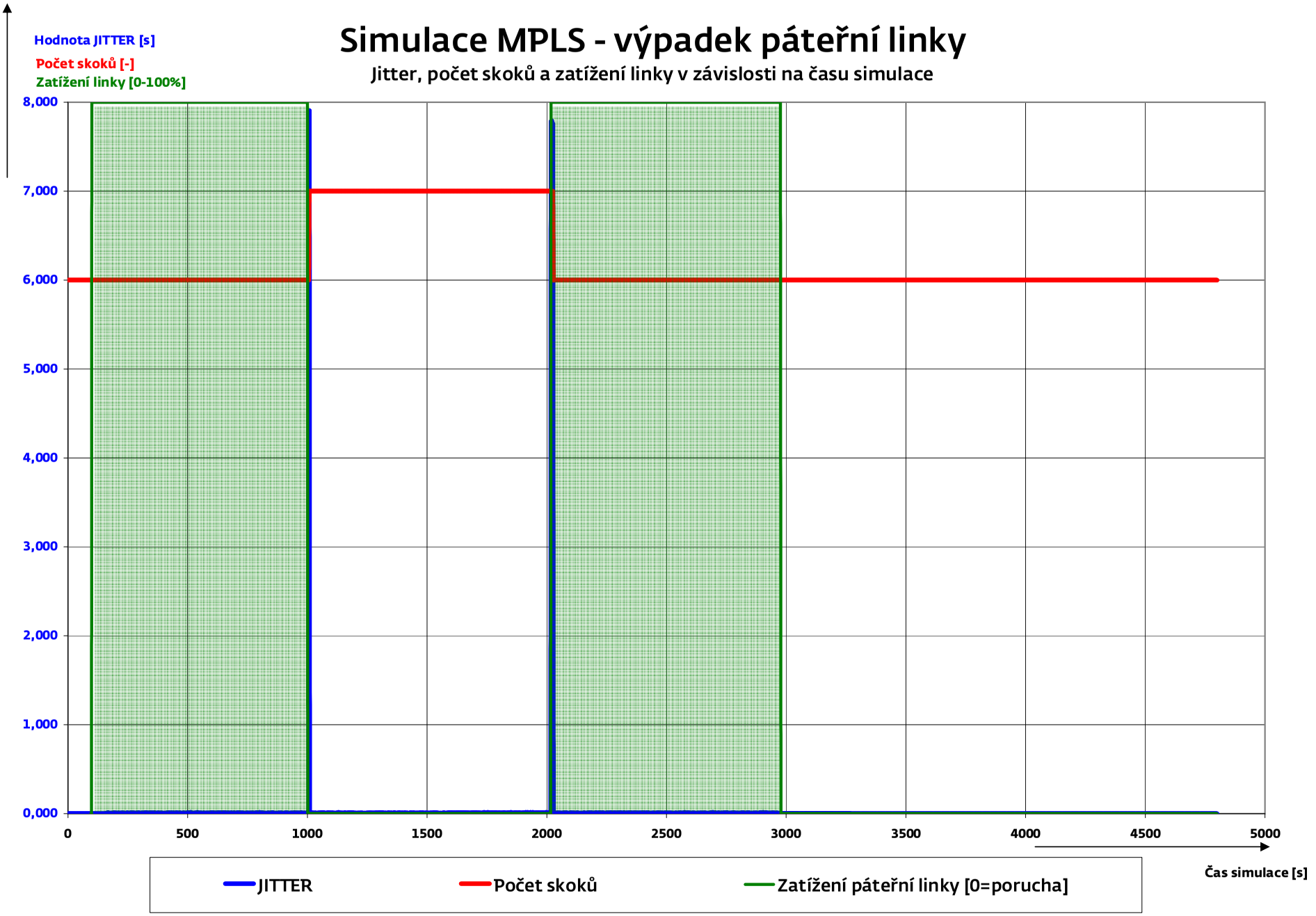
k výpadku hlasové informace přenášené sítí. Vzhledem k plné saturaci všech linek a vzhledem k celkovému zatížení sítě je však toto chování předvídatelné.



Obr. 5.9: Průměrná hodnota parametru **jitter** hlasového přenosu VoIP - výpadek linky



Obr. 5.10: Průměrná hodnota parametru **jitter** hlasového přenosu VoIP - výpadek linky - s vyobrazeným průběhem výpadku páteřní linky



Obr. 5.11: Simulace výpadku páteřní linky - vliv na **jitter** a počet skoků (HOP) při průchodu paketu sítí

5.4 MPLS s TE tunelem

V tomto scénáři použiji pro další rozšíření scénář modelu sítě MPLS s QoS, který jsem vytvořil v kapitole 5.2 této diplomové práce. Cílem této kapitoly bude sestavení funkčního TE tunelu v rámci MPLS sítě. Úkolem TE tunelu bude propojení jednotlivých dílčích firemních poboček označených **FIRMA_POBOCKA_1** (umístění v uzlu Cheb) a **FIRMA_POBOCKA_2** (umístění v uzlu Zlín). V simulaci se zaměřením na důkaz funkce TE tunelu, na směrování paketů v tomto případě a na hodnotu zpoždění přenosu paketů mezi jednotlivými firemními pobočkami vzhledem k reálné použitelnosti.

5.4.1 Vytvoření scénáře

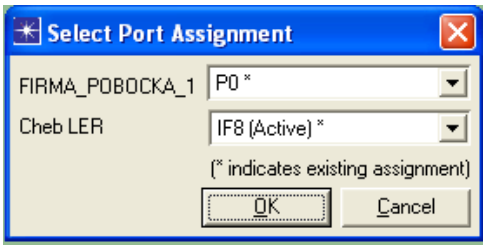
—TATO ČÁST DIPLOMOVÉ PRÁCE JE NAPSÁNA FORMOU NÁVODU—

1. Nejprve bude zapotřebí zkopírovat scénář vytvořený v kapitole 5.2 této práce. To provedeme s pomocí menu **Scenarios** → **Duplicate Scenario...**. Zadáme jméno nového scénáře na **MPLS_TE_TUNNEL** a potvrdíme. Po provedení této akce se zároveň do daného scénáře přepneme.

2. Pro potřeby našeho scénáře bude zapotřebí tento scénář patřičně modifikovat. V uzlu Zlín nejprve přejmenujeme uživatelskou podsíť **Zlin_Users_1** na **FIRMA_POBOCKA_2** a s použitím předchozích znalostí propojíme tento prvek přímo s hraničním směrovačem LER **Zlin LER**.

3. Stejným způsobem jako v předchozím kroku přejmenujeme podsíť **Cheb_Users_1** na **FIRMA_POBOCKA_1**. Opět zajistíme přímé propojení k hraničnímu směrovači **Cheb LER**.

4. Před vytvořením samotného TE tunelu, bude zapotřebí, abychom u jednotlivých LER směrovačů identifikovali rozhraní, která budou sloužit jako 'zákaznická' rozhraní v rámci TE tunelu. Všechny pakety přicházející na tomto rozhraní budou LER směrovačem automaticky odeslána v rámci TE tunelu a opačně. V jednotlivých podsítích zjistíme použité rozhraní tak, že označíme linku spojující dílčí firemní pobočku **FIRMA_POBOCKA** s hraničním směrovačem LER. Kliknutím pravého tlačítka myši a zvolením nabídky **Edit ports** získáme informace o jednotlivém přirazení portů na obou síťových prvcích viz. obrázek 5.12. V případě této modelové simulace bylo přiřazení



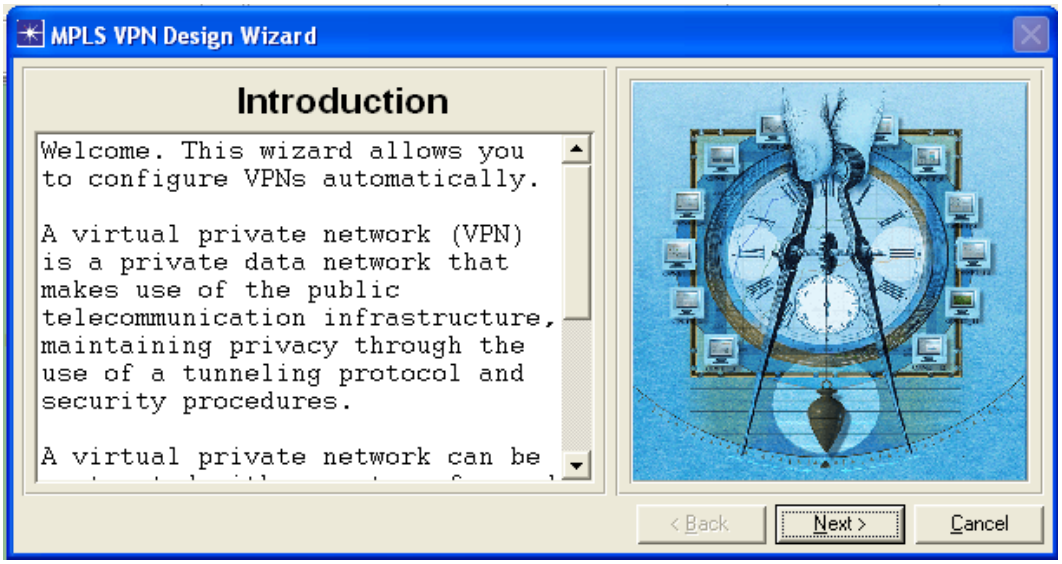
Obr. 5.12: Konfigurace přiřazení portů jednotlivé linky

portů dle tabulky 5.1 a v dalším textu bude pracováno s těmito rozhraními.

