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

SIMULAČNÍ SCÉNÁŘE V PROSTŘEDÍ IT GURU PRO PŘEDMĚT KOMUNIKAČNÍ TECHNOLOGIE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ŠTĚPÁN RYBKA

BRNO 2013

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

SIMULAČNÍ SCÉNÁŘE V PROSTŘEDÍ IT GURU PRO KOMUNIKAČNÍ TECHNOLOGIE

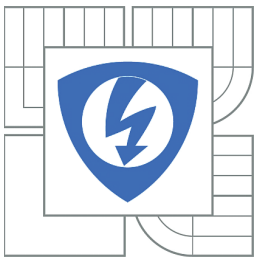
SIMULATION SCENARIOS IN IT GURU ENVIRONMENT FOR COMMUNICATION TECHNOLOGY COURSE

DIPLOMOVÁ PRÁCE
MASTER´S THESIS

AUTOR PRÁCE
AUTHOR
Bc. ŠTĚPÁN RYBKA

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. JAN JEŘÁBEK, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Štěpán Rybka ***ID:*** 119593
Ročník: 2 ***Akademický rok:*** 2012/2013

NÁZEV TÉMATU:

Simulační scénáře v prostředí IT Guru pro předmět Komunikační technologie

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s obsahem předmětu Komunikační technologie. V rámci diplomové práce navrhněte a detailně zpracujte 4 úlohy pro prostředí OPNET tak, aby byly funkční ve verzi IT Guru. Vytvořte ke každé úloze i vzorové řešení a návod k řešení pro studenty. Úlohy musí v širším rámci spadat do obsahu předmětu a délkou musí odpovídat tříhodinovému cvičení předmětu. Zaměřte se především na témata, jako jsou srovnání různých aplikačních protokolů, srovnání různých nastavení protokolů pro přenos hlasu, srovnání různých síťových prvků, porovnání vlastností jednotlivých síťových technologií, vliv topologie na funkčnost sítě, různé přístupy k řešení WAN sítí či směrování různých úrovní. Ve všech vytvořených úlohách klaďte důraz na vysvětlení podstaty problému.

DOPORUČENÁ LITERATURA:

[1] JEŘÁBEK, J. Pokročilé komunikační techniky, Skriptum FEKT VUT v Brně, 194 stran, 2013.
[2] FOROUZAN, B. A. TCP/IP Protocol Suite, McGraw-Hill Higher Education, Boston, 4. vydání, 979 stran, 2010.
[3] OPNET Technologies, OPNET Modeler Product Documentation Release 16.0, 2010.

Termín zadání: 11.2.2013 ***Termín odevzdání:*** 29.5.2013

Vedoucí práce: Ing. Jan Jeřábek, Ph.D.
Konzultanti diplomové práce:

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

UPOZORNĚNÍ:

ABSTRAKT

Tato diplomová práce s názvem Simulační scénáře v prostředí IT Guru pro Komunikační technologie se zabývá návrhem laboratorních úloh pro výše zmíněný předmět. V první části práce je teoretický úvod k použitým protokolům, v praktické části jsou samotné úlohy. Práce obsahuje 4 laboratorní úlohy s detailním popisem postupu práce a názornými obrázky. První dvě úlohy se zaměřují na aplikační protokoly, s kterými přijde uživatel každodenně do styku. Další dvě se orientují na směrovací protokoly, konkrétně na vnitřní směrovací protokol IS-IS a vnější směrovací protokol BGP. K tvorbě byla použita bezplatná verze simulačního programu OPNET Modeler, IT Guru, verze 9.1.A.

KLÍČOVÁ SLOVA

Protokol, laboratorní úloha, OPNET, IT Guru, směrování

ABSTRACT

This thesis called Simulation scenarios in an IT Guru enviroment for Communication Technologies deals with the design of laboratory experiments for the above-mentioned subject. The first part is a theoretical introduction to the protocols. The practical part contains the tasks itself. The work includes four laboratory tasks with a detailed description of the work and an illustrative pictures. The first two tasks are focused on the application protocols, with which the user comes into contact with every day. The other two are focused on routing protocols, precisely the internal routing protocol IS-IS and BGP external routing protocol. A free version of simulation software OPNET Modeler and IT Guru enviroment version 9.1.A were used to create the work.

KEY WORDS

Protocol, laboratory exercise, OPNET, IT Guru, routing

RYBKA, Š. *Simulační scénáře v prostředí IT Guru pro předmět Komunikační technologie.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 102 s. Vedoucí diplomové práce Ing. Jan Jeřábek, Ph.D..

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Simulační scénáře v prostředí IT Guru pro předmět Komunikační technologie jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této semestrální práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona c. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonu (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

..... podpis autora

Mé poděkování patří vedoucímu práce Ing. Janu Jeřábkovi, Ph.D., jehož cenné rady přispěly ke vzniku této práce.

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Purkynova 118, CZ-61200 Brno, Czechia

http://www.six.feec.vutbr.cz

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

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Důvodem zvolení této diplomové práce byla skutečnost, že jsem chtěl vytvořit dílo, které by bylo dále využíváno pro vzdělávání studentů. Cílem diplomové práce je vytvoření laboratorních úloh pro předmět Komunikační technologie. Tento předmět je vyučován v zimním semestru druhého ročníku bakalářského studijního oboru Teleinformatika. Komunikační technologie je jedním z prvních předmětů, kde se studenti seznamují s komunikačními modely, strukturou sítí a základním popisem síťových modelů.

V úvodní části se věnuji ukázce simulačního programu OPNET IT Guru, verze 9.1.A a jeho hlavních částí, který je použit pro tvorbu laboratorních úloh. Pro použité protokoly, které jsou využívány v laboratorních úlohách, je vypracován teoretický úvod. První dvě úlohy se zaměřují na seznámení s aplikačními protokoly Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Telecomunication Network (TELNET) a Voice over Internet Protocol (VoIP). Další dvě se věnují směrovacím protokolům Intermediate System to Intermediate System (IS-IS) a Border Gateway Protocol (BGP).

Laboratorní úlohy jsou rozděleny do několika po sobě jdoucích kroků, návrh sítí, simulování a vyhodnocení výsledků. V úlohách používám protokoly, které jsou popsány v teoretickém úvodu. Po zvládnutí úlohy student porozumí dané problematice a své poznatky bude konzultovat s vyučujícím. K laboratorním úlohám je vytvořen podrobný návod vypracování. V posledních dvou je na závěr vytvořen samostatný úkol, který bude vybízet studenty k zamyšlení se nad problematikou. Pro tyto dvě úlohy je vypracováno vzorové řešení samostatných úkolů. Vzhledem k náročnosti konfigurování síťového provozu je pro poslední dvě úlohy připravena přednastavená síť. Všechny zdrojové kódy vytvořených úloh v programu IT Guru jsou přiloženy na CD.

1 Program Opnet IT Guru

Simulační prostředí Opnet IT Guru je výrobkem společnosti OPNET Technologies, Inc. Firma je předním poskytovatelem simulačních řešení pro vytváření sítí, síťových provozů a aplikací v nich používaných [1]. Program Opnet poskytuje široký přehled nad všemi druhy sítí, monitorování jejich infrastruktury, hluboký sběr dat a jejich následnou analýzu. To vše umožňuje perfektní simulování a určení příčin problémů.

Společnost nabízí dvě verze programu, Opnet Modeler a Opnet Modeler IT Guru. První z nich je program, který umožňuje analyzovat simulované sítě a porovnávat vliv různých technologických návrhů [2]. Obsahuje širokou sadu protokolů a technologií prakticky pro všechny typy sítí, zahrnující například VoIP, TCP, OSPFv3, MPLS, IPv6 a mnoho dalších. Druhým je Opnet Modeler IT Guru, což je ochuzená verze programu Opnet Modeler. Je navržen pro vytváření a testování na úrovni školních laboratorních manuálů [3]. Její hlavní výhoda je bezplatné stažení z oficiálních stránek firmy. OPNET Technologies umožňuje 6-ti měsíční obnovitelnou licenci. Díky těmto vlastnostem IT Guru nabízí značnou výhodu pro studenty, bezplatně si stáhnout program a používat zdarma kvalitní simulační program i mimo školní budovu. Proto jsme pro řešení laboratorních úloh zvolili tuto ochuzenou verzi.

1.1 Prostředí programu IT Guru

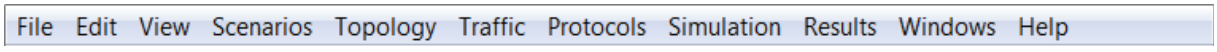
1.1.1 Grafické rozhraní

Hlavní okno po zapnutí programu OPNET IT Guru se jmenuje System window [4]. V tomto okně můžeme vytvářet nové projekty, File → New → Project, nebo otevírat projekty rozpracované, viz obr. 1.1.

Obr. 1.1: System window programu IT Guru

1.1.2 Project Editor

Okno Project Editor je hlavním oknem programu, zde se prakticky vytvářejí nové projekty, vybírají se požadované statistiky a spouští se tu simulace. Celý Project editor můžeme rozdělit na několik důležitých částí. Hlavní lišta, viz obr. 1.2, pomocí které můžeme jako v jiných programech, přistupovat k dalším funkcím programu.



Obr. 1.2: Hlavní lišta programu IT Guru

Další částí jsou akční tlačítka, viz obr. 1.3.

Obr. 1.3: Akční tlačítka programu IT Guru

 **Objects Palette**: Paleta, kde si můžeme vybrat objekty pro náš projekt, pracovní stanice, servery, linky, síťové prvky, atd.

 **Check links**: Toto tlačítko kontroluje, jsou-li všechny linky v síti nakonfigurovány správně. Většinou se používá před samotnou simulací.

 **Link Failure**: Tlačítko pro shození linky nebo určitého uzlu.

 **Link Recovery**: Tlačítko pro opětovné nahození objektu po předchozím použití Link Failure.

 **Return to Parent Subnet**: Tlačítko používané, když se topologie zvětšuje a některé části jsou situované v subsítích.

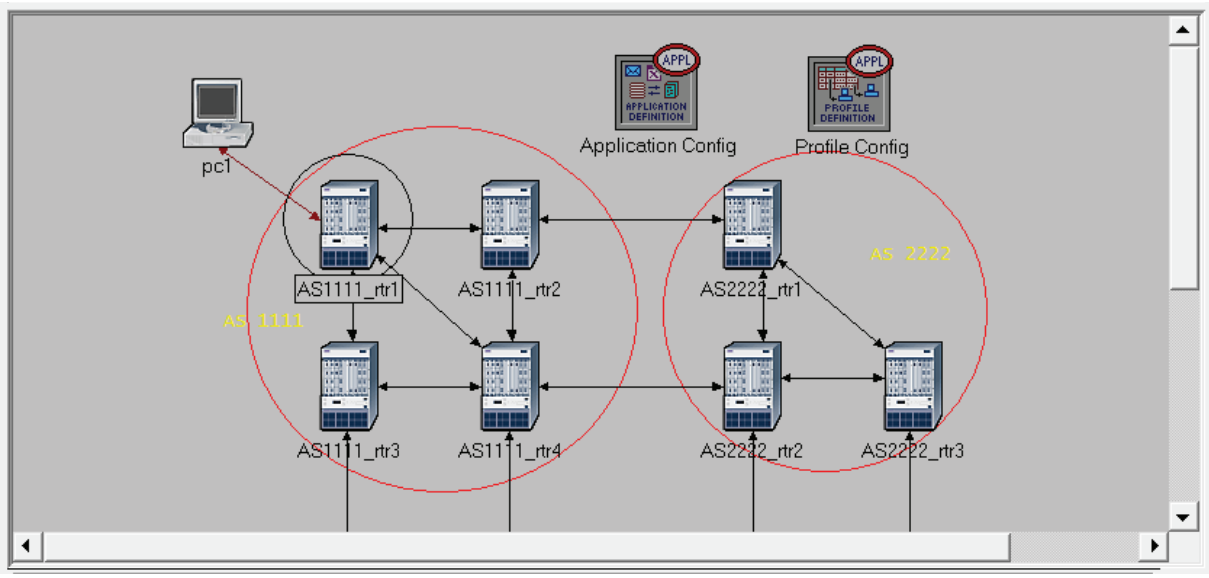
 **Zoom / Unzoom**: Pro přiblížení a oddálení pohledu na pracovní plochu.

 **Run**: Otevře okno, kde konfigurujeme nastavení simulace pro aktuální scénář.

 **View Results**: Po proběhnutí simulace si pomocí tohoto tlačítka zobrazíme výsledky.

 **Hide / Show All Graphs**: Zobrazí nebo skryje všechny grafy, nebo statistiky, které jsme zvolili v Choose Individual Statistics.

Další částí je pracovní plocha, viz obr. 1.4. Toto je místo, kam vkládáme objekty, které budou tvořit síť.



Obr. 1.4: Pracovní plocha programu IT Guru

2 Protokoly

V první laboratorní úloze, která je zaměřená na aplikační protokoly, tak i ve druhé se používají aplikační protokoly. Ty jsou definovány v aplikační vrstvě vrstvového modelu TCP/IP. Tato vrstva zahrnuje mnoho protokolů, které uživatel pracující na počítači a komunikující po síti denně využívá. Jsou to například FTP (File Transfer Protocol), HTTP (HyperText Transfer Protocol), IMAP (Internet Message Access Protocol), SSH (Secure Shell), SIP (Session Initiation Protocol), SMTP (Simple Mail Transfer Protocol), Telnet (Telecomunication Network) a mnoho dalších.

2.1 TELNET

Telnet (Telecomunications Network protocol) je jeden z nejstarších protokolů [5]. První zmínka je z roku 1969, standardizován byl roku 1980 jako norma RFC-764 a o tři roky později byla nahrazena novější normou RFC-854. Telnet emuluje terminál na bázi protokolu TCP/IP. Je to tedy zařízení, pomocí kterého můžeme komunikovat s počítačem.

Podmnožinou Telnetu je NVT (Network Virtual Terminal). Můžeme si představit jako, kdyby se Telnet skládal ze dvou vrstev, tedy spodní vrstvy NVT a vrstvy vrchní, samotného protokolu Telnet. NVT má na starosti reprezentaci dat, například synchronizaci v jaké abecedě budou koncové uzly komunikovat. Tento protokol využívají, kromě Telnetu i FTP, POP3, SMTP a HTTP a další. Telnet umožňuje práci i na vzdáleném počítači. Komunikuje se přes TCP port číslo 23. Pokud uživatel komunikuje se vzdáleným počítačem, označuje se jako server.

Telnet definuje několik řídících funkcí [6]. Například IP (Interrupt Process), AO (Abort Output), AYT (Are You There), EC (Erase Character) a EL (Erase Line). Interupt process neboli přerušení procesu pozastavuje, přerušuje nebo ukončuje provoz uživatelského procesu. Funkce Abort Output jak už naznačuje překlad, zahazuje výstup, aniž by výstup vypsal do terminálu uživatele. Are You There poskytuje uživateli důkaz, že proces stále běží. Může být vyvolána uživatelem, když je systém neočekávaně nečinný na dlouhou dobu. Erase Character maže poslední znak ve streamu dat pocházejícího od uživatele. Erase Line maže všechny data na aktuálním vstupu. A další funkce EOF (End of File), BRK (Break process), WILL, WONT, DO, DONT atd.

Telnet definuje 4 komunikační režimy [5]. Poloduplexní, znakový, pseudořádkový a řádkový režim. Poloduplexní režim (Half-duplex) používá ke komunikaci jednu sdílenou

frekvenci, když tedy jeden vysílá, druhý musí naslouchat. Po ukončení přenosu se posílá příkaz <IAC GA> nebo také v překladu přepínám. Tento režim se dnes už nepoužívá. Znakový režim (Character at a time) je nyní nejvíce používaný režim. Je implicitně nastaven na obou koncích, tedy jak u uživatele, tak i u serveru. Uživatel posílá znak po znaku, který je zabalen do IP záhlaví a záhlaví TCP segmentu. Podle znakového režimu je odvozen režim pseudořádkový (Line at a time). Zatímco znakový režim vyžaduje, aby byli spuštěné dvě volby najednou, ECHO a SUPPRESS-GO-AHEAD. U pseudořádkového režimu je vždy jedna volba neaktivní. Volba echo zabezpečuje, že stisknutá klávesa se zobrazí na terminálu a druhá volba potlačuje poloduplexní režim. Řádkový režim (Line mode) nejdříve zpracuje celý řádek na straně uživatele a až poté je tento řádek poslán k serveru.

2.2 FTP

FTP (File Transfer Protocol) slouží k přenosu dat přes počítačovou síť pomocí TCP/IP [5]. Podobně jako protokol Telnet je i FTP jeden z nejstarších protokolů. Poprvé byl vyhlášen normou RFC-114 roku 1971, následně prošel úpravami v normách RFC-959, RFC-2228 a RFC-2640.

Koncový uživatel pracuje s FTP pomocí uživatelského rozhraní, které si můžeme představit jako FTP program, internetový prohlížeč nebo jako grafickou utilitu. Toto uživatelské rozhraní komunikuje s vrstvou nazvanou interpret příkazů. Ta obsahuje sadu příkazů a komunikuje se stejnou vrstvou na straně serveru. Dále může pracovat s lokálním souborovým systémem. Pro přenos dat je nutné, aby se obě vrstvy interprety příkazů domluvily na vlastnostech přenosu. Poté dají příkaz k vrstvě přenosu, ta komunikuje s lokálním souborovým systémem a dokáže na jedné straně soubor číst a na straně druhé ho zapisovat. Pro komunikaci mezi serverem a uživatelem se používají dvě linky. Po příkazové lince klient posílá požadavky na server, který poslouchá na portu 21. Po datovém kanálu prochází data, zde se může role serveru a klienta navzájem vyměnit. S tím souvisejí dva režimy komunikace, pasivní a aktivní.

FTP má 4 typy přenášených souborů, implicitní, binární, lokální a EBCDIC (Extended Binary Coded Decimal Interchange Code). Implicitní typ používá NVT protokol pro reprezentaci dat a používá i příkazový řádek. Typ EBCDIC se používá jen tehdy, pokud tento kód znají oba dva koncové uzly. Binární typ vysílá tok bajtů. Poslední lokální typ se v dnešní době nepoužívá, sloužil pro spojení dvou systému, které neměly stejnou

reprezentaci dat. Dále FTP rozlišuje strukturu přenášeného souboru na souborovou, strukturu tvořenou větami a stránkovou strukturu. Rozlišuje i několik módů přenášených dat. Mód ve tvaru datového toku, blokový mód a kompresní mód.

FTP používá celkem 32 příkazů, mezi které patří například: PASS heslo, CWD nastavení aktuálního adresáře, QUIT ukončení spojení, PASV přechod do pasivního režimu, RETR stažení souboru ze serveru a další.

2.2.1 Aktivní režim

Aktivní režim je klasickým módem komunikace FTP protokolu. Tento režim většinou podporují softwaroví klienti, komunikace probíhá v příkazovém řádku. Naopak většina prohlížečů podporuje pasivní režim. FTP server by měl podporovat režimy oba.

Aktivní režim je charakteristický tím, že zřizuje aktivní datový kanál pro přenos informací. Prvním krokem pro zahájení komunikace je žádost klienta o přidělení volného portu, musí být větší než 1023. Přes tento port se spojí s TCP portem 21 na straně serveru. Dalším je přenesení IP adresy a klient se serverem naváže spojení. Všechny údaje se přenášejí v desítkové soustavě. Nyní je možné pracovat se soubory.

2.2.2 Pasivní režim

V některých případech může být aktivní datový kanál mezi serverem a klientem na obtíž, to řeší pasivní režim. Pro pasivní režim nastavuje klient dva kanály, jeden pro příkazový a druhý pro datový. Toho se může využít třeba na ochranu proti filtraci přenosu na směrovači. Prvním krokem pro připojení je dotaz klienta na server o alokaci portu. Ten mu odpoví zprávou, ve které je číslo alokovaného portu a IP adresa serveru, na kterém si klient vytvoří datový kanál.

2.3 HTTP

Protokol HTTP (HyperText Transfer Protocol) je oproti předešlému Telnetu a FTP mladší [5]. Základem byl návrh vědců z organizace CERN (European Organization of Nuclear Research) roku 1990. První verzí byla 0.9, v současnosti se používá verze 1.1 z roku 1999, který byl vyhlášen v normě RFC-2616.

Pomocí protokolu HTTP se přenáší hypertexty mezi entitami sítě. Jako hlavní typ komunikace je komunikace klient-server. Klient napíše do internetového prohlížeče URI

(Uniform Resource Identifier), jednotný identifikátor zdroje. Klient vezme jméno serveru, pošle ho DNS serveru, který mu vrátí přeložené jméno jako IP adresu serveru. Následně klient vymění jméno za IP adresu a naváže spojení s požadovaným serverem. Vše za pomocí TCP protokolu. Dále klient pošle HTTP request a server mu odpoví. Tím se klientovi v prohlížeči zobrazí webová stránka. Každá webová stránka obsahuje určitý počet objektů a na stažení všech objektů je potřeba stejný počet HTTP žádostí. V předchozí verzi bylo nutné pro každý objekt vytvořit nové TCP spojení, ve verzi 1.1 se navazuje spojení pouze jednou. Klient může poslat více žádostí najednou, nemusí čekat, až mu server odpoví na první request. Komunikace mezi klientem a serverem se dělí jen na žádosti a odpovědi. Jistým omezením může být asynchronní přenos HTTP protokolu, kdy nejde vytvořit odpověď bez žádosti.

Mezi entity protokolu HTTP patří proxy, brána a tunel. Všechny tyto systémy mohou a nemusejí ležet mezi klientem a serverem a spojení se navazuje vždy mezi sousedními uzly.

2.3.1 Proxy

Systém proxy se dělí na dvě části, klientskou a serverovou část. Klientská část přebírá žádosti od serverové části a navazuje spojení s cílovým serverem. Naopak serverová část přebírá žádosti od klienta a předává je klientské části. Proxy se většinou vkládá například jako hranice mezi vnitřní sítí a internetem. Proxy má možnost měnit i přepisovat HTTP žádost. Odpověď si ukládá do paměti cash, to by později při stejném dotazu znamenalo rychlejší odpověď. Proxy může zjišťovat, jestli má klient právo na zobrazení stránky a oprávnění k určité žádosti. Také jestli má oprávnění přistupovat na určité servery. Pomocí proxy je možné zakazovat přístup na některé servery, může jít třeba o servery s erotickou tématikou, nebo omezení zaměstnavatele, aby si zaměstnanec neprohlížel servery, které nepotřebuje k práci. Dále může zjišťovat, jestli má vůbec klient oprávnění proxy používat. Proxy také rozlišuje, jestliže jde žádost z vnitřní sítě či ze strany internetu. Z vnitřní sítě může mít klient veškerá oprávnění a ze strany internetu může požadovat například heslo, to slouží jako ochrana proti útokům. Dále může přenášená data kontrolovat, zda neobsahují viry.

2.3.2 HTTP dotaz

HTTP request je svou strukturou podobný elektronické poště. Skládá se ze 4 částí. První část je název metody. Druhá část je záhlaví, to je tvořeno hlavičkami. Každá hlavička začíná klíčovým slovem, následuje dvojtečka, parametry hlavičky a konec řádku. Třetí část je prázdný řádek, který se definuje příkazem CRLF. Poslední částí jsou data.

HTTP definuje 8 metod, GET, POST, OPTIONS, HEAD, CONNECT, PUT, TRACE a DELETE. Metoda GET slouží pro žádost klienta na data uložená na serveru. Metodou POST se naopak odesílají data na server. Metoda TRACE je podobná příkazu tracerout, touto metodou zjistíme kolik bran nebo proxy leží mezi klientem a serverem. OPTIONS zjišťuje informace o vlastnostech serveru.

2.3.3 HTTP odpověď

HTTP response se skládá ze tří částí. První je verze protokolu. Druhou částí je výsledkový kód, který určuje výsledek operace. Výsledkové kódy jsou trojciferné, je jich celkem 5. První začínající jedničkou (1xx) informují o zpracování, které dále pokračuje. Druhé začíná dvojkou (2xx) a dokládá nám, že operace se zdařila. Třetí (3xx) informuje o přesměrování, které se bude týkat jiného jednoznačného identifikátoru URI. Další začínající čtyřkou (4xx) informuje o chybě klienta a poslední začínající pětkou (5xx) informuje naopak o chybě serveru.

2.4 VoIP

VoIP (Voice over Internet Protocol) v překladu přenos hlasu po IP sítích [7]. Přenos hlasu potřebuje podobné nebo stejné chování IP sítě jako sítě telefonní. Proto je potřeba při komunikaci využít signalizace, která umožňuje vytvořit určitou kvalitu linky mezi koncovými spoji a umožňuje ověřit danou úroveň kvality služeb QoS (Quality of Service). Provoz hlasu je obecně také citlivý na zpoždění a jitter. Pokud se tak nestane a přenos hlasu nebude prioritou, koncové uzly neboli uživatelé to pocítí zhoršenou kvalitou.

Hlavní požadavky na VoIP jsou trvalá dostupnost služby, zpoždění v jednom směru 150 ms, ochrana proti odposlechu přenosu a zaručeně vysoká kvalita a odezva v reálném čase. Přenos hlasu se odlišuje od přenosu dat. Hlasová komunikace nepotřebuje takovou šířku pásma jako přenos dat, ale vyžaduje šířku konstantní. „Maximální jednosměrná

latence by se měla pohybovat pod 180 ms a maximální kolísání zpoždění pod 30 ms [7].“ Maximální ztrátovost paketů by měla být okolo 1%.

Pro potřebu přenosu je nutné nejdříve signál upravit, tedy analogový signál digitalizovat, použít A/D převodník. Postupně se provedou tři kroky: vzorkovaní, kvantování a kódování. Vzorkováním se vytvářejí vzorky, provádí se například kmitočtem 8 kHz. Přičemž platí, podle vzorkovacího teorému, že kmitočet vzorkovacího signálu musí být minimálně dvakrát větší, než je maximální kmitočet signálu vzorkovaného. Dalším krokem je kvantování, kdy je každý vzorek přiřazen do určité kvantizační úrovně, například použitím 8 bitového převodníku získáme 256 kvantizačních úrovní. Tím, že hodnota vzorku není pokaždé stejná jako nejbližší hodnota kvantizační úrovně, které je přiřazena, vzniká kvantizační šum. Posledním krokem je kódování. Každé úrovni je přiřazena binární hodnota čísel. U kódování hlasu se například používají kodeky řady G.7xx, nejčastěji G.711, G.723.1 a G.729.

2.4.1 Signalizace

Signalizace je nejdůležitější mechanizmus pro přenos hlasu. Jeho úloha je navazování a ukončování spojení, popis relací a přenos dat pro zabezpečení. Umožňuje i další funkce jako například služby ústředen nebo roaming a další.

Pro VoIP existují dvě rozdílné řešení signalizace, H.323 a SIP. Obě dvě používají stejné kodeky a transportní protokoly.

2.4.2 SIP

SIP (Session Initation Protocol) je jednoduchý protokol pro ovládání interaktivních relací nejen pro VoIP. Provádí lokalizaci, navazuje spojení, poskytuje služby dostupnosti uživatele a uživatelské možnosti. Naproti tomu neposkytuje management interaktivních relací, nedokáže zajistit QoS, ale spolupracuje s protokoly, které ji poskytují.

Oproti H.323 má SIP menší režii. K navázání spojení mu vystačí tři zprávy. Prakticky zjistí, kde se koncový uživatel nachází, jaké vlastnosti má jeho zařízení a o další funkce žádá jiné protokoly. SIP používá mnohem jednodušší textový formát. Struktura SIP se skládá ze serverů a uživatelských agentů (UA, User Agent). Servery předávají požadavky na ostatní servery a klienty.

Server by měl být připraven k přijímání žádosti o libovolné IP adrese, portu nebo transportní kombinaci [8]. Naslouchá na defaultních SIP portech, port 5060 pro TCP

a UDP a 5061 pro TLS přes TCP. Pro každý port a rozhraní, kde server naslouchá na UDP, musí zároveň naslouchat i pro TCP, protože zpráva se může poslat i pomocí protokolu TCP, například pokud by byla dlouhá. Zasílá odpověď (response) na zdrojovou adresu, ze které přišel požadavek (request). UA reprezentuje koncový systém. Obsahuje UAC (User Agent Client), který generuje požadavky a UAS (User Agent Server), který na ně odpovídá. UAC tedy například kliknutím vytvoří požadavek a UAS příjme žádost a vygeneruje odpověď na základě uživatelského vstupu. Komunikace mezi UAS a UAC závisí v základě na dvou faktorech, zda je odpověď uvnitř nebo mimo dialog a na kterou metodu žádosti je metoda založena.

2.4.3 H.323

Pojem H.323 nepopisuje jeden protokol, ale soubor protokolů [5]. H. 323 pro svou práci používá doporučení H.320 a Q.931. Používá se k přeměně signalizace mezi paketovým přenosem a přenosem v telefonní síti. H.323 se stará o navazování a ukončování spojení, digitalizaci a paketizaci. Dále kóduje a komprimuje hlas, pomocí kodérů řady G.7xx a signalizaci volání RAS (Registration, Admission, Status) pomocí protokolu H.255, který zabezpečuje komunikaci mezi koncovými uzly sítě a navazuje spojení mezi uzly. Pomocí H.245 se provádí řídící signalizace, která je zodpovědná za řízení provozu mezi uzly, výměnu informací o technických vlastnostech koncových uzlů a další. Obě signalizace, tedy H.255 a H.245, pracují nad TCP. H.255 zabezpečuje spojení mezi koncovými uzly nebo mezi koncovým uzlem a zařízením gatekeeper pomocí výměny zpráv po signalizačním kanálu. H.245 komunikuje mezi koncovými uzly a vyměňuje zprávy o jejich vlastnostech.

H.323 definuje celkem 4 entity, terminál, gatekeeper, MCU a bránu. Terminál si můžeme představit jako PC nebo jiné speciální zařízení. Terminál musí podporovat protokoly H.245 a H.225, je povinný pro komunikaci pomocí H.323. Dále musí podporovat protokoly RTP nebo RTCP a předem určený kodér pro kódování hlasu (G.7xx), nejčastěji se používá kodér G.711, který používá PCM a rychlost 64 kbit/s. Gatekeeper neboli vrátný je nepovinná složka, která se nepodílí na přenosu hlasu mezi koncovými uzly. Má na starosti řídící služby pro terminály a brány, určuje adresaci, šířku pásma, signalizaci a směrování v síti. Brána je zodpovědná za komunikaci mezi terminály dvou různých sítí, tedy sítě H.323 a jinými. MCU zabezpečuje provoz a komunikaci mezi více terminály a bránami.

Velkou nevýhodou je závislost H.323 na protokolu TCP, což je příčinou velké režie přenosu. Další nevýhodou je, že se nezabývá zabezpečením potřebné šířky pásma pro přenos, proto používá protokol RSVP (Resource Reservation Protocol).

Tab. 2.1: Rozdíly mezi SIP a H.323 [5]

	SIP	**H.323**
Specifikace	IETF	ITU-T
Použitý model	Internet/WWW (klient-server)	Telefonie (peer-to-peer)
Složitost	Střední	Vysoká
Kódování	Textové	Binární
Prvky	UA, UAS	Terminál, brána, gatekeeper
Transportní protokol	Hlavně UDP	Hlavně TCP

3 Směrovací protokoly

Můžeme říci, že směrování je jeden z nejdůležitějších úkonů při přenosu dat v rozsáhlé síti mezi koncovými uživateli. Nebýt toho, data by se nejspíš ztratila. Pro směrování se používají směrovací protokoly, jako RIP, OSPF, IS-IS, BGP a další. Tyto protokoly můžeme rozdělit podle několika kritérií a to podle působení uvnitř autonomního systému nebo vně, tedy na IGP (Interior Gateway Protocol) nebo EGP (Exterior Gateway Protocol). Další rozdělení můžeme provést podle způsobu učení sítě na protokoly s úplnou znalostí sítě (Link state), kam patří OSPF a IS-IS. Nebo na protokoly se zprostředkovatelnou znalostí sítě neboli Distance vector, např. RIP nebo BGP.

