VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019 Bc. Aneta Koláčková

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SIMULAČNÍ SCÉNÁŘE PRO ANALÝZU CHOVÁNÍ TRANSPORTNÍCH SÍTÍ
SIMULATION SCENARIOS FOR ANALYSIS OF BEHAVIOR OF TRANSPORT NETWORKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Aneta Koláčková**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Jan Jeřábek, Ph.D.**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Studentka: Bc. Aneta Koláčková ***ID:*** 173677
Ročník: 2 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Simulační scénáře pro analýzu chování transportních sítí

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku protokolů sady TCP/IP (Transmission Control Protocol / Internet Protocol), směrovacích protokolů a fungování transportní části mobilních sítí nejnovějších generací. Prostudujte možnosti emulace těchto sítí, zejména v případě využití zařízení od společnosti Cisco. V rámci diplomové práce vytvořte dva komplexní scénáře, které umožní pomocí simulace nastavovat především transportní část sítě, ověřovat si chování a analyzovat dopad jednotlivých způsobů konfigurace. Práce musí obsahovat detailní návody a vzorová řešení. Rozsahem se musí jednat o scénáře realizovatelné za přibližně 2 hodiny času.

DOPORUČENÁ LITERATURA:

[1] FOROUZAN, Behrouz A. TCP/IP protocol suite. 4th ed. Boston: McGraw-Hill Higher Education, 2010, xxxv, 979 s. ISBN 978-0-07-337604-2.

[2] JEŘÁBEK, J. Komunikační technologie. Skriptum FEKT Vysoké učení technické v Brně, 2018. s. 1-172.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 16.5.2019

Vedoucí práce: doc. Ing. Jan Jeřábek, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Diplomová práce se zabývá problematikou mobilních sítí, protokolů sady TCP/IP a směrovacích protokolů. Zaměřuje se zejména na fungování transportní části mobilních sítí třetí a čtvrté generace a snaží se najít nejvhodnější možnosti jejich emulace. V práci jsou porovnány dva vybrané emulační programy prostřednictvím série testů na serveru vytvořeném pro tyto účely. Hlavním výstupem jsou dva detailně popsané komplexní scénáře a jejich vzorová řešení. První scénář se zabývá celkovou konfigurací transportní sítě. Druhý scénář navazuje na první a je založen na hledání chyb v konfiguraci. Pomocí těchto scénářů je pak analyzováno chování transportní části mobilní sítě.

KLÍČOVÁ SLOVA

Simulační scénáře, GNS3, EVE-NG, ESXi, mobilní transportní síť, Cisco, směrovací protokoly

ABSTRACT

This master's thesis deals with the issue of mobile networks, protocol suite TCP/IP and routing protocols. It focuses in particular on the functioning of the transport part of the third and fourth generation mobile networks and intends to find the most suitable options for their emulation. The master's thesis compares two selected emulation programs through a series of tests that were performed on a server created for these purposes. The main outputs are two detailed comprehensive scenarios and solutions have been developed. The first scenario deals with the overall transport network configuration. The second scenario is based on the first one and is focused on troubleshooting of particular issues in this network. Their implementation helps to analyze the behavior of the transport part of the mobile network.

KEYWORDS

Simulation Scenarios, GNS3, EVE-NG, ESXi, Mobile Transport Network, Cisco, Routing Protocols

KOLÁČKOVÁ, Aneta. *Simulační scénáře pro analýzu chování transportních sítí*. Brno, Rok, 190 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Jan Jeřábek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Simulační scénáře pro analýzu chování transportních sítí“ jsem vypracovala 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 autorka uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autorky

PODĚKOVÁNÍ

Děkuji především vedoucímu diplomové práce, panu doc.Ing. Janu Jeřábkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy. Poděkování patří také panu Ing. Michalu Trávníčkovi za cenné rady týkající se problematiky této diplomové práce. Na závěr bych ráda poděkovala své rodině za obrovskou podporu během studia.

Brno ....

podpis autorky

Tato práce vznikla jako součást klíčové aktivity KA6 - Individuální výuka a zapojení studentů bakalářských a magisterských studijních programů do výzkumu v rámci projektu OP VVV Vytvoření double-degree doktorského studijního programu Elektronika a informační technologie a vytvoření doktorského studijního programu Informační bezpečnost, reg. č. CZ.02.2.69/0.0/0.0/16_018/0002575.

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Projekt je spolufinancován Evropskou unií.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Následující diplomová práce se věnuje problematice protokolů sady TCP/IP, směrovacím protokolům a fungování transportní části mobilních sítí třetí a čtvrté generace. Cílem práce je porovnat nejvhodnější možnosti emulace těchto sítí a vytvořit dva komplexní scénáře, díky nimž bude možné analyzovat chování transportní části mobilní sítě. Transportní síť je často opomíjenou částí v architektuře mobilní sítě a přitom hraje klíčovou roli, bez které by nedošlo ke spojení přístupové a páteřní sítě.

V úvodní kapitole bude popsána architektura mobilních sítí třetí a čtvrté generace, počínaje přístupovou částí sítě, přes transportní síť až po jádro sítě. Systematicky bude popsán typ zařízení, technologie, která se v dané části nachází, rozhraní a funkce, které jednotlivé zařízení zastává. Záměrem je přiblížit postavení transportní části u mobilních sítí.

Druhá kapitola se bude zabývat protokoly používanými v transportní síti. Celá transportní síť je založená na protokolu IP, tudíž se bude uvedená kapitola věnovat internetovým protokolům IPv4 (Internet Protocol v4) a IPv6 (Internet Protocol v6), konkrétně adresování a struktuře datagramů. U IPv6 budou zahrnuty i přechodové mechanismy. V kapitole budou dále popsány směrovací metody, statické směrování, OSPFv2 (Open Shortest Path First v2) a OSPFv3 (Open Shortest Path First v3). V neposlední řadě zde bude rozebrán protokol BGP (Border Gateway Protocol) a protokol BFD (Bidirectional Forwarding Detection), které hrají důležitou roli v dnešních sítích. Součástí kapitoly je i QoS (Quality of service) a protokol pro správu sítě SNMP (Simple Network Managment Protocol).

Třetí kapitola bude věnována popisu softwarových a hardwarových prostředků využitých pro emulaci transportní sítě. Rozebereme dostupné emulátory a typy hyperviserů. Na to navážeme popisem vytvoření vlastního serveru s ESXi. Detailně budou rozebrány vybrané emulační programy EVE-NG (Emulated Virtual Environment – Next Generation) a GNS3 (Graphical Network Simulator-3), rovněž s popisem jejich instalace na vytvořený server s ESXi. Poté se v téže kapitole soustředíme na Cisco IOS a programy jako jsou Ostinato, Wireshark a PowerSNMP.

Ve čtvrté kapitole se zaměříme na analýzu a srovnání emulačních programů EVE-NG a GNS3. Poukážeme na zjištěné rozdíly v instalaci a v práci s oběma programy na základě testování tří parametrů: zatížení CPU, doby načtení obrazů a zatížení paměti RAM. Tato kapitola obsahuje také popis výchozí transportní sítě pro testování.

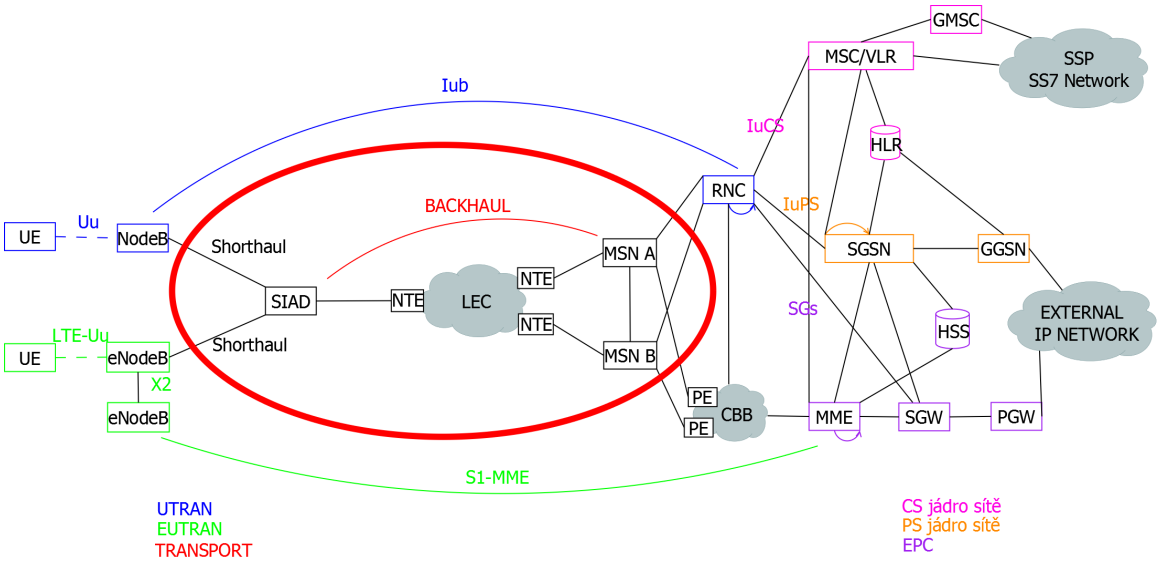
Závěrečná pátá kapitola představí dva simulační scénáře. Scénář 1 se zaměřuje na konfiguraci celé transportní sítě, včetně simulace základnových stanic NodeB a eNodeB, především z pohledu QoS. Scénář 2 má odlišný koncept a je úzce spojen

se Scenářem 1. Jeho podstata spočívá v hledání chyb, které se často v mobilní transportní síti objevují. Oba scénáře budou detailně popsány, včetně vzorových řešení.

V závěru shrneme veškeré poznatky uvedené v této diplomové práci.

1 Mobilní sítě

V následující kapitole je představena obecná struktura mobilních sítí 3G a 4G. Podstatou je přiblížit postavení transportní sítě v mobilních sítích a také objasnit samotnou architekturu mobilní sítě. Jsou zde popsány jednotlivé části s důrazem na jejich význam a funkci, kterou zajišťují. Celkový přehled je vyznačen na Obr. 1.1 a následující podkapitoly na něj navazují. Veškeré zkratky uvedené na tomto obrázku jsou vysvětleny v průběhu popisu jednotlivých komponentů.



Obr. 1.1: Celkový přehled základních součástí mobilních sítí 3G, 4G a jejich propojení

Je nutné zmínit, že architektura mobilní sítě podléhá standardizaci. Zabývá se jí společenství 3GPP (The 3rd Generation Partnership Project). To zaštiťují standardizační organizace ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC. [1] Společenství původně vzniklo s cílem vytvořit globálně použitelnou specifikaci systému mobilních telefonů 3G, později bylo však rozšířeno o další generace a nyní zajišťuje i 5G standardy.

1.1 Přístupová síť (RAN – Radio Access Network) pro sítě 3G a 4G

Přístupová síť je ta část mobilní sítě, která využívá rádiové spojení. Její součástí jsou UE (User Equipment), t.j. mobilní telefon, počítač či jiné zařízení a základnové stanice (NodeB, RNC a eNodeB).

Síť rozlišujeme dle použité technologie. Pokud se jedná o 3G síť, ve většině případů se používá technologie UMTS (Universal Mobile Telecommunications System), kde mluvíme o přístupové síti UTRAN (Universal Terrestrial Radio Access Network). V případě 4G mluvíme o technologii LTE(-A) (The Long Term Evolution (- Advanced)) a její přístupová část je nazývána EUTRAN (Evolved Universal Terrestrial Radio Access Network). Architektury jsou si podobné, nicméně využívají odlišné technologie přístupu ve směru od a k uživateli.

1.1.1 UTRAN – Universal Terrestrial RAN (3G)

UMTS podporuje kmitočtová pásma od 0,8 GHz až do 2,1 GHz. Toto pásmo zajišťuje dobrou propustnost přes překážky a nízký útlum. Kmitočtová pásma se dělí na párovou část a nepárovou část. Párová část využívá frekvenční duplex FDD (samostatné frekvenční kanály pro downlink a uplink) a nepárová část využívá časový duplex TDD (pro přenos jsou vyhrazeny časové sloty v jednom frekvenčním kanálu). [2] UMTS pro obě varianty používá šířku kanálu 5 MHz (2 x 5 MHz pro FDD, 5 MHz pro TDD). Vzhledem k tomu, že se doporučuje použít přednostně FDD, je pro tuto technologii vyhrazeno mnohem širší pásmo. Vyhrazená kmitočtová pásma se liší podle území, např. Evropa a USA používají odlišná pásma. [3]

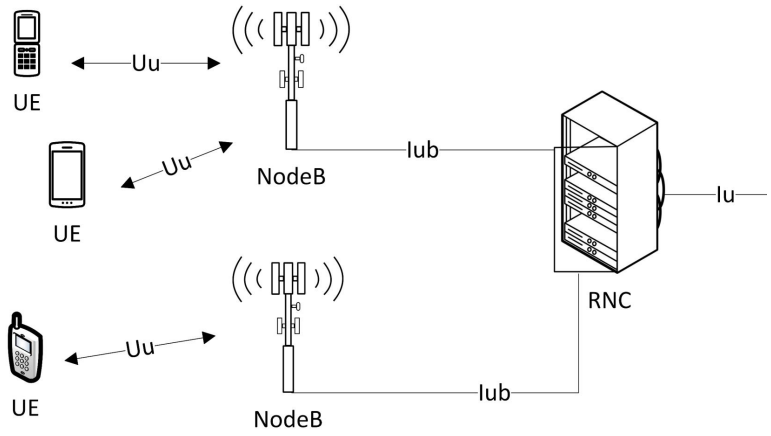
UMTS technologie využívají širokopásmový mnohonásobný přístup s kódovým dělením – WCDMA (Wideband Code Division Multiple Access). Tato technika rozloží informační bity uživatelů do velmi širokého pásma a tím se zvýší kapacita systému. Tím se umožní většímu počtu uživatelů připojit se k jedné základové stanici a za cenu snížení uživatelské přenosové rychlosti dobře odlišit datový signál od ostatních. [2] [3]

Architektura UTRAN

Přístupová síť UTRAN se dělí na menší části, jak lze vidět na Obr. 1.2.