Tab. 5.1: Přiřazení síťových rozhraní směrovačů LER

Název LER	**Rozhraní pro propojení s firemní pobočkou**
Cheb LER	IF8
Zlín LER	IF9

5. Dalším naším úkolem bude vyvoření obousměrného MPLS TE tunelu mezi jednotlivými uzly Cheb a Zlín. Tohoto docílíme tak, že vyvoláme následující položky z hlavního menu simulátoru Opnet: **Protocols** → **MPLS** → **Deploy MPLS VPNs...**.
6. Uvítá nás průvodce pro vytvoření MPLS TE tunelu viz. obrázek 5.13. V kontextové nabídce průvodce zvolíme položku **Next** a v následující nabídce zvolíme položku **Layer-2 VPNs** a ponecháme zaškrtnutou položku **Duplicate scenario before changes**. Pokračujeme stisknutím tlačítka **Next**.
7. V následujícím kroku průvodce bude zapotřebí, abychom jako administrátoři sítě MPLS nadefinovali jednotlivé LSR směrovače, které budou daným TE tunelem propojeny. Výběr provedeme stisknutím tlačítka **Add VPN**. Automaticky se nám do tabulky umístí TE tunel se atributy Status-New, VC ID-100, Peers-Specify..., Switching-N/A. Nastavení označení TE tunelu ponecháme, pouze klikneme na klíčové slovo **Specify...** ve sloupci Peers.
8. Otevře se výběrový dialog se žádostí o výběr jednotlivých koncových prvků TE tunelu. Z našeho modelu sítě vybereme prvky **LER Cheb** a **LER Zlín** s použitím tlačítka **»**. Pro jednotlivé vybrané LER směrovače vybereme patřičné síťové rozhraní dle 4. bodu tohoto postupu a vybereme jej jako **Interface** pro daný řádek výběru. Po dokončení pokračujeme stisknutím tlačítka **OK**.



Obr. 5.13: Průvodce pro sestavení TE tunelu

9. Pokračujeme stisknutím tlačítka **Next**.

10. Nastavené údaje TE tunelu překontrolujeme a ukončíme průvodce s pomocí tlačítka **Finish**.

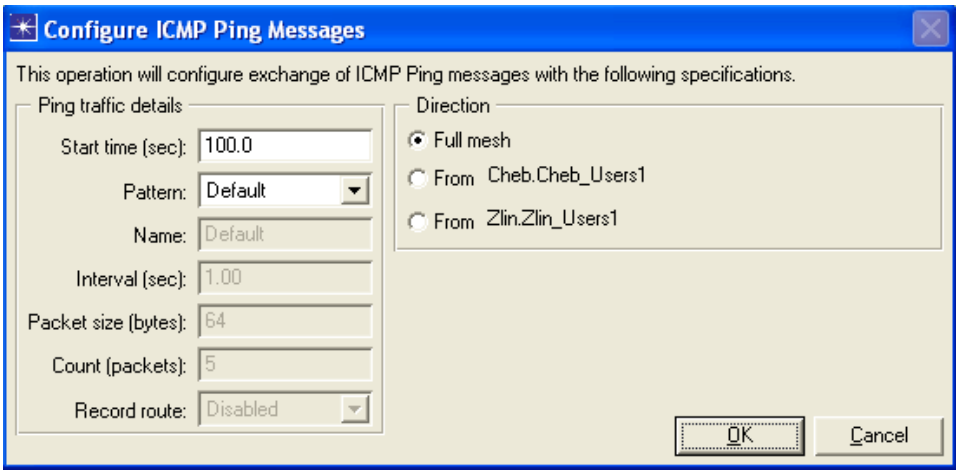
11. Nyní máme obousměrný TE tunel úspěšně vytvořený. Jeho existenci ověříme fyzicky v modelu simulace.

12. Před spuštěním samotné simulace nám zbývá nastavit parametry pro statistiky. Označíme si obě firemní pobočky **FIRMA_POBOCKA_1** a **FIRMA_POBOCKA_2** a pokračujeme s pomocí hlavního menu výběrem sekvence **Protocols** → **IP** → **Demands** → **Configure Ping Traffic on selected Nodes...**.

13. Ve vyvolané nabídce nastavíme parametry pro ICMP PING dle obrázku 5.14, tedy Full mesh, Start time: 100 sec, Pattern: default.

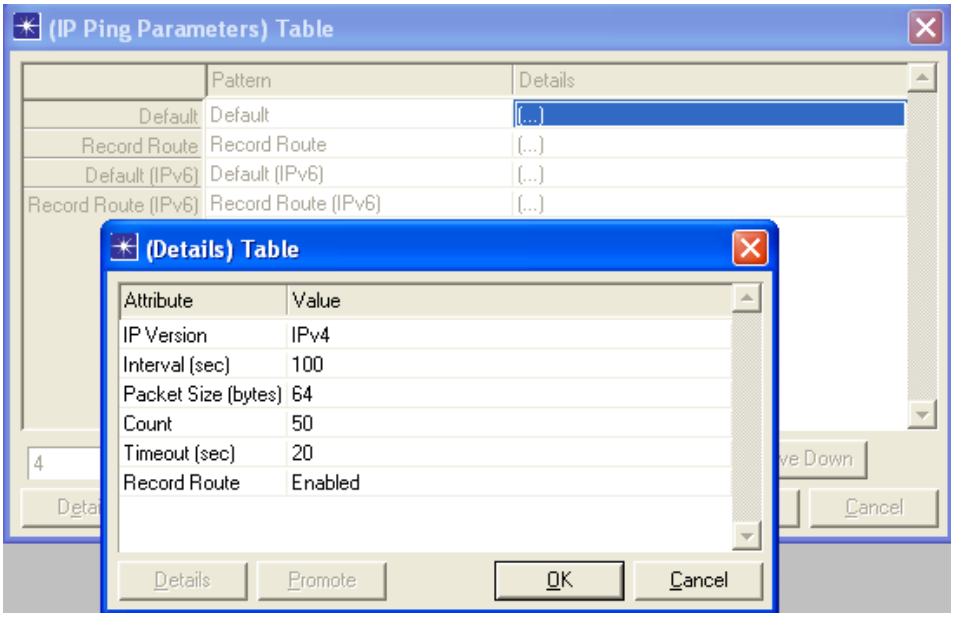
14. Abychom mohli parametry žádosti typu ICMP REQUEST dále a lépe definovat, vložíme do hlavního modelu z palety objektů objekt s názvem **IP Attribute Config**.

15. Zvolením položky **Edit attributes** na právě vloženém objektu přejdeme do konfigurace, kde nastavíme atribut **IP Ping Parameters** na hodnotu **Edit...**. Položku **Default** změníme dle obrázku 5.15. na následující hodnoty:



Obr. 5.14: Konfigurace žádosti ICMP REQUEST mezi pobočkami

IP version: IPv4
Interval (sec): 100
Packet size (bytes): 64
Count: 50
Timeout (sec): 20
Record Route: Enabled
Toto nastavení způsobí, že bude mezi jednotlivými subsítěmi vzájemně odesláno 50 paketů typu ICMP REQUEST s časovým rozdílem 100 sekund.



Obr. 5.15: Nastavení atributů pro PING

16. Pro nastavení statistik zvolíme při současném označení našich firemních poboček a kliknutí pravého tlačítka myši položku **Choose individual DES statistics...** a vybereme **Node statistics** → **IP**. Poté můžeme spustit simulaci s pomocí menu **DES** → **Configure/Run Discrete Event Simulation...**. Nastavíme délku simulace **Duration** na hodnotu **80 minutes** a spustíme simulaci s pomocí tlačítka **Run**. Program OPNET provede simulaci.

17. Pro zobrazení výsledků simulace označíme jednu z firemních poboček a kliknutím pravého tlačítka myši a výběrem volby **View results** zobrazíme výsledky simulace, kde dále pokračujeme po linii **Object Statistics** → **MPLS_SIM** → **Cheb** → **FIRMA_POBOCKA_1** → **IP**. Zaškrtneme volbu **Ping Requests Sent(packets)** a **Ping Requests Received(packets)**. Abychom mohli jednoznačně určit rozdíly mezi odeslanými a přijatými ICMP pakety zvolíme v sekci **Presentation** volbu **Overlaid statistics**.

18. Další statistiky, které si můžeme zobrazit, jsou statistiky pro zpoždění **Ping Response Time (sec)**.

5.4.2 Výsledky simulace

Pro vyhodnocení naší simulace a chování MPLS bude důležité pro začátek shrnout několik důležitých hodnot, které jsme v simulaci nastavili:

- 100 sekund od začátku simulace - sestavení MPLS TE tunelu
- 200 sekund od začátku simulace - start toků na pozadí sítě
- 2880 sekund od začátku simulace - konec toků na pozadí sítě

Z obr. 5.12 můžeme blíže analyzovat funkci sestaveného TE tunelu. Z naší virtuální firemní pobočky č.1 jsou odesílány ICMP pakety (ping request) na pobočku č.2, která odpovídá zpět odesílateli s pomocí paketů ICMP echo reply. Z porovnání obou sloupcových grafů je na první pohled opticky zřejmé, že jednotlivé firemní pobočky mezi sebou mají navázané přímé spojení, které je stabilní a že po celou dobu simulace byly přijaty odpovědi na všechny odeslané PING pakety a nedošlo tak k jakýmkoliv ztrátám na přenosu.

Jiný úhel pohledu na tyto diskutované výsledky - velikost odezvy mezi jednotlivými pobočkami v sekundách - nám dává obr. 5.13, který jasně zobrazuje výborné

parametry MPLS tunelu (hodnota zpoždění kolísá mezi 3 ms až 9 ms) a odpovídá tak praktickým očekáváním reálného systému v provozu.