3.1 Protokoly Distance Vector

Nazývané též algoritmy vektorů vzdáleností [5]. Poprvé byl tento princip použit v síti ARPANET v roce 1969. Směrovače mají po připojení do sítě informace pouze o přímých sousedech, tedy s metrikou 0. Cestu k ostatním sítím si každý směrovač vypočítává sám, pouze z informací, které dostane od svých sousedů, nikoli na základě znalosti celé sítě. Zprávy od sousedů obsahují seznamy dvojic, *V* a *D*. Parametr *D* znamená vzdálenost a *V* identifikuje cíl, proto název Distance Vector. Směrovací informace se vyměňují periodicky (30 - 90 s). Jelikož mají znalost jen o sousedních směrovačích, jsou tyto protokoly náchylné ke vzniku smyček. Mají též velmi pomalou konvergenci, vzdálené směrovače se o změně v sítí dozvědí po delší chvíli, v řádech minut. Výhodou je jednoduchost protokolu. Zástupci této skupiny jsou např. BGP a RIP.

3.2 Protokoly Link State

Neboli algoritmy stavu spojů. Byly vytvořeny kvůli nedostatku algoritmů vektorů vzdáleností, konvergenci. Je to doba, která je potřebná k informování všech uzlů o změně v síti. Tyto protokoly mají znalost celé sítě, k tomu se využívá záplavové směrování. Vysílá se pomocí paketu LSP, Link State Packet. Postup je takový, že nejdříve si směrovač zjistí informace o všech uzlech v síti. Následně si vypočítá metriku, nekratší cestu k cíli, s tím že se považuje za kořen stromu. K tomu se používá Dijkstrův algoritmus, jehož výsledkem je směrovací tabulka. Největší výhodou je tedy rychlá konvergence, kdy se

informace o změně v síti se posílají okamžitě po změně. Tyto algoritmy využívají protokoly OSPF, IS-IS.

3.3 Autonomní systémy - AS

Pod pojmem autonomní systém si můžeme představit skupinu směrovačů spadající pod stejnou správu, například velký internetový poskytovatel (ISP, Internet Service Provider) [9]. Řídí se jednotným směrovacím protokolem. Autonomní systémy se označují jednoznačným 16 bitovým číslem.

Při směrování uvnitř AS se používají protokoly IGP, pro spojování AS se používají EGP. Při pohledu z venku je znám externím směrovacím protokolům pouze hraniční směrovač AS (Border Gateway). Stejné je to z vnitřního pohledu.

AS můžeme rozdělit na dvě skupiny, podle počtu připojení k jiným AS na single-homed a multi-homed autonomní systémy. Single-homed je připojený pouze k jednomu dalšímu AS a to pouze jednou linkou. Autonomní systémy typu multi-homed mohou být připojeny k více AS s více linkami ke každému z nich. AS multi-homed pak mouhou být tranzitními systémy, tedy spojovat AS.

3.4 Protokol IS-IS

Směrovací protokol IS-IS, Intermediate System to Intermediate System, je často porovnáván s protokolem OSPF [10]. V současnosti je IS-IS nejvíce používán poskytovateli internetu, ISP. Je to výborný směrovací protokol pro velké sítě, protože je jednoduchý a stabilní.

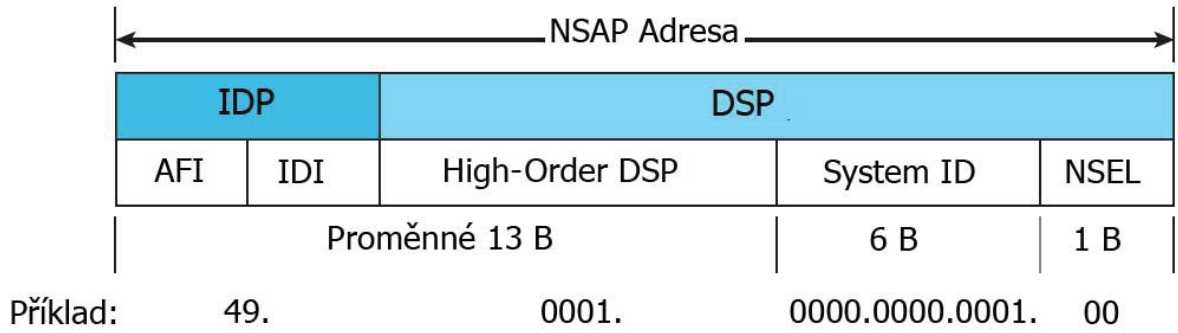
IS-IS je směrovací protokol typu link state. Původně byl navržen pro OSI (Open Systems Interconnection) sítě, později i pro IP. IS-IS rozděluje směrovací doménu na více domén. Pro výměnu směrovacích informací používá LSP a pro výpočet nejlepší cesty k cíly používá Dijkstrův algoritmus. Protokol IS-IS rozděluje v rámci AS směrovací prostor na levely, Level 1 a Level 2. Ve skutečnosti OSI rozděluje prostor na 4 úrovně, Cisco podporuje jen 2.

IS-IS používá 4 druhy paketů, Hello pakety, LSP (Link-state PDU), CSNP (Complete Sequence Number PDU) a PSNP (Partial Sequence Number PDU). Hello pakety se posílají každých 10 s. LSP se používá pro výměnu směrovacích informací. Je číslované od 0x00000001 do 0xFFFFFFFF. Životnost LSP je 1200 s a periodicky se po 15 min

obnovují. Pomocí PSNP je možné potvrdit nebo vyžádat konkrétní LSP. Pomocí CSNP se každých 10 s posílá úplný seznam LSP.

3.4.1 Adresace

Pro adresaci používá IS-IS NSAP (Network Service Access Point), viz obr. 3.1. Délka je proměnlivá, od 8 do 20B.



Obr. 3.1: Formát NSAP adresy [10]

NSAP se dělí na IDP (Initial Domain Part) a DSP (Domain Specific Part). IDP obsahuje AFI (Authority and Format Identifier), který specifikuje orgán, který přiřazuje tuto adresu. Například 49 se používá pro privátní adresy. Dále obsahuje IDI (Initial Domain Identifier), který identifikuje subdoménu za AFI, například 47.0006 byla přidělena americkému ministerstvu obrany. DSP obsahuje informace potřebné ke směrování uvnitř domény. Obsahuje HO-DSP (high-order domain-specific part), the system ID a NSEL. HO-DSP dělí domény na oblasti. The system ID definuje jednotlivé zařízení, je podobné IP adrese. NSEL identifikuje proces na zařízení, pokud je tato hodnota nulová, mluvíme o NSAP jako o NET (Network Entity Title).

Adresu můžeme rozdělit na 3 části, Area Address, System ID a NSEL (Network selector). Area Address musí být 1B dlouhá a skládá se ze dvou částí, AFI a area ID. Někdy se jí říká prefix. System ID, neboli ID oblasti, je maximálně 6B dlouhé. Je to unikátní číslo v síti, většinou se používá MAC adresa zařízení. NSEL většinou obsahuje nulovou hodnotu, což označuje směrovač.

3.4.2 Úrovně směrování

OSI sice definuje 4 úrovně směrování, prakticky se používají jen dvě, Level 1 a Level 2. Směrování Level 1 popisuje směrování ve vnitřní oblasti. Obsahuje informace o topologii

samotné oblasti. Level 2 se používá pro směrování mezi oblastmi. Směrovače si vyměňují adresy oblastí (prefixy) a zjišťuje nejkratší cestu k nim.

3.4.3 Typy směrovačů

IS-IS definuje tři typy směrovačů, které těsně korespondují s úrovněmi směrování. Směrovač Level 1 se naučí cesty jen ve své oblasti. Směrovač Level 2 se naučí jen cesty, které existují mezi oblastmi. Poslední typ směrovače je Level 1-2, říká se mu hraniční směrovač. Leží na hranici oblasti a učí se jak cesty uvnitř oblasti, tak cesty spojující tyto oblasti. Prakticky tedy spojuje Level 1 a Level 2 směrovače.

3.4.4 Porovnání s OSPF

Oba jsou to link state směrovací protokoly, kteří podporují VLSM (Variable-Length Subnet Mask) adresování s maskou podsítě proměnné délky. Používají SPF algoritmus (Shortest Path First algorithm) pro výpočet nejkratší cesty k cíli a oba velice rychle reagují na změny v síti.

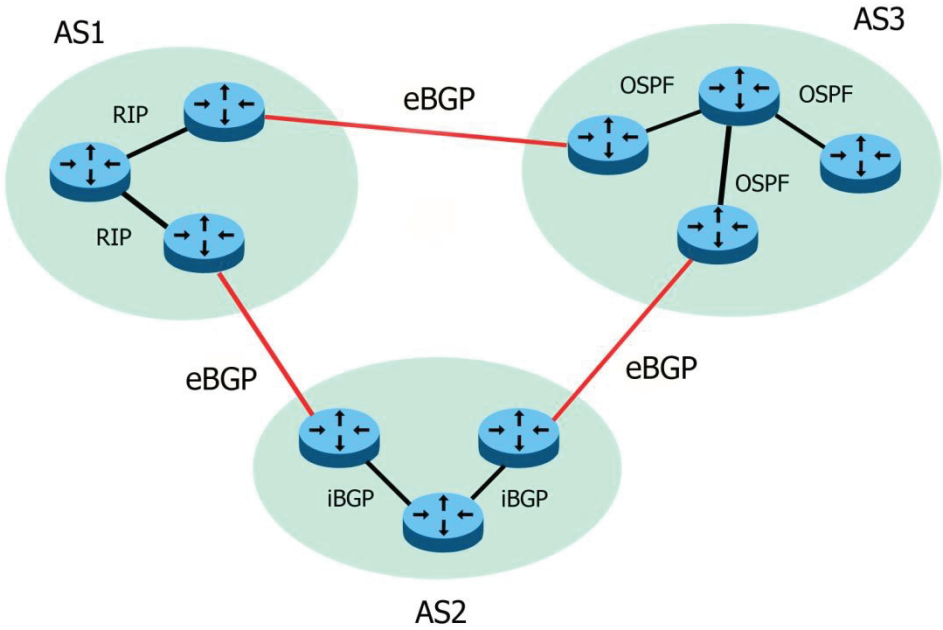
Hranice oblastí pro OSPF jsou tvořeny směrovači ABR, kdežto u IS-IS hranice leží na linkách. Vzhledem k posílání mnoho malých LSA v OSPF se výměna nových informací rozšiřuje pomaleji oproti IS-IS. OSPF pracuje nad IP vrstvou, IS-IS nad vrstvou linkovou. Metrika je u IS-IS konstantní, rovná 10, u OSPF je proměnná. IS-IS je možné rozšíření paketu o nová pole, u OSPF nikoli.

3.5 Směrovací protokol BGP

Border Gateway Protocol, je zástupce vnějších směrovacích protokolů EGP [11]. Pro směrování v síti je velice důležitý, i když se s ním běžný uživatel nesetká. Nejdůležitějším úkolem BGP je flexibilita, snadné spojení a výměna informací mezi AS. Výměna těchto informací je posílána sítí na základě znalosti sousedních směrovačů, tyto údaje musí být nakonfigurovány ručně. Pro přenos informací se používá protokol TCP.

Autonomní systémy tedy využívají BGP protokol k distribuci informací o jeho síti a o sítích, do kterých je schopen doručit pakety přes svůj AS. Okolní AS se z těchto informací rozhodnou, zda použít právě tento AS nebo jiný ke směrování do jiné sítě. Uzlům, které se mezi sebou dorozumívají prostřednictvím protokolu BGP se říká peer a jejich komunikaci peering.

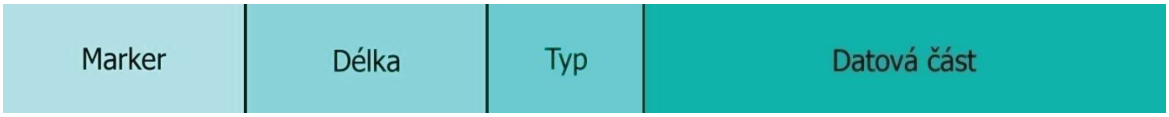
BGP se nemusí používat výhradně jen pro spojování AS. Možné je použití i uvnitř AS, poté se nazývá interior BGP. Protokolu BGP, který směruje data mezi AS, se někdy říká exterior BGP. Na obr. 3.2 je vidět možné použití směrovacích protokolů uvnitř autonomních systémů a mezi sebou nazvájem. V několika případech není vhodné použít iBGP. Například když je síť připojená pouze jednou linkou k ISP, pokud síť nemá dostatečně výkonné směrovače pro zajištění náročného BGP směrování nebo jestliže správci neznají BGP protokol. Na druhou stranu situace, kdy síť obsahuje výkonné směrovače a správci znají BGP, nebo je síť připojená k více ISP, tak je možné iBGP implementovat.



Obr. 3.2: Ukázka možného použití směrovacích protokolů v AS a mezi AS

3.5.1 Formát zprávy

Délka zprávy je proměnlivá, záleží na nepovinné datové části zprávy. Maximální délka je 4096 B. Struktura je na obr. 3.3.



Obr. 3.3: Struktura BGP zprávy [11]

Marker je pole o délce 16 B a zajišťuje kompatibilitu se staršími verzemi. Pole *Délka* udává celkovou délku zprávy a má délku 2 B. V poli *Typ* je uložena hodnota reprezentující typ zprávy a je dlouhé 1 B. BGP definuje celkem 4 zprávy, Open, Update nebo NLRI Update, Notification a KeepAlive. Posledním polem je *Datová část*, která je nepovinná a má maximální délku 4077 B.

3.5.2 Zpráva NLRI Update

Network Layer Reachability Information Update neboli aktualizace dostupnosti síťové vrstvy. Tato zpráva je určena pro přenos aktualizací směrovacích informací. Je přenášena pouze při změně topologie, viz obr 3.4.

Obr. 3.4: Formát zprávy NLRI Update [11]

Délka pole informujícího o stažených (neplatných) routách informuje o délce dalšího pole a je dlouhé 2 B. *Neplatné (stahované) routy* obsahuje seznam neplatných rout a má proměnnou délku. Pole *Celková délka pole atributy routy* udává celkovou délku dalšího pole a je dlouhé 2 B. Pole *Atributy routy* obsahuje atributy, každý se skládá ze tří částí, typ atributu (2 B), délka atributu (1 B) a hodnota atributu. Atribut ORIGIN definuje původce směrovací informace. Atribut AS_PATH definuje cestu k cíli tvořenou seznamem AS. Atribut NEXT_HOP udává unicastovou adresu dalšího skoku v síti. Atribut LOCAL_PREF nastavuje v rámci AS preference pro různé routy směrem z AS. Atribut MULTI_EXIT_DISC je nepovinný atribut, pomocí kterého můžeme ovlivnit provoz do AS. Posledním polem je NLRI, to je pole proměnné délky. Může zde být uložena 1 nebo více NLRI informací, délka a prefix.

3.5.3 Průběh směrovaní

Směrování mezi AS je složitější než uvnitř AS, protože směrovací tabulky obsahují mnohem více záznamů a vyhledávání je tedy složitější. BGP využívá víc parametrů pro určení nejlepší cesty k cíli, například vzdálenost, nastavitelné atributy administrátorem, pravidla zohledněná vůči zdroji a cíli atd. Směrovač tedy není jediný, který ovlivňuje směrování. Administrátor může aplikací atributů efektivně měnit konfiguraci svého AS.

Směrovač obsahuje celkem 3 směrovací tabulky, Loc-RIB, Adj-RIB-In a Adj-RIB-Out. Loc-RIB je vlastní směrovací tabulka směrovače. Adj-RIB-In je tabulka, která obsahuje NLRI informace obdržené od připojených sousedů. V tabulce Adj-RIB-Out se udržují NLRI informace pro odeslání k těmto sousedům.

Pokud v Adj-RIB-In přibude nová cesta, je na BGP procesu, jestli ji zařadí do Loc-RIB nebo její záznam aktualizuje. To určuje pomocí povinných nebo nakonfigurovaných parametrů. Atribut AS_PATH je seznam autonomních systémů, přes které musí přenos přejít, aby se dostal do cíle. Proces vybírá vždy tu s menším počtem záznamů. Dalším atributem je váha (weight). Je to lokální parametr, kterým upřednostňujeme některé linky před ostatními, tím že jím přiřazujeme váhu spoje. Pomocí atributu LOC_PREF neboli lokální preference upravujeme přenos z našeho AS. Používá se tehdy, pokud existuje více cest do stejného AS. Pomocí atributu MED ovlivňujeme provoz do našeho AS, tím že prakticky znevýhodníme metriku jedné linky oproti druhé.

4 Laboratorní úlohy

V rámci diplomové práce jsou vytvořeny 4 podrobné návody pro počítačová cvičení předmětu Komunikační technologie. Laboratorní úlohy jsou vytvořeny v programu Opnet IT Guru. Návod k úlohám je co nejvíce detailní, student by při jeho dodržování měl dosáhnout úspěšného zvládnutí úlohy. Také by měl být schopen vysvětlit vyučujícímu podstatu nabytých informací. Každý krok a každé jednotlivé nastavení je vysvětleno a doplněno obrázkem. Úlohy jsou logicky dělené do několika na sebe navazujících částí, úvod, výstavba sítě, nastavení parametrů pro simulaci provozu, nakonfigurování příslušných síťových prvků, simulaci provozu a následné vyhodnocení výsledků.

V návodech pro úlohy jsou použity grafické výstupy přímo z IT Guru, které nejsou typograficky kvalitní. Důvodem je názorná ukázka pro studenty, díky kterým si mohou kontrolovat správnost nastavení parametrů a simulačních výstupů programu.

První úloha se zaměřuje na srovnání aplikačních protokolů, HTTP, FTP a Telnet. Každá aplikace je vložena do jednoho profilu. Ty se spouštějí po sobě od prvního do posledního. Student si pak z grafů může vyčíst rozdíly.

Druhá úloha je zaměřená na protokol VoIP, kde je opět vytvořeno několik profilů. Obsahují stejnou aplikaci pro přenos hlasu, ale každá je pozměněná použitím jiných kodeků. V druhé části je do profilů vložena signalizace SIP a je pozměněna síť, kde je přidán proxy server pro podporu protokolu SIP.

Třetí úloha se zaměřuje na směrovací protokol IS-IS. Před samotnou úlohou je jednoduchý úvod, který studentům přiblíží směrování v sítí. V první části si student otevře připravenou topologii a nakonfiguruje si IP adresaci jednoho uzlu. Toto řešení jsem volil kvůli časové náročnosti. Následně se síť překonfiguruje na směrování pomocí IS-IS protokolu. Dalším bodem je vytvoření dvou skupin podle parametru NET (Network Entity Title) a s tím spojené rozdělení směrovačů podle úrovně. Další částí je sloučení vytvořených oblastí. Na konci laboratorní úlohy je vytvořený samostatný úkol.

Poslední čtvrtá úloha seznamuje studenty se směrovacím protokolem BGP. Stejným způsobem jako pro třetí úlohu i zde jsem připravil síť. Postupně student nastaví IP adresaci a nastavení BGP parametrů pro jeden uzel z důvodů časové náročnosti nastavování parametrů. Vytvořená síť je rozdělena na 4 autonomní systémy. V dalších bodech si student postupně vyzkouší nastavování atributů, díky kterým usměrňuje cestu přenosu v síti. Na konci úlohy je opět samostatný úkol.

5 Aplikační protokoly

5.1 Úvod k laboratorní úloze

Tato úloha se zabývá srovnáním protokolů pracujících na aplikační vrstvě komunikačního modelu. FTP (File Transfer Protocol) je jeden z nejstarších protokolů. Používá se pro přenos dat přes komunikační síť. HTTP (Hypertext Transfer Protocol) se používá pro výměnu hypertextových dokumentů. Spolu s elektronickou poštou se jedná o nejvyužívanější protokol. Dalším protokolem je Telnet (Telecomunication Network), který určený pro připojení uživatele ke vzdálenému počítači. A v posledním případě se setkáme v laboratorní úloze s elektronickou poštou neboli Email. Jako komunikační prostředí je zvolen ethernet.

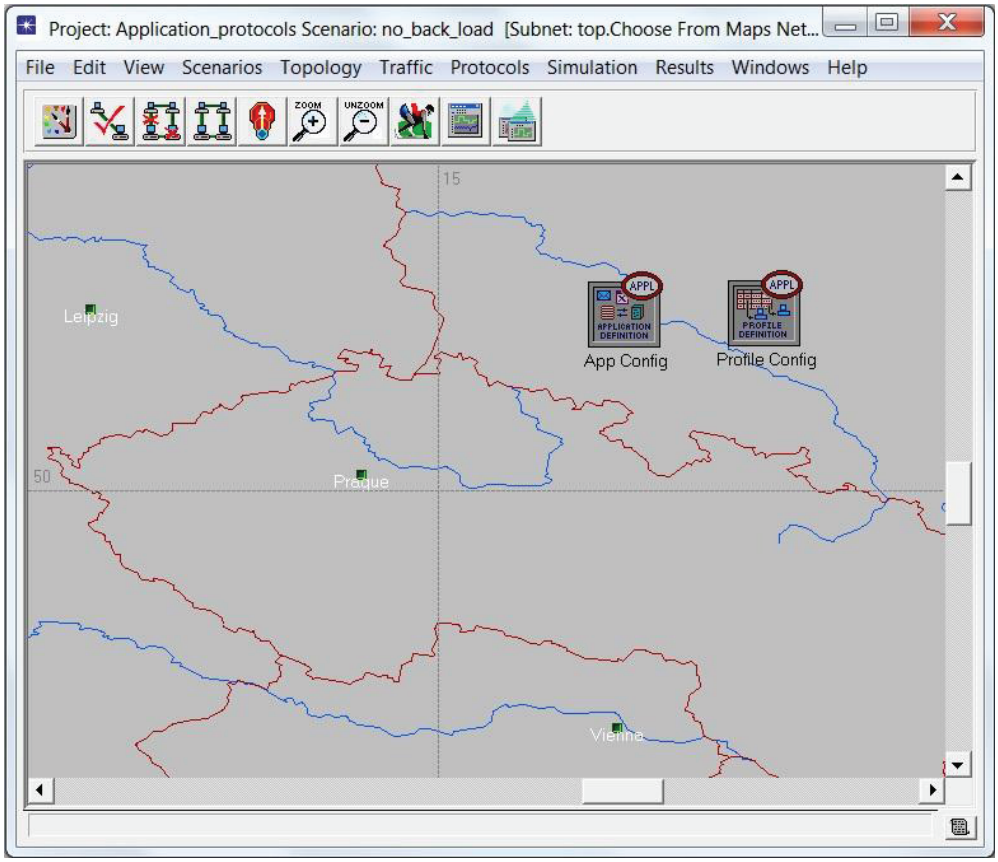
5.2 Vytvoření topologie

1. Spustíme program OPNET IT GURU academic edition.
2. Z nabídky zvolíme **File/new...** (Ctrl+N), v okně ponecháme položku *„Project“* a potvrdíme tlačítkem **OK**.
3. V dalším okně zadáme název projektu například: **Application_protocols** a název scénáře: **no_back_load** a klikněte na **Next**.
4. Pro vytvoření prázdného scénáře vybereme *„Create Empty Scenario“* a potvrdíme **Next**.
5. V další nabídce vybereme *„Choose From Maps“* a potvrdíme. V rolovací nabídce vyhledáme *„Europe“* a potvrdíme.
6. V další záložce vybereme předpřipravené soubory, které budeme používat: *„ethernet, LANs, links, routers“* a potvrdíme. V posledním okně klikneme na OK. Tím se vytvoří projekt.
7. Objeví se mapa Evropy a okno **Object palette**, na hlavní liště ikona  . Tlačítkem **ZOOM** opět na hlavní liště se přiblížíme na Českou Republiku.
8. Z okna Object Palette vybereme a přeneseme na plochu projektu dva objekty: *„Application Config“*, který se používá pro výběr a nastavení aplikací a *„Profile Config“*, kde se nastavují parametry těchto aplikací.
9. Nyní budeme upravovat blok Application Config. Zde můžeme vybírat z již hotových aplikací nebo si vlastní vytvářet. Pravým tlačítkem tedy klikneme na blok a zvolíme *„Edit Attributes“,* kde nastavujeme vlastnosti objektů. V položce *„name“*

přejmenujeme blok na **App Config**. Dále v řádku *„Application Definitions“* vybereme atribut *„None“* a klikneme na **Default**. Tím jsme vybrali všechny vytvořené podporované aplikace. Jsou to tyto:

- Database Accsess (Heavy), Database Accsess (Light)
- eMail (Heavy), eMail (Light)
- File Transfer (Heavy), File Transfer (Light)
- File Print (Heavy), File Print (Light)
- Telnet Session (Heavy), Telnet Session (Light)
- Video Conferencing (Heavy), Video Conferencing (Light)
- Voice over IP Call (PCM Quality), Voice over IP Call (GSM Quality)
- Web Browsing (Heavy HTTP 1.1), Web Browsing (Light HTTP 1.1)

10. Nyní klikneme na Profile Config pravým tlačítkem a podobně jako u Application config ho přejmenujeme na **Profile Config**, viz obr. 5.1.



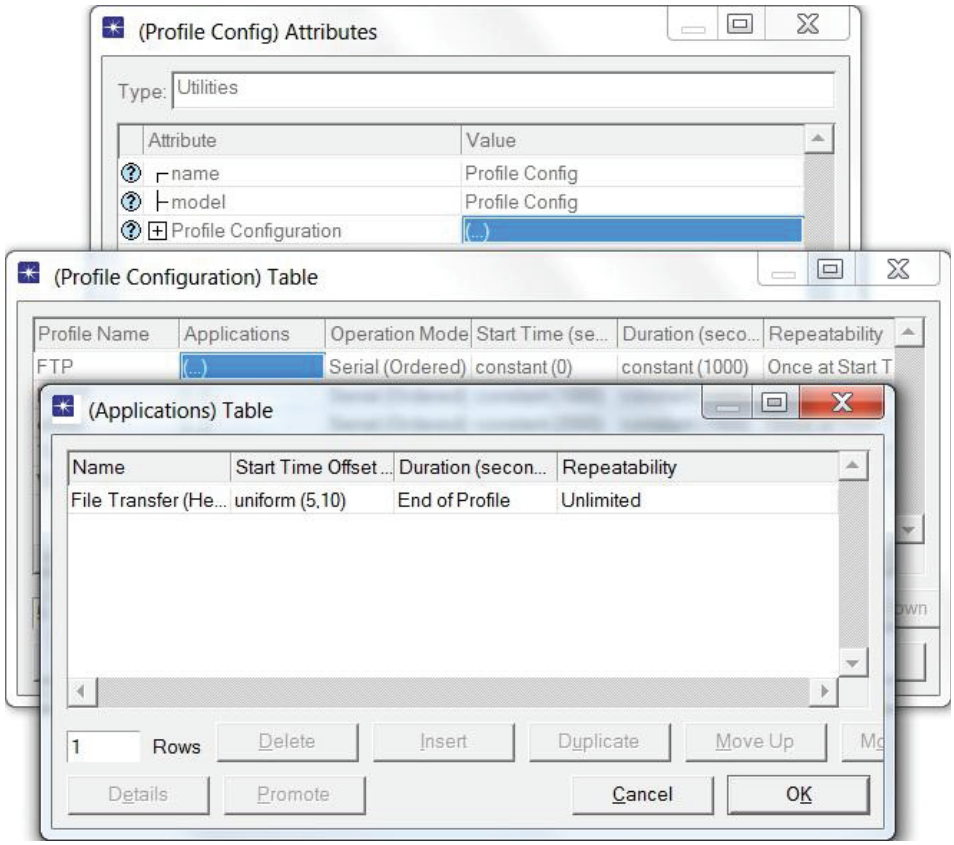
Obr. 5.1: Ukázka Application Config a Profile Config

11. V tomto bloku ještě zůstaneme. Klikneme v řádku Profile Configuration na položku None a zvolíme **Edit**. Otevřelo se okno *„Profile Configuration Table“*. Zde můžete vytvářet profily. Každý profil může obsahovat několik aplikací.

12. Nyní si nastavíme jeden profil. V nabídce *„Rows“* změníme hodnotu na **1**. Další hodnoty zadáme takto:

    - Profile Name: FTP
    - Operation Mode: Serial (Ordered) - profily se budou spouštět jeden za druhým od horního až k poslednímu
    - Start Time (seconds) constant (0) - profil bude startovat v 0s
    - Duration constant (1000) - doba trvání profilu bude 1000s

13. Dále nastavíme aplikaci, která poběží v našem profilu. V jednom profilu můžeme spouštět více aplikací. Klikneme ve sloupci *„Applications“* na **None** a z rolovací nabídky vybereme **edit**. Objeví se okno *„Applications Table“*. Zde nastavíme hodnotu *„Rows“* na **1**. Klikneme na položku **Name** a vybereme aplikaci. V našem případě to bude *„File Transfer (Heavy)“*, viz obr. 5.2.



Obr. 5.2: Nastavení FTP profilu

14. Nyní si vytvoříme v okně Profile Configuration Table další tři profily: HTTP, eMail, Telnet a Všechny. Vytvoříme je podobně jako FTP.