- **UE (User Equipment)** - jedná se o první část, nejčastěji mobilní telefon, který obsahuje USIM kartu[1]. Má na starost veškerou komunikaci na Uu rozhraní.
- Dalšími částmi jsou základnové stanice **NodeB** – jsou tvořeny anténou a DU (Digital Unit). Anténa je umisťována na střechách či stožárech a je spojena koaxiálním kabelem s DU jednotkou nacházející se v přístřešku pod anténou (*Cell Site*). Základnová stanice zajišťuje zpracovávání hovorů, správu radiových zdrojů a nastavení kanálů, sledování výkonu a synchronizaci. Zajišťuje tedy vše potřebné pro připojení UE přes Uu rozhraní.

[1] USIM karta je karta obsahující specifické údaje o uživateli, jako je ID, autentizační algoritmus atd. Je analogií ke kartě SIM používané v 2G sítích, se kterou bývá zaměňována. [3]



Obr. 1.2: Rozdělení architektury UTRAN na jednotlivá zařízení s popisem spojení

- Poslední částí je **RNC** – řídí spojení k UE a provádí řízení a správu pro NodeB. Obvykle má na starost několik desítek NodeB. Dohlíží na komunikaci na Iu lince, která je spojením s páteřní částí sítě a Iub linky, nacházející se mezi NodeB a RNC. Dále pak řídí výkon, přístup a radiové prostředky. Má na starost i bezpečnost UTRAN a uchovává v sobě databázi potřebných údajů pro UE a NodeB. Zajišťuje také QoS značkování dat přicházejících od UE na NodeB.

1.1.2 EUTRAN – Evolved Universal Terrestrial RAN (4G)

EUTRAN je označení pro přístupovou síť u technologie LTE(-A). Pro LTE(-A) je vyhrazeno větší spektrum frekvenčních pásem než pro UMTS. Dle Českého telekomunikačního úřadu (ČTU) jsou pro LTE vyhrazena pásma 800 MHz, 1800 MHz, 2100 MHz a 2600 MHz. [4] Technologie LTE(-A) využívá stejně jako UMTS rozdělení na párovou a nepárovou část, tedy FDD a TDD.

Rozdíl nastává u přístupové metody. LTE(-A) ve směru k uživateli (downlink) používá přístup OFDMA (Orthogonal frequency division multiple access), ve směru od uživatele (uplink) pak SC-FDMA (Single-carrier frequency division multiple access). Vychází z OFDM (Orthogonal Frequency Division Multiplexing), které je založeno na přenosu na více nosných a kombinuje časový TDMA a frekvenční FDMA přístupový multiplex. [3]

Technologie LTE(-A) využívá na přístupové vrstvě další funkce, díky kterým je možné dosahovat vyšších přenosových rychlostí a nižší latence. Jedná se například o podporu MIMO (Multiple Input Multiple Output), tvarování charakteristiky an-

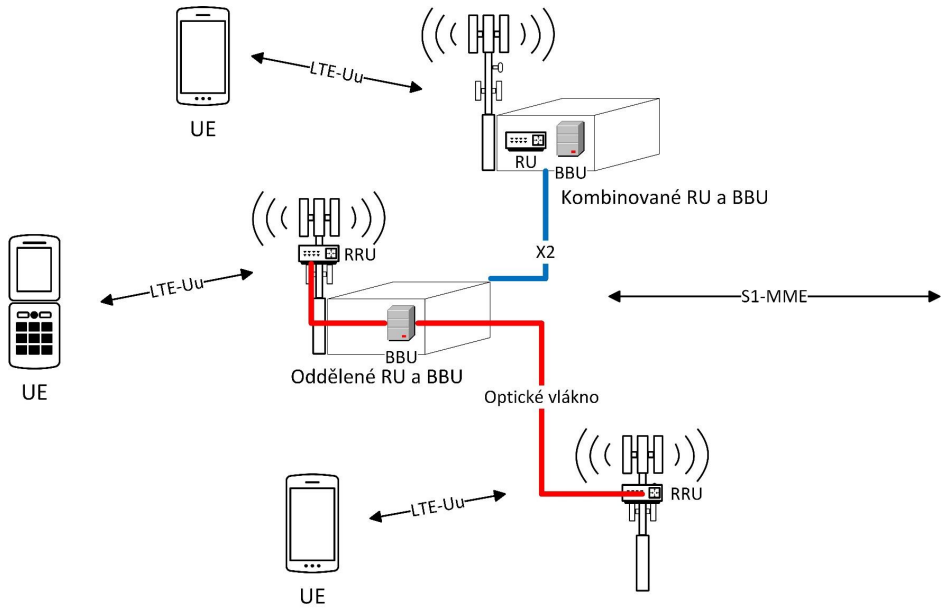
ténního systému (beamforming) nebo diverzity vysílání a příjmu (TX/RX diversity).

Architektura EUTRAN

Z důvodu zjednodušení topologie, snížení latence a k celkovému urychlení se přístupová síť EUTRAN skládá pouze z jediného zařízení a tím je základová stanice eNodeB. Narozdíl od UMTS zde není žádná nadřazená entita jako je RNC pro NodeB. eNodeB nahrazuje funkce zajišťované oběma zařízeními. Řeší správu rádiových zdrojů, rádiových nosičů, rádiové pokrytí, vysílání referenčního signálu a systémových informací. Má na starosti *handover*, přidělování rádiových zdrojů k terminálům UE dle QoS (Quality of service) a veškerou komunikaci směrem do páteřní sítě.

Samotná základová stanice se skládá ze dvou částí: z modulu označovaného jako BBU (Base-Band Unit), jenž má na starosti zpracování signálu v základním pásmu; a RU (Radio Unit), jenž zajišťuje převod mezi vysíláním a příjmem signálu v rádiové oblasti. Mezi těmito moduly je specifikováno rozhraní, které umožňuje jejich oddělení až na kilometry pomocí optických vláken. V takovém případě se jedná o RRU (Remote Radio Unit). [5]

Architektura je znázorněna na Obr. 1.3.



Obr. 1.3: Rozdělení architektury EUTRAN na jednotlivá zařízení s popisem spojení

- **UE (User Equipment)** - může se jednat o mobilní telefon, notebook s mobilním širokopásmovým adaptérem nebo o podobná zařízení. Důležitá je především podpora technologie LTE. S eNodeB je spojen rádiovým rozhraním LTE-Uu.

- Následuje základnová stanice **eNodeB** – její anténní část bývá také umísťována na střechách či stožárech, dle potřeby pokrytí oblasti. Hlavní jednotka se nachází v *Cell Site*, nejčastěji v blízkosti antény. Jak bylo zmíněno výše, je tvořena dvěmi částmi - RU a BBU. eNodeB plní mnoho funkcí samostatně (zastává i funkci RNC). Proto existuje mezi blízkými eNodeB spojení, které je nazýváno X2 a pomocí kterého mezi sebou mohou komunikovat. Zajišťuje např. X2 handover. Spojení dále směrem do páteřní sítě je označováno S1-MME. Data zaslaná od UE jsou na eNodeB značkována pro zajištění QoS při průchodu sítí.

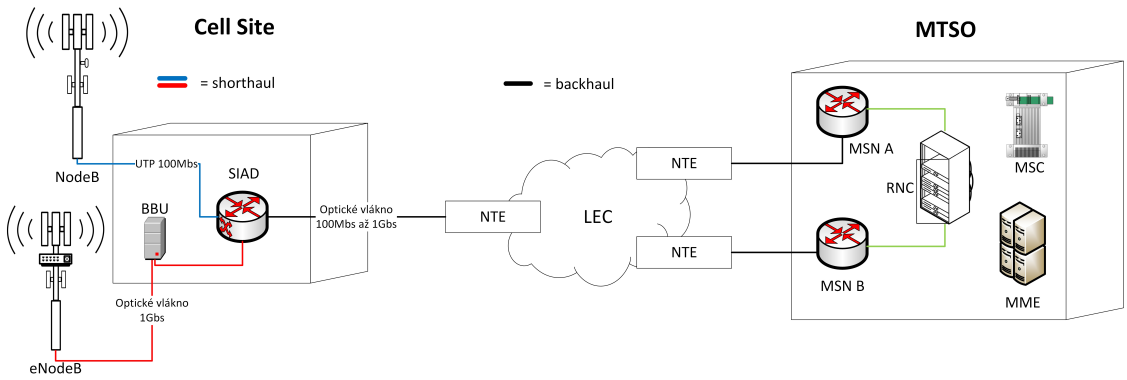
1.2 Transportní síť

V této části se budeme detailněji zabývat transportní sítí. Transportní síť je část RAN sítě, která se stará o přenos dat od uživatele do páteřní sítě a naopak. Není závislá na technologii použité v radiové části. Transportní část je stejná pro 2G, 3G, 4G i 5G sítě. Z pohledu transportní sítě se liší pouze v QoS a náročnosti dle šířky pásma. Transportní síť bývá často opomínána a málo rozebírána, je ale klíčovou součástí, bez které by základové stanice rozmístěné v různých lokalitách jen stěží komunikovaly se sítí páteřní.

Z Obr. 1.4 je patrné, že transportní část začíná za NodeB a eNodeB. Ty jsou připojeny do SIAD (Smart Integrated Access Device) směrovače a toto spojení bývá nazýváno *Shorthaul*. SIAD směrovač se nachází u věže, kde se nachází anténa, nejčastěji v přístřešku pod ní (*Cell Site*). Poté je ze SIADu vedeno spojení, nejčastěji přes optické vlákno, do přepínače NTE (Network Terminal Endpoint), který je hraniční částí lokálního výměnného nositele, tzv. LEC (Local Exchange Carrier). LEC může být pro každý *Cell Site* odlišný, dle nasmlouvaných dohod operátorů. Spojení dále vede od LEC přes NTE přepínače do robustnějších směrovačů označovaných MSN (Multi-Service Node), které bývají v páru. Až zde se nachází spojení s RNC. Místo, kde se nachází MSN, RNC a další zařízení jako jsou MME (Mobility Management Entity) a MSC (Mobile Switching Center), o kterých bude řeč v následujících kapitolách, je nazýváno *Central Office* nebo také ve zkratce MTSO (Mobile Telephone Switching Office). *Cell Site* a MTSO od sebe mohou být vzdáleny až desítky kilometrů.

SIAD (Smart Integrated Access Device)

Směrovač, který je součástí každého *Cell Site*. Jeho úkolem je směrovat veškerá data získaná ze *Shorthaulu* směrem k MSN A a B přes IP protokol. Často využívanými



Obr. 1.4: Architektura transportní sítě

směrovači jsou Cisco ASR, které splňují vysoké požadavky na šířku pásma. Jejich výhodou je také schopnost propojení L2 a L3 vrstvy a agregace dat z více zdrojů.

NodeB je připojeno do SIAD přes UTP kabel rozhraním RJ45. Připojení dosahuje rychlosti do 100 Mb/s. U eNodeB se využívá připojení přes optické vlákno a rychlost toku dat je od 100 Mb/s do 1 Gb/s.

Pakety přicházející od základnových stanic NodeB a eNodeB na SIAD směrovač jsou již označkované pro zajištění QoS. SIAD tedy jen prochází údaje v hlavičce paketu a zachází s nimi dle vlastního nastavení QoS.

SIAD má pouze jeden *Backhaul* port, využívá však L2 vrstvy a BDI viz Kap.3.5. *Backhaul* port je propojen nejčastěji optickým kabelem do NTE přepínače (dříve kabely metalické, v odlehlých oblastech stále mikrovlnné spoje).

LEC (Local Exchange Carrier)

NTE přepínač je vstupní branou do LEC. Jak již název napovídá, jedná se o telefonní společnosti spravující telefonní hovory v určité oblasti. Tento koncept pochází z USA, kde v roce 1984 došlo k nucenému zrušení tehdejšího monopolu, který měla společnost American Telephone & Telegraph.

LEC mohou spravovat oblasti velikosti států, měst i odlehlých míst. Tyto společnosti mohou být přímo vlastníky zařízení nebo si je pronajímat od velkých operátorů. Jsou ale vázány povinnostmi a pravidly, které musí dodržovat např. povinnost dalšího prodeje. LEC jsou povinny neukládat nepřiměřené diskriminační podmínky nebo jiná omezení při prodeji svých telekomunikačních služeb. To znamená, že musí umožnit jiným operátorům využit kapacity svých zařízení za přiměřené ceny. Ačkoliv se operátoři snaží využívat služeb LEC minimálně a upřednostňují vlastní telekomunikační síť, často to není možné a proto musí těmto společnostem za přenos platit.