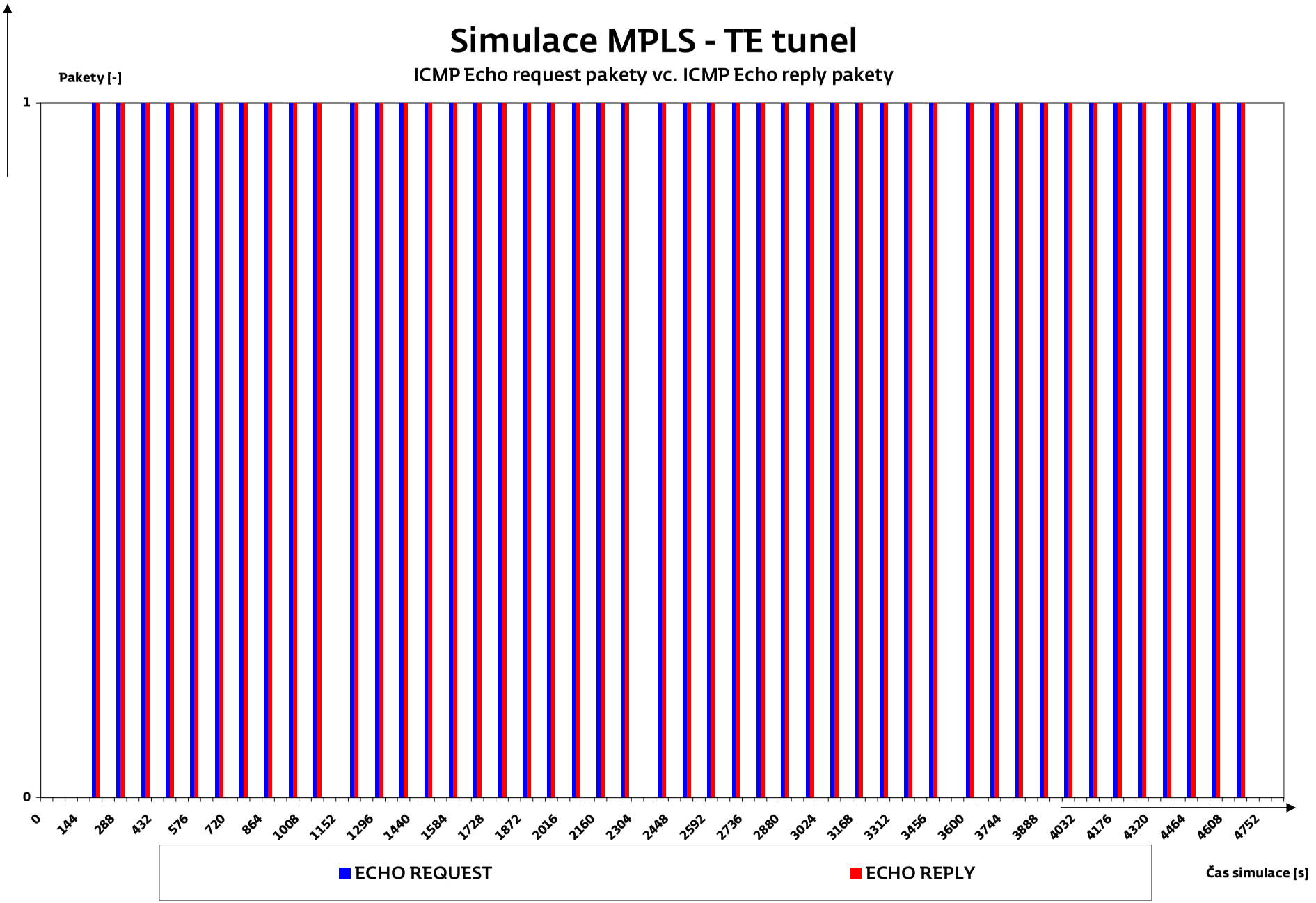
Další zajímavou analýzou je výpis (trace) cest na úrovni MPLS sítě, který nám ukáže fyzické cesty, kterými pakety TE tunelem procházejí. Převzatý výpis ze simulačního rozhraní OPNET reprezentuje tabulka 5.2.

Tab. 5.2: Trasování paketů TE tunelu na úrovni MPLS sítě

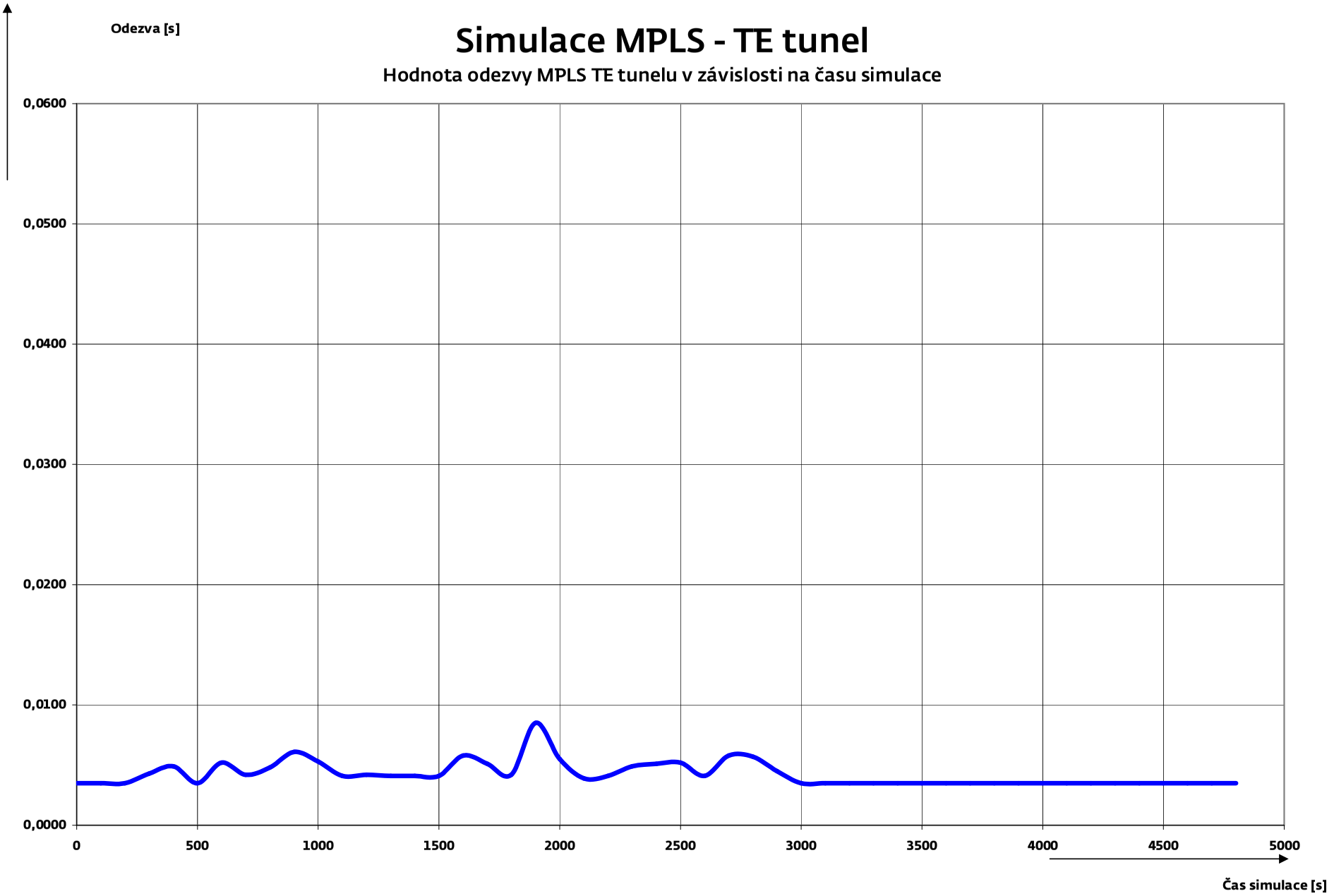
IP adresa	**Název směrovače**	**MPLS Label**
192.0.0.2	Cheb.FIRMA_POBOCKA_1	–
192.0.2.2	Cheb.Cheb LER	17
192.0.5.2	Cheb.Cheb LSR	17
192.0.30.2	Plzen.Plzen LSR	17
192.0.38.2	Ceske_Budejovice.CBud LSR	16
192.0.54.2	Jihlava.Jihlava LSR	16
192.0.68.2	Brno.Brno LSR	16
192.0.69.2	Zlin.Zlin LSR	–
192.0.52.4	Zlin.Zlin LER	–
192.0.52.1	Zlin.FIRMA_POBOCKA_2	–
192.0.52.1	Zlin.FIRMA_POBOCKA_2	–
192.0.69.2	Zlin.Zlin LER	17
192.0.68.1	Zlin.Zlin LSR	17
192.0.54.1	Brno.Brno LSR	17
192.0.38.1	Jihlava.Jihlava LSR	16
192.0.30.1	Ceske_Budejovice.CBud LSR	16
192.0.5.1	Plzen.Plzen LSR	16
192.0.2.1	Cheb.Cheb LSR	–
192.0.0.1	Cheb.Cheb LER	–
192.0.0.2	Cheb.FIRMA_POBOCKA_1	–

Z tabulky 5.2 vyplývá, že pakety při cestě tunelem k příjemci a zpět k odesílateli putují totožnou cestou, která je naprosto symetrická a je automaticky kalkulována MPLS sítí pro dosažení nejmenšího zpoždění přenosu.

Z výše uvedeného vyhodnocení vyplývá, že se nám v simulaci podařilo vytvořit plně funkční TE tunel, který vytváří virtuální propojení dvou firemních poboček. Demonstrací metodou PING jsme ověřili průchodnost trasy a ověřili jsme velikost zpoždění paketů při průchodů sítí, stejně tak jako jejich fyzickou cestu.