HTTP :

- Profile Name: HTTP
- Operation Mode: Serial (Ordered)
- Start Time (seconds) constant (1000)
- Duration constant (1000)

EMail:

- Profile Name: eMail
- Operation Mode: Serial (Ordered)
- Start Time (seconds) constant (2000)
- Duration constant (1000)

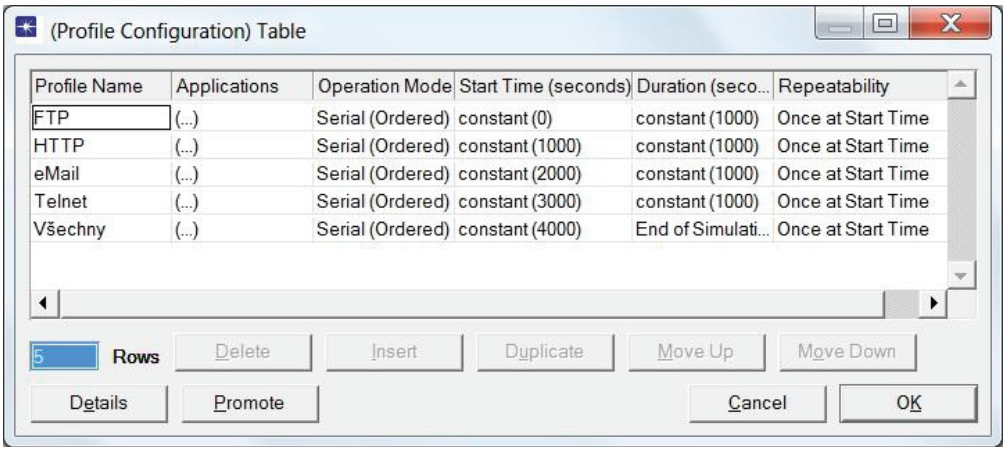
Telnet:

- Profile Name: Telnet
- Operation Mode: Serial (Ordered)
- Start Time (seconds) constant (3000)
- Duration constant (1000)

Všechny:

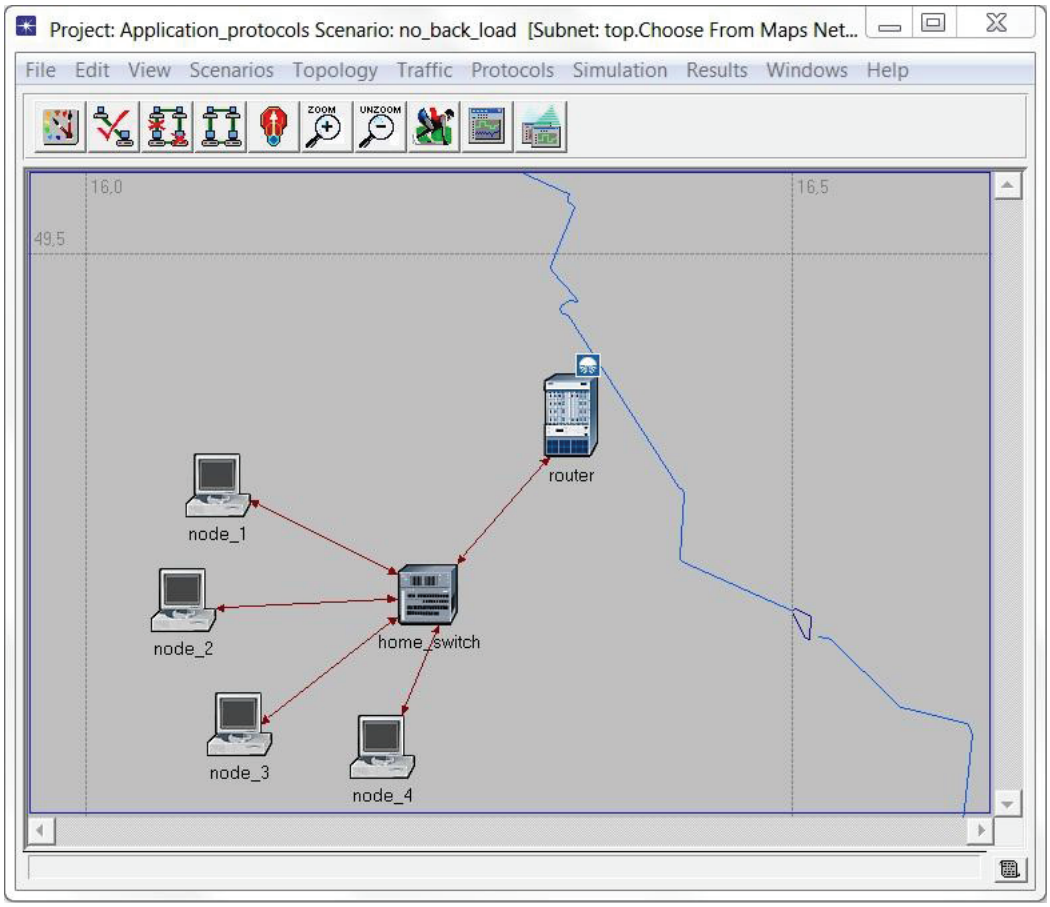
- Profile Name: Všechny
- Operation Mode: Serial (Ordered)
- Start Time (seconds) constant (4000)
- Duration constant (1000)

V záložce Applications každého profilu vložíme příslušnou aplikaci. Do profilu Všechny pak vložíme všechny aplikace, které budeme používat, viz obr. 5.3.



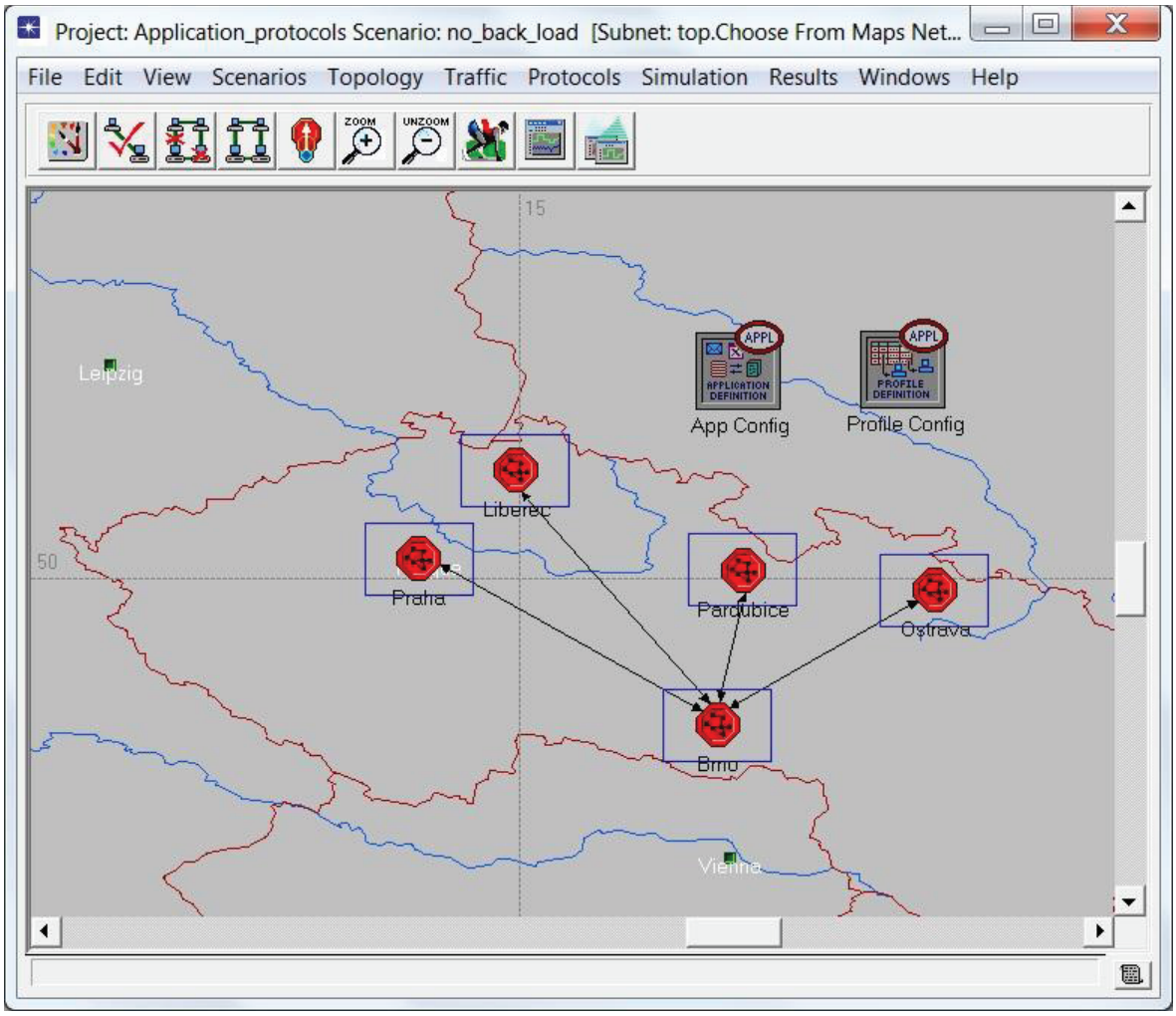
Obr. 5.3: Vytvořené profily v objektu Profile config

15. Nyní jsme vytvořili profily a dále vystavíme síť. Otevřeme si Object palette a na pracovní plochu vložíme blok **subnet**. Dvojklikem na blok se dostaneme do podsítě. Z palety nástrojů vložíme bloky a pojmenujeme je: *„ethernet16_switch“*- home_switch, *„BN_ASN_4s_e4_f2_fe4_sl12_tr4“*- router a 4x *„ethernet_wkstn“*. Nakonec pospojujeme technologií *„10BaseT“*, viz obr. 5.4.

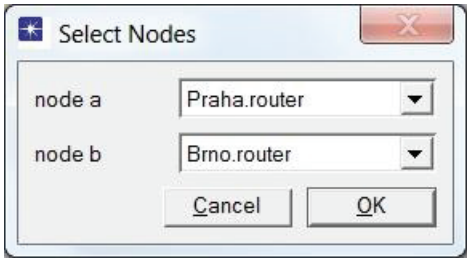


Obr. 5.4: Objekty pro vystavění podsítě

16. Vrátíme se o úroveň výše ikonou na hlavní liště. Naši podsíť si pomocí klávesových zkratek CTRL+C, CTRL+V nakopírujeme. Jednotlivé podsítě pojmenujeme (Brno, Praha, Liberec, Pardubice a Ostrava), viz obr. 5.5. Podsíť Brno spojíme s ostatními technologií *„PPP_DS0“*, kterou najdeme opět v Object Pallette. Po propojení vyskočí okno *„Select Nodes“*, kde nadefinujeme, které uzly obou podsítí chceme spojit. Z rolovacích lišt vybereme na obou stranách **router**, viz obr. 5.6.

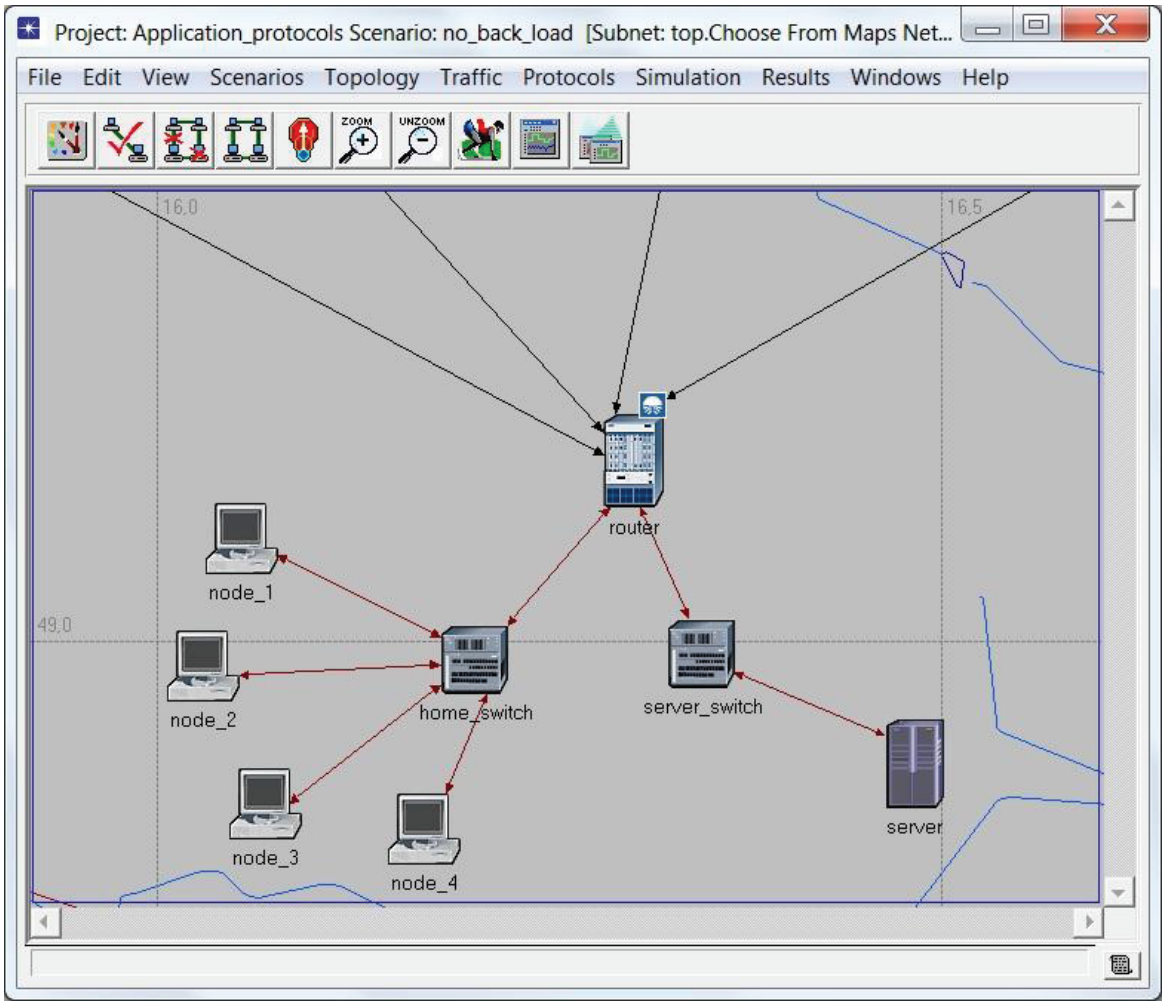


Obr. 5.5: Síť tvořená čtyřmi podsítěmi



Obr. 5.6: Spojení uzlů podsítí Praha a Brno

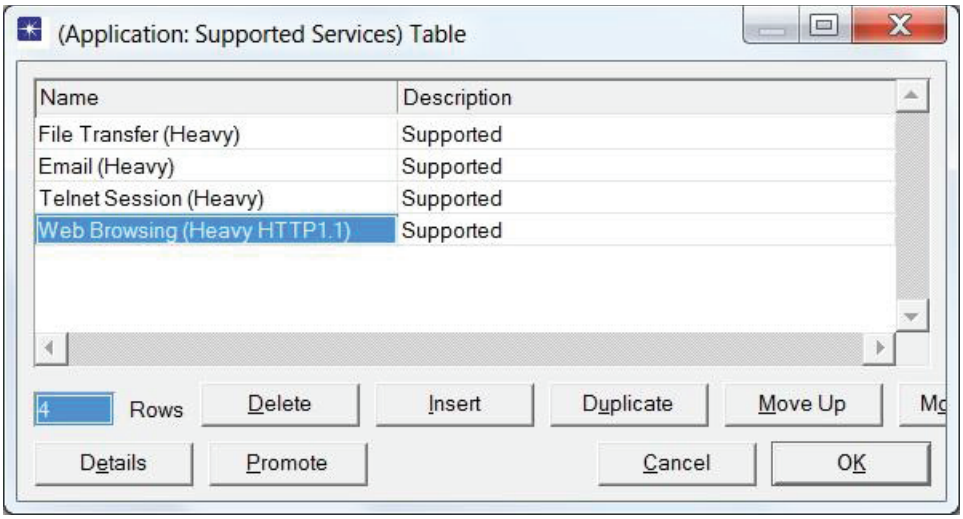
17. Nyní nám zbývá ještě vytvořit server, který nám bude distribuovat aplikace. Otevřeme proto podsíť Brno. Z Object Palette si do okna vložíme a pojmenujeme *„ethernet_server“* jako **server** a *„ethernet16_switch“* jako **server_switch**. Nakonec propojíme server, switch a router technologií *„10BaseT“*, viz obr. 5.7.



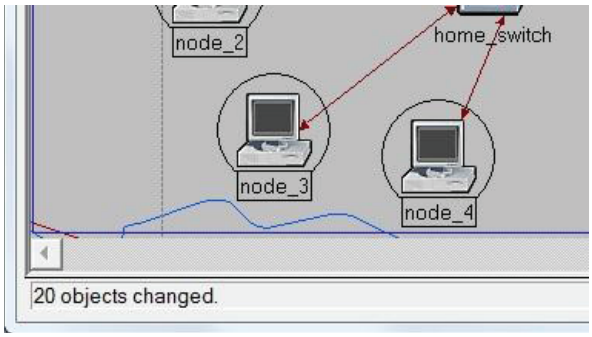
Obr. 5.7: Podsíť Brno

18. Nyní definujeme provoz na serveru. Zůstaneme v podsíti Brno a otevřeme si atributy serveru, Edit Attributes. V řádku *„Applicaton: Supported Services“* klikneme na **None** a zvolíme **Edit**. Objeví se okno *„Application: Supported Services Table“*, kde změníme hodnotu *„Rows“* na **4**. Ve sloupci **Name** znovu klikneme na **None** a vybereme naše aplikace, tedy: File Transfer (Heavy), Email (Heavy), Telnet Session (Heavy) a Web Browsing (Heavy HTTP 1.1). Ve sloupci *„Description“* ponecháme **Supported**, viz obr. 5.8. Tím jsme právě nastavili server, aby podporoval aplikace, které požadujeme.
19. Dále nastavíme podporované profily na koncových uzlech. Pravým tlačítkem klikneme na pracovní stanici a vybereme *„Select Similar Nodes“*. Tím se označí všechny stejné

objekty ve scénáři. Dále si zobrazíme jejich atributy. V řádku „Application: Supported Profiles“ změníme hodnotu na **Edit**. Zde změníme *„Rows“* na **5**. Postupně přidáme všechny nadefinované profily. Potvrdíme **Ok**, viz obr. 5.9. Než potvrdíme okno atributů koncových stanic, nesmíme zapomenout na zatrhnutí tlačítka **Apply Changes to Selected Objects**. To nám zařídí, že profily, které jsme přiřadili jedné koncové stanici, se přidají všem stanicím. Jako kontrola by nám program měl napsat **20 objects changed** v levém dolním rohu okna projektu po odkliknutí tlačítka **Ok**, viz obr. 5.10. Těmito kroky jsme nastavili na pracovních stanicích a serveru podporu aplikací.



Obr. 5.9: Aplikace, které podporuje server



Obr. 5.10: Potvrzení přidání profilů všem koncovým stanicím

5.3 Testování sítě

1. Nyní se vrátíme na vyšší level do síťové hierarchie. Pravým tlačítkem klikneme na plochu a v menu vybereme *„Choose Individual Statistics“*. Otevře se nám okno, kde budeme vybírat, které události chceme při simulaci provozu na síti sledovat. Rozklikneme *„Global Statistics“* a zde vybereme **Email**, kde označíme možnosti

„Download Response Time (sec)“, časovou odezvu stahování Emailu a *„Traffic Sent (packets/sec)“*, počet poslaných paketů za časovou jednotku, viz obr. 5.11. Podobně vybereme ostatní statistiky, které chceme sledovat u Ftp, HTTP, Ethernet a Remote Login.

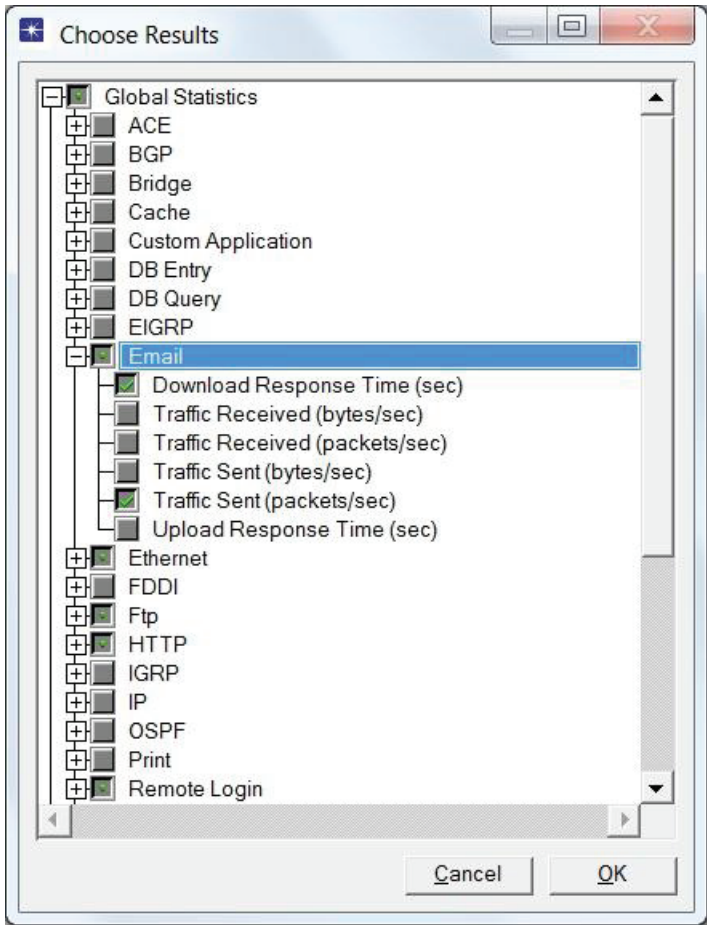
Ftp: *„Download Response Time (sec)“* a *„Traffic Sent (packets/sec)“*

HTTP: *„Page Response Time (seconds)“* a *„Traffic Sent (packets/sec)“*

Remote Login: *„Download Response Time (sec)“* a *„Traffic Sent (packets/sec)“*

Ethernet: *„Delay (sec)“*, tímto budeme sledovat zpoždění na lince

2. To je vše v části Global Statistics, nyní si otevřeme Link Statistics, dále point-to-point, kde zatrhneme: *„queuing delay →“*, zpoždění při řazení do front
   „throughput (packets/sec) →“, propustnost
   „utilization→“, využití linky



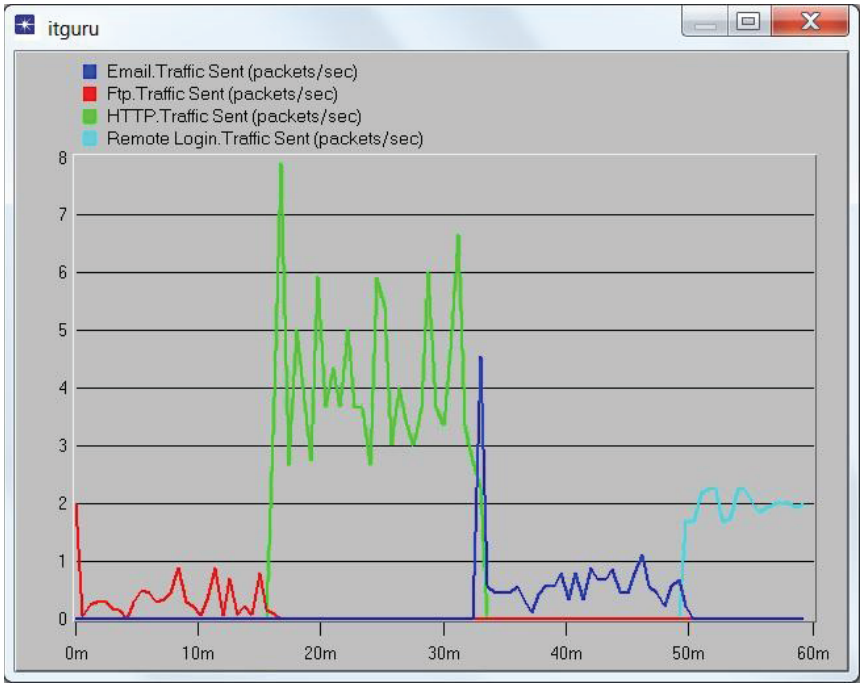
Obr. 5.11: Definování statistik

3. Právě jsme nastavili statistiky, které chceme sledovat. Nyní budeme simulovat síť. Nejdříve si projekt uložíme. Nastavovat simulace lze v záložce *„Scenarios/Manage Scenarios...“*. Zde v kolonce *„results“* zvolíme možnost **<collect>** a ostatní veličiny

měnit nebudeme. Potvrdíme tlačítkem **OK**, simulace se spustí a po skončení simulace okno zavřeme.

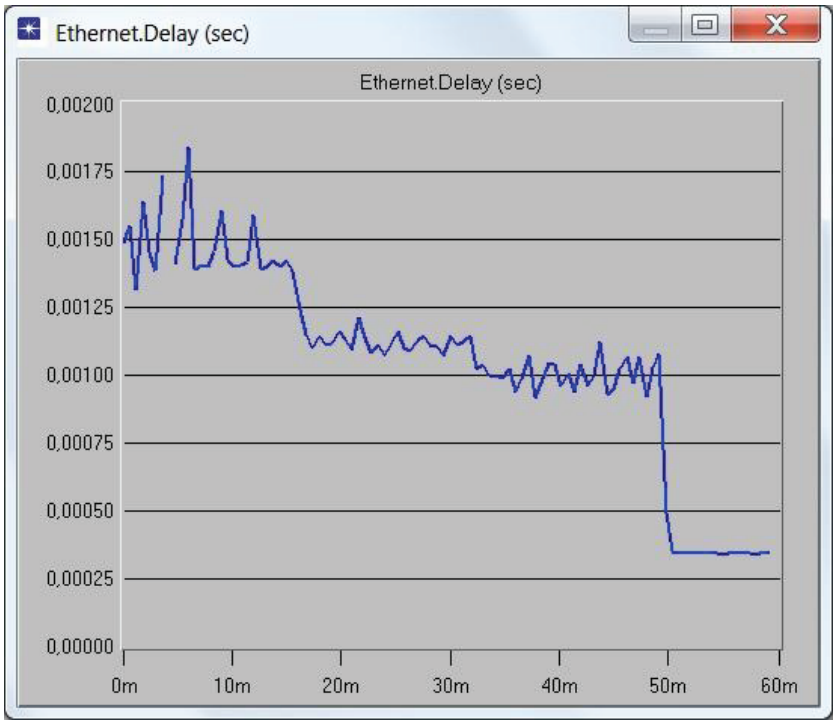
5.4 Vyhodnocení výsledků

1. Pravým tlačítkem klikneme na plochu a zvolíme možnost *„View Results“*. V levé části okna máme námi vybrané statistiky, v pravé části se vykreslují grafy a můžeme zde ovlivňovat jak. Klikneme proto na *„Stackted Statistic“* a vybereme možnost **Overlaid Statistics**. To nám zajistí, že se křivky budou vykreslovat do jednoho grafu. V levé části klikneme na *„Global Statistics“* a postupně otevřeme všechny aplikace a vybereme možnost *„Traffic Send (packets/sec)“*, tlačítkem **Show** si můžete zobrazit graf ve větší velikosti, viz obr. 5.12.



Obr. 5.12: Počet poslaných paketů za jednotku času

2. Zkuste si zobrazit další statistiky, například *„Delay(sec)“* - zpoždění , *„Response Time“* - dobu odezvy, u jednotlivých aplikací. Nebo definovat jiné statistiky podobně jako v bodě 1. Ve složce *„Object statistics“* si můžete zobrazit statistiky sledované na linkách, jako například propustnost, zpoždění na ethernetu, viz obr 5.13, a další.



Obr. 5.13: Zpoždění na ethernetu

6 Protokol VoIP a jeho varianty

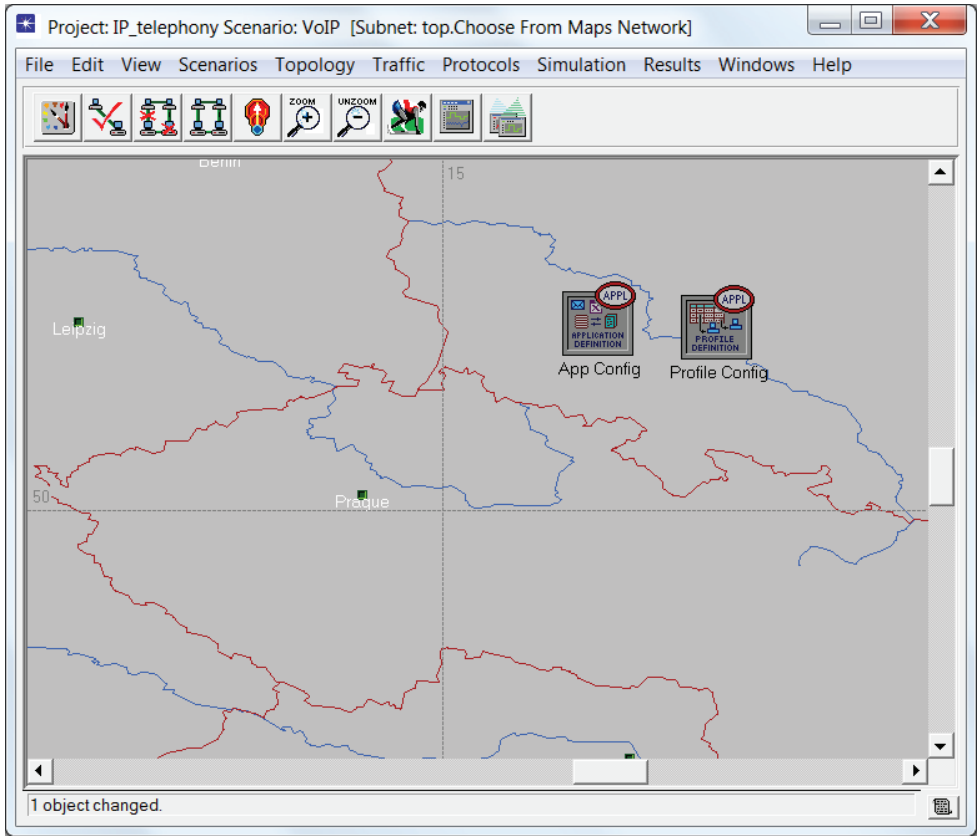
6.1 Úvod k laboratorní úloze

Laboratorní úloha se zabývá srovnáním protokolů, které zajišťují přenos hlasu přes prostředí internetu.

6.2 Vytvoření topologie

1. Spustíme program OPNET IT GURU academic edition.
2. Z nabídky zvolíme **File/new...** (Ctrl+N), v okně ponecháme položku *„Project“* a potvrdíme tlačítkem **OK**.
3. V dalším okně zadáme název projektu například: **IP_telephony** a název scénáře: **VoIP** a klikněte na **Next**.
4. Pro vytvoření prázdného scénáře vybereme *„Create Empty Scenario“* a potvrdíme **Next**.
5. V další nabídce vybereme *„Choose From Maps“* a potvrdíme. V rolovací nabídce vyhledáme *„Europe“* a potvrdíme.
6. V další záložce vybereme předpřipravené soubory, které budeme používat: *„ethernet, internet_toolbox, LANs, links a routers“* a potvrdíme. V posledním okně klikneme na OK. Tím se vytvoří projekt.
7. Objeví se mapa Evropy a okno **Object palette**, na hlavní liště ikona  . Tlačítkem **ZOOM** opět na hlavní liště se přiblížíme na Českou Republiku.
8. Z okna Object Palette vybereme a přeneseme do plochy projektu dva objekty: *„Application Config“*, který se používá pro výběr a nastavení aplikací a *„Profile Config“*, kde se nastavují parametry těchto aplikací.
9. Pravým tlačítkem klikneme na blok *„Application Config“* a zvolíme *„Edit Attributes“,* kde nastavujeme vlastnosti objektů. V položce *„name“* přejmenujeme blok na **App Config**. Stejným způsobem přejmenujeme blok *„Profile Config“* na **Profile Config**, viz obr. 6.1.
10. Nyní budeme nastavovat objekt **App Config**. Znovu otevřeme okno atributů a dále v řádku *„Application Definitions“* vybereme atribut *„Edit“*. Objeví se okno *„(Application Definitions) table“*, kde změníme hodnotu *„Rows“* na **3**. Do prvního řádku v kolonce *„Name“* napíšeme jméno aplikace **PCM_quality_speach**. Ve sloupci *„Description“* klikneme na **None** a zvolíme **Edit**. Objeví se okno *„(Description)*

Table“. V řádku s názvem *„Voice“* změníme hodnotu z Off na **PCM Quality Speach**, viz obr. 6.2. Nadefinujeme další dvě aplikace. První z nich bude mít název **GSM_quality_speach** a v okně *„(Description) Table“* změníme hodnotu na **GSM Quality Speach.** Třetí aplikace bude mít název **IP_telephony** a hodnotu v okně *„(Description) Table“* **IP Telephony**. Vše potvrdíme **OK**. Tímto jsme nastavili aplikace, které budeme dále využívat.