LEC používají různé technologie a vše je řešeno vzájemnými smlouvami, které stanový šířku pásma nebo přenosové rychlosti. Z toho důvodu se tato práce nebude

zabývat simulací této části transportní sítě. Je důležité také zmínit, že z pohledu SIAD a MSN, není síť LEC viditelná. [6]

MSN (Multi-Service Node) směrovače

Jedná se o typ směrovačů nacházejících se v MTSO (Mobile Telephone Switching Office). Zajišťují spojení z RAN do páteřní sítě CBB (Core Backbone) a jsou posledními uzly před PE (Provider Edge) směrovači páteřní sítě. Často je využíváno směrovačů od firmy Cisco a to ASR 9000, které byly vyvinuty přímo pro hraniční, páteřní a mobilní transportní sítě. Splňují požadované parametry jako je vysoká škálovatelnost, flexibilita, plná podpora L2 a L3 vrstev, atd.

Každý z těchto výkonných směrovačů má na starost okolo 200-400 SIAD směrovačů. Bývají vždy umístěny v páru z důvodu rozložení zátěže a zálohy v případě výpadku jednoho z nich. MSN směrovače, které tvoří pár (označovány MSN A a B), mají přidělené množství *Cell Site* rovnoměrně (rozložení 50% na 50%). Například MSN A je pro určité *Cell Site* primární směrovač. Veškerý tok dat je směrován jeho směrem a MSN B zůstává pro tyto *Cell Site* ve stavu *Standby*. Data na něj budou přesměrována jen v případě výpadku. To stejné platí naopak.

Není zde využito protokolů pro vyvažování zátěže (*loadbalancing*). Důvodem je nezatěžování sítě a tudíž rychlejší odezva.

Vzájemná komunikace mezi směrovači MSN A a B probíhá přes 40 Gb/s spojení. Obě MSN zařízení bývají součástí jedné MTSO.

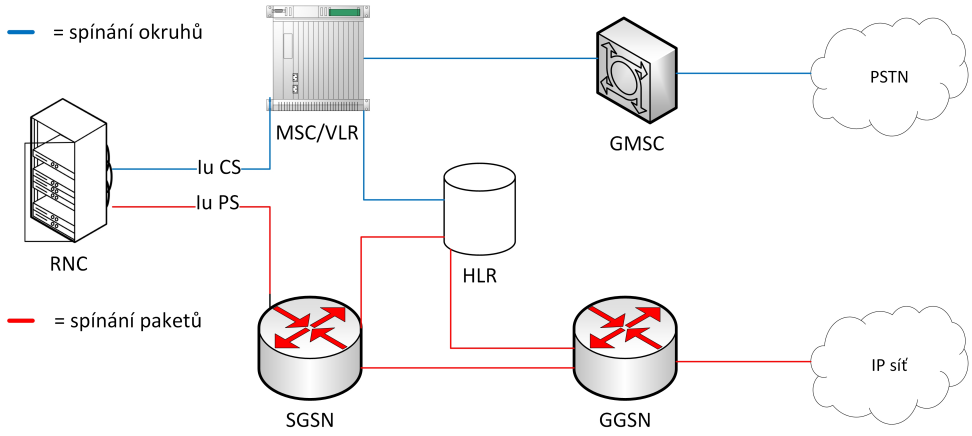
QoS je opět zajištěno dle nastavených parametrů na MSN směrovačích a pakety si i nadále ponechávají značky přidělené od základnových stanic NodeB a eNodeB.

1.3 UMTS (Universal Mobile Telecommunications System): páteřní síť 3G

Jak již bylo řečeno, UMTS lze rozdělit na dvě části – přístupovou síť UTRAN, která je popsána v Kap. 1.1.1 a páteřní síť. UMTS poskytuje dva hlavní typy přenosu uživatelské informace. Prvním typem je přepínání okruhů (Circuit Switched), typický pro hlasové služby. Druhým je paketový přenos (Packet Switched), který je pro hlasové služby a datové přenosy. Rozhraní mezi UTRAN a UMTS páteřní sítí je nazýváno **Iu**. Rozhraní Iu pracuje v režimech, které jsou definovány při sestavování spojení.

- rozhraní Iu CS (Circuit Switched) pro hlasové služby
- rozhraní Iu PS (Packet Switched) pro hlasové služby a datové spojení

Páteřní síť pro oba uvedené druhy provozu dělíme na přepojování okruhů a přepojování paketů, viz Obr. 1.5.



Obr. 1.5: Rozdělení architektury UMTS

U spínání okruhů hovoříme o zařízeních:

MSC (Mobile Switching Center) - telefonní ústředna

Koncept je stejný jako u 2G sítí. MSC je ústřední částí celého systému přepojování okruhů. Zajišťuje nastavení volání, směrování hovorů, spojování hovorů a předávání spojení. MSC hlásí změny polohy sítí. Díky tomu je mobilní stanice dostupná příchozím hovorům, když se uživatel pohybuje. Je částí sítě, která zaručuje, že hovor bude přesměrován bez přerušení hovoru. [19]

HLR (Home Location Register) - domovský registr účastníků

HLR je centrální databáze všech účastníků mobilní sítě. Obsahuje informace o každé SIM v síti (může obsahovat záznamy až několika set tisíc účastníků). Má na starost autentizaci a je jediným místem, kde jsou uloženy šifrovací klíče. V každé síti se musí nacházet minimálně jedno HLR. [2] V praxi je jich větší počet a vyskytují se vždycky v páru, tak aby se zabránilo přetížení a vyhlo se „jedinému bodu selhání“.

VLR (Visitor Location Register) - registr dočasných účastníků

VLR je databáze účastníků obsahující data potřebná pro řízení pohybu uživatele. Logicky souvisí s určitou telefonní ústřednou MSC. Registr dočasných uživatelů bývá často i fyzicky její součástí. V momentě, kdy se účastník přesune do oblasti jiné ústředny MSC, jsou jeho informace zkopírovány z HLR do VLR, kde jsou uchovávány originální data. Výhodou je snížení množství signalizace mezi MSC a HLR. Větší počet poblíž se nacházejících MSC může mít také společný VLR.

GMSC (Gateway Mobile Switching Center) - výchozí brána telefonní ústředny

GMSC zajišťuje spojení UMTS standardu se sítěmi využívajícími technologii přepojování okruhů. Konvertuje formát ze sítí s přepojováním okruhu na protokoly pro mobilní sítě. Agreguje veškerý tok dat z PSTN (Public Switched Telephone Network). V případě příchozího PSTN hovoru vyhledá GMSC vhodný MSC a přesměruje hovor. V případě odchozího hovoru zajistí PSTN směrování.

Jelikož celková architektura počátečních verzí 3G byla převzata ze sítí 2G, je zachován koncept rozdělení na přepojování okruhů a přepojování paketů, jak je zmíněno v úvodu této podkapitoly. Výhodou přepojování paketů, na rozdíl od přepojování okruhů, je datový přenos společně s hlasovými službami, který přinesla u sítí 2G technologie GPRS (General Packet Radio Service). Proto zařízení tvořící páteřní síť 3G u přepojování paketů nesou v názvu stále tuto technologii, ačkoliv používají technologie HSDPA (High-Speed Downlink Packet Access) a HSUPA (High-Speed Uplink Packet Access). Zařízení jsou zobrazena na Obr. 1.5 a jedná se o následující 2 uzly:

SGSN (Serving GPRS Support Node) - obslužný uzel podpory GPRS

Tato ústředna je základem systému pro paketové přenosy. Má na starost autentizaci, šifrovaní, *mobility management* (připojení/odpojení a řízení polohy) a zajišťuje doručování datových paketů pro UE. Uchovává si kopii profilu účastníka, kterou získá z HLR, na který je SGSN připojen. Je ekvivalentem MSC a VLR u přepínaní paketů. S RNC je SGSN spojeno IuPS.

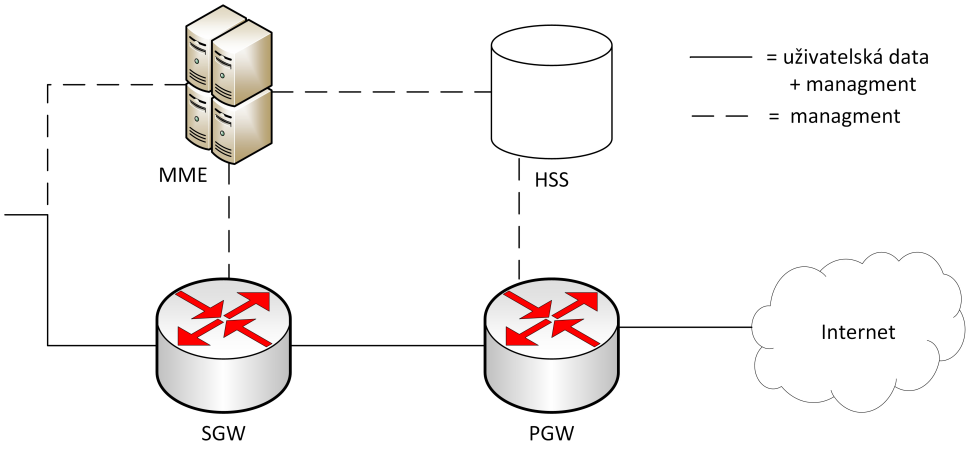
GGSN (Gateway GPRS Support Node) - bránový uzel podpory GPRS

Tento uzel je zodpovědný za propojení sítě UMTS s vnějšími sítěmi s přepojováním paketů, jako je Internet a X.25. GGSN převádí pakety přicházející z SGSN do správného datového protokolu a posílá je do odpovídající paketové sítě. V opačném směru dojde k převodu adresy příchozích datových paketů na adresu cílového účastníka; tyto pakety jsou pak poslány odpovědnému SGSN. Bránový uzel má na starost řízení zásob IP adres a také chrání páteřní síť filtrováním dat a plní funkci firewallu. Je ekvivalentem k GMSC u přepojování okruhů.[22]

1.4 LTE (Long Term Evolution): páteřní síť EPC (Evolved Packet Core) 4G

LTE se vyznačuje ryze paketovým systémem a již nepoužívá přepojování okruhů. Je plně založený na protokolu IP s rozdělením na kontrolní a datové nosiče (*control plane* a *user plane*). Díky implementaci LTE došlo ke zvýšení datových rychlostí z Mb/s na Gb/s. LTE také přineslo nižší latenci, vyšší spolehlivost a zvýšení spektrální účinnosti.

EPC (Evolved Packet Core), jak se také jinak označuje páteřní síť LTE, je jednodušší než páteřní síť UMTS, protože zde nejsou prvky pro přepínání okruhů. Obsahuje nové bloky MME (Mobility Management Entity), HSS (Home Subscriber Server), SGW (Serving Gateway) a PGW (Packet Gateway). Jsou zobrazeny na Obr. 1.6 a jejich funkce popsány níže.



Obr. 1.6: Rozdělení architektury LTE

MME (Mobility Management Entity)

Jedná se o hlavní řídicí prvek sítě LTE. Má na starost větší počet eNodeB. Dohlíží na přístup do sítě a zajišťuje autentizaci. Porovnává odezvy od UE a HSS. Neprocházejí přes něj však uživatelská data. MME provádí šifrování komunikace a chrání tak proti odposlechu. Sleduje pohyby všech účastníků v dané oblasti a po přihlášení UE do sítě pošle jeho údaje do HSS. MME si ukládá kopie o UE, které získá z HSS během jeho obsluhování. Pokud dojde k přepojení na jiný MME, zašle kopii uživatelského profilu do nového bloku a svoji kopii smaže.[21]

MME je také zařízením, jenž zajišťuje řídící funkce mezi LTE a 3G přístupovou částí sítě pomocí spojení se SGSN (Serving GPRS Support Node), viz Obr. 1.1, spojení je označováno SGs. SGs rozhraní slouží jako spojení mezi MME a MSC,

probíhá zde tzv. *CS Fallback*. *CS Fallback* je přesměrování z 4G na přepínání okruhů u technologie 3G, a to z důvodu odlišných technologií. Bez tohoto přesměrování by nebylo možné hovor mezi 4G a 3G uskutečnit.

HSS (Home Subscriber Server)

HSS je databáze uchovávající informace o všech účastnících v síti. Je spojena se všemi MME v síti. Zajišťuje funkce jako jsou *mobility management* nebo podpora sestavování hovoru a provádí autentizaci a autorizaci přístupu. Je založena na HLR (Home Location Register) v UMTS sítích.

SGW (Serving Gateway)

SGW je servisní brána starající se o směrování uživatelských datových paketů. Slouží jako kotva při *handoveru* mezi dvěma eNodeB i mezi LTE a jinými 3GPP technologiemi. V případě soudem povoleného odposlechu právě zde dochází replikovaní uživatelského provozu.

PGW (Packet Gateway)

Brána PGW je směrovač mezi EPC a jinými paketovými sítěmi. Přiděluje UE IP adresy. UE může být připojen zároveň k několika PGW pro připojení k více sítím. Dalšími funkcemi jsou kontrola dodržování politiky směrování, filtrování provozu nebo možnost legálního odposlouchávání. Je také klíčovým bodem pro komunikaci mezi 3GPP a non-3GPP technologiemi jako je např. WiMAX.

1.5 CBB (Core Backbone) páteřní síť

Na závěr úvodní kapitoly a pro pochopení celé problematiky sítě je nutno zmínit CBB. CBB síť stojí mezi transportní a páteřní částí (EPC a UMTS) mobilní sítě.

Má specifické postavení v síti, které se liší dle operátora a jím pokryté oblasti. Je to tedy interní síť operátora, do které ústí veškeré jeho poskytované služby.

Úkolem CBB pro mobilní síť je směrovaní dat. V případě 4G jde o směrování od MSN zařízení na jedné straně např. státu, k požadovanému MME a SGW/PGW na straně druhé. U 3G sítě se CBB využívá především pro přepojování paketů, pro přepojování okruhů se používá starší SS7 síť.