Obr. 5.16: Ping mezi jednotlivými firemními pobočkami na konci TE tunelu



Obr. 5.17: Odezva mezi jednotlivými firemními pobočkami na konci TE tunelu

5.5 MPLS s TE tunelem a vliv výpadku páteřní linky

V tomto scénáři použiji pro další rozšíření scénář modelu sítě MPLS s TE tunelem, který jsem vytvořil v kapitole 5.4 této diplomové práce. Cílem této kapitoly zkoumání vlivu výpadku páteřní přenosové linky na model TE tunelu v rámci MPLS sítě. Simulace bude zkoumat dopad výpadku na přenos dat mezi jednotlivými dílčími firemními pobočkami označenými **FIRMA_POBOCKA_1** (umístění v uzlu Cheb) a **FIRMA_POBOCKA_2** (umístění v uzlu Zlín).

5.5.1 Vytvoření scénáře

—TATO ČÁST DIPLOMOVÉ PRÁCE JE NAPSÁNA FORMOU NÁVODU—

1. Nejprve bude zapotřebí zkopírovat scénář vytvořený v kapitole 5.4 této práce. To provedeme s pomocí menu **Scenarios** → **Duplicate Scenario...**. Zadáme jméno nového scénáře na **MPLS_TE_TUNNEL_FAILURE** a potvrdíme. Po provedení této akce se zároveň do daného scénáře přepneme.

2. Model TE tunelu vč. statistik máme již z předchozího scénáře vytvořený, proto nezbývá než zavést do modelu sítě 'umělou' poruchu páteřní linky a přezkoumat s pomocí statistik dopad na celou síť. Přejdeme do palety objektů a vyhledáme objekt s názvem **Failure Recovery**, který umístíme vedle naší hlavní topologie sítě.

3. Označíme si právě vložený objekt Recovery config typu Failure Recovery a zvolíme možnost **Edit Attributes**.

4. Změníme položku **Failure/Recovery Modeling** na hodnotu **Enabled**, čímž simulačnímu prostředí dáme na jevo, že následující konfigurace bude aktivní, a že při simulaci má program kalkulovat s případnými výpadky či obnoveními linek - dle konfigurace.

5. U atributu **Failure/Recovery Specification** klikneme na položku **Edit...** a do nově vzniklé tabulky vložíme dva nové řádky s následujícími záznamy:
   $MPLS_SIM.Plzen <-> C.Budejovice$, Time=1000 s, Status=fail
   $MPLS_SIM.Plzen <-> C.Budejovice$, Time=2000 s, Status=recovery
   Těmito konfiguračními direktivami dáváme simulačnímu prostředí signál k

vytvoření výpadku linky Plzeň-České Budějovice v čase 1000 s od počátku simulace a k jejímu opětovnému obnovení v čase 2000 s.

6. U atributu **Node Failure Mode** musíme zajistit, aby byl tento atribut nastaven na hodnotu **Node Only** a tím máme konfiguraci výpadku páteřního spoje v rámci rozšířeného modelu sítě hotovou.

7. Statistiky máme nastavené již z předchozí simulace. Spustíme simulaci s pomocí menu **DES** → **Configure/Run Discrete Event Simulation...**. Nastavíme délku simulace **Duration** na hodnotu **80 minutes** a spustíme simulaci s pomocí tlačítka **Run**. Program OPNET provede simulaci.

8. Pro zobrazení výsledků simulace označíme jednu z firemních poboček a kliknutím pravého tlačítka myši a výběrem volby **View results** zobrazíme výsledky simulace, kde dále pokračujeme po linii **Object Statistics** → **MPLS_SIM** → **Cheb** → **FIRMA_POBOCKA_1** → **IP**. Zaškrtneme volbu **Ping Requests Sent(packets)** a **Ping Requests Received(packets)**. Abychom mohli jednoznačně určit rozdíly mezi odeslanými a přijatými ICMP pakety zvolíme v sekci **Presentation** volbu **Overlaid statistics**.

9. Další statistiky, které si můžeme zobrazit, jsou statistiky pro zpoždění **Ping Response Time (sec)**.

5.5.2 Výsledky simulace

Pro vyhodnocení naší simulace a chování MPLS bude důležité pro začátek shrnout několik důležitých hodnot, které jsme v simulaci nastavili:

- 100 sekund od začátku simulace - sestavení MPLS TE tunelu
- 200 sekund od začátku simulace - start toků na pozadí sítě
- 1000 sekund od začátku simulace - výpadek linky Plzeň-České Budějovice
- 2000 sekund od začátku simulace - obnovení linky Plzeň-České Budějovice
- 2880 sekund od začátku simulace - konec toků na pozadí sítě

Na obr. 5.18 můžeme sledovat funkci sestaveného TE tunelu, kdy z firemní pobočky č.1 odchází ICMP pakety na pobočku č.2, která na jejich příchod odpovídá (PING). Z porovnání obou sloupcových grafů vidíme, že pobočky mezi sebou mají navázané přímé spojení, PING pakety prochází, ovšem v okamžiku výpadku páteřní trasy

dojde při přenosu k výpadku jednoho z paketů. V tomto okamžiku síť provádí rekalkulaci TE tunelu a navázání nového spojení, což je přímým důsledkem pozorovaných ztrát. V případě znovuobnovení původně poruchové linky dojde pro změnu ke značnému zpoždění čtyř paketů, což je opět přímý důsledek přesměrování toku dat. Tyto výsledky demonstrují, že při plně saturovaných přenosových trasách není pro MPLS tunel snadné nalézt náhradní trasu, nicméně se to podaří, bez toho, aniž by došlo k větší degradaci kvality dodávaných služeb.

Jiný úhel pohledu na tuto simulaci - velikost odezvy mezi jednotlivými pobočkami v sekundách - nám dává obr. 5.19. V případě simulace výpadku páteřní linky dosáhnou v okamžiku přesměrování provozu odezvy hodnot až 20000 ms, což je pro jakékoliv hlasové přenosy degradující. Tento jev v simulaci si vysvětluji tak, že vzhledem k tomu, že společně s VoIP hovorem se TE tunelem šíří další, velký objem dat, není takto úzký přenosový kanál schopen v krátkém časovém horizontu situaci stabilizovat. Dalším aspektem je poté systém detekce chyb a systém obnovy použitý pro simulaci. Užité systémy mají menší systémové nároky, ale zároveň se vyznačují delší dobou, která je zapotřebí pro obnovení. V praxi se tak situace běžně řeší vytvořením separátních TE tunelů pro jednotlivé druhy provozu (např. data, hlas..), čímž dojde k vzájemné priorizaci přenosových cest a interakci tunelů v souvislosti s kapacitou jednotlivých linek.

Způsob, jakým se bude síť chovat v případě výpadku jedné z páteřních linek (Plzeň-České Budějovice), nám blíže přiblíží tabulka 5.3.

Tab. 5.3: Trasování paketů TE tunelu na úrovni MPLS sítě

IP adresa	**Název směrovače**	**MPLS Label**
192.0.0.2	Cheb.FIRMA_POBOCKA_1	–
192.0.2.2	Cheb.Cheb LER	17
192.0.5.2	Cheb.Cheb LSR	17
192.0.30.2	Plzen.Plzen LSR	17
192.0.38.2	Ceske_Budejovice.CBud LSR	16
192.0.54.2	Jihlava.Jihlava LSR	16
192.0.68.2	Brno.Brno LSR	16
192.0.69.2	Zlin.Zlin LSR	–
192.0.52.4	Zlin.Zlin LER	–
192.0.52.1	Zlin.FIRMA_POBOCKA_2	–
192.0.52.1	Zlin.FIRMA_POBOCKA_2	–
192.0.69.2	Zlin.Zlin LER	–
192.0.68.1	Zlin.Zlin LSR	–
192.0.53.1	Brno.Brno LSR	–
192.0.7.2	Praha.Praha LSR	–
192.0.5.1	Plzen.Plzen LSR	–
192.0.2.1	Cheb.Cheb LSR	–
192.0.0.1	Cheb.Cheb LER	–
192.0.0.2	Cheb.FIRMA_POBOCKA_1	–