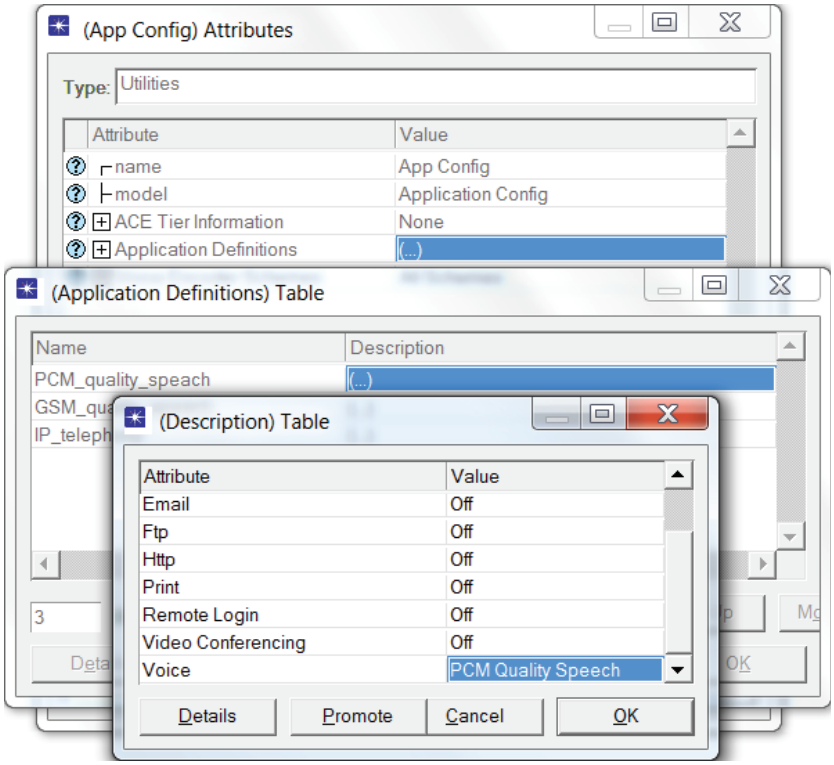
Obr. 6.1: Ukázka bloků Application Config a Profile Config

11. V dalším bodě budeme nastavovat objekt **Profile Config**. Stejným způsobem si zobrazíme atributy. Otevře se okno *„(Profile Configuration) table“*. Zde změníme hodnotu *„Rows“* na **3**. první řádek nadefinujeme takto:

Profile Name:	PCM_Quality
Operation Mode:	Simultaneous, všechny profily se budou spouštět zároveň, ostatní ponecháme stejné.
Start Time (sec):	constant (0), spuštění profilu
Duration (sec):	constant (1000), doba trvání profilu

12. Dále nastavíme aplikaci, která poběží v našem profilu. V jednom profilu můžeme spouštět více aplikací. Klikneme ve sloupci *„Applications“* na **None** a z rolovací

nabídky vybereme **edit**. Objeví se okno *„Applications Table“*, zde nastavíme hodnotu *„Rows“* na **1**. Klikneme na položku **Name** a vybereme aplikaci. V našem případě to bude *„PCM_quality_speach“*, v kolonce *„Start Time Offset (seconds)“* nastavíme **constant (0)** a *„Duration (seconds)“* hodnotu **exponecial (120)**, viz obr. 6.3. Okno potvrdíme. Podobně nastavíme další dva profily.



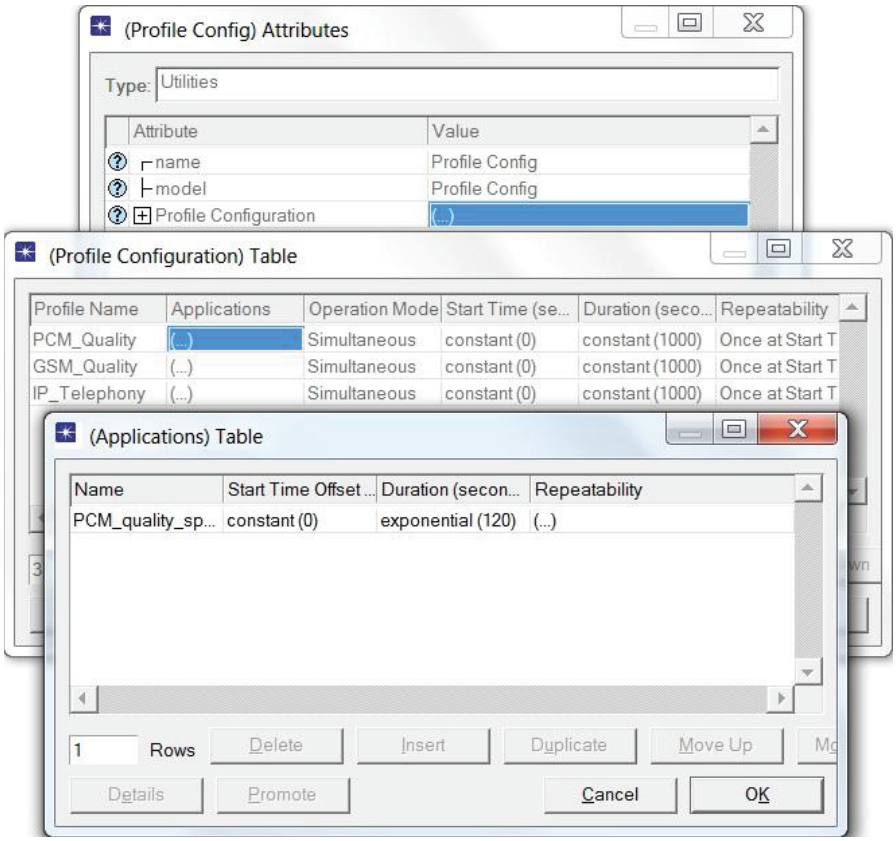
Obr. 6.2: Postupné nastavení aplikace PCM_quality_speach

Profil **PCM_Quality**:

Profil Name:	PCM_Quality
Operation Mode:	Simultaneous
Start Time (sec):	constant (0)
Duration (sec):	constant (1000)

Profil **IP_Telephony**:

Profil Name:	IP_Telephony
Operation Mode:	Simultaneous
Start Time (sec):	constant (0)
Duration (sec):	constant (1000)



Obr. 6.3: Postupné nastavení profilu PCM_Quality

13. Máme nastavené aplikace a profily a nyní si vystavíme síť. Otevřeme si okno *„Object Palette“* a z něho si na plochu vložíme tyto objekty: „ip32_cloud“, 3x „ethernet4_slip8_gtwy“ a 6x „ethernet_wkstn“. Objekty si pojmenujeme.

    ip32_cloud: internet

    ethernet4_slip8_gtwy: router_1, router_2 a router_server

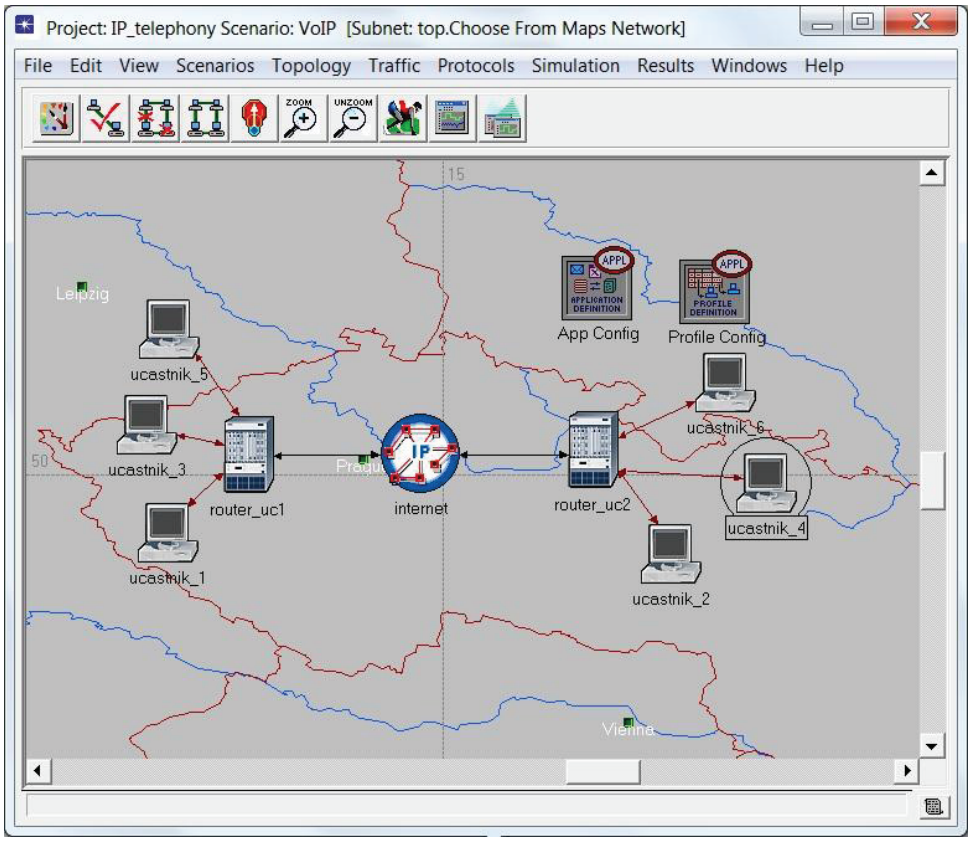
    ethernet_wkstn: ucastnik_1 - ucastnik_6

14. Objekty pospojujeme. Účastníky spojíme se směrovači technologií **10BaseT** a všechny směrovače s blokem internet technologií **PPP_DS0**, viz obr. 6.4. Obě najdeme v paletě objektů.

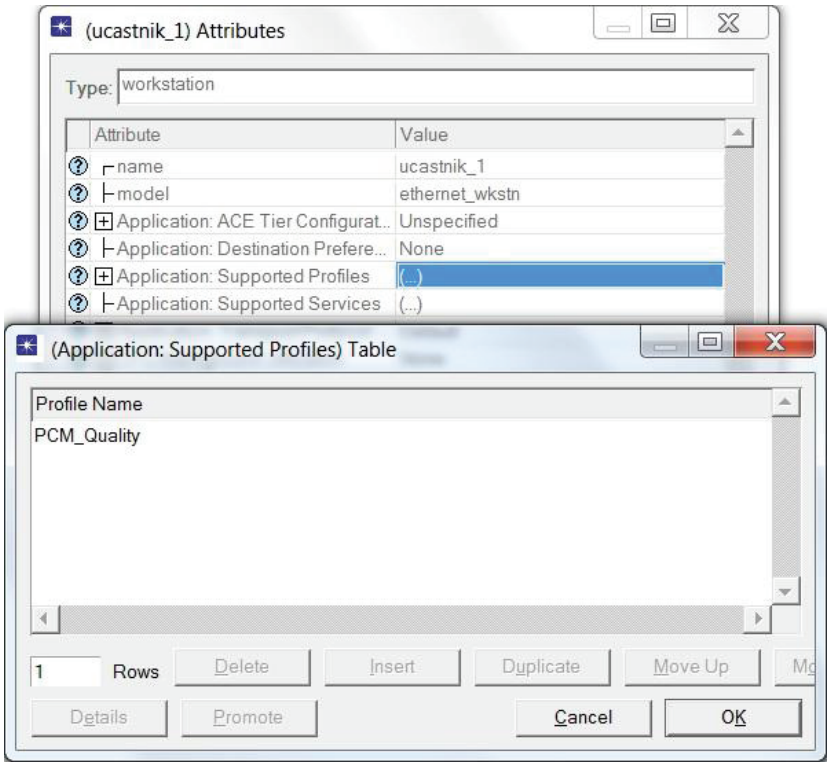
15. Dále nastavíme, aby účastníci podporovali zvolené aplikace. S pomocí klávesy CTRL vybereme dva objekty: **ucastnik_1** a **ucastnik_2** a zobrazíme si jejich parametry. A v kolonce *„Application: Supported Profiles“* změníme hodnotu **None** na **Edit**. Objeví se okno *„(Application: Supported Profiles) Table“*, zde upravíme hodnotu *„Rows“* na **1** a následně vložíme profil **PCM_Quality** a potvrdíme **Ok**. To stejné uděláme v kolonce *„Application: Supported Services“*, viz obr. 6.5. Nezapomeneme zaškrtnout kolonku *„Apply Changes to Selected Objects“*, ta nám zajistí, že nebudeme

muset nastavovat oba účastníky nadvakrát. Pro kontrolu, že nám změny proběhly se v levém dolním rohu, se napíše *„2 Object Changes“*, viz obr. 6.6.

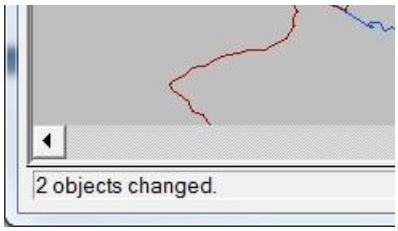
16. Stejným způsobem nadefinujeme podporované profily a služby pro další dva, kde bude **GSM_Quality** a poslední dva kde bude **IP_Telephony**.



Obr. 6.4: Vytvořená síť se všemi objekty



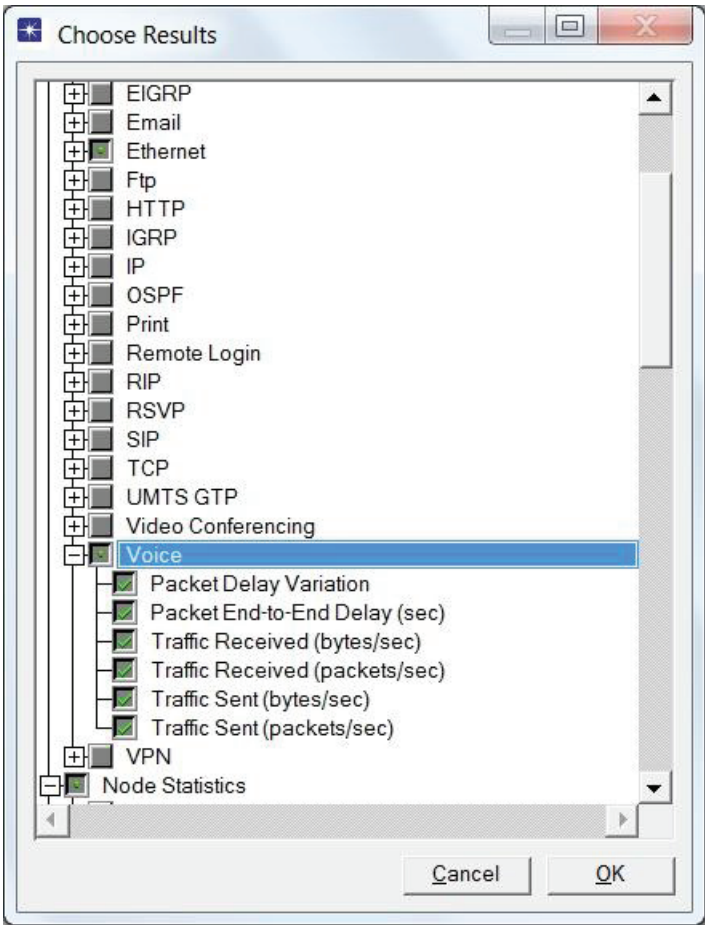
Obr. 6.5: Účastník 1 podporuje aplikaci PCM_Quality



Obr. 6.6: Potvrzení změny dvou objektů

6.3 Testování sítě

1. Nyní nastavíme jaké události při testování chceme sledovat. Pravým tlačítkem klikneme na plochu a vybereme možnost *„Choose Individual Statistics“*. Z nabídky *„Global Statistics“* vybereme záložku **Voice**, viz obr. 6.7. V záložce *„Node Statistics“* to budou **Voice Application**, **Voice Called Party** a **Voice Calling Party**. V záložce „Link Statistics“ **point-to-point**.



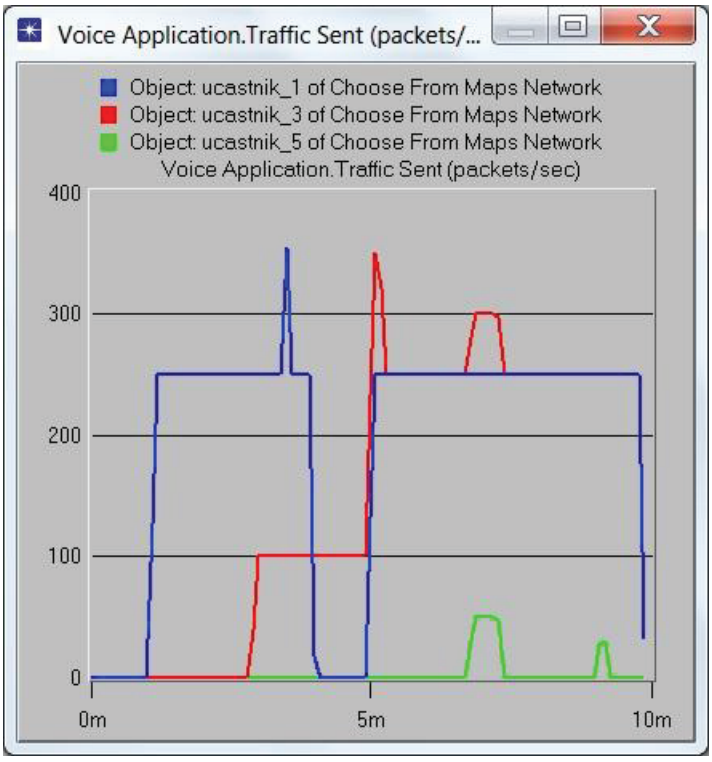
Obr. 6.7: Definování sledovaných globálních statistik

2. Právě jsme nastavili statistiky, které chceme sledovat. Nyní budeme simulovat síť. Nejdříve si projekt uložíme. Nastavovat simulace lze v záložce *„Scenarios → Manage Scenarios...“*. Zde v kolonce *„results“* zvolíme možnost **<collect>**, v *„Sim Duration“* nastavíme hodnotu **10** a časové jednotky *„Time Units“* změníme na **minute (s)**. Potvrdíme tlačítkem **OK**, simulace se spustí. Po skončení simulace okno zavřeme.

6.4 Vyhodnocení výsledků

1. Pravým tlačítkem klikneme na plochu a zvolíme možnost *„View Results“*. V levé části okna máme námi vybrané statistiky, v pravé části se vykreslují grafy a můžeme zde ovlivňovat jak. Klikneme proto na *„Stackted Statistic“* a vybereme možnost **Overlaid Statistics**. To nám zajistí, že se křivky budou vykreslovat do jednoho grafu. V levé části klikneme na *„Object Statistics“* postupně otevřeme účastníky 1, 3 a 5, dále „Voice Application“ a vybereme možnost **Traffic Sent (packet/sec),** tlačítkem **Show** si

můžeme zobrazit graf ve větší velikosti, viz obr. 6.8. Je tu vidět rozdíl kolik paketů posílají účastníci, kteří používají jiné aplikace.



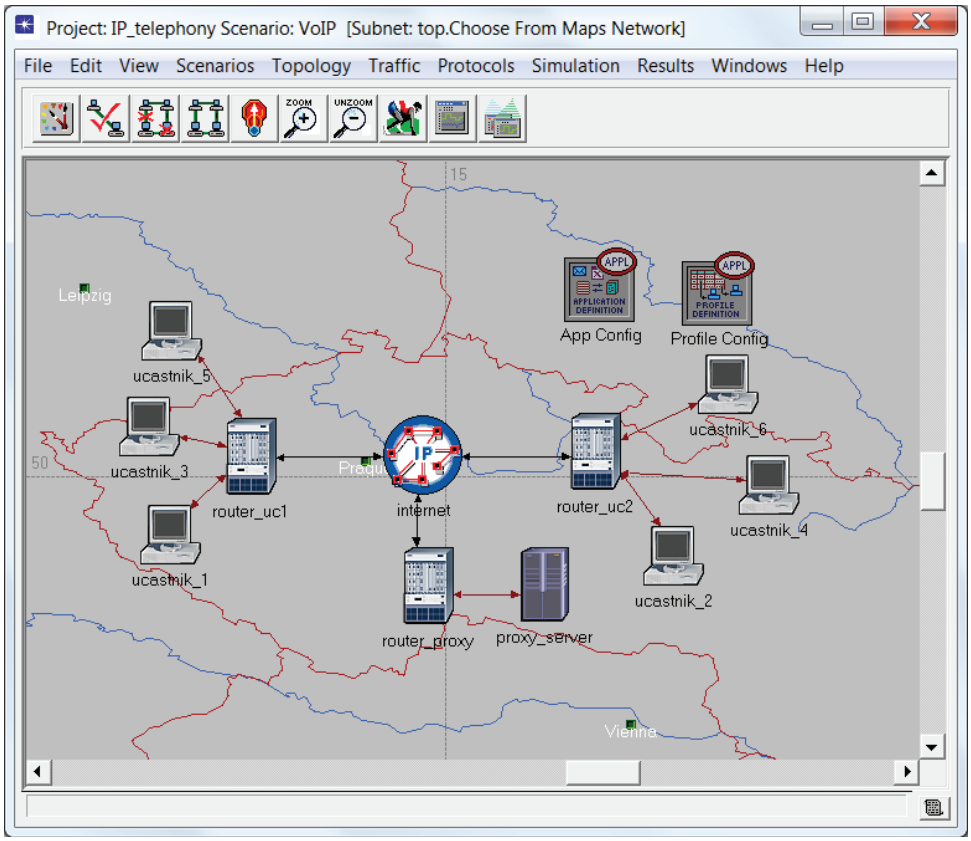
Obr. 6.8: Statistika Traffic Sent pro všechny aplikace

3. Můžeme si zobrazit další statistiky. Například *„Global statistics → Voice → Traffic sent“*, kde můžete vidět celkový počet přenesených paketů. Nebo *„Global statistics → Ethernet → Delay“*, celkové zpoždění na lince.

6.5 Protokol SIP

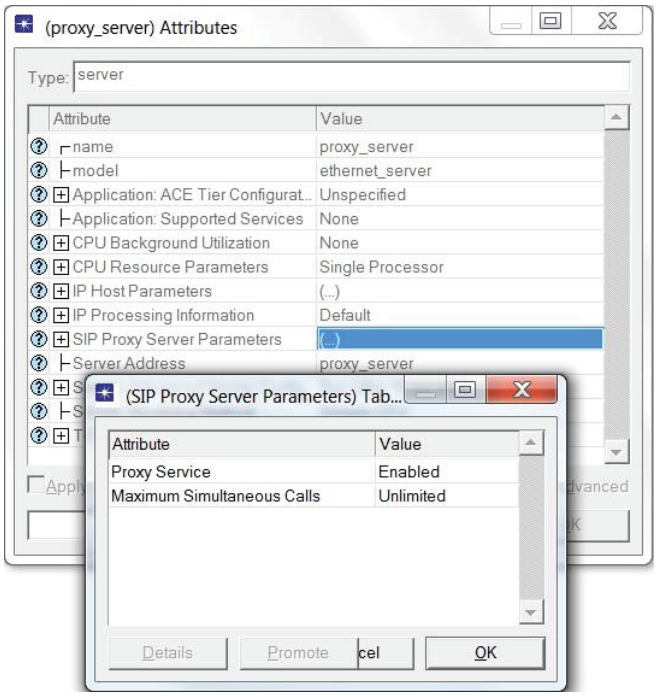
1. Nyní implementujeme protokol **SIP** (Session Initiation Protocol) neboli česky, protokol pro inicializaci relací. Pro posouzení změn si vytvoříme druhý scénář. Na hlavní liště klikneme na *„Scenarios“* a zvolíme *„Duplicate Scenario...“* nebo také pomocí kláves CTRL + SHIFT + D, pojmenujeme ho **SIP**. Bude za potřebí do naší sítě doplnit pár objektů.

2. Otevřeme si paletu objektů a na plochu si vložíme *„ethernet4_slip8_gtwy“* a *„ethernet_server“*. Oba pojmenujeme **router_server** a **proxy_server**. Proxy_server propojíme se směrovačem technologií **1000BaseT** a směrovač s internetem technologií **PPP_DS0**, viz obr. 6.9.



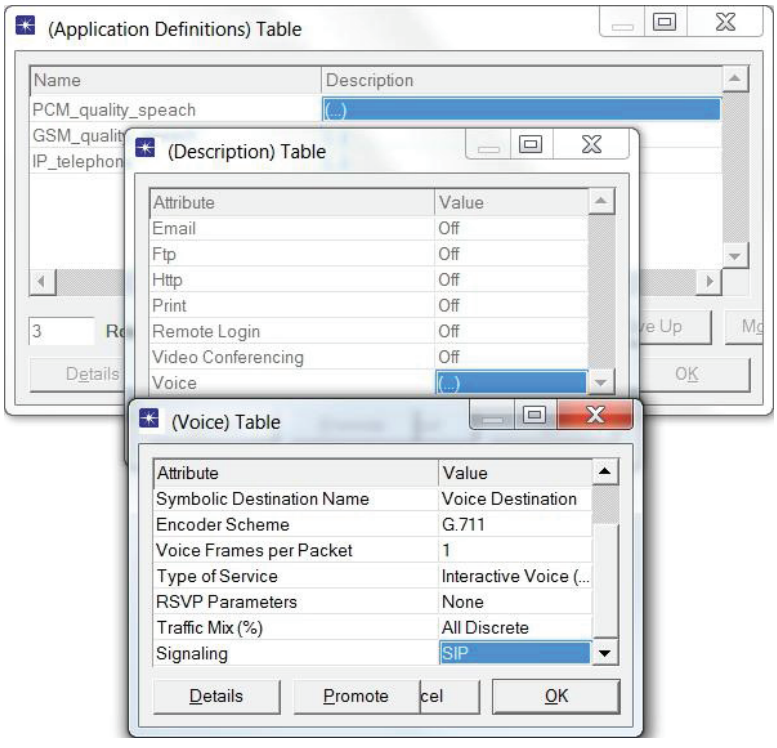
Obr: 6.9: Doplnění sítě o proxy_server a směrovač

3. Otevřeme si atributy proxy_serveru, zde budeme definovat dva údaje. Nejdříve klikneme v řádku „*SIP Proxy Server Parameters*“ na **(...)** → **Proxy Service, Disable** změníme na **Enable** a potvrdíme. Dále nastavíme adresu serveru. Ve stejnojmenném řádku zvolíme Edit a pojmenujeme adresu **proxy_server**, viz obr. 10.



Obr. 6.10: Nastavení atributů proxy serveru pro parametry SIP

4. Dále upravíme aplikace, aby podporovaly SIP protokol. Stejným postupem jako v bodě 10. Tedy atributy „*App Config* → *Application Definitions* → *PCM_quality_speach* → *Voice* → *Edit* → *Signaling* → *SIP*“. Stejným způsobem postupujeme i u dvou zbývajících aplikací, viz obr. 6.11.

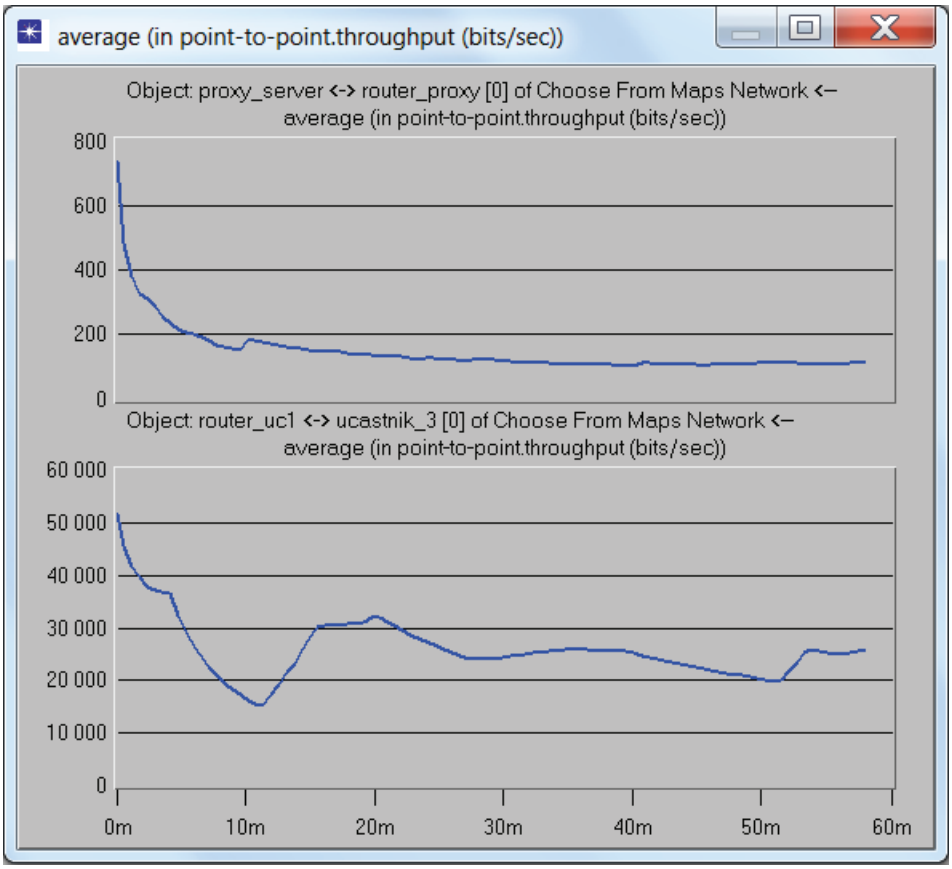


Obr. 6.11: Změněná aplikace na podporu protokolu SIP

5. Nakonec přidáme podporu protokolu SIP i účastníkům. Pravým tlačítkem klikneme na pracovní stanici a vybereme „*Select Similar Nodes*“, tím označíme všechny účastníky. Vybereme jejich atributy a vybereme **SIP UAC Parameters**. Hodnotu „UAC Service“ změníme na **Enable**. Dále klikneme na hodnotu **Unspecified** v řádku „Proxy Server Specification“ a po kliknutí na **specify…** vložíme **proxy_server**. Tím i účastníci podporují SIP.

6.6 Testování sítě

1. Nastavíme sledované charakteristiky. Otevřeme si okno „Choose Results“, všechny charakteristiky, které nám zde zbyly po předchozím scénáři, odebereme a vybereme nové. V záložce „*Node Statistics*“ **SIP UAC**, **SIP UAS**, **Voice Application**, **Voice Called Party** a **Voice Calling Party**. Další v záložce „*Link Statistics*“ **point-to-point**.
2. Projekt si uložíme. Spustíme si simulaci opět přes menu „*Scenarios → Manage Scenarios*“. Tady upravíme ve sloupci „results“ hodnotu na **<collect>** a změníme čas simulace na **1 hodinu**. Potvrdíme tlačítkem **Ok** a simulace se spustí, po skončení zavřeme okno.
3. Nyní si zobrazíme výsledky, opět pomocí pravého tlačítka a výběru „*View Results*“. Zobrazíme si propustnost různých linek sítě, například linky mezi proxy serverem a routrem, který je s ním spojený. *„Object Statistics → Choose From Maps Network → proxy_server ↔ router_proxy [0] → point-to-point → throughput (bits/sec)→“*. Pro porovnání si zobrazíme propustnost například pro účastníka 3. Výběr provedeme stejným způsobem. *„Object Statistics → Choose From Maps Network → router_uc1 ↔ ucastnik_3 [0] → point-to-point → throughput (bits/sec)→“*, viz obr. 6.12.



Obr. 6.12: Rozdíl propustnosti linek u SIP serveru a účastníka

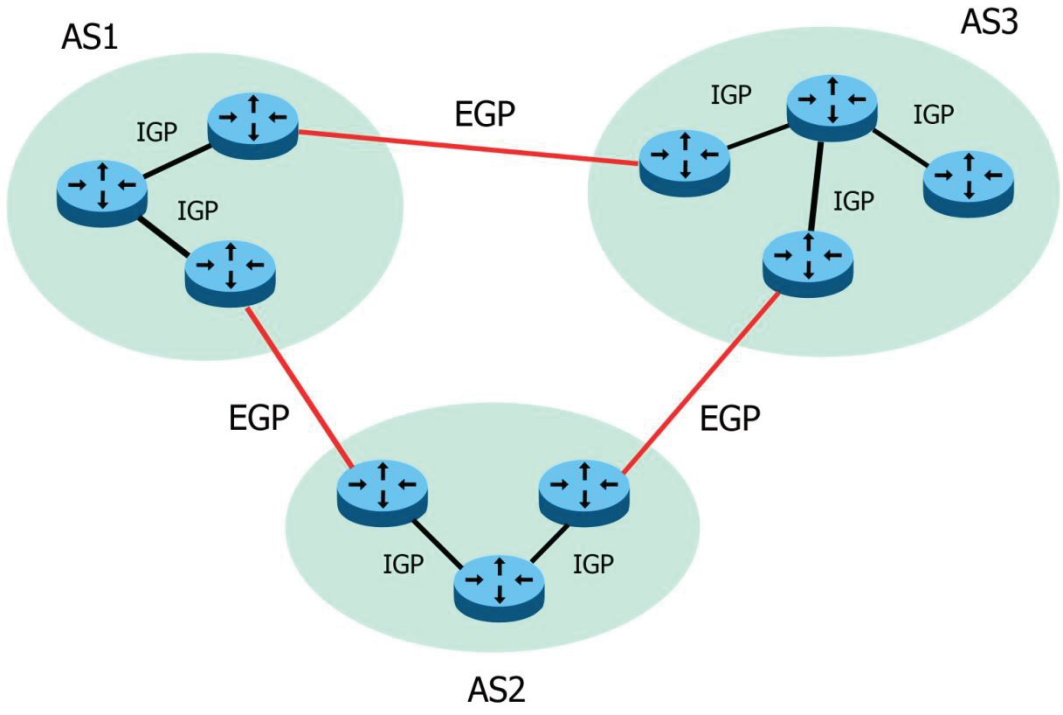
4. Seznamte se s dalšími charakteristikami přenosu a zobrazte si např. v záložce „*Global Statistics → Voice → Traffic Sent (bytes/sec)*“ nebo v záložce „*Object Statistics → Choose From Maps Network → proxy_server → SIP UAS → Active Calls* “ si zobrazte počet vytvořených hovorů.