Celá síť CBB je postavena na protokolech BGP, MPLS a OSPF. Tvoří ji pouze několik desítek směrovačů.

2 Protokoly využívané v transportní síti

Druhá kapitola popisuje protokoly, které zde jsou podrobně popsány. Celá transportní síť je postavena na protokolu IP. Využívá obě jeho verze, IPv4 pro mobilní sítě 3G a IPv6 převážně pro 4G. Zmíněny jsou i přechodové mechanismy. Důležitou úlohu hraje v transportní síti směrování, kterému se tato kapitola bude podrobněji věnovat. Konkrétně se zaměří na směrování statické, detailněji protokol OSPF a protokol BGP.

Dále je zde popsán protokol BFD, který je vázaný na směrovací protokoly použité v transportní síti. Následně se kapitola věnuje využití QoS a protokolu SNMP pro správu sítě.

2.1 Internetový protokol IPv4

Protokol IP je paketově orientovaný přenosový mechanismus využívaný TCP/IP protokoly na síťové vrstvě. Je to nespolehlivý a nespojovaný datagramový protokol, který funguje způsobem *best effort*. Tedy snaží se o co nejlepší doručení.

Díky tomu, že protokol IP je nespojovaný, je každý datagram zpracováván samostatně a může tak následovat jinou cestu ke stejnému cíli. Zvyšuje se tak pravděpodobnost poškození, ztracení anebo doručení paketu ve špatném pořadí. Z toho důvodu je protokol IP závislý na protokolech vyšších vrstev, které tyto problémy řeší. Pokud je při přenosu důležitá spolehlivost, může být např. spárován s protokolem TCP (Transmission Control Protocol). [23]

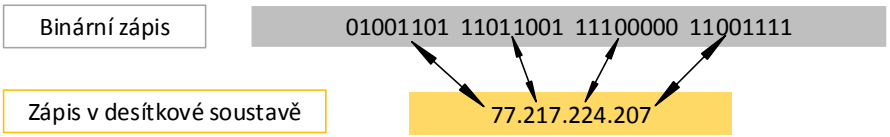
2.1.1 Adresování v IPv4

Podstatou adresování v IPv4 je unikátním způsobem identifikovat jednotlivá zařízení v Internetu tak, aby byla umožněna komunikace mezi všemi zařízeními.

IPv4 adresa je 32-bitová a pochází z adresního prostoru. Adresní prostor je celkový počet adres užívaných protokolem IPv4. U IPv4 se jedná o 2^{32} adres (4 294 967 296).

Adresy je možné zapisovat v binární podobě jako čtveřici oktetů, kde každý oktet (byte) má osm bitů (bit 1 nebo 0). Pro lepší zapamatování jsou adresy častěji zobrazovány v desítkové soustavě a oddělené tečkami. Jelikož jeden byte má osm bitů, dosahuje číslo v desítkové soustavě pouze hodnot mezi 0 a 255, viz Obr. 2.1. [23]

IP adresa se skládá z několika částí. Základem jsou dvě části - prefix identifikující adresu sítě a adresu stanice. V době, kdy bylo IP adresování na samém začátku, byl využíván systém adresování pomocí tříd (classful). Adresní prostor byl rozdělen do tříd od A do E. Zde byl prefix pevně dán třídou, do které adresa patřila. Od tohoto



Obr. 2.1: Zápis v binární a desítkové soustavě

konceptu, který nebyl zdaleka ideální, se ustoupilo a přešlo se na tzv. beztřídní adresování (classless). [24]

U beztřídního adresování můžeme prefix určit z masky podsítě (subnet mask). Maska podsítě se nachází za IPv4 adresou. Obsahuje v binárním tvaru jedničky následované nulami. Jedničky nám udávají část síťového prefixu v IPv4 adrese a nuly naopak část adresy stanice. Maska podsítě se zapisuje také v desítkové soustavě oddělené tečkami. Příkladem síťové masky je 255.255.255.0. Lze se setkat i se zkráceným zápisem masky, označovaným jako CIDR (Classless Inter–Domain Routing). Zapisuje se v binární formě jako IP adresa následovaná lomítkem a číslem, které vyjadřuje počet jedničkových bitů v masce podsítě. Síťová maska 255.255.255.0 se tedy zapíše jako /24. [25]

Ačkoliv se již plně přešlo na beztřídní adresování s podsíťováním, koncept tříd stále zůstává orientačním bodem. V Tab 2.1 lze vidět rozdělení na třídy v historickém třídním systému, i nynější rozdělení beztřídního adresování s důrazem na délky prefixů.

Tab. 2.1: Rozdělení IPv4 na třídy classful a classless podle délky prefixů

Třída dělené sítě	První oktet	Délka prefixu - classful	Délka prefixu - classless
A	0-127	/8	/9 - /30
B	128-191	/16	/17 - /30
C	192-223	/24	/25 - /30
D	224-239	Multicastové adresy	Multicastové adresy
E	240-255	Experimentální adresy	Experimentální adresy

Druhy IPv4 adres

IPv4 adresy se dále dělí na vnitřní a vnější. Vnitřní jsou ty adresy, které lze používat jen v uzavřených sítích. Nelze je směrovat dále v Internetu. Nazývají se privátní a je pro ně přesně vyčleněný rozsah, viz Tab. 2.2. [24] Opakem jsou sítě vnější, používané

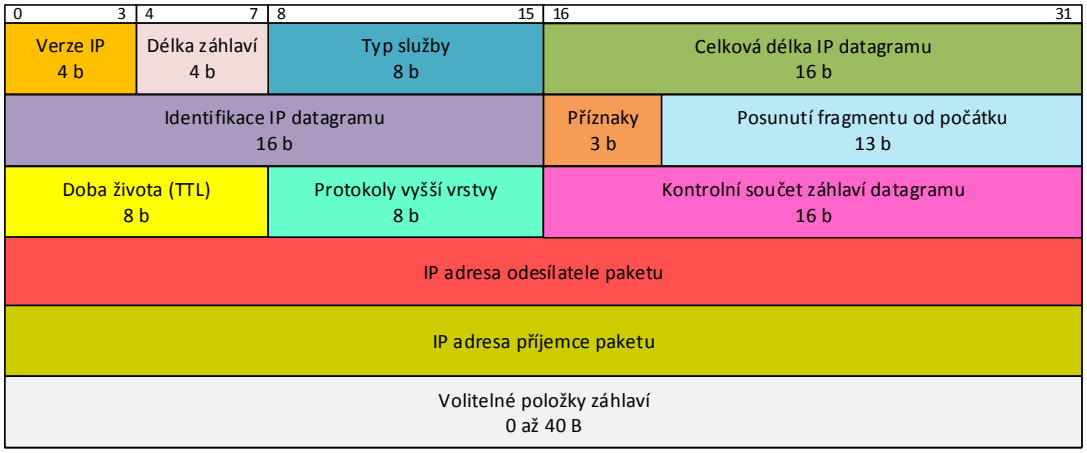
pro směrování mimo uzavřené sítě. Při emulaci transportní sítě jsou v této práci použity privátní i vnější IPv4 adresy.

Tab. 2.2: Rozsahy pro privátní IP adresy

Třídy	Rozsah IP adres	Počet sítí	Počet hostů v 1 síti
A	10.0.0.0 – 10.255.255.255	1	16 777 214
B	172.16.0.0 – 172.31.255.255	16	65 534
C	192.168.0.0 – 192.168.255.255	256	254

2.1.2 IPv4 datagramy

IPv4 datagramy, též nazývané pakety, zavádí jednotnou abstrakci formátu datových jednotek na síťové vrstvě. [24] Paket má proměnlivou velikost a je tvořen ze dvou částí. První je hlavička, jejíž velikost je od 20 do 60 bytů a obsahuje informace nezbytné pro směrování a doručení paketu. Druhou částí jsou samotná data. Minimální velikost celého paketu je 20 bytů, maximální pak 65 535 bytů. Při popisu hlavičky IPv4 datagramu, viz Obr. 2.2, bylo čerpáno z [23] a [24].



Obr. 2.2: Hlavička IPv4 datagramu - paketu [24]

- **Verze IP** - pole definující verzi IP protokolu. Momentálně hovoříme o verzi 4, v další kapitole bude zmíněna další možná a to verze 6 (IPv6). Pokud zařízení nepodporuje danou verzi, paket je zahozen.
- **Délka záhlaví** - pole definující celkovou délku hlavičky datagramu. Je potřebné z důvodu proměnlivosti velikosti hlavičky (od 20 bytů do 60 bytů) a kvůli volitelným položkám záhlaví.

- **Typ služby** - pole identifikující prioritu paketu. Je využíváno u QoS. (Quality of service)
- **Celková délka IP datagramu** - pole definující délku celého datagramu včetně hlavičky. Maximální hodnota, jak bylo v úvodu podkapitoly již zmíněno, je 65 535 bytů.
- **Identifikace IP datagramu** - pole sloužící k rozlišení a určení příslušnosti k sobě patřících fragmentů.
- **Příznaky** - pole související opět s fragmentací. Obsahuje požadavek DF (*don't fragment*) nebo MF (*more fragments*). Informují příjemce o tom, jak mají zacházet s paketem.
- **Posunutí fragmentu od počátku** - určuje pozici obsahu dat v rámci fragmentovaného paketu. Usnadňuje znovu jeho poskládaní na straně příjemce.
- **Doba života (TTL)** - údaj značící maximální počet skoků, který může IP datagram uskutečnit, dokud nedojde k jeho zahození.
- **Protokoly vyšší vrstvy** - pole nesoucí informaci o protokolu vyšší vrstvy. Obvykle se jedná o transportní protokoly TCP (Transmission Control Protocol nebo UDP (User Datagram Protocol).
- **Kontrolní součet záhlaví datagramu** - počítá se pouze ze záhlaví datagramu. Kontrola probíhá na každém uzlu. V případě, že součet nesedí (chyba), dojde k zahození datagramu.
- **IP adresa odesílatele paketu/příjemce paketu** - jedná se o nejdůležitější části IP datagramu, bez nichž by nebylo možné pakety směrovat. Tyto adresy zůstávají neměnné během celého přenosu.
- **Volitelné položky záhlaví** - jsou nepovinné a používají se minimálně. Někdy bývají dokonce zakázány.

2.2 Internetový protokol IPv6

IPv6 je protokol, který je nástupcem předešlého protokolu IPv4. K zavedení IPv6 protokolu vedlo několik důvodů. Nebyl to pouze problém s vyčerpáním adresního prostoru u IPv4, ale také velké rozrůstání rozsahu směrovacích tabulek, nadbytečná režie, nedostatečná podpora multimédií nebo neexistující *end-to-end* komunikace. [26]

Verze 6 odpověděla na výše zmíněné problémy následujícími změnami. Informace byly čerpány ze zdrojů [23] a [26]:

- **Větší adresní prostor** - IPv6 adresa má 128 bitů. Ve srovnání s 32 bitovou adresou u IPv4 je to navýšení o 2^{96} adres.
- **Zredukování směrovacích tabulek** - pomocí hierarchického směrování, díky velkému počtu adres v jedné síti.

- **Zjednodušený formát záhlaví** - méně povinných položek.
- **Podpora zabezpečení** - možnost autentizace a kryptografického zabezpečení, které zvyšuje důvěryhodnost paketu.
- **Mechanismy pro zajištění QoS** - značkování toku dat v záhlaví.
- **Úbytek nadbytečné režie** - zvyšuje se tak rychlost zpracování ve směrovačích (nepočítá CRC, absence fragmentace).

Protokol v6 přináší nicméně také problémy. Související především s velkým počtem adres. Nelze projít všechny IPv6 adresy v jedné síti v rozumném čase, abychom zjistili, které adresy jsou funkční a které nikoliv. Znamená to rovněž vetší náklady pro poskytovatele a nutnost řešit koexistenci IPv6 se stávající IPv4, viz Kap. 2.2.3.

2.2.1 Adresování v IPv6

Adresní prostor se rozšířil na 2^{128} IP adres. Je tedy velmi nepravděpodobné, že by došlo k jeho vyčerpání. Hospodaření s adresním prostorem je tedy neefektivní a nedává si za cíl šetřit IP adresami. [26]

IPv6 adresa je zapisována po 8 skupinách o 4 hexadecimálních číslicích oddělených dvojtečkou, např.:

6000:0000:0000:0000:0ABC:DEF1:0345:789A

Každá skupina oddělená dvojtečkami je tvořena 16 bity. Jelikož adresy jsou pro člověka těžko zapamatovatelné, byl zaveden tzv. zkrácený zápis. Ten umožňuje nahradit souvislou posloupnost nul znakem „::“ (pouze jednou v zápisu). Je možné i vynechat úvodní nulu v každé skupině. Po těchto úpravách bude nejkratší možný zápis IP adresy vypadat následovně [26]:

6000::ABC:DEF1:345:789A

IPv6 adresy dále obsahují prefix, stejně tak jako IPv4. Prefix je začátek IP adresy, který určuje adresu sítě (nebo podsítě). Zapisuje se lomítkem a je umístěný za IPv6 adresou, např. /64. Druhá část adresy určuje adresu stanice.