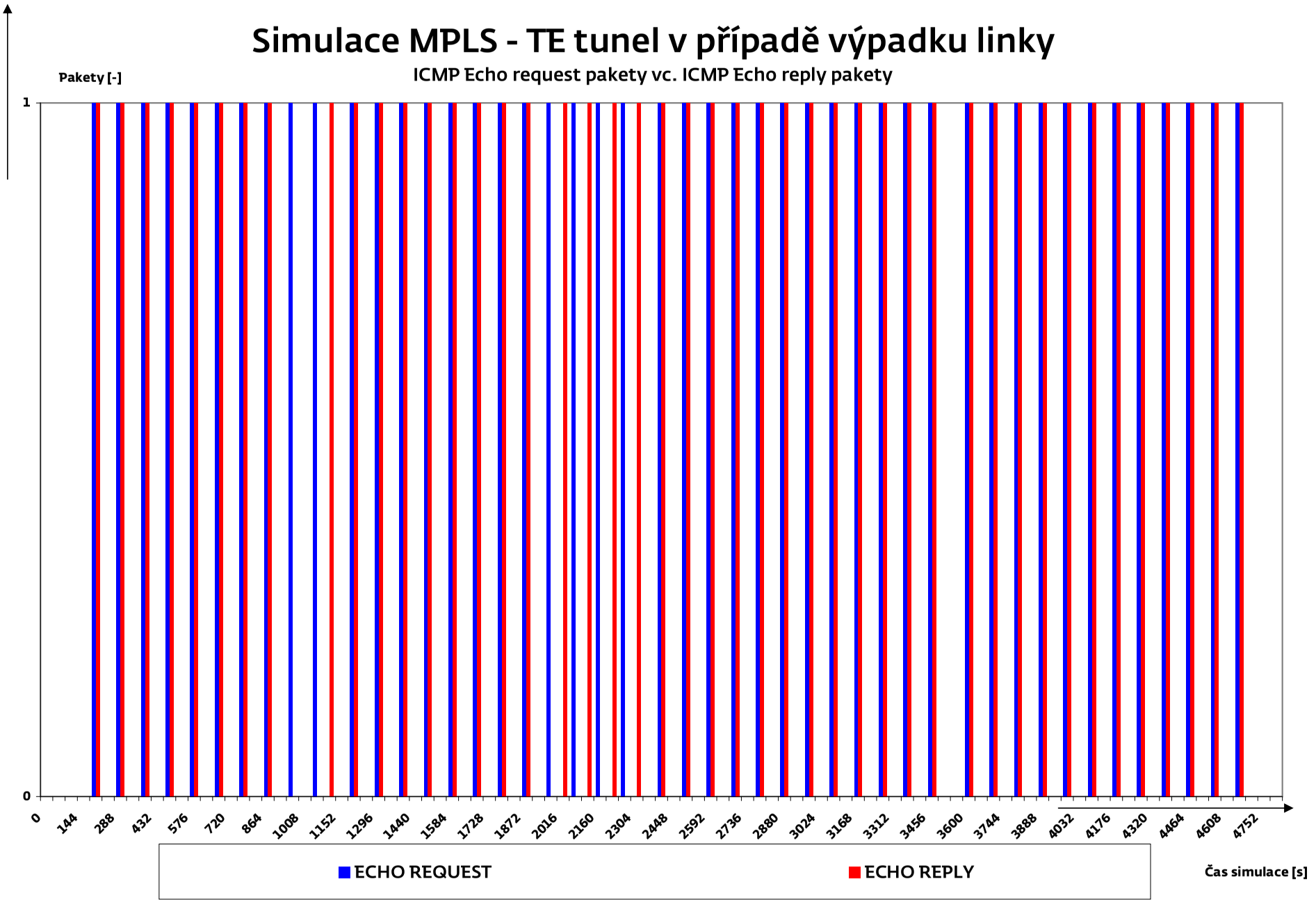
Tabulka ukazuje, že v případě výpadku páteřní linky si MPLS vytvořila novou cestu pro vlastní TE tunel. Ten je při cestě paketů zpět využit pro přenos námi zkoumaného datového provozu.

Cestu, kterou MPLS síť pro TE tunel v rámci situace sestaví lze přitom ovlivnit - je určena dvěma faktory:

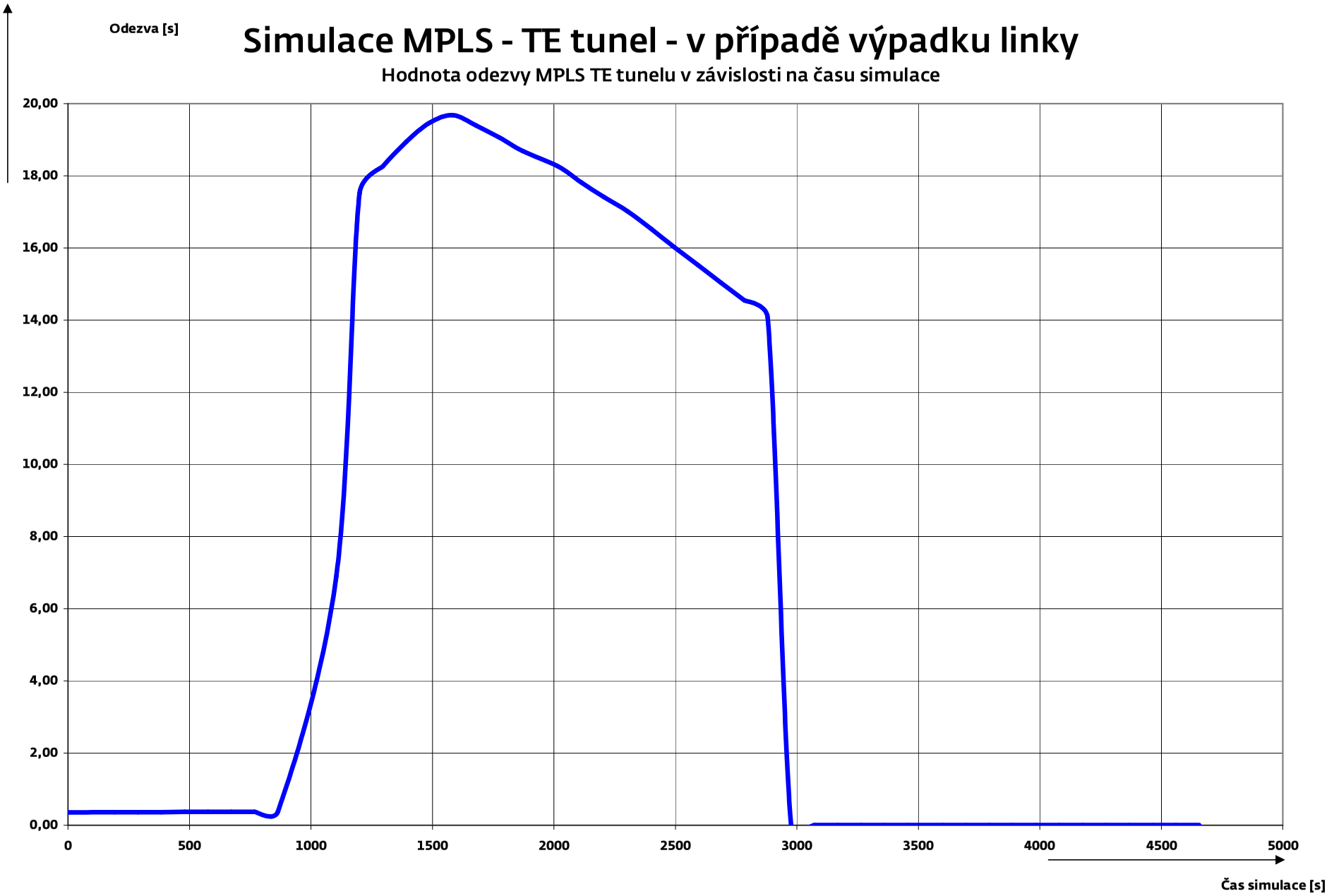
1. IGP metrikou linky (v našem případě dána OSPF protokolem)
2. TE metrikou tunelu

V případě, že není nastavena TE metrika tunelu, pak automaticky tato hodnota odpovídá IGP metrice linky. Užití dvou metrik pro přenosové linky je implementováno u MPLS z toho důvodu, že lze rozlišit typy aplikací a tak například hlasový přenos může v rámci TE tunelu o své cestě rozhodovat s pomocí TE metriky tunelu, navázané na kalkulaci zpoždění dané přenosové linky, zatímco běžná data přenášená s pomocí jiného TE tunelu mohou být směrována s užitím IGP metriky. Různé druhy provozu tak mohou mít různé cesty a váhy linek v závislosti na typu provozu. OPNET umožňuje jak nastavení IGP metriky pro každý LSP, tak i nastavení TE metriky pro daný TE tunel **LSP metric**, **TE parameters**, **HOP Limit**. V případě naší simulace tak dojde k vytvoření nové cesty v rámci MPLS sítě na základě IGP metriky linky, protože explicitní nastavení metriky TE tunelu nebylo provedeno.

Z výše uvedeného vyhodnocení vyplývá, že se nám v simulaci podařilo vytvořit plně funkční TE tunel, který vytváři virtuální propojení dvou firemních poboček. Během simulace došlo k výpadku jedné z přenosových tras, přes kterou TE tunel prochází. Demonstrací metodou PING jsme ověřili průchodnost trasy a zároveň jsme odhalili ztrátovost a nárůst zpoždění, který nastává při rekalkulaci přenosových tras v souvislosti s výpadkem. Dále jsme simulovali a vyhodnotili velikost zpoždění při průchodu paketů mezi firemními pobočkami v závislosti na čase simulace, které je neuspokojivé a pro přenos hlasového hovoru nevyhovující, nicméně pro danou simulaci odpovídající vzhledem ke stoprocentní saturaci všech přenosových tras. Na závěr jsme přezkoumali fyzickou trasu paketů v rámci MPLS TE tunelu a došli jsme k závěru, že trasa je vzhledem k výpadku jedné z přenosových tras pozměněná a nesymetrická, což potvrzuje funkčnost rekalkulace tunelu v rámci sítě.



Obr. 5.18: Ping při výpadku páteřní linky mezi jednotlivými firemními pobočkami na konci TE tunelu



Obr. 5.19: Odezva mezi jednotlivými firemními pobočkami na konci TE tunelu - simulace při výpadku linky

6 ZÁVĚR

Jak bylo uvedeno v úvodu této diplomové práce, jejím cílem bylo seznámit čtenáře se základními principy protokolu MPLS a odhalit souvislosti s řízením kvality služeb. Čtenář měl možnost s pomocí uvedených návodů vytvořit pět scénářů simulace MPLS sítě. První ze scénářů nebral v potaz řízení kvality služeb, zatímco druhý scénář řízení kvality služeb přímo implementoval. Třetí scénář zkoumal vliv výpadku páteřní linky na MPLS síť jako takovou a kvalitu služeb v ní obsaženou. Čtvrtý a pátý scénář se zabýval TE tunely, jejich sestavením, funkcí a reakcí v rámci MPLS sítě na výpadek linky v přenosové cestě.