7 Směrovací protokol IS-IS

7.1 Úvod k laboratorní úloze

Směrování je bezesporu jedna z nejdůležitějších operací při komunikaci mezi dvěma vzdálenými uživateli v počítačové síti. Přenos probíhající přes mnoho síťových prvků by byl bez něho ztracen.

Prakticky můžeme směrovací protokoly rozdělit do dvou skupin. První skupinou jsou EGP (Exterior Gateway Protocols) neboli vnější směrovací protokoly, mezi které patří BGP (Border Gateway Protokol). Druhou skupinou jsou IGP (Interior Gateway Protocols) neboli vnitřní směrovače, mezi které patří RIP (Routing Information Protocol), OSPF (Open Shortest Path First), IGRP (Interior Gateway Routing Protocol), EIRGP (Enhanced Interior Gateway Routing Protocol) a posledním je IS-IS (Intermediate System to Intermediate System), kterým se budeme v laboratorní úloze zabývat. Vnitřní směrovací protokoly jsou používány uvnitř autonomních systémů, naopak vnější tyto autonomní systémy spojují, viz obr. 7.1.

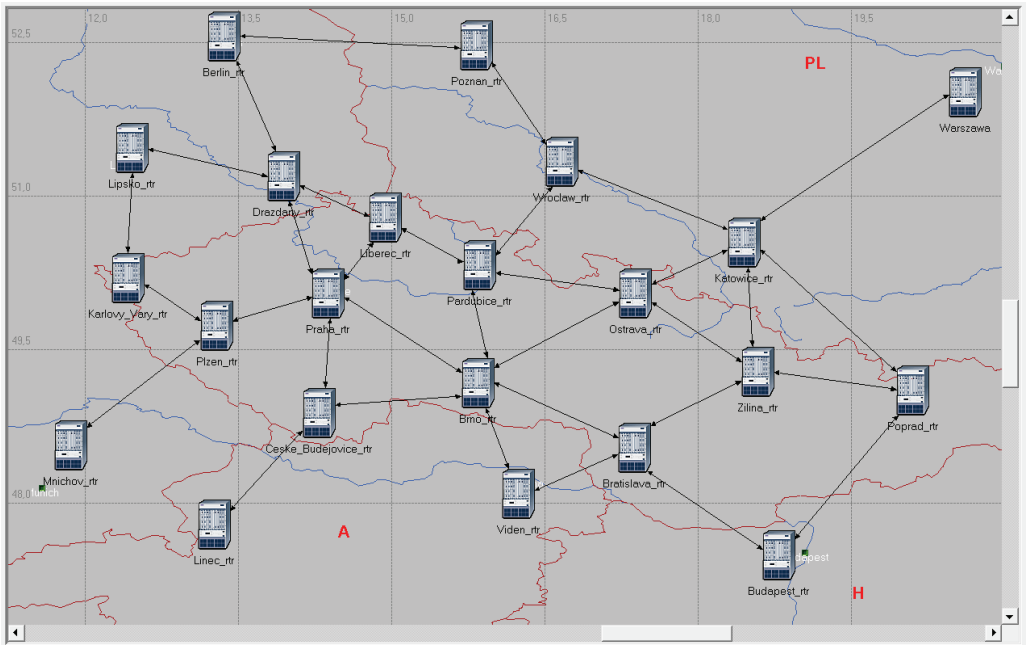


Obr. 7.1: Ilustrativní ukázka směrovacích protokolů v a mezi AS

Směrovací protokol IS-IS používá algoritmus stavu spojů, tím je podobný protokolu OSPF. Metriku určuje podle ceny cesty, součet všech cest mezi zdrojem a cílem. Metrika nemusí být samotný rozhodující faktor, další jsou propustnost, zpoždění a chybovost. IS-IS může rozdělit směrovače do dvou úrovní neboli oblastí. Směrovače Level 1, které budou znát svoji oblast a směrovače Level 2, které budou směrovat provoz mezi těmito oblastmi.

7.2 Počáteční nastavení

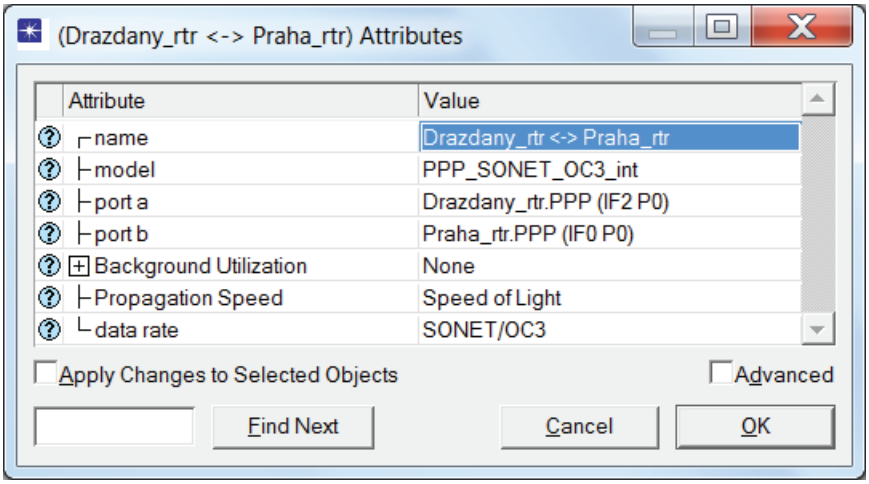
1. Spustíme program OPNET IT GURU Academic Edition.
2. Z nabídky zvolíme **File/Open…** (Ctrl+O), v otevřeném okně najdeme projekt *„pripravena_sit_IS_IS“* a potvrdíme **OK**. Zobrazí se okno s připravenou topologií, viz obr. 7.2.



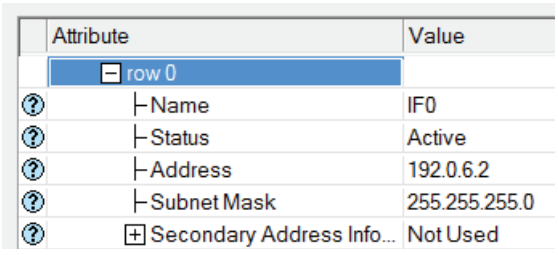
Obr. 7.2: Připravená síť pro laboratorní úlohu

3) Abychom postupným ukládáním nepřepsali připravenou síť, uložíme si projekt pod jiným názvem. *„File → Save as“* a pojmenujeme ho **lab_IS_IS**.
4) Pro komunikaci v síti je nutné přiřadit každému z uzlů jedinečnou adresu v rámci celé sítě. V připravené síti je toto už téměř hotové. Nakonfigurujeme si tedy jen jeden uzel.
5) Jediný uzel, který není nakonfigurován, je uzel Praha. Zobrazíme si tedy jeho atributy. Pravým tlačítkem klikneme na směrovač a vybereme *„Edit Attributes“*. Dále na položku *„IP Routing parameters → Interface Information“*. Zde vybereme rozhraní,

které chceme nastavit. Která linka je přiřazena, ke kterému portu zjistíme tak, že pravým tlačítkem klikneme na určitou linku a zobrazíme si její atributy, viz obr. 7.3. Zde je vidět, že linka do Drážďan je na směrovači Praha_rtr připojená do portu IF0. Otevřeme si znovu okno směrovače Praha_rtr a rozklikneme *„row 0“*, kde změníme položku *„Address“* na **192.0.6.2** a položku *„Subnet Mask“* na **255.255.255.0,** viz obr. 7.4. Další rozhraní a loopback nastavíme stejným způsobem pomocí tab. 7.1. Konfigurace pro loopback není v záložce *„Interface Information“,* ale v *„Loopback Interfaces“.*



Obr. 7.3: Atributy linky Drazdany_trt ↔ Praha_rtr



Obr. 7.4: Nastavení adresy a masky sítě pro rozhraní IF0

Tab. 7.1. Nastavení rozhraní pro směrovač Praha_rtr

Rozhraní	IP adresa	Maska sítě
IF0	192.0.6.2	255.255.255.0
IF1	192.0.19.1	255.255.255.0
IF2	192.0.18.1	255.255.255.0
IF3	192.0.17.1	255.255.255.0
IF4	192.0.15.2	255.255.255.0
Loopback	192.0.16.1	255.255.255.0

6) Nyní máme nastavené všechny adresy. Jako kontrolu, že jsme nastavení provedli správně, si můžeme zobrazit směrovací tabulku. Označíme uzel **Praha_rtr**. Na hlavní liště zvolíme *„Protocols → IP → Routing → Export routing table for selected routers“*. Spustíme si simulaci, tím že na hlavní liště zvolíme *„Scenario → Manage Scenario“*, zobrazí se nám okno, kde v kolonce *„Results“* zvolíme **<collect>**. Potvrdíme **Ok**. Tím spustíme simulaci, po skončení okno zavřeme. Dále na hlavní liště klikneme na *„Results → Open simulation log“*, v otevřeném okně nalezneme uzel **Praha_rtr** a kliknutím ve sloupci *„Message“* na **COMMON ROUTE TABLE** se nám zobrazí směrovací tabulka, viz obr. 7.5.

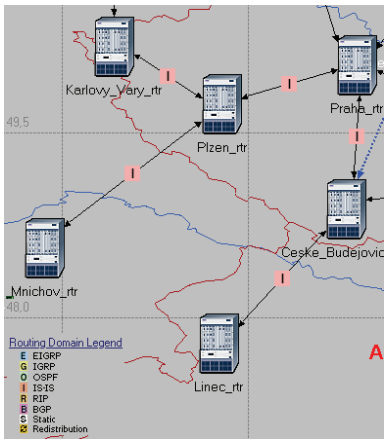
```
COMMON ROUTE TABLE snapshot for:

  Router name: Choose From Maps Network.Praha_rtr
      at time: 3600,00 seconds

ROUTE TABLE contents:

   Dest. Address     Subnet Mask      Next Hop       Interface Name    Metric      Protocol    Insertion Time
  ---------------  ---------------  ---------------  ---------------   ------     --------    --------------

  192.0.6.0        255.255.255.0    192.0.6.2        IF0               0           Direct          0,000
  192.0.19.0       255.255.255.0    192.0.19.1       IF1               0           Direct          0,000
  192.0.18.0       255.255.255.0    192.0.18.1       IF2               0           Direct          0,000
  192.0.17.0       255.255.255.0    192.0.17.1       IF3               0           Direct          0,000
  192.0.15.0       255.255.255.0    192.0.15.2       IF4               0           Direct          0,000
  192.0.16.0       255.255.255.0    192.0.16.1       Loopback          0           Direct          0,000
  192.0.11.0       255.255.255.0    192.0.15.1       IF4               1           RIP             5,000
  192.0.12.0       255.255.255.0    192.0.15.1       IF4               1           RIP             5,000
  192.0.13.0       255.255.255.0    192.0.15.1       IF4               1           RIP             5,000
  192.0.5.0        255.255.255.0    192.0.19.2       IF1               1           RIP             5,000
  192.0.20.0       255.255.255.0    192.0.19.2       IF1               1           RIP             5,000
  192.0.21.0       255.255.255.0    192.0.19.2       IF1               1           RIP             5,000
  192.0.33.0       255.255.255.0    192.0.17.2       IF3               1           RIP             5,000
  192.0.34.0       255.255.255.0    192.0.17.2       IF3               1           RIP             5,000
  192.0.35.0       255.255.255.0    192.0.17.2       IF3               1           RIP             5,000
  192.0.3.0        255.255.255.0    192.0.6.1        IF0               1           RIP             5,001
```

Obr. 7.5: Směrovací tabulka směrovače Praha_rtr

7) Ve směrovací tabulce si můžeme všimnout, že námi nakonfigurované adresy jsou připojené přímo s metrikou 0. Ostatní adresy jsou naučené.

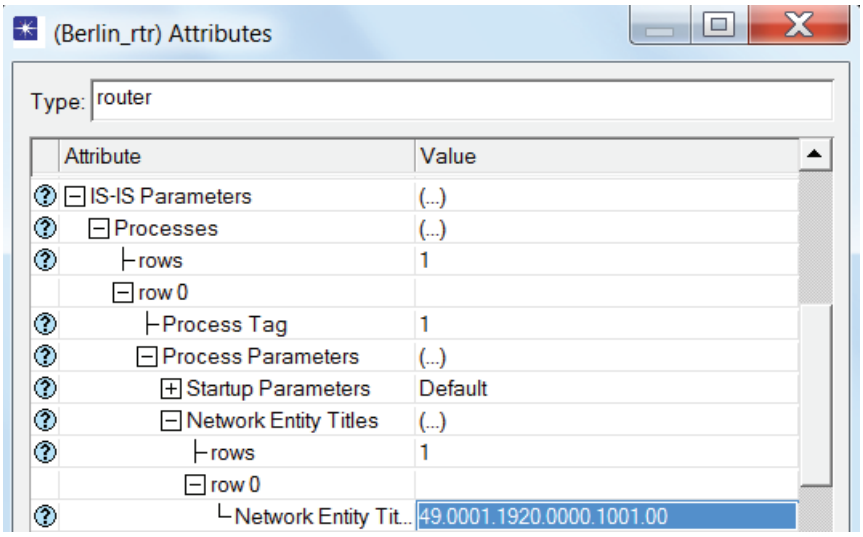
7.3 Konfigurace směrovacího protokolu IS-IS

1) Na směrovačích je standardně nastaven směrovací protokol RIP. Nastavíme tedy protokol IS-IS. Na hlavní liště klikneme na *„Protocols → IP → Routing → Configure Routing Protocols“*. Zatrhneme kolonku **IS-IS** a ponecháme *„Apply the above selection to subinterfaces“* a *„All interfaces (including loopback)“*. Potvrdíme **OK**. Tím jsme změnili směrovací protokol, na linkách se nám objevilo písmeno **I**, které nám to potvrzuje, viz obr. 7.6.



Obr. 7.6: Nastavení směrovacího protokolu IS-IS

2) Aby nám komunikace směrovacího protokolu fungovala správně, musíme nastavit **Network Entity Title** (NET) neboli ID. Tím můžeme rozdělit směrovače na několik skupin. Adresa NET je složená z několika částí. Například adresa **49**.**0001**.**1920.0000.1001**.**00**. Adresace začíná vždy samostatným oktetem neboli bajtem **49**. Následuje ID oblasti, **0001**. Dále ID směrovače, kde se většinou používá IP nebo MAC adresa. A poslední je jeden oktet, který je většinou nulový.
3) Zobrazíme si atributy směrovače **Berlin_rtr**, pravým tlačítkem klikneme na objekt a zvolíme *„Edit Atributes“*. Dále *„IS-IS Parameters → Processes → row 0 → Process Parameters → Network Entity Titles → row 0“* a do řádku *„Network Entity Title“* vložíme adresu **49.0001.1920.0000.1001.00**, viz obr. 7.7. V tomto scénáři budeme mít pouze jednu oblast, tedy u všech směrovačů bude ID oblasti 0001. Jako adresu jsme zvolili IP adresu loopbacku. Nastavení ostatních směrovačů opakujeme stejným způsobem podle tab. 7.2.



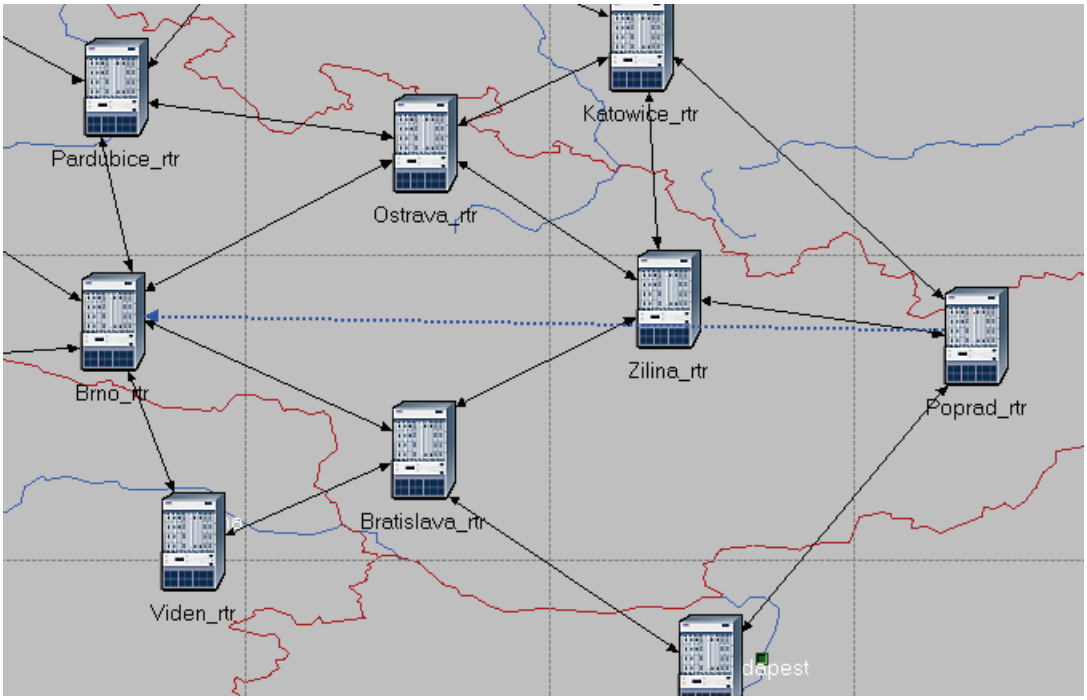
Obr. 7.7: Nastavení NET na směrovači Berlin_rtr

Tab. 7.2: Hodnoty NET všech směrovačů v síti

Směrovač	Network Entity Title
Berlin_rtr	49.0001.1920.0000.1001.00
Bratislava_rtr	49.0001.1920.0004.0001.00
Brno_rtr	49.0001.1920.0003.0001.00
Budapest_rtr	49.0001.1920.0004.2001.00
Ceske_Budejovice_rtr	49.0001.1920.0003.4001.00
Drazdany_rtr	49.0001.1920.0000.4001.00
Karlovy_Vary_rtr	49.0001.1920.0001.0001.00
Katowice_rtr	49.0001.1920.0005.4001.00
Liberec_rtr	49.0001.1920.0002.0001.00
Linec_rtr	49.0001.1920.0003.6001.00
Lipsko_rtr	49.0001.1920.0000.8001.00
Mnichov_rtr	49.0001.1920.0001.4001.00
Ostrava_rtr	49.0001.1920.0005.1001.00
Pardubice_rtr	49.0001.1920.0002.7001.00
Plzen_rtr	49.0001.1920.0001.2001.00
Poprad_rtr	49.0001.1920.0004.5001.00
Poznan_rtr	49.0001.1920.0002.2001.00
Praha_rtr	49.0001.1920.0001.6001.00
Viden_rtr	49.0001.1920.0003.7001.00
Waszawa_rtr	49.0001.1920.0005.6001.00
Wroclaw_rtr	49.0001.1920.0002.4001.00
Zilina_rtr	49.0001.1920.0004.8001.00

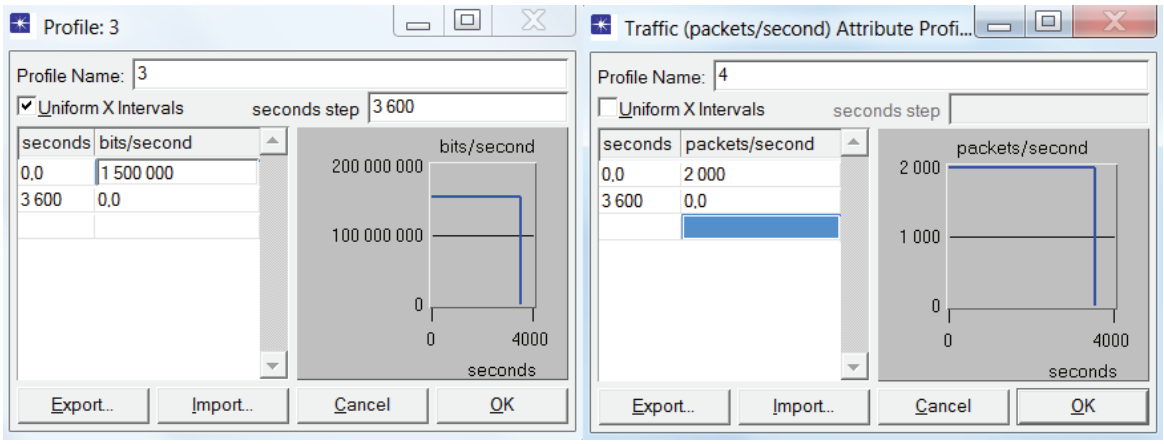
4) Nyní máme protokol IS-IS nakonfigurovaný. Pro všechny cesty je metrika stejná. Proto bude přenos mezi dvěma uzly, které mají více možných cest mezi sebou, rozdělen mezi

tyto cesty. Zobrazíme si paletu objektů, na hlavní liště  a z vyskakovací lišty, v horní části okna, vybereme *„internet_toolbox“*. Zde najdeme objekt **ip_traffic_flow**, klikneme na něj a spojíme uzly **Brno_rtr** a **Poprad_rtr,** viz obr. 7.8.



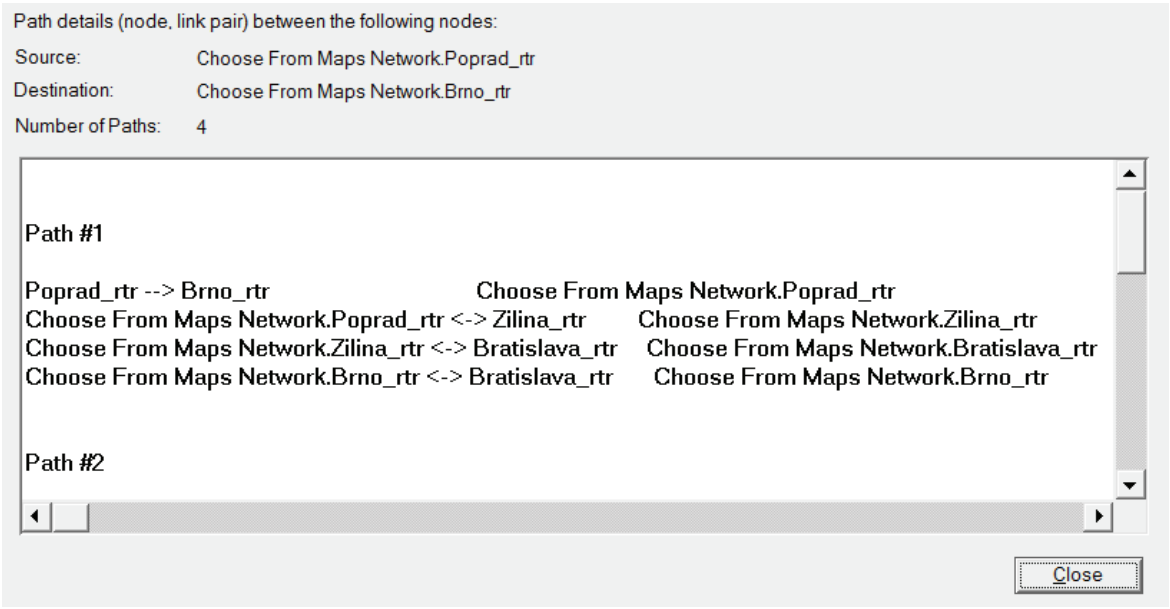
Obr. 7.8: Ip traffic flow mezi směrovači Poprad_rtr a Brno_rtr

5) Nastavíme velikost přenosu, v atributech objektu ip_traffic_flow, *„Edit Atributes“*. V řádku *„Traffic (bits/second)“* klikneme na **NONE** a vybereme **Edit**. V prvním řádku vložíme **0 seconds** a **1 500 000 bits/second**, v druhém řádku pak **3600 seconds** a **0 bits/second** a potvrdíme **OK**. V řádku *„Traffic (packets/second)“* klikneme opět na **NONE** a zvolíme **Edit**. V prvním řádku vložíme **0 seconds** a **2000 packets/second**, v druhém řádku pak **3600 seconds** a **0 packets/sekond,** viz obr. 7.9.



Obr. 7.9: Nastavení přenosu mezi uzly Brno_rtr a Poprad_rtr

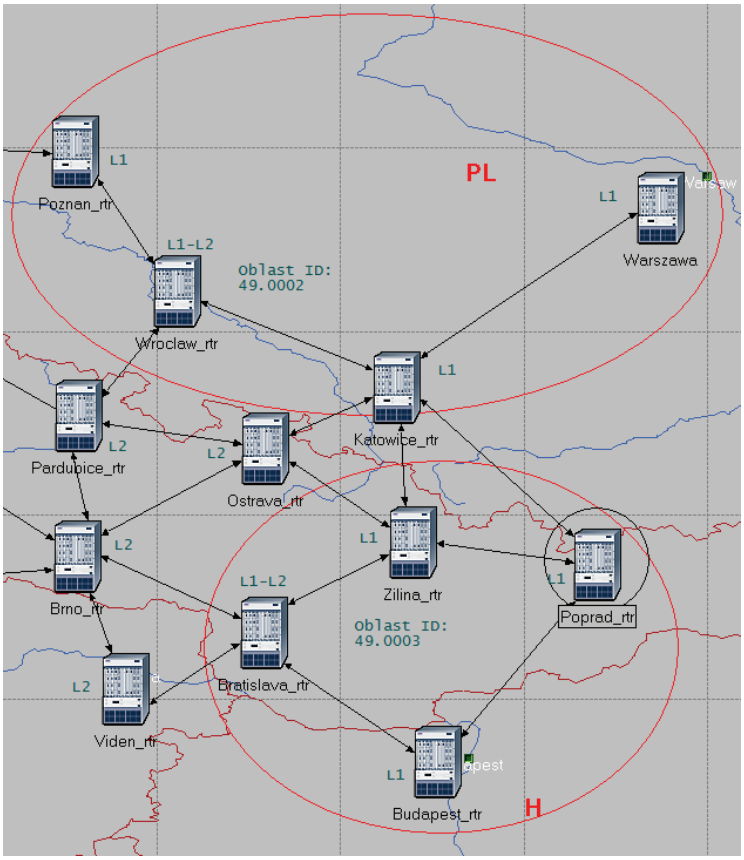
6) Nyní spustíme simulaci, na hlavní liště zvolíme *„Scenarios → Manage Scenario“*, zobrazí se nám okno, kde v kolonce *„Results“* zvolíme **<collect>**. Potvrdíme **Ok**, po proběhnutí okno zavřeme.
7) Na hlavní liště zvolíme *„Protocols → IP → Demands → Display Routes for Configured Demands...“*. Objeví se okno, kde se postupně proklikáme *„Sources (Choose from Maps Network) → Poprad_rtr → Brno_rtr → 1“* a v pravé částí okna klikneme na **Details**. Zobrazí se okno, kde je výpis všech možných tras z Popradu do Brna. Vidíme tedy, že jsou možné trasy 4, viz obr. 7.10.



Obr. 7.10: 4 možné cesty z Popradu do Brna

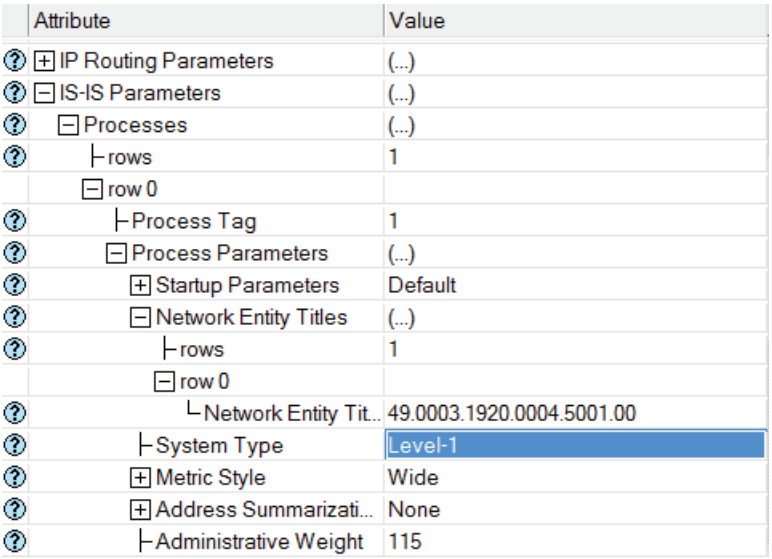
7.4 Rozdělení směrovačů do skupin podle ID

1) Scénář si duplikujeme, *„Scenarios → Duplicate Scenario...“*, nebo pomocí klávesové zkratky **Ctrl + Shift + D**. Nový scénář pojmenujeme **IS_IS_skupiny**. Potvrdíme **OK**.
2) Protokol IS-IS umožňuje rozdělení směrovačů do dvou úrovní, Level 1 a Level 2. Směrovač Level 1 je takzvaný místní a komunikuje pouze uvnitř své oblasti. Pokud musí přenést data mimo svojí oblast, směruje přenos k hraničnímu směrovači neboli Level 1-2 směrovači. Ten už směruje přenos mimo oblast přes páteřní směrovače, Level 2 směrovače.
3) Pro ukázku si zobrazte směrovací tabulku pro uzel **Poprad_rtr**, nejdřív ovšem musíme zvolit export směrovací tabulky. Označíme tedy směrovač **Poprad_rtr**. Na hlavní liště zvolíme *„Protocols → IP → Routing → Export routing table for selected routers“* a spustíme simulaci. Zobrazte si směrovací tabulku, stejným způsobem jako v předchozím případě pro uzel Praha_rtr. Nyní má zhruba 110 položek. Rozdělením směrovačů na oblasti se počet zmenší. Nyní vytvoříme dvě nové oblasti, viz obr. 7.11, dvě postačují pro vysvětlení problému. Výběr směrovačů a oblastí je zcela náhodný.



Obr. 7.11: Vytvoření dvou skupin směrovačů podle NET a levelu

4) Zobrazíme si atributy směrovače Poprad_rtr. Pravým tlačítkem klikneme na objekt a zvolíme *„Edit atributes“*. Dále *„IS-IS Parameters → Processes → row 0 → Process Parameters → Network Entity Titles → row 0“* a do řádku *„Network Entity Title“* vložíme adresu **49.0003.1920.0004.5001.00.** Jelikož bude směrovač Poprad_rtr typu místní, zvolíme v záložce *„Process Parameters → System type“* na **Level 1**, viz obr. 7.12.



Obr. 7.12: Změna NET a Levelu u směrovače Poprad_rtr

5) Nyní upravíme stejným způsobem ostatní směrovače podle tab. 7.3.