Adresy se u protokolu verze 6 dělí do tří základních skupin:

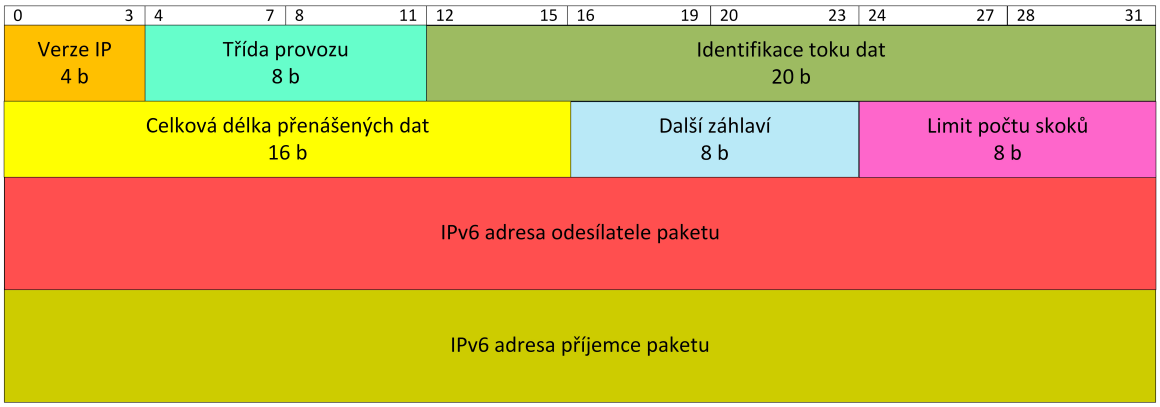
- **Individuální (unicast)** - adresy určující konkrétní stanici v síti. Dělí se dále na několik typů:
  - Globální unikátní
  - Lokální unikátní
  - Linkové unikátní
  - Nespecifikované
  - IPv4 kompatibilní
  - Lokální smyčky

- **Skupinové (multicast)** - adresy určující skupinu stanic, kterým má být paket doručen. Zastupují i všesměrové adresy (broadcast), které nejsou v IPv6 samostatně.
- **Výběrové (anycast)** - adresy také označující skupinu adres. Rozdíl oproti skupinovým spočívá v tom, že obvykle dojde k doručení pouze jednomu členu a to obvykle tomu, který je nejblíže.

U IPv6 je podstatnou změnou oproti IPv4 skutečnost, že síťové rozhraní obsahuje často více než jednu adresu. [26]

2.2.2 IPv6 datagramy

IPv6 datagramy jsou tvořeny dvěma částmi. První je povinná základní hlavička, následovaná druhou částí - přenášenými daty. Součástí přenášených dat je i rozšiřující záhlaví. Povinná základní hlavička má pevně danou velikost, 40 bytů. Rozšiřující záhlaví společně s přenášenými daty může mít až 65 535 bytů. Přestože došlo ke zjednodušení základní hlavičky IPv6, zvětšila se velikost oproti základní hlavičce IPv4 na dvojnásobek. Je to dáno především čtyřikrát delšími adresami u IPv6. Na Obr. 2.3 je znázorněna hlavička základního záhlaví. Při jejím popisu bylo čerpáno z [23] a [26].



Obr. 2.3: Hlavička IPv6 datagramu - paketu [24]

- **Verze IP** - definuje verzi protokolu, u IPv6 má pole hodnotu 6.
- **Třída provozu** - umožňuje nastavit prioritu paketu a upřednostnit některé pakety před ostatními. K zajištění kvality služeb - QoS.
- **Identifikace toku dat** - v tomto poli dochází k označování toku dat. Pakety označené stejným číslem jsou posílány stejnou cestou v síti. Usnadňuje to směrování paketů. Tato metoda není zatím plně využívána.
- **Celková délka přenášených dat** - udává délku přenášených dat bez základního záhlaví.

- **Další záhlaví** - nese informace o rozšiřujícím záhlaví. Pokud rozšiřující záhlaví není přítomno, ukazuje pole na přítomnost protokolu vyšší vrstvy (UDP nebo TCP). U IPv4 se toto pole nazývá „Protokoly vyšší vrstvy“.
- **Limit počtu skoků** - odpovídá poli „Doba života (TTL)“ u IPv4. Má i stejný význam. Slouží jako ochrana před zacyklením paketů v síti. Při průchodu přes směrovače se hodnota postupně dekrementuje a následně je paket odstraněn.
- **IPv6 adresa odesílatele paketu/příjemce paketu** - obojí je o velikosti 128 bitů.

2.2.3 Přechodové mechanismy

Internet je tvořen velkým počtem systémů, které využívají IPv4. K plnému přechodu na IPv6 nedojde okamžitě. Proto existují řešení, která zajistí postupný, nicméně dlouhodobý přechod a umožní koexistenci obou protokolů bez komplikací. [23] Jedná se o tyto tři metody:

Souběh internetových protokolů - Dual Stack

Před přechodem na IPv6 se doporučuje využívat Dual Stack protokol, který plně podporuje jak IPv4, tak i IPv6. Podpora tohoto protokolu se ale odráží na ceně zařízení a celkovém zvýšení nákladů. Další nevýhodou zůstává stále velká potřeba IPv4 adres. I přesto se jedná o nejpřijatelnější metodu pro následující roky. [24] Dual Stack se také využívá v mobilních transportních sítích, viz Kap.4 a Kap.5.

Tunelování

Tunelování je metoda, kdy spolu dva IPv6 systémy chtějí komunikovat, ale spojuje je IPv4 síť. Pro přechod touto sítí musí mít pakety IPv4 adresu. Řešením je zapouzdření IPv6 paketu do IPv4 při vstupu do IPv4 sítě. Při odchodu z IPv4 sítě dojde k odpouzdření. Aby bylo jasné, že se jedná o IPv4 paket nesoucí IPv6 paket jako data, používá se v hlavičce hodnota 41 u položky „Protokoly vyšší vstvy“. [23]

Překlad adres

Tuto metodu lze využít v situacích, kdy většina systému přešla na IPv6, ale v síti stále existují systémy využívající IPv4. Příkladem je situace, kdy odesílatel chce využít IPv6, ale příjemce IPv6 nerozumí. Tunelování zde logicky nefunguje, protože platí podmínka, že paket musí být v IPv4, aby příjemce mohl datagram zpracovat. Musí dojít k překladu. Tato technika se nazývá NAT-PT (Network Address Translator - Protocol Translator).

2.3 Směrovací metody v transportní síti

Následující podkapitola pojednává o směrovacích protokolech na transportní vrstvě. Směrování je proces, který zajistí cestu od zdroje k cíli přes komunikační síť a následně i přenosu uživatelské informace po této cestě. [27]

Transportní síť je založena na statických cestách, protokolu OSPF (Open Shortest Path First) a protokolu BGP (Border Gateway Protocol). Jejich detailnější popis je uveden níže.

2.3.1 Statické směrování

Jedná se o nejjednodušší metodu. V jednotlivých uzlech sítě je definováno, která výstupní linka má být zvolena pro konkrétního adresáta. Nastavení je fixní a nereaguje pružně na změny v síti. Výhodou této metody je nulová zátěž sítě co se týče výměny směrovacích informací. [24]

Důležité je podotknout, že statické směrování má výchozí administrativní vzdálenost[1] rovnu 1. Pokud není hodnota ručně změněna, má statické směrování přednost před dynamickým.

U IPv6 má statické směrování stejný význam jako u IPv4. Bývá však nutné na směrovačích IPv6 směrování povolit.

2.3.2 OSPFv2 (Open Shortest Path First version 2)

Protokol OSPF je dynamickým protokolem typu *link-state*. Řadí se mezi protokoly IGP (Interior Gateway Protocol), tedy uvnitř autonomního systému. Tento typ směrovacích protokolů představuje pokročilejší způsob hledání „nejlepší“ cesty. Umožňuje přiřadit linkám bezrozměrné metriky, tzv. ceny. Cena je na rozhraní přiřazena automaticky na základě šířky pásma (čím větší šířka pásma, tím menší metrika) nebo také ručně administrátorem. Na základě této metriky hledají směrovače cestu k cíli s nejmenší celkovou metrikou. [27] K tomuto hledání využívá protokol OSPF algoritmus Shortest Path.

Sekce tvořící tuto kapitolu mají za cíl objasnit základy protokolu OSPF. Text je koncipován s ohledem na využití protokolu v Cisco zařízeních.

Algoritmus Shortest Path

Nejkratší cesta je vypočítávána s pomocí Dijkstrova algoritmu. Shortest Path algoritmus považuje každý směrovač za kořenový a vypočítává nejkratší cesty ke každému

[1] Administrativní vzdálenost je vlastnost používaná na směrovačích k určení nejlepší cesty mezi více směrovacími protokoly. Definuje spolehlivost protokolu a priorizuje lepší nižším číslem. [25]

dostupnému místu tak, že sčítá metriky. Každý směrovač tak má vlastní pohled na topologii, přestože všechny směrovače vytvoří nejkratší „Shortest Path Tree“ s využitím stejné *link-state* databáze. [28]

Typy směrovačů v OSPF

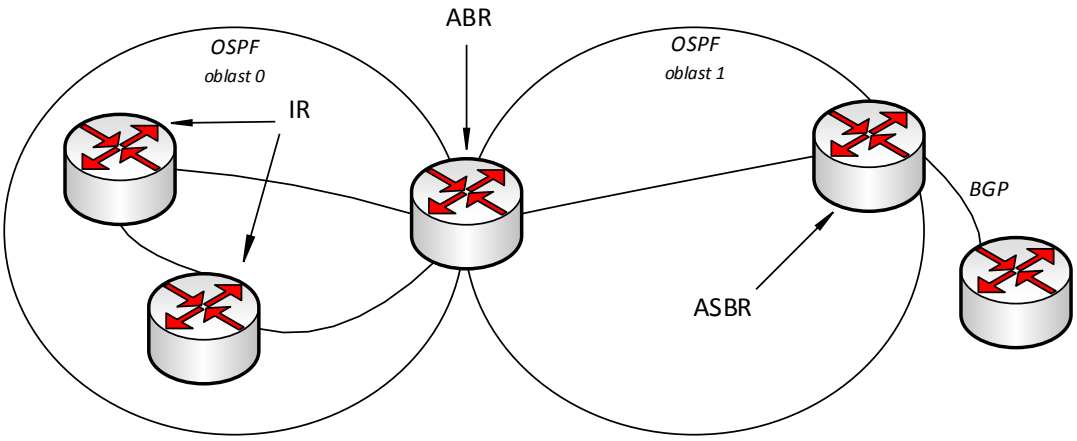
V případě jakékoliv změny v síti využívá OSPF záplavové vyměňování updatů mezi směrovači (dynamický protokol). Díky tomu je nám umožněno rozdělit síť na oblasti a vytvořit hranice. Updaty jsou posílány buďto jen v daných oblastech nebo i mimo ně, dle jejich typu. Čerpáno bylo z [28].

Směrovače, které patří do více oblastí, jsou nazývány **area border router (ABR)**. Ty mají za úkol šířit informace ze směrovačů nebo informace o změnách ve směrovaní mezi oblastmi.

Směrovač, který má všechny svoje porty ve stejné oblasti se nazývá **internal router (IR)**.

Má-li směrovač funkci výchozí brány, tj. nachází se mezi OSPF a jiným směrovacím protokolem (IGRP, EIGRP, RIP, BGP), je nazýván **autonomous system border router (ASBR)**. Jakýkoliv směrovač může být ABR nebo ASBR.

Jednotlivé typy směrovačů lze vidět na Obr. 2.4.



Obr. 2.4: Typy směrovačů u protokolu OSPF [28]

DR a BDR směrovače

Jak již bylo uvedeno, OSPF využívá záplavové vyměňování updatů. Z důvodu minimalizování množství dat vyměňovaných mezi jednotlivými segmenty, určí OSPF

jeden ze směrovačů jako DR tedy *Designated Router* a druhý jako BDR neboli *Backup Designated Router*.

Účelem tohoto určení je vytvoření centrálního bodu pro vyměňování dat. Místo toho, aby si směrovače vyměnily všechny informace mezi sebou, vymění si informace pouze s DB a BDR a poté tyto informace rozesílají dále.

Výběr informací probíhá pomocí „Hello paketů“. Směrovače si rozešlou Hello pakety mezi sebou a navzájem porovnají OSPF prioritu. Směrovač s nejvyšší prioritou se stane DR. Stejný proces platí pro BDR. V případě že je hodnota priorit stejná, rozhoduje „router ID“. Ve výchozím nastavení je priorita nastavena na 1. Pokud je hodnota priority 0, rozhraní nebude soutěžit o výběr DR nebo BDR a bude směrovač označen jako DROTHER. [28]

Zprávy protokolu OSPF

OSPF má pro komunikaci definované tři základní typy zpráv, které souvisí s dílčími procesy směrovacího protokolu [25] [27] :

- **Hello zpráva** - zpráva, která slouží k navázání spojení. Pravidelně se vysílá v intervalu 10 s po všech rozhraních směrovače. Slouží pro detekci a následnou pravidelnou kontrolu dostupnosti sousedních směrovačů.
- **Zpráva s popisem databáze (Database Description)** - zpráva, díky které dochází k výměně informací o topologii sítě mezi sousedními směrovači po navázání spojení pomocí „Hello zpráv“.
- **LSA zprávy (Link-State Advertisements)** - zprávy sloužící k aktualizaci jednotlivých části databáze. Jsou rozděleny do 5 základních typů. LSA zprávy jsou generovány dle typu oblasti a směrovače:
  - *LSA typ 1 (Router LSA)* - informuje o stavu a ceně linky. Týká se pouze jedné konkrétní oblasti. Jeho součástí je číslo sítě, maska podsítě a údaj, o jaký typ směrovače se jedná.
  - *LSA typ 2 (Network LSA)* - je generován DR směrovačem. Rozesílá všesměrově informace o všech připojených směrovačích pro danou oblast.
  - *LSA typ 3 (Summary LSA)* - je tvořen směrovačem na pomezí dvou OSPF oblastí, neboli ABR. Popisuje síť uvnitř a vně oblasti. Může nést informace o výchozí cestě. Rozesílá se obvykle v celém AS (Autonomous System).
  - *LSA typ 4 (ASBR Summary LSA)* - tento typ zpráv nám říká, jak se dostat k ASBR. Je generován ABR, ale není ve stejné oblasti jako ASBR. Přenáší se přes oblasti.
  - *LSA typ 5 (External LSA)* - přenáší informace o vnějších sítích, tj. sítích mimo náš AS. Je generován ASBR směrovačem a šíří se v AS.