Z poznatků uvedených ve vyhodnoceních jednotlivých simulací je zřejmé, proč dnešní telekomunikační operátoři mohutně nasazují protokol MPLS na své přenosové sítě. Kromě možnosti propojení několika typů různých sítí MPLS nasazuje mocný nástroj pomocí nativního zpracování QoS, kdy dochází k přenosu ToS třídy společně s daty již v hlavičcce MPLS. Asi největší silou MPLS je rychlejší zpracování dat než je tomu u klasických dynamicky-směrovacích metod a možnost využití Traffic Engeneeringu, který reprezentuje silný nástroj pro adaptace služeb na základě přání zákazníka. Simulované hodnoty odpovídají teoretickým očekáváním a do jisté míry i mým osobním očekáváním z praktického hlediska a vzhledem k dosavadním zkušenostem, které jsem měl možnost v oblasti síťových technologií načerpat. OPNET je skvělým simulačním nástrojem a v akademických i komerčních sférách má své jisté využití pro simulace sítí různých rozsahů a charakteristik.

Porovnáním výsledků simulací této práce se mohl čtenář přesvědčit o vlivu nastavení parametru ToS na řízení kvality služeb v takovéto síti, o vlivu výpadku linky na chování sítě jako takové a v neposlední řadě se čtenář seznámil se základními a rozšířenými principy MPLS sítí vč. praktického dopadu na možnost dosažení kvality služeb.

MPLS technologie se jeví jako velmi perspektivní, a proto je v dnešní době tak rozšířeným nástrojem, který umožňuje poskytovatelům služeb uspokojovat neustále se zvyšující dynamické nároky jejich uživatelů. Dle informací uvedených v [6] lze podle současného stavu evropského projektu Géant považovat MPLS za hlavní přenosovou technologii mezinárodní páteřní sítě. Rovněž většina národních výzkumných a vzdělávacích sítí již nasadilo MPLS, aby tak mohla čerpat z předností této technologie.

REFERENCE

[1] BOLDIŠ, P. *Bibliografické citace dokumentů podle ČSN ISO 690 a ČSN ISO 690-2* [online]. 2001, poslední aktualizace 11.11.2004 [cit. 17.2.2005]. Dostupné z URL: <http://www.boldis.cz/citace/citace.html>.

[2] PEPELNIAK, I.; GUICHARD, J. *MPLS and VPN Architectures*. 1. vyd. Indianapolis: Cisco Press, 2000. ISBN: 978-1587050022.

[3] ROSEN, E.; VISWANATHAN, A.; CALLON, R. *Multiprotocol Label Switching Architecture*. IETF Networking Group [online]. 2001 [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc3031.txt>.

[4] BRADLEY, M. *Quality of Service*. about.com [online]. 2007 [cit. 17.4.2010]. Dostupné z URL: <http://computernetworking.about.com/od/networkdesign/g/bldef_qos.htm>.

[5] Skupina autorů WIKIPEDIA *Wikipedia - MPLS*. wikipedia.org [online]. 2009, [cit. 17.4.2010]. Dostupné z URL: <http://en.wikipedia.org/wiki/Multiprotocol_Label_Switching>.

[6] SHARMA, V.; HELLSTRAND, F.; METANOIA INC.; NORTEL NETWORKS *RFC3469 - Framework for MPLS based Recovery*. IETF Networking Group [online]. únor 2003, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc3469.txt>.

[7] WELCHER, J., P. *Introduction to MPLS*. Vysoká škola báňská - Technická univerzita Ostrava [online]. 2000, [cit. 17.4.2010]. Dostupné z URL: <http://www.cs.vsb.cz/grygarek/TPS/MPLS/Introduction_to_MPLS.htm>.

[8] IETF Networking Group *Multiprotocol Label Switching*. IETF Networking Group [online]. 2009, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/dyn/wg/charter/mpls-charter.html>.

[9] Skupina autorů WIKIPEDIA *Internet*. wikipedia.org [online]. 2009, [cit. 17.4.2010]. Dostupné z URL: <http://cs.wikipedia.org/wiki/Internet>.

[10] IEEE802.ORG *802.1p* ieee802.org [online]. 2009, [cit. 17.4.2010]. Dostupné z URL: <http://www.ieee802.org/1/>.

[11] ROSEN, E.; REKHTER, Y. *RFC2547 - BGP/MPLS VPNs*. IETF Networking Group [online]. březen 1999, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc2547.txt>.

[12] AWDUCHE, E.; MALCOLM, J.; AGOGBUA, J.; O'DELL, M.; MCMANUS, J.; UUNET *RFC2707 - Requirements for Traffic Engineering Over MPLS*. IETF Networking Group [online]. září 1999, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc2702.txt>.

[13] DEMICHELIS, C.; CHIMENTO, P.; TELECOMITALIA LAB; ERICSSON IPI *RFC3393 - IP Packet Delay Variation Metric for IP Performance Metrics*. IETF Networking Group [online]. 2002, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc3393.txt>.

[14] Skupina autorů WIKIPEDIA *Packet delay variation*. wikipedia.org [online]. 6.10.2009, [cit. 17.4.2010]. Dostupné z URL: <http://en.wikipedia.org/wiki/Packet_delay_variation>.

[15] AWDUCHE, D.; BERGER, L.; GAN, D.; LI, T.; SROMOVASAN, V.; SWALLOW, G. *RSVP-TE: Extensions to RSVP for LSP Tunnels (RFC 3209)*. RFC [online]. 2001, [cit. 17.4.2010]. Dostupné z URL: <http://rfc-3209.rfc-list.net/>.

[16] BRADEN, R.; ZHANG, L.; BERSON, S.; HERZOG, S.; JAMIN, S. *Resource Reservation Protocol (RSVP) (RFC2205)*. RFC [online]. 1997, [cit. 17.4.2010]. Dostupné z URL: <http://rfc-2205.rfc-list.net/>.

[17] LE FAUCHEUR, F., ED.; CISCO SYSTEMS, INC. *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*. IETF Networking Group [online]. 2005, [cit. 17.4.2010]. Dostupné z URL: <http://tools.ietf.org/html/rfc4124>.

[18] LE FAUCHEUR, F., ED.; CISCO SYSTEMS, INC.; LAI, W. *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*. IETF Networking Group [online]. 2005, [cit. 17.4.2010]. Dostupné z URL: <http://tools.ietf.org/html/rfc4125>.

[19] LE FAUCHEUR, F, ED.; CISCO SYSTEMS, INC. *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*. IETF Networking Group [online]. 2005, [cit. 17.4.2010]. Dostupné z URL: <http://tools.ietf.org/html/rfc4127>.

[20] HABERMAN, B.; THALER, D. *RFC3036 - LDP Specification*. IETF Networking Group [online]. 2001, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc3036.txt>.

[21] ASH, J.; LEE, Y.; ASHWOOD-SMITH, P.; JAMOUSSI, B.; FEDYK, D.; SKALECKI, D.; LI, L. *RFC3214 - LSP Modification Using CR-LDP*. IETF Networking Group [online]. 2002, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc3214.txt>.

[22] REKHTER, Y.; AGGARWAL, R. *RFC4781 - Graceful Restart Mechanism for BGP with MPLS*. IETF Networking Group [online]. 2007, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc4781.txt>.

[23] SANGLI, S.; CHEN, E.; FERNANDO, R.; SCUDDER, J.; REKHTER, Y. *RFC4724 - Graceful Restart Mechanism for BGP*. IETF Networking Group [online]. 2007, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc4724.txt>.

[24] KOMPELLA, K.; SWALLOW, G. *RFC4379 - Detecting MPLS Data Plane Failures*. IETF Networking Group [online]. 2006, [cit. 17.4.2010]. Dostupné z URL: <http://www.ietf.org/rfc/rfc4379.txt>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ATM	Asynchronous transfer mode - standard pro vysokorychlostní síť ovou architekturu
BGP	Border Gateway Protocol
CBR	Constant bit rate
CLP	Cell Loss Priority
CoS	Class Of Service
DiffServ	IP Differentiated Services
DSCP	DiffServ Code Point
FEC	Forward equivallence class
HomePNA	Home networking over coaxial and phone lines
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IntServ	IP Integrated Services
IPTV	IP Television
IS-IS	Intermediate system to intermediate system - protokol využívaný směrovači k určení nejlepší přenosové cesty
LDP	Label Distribution Protocol
LER	Label Edge Router
LIB	Label Information Base
LSP	Label Switched Path
LSR	Label Switch Router
MPLS	Multiprotocol label switching
OSI	Open system interconnection reference model
OSPF	Open Shortest Path First

PML	Path Merge LSR - směrovač LSR zopodvědný za propojení nové záložní cesty s částí původní funkční přenosové trasy
PSL	Path Switch LSR - směrovač LSR v topologii za poruchou, který je zodpovědný za přepínání nebo přesměrování provozu z pracovní cesty na cestu záložní
QoS	Quality of Service
RSVP	Resource reservation protocol
RSVP-TE	Resource reservation protocol - Traffic Engeneering
SDH	Synchronous Digital Hierarchy
SONET	Synchronous Optical Networking
TCP/IP	Transmission Control Protocol / Internet Protocol
TE	Traffic Engeneering
TTL	Time To Live
UDP	User Datagram Protocol
VoIP	Voice over IP