Tab. 7.3: Změny úrovní a NET pro všechny směrovače

Směrovač	Level	Network Entity Title
Berlin_rtr	Level 2	Zůstane stejný
Bratislava_rtr	Level 1-2	49.0003.1920.0004.0001.00
Brno_rtr	Level 2	Zůstane stejný
Budapest_rtr	Level 1	49.0003.1920.0004.2001.00
Ceske_Budejovice_rtr	Level 2	Zůstane stejný
Drazdany_rtr	Level 2	Zůstane stejný
Karlovy_Vary_rtr	Level 2	Zůstane stejný
Katowice_rtr	Level 1	49.0002.1920.0005.4001.00
Liberec_rtr	Level 2	Zůstane stejný
Linec_rtr	Level 2	Zůstane stejný
Lipsko_rtr	Level 2	Zůstane stejný
Mnichov_rtr	Level 2	Zůstane stejný
Ostrava_rtr	Level 2	Zůstane stejný

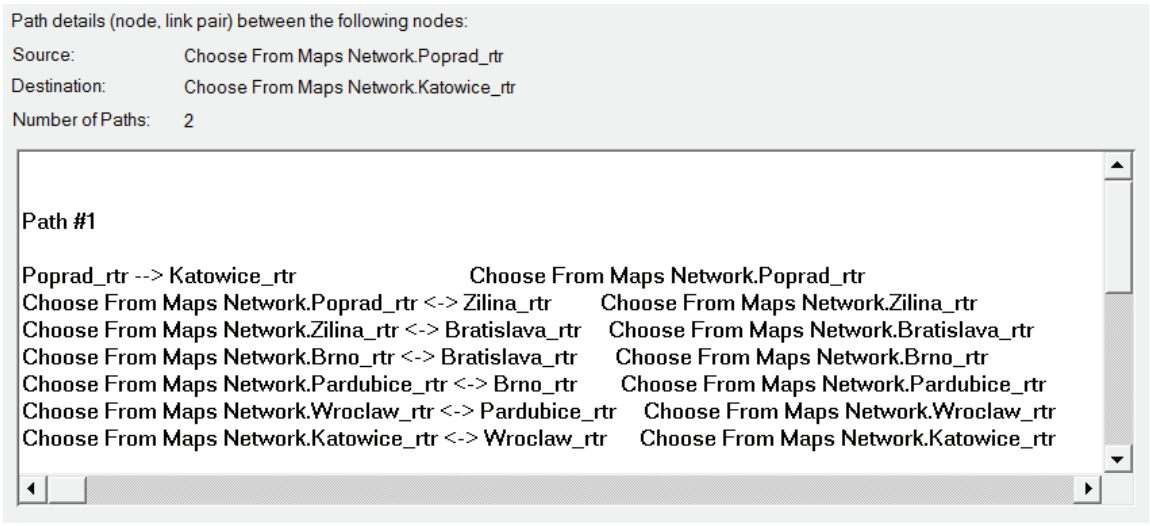
Pardubice_rtr	Level 2	Zůstane stejný
Plzen_rtr	Level 2	Zůstane stejný
Poprad_rtr	Level 1	49.0003.1920.0004.5001.00
Poznan_rtr	Level 1	49.0002.1920.0002.2001.00
Praha_rtr	Level 2	Zůstane stejný
Viden_rtr	Level 2	Zůstane stejný
Waszawa_rtr	Level 1	49.0002.1920.0005.6001.00
Wroclaw_rtr	Level 1-2	49.0002.1920.0002.4001.00
Zilina_rtr	Level 1	49.0003.1920.0004.8001.00

6) Ověříme si, kudy jde přenos mezi dvěma různými oblastmi a zároveň jejich místními směrovači. Pomocí zkopírování objektu **ip_traffic_flow** (Ctrl C, Ctrl V) si propojíme uzly **Poprad_rtr** a **Katovice_rtr**.
7) Spustíme simulaci *„Scenarios → Manage Scenario“*, zobrazí se nám okno, kde v kolonce *„Results“* pro scénář **IS_IS_skupiny** zvolíme **<collect>**. Potvrdíme **Ok**. Po proběhnutí okno zavřeme.
8) Nyní si zobrazíme opět směrovací tabulku směrovače Poprad_rtr *„Results → Open simulation log“*, kde najdeme směrovač, viz obr. 7.13. Změna je výrazná, směrovač vidí jen svoji oblast. Všimněme si, že na posledním řádku se objevila adresa 0.0.0.0 (default gateway) pro cestu mimo oblast 49.0003.

```
COMMON ROUTE TABLE snapshot for:

  Router name: Choose From Maps Network.Poprad_rtr
      at time: 3600,00 seconds

ROUTE TABLE contents:

   Dest. Address     Subnet Mask       Next Hop       Interface Name    Metric      Protocol    Insertion Time
   ---------------  ---------------  ---------------  ---------------   ------      --------    --------------

   192.0.47.0       255.255.255.0    192.0.47.2       IF0               0           Direct           0,000
   192.0.46.0       255.255.255.0    192.0.46.2       IF1               0           Direct           0,000
   192.0.44.0       255.255.255.0    192.0.44.2       IF2               0           Direct           0,000
   192.0.45.0       255.255.255.0    192.0.45.1       Loopback          0           Direct           0,000
   192.0.31.0       255.255.255.0    192.0.44.1       IF2               30          IS-IS            0,500
                                     192.0.46.1       IF1               30          IS-IS            0,500
   192.0.38.0       255.255.255.0    192.0.44.1       IF2               30          IS-IS            0,500
                                     192.0.46.1       IF1               30          IS-IS            0,500
   192.0.41.0       255.255.255.0    192.0.46.1       IF1               20          IS-IS            0,500
   192.0.39.0       255.255.255.0    192.0.44.1       IF2               20          IS-IS            0,500
   192.0.40.0       255.255.255.0    192.0.44.1       IF2               20          IS-IS            0,500
                                     192.0.46.1       IF1               20          IS-IS            0,500
   192.0.49.0       255.255.255.0    192.0.46.1       IF1               20          IS-IS            0,500
   192.0.50.0       255.255.255.0    192.0.46.1       IF1               20          IS-IS            0,500
   192.0.48.0       255.255.255.0    192.0.46.1       IF1               10          IS-IS            0,500
   192.0.42.0       255.255.255.0    192.0.44.1       IF2               10          IS-IS            0,500
   0.0.0.0          0.0.0.0          192.0.44.1       IF2               20          IS-IS            0,500
                                     192.0.46.1       IF1               20          IS-IS            0,500
```

Obr. 7.13: Změna směrovací tabulky pro Poprad_rtr

9) Pomocí cesty *„Protocols → IP → Demands → Display Routes for Configured Demands...“,* a postupným proklikáním, si zobrazíme, kudy jde přenos. Zde je vidět, že přenos mezi Poprad_rtr a Katowice_rtr musí jít přes hraniční směrovače obou oblastí, viz obr. 7.14.



Obr. 7.14: Cesta z Popradu do Katowic přes hraniční směrovače obou oblastí

7.5 Sloučení skupin podle NET

1) Scénář si duplikujeme, *„Scenarios → Duplicate Scenario…“*. Pojemnujeme ho **IS_IS_slouceni**.
2) Protože je zbytečné vést přenos přes hraniční směrovače, když jsou směrovače spojeny přímou linkou, můžeme přidat NET adresu cizí oblasti do směrovače. Otevřeme si atributy směrovače Žilina_rtr *„Edit atributes“*. Dále *„IS-IS Parameters → Processes → row 0 → Process Parameters → Network Entity Titles“* a přidáme jeden řádek. Zde vložíme ID sousední oblasti, **49.0002.1920.0005.4001.00**.
3) Zkopírujeme ip_traffic_flow a spojíme cestu z Katowice_rtr do Zilina_rtr.
4) Spustíme simulaci *„Scenarios → Manage Scenarios…“*, zobrazí se nám okno, kde v kolonce *„Results“* pro scénář **IS_IS_slouceni** zvolíme **<collect>**. Potvrdíme **Ok** a po proběhnutí okno zavřeme.
5) Opět si zobrazíme cestu přenosu, *„Protocols → IP → Demands → Display Routes for Configured Demands…“*, kde můžeme vidět, že přenos jde přímo, viz obr. 7.15.

Path details (node, link pair) between the following nodes:

Source: Choose From Maps Network.Katowice_rtr

Destination: Choose From Maps Network.Zilina_rtr

Number of Paths: 1

Path #1

Katowice_rtr --> Zilina_rtr Choose From Maps Network.Katowice_rtr
Choose From Maps Network.Zilina_rtr <-> Katowice_rtr Choose From Maps Network.Zilina_rtr

Obr. 7.15: Stejná cesta po sloučení oblastí

7.6 Nastavení výpadků při přenosu

1) Scénář si duplikujeme, *„Scenarios → Duplicate Scenario…“*. Pojmenujeme ho **IS_IS_vypadek**.
2) Zkopírujeme si objekt *„ip_traffic_flow“* mezi uzly Poprad_rtr a Katowice_rtr a propojíme si uzly **Liberec_rtr** a **Karlovy_Vary_rtr**.
3) Zobrazíme se paletu objektů  . Z vyskakovací lišty vybereme *„Utilities“*, kde najdeme objekt **Failure Recovery** a vložíme ho na plochu.
4) Otevřeme si jeho atributy. Dále *„Node Failure/Recovery Specification“*, zde přidáme dva řádky. V tomto objektu můžeme nastavit vypadnutí a následné nahození linky nebo uzlu, přesněji na směrovači Drazdany_rtr. Budeme simulovat rozložení přenosu na linkách při těchto událostech. Nastavení provedeme podle obr. 7.16.

Attribute	Value
name	fail_rec
model	Failure Recovery
Failure/Recovery Modeling	Enabled
Link Failure/Recovery Specifica...	(...)
Link Failure/Recovery Specifica...	NOT_USED
Node Failure Mode	Node only
Node Failure/Recovery Specific...	(...)
rows	2
row 0	
Name	Choose From Maps Network.Drazdany_rtr
Time	1 000
Status	Fail
row 1	
Name	Choose From Maps Network.Drazdany_rtr
Time	2 000
Status	Recover

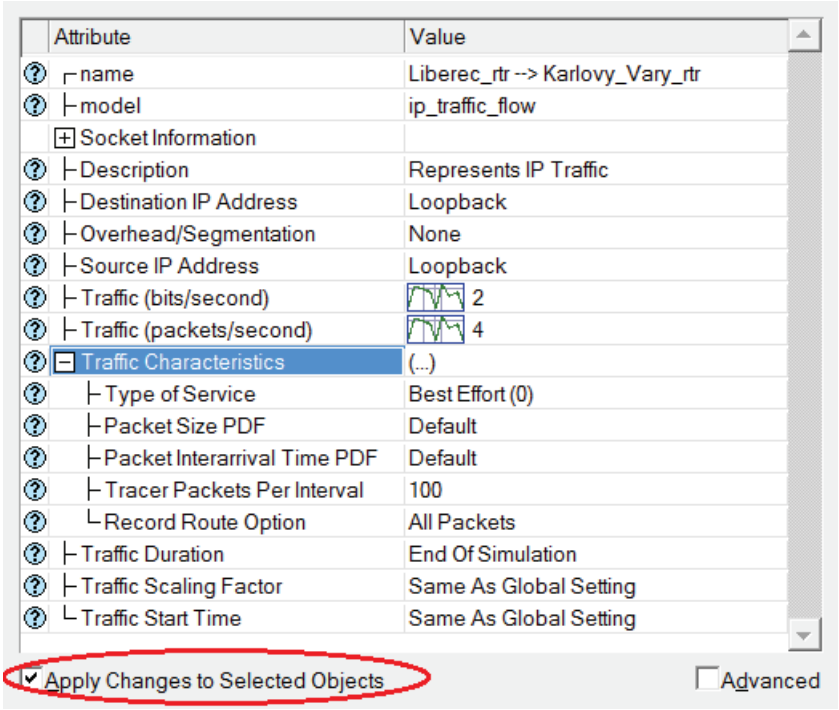
Obr. 7.16: Nastavení výpadku a nahození uzlu Drazdany_rtr

5) Zůstaneme v atributech Failure Recovery. V kolonce *„Node Failure/Recovery Specification“* přidáme opět dva řádky. Stejným způsobem nastavíme výpadek linek mezi **Poprad_rtr** a **Katowice_rtr** a **Zilina_rtr** a **Katowice_rtr**. Podle obr. 7.17.

Attribute	Value
name	fail_rec
model	Failure Recovery
Failure/Recovery Modeling	Enabled
Link Failure/Recovery Specifica...	(...)
rows	2
row 0	
Name	Choose From Maps Network.Katowice_rtr <-> Poprad_rtr
Time	500
Status	Fail
row 1	
Name	Choose From Maps Network.Zilina_rtr <-> Katowice_rtr
Time	1 000
Status	Fail
Link Failure/Recovery Specifica...	NOT_USED
Node Failure Mode	Node only
Node Failure/Recovery Specific...	(...)

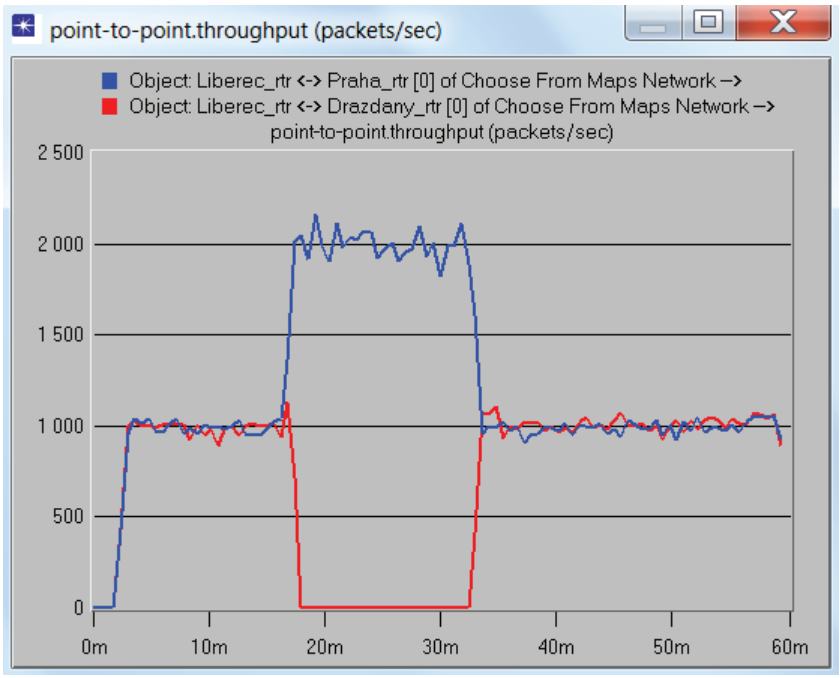
Obr. 7.17: Výpadek linek mezi Poprad_rtr, Katowice_rtr a Zilina_rtr

6) Pro měření upravíme všechny objekty ip_traffic_flow. Jeden označíme, pravým tlačítkem klikneme a v menu vybereme **Select Similar Demands**. Označily se nám všechny 3 linky. Nyní si zobrazíme jejich atributy, *„Edit Atributes“*. V řádku *„Traffic Characteristics → Tracer Packets Per Interval“* nastavíme hodnotu na **100** a v kolonce *„Traffic Characteristics → Record Route Option“* **All packets.** Pro uložení změn na všech linkách nezapomeneme zatrhnout v levém dolním rohu **Apply Changes to Selected Objects,** viz obr. 7.18**.** To nám zajistí výpočet cesty vícekrát při simulaci a my budeme moci sledovat změny.



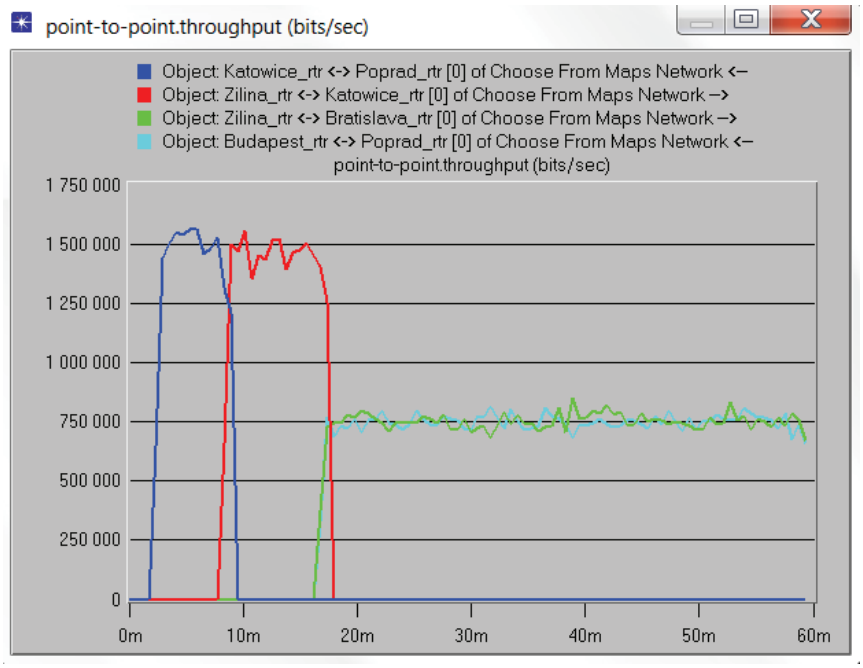
Obr. 7.18: Nastavení Objektu ip_traffic _flow

7) Podobně jako v předchozí části, vložíme číslo NET, **49.0003.1920.0005.4001.00**, do Katowice_rtr, *„IS-IS Parameters → Processes → row 0 → Process Parameters → Network Entity Titles“*.
8) Nyní si vybereme statistiky, které budeme sledovat. Pravým tlačítkem klikneme na plochu a vybereme **Choose Individual Characteristics**. V okně vybereme *„Link Statistics*“, kde označíme **point-to-point** a potvrdíme OK.
9) Spustíme simulaci. Na hlavní liště *„Scenarios → Manage scenarios…“* a spustíme scénář **IS_IS_vypadek**. Po dokončení okno zavřeme.
10) Zobrazíme si výsledky. Pravým tlačítkem klikneme na plochu a vybereme **View Results**. V pravé části místo *„Stacked statistics“* vybereme **Overlaid statistics**. V nabídce v levé části si najdeme linky *„Liberec_rtr ↔ Drazdany_rtr“* a *„Liberec_rtr ↔ Praha_rtr“*. Obě si otevřeme a z části *„point-to-point“* vybereme **throughput(packets/sec)** →. Na pravé straně se zobrazí průběhy. Plně si je zobrazíme tlačítkem **Show**. Z grafu je patrné, že před vypadnutím směrovače Drazdany_rtr je provoz rozložen na obě linky. Po vypadnutí jde celý přenos přes uzel Praha_rtr a po zpětném nahození je přenos opět rozložen stejně jako před výpadkem, viz obr. 7.19.



Obr. 7.19: Propustnost v bit/s na dvou linkách ze směrovače Liberec_rtr

11) Dále si zobrazíme změny přenosu kvůli vypadnutým linkám mezi uzly Poprad_rtr ↔ Katowice_rtr a Zilina_rtr ↔ Katowice_rtr, *„Protocols → IP → Demands → Display Routes for Configured Demands…“*. Před výpadkem jde přenos přímo mezi Popradem a Katowicemi. Po výpadku linky mezi těmito směrovači přenos probíhá přes uzel Zilina_rtr. Po vypadnutí linky mezi Žilinou a Katowicemi protokol IS-IS najde cestu opět přes hraniční směrovač Bratislava_rtr, viz obr. 7.20.



Obr. 7.20: Provoz mezi dvěma oblastmi po vypadnutí linek

7.7 Samostatný úkol

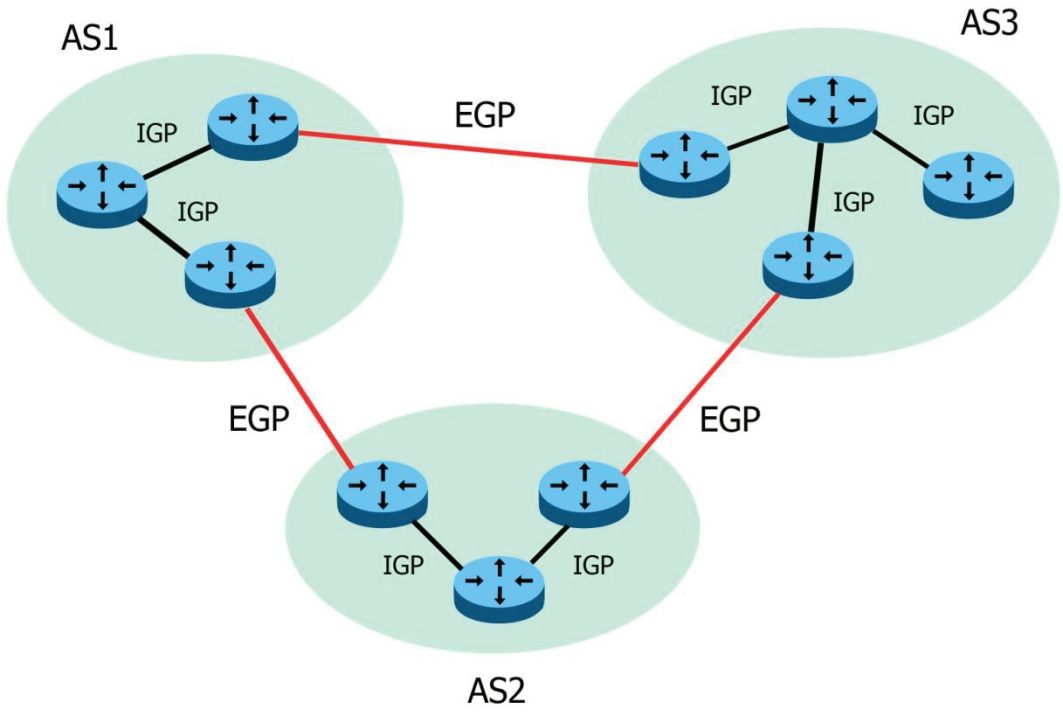
1. Duplikujte si scénář IS_IS_skupiny. Nazvěte si ho **IS_IS_490004_ukol**. V novém scénáři vytvořte novou skupinu podle NET ID, oblast bude mít označení 49.0004. V této skupině budou uzly Berlin_rtr, Drazdany_rtr, Lipsko_rtr a Karlovy Vary_rtr. V nové oblasti bude Drazdany_rtr hraniční směrovač (Level 1-2), ostatní budou pouze lokální směrovače (Level 1). Proveďte simulaci mezi lokálními směrovači mezi dvěmi oblastmi, například mezi Berlin_rtr a Poznan_rtr . Ověřte, že přenos jde přes hraniční směrovače.
2. Opět si duplikujte scénář IS_IS_490004_ukol, nový si nazvěte ho **IS_IS_slouceni_ukol**. Proveďte sloučení nové oblasti s oblastí 49.0002, mezi Berlin_rtr a Poznan_rtr. Opakujte přenos a ověřte, jestli se přenos změnil a prochází nyní mezi těmito uzly přímo.
3. Své výsledky konzultujte s učitelem.

8 Směrovací protokol BGP

8.1 Úvod k laboratorní úloze

Směrování je bezesporu jedna z nejdůležitějších operací při komunikaci mezi dvěma vzdálenými uživateli v počítačové síti. Přenos probíhající přes mnoho síťových prvků by byl bez něho ztracen.

Prakticky můžeme směrovací protokoly rozdělit do dvou skupin. První skupinou jsou EGP (Exterior Gateway Protocols) neboli vnější směrovací protokoly, mezi které patří BGP (Border Gateway Protokol), kterým se budeme zabývat. Druhou skupinou jsou IGP (Interior Gateway Protocols) neboli vnitřní směrovače, mezi které patří RIP (Routing Information Protocol), OSPF (Open Shortest Path First), IGRP (Interior Gateway Routing Protocol), EIRGP (Enhanced Interior Gateway Routing Protocol) a IS-IS (Intermediate System to Intermediate System). Vnitřní směrovací protokoly jsou používány uvnitř autonomních systémů, naopak vnější tyto autonomní systémy spojují, viz obr. 8.1.

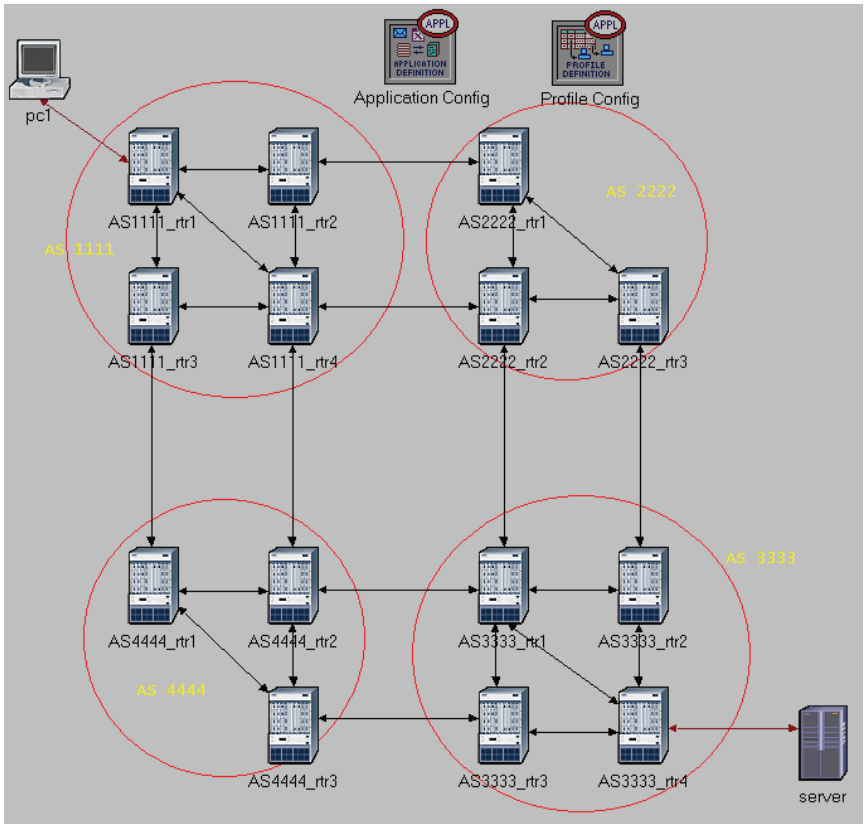


Obr. 8.1: Ilustrativní ukázka směrovacích protokolů v a mezi AS

Protokol BGP se používá pro spojování autonomních systémů, koncový uzel s ním nepřijde do styku. BGP je směrovací protokol typu distance vector, informace o své tabulce rozesílá svým sousedům. Spojení mezi sousedy musí být nakonfigurováno ručně. Pro přenos se používá protokol TCP (port 179). Hraniční směrovače se navzájem informují, k jakým sítím mají přístup, z této informace a pak hraniční směrovač rozhodne kudy přenos směrovat. BGP lze nakonfigurovat i uvnitř AS, internal BGP (iBGP). BGP má výbornou škálovatelnost, malou zátěž sítě, ale oproti tomu má velkou zátěž na CPU a složitou konfiguraci.

8.2 Počáteční nastavení

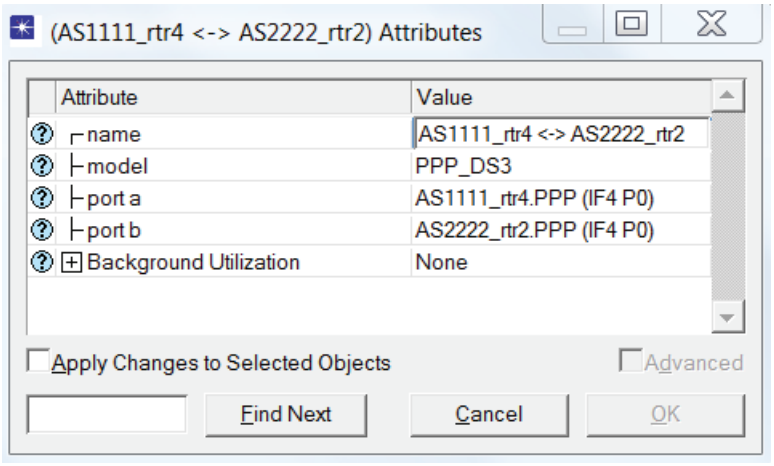
1) Spustíme program OPNET IT GURU Academic Edition.
2) Z nabídky zvolíme **File/Open...** (Ctrl+O), v otevřeném okně najdeme projekt *„pripravena_sit_BGP“* a potvrdíme **OK**. Zobrazí se okno s připravenou topologií, viz obr. 8.2.



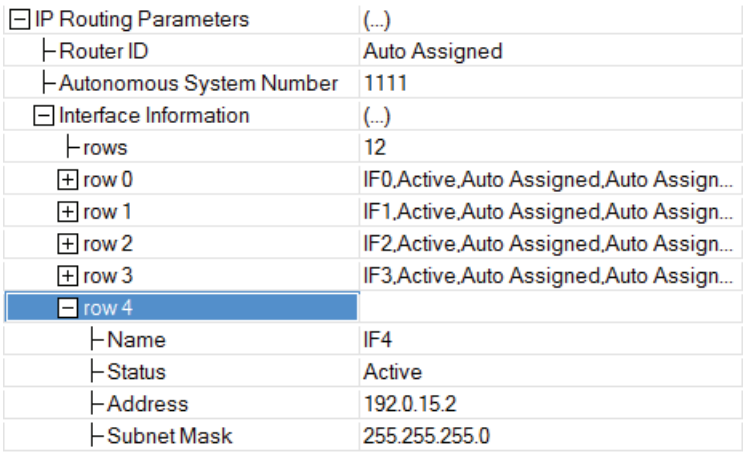
Obr. 8.2: Připravená síť pro laboratorní úlohu

3) Abychom postupným ukládáním nepřepsali připravenou síť, uložíme si projekt pod jiným názvem. *„File → Save as“* a pojmenujeme ho **lab_BGP**.

4) Pro komunikaci v síti je nutné přiřadit každému z uzlů jedinečnou adresu v rámci celé sítě. V připravené síti je toto už téměř hotové. Nakonfigurujeme si tedy jen jeden uzel.
5) Jediný uzel, který není nakonfigurován je uzel **AS1111_rtr4**. Zobrazíme si tedy jeho atributy. Pravým tlačítkem klikneme na směrovač a vybereme *„Edit Attributes“*. Dále na položku *„IP Routing parameters → Interface Information“*. Zde vybereme rozhraní, které chceme nastavit. Která linka je přiřazena, ke kterému portu zjistíme tak, že pravým tlačítkem klikneme na určitou linku a zobrazíme si její atributy, jako například na obr. 8.3. Tady je vidět, že linka směřující ke směrovači AS2222_rtr2 je na směrovači AS1111_rtr4 připojená do portu IF4. Přesuneme se k oknu směrovače AS1111_rtr4 a rozklikneme *„row 4“*, kde změníme položku *„Address“* na **192.0.15.2** a položku *„Subnet Mask“* na **255.255.255.0**, viz obr. 8.4. Další rozhraní a loopback nastavíme stejným způsobem pomocí tab. 8.1. Konfigurace pro loopback není v záložce *„Interface Information“*, ale v *„Loopback Interfaces“*.