Existují i další, ale již méně časté typy jako např. LSA typu 7.

  - *LSA typ 7 (NSSA External LSA)* - je generován ASBR v NSSA. Šíří se pouze v dané oblasti. Jakmile se dostane na ABR dané oblasti, je LSA typu 7 převedena na LSA typu 5 a distribuována dále do OSPF.

Druhy oblastí - rozdělení OSPF

OSPF bývá využíván i ve větších sítích. Jeho výhodou je, že jej lze rozdělit do různých oblastí dle potřeb. Sníží se tím výpočty nejkratší cesty, zmenší se směrovací tabulky a sníží zasílání updatů.

- **Páteřní oblast neboli area 0** - v případě, že je v OSPF síti zahrnuto více oblastí, platí povinnost, že jedna z oblastí musí být páteřní neboli area 0. Páteřní oblast musí být ve středu všech ostatních oblastí, nejlépe tak, aby byly všechny oblasti fyzicky připojeny. Je to z toho důvodu, že OSPF předpokládá, že všechny oblasti budou zasílat svoje směrovací informace do páteřní oblasti a ta je poté bude šířit mezi ostatní. [28]
  - *Virtuální linky* - v případě, že nějaká oblast nemůže být fyzicky připojena k páteřní oblasti, využívá se takzvaných virtuálních linek. Virtuální linka musí být vytvořena mezi dvěma ABR směrovači, které patří do stejné oblasti a jeden z nich musí být zároveň připojen k oblasti 0.
- **Stub oblast** - je charakteristická tím, že má pouze jednu cestu ven z dané oblasti, a to do páteřní oblasti. Není zde povolena redistribuce z jiných protokolů do OSPF.

  Nastavením oblasti jako „Stub“ snížíme paměťovou náročnost na směrovače uvnitř této oblasti. Nevýhodou naopak je, že nemůže být použita jako transitní oblast pro virtuální linky. A stejně tak v oblasti Stub nenalezneme ASBR směrovače.

  Pro sestavení Stub oblasti je nutné, aby všechny směrovače uvnitř byly nastaveny jako Stub. Ve Stub oblasti se šíří LSA zprávy typu 1, 2, 3 a nešíří se v ní LSA typu 4, 5. Díky posílání LSA 3 je možné, aby směrovače ve Stub oblasti byly schopny směrovat pakety do vnějších destinací bez potřeby uchovávat všechny jednotlivé vnější cesty.
- **Totally Stubby oblast** - liší se právě v šíření LSA zpráv. Nešíří LSA typu 4, 5 a ani 3. Šíří pouze 1 a 2. To znamená, že veškerá odchozí data jsou směrována jedinou výchozí cestou stanovenou ABR.
- **Not-so-stubby oblast** - je to oblast, která má z pohledu OSPF pouze jednu cestu ven, ale může mít cestu ven přes jiný směrovací protokol. NSSA obsahuje právě ASBR. V NSSA se šíří LSA zprávy typu 7, které jsou generovány ASBR a ty jsou poté na ABR konvertovány na LSA typu 5 a šířeny dále do OSPF. [29]

Změny u OSPFv3

Verze 3 je vyhrazena pro IPv6. Od verze 2, která je pro IPv4, se příliš neliší, avšak je mezi nimi pár důležitých rozdílů. Čerpáno bylo z [30].

- Ke změnám došlo u LSA zpráv. Význam LSA Typu 1 a 2 byl pozměněn a došlo k přidání dvou nových typů LSA, typu 8 (Link LSA) a typu 9 (Intra-area Prefix LSA). Jednou z největších výhod je vyloučení IP adresování z výpočtů Shortest Path First stromu. Typ 9 nově přebírá správu informací uvnitř oblasti, kterou měla dříve na starost zpráva Typu 2. Je to způsobeno IP adresováním, které probíhá nezávisle na LSA zprávách používaných pro kalkulaci nejkratšího stromu a přidávání nebo upravování podsítí uvnitř OSPF. Neovlivní to tak integritu stromu a nenutí provádět přepočítávání.
- Další změnou související s příchodem nové LSA zprávy - Typu 8 je vytváření sousedství. U IPv4 byly využívány adresy na portech a byly propagovány jako součást zprávy Typu 1. IPv6 u OSPFv3 používá lokální linkové adresy pro navázání sousedství. Jsou distribuovány v konkretní oblasti právě pomocí Typu 8 (Link LSA).
- Změna nastala i u číslování LSA. U OSPFv3 je možné z čísla LSA zprávy zjistit její dosah (kam až je zasílána). Rozhodující jsou první dva bity. Zpráva začínající 0x0 je vázána na lokální linku (**0**x**0**008 - Typ 8). Zpráva 0x2 je rozesílána uvnitř oblasti (**0**x**2**004 - je označení pro Typ 4) a zprávy začínající 0x4 jsou rozesílány v celém AS (**0**x**4**005 - pro Typ 5).
- Rozdílná je podpora více instancí OSPF na jedné společné lince neboli paralelnost.
- Lepší autentizace zpráv na základě IPsec.

2.3.3 BGP (Border Gateway Protocol)

BGP neboli Border Gateway Protocol je dynamickým protokolem typu path-vector. Řadí se mezi protokoly EGP (Exterior Gateway Protocol), tedy mezi autonomní systémy. Jeho úkolem je dosažení flexibility, možnosti snadné volby propojení a výměny směrovacích tabulek mezi AS. Je založen na použití TCP spojení na portu 179. [31]

BGP se dělí na externí a interní. Vazba mezi BGP směrovači v různých AS je nazývána eBGP (External Border Gateway Protocol). Vazba mezi BGP směrovači uvnitř stejné AS je iBGP (Internal Border Gateway Protocol). Liší se pak v typech zpráv, které mezi sebou směrovače posílají a ve směrovací politice.

Základní princip BGP

Na začátku komunikace dojde k výměně kompletních směrovacích informací ve formě záznamu NLRI (Network Layer Reachability Information). Dle potřeby si směrovače vyměňují také aktualizace. Pokud dojde k rozpadu TCP spojení, vyhodnotí to směrovače jako ztrátu dosažitelnosti a odstraní ze svých směrovacích tabulek všechny cesty, které byly přes tento uzel dostupné.

AS přes BGP sděluje sousedním AS, jakým IP sítím je schopen doručit IP datagramy a přijímá informace o jiných sítích přes sousední AS. Rozhoduje se na základě směrovací politiky, který sousední AS k dané síti využije. [31]

BGP zprávy

Protokol BGP používá 4 typy zpráv:

- **Open** - slouží k zahájení BGP komunikace mezi dvěma směrovači.
- **Keepalive** - využívá se k udržení BGP komunikace, dává sousednímu směrovači najevo, že spojení je stále funkční.
- **Update** - zpráva nese samotnou směrovací informaci NLRI. Obsahuje i informaci o sítích, které přestaly být použitelné a mají být odstraněny.
- **Notification** - obsahuje chybové hlášení o vypovězení nebo zamítnutí BGP spojení.

Stavy směrovače v BGP

Sousední směrovače mezi sebou mohou mít až 6 typů stavů v protokolu BGP.

- **Idle** - stav, kdy směrovač odmítá spojení nebo při chybě. Lze jej nastavit i ručně.
- **Connect** - BGP proces, při kterém dochází k rozpoznání příchozího TCP spojení a směrovač čeká na dokončení tohoto spojení. Po dokončení přejde do OpenSent.
- **Active** - stav, kdy BGP na směrovači zahájí TCP spojení a čeká na dokončení „3-way handshake“. Po dokončení přejde do OpenSent.
- **OpenSent** - stav, kdy TCP spojení existuje a BGP Open zpráva byla zaslána sousedovi, ale odpovídající Open zpráva od sousedního směrovače ještě nebyla přijata.
- **OpenConfirm** - Open zpráva od souseda byla obdržena a čeká pouze na Keepalive zprávu.
- **Established** - obousměrné spojení je vytvořeno a může dojít k zasílání Update a Keepalive zpráv.

Směrovací politika

BGP je postaven na algoritmu path-vector,díky kterému lze nalézt nejkratší cesty do všech AS. Pro explicitní ovlivňování směrovací politiky je nutné použít mechanismus, který by vyjádřil preferenci na základě kriterií. K tomuto účelu je v protokolu BGP využito tzv. atributů. Atributy mají odlišnou důležitost a dělí se do čtyř kategorií [32]:

- **Povinné atributy** (well-known mandatory) - musí být povinně připojeny ke každé cestě a každá implementace BGP jim musí rozumět.
- **Volitelné atributy** (well-known discretionary) - každá implementace BGP jim musí rozumět, avšak jejich připojení k cestě není povinné.
- **Transitivní** (optional transitive) - implementace BGP jim nemusí rozumět. V případě, že jim nerozumí, předává se atribut dále beze změny.
- **Netransitivní** (optional nontransitive) - implementace BGP jim nemusí rozumět. V případě, že jim nerozumí, atribut se dále nepředává.

Seznam některých atributů v rámci Cisco zařízení je uveden v Tab. 2.3.

Tab. 2.3: Seznam nejvyužívanějších atributů [32]

Atribut	Typ
WEIGHT	Cisco proprietární
ORIGIN	Povinný
AS_PATH	Povinný
NEXT_HOP	Povinný
LOCAL_PREF	Volitelný
ATOMIC_AGGREGATE	Volitelný
AGGREGATOR	Transitivní
COMMUNITY	Transitivní
MULTI_EXIT_DISC (MED)	Netransitivní

Význam jednotlivých atributů je popsán níže.

- **WEIGHT** - Povinný atribut, který je lokální (neposílá se sousedním směrovačům), vyšší hodnota znamená vyšší preferenci.
- **ORIGIN** - Určuje původ příchozí aktualizace. Nejlepší je, přichází-li aktualizace z IGP (Interior Gateway Protocol), poté z EGP (Exterior Gateway Protocol) a jako poslední Incomplete (cesta redistribuovaná do BGP).
- **AS_PATH** - Série AS čísel, přes které vede cesta k cíli, přičemž každý router přidá svoji AS. Slouží k zabránění smyčkám (pokud cesta již je v seznamu, je odmítnuta). Čím je daná cesta kratší, tím lépe.

- **NEXT_HOP** - Udává IP adresu „next-hop“ směrovače, která se používá pro dosažení cíle.
- **LOCAL_PREF** - Tímto atributem je možno upřednostnit výstup z AS. Čím vyšší je jeho hodnota, tím lépe.
- **ATOMIC_AGGREGATE** - Upozorňuje sousední AS, že směrovač sumarizoval cesty.
- **AGGREGATOR** - Specifikuje IP adresu a číslo AS směrovače.
- **COMMUNITY** - Používá se pro filtrování příchozích nebo odchozích cest a umožňuje jejich označení. Používá se také pro cílové cesty, které mají některé vlastnosti stejné a mají stejnou směrovací politiku.
- **MULTI_EXIT_DISC (MED)** - Slouží pro ovlivnění výběru vstupního bodu do AS. Upřednostňovány jsou ty vstupy do AS, které mají nastaveny MED atributy na nižší hodnoty.

2.4 BFD (Bidirectional Forwarding Detection)

Protokol BFD byl navržen tak, aby rychle rozpoznal výpadky v komunikaci na cestě mezi systémy. Dokáže je rozpoznat na jakémkoliv typu spojení nezávisle na médiu. Pracuje pod libovolným datovým protokolem (síťové vrstvy, linkové vrstvy, tunely atd.) mezi dvojicí sousedních směrovačů. Využívá unicast přenos a *point-to-point* mód. Pakety jsou přenášeny v datové části bez ohledu na použitý zapouzdřovací protokol. [33]

BFD funguje jako jednoduchý „Hello protokol“, který je podobný detekčním částem dobře známých směrovacích protokolů. Jinými slovy, sousední směrovače naváží komunikaci a periodicky si zasílají zprávy. Jestliže po určité době nedorazí odpověď, můžeme linku považovat za přerušenou a příslušný protokol, pro který byl BFD protokol v provozu, je o tom informován.

Podstatný je i fakt, že BFD protokol nevyužívá svůj vlastní objevovací mechanismus, protože využívá objevovací mechanismus protokolu ke kterému je přiřazen. Např. u OSPF je pro navázání BFD sousedství použit OSPF „Hello protokol“.

2.4.1 Operační módy

Protokol BFD má dva operační módy. Asynchronní mód a mód na vyžádání. K oběma režimům lze navíc přidat funkci ozvěny (echo). Jako zdroj byl použit [33].

Asynchronní mód

V tomto módu si systémy periodicky zasílají BFD kontrolní pakety. Pokud část těchto paketů v řadě není obdržena prostřednictvím druhého systému, je spojení

označeno za nefunkční.

Mód na vyžádání

V tomto režimu se předpokládá, že systém má nezávislý způsob jak ověřit, zda je připojen k druhému systému. Po navázání BFD spojení může jeden systém požádat ten druhý, aby zastavil odesílání BFD kontrolních paketů. Výjimkou je, kdy systém potřebuje explicitně ověřit konektivitu. V takovém případě se vymění krátká sekvence BFD kontrolních paketů. Pokud protější strana odpoví, komunikace opět utichne.