A PŘÍLOHY

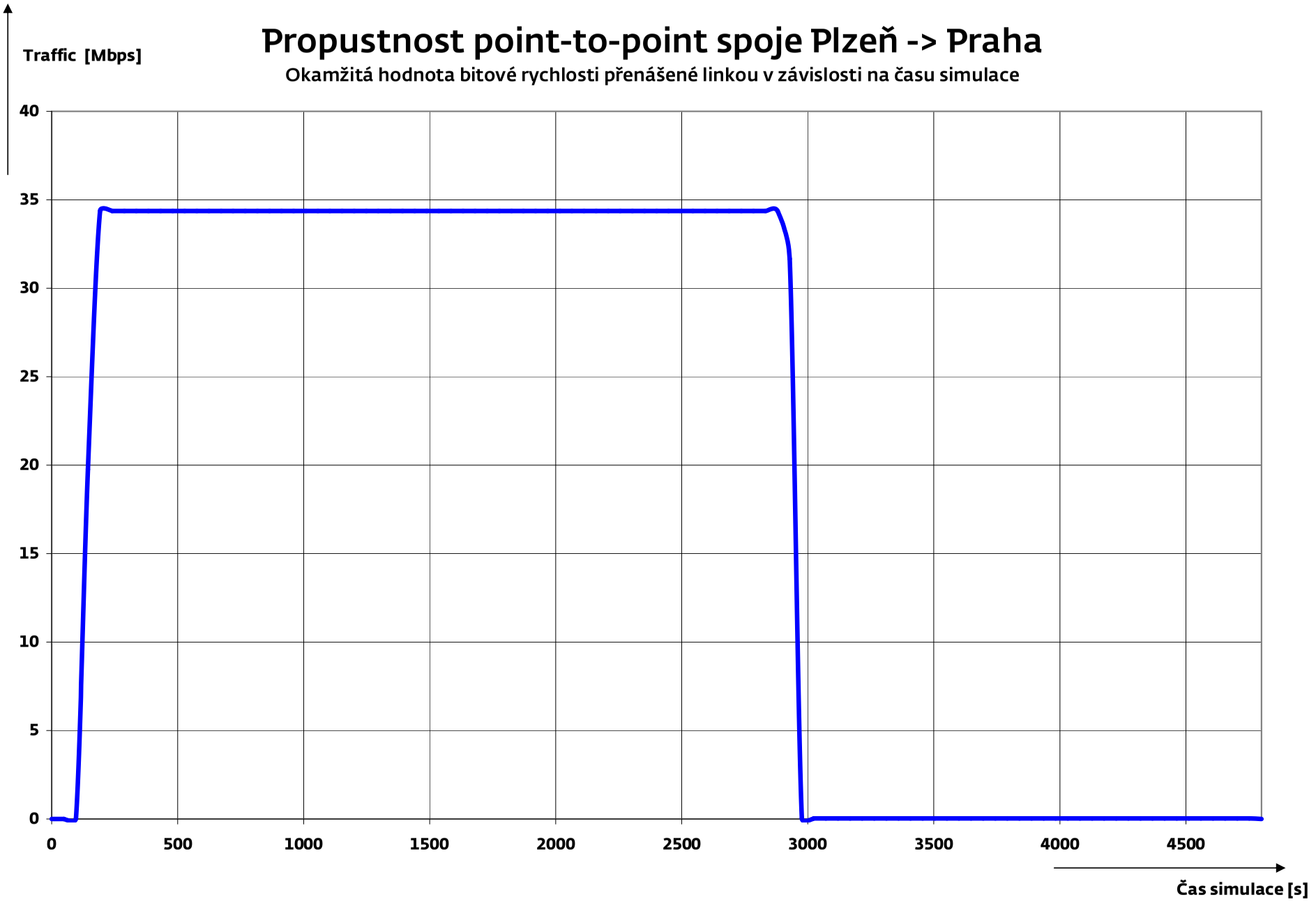
A.1 Propustnost point-to-point spoje Plzeň-Praha v závislosti na času

A.2 Propustnost point-to-point spoje Plzeň-České Budějovice v závislosti na času

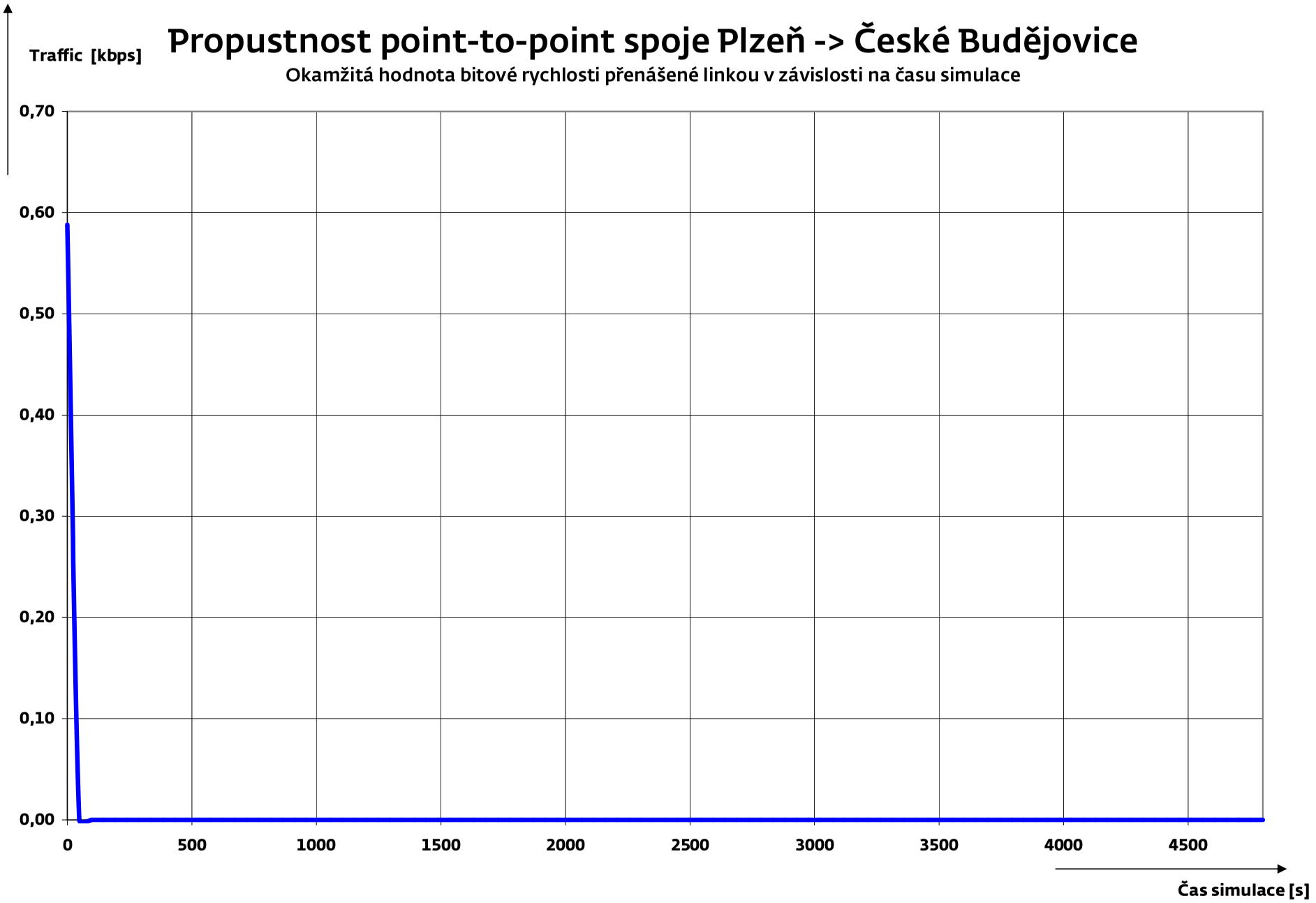
A.3 Zatížení point-to-point spoje Plzeň-Praha v závislosti na času

A.4 Packet Delay Variation hlasových služeb simulace v závislosti na času

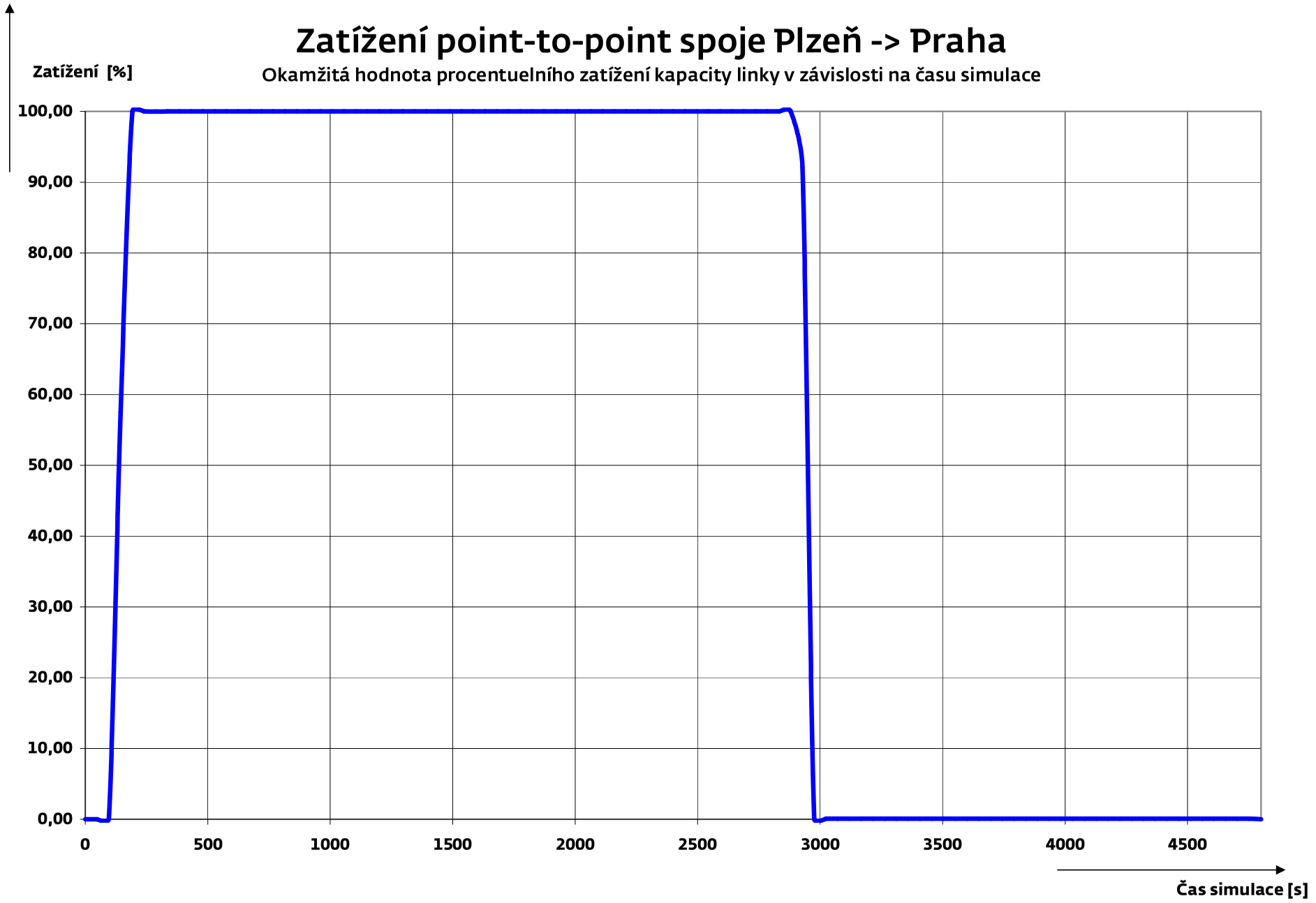
A.5 End-to-end zpoždění hlasových služeb simulace v závislosti na času