Obr. 8.3: Atributy linky mezi směrovači AS1111_rtr4 a AS2222_rtr2



Obr. 8.4: Nastavení adresy a masky pro rozhraní IF4 pro směrovač AS1111_rtr4

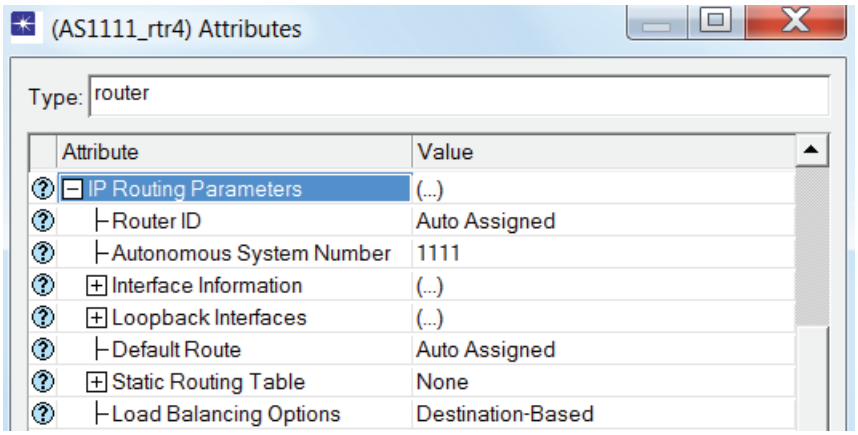
Tab. 8.1. Nastavení IP adres pro rozhraní pro směrovač Praha_rtr

Rozhraní	IP adresa	Maska sítě
IF4	192.0.15.2	255.255.255.0
IF5	192.0.34.1	255.255.255.0
IF6	192.0.8.2	255.255.255.0
IF10	192.0.4.2	255.255.255.0
IF11	192.0.9.2	255.255.255.0
Loopback	192.0.5.1	255.255.255.0

6) Nyní máme nastavené všechny adresy. Komunikace by nám měla fungovat po celé sítí. Dále si nastavíme BGP parametry.

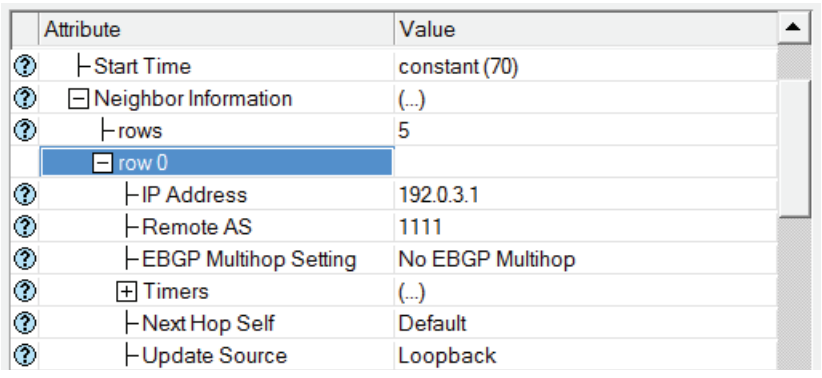
8.3 Konfigurace směrovacího protokolu BGP

1) Pro směrování pomocí BGP protokolu je nutné nastavit několik parametrů, číslo autonomního systému (AS) a informace o sousedních směrovačích (Neighbor Information). Oba parametry je nutné nastavit na všech směrovačích. Protože by tato konfigurace zabrala mnoho času, jsou opět parametry nastaveny na všech směrovačích, kromě směrovače **AS1111_rtr4**.
3) V naší síti jsou vytvořeny 4 autonomní systémy. Jak už napovídají názvy směrovačů, jsou to **AS1111**, **AS2222**, **AS3333** a **AS4444**. Tím je síť rozdělena na 4 oblasti, mezi kterými bude probíhat směrování pomocí BGP protokolu.
4) Zobrazíme si atributy směrovače AS1111_rtr4. Pravým tlačítkem klikneme na směrovač a vybereme *„Edit Attributes“*. Jako první umístíme směrovač do autonomního systému. V záložce *„IP Routing Parameters → Autonomous System Number“* na místo *„Auto Assigned“* napíšeme **1111**, viz obr. 8.5. Tím jsme přiřadili směrovač do AS.



Obr. 8.5: Nastavení AS na směrovači AS1111_rtr1

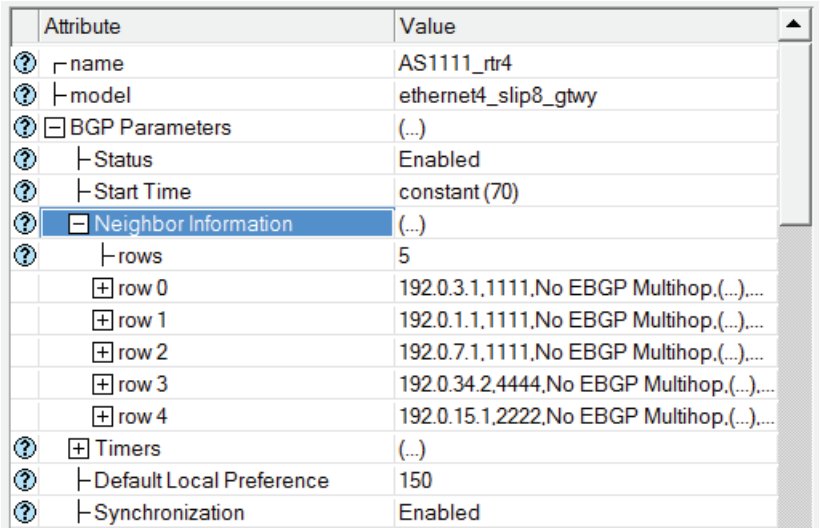
5) Zůstaneme v atributech směrovače a přesuneme se do záložky, kde budeme konfigurovat informace o sousedních, *„BGP Parameters → Neighbor Information“*. AS1111_rtr4 je připojen k 5 sousedům, *„0 rows“* změníme na 5.
6) Při konfigurování sousedů používáme dva atributy, IP adresu souseda a číslo jeho AS. Jakou IP adresu souseda zvolíme, určuje v jakém je sousední směrovač AS vůči konfigurovanému směrovači. Pro sousedy ve stejném AS bude IP adresa souseda jeho loopback. Pro první záznam si otevřeme první řádek „row 0“. Směrovač AS1111_rtr2, který je ve stejném AS, má atributy *„IP Address“* **192.0.3.1**, *„Remote AS“* **1111** a *„Update Source“* **Loopback**, viz obr. 8.6. Atributy pro všechny sousedy nastavíme stejným způsobem podle tab. 8.2. Pro směrovače v jiném AS použijeme IP adresu protilehlého portu, ke kterému je směrovač připojen. Nastavení všech směrovačů sousedících s AS1111_rtr4 jsou vidět na obr. 8.7.



Obr. 8.6: Nastavení Neighbor Information u AS1111_rtr4 pro AS1111_rtr2

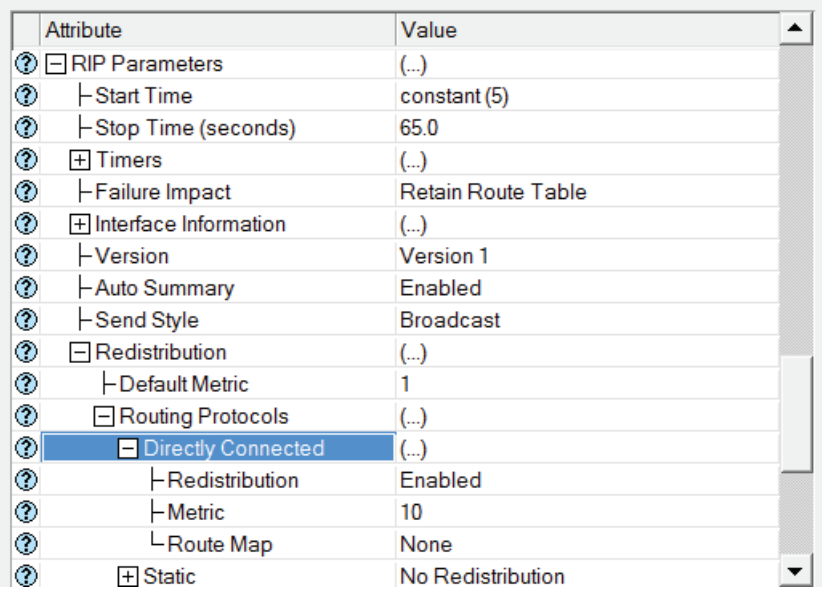
Tab. 8.2: Nastavení Neighbor Information pro směrovač AS1111_rtr4

Směrovač	IP address	Remote AS	Update Source
AS1111_rtr1	192.0.1.1	1111	loopback
AS1111_rtr2	192.0.3.1	1111	loopback
AS1111_rtr3	192.0.7.1	1111	loopback
AS2222_rtr2	192.0.15.1	2222	-
AS4444_rtr2	192.0.34.2	4444	-



Obr. 8.7: Nastavení všech sousedů pro směrovač AS1111_rtr4

7) Protože uvnitř autonomního systému budeme směrovat provoz pomocí protokolu RIP, musíme povolit redistribuci. Opět v atributech AS1111_rtr4 si najdeme *„RIP Parameters → Redistribution → Routing Protocols → Directly Connected“* a povolíme *„Redistribution“* na **Enable** a *„Metric“* na **10**, viz obr. 8.8.



Obr. 8.8: Redistribuce RIP parametru pro směrování uvnitř AS

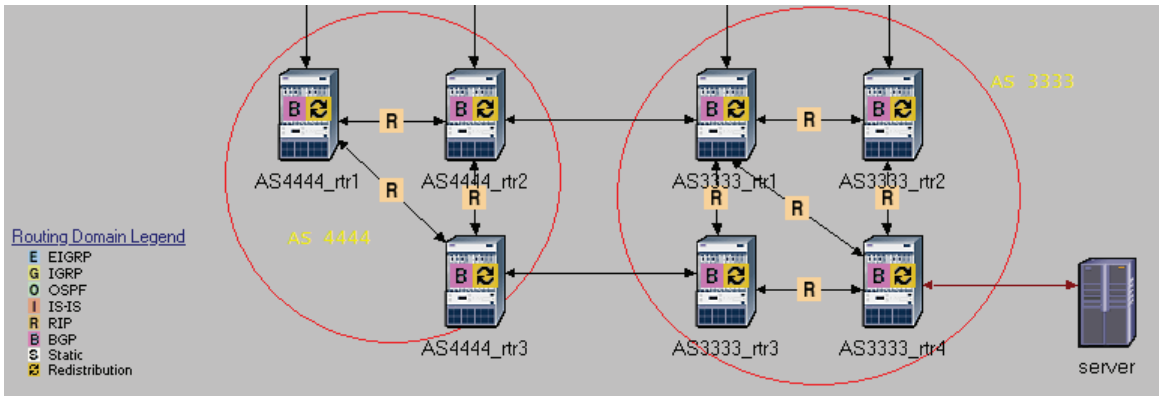
8) Nyní máme nastavené vše potřebné pro směrování pomocí BGP. Jako kontrolu, že jsme nastavení provedli správně, si můžeme zobrazit směrovací tabulku. Označíme uzel **AS1111_rtr4**. Na hlavní liště zvolíme *„Protocols → IP → Routing → Export routing table for selected routers“*. Spustíme si simulaci, tím že na hlavní liště zvolíme *„Scenarios → Manage Scenarios...“*. Zobrazí se nám okno, kde v kolonce *„Results“* zvolíme **<collect>**. Potvrdíme **Ok**. Tím spustíme simulaci, po skončení okno zavřeme. Dále na hlavní liště klikneme na *„Results → Open simulation log“* V otevřeném okně ve sloupci *„Node“* nalezneme uzel **AS1111_rtr4** a kliknutím ve sloupci *„Message“* na **COMMON ROUTE TABLE** se nám zobrazí směrovací tabulka, viz obr. 8.9. Kde si můžeme ověřit, přes které porty a jakými směrovacími protokoly se naučil všechny routy.

```
COMMON ROUTE TABLE snapshot for:

  Router name: Enterprise Network.AS1111_rtr4
      at time: 10800,00 seconds

ROUTE TABLE contents:

   Dest. Address     Subnet Mask       Next Hop       Interface Name    Metric      Protocol    Insertion Time
   -------------  ---------------  ---------------  ---------------   ------      --------    --------------

  192.0.8.0        255.255.255.0    192.0.8.2        IF6               0           Direct           0,000
  192.0.4.0        255.255.255.0    192.0.4.2        IF10              0           Direct           0,000
  192.0.9.0        255.255.255.0    192.0.9.2        IF11              0           Direct           0,000
  192.0.5.0        255.255.255.0    192.0.5.1        Loopback          0           Direct           0,000
  192.0.15.0       255.255.255.0    192.0.15.2       IF4               0           Direct           0,000
  192.0.34.0       255.255.255.0    192.0.34.1       IF5               0           Direct           0,000
  192.0.2.0        255.255.255.0    192.0.4.1        IF10              1           RIP              5,001
  192.0.3.0        255.255.255.0    192.0.4.1        IF10              1           RIP              5,001
  192.0.10.0       255.255.255.0    192.0.4.1        IF10              11          RIP              5,001
  192.0.6.0        255.255.255.0    192.0.9.1        IF11              1           RIP              5,001
  192.0.7.0        255.255.255.0    192.0.9.1        IF11              1           RIP              5,001
  192.0.37.0       255.255.255.0    192.0.9.1        IF11              11          RIP              5,001
  192.0.0.0        255.255.255.0    192.0.8.1        IF6               1           RIP              5,002
  192.0.1.0        255.255.255.0    192.0.8.1        IF6               1           RIP              5,002
  192.0.17.0       255.255.255.0    192.0.15.1       IF4               10          BGP             70,006
  192.0.14.0       255.255.255.0    192.0.15.1       IF4               10          BGP             70,006
  192.0.16.0       255.255.255.0    192.0.15.1       IF4               10          BGP             70,006
  192.0.11.0       255.255.255.0    192.0.15.1       IF4               10          BGP             70,006
  192.0.12.0       255.255.255.0    192.0.15.1       IF4               10          BGP             70,006
  192.0.13.0       255.255.255.0    192.0.15.1       IF4               10          BGP             70,006
  192.0.19.0       255.255.255.0    192.0.15.1       IF4               10          BGP             70,006
  192.0.35.0       255.255.255.0    192.0.34.2       IF5               10          BGP             70,008
  192.0.31.0       255.255.255.0    192.0.34.2       IF5               10          BGP             70,008
  192.0.33.0       255.255.255.0    192.0.34.2       IF5               10          BGP             70,008
  192.0.29.0       255.255.255.0    192.0.34.2       IF5               10          BGP             70,008
  192.0.32.0       255.255.255.0    192.0.34.2       IF5               10          BGP             70,008
  192.0.36.0       255.255.255.0    192.0.34.2       IF5               10          BGP             70,008
  192.0.38.0       255.255.255.0    192.0.34.2       IF5               10          BGP             70,008
  192.0.30.0       255.255.255.0    192.0.34.2       IF5               10          BGP             70,010
  192.0.28.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,010
  192.0.24.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,010
  192.0.20.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,010
  192.0.22.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,010
  192.0.21.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,010
  192.0.23.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,010
  192.0.26.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,011
  192.0.27.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,011
  192.0.25.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,011
  192.0.39.0       255.255.255.0    192.0.15.1       IF4               0           BGP             70,011
  192.0.18.0       255.255.255.0    192.0.15.1       IF4               10          BGP             70,012
```

Obr. 8.9: Směrovací tabulka uzlu AS1111_rtr4

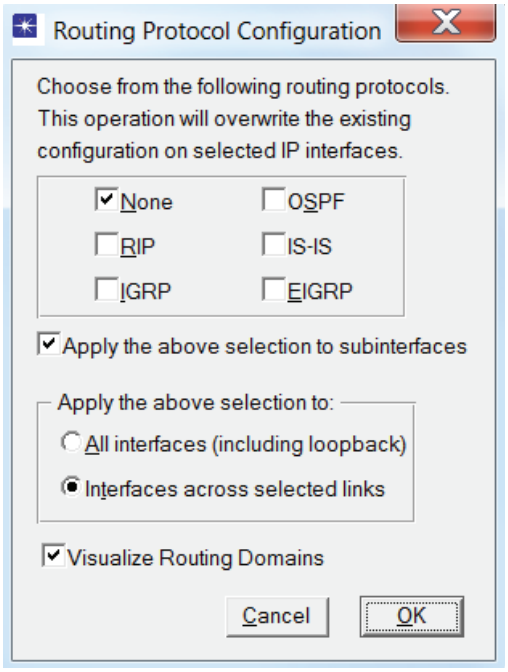
8.4 Internal Border Gateway Protocol - iBGP

1) Směrovací BGP protokol se většinou používá pro směrování ve velkých sítí pro spojení autonomních systémů, ve kterých se většinou o směrování starají směrovací protokoly IGP, např. RIP, OSPF a další. Protokol BGP jde ovšem použít i pro toto směrování, poté mu říkáme **internal BGP** (iBGP).
2) Nejdříve si scénář duplikujeme, na hlavní liště *„Scenarios → Duplicate Scenario..“*. Objeví se okno, ve kterém nový scénář pojmenujeme **iBGP** a potvrdíme **OK**.
3) Zobrazíme si směrovací domény v naší síti, na hlavní liště klikneme na *„Protocols“* a pak dále *„IP → Routing → Visualize Routing Domains“* nebo klávesovou zkratkou Ctrl + Shift + V, viz obr. 8.10.



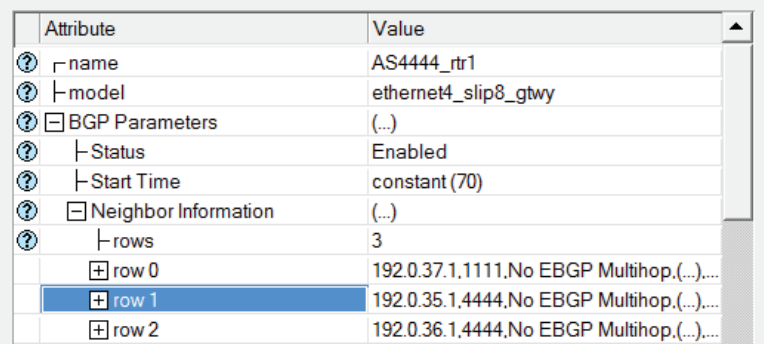
Obr. 8.10: Zobrazení směrovacích domén

4) Směrování iBGP budeme nastavovat pro AS4444. Nejdříve označíme všechny tři linky, které spojují AS4444_rtrx směrovače. Na těchto linkách deaktivujeme protokol RIP. Na hlavní liště klikneme na *„Protocols“* a pak dále *„IP → Routing → Configure Routing Protocols..“*. Objeví se okno, kde zatrhneme **None**, viz obr. 8.11, a potvrdíme.



Obr. 8.11: Změnění Směrovací protokolu na linkách v AS4444

5) Dále změníme informace o sousedních směrovačích. Na směrovači AS4444_rtr1 si zobrazíme atributy, pravým tlačítkem a **Edit Atributes**. Dále *„BGP Parameters → Neighbor Information“*. Pro sousedy v AS4444 změníme IP adresy, v *„row 1“* změníme adresu z 192.0.33.1 na **192.0.35.1** a v *„row 2“* změníme adresu 192.0.32.1 na **192.0.36.1**, a v obou změníme *„Update Source“* na **Not Used**, viz obr. 8.12.



Obr. 8.12: Změna IP adres pro sousední směrovače v AS4444 pro AS4444_rtr1

6) Pro další dva směrovače, AS4444_rtr2 a AS4444_rtr3, provedeme stejné změny podle tab. 8.3. Nezapomeneme opět změnit parametr *„Update Source“* na **Not Used**.

Tab. 8.3: Změna IP adres pro uzly AS4444_rtr2 a AS4444_rtr3

Směrovač	**Původní IP adresa**	**Nová IP adresa**
AS4444_rtr2	192.0.38.1	192.0.35.2
AS4444_rtr2	192.0.32.1	192.0.31.2
AS4444_rtr3	192.0.33.1	192.0.31.1
AS4444_rtr3	192.0.38.1	192.0.36.2

7) Nyní máme vše potřebné zajištěno. Stejným způsobem jako v kapitole 2 a části 9, si spustíme simulaci, *„Scenarios → Manage Scenarios...“*. Po proběhnutí si zobrazíme směrovací tabulku směrovače AS4444_rtr2, *„Results → Open simulation log“*, viz obr. 8.13. Zde vidíme, že všechny routy jsou naučeny pomocí BGP protokolu.

COMMON ROUTE TABLE snapshot for:

Router name: Enterprise Network.AS4444_rtr2
at time: 10800,00 seconds

ROUTE TABLE contents:

Dest. Address	Subnet Mask	Next Hop	Interface Name	Metric	Protocol	Insertion Time
192.0.33.0	255.255.255.0	192.0.33.1	Loopback	0	Direct	0,000
192.0.35.0	255.255.255.0	192.0.35.1	IF4	0	Direct	0,000
192.0.31.0	255.255.255.0	192.0.31.1	IF5	0	Direct	0,000
192.0.34.0	255.255.255.0	192.0.34.2	IF10	0	Direct	0,000
192.0.30.0	255.255.255.0	192.0.30.2	IF11	0	Direct	0,000
192.0.28.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,006
192.0.24.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,006
192.0.20.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.22.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.19.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.21.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.23.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.26.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.27.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.29.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.25.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.39.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,007
192.0.8.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.4.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.9.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.5.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.2.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.3.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.10.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.6.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.7.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.37.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.0.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.1.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,009
192.0.17.0	255.255.255.0	192.0.34.1	IF10	0	BGP	70,009
192.0.14.0	255.255.255.0	192.0.34.1	IF10	0	BGP	70,009
192.0.16.0	255.255.255.0	192.0.34.1	IF10	0	BGP	70,009
192.0.11.0	255.255.255.0	192.0.34.1	IF10	0	BGP	70,009
192.0.12.0	255.255.255.0	192.0.34.1	IF10	0	BGP	70,009
192.0.13.0	255.255.255.0	192.0.34.1	IF10	0	BGP	70,010
192.0.18.0	255.255.255.0	192.0.30.1	IF11	10	BGP	70,013
192.0.15.0	255.255.255.0	192.0.34.1	IF10	10	BGP	70,015

Obr. 8.13: Směrovací tabulka AS4444_rtr2 naučená pouze protokolem BGP

8.5 Směrovací parametry BGP

BGP protokol definuje zprávu NLRI (Network Layer Reachability Information Update), která přenáší aktualizace směrovacích informací. Tato zpráva mimo jiné obsahuje pole Atributy routy. Zde jsou definovány atributy, které ovlivňují směrování v síti, například As Path, Next Hop, Local Preference, Multi Exit Discriminator a další. Na ně se nyní podíváme.

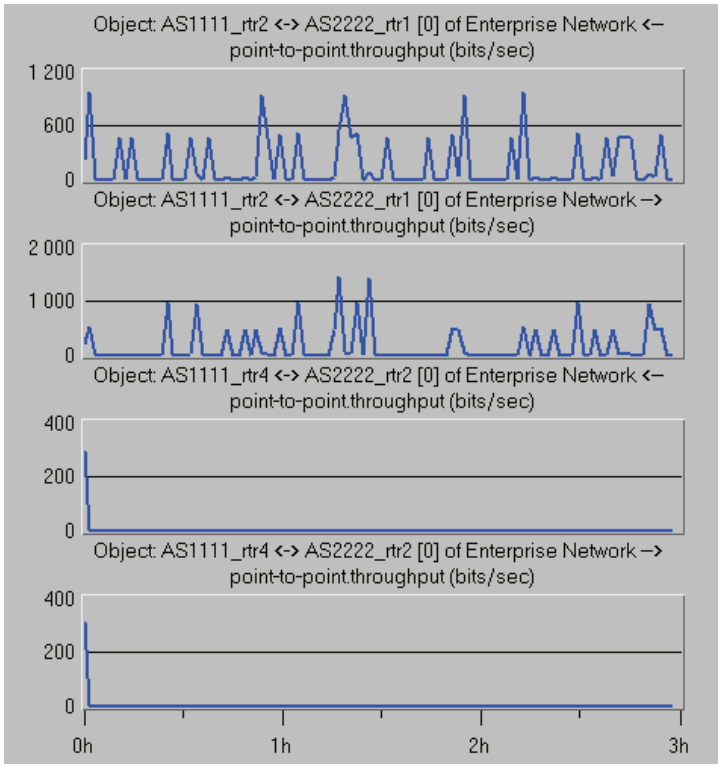
Nastavování se provádí pomocí *„Route Map Configuration“*, které obsahuje položky *„Match Info“*, a *„Set Info“*. Match Info funguje jako podmínka, pokud směrovací protokol nalezne shodu v Match Info, přesune se do Set Info a pokud ne, přeskočí na další Match Info. V Set Info se definuje co se má pro splnění udělat.

8.5.1 Atribut Multi Exit Discriminator (MED)

1) Nejdříve si scénář duplikujeme, na hlavní liště *„Scenarios → Duplicate Scenario..“*. Objeví se okno, ve kterém nový scénář pojmenujeme **BGP_MED** a potvrdíme **OK**.

2) Parametr Multi Exit Dicriminator ovlivňuje provoz směrem do Autonomního systému. Nejdříve si spustíme simulaci, abychom později viděli změnu provozu. V naší síti je vytvořená aplikace **email**, kterou **pc1** i **server** podporují. Nic tedy nastavovat nemusíme. Spustíme simulaci, tím že na hlavní liště zvolíme *„Scenarios → Manage Scenarios…“*, zobrazí se nám okno, kde v kolonce *„Results“* pro scénář **BGP_MED** zvolíme **<collect>**. Potvrdíme **Ok**. Po skončení zavřeme.

3) Nyní pravým tlačítkem klikneme na plochu a vybereme **View Results**. V levém okně je seznam zvolných statistik, vybereme *„Object Statistics → Enterprise Network → AS1111_rtr2 ↔ AS2222rtr1 → point-to-point“* a zde zvolíme **throughput (bits/s)** → a **throughput (bits/s)** ←. Stejné statistiky vybereme i pro linku AS1111_rtr4 ↔ AS2222_rtr2. V pravé části se nám zobrazili průběhy, viz obr. 8.14. Na průbězích můžeme vidět, že provoz jde po lince mezi AS1111_rtr2 a AS2222_rtr1. To nyní pomocí MED změníme.



Obr. 8.14: Propustnost v síti bez použitých pravidel pro směrování

4) Pomocí MED upravíme provoz do sítě. Chceme tedy, aby šel přes AS1111_rtr4. Implementace pravidla je jednoduchá, musíme nejdříve pravidlo vytvořit a poté ho implementovat na směrovač. Pravidla budeme vytvářet pro směrovače AS1111_rtr2 a AS1111_rtr4. Otevřeme si jejich atributy, dále *„IP Routing Parameters → Route Map*

Configuration“, kde změníme *„rows“* na **1**, řádek otevřeme. V řádku *„Map Label“* pojmenujeme konfiguraci **MED**, dále *„Map Configuration → row 0“*. Zde budeme měnit obě záložky, otevřeme si *„Match Info“*, přidáme jeden řádek. Otevřeme si ho a v položce *„Match Value“*, pro oba směrovače, změníme na **192.0.40.1 255.255.255.0**, IP adresa pc1. V *„Match Info“* je to vše, přesuneme se do *„Set Info“*. Zde opět vytvoříme jeden řádek. Hodnoty nastavíme následovně, *„Set Attribute“* na **Multi Exit Discriminator**, *„Set Value“* pro **AS1111_rtr2** na **200** a pro **AS1111_rtr4** bude **50**, viz obr. 8.15. Tím jsme oběma linkám přiřadili rozdílnou metriku, kterou se bude směrovací protokol řídit a rozhodovat kudy bude přenos směrovat.

Attribute	Value
Route Map Configuration	(...)
rows	1
row 0	
Map Label	MED
Map Configuration	(...)
rows	1
row 0	
Term	10
Match Info	(...)
rows	1
row 0	
Match Property	IP Address
Match Condition	Equals
Match Value	192.0.40.1 255.255.255.0
Set Info	(...)
rows	1
row 0	
Set Attribute	Multi Exit Discriminator
Set Operation	Set As [=]
Set Value	200

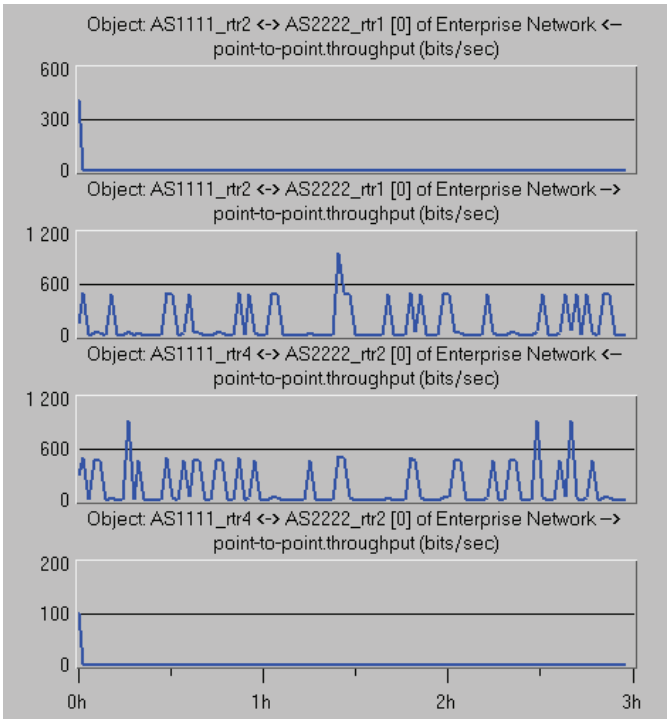
Obr. 8.15: Nastavení parametrů MED pro směrovač AS1111_rtr2

5) Nyní je potřeba povolit směrovací politiku na obou směrovačích. Zůstaneme v jejich atributech a přesuneme se do *„BGP parameters → Neighbor Information“,* kde u obou směrovačů otevřeme řádek souseda v AS 2222. Dále pak v záložce *„Routing Policies“* změníme řádek na **1** a otevřeme si ho. V *„Route Map“* vybereme naše vytvořené pravidlo **MED** a *„Applicable Direction“* změníme na **Out**, viz obr. 8.16.

Attribute	Value
BGP Parameters	(...)
Status	Enabled
Start Time	constant (70)
Neighbor Information	(...)
rows	3
row 0	192.0.1.1,1111,No EBGP Multihop,(...),...
row 1	192.0.5.1,1111,No EBGP Multihop,(...),...
row 2	
IP Address	192.0.10.2
Remote AS	2222
EBGP Multihop Setting	No EBGP Multihop
Timers	(...)
Next Hop Self	Default
Update Source	Not Used
Prefix Limit	No Max Limit
Weight	100
Send-Community	Disabled
Routing Policies	(...)
rows	1
row 0	
Route Map	MED
Applicable Direction	Out

Obr. 8.16: Implementace pravidla MED do BGP parametrů směrovače

6) Nyní opět spustíme simulaci, stejným způsobem jako v bodě 2 a následně si zobrazíme stejné průběhy jako v bodě 3, viz obr. 8.17. Zde můžeme vidět, že po použití **Multi Exit Discriminator** se změnilo směrování ve směru do AS. Nyní jde přes směrovač AS1111_rtr4.



Obr. 8.17: Propustnost v síti s použitým pravidlem MED pro směrování

8.5.2 Atribut Local Preference

1) Scénář si opět duplikujeme, na hlavní liště *„Scenarios → Duplicate Scenario..“*. Objeví se okno, ve kterém nový scénář pojmenujeme **BGP_Local_Preference** a potvrdíme **OK**.
2) Atribut Local Preference je v podstatě stejný jako MED. Jediný rozdíl je, že jím ovlivňujeme provoz z AS ven. Tedy v opačném směru než MED. Nastavení je velmi podobné. Opět budeme chtít přenos vést přes směrovač AS1111_rtr4.
3) Otevřeme si atributy stejných směrovačů jako v předešlém úkolu, dále *„IP Routing Parameters → Route Map Configuration“*, zde přidáme jeden řádek a otevřeme ho. V řádku *„Map Label“* pojmenujeme konfiguraci **Local_preference**, dále *„Map Configuration → row 0“*. Zde budeme nyní měnit záložku *„Set Info“*. Vytvoříme jeden řádek. Hodnoty nastavíme následovně, *„Set Attribute“* na **Local Preference** a *„Set Value“* pro **AS1111_rtr2** na **100** a pro **AS1111_rtr4** bude **500**, viz obr. 8.18. Tím jsme oběma linkám přiřadili rozdílnou hodnotu, kterou se bude směrovací protokol řídit a rozhodovat kudy bude přenos směrovat.