Tento mód se využívá v situacích, kdy by velký počet periodické komunikace mohl být na obtíž. Např. u systému s velkým počtem BFD relací.

Přídavná funkce Echo

Je-li funkce Echo aktivní, přenáší se proud paketů k druhému systému a ten odpovídá na tyto pakety zpátky stejnou cestou (komunikace ve smyčce). Pokud odpovědi nepřijdou zpět k prvnímu systému, linka je označena za nefunkční. Echo funkce může být povolena i pouze v jednom směru.

V případě asynchronního módu dojde ke snížení periodického zasílání kontrolních BFD paketů, protože Echo funkce zajistí úlohu rozpoznání protějšku. Samostatný asynchronní mód má však výhodu v tom, že na rozdíl od Echo funkce potřebuje pouze poloviční počet paketů k tomu, aby detekoval změnu. Echo funkce avšak zase skutečně ověří danou cestu k protějšku a rozpozná některé typy výpadků, které by samostatný asynchronní mód nerozpoznal.

U módu na vyžádání jsou kontrolní BFD pakety zcela eliminovány a detekce zajištěna přes Echo funkci.

2.5 QoS (Quality of service)

QoS neboli Quality of service je velmi obsáhlé téma, proto se tato podkapitola zaměřuje pouze na ty části, které jsou pro tuto práci relevantní.

QoS představuje schopnost sítě zajišťovat lepší služby vybraným přenosům prostřednictvím různých technologií. Nejrozšířenější jsou především tyto tři typy mechanismu QoS:

- **Best-effort** - metoda největšího úsilí. Tato metoda se snaží každý paket co nejrychleji a nejefektivněji přenést k cíli, bez jakýchkoliv záruk. Tato metoda má nulové QoS. [27]
- **Mechanismus Integrovaných služeb (IntServ)** - vychází z modelu, kdy je před přenosem zajištěna potřebná kvalita přenosového kanálu. Aplikace

oznámí počítačové síti svoje požadavky na přenos dat ve formě QoS. Počítačová síť ověří, zda k tomu má k dispozici prostředky a rozhodne, zda je schopna požadavkům vyhovět nebo zda je aplikace musí snížit.

- **Mechanismus Diferencovaných služeb (DiffServ)** - jedná se o nejrozšířenější řešení pro zajištění kvality služeb. Tento mechanismus je využíván i v praktické části této práce. Jeho princip spočívá v rozdělení síťového provozu do několika tříd a zajištění odlišného zacházení s jednotlivými třídami. Označování paketů probíhá pouze při vstupu do sítě.

2.5.1 Diferencované služby (DiffServ)

Dále je již popsán pouze mechanismus Diferencovaných služeb neboli DiffServ se zaměřením na protokoly IPv4 a IPv6 a s použitím na Cisco zařízeních.

Proces klasifikace paketů začíná na vstupním směrovači do sítě. Výběr značení může být proveden na základě několika faktorů (IP adresa odesílatele nebo adresáta, číslo portu apod.) nebo mohou být pakety již klasifikovány aplikací, která je posílá do sítě. V této práci byla použita druhá z možností.

Způsob značení paketů se rozlišuje dle použité technologie. U IPv4 je značka obsažena v záhlaví IPv4 datagramu v poli „Typ služby“ (dříve toto pole obsahovalo IP precedence). V případě IPv6 je značka v záhlaví IPv6 datagramu v poli „Třída provozu“. Pole, ve kterém bývá značka uložena, má 8 bitů a je nazýváno DS . V poli DS představuje 6 bitů značka DSCP neboli Differentiated Services Codepoint, zbylé 2 bity jsou rezervované pro budoucí využití. [27]

V jednotlivých směrovačích dochází k zacházení s pakety bez ohledu na ostatní směrovače, tzv. PHB-per-hop-behaviour. Toto chování lze rozdělit na: [35]

- *Default PHB* - využívá metodu best-effort.
- *Expedited Forwarding (EF) PHB* - jinak také urychlené posílání. Pro pakety, jež vyžadují malé zpoždění, kolísání a nízkou ztrátovost.
- *Assured Forwarding (AF) PHB* - neboli zajištěné posílání. AF chování zařazuje pakety do jedné ze čtyř tříd. Jednotlivým třídám je ve směrovačích přidělen určitý objem prostředků. V rámci tříd je každému paketu přiřazena jedna ze tří priorit zahození paketů, v případě že by došlo k zahlcení. Nejprve mají přednost pakety s nižší hodnotou v poli priorita.
- *Class Selector PHB* - zajišťuje zpětnou kompatibilitu s klasifikováním dle IP precedence.

V Tab. 2.4 jsou vypsány nejčastěji využívané hodnoty v poli DS.

Tab. 2.4: Seznam nejčastěji používaných DSCP hodnot [36]

DSCP hodnota	Desítkový zápis	Značení	Pravděpodobnost zahození
101 110	46	Expedited forwarding (EF)	-
000 000	0	Best effort	-
001 010	10	AF11	Nízká
001 100	12	AF12	Střední
001 110	14	AF13	Vysoká
010 010	18	AF21	Nízká
010 100	20	AF22	Střední
010 110	22	AF23	Vysoká
011 010	26	AF31	Nízká
011 100	28	AF32	Střední
011 110	30	AF33	Vysoká
100 010	34	AF41	Nízká
100 100	36	AF42	Střední
100 110	38	AF43	Vysoká
001 000	8	CS1	-
010 000	16	CS2	-
011 000	24	CS3	-
100 000	32	CS4	-
101 000	40	CS5	-
110 000	48	CS6	-
111 000	56	CS7	-

2.6 SNMP (Simple Network Managment Protocol)

SNMP je standardizovaný protokol pro správu zařízení v síti za pomocí sady protokolů TCP/IP. Poskytuje základní operace pro sledování a údržbu sítě prostřednictvím výměny zpráv. SNMP zpráva je paket zasílaný přes Ethernet UDP/IP na portu 161.

SNMP je protokolem aplikační vrstvy. Je vytvořen tak, aby mohl monitorovat zařízení od různých výrobců instalovaných na rozdílných fyzických zařízeních. [23]

Existují tři verze:

- **SNMPv1** - verze postrádající prostředky pro komunikaci mezi manažery. Agent není schopen dodávat manažerovi detailnější informace. Hlavním nedostatkem je však slabé zabezpečení.
- **SNMPv2** - v této verzi došlo k mnoha vylepšením jako je schopnost komuni-

kace mezi manažery, zvýšená bezpečnost nebo schopnost požadavku na větší množství dat. Z důvodu velké složitosti v2 (označované i 2c) v oblasti zabezpečení byla vytvořena verze Community-Base, označována v2c. Tato podverze však využívá zabezpečení SNMPv1, které je nižší.

- **SNMPv3** - verze, která definuje koexistenci SNMPv1, v2c a v3. Přináší velkou míru zabezpečení díky SHA, MD5 autentizačním protokolům a DES56 šifrování.

2.6.1 Manažeři a agenti

SNMP využívá koncept manažerů a agentů. Manažer bývá obvykle host s klientským SNMP programem, který dohlíží na skupinu směrovačů nebo serverů (agentů) využívajících SNMP serverový program. Protokol je založen na jejich vzájemné interakci.

Agent uchovává informace o výkonu v databázi a manažer k nim má přístup. Manažer také může přikázat agentovi provést určité akce. Agenti se také podílejí na procesu řízení. Serverový program běžící na agentovi provádí kontrolu systému a pokud si všimne něčeho neobvyklého, může poslat manažerovi varovnou zprávu (nazývanou *trap*). [23]

Součásti managementu

Pro vykonávání správy využívá SNMP dva další protokoly: SMI (Structure of Management Information) a MIB (Management Information Base).

- **Role SNMP** - SNMP definuje formát paketů vyměňovaných mezi správcem a agentem. Čte a mění stav objektů (hodnoty proměnných) v SNMP paketech.
- **Role SMI** - SMI definuje obecná pravidla pro pojmenování a definování objektů a ukazuje, jak je kódovat.
- **Role MBI** - MIB vytváří strukturu pojmenovaných objektů, jejich typů a vzájemných vztahů v zařízení, které je spravováno. Do struktury přistupujeme pomocí SNMP.

2.6.2 Typy SNMP operací

SNMP definuje 8 typů operací [23]. Jedná se o příkazy sloužící ke komunikaci mezi manažerem a agentem nebo manažerem a manažerem:

- **GetRequest** - požadavek od manažera k agentovi pro načtení proměnné.
- **GetNextRequest** - požadavek od manažera k agentovi pro zjištění dostupných proměnných a jejich hodnot.

- **GetBulkRequest** - požadavek od manažera k agentovi pro více interakcí GetNextRequest.
- **SetRequest** - požadavek od manažera k agentovi na změnu hodnoty proměnné.
- **Response** - odpověď agenta k manažerovi na *GetRequest* nebo *GetNextRequest*. Obsahuje hodnoty vyžádané manažerem.
- **Trap** - zpráva zaslaná od agenta k manažerovi s hlášením o události.
- **InformRequest** - zpráva zaslaná od jednoho manažera k druhému manažerovi s požadavkem na zaslání proměnných od agentů pod jeho správou.
- **Report** - nahlášení chyby mezi manažery (není využíváno).

3 Emulace transportní sítě

Tato kapitola popisuje softwarové a hardwarové prostředky využité pro emulaci transportní sítě.

Jelikož investice do reálných síťových zařízení není levná záležitost, byly vyvinuty síťové simulátory a emulátory, které umožňují testovat chování sítě bez nutnosti nákupu fyzických síťových zařízení.

Na začátku této kapitoly zmíníme a popíšeme dostupné emulátory. Poté se budeme zabývat hypervisory a konkrétním použitým hardwarem včetně popisu jeho instalace.

Následně zde budou popsány dvě volně dostupná prostředí pro emulaci, které dominují trhu a byly zvoleny jako nejideálnější řešení. Podkapitoly obsahují popis instalace samotných prostředí a importování emulátorů. Jejich rozdílnost, výhody a nevýhody pro emulaci transportní sítě budou porovnány v Kap.4.

Dále pak tato kapitola řeší otázku výběru již konkrétních virtuálních zařízení pro potřeby transportní sítě. Stejně tak si klade otázku, které pomocné programy je potřeba použít a nainstalovat k emulaci transportní sítě.

Pro tuto práci není uvažována varianta instalace virtuálních zařízení přímo na hardware, např. na server, bez nutnosti použití programů. Je to dáno potřebami koncových uživatelů vytvořené emulace transportní sítě. Ti tuto variantu nebudou používat.

3.1 Emulátory

Emulátory jsou hardwarová nebo softwarová zařízení, která umožňují počítačovému systému (jinak také *hostitelský systém*) imitovat funkce jiné počítačové platformy (jinak také *host*). Umožňují hostitelskému systému spustit software, nástroje, periferní zařízení atd., které jsou pro systém hosta vytvořeny. Emulace je speciálním případem virtualizace. [7].

V této práci jsou použity programy EVE-NG (Emulated Virtual Environment – Next Generation) a GNS3 (Graphical Network Simulator 3), které využívají emulátory. Jsou graficky velmi názorné, díky čemuž lze snadno a podle potřeby měnit topologii, testovat provoz nebo zkoušet chování protokolů ve specifických podmínkách.

Oba programy podporují stejné emulátory:

Dynamips

Dynamips je emulační program, který slouží k emulaci Cisco hardwarové platfomy, konkrétně série 1700, 2600, 2691, 3600, 3725, 3745 a 7200. Je podporován operačními systémy FreeBSD, Linux, Mac OS X nebo Windows. Umožňuje emulovat hardware Cisco zařízení tak, že je schopen přímo spustit Cisco IOS v emulátoru. [8] V roce 2007 byl již jeho oficiální vývoj zastaven, nicméně je podporován ze strany GNS3 a dobrovolníků. V dnešní době je však považován za starší technologii.

IOU/IOL

IOU (IOS na Unixu) nebo IOL (IOS na Linuxu) je simulátor určený pouze pro proprietární užití firmou Cisco. IOL je kompilovaný pro architekturu i386. Naopak IOU odkazuje na Unixovou (Solaris) verzi kompilovanou pro Sparc architekturu. [9]

Použití IOU/IOL je limitováno licencí, kterou může vystavit pouze Cisco. Bez přítomnosti *iourc* licence nelze IOU/IOL image spustit.

Jedná se především o obrazy L2 a L3 přepínačů. Jelikož zde chybí oficiální podpora ze strany Cisco pro využití programů jako jsou GNS3 a EVE-NG, jsou obrazy často chybové. Řešením se jeví IOSvL2, který je podporován, nicméně jeho nevýhodou je větší hardwarová náročnost.

QEMU

QEMU je generický open-source emulátor pro hardwarovou a softwarovou virtualizaci. [10] Má dva operační módy:

- Emulace v uživatelském režimu (emulátor) – QEMU spustí OS a programy vytvořené pro jedno zařízení (např. ARM vývojovou desku) na jiném zařízení (PC). Použitím dynamického binárního překladu dosahuje dobrého výkonu. Je schopen provádět několik virtuálních CPU a také více vláken paralelně.
- Kompletní emulace systému (virtualizér) – QEMU dosahuje téměř nativního výkonu[1] tím, že spustí kód hosta přímo na hostitelském CPU. Je podporován u hypervisoru Xen nebo v modulu jádra KVM v Linuxu. V tomto módu používá jedno vlákno pro emulování všech virtuálních CPU a hardwaru.