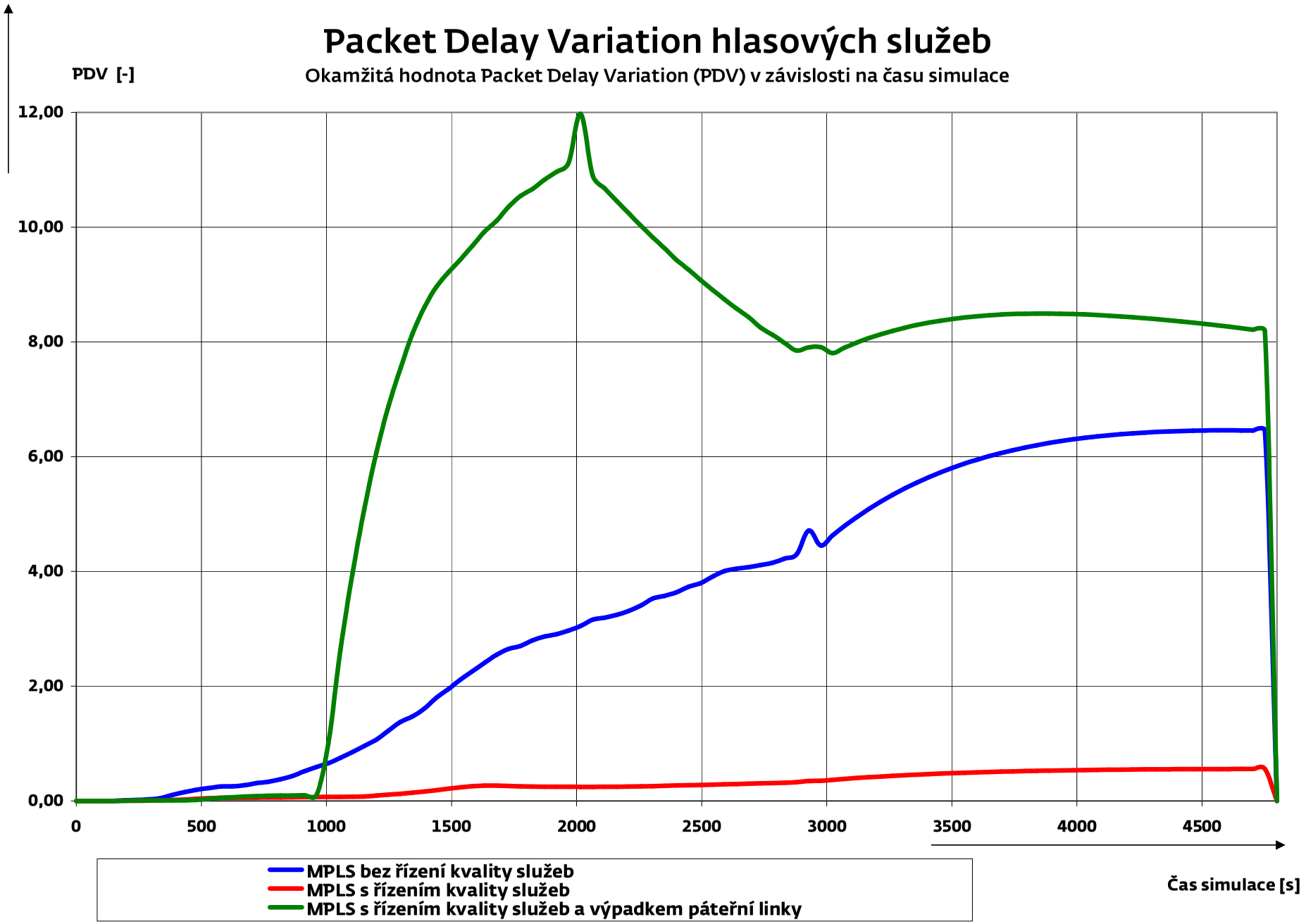
Obr. A.1: Propustnost point-to-point spoje Plzeň-Praha v závislosti na času



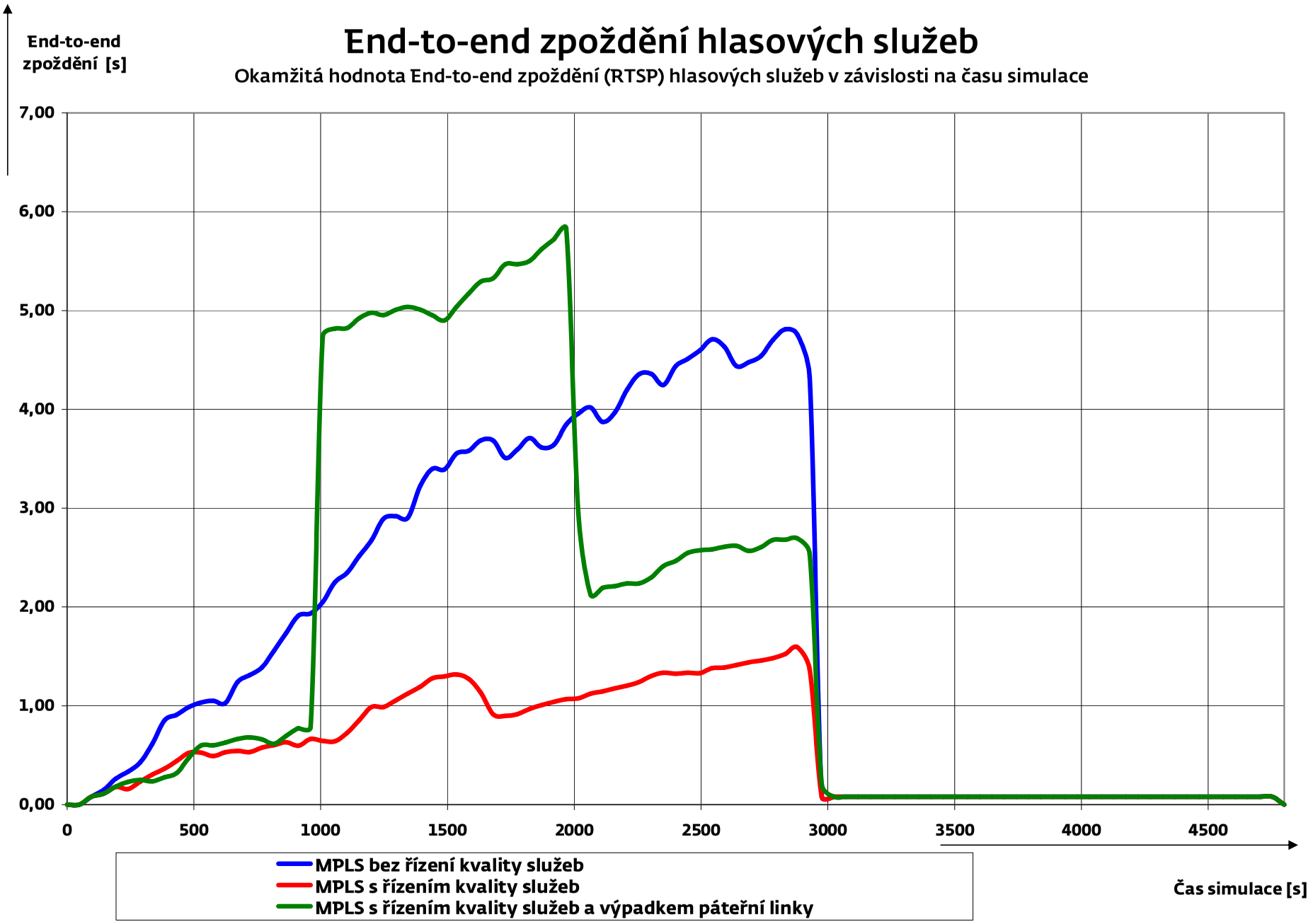
Obr. A.2: Propustnost point-to-point spoje Plzeň-České Budějovice v závislosti na času



Obr. A.3: Zatížení point-to-point spoje Plzeň-Praha v závislosti na času



Obr. A.4: Packet Delay Variation hlasových služeb simulace v závislosti na času



Obr. A.5: End-to-end zpoždění hlasových služeb simulace v závislosti na času