Attribute	Value
⊟ Route Map Configuration	(...)
├ rows	2
⊞ row 0	MED,(...),Not Used
⊟ row 1	
├ Map Label	Local_preference
⊟ Map Configuration	(...)
├ rows	1
⊟ row 0	
├ Term	10
⊞ Match Info	None
⊟ Set Info	(...)
├ rows	1
⊟ row 0	
├ Set Attribute	Local Preference
├ Set Operation	Set As [=]
└ Set Value	100

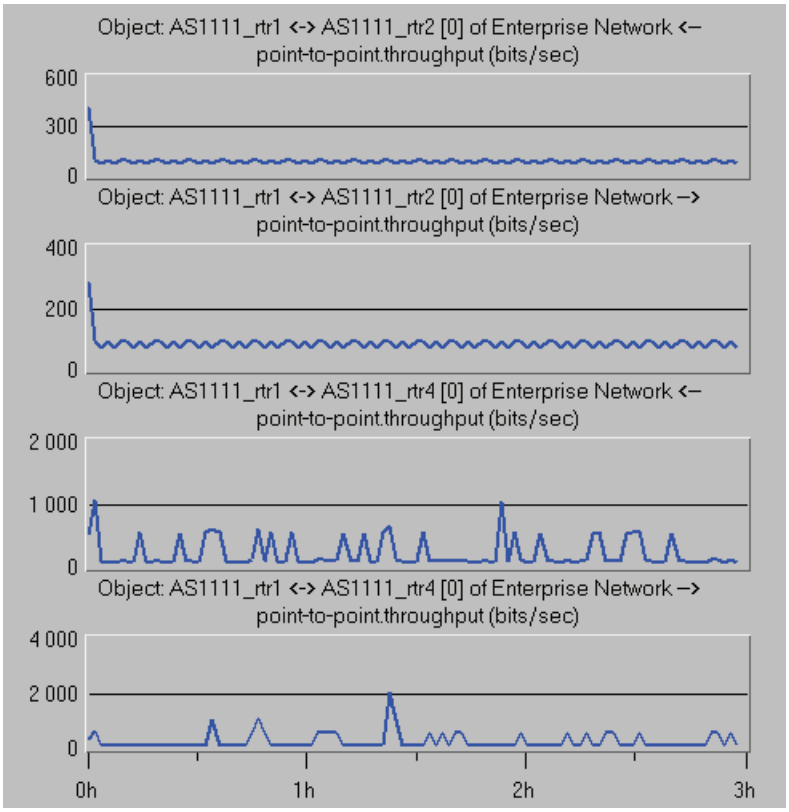
Obr. 8.18: Nastavení parametrů MED pro směrovač AS1111_rtr2

4) Opět povolíme směrovací politiku na směrovačích. *„BGP parameters → Neighbor Information“,* kde u obou směrovačů otevřeme řádek souseda v AS 2222. Dále pak v záložce *„Routing Policies“* přidáme řádek a otevřeme si ho. V *„Route Map“* vybereme naše vytvořené pravidlo **Local_preference** a *„Applicable Direction“* změníme na **In**, viz obr. 8.19.

Attribute	Value
row 2	
IP Address	192.0.10.2
Remote AS	2222
EBGP Multihop Setting	No EBGP Multihop
Timers	(...)
Next Hop Self	Default
Update Source	Not Used
Prefix Limit	No Max Limit
Weight	100
Send-Community	Disabled
Routing Policies	(...)
rows	2
row 0	MED,Out
row 1	
Route Map	Local_preference
Applicable Direction	In

Obr. 8.19: Implementace pravidla Local_preference do BGP parametrů směrovače

5) Nyní opět spustíme simulaci a následně si zobrazíme stejné průběhy, viz obr. 8.20. Zde můžeme vidět, že po použití **Local Preference** se změnilo směrování ve směru z AS. Nyní jde přes směrovač AS1111_rtr4.



Obr. 8.20: Propustnost v síti s použitým pravidlem Local_preference pro směrování

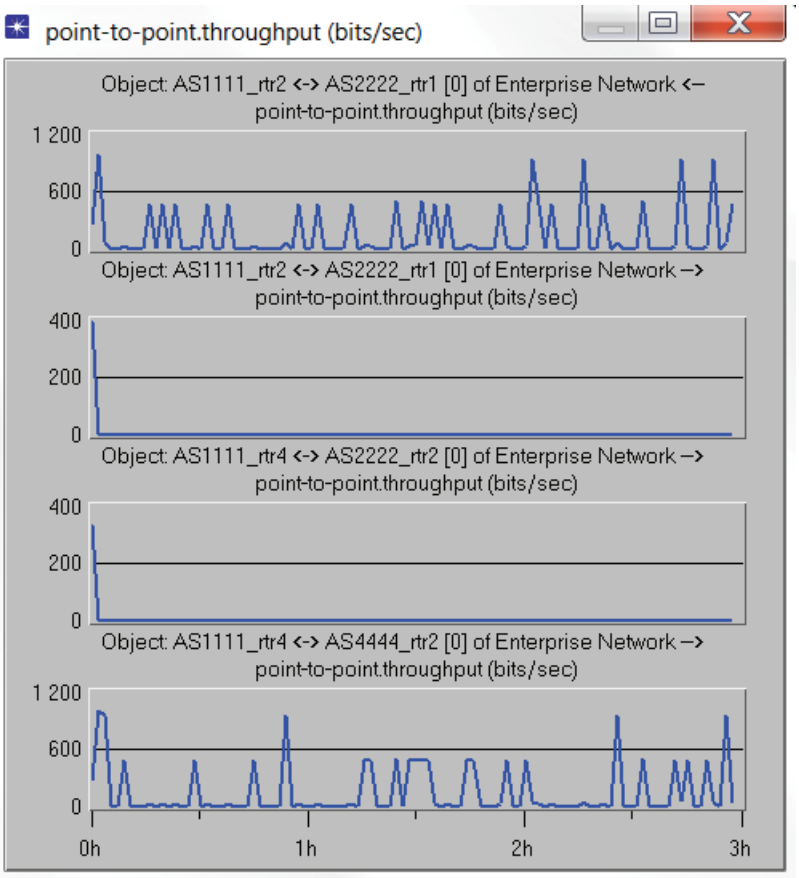
8.5.3 Atribut AS Path a Next Hop

1) Další z atributů je AS Path a Next Hop. AS Path je povinný parametr, který při průchodu zapisuje čísla AS a tím zabraňuje smyčkám v síti. Next Hop je informace dalšího skoku. Nyní si duplikujeme původní scénář bez atributů MED a Local Preference. Přepneme se do scénáře BGP *„Scenarios → Switch to Scenario“* a z nabídky vybereme **BGP**. Poté opět duplikujeme, nový scénář pojmenujeme **BGP_AS_Path_Next_Hop** a potvrdíme **OK**. Nyní jde provoz stejně jako na obr. 8.14.
2) Otevřeme si atributy AS1111_rtr2, proklikáme se do *„Route Map Configuration“*. Nový *„Map Label“* pojmenujeme, výběrem z lišty, na **Route Map 1**. Dále v *„Match Info“* do pole *„Match Property“* vložíme **AS Path**, *„Match Conditions“* **Contains** a do *„Match Value“* **4444**. V záložce *„Set Info“* nastavíme následovně, *„Set Atribute“* na **Local Preference** a *„Set Value“* na **10**.
3) Následně povolíme politiku v nastavení sousedních informací v BGP do AS 2222, *„BGP parameters → Neighbor Information → row 2 → Routing Policies → row 0“*. Kde změníme *„Route Map“* na **Route Map 1** *„Applicable Direction“* ponecháme na **In**. Výsledek je takový, že směrem z AS jde nyní přenos přes AS1111_rtr4.
4) Nyní nastavíme AS1111_rtr4 podle tab. 8.4.

Tab. 8.4: Nastavení politiky směrování na směrovači AS1111_rtr4

IP Routing Parameters → Route Map Configuration → row 0	
Map Label	Route Map 1
Match Info → Match Property	AS Path
Match Info → Match Conditions	Contains
Match Info → Match Value	3333
Set Info → Set Attribute	Next Hop
Set Info → Set Value	192.0.34.2
BGP Parameters → Neighbor Information → soused v AS 2222 → Routing Policies	
Route Map	Route Map 1
Applicable Direction	In

5) Nyní si spustíme simulaci a zobrazíme si průběhy, viz obr. 8.21. Zde můžeme vidět, že jsme donutili přenos z AS 1111, pomocí nejdříve AS Path a později Next Hop, aby šel přes AS 4444.



Obr. 8.21: Průběh přenosu přes směrovač AS4444_rtr2

8.6 Samostatný úkol

1) Nejprve si duplikujte scénář BGP, který si pojmenujte **BGP_ukol**.
2) Zkopírujte pc1 a připojte ho k AS4444_rtr1, zkopírujte i server, který připojte k AS2222_rtr3, původní objekty pc1 a serveru smažte.
3) Nastavte IP adresy na portech směrovačů, ke kterým byly připojeny pc1 a server. Na jakém portu zjistíte z atributů linky.
4) Nasměrovat přenos do a ven z AS přes směrovač AS4444_rtr3. Pomocí atributů Multi Exit Discriminator a Local Preference.

9 Závěr

Výsledkem mé diplomové práce jsou 4 laboratorní úlohy pro výuku počítačových cvičení předmětu Komunikační technologie. Úlohy byly vypracovány v simulačním programu IT Guru Academic Edition 9.1.A od společnosti OPNET Technologies, Inc. Výhodou této verze je bezplatné použití, které student může využít i mimo školní budovu.

Práce je rozdělena do několika částí. Úvod práce je věnován seznámení s programem IT Guru. V další části je teoretické zpracování protokolů, které jsou používány v laboratorních úlohách. Poslední částí jsou vypracované laboratorní úlohy, které jsou navrženy, tak aby studentovi co nejvíce přiblížily popisovanou problematiku a pomohly ji pochopit.

Laboratorní úlohy jsou detailně vypracovány, každý krok je přesně popsán a ilustrován obrázkem. Úlohy jsou logicky rozděleny do několika částí, jako příprava a výstavba sítě, simulování provozu a vyhodnocení výsledků.

První úloha se věnuje seznámení s aplikačními protokoly FTP (File Transfer Protocol), HTTP (HyperText Transfer Protocol) a TELNET (Telecomunication Network). Protokoly jsou porovnávány z hlediska síťových nároků. Druhá úloha se zabývá telefonií VoIP (Voice over Internet Protocol). Jsou zde použity různé typy kodeků a typy signalizací H.323 a SIP (Session Initation Protocol). Třetí úloha je zaměřena na směrovací protokol typu link state, IS-IS, který se používá pro směrování provozu v autonomních systémech. Poslední úloha se zabývá směrovacím protokolem BGP, který se používá pro směrování mezi autonomními systémy. Na konci posledních dvou laboratorních úloh je samostatný úkol, který napomůže k lepšímu pochopení problematiky.

Součástí práce je vzorové řešení samostatných úkolu, které jsou umístěny v příloze. Zdrojové kódy vytvořených programů jsou na přiloženém CD.

Literatura

[1] OPNET Technologies, Inc. *OPNET Technologies: About OPNET* [online]. 2013 [cit. 2013-05-24]. Dostupné z URL: ⟨http://www.opnet.com/corporate/⟩.

[2] OPNET Technologies, Inc. *Network Modelink | Network Simulation* [online]. 2013 [cit. 2013-05-24]. Dostupné z URL: ⟨http://www.opnet.com/solutions/network_rd/modeler.html⟩.

[3] OPNET Technologies, Inc. *OPNET IT Guru Academic Edition* [online]. 2013 [cit. 2013-05-24]. Dostupné z URL: ⟨http://www.opnet.com/university_program/itguru_academic_edition⟩.

[4] Universitat Ramon Llull. *Security Labs in OPNET IT Guru* [online]. 2005-06-18 [cit. 2013-05-24]. Dostupné z URL: ⟨http://www.opnet.com/university_program/teaching_with_opnet/textbooks_and_materials/materials/Security_labs_ITGAE.zip⟩.

[5] KABELOVÁ, Alena, Dostálek, Libor. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5. vydání. Brno: Computer Press, 2008. 488s. ISBN 978-80-251-2236-5.

[6] POSTEL, J., REYNOLDS, J.. *Telnet Protocol Specification*, RFC 854. Internet Engineering Task Force. Květen 1983. 16s. [cit. 19. 3. 2013]. Dostupné z URL: ⟨http://tools.ietf.org/html/rfc854⟩.

[7] PUŽMANOVÁ, Rita. *Moderní komunikační sítě od A do Z*. 2. vydání. Brno: Computer Press, 2006. 432s. ISBN 80-251-1278-0.

[8] ROSENBERG, Jonathan. *SIP: Session Initiation Protocol, RFC 3261. Internet Engineering Task Force*. Červen 2002. 169s. [cit. 17. 3. 2013]. Dostupné z URL: ⟨http://www.ietf.org/rfc/rfc3261.txt⟩.

[9] GRYGAREK, Petr. *Směrovací protokol BGP* [online]. 2009-05-27 [cit. 2013-05-22]. Dostupné z URL: ⟨ http://www.cs.vsb.cz/grygarek/SPS/lect/BGP/BGP.html⟩.

[10] CONLAN, J. Patric. *Cisco Network Professional's Advanced Internetworking Guide*. 1. vydání. Hoboken (USA): Wiley Publishing, 2009. 887s. ISBN 978-0470383605.

[11] JEŘÁBEK, J.. *Pokročilé komunikační techniky (MPKT).* Skripta VUT v Brně. 2013-02-04 [cit. 2013-05-20]. 194s. ISBN 978-80-214-4636-6.

Seznam zkratek

ABR	Area Border Router
AFI	Authority and Format Identifier
AO	Abort Output
ARPANET	Advanced Research Projects Agency Network
AS	Autonomous System
AYT	Are You There
BGP	Border Gateway Protocol
BRK	Break process
CERN	European Organization of Nuclear Research
CPU	Central Processing Unit
CSNP	Complete Sequence Number PDU
DNS	Domain Name System
DSP	Domain Specific Part
EBCDIC	Extended Binary Coded Decimal Interchange Code
eBGP	exterior Border Gateway Protocol
EC	Erase Character
EGP	Exterior Gateway Protocol
EIRGP	Enhanced Interior Gateway Routing Protocol
EL	Erase Line
eMail	electronic Mail
EOF	End of File
FTP	File Transfer Protocol
GSM	Global System for Mobile Communications
HO-DSP	High-Order Domain-Specific Part
HTTP	Hypertext Transfer Protocol
iBGP	internal Border Gateway Protocol
IDI	Initial Domain Identifier
IDP	Initial Domain Part
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol
IMAP	Internet Message Access Protocol

IP	Internet Protocol
IP	Interrupt Process
IPv6	Internet Protocol version 6
IS-IS	Intermediate System to Intermediate Systém
ISP	Internet Service Provider
ITU-T	International Telecommunication Union - Telecommunication
LSA	Link-State Advertisement
LSP	Link State Packet
LSP	Link-State PDU
MAC	Media Access Control
MCU	Multipoint Cotrol Unit
MED	Multi Exit Discriminator
MPLS	Multiprotocol Label Switching
NET	Network Entity Title
NRLI	Network Layer Reachability Information
NSAP	Network Service Access Point
NSEL	Network selector
NVT	Network Virtual Terminal
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PCM	Pulse-Code Modulation
PDU	Protocol Data Unit
POP3	Post Office Protocol
PPP	Point-to-Point Protocol
PSNP	Partial Sequence Number PDU
QoS	Quality of Service
RAS	Registration, Admission, Status
RFC	Request For Comments
RIB	Routing Information Bases
RIP	Routing Information Protocol
rtr	router
SIP	Session Initation Protocol
SMTP	Simple Mail Transfer Protocol

SPF	Shortest Path First
SSH	Secure Shell
TCP	Transmission Control Protocol
Telnet	Telecommunication Network
TLS	Transport Layer Security
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
VLSM	Variable-Length Subnet Mask
VoIP	Voice over Internet Protocol

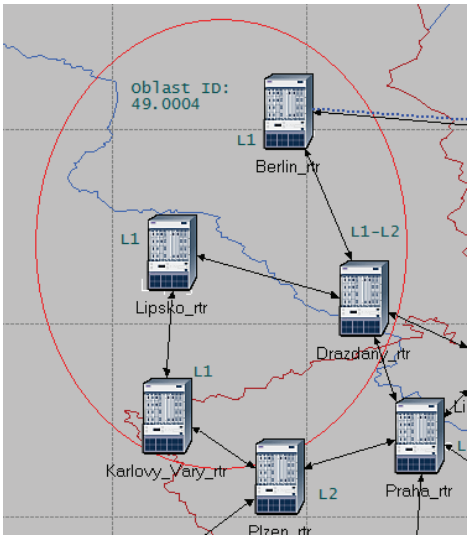
Přílohy

A. Řešení úkolů pro laboratorní úlohu Směrovací protokol IS - IS

Tato laboratorní úloha je zaměřena na představení a pochopení směrovacího protokolu IS-IS. Postupně si student vytvoří síť, nastaví směrovaní pomocí IS-IS. Celou síť rozdělí na oblasti podle úrovně směrovačů, Level 1 místní směrovače, Level 2 páteřní směrovače a Level 1-2 hraniční směrovače. V dalším bodu se sloučí dvě vytvořené oblasti v jednu. V poslední části si ověří chování směrování při výpadku uzlů nebo linek. Následují samostatné úlohy.

A.1 Úkol 1

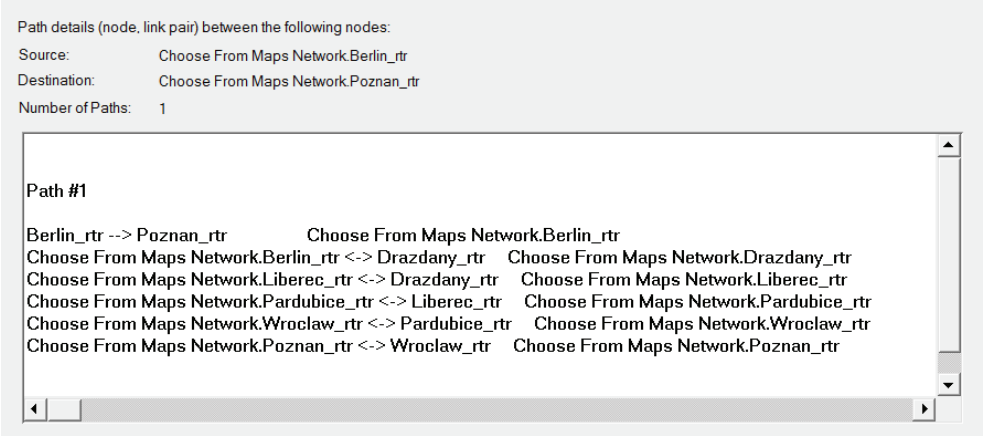
1) V prvním úkolu má student vytvořit další oblast ze směrovačů Berlin_rtr, Drazdany_rtr, Lipsko_rtr a Karlovy Vary_rtr. Network Entity Title bude 49.0004. V této oblasti je jako hraniční směrovač konfigurován uzel Drazdany_rtr (Level 1-2), ostatní směrovače jsou lokální (Level 1), viz obr. A. 1.



Obr. A. 1: Vytvoření nové oblasti podle NET a levelu

2) Pro nastavení je nutné změnit NET v *„IS-IS Parameters → Processes → row 0 → Process Parameters → Network Entity Titles → row 0 → Network Entity Title“* číslo oblasti na 49.0004 ke všem směrovačům. Ve stejné složce je nutné upřesnit level směrovače.

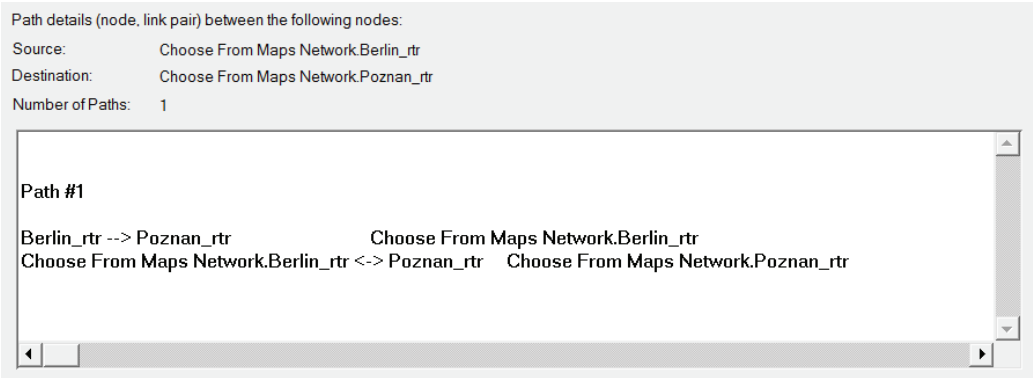
3) Po nastavení těchto parametrů si student zkopíruje objekt ip_trafic_flow a vystaví ho mezi uzly Berlin_rtr a Poznan_rtr a spustí simulaci.
4) Ověří si správné nastavení zobrazením cesty přenosu pomocí *„Protocols → IP → Demands → Display Routes for Configured Demands... → Sources (Choose from Maps Network) → Berlin_rtr → Poznan_rtr → 1“*, v pravé části vybere detaily a zobrazí se mu cesta přenosu, viz obr. A. 2. Místní směrovač Berlin_rtr nezná přímou cestu do uzlu Poznan_rtr, proto musí přenos směrovat přes Drazdany_rtr, který tuto znalost má.



Obr. A. 2: Přenos z Berlin_rtr do Poznan_rtr přes hraniční směrovač Drazdany_rtr

A.2 Úkol 2

1) V druhém úkolu student sloučí dvě oblasti, 49.0002 a 49.0004. Výsledkem je, že přenos nejde přes hraniční směrovače, ale je směrován přímo mezi Berlin_rtr a Poznan_rtr. K tomu je nutné přiřadit do NET směrovače Poznan_rtr záznam o oblasti 49.0004. Přesněji to bude **49.0004.1920.0002.2001.00**.
2) Znovu si spustí simulaci a ověří si, kudy jde přenos, nyní jde přímo, viz obr. A. 3.

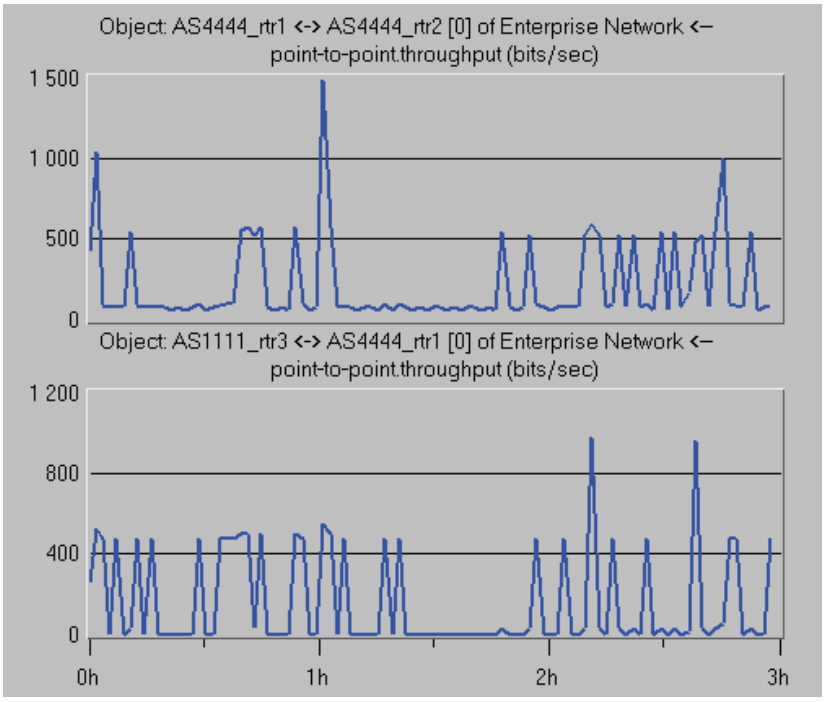


Obr. A. 3: Přímý přenos mezi Berlin_rtr a Poznan_rtr díky sloučení oblastí

B. Řešení úkolů pro laboratorní úlohu Směrovací protokol BGP

Tato úloha je zaměřena na pochopení směrování pomocí směrovacího protokolu. Student si nejdříve otevře připravenou síť. Doplní IP adresy a informace o BGP sousedech. Následují další scénáře. V prvním si v AS 4444 nakonfiguruje iBGP. V dalších konfiguruje atributy BGP protokolu, Multi Exit Discriminator, Local Preference, AS Path a Next Hop. Následuje samostatný úkol.

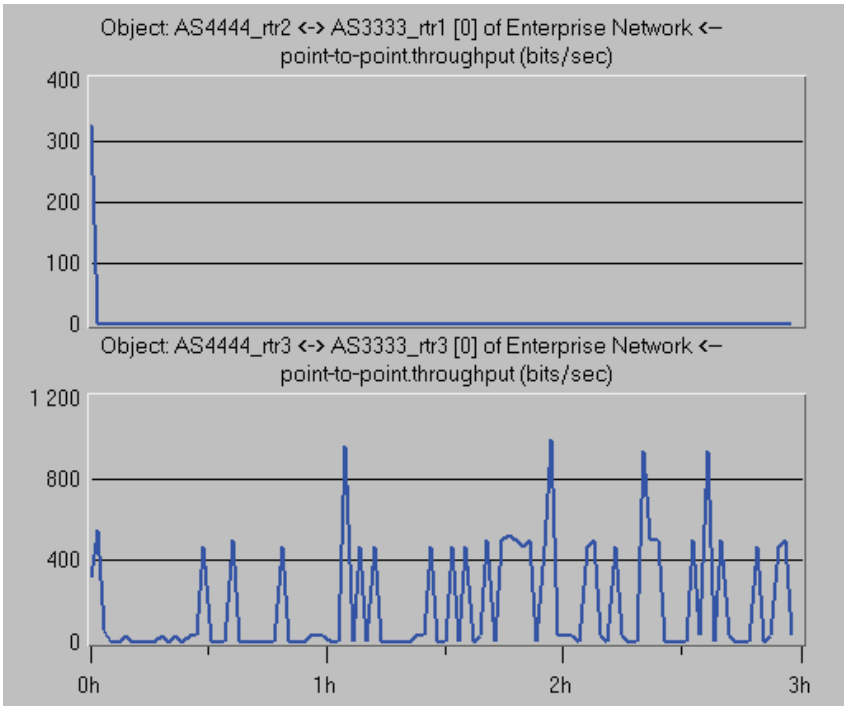
1) Student si duplikuje původní scénář BGP a nazve si ho **BGP_ukol**.
2) Změní si místo připojení pro pc1 a server. Pc1 zkopíruje a připojí ho k AS4444_rtr1, pc1 se automaticky přejmenuje na pc2. Původní pc1 smaže. Server zkopíruje a připojí k AS2222_rtr3, původní server smaže.
3) Nastaví se na směrovačích IP adresy nově připojených objektů. Oba objekty se defaultně připojí na port IF0, možnost zjištění podle atributů linky, kterou jsou propojeni. Pro pc1 to bude u AS4444_rtr1 IF0 IP adresa 192.0.40.2 a maska 255.255.255.0. Pro server bude IP adresa na IF0 192.0.50.2 a maska 255.255.255.0.
4) Student si zobrazí, kudy jde přenos, viz obr. B. 1. Přenos směrem od pc k serveru jde přes AS1111_rtr3 a směrem od serveru k pc jde přes směrovač AS4444_rtr2.



Obr. B. 1: Směr přenosu od pc serveru a zpět

5) Pomocí atributů MED a Local Preference student změní cestu přenosu, nyní půjde směrem do AS 4444 přes směrovač AS4444_rtr3 a směrem ven z AS 4444 přes směrovač AS4444_rtr2.

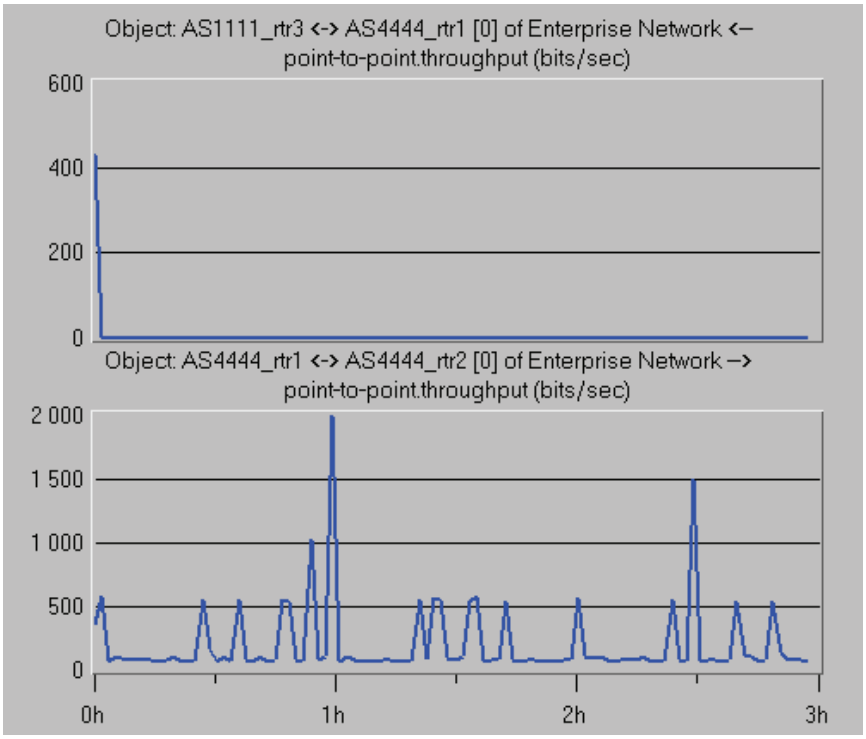
6) Jako první popíšu atribut MED. Nastavení provedeme na AS4444_rtr2 a AS4444_rtr3, kde upřednostníme spodní linku. *„IP Routing Parameters → Route Map Configuration → Map Configuration“*. V *„Match Info“* změníme hodnotu *„Match Value“* na **192.0.40.1 255.255.255.0**. V *„Set Info“* hodnoty nastavíme následovně, *„Set Attribute“* na **Multi Exit Discriminator** a *„Set Value“* pro **AS4444_rtr2** na **200** a pro **AS4444_rtr3** to bude **50**. Tím jsme upřednostnili spodní linku. Dále je potřeba povolit politiku směrování na sousední směrovače **AS 3333**. V *„BGP parameters → Neighbor Information“* si najdeme správný řádek a v *„Routing Policies → Route Map“* vybereme naše vytvořené pravidlo **MED** a *„Applicable Direction“* změníme na **Out**. Spustíme simulaci a ověříme si provoz, viz obr. B. 2. Zde je vidět, že se cesta provozu změnila.



Obr. B.2: Změna přenosu do AS 4444 pomocí atributu MED

7) Změníme přenos ven z AS 4444. Nyní jde přes AS1111_rtr3. Po změně půjde přes AS4444_rtr2. To provedeme pomocí Local Preference na AS4444_rtr1. *„IP Routing Parameters → Route Map Configuration → Map Configuration“*. Zde vytvoříme dva řádky, jeden pro linku k AS 1111 a druhou pro linku k AS4444_rtr2. V obou dvou budeme nastavovat jen parametr *„Set Info“*. V *„Set Attribute“* nastavíme atribut **Local Preference**

a *„Set Value“* pro linku do **AS 1111** nastavíme na **100** a pro linku do **AS4444_rtr3** nastavíme na **200**. Opět musíme povolit politiku. V *„BGP parameters → Neighbor Information“* si najdeme správný řádek a v *„Routing Policies → Route Map“* vybereme nové pravidlo a směr **In**. Spustíme simulaci a ověříme výsledek, viz obr. B. 3.



Obr. B. 3: Změna přenosu do AS 4444 pomocí atributu Local Preference