Díky trendu virtualizace nabízí Většina výrobců obrazy v podobě QEMU. Příkladem jsou Cisco ASA (XRv, IOSv L3-L2, CSR, Firepower), Juniper vMX, Alcatel 7750SR, Huawei USG6000v, PaloAlto FW a další. Pro emulaci síťových zařízení se více doporučuje QEMU, než starší Dynamips.

[1] Tzn. výkon, který se dosahuje na platformě, pro kterou byl program napsán.

VPCS

Jedná se o emulátor osobního počítače. Emulace není paměťově náročná. PC je bez GUI a je zde možnost pouze jednoduchých příkazů typu `ping`.

Dalšími podporovanými emulátory jsou např. **Vmware** a **Docker**.

Přes veškerou snahu mají emulační prostředí a emulátory svá omezení. Je to z toho důvodu, že jejich cílem není zařazení do produkční sféry, ale pouze vzdělávací funkce. Z pohledu výrobců je to pochopitelné. Pokud by emulátory plně nahrazovaly funkci reálných zařízení, docházelo by k jejich zneužití.

Emulátory nejsou schopny nahradit ASIC hardware, který je typický pro Cisco Catalyst přepínače, což způsobuje komplikaci při jejich emulování. V dnešní době již však existuje několik řešení. Prvním řešením při užití emulátoru Dynamips je přidání virtuálního přepínacího modulu, např. NM-16ESW, k jednomu z virtuálních směrovačů, čímž se vytvoří L3 přepínač. Druhou variantou je pak použití IOU/IOL obrazů nebo QEMU IOSv, které poskytují při konfiguraci více možností než přepínací modul u Dynamips. Třetím řešením je poté zakomponování reálného přepínače do topologie, což prostředí pro emulaci umožňují.

Další omezení se týká výkonu sítě. U emulátoru Dynamips není podporována hardwarová akcelerace a propustnost je limitována od 1,5 Mb do 800 Mb za sekundu, dle použitého IOS obrazu. [11] U QEMU je ve většině případů propustnost limitovaná licencí, opět aby nedocházelo k zařazení edukační verze obrazů do produkční sféry.

Veškeré tyto nedostatky mají nicméně pouze minimální vliv na studijní a testovací použití.

3.2 Hypervisor

Dnešní doba je příznivá pro síťové inženýry. Pomocí virtualizace a emulátorů zmíněných výše se dá vyvarovat nákupu reálných zařízení. Jedinou překážkou je nutnost dostatečné kapacity výpočetních prostředků, jelikož emulace/virtualizace reálných zařízení je dosti náročná.

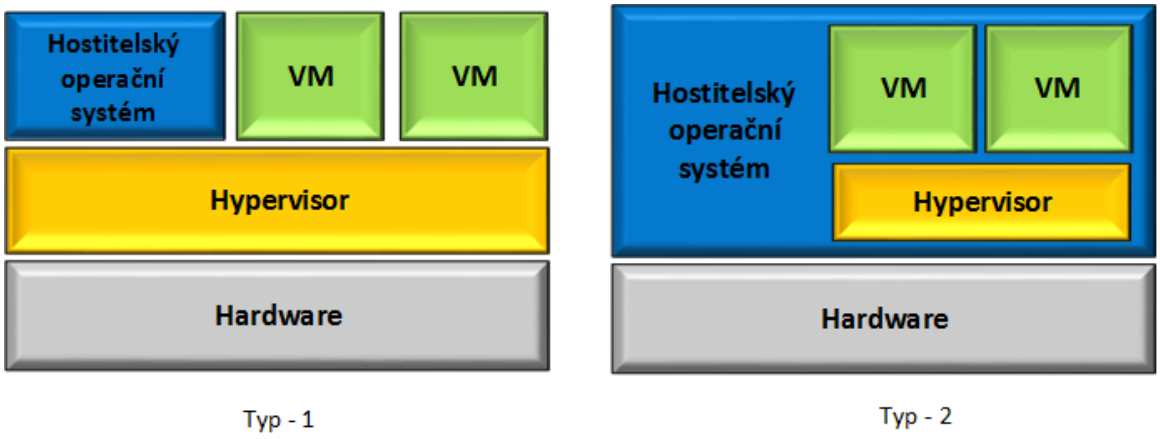
Ve velkých virtuálních laboratořích se využívá serverů, na nichž je pomocí hypervisorů možno rozdělit fyzický hardware na více virtuálních zařízení. Hypervisory se dělí na dva typy: [14]

Typ 1 – nativní

Hypervisor je spuštěn přímo na hostitelském hardwaru. Často je nazýván jako „bare metal“. Chová se jako vlastní operační systém a jednotlivá virtuální zařízení fungují přímo pod hypervisorem, viz Obr. 3.1 Typ-1. Díky eliminaci hostitelského operačního systému dosahují tyto typy hypervisorů výborného výkonu a podporují velké množství virtuálních zařízení. Příkladem je ESXi od firmy VMware.

Typ 2 – hostovaný

Hypervizor běží na hostitelském operačním systému. Jednotlivá virtuální zařízení pracují pod hypervisorem. Pokud chce mít virtuální zařízení přístup k hardwaru, musí projít skrz hypervisor i operační systém, viz Obr. 3.1 Typ-2. To způsobuje nižší výkon. Příkladem je VMWare Workstation nebo VirtualBox.



Obr. 3.1: Typy hypervisorů: Typ 1 - nativní, Typ 2 - hostovaný

Z jednoznačných důvodů je pro tuto práci vhodnější využít nativní hypervisor, a to i přesto, že vyžaduje specifické požadavky na hardware jako je podpora virtualizace v CPU a BIOSu. Proto je pro praktickou část této práce vyhrazen herní počítač bez operačního systému, splňující zmíněné požadavky, na který je nainstalován ESXi od VMwaru.

3.2.1 Specifikace vlastního zapojení

Vlastní zapojení je tvořeno z počítače o následujících parametrech:

- Základová deska: ASUSTeK Computer INC. P7H55-M
- Procesor: Intel(R) Core(TM) i5 CPU650 @3.20GHz
- Operační paměť (RAM): 16 GB (4x4GB) Kingston Genesis DDR3
- Interní paměť (ROM): 1 TB Samsung HD103SJ

- Grafická karta: ATI Radeon HD 5600 Series 1 GB
- Síťová karta: Realtek 8111E Gigabit LAN Controller

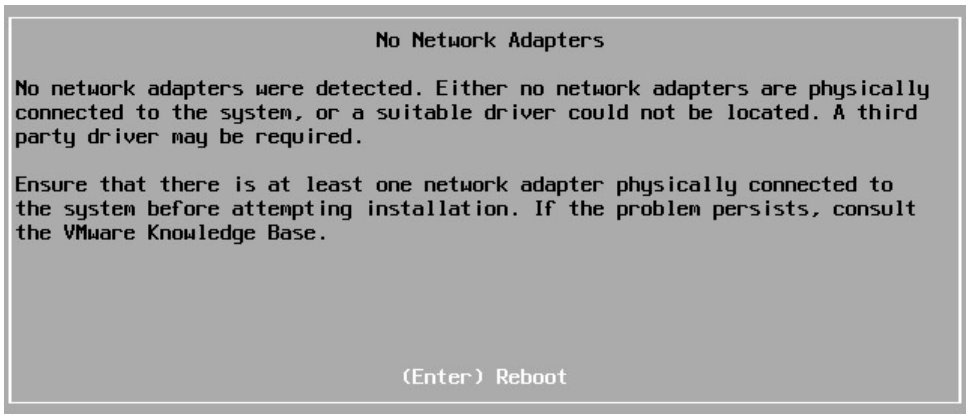
Díky podpoře virtualizace u této konfigurace byl nainstalován na PC nativní hypervisor, tedy ESXi od VMwaru.

ESXi je freeware, Typ-1 („bare metal“) hypervisor, který se instaluje přímo na fyzické zařízení bez operačního systému. Standardně bývá využíván na serverech, nicméně díky již zmiňované podpoře virtualizace u PC bylo možné ESXi nainstalovat na daný PC a vytvořit z něj server pro virtuální stroje.

Instalace ESXi

Instalace ESXi na osobní PC není jednoduchou záležitostí, jelikož se jedná o nestandardní implementaci. VMware udává na oficiálních stránkách[2] seznam podporovaného hardwaru. Osobní PC nejsou v tomto seznamu vůbec zahrnuty.

ESXi má několik verzí, z nichž je nejaktuálnější je verze 6.7. Při prvotní instalaci verze 6.7 byla zahlášena chyba viz Obr. 3.2.



Obr. 3.2: Chybové hlášení - „No Network Adapters“

Stejné hlášení se objevilo i při testování starších verzí 6.5 a 6.0. K překonání této překážky bylo třeba si vytvořit „custom verzi“ (neboli vlastní) ESXi image. Originální image byl upraven tak, aby rozeznal síťovou kartu Realtek 8111E. Nejprve byla vyzkoušena verze 6.7, kde instalace custom verze nebyla úspěšná. Instalace verze 6.5 již byla dokončena úspěšně, ale systém při bootování vždy zamrzl. Ideální se ukázala nakonec být verze 6.0 a ta také byla na PC nainstalována.

PC s ESXi byla přidělena statická adresa 192.168.1.190 ve vnitřní síti. Pro připojení zvenčí byl nastaven PAT (Port Address Translation). Veřejná IP adresa je 93.99.192.179:443 (připojení přes https). Dále byl vytvořen login a heslo.

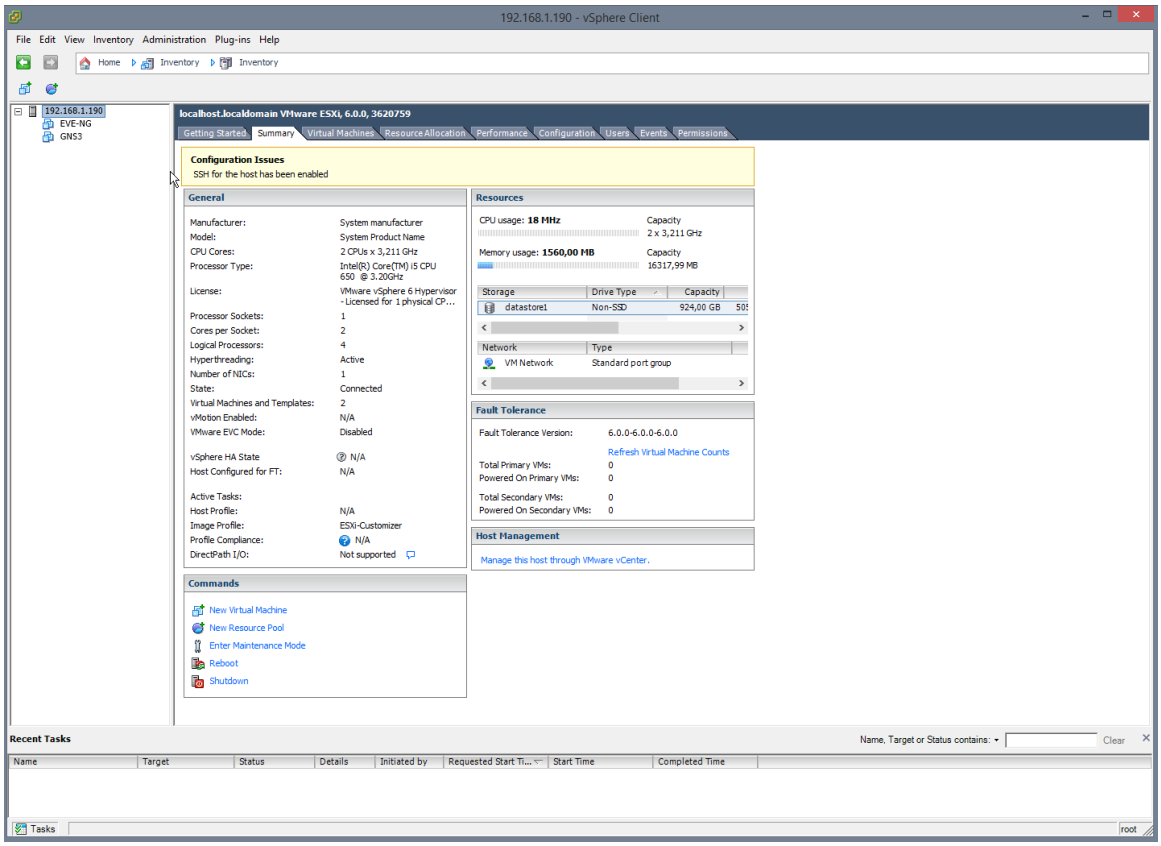
[2] www.vmware.com/resources/compatibility/pdf/vi_systems_guide.pdf

Klient pro správu serveru - vSphere klient

VSphere klient je grafické prostředí, které je nezbytně nutné pro správu ESXi serveru. Přes vSphere klienta je možné přidávat a odebírat virtuální stroje, přidělovat jim paměť, sledovat vytíženost serveru, apod.

Existují dvě možnosti jak vSphere klienta spustit. Obě možnosti jsou nabídnuty po připojení se k adrese serveru (192.168.1.190 nebo 93.99.192.179:443). První možností je stáhnout program přímo do vlastního PC (s Windows). Druhou variantou je využití webového klienta. Vždy je nutné zadat login a heslo.

Grafické prostředí u verze s instalací je vidět na Obr. 3.3. Webové prostředí je téměř totožné.



Obr. 3.3: Grafické prostředí vSphere klienta u verze s instalací

3.3 EVE-NG (Emulated Virtual Environment – Next Generation)

Jak již název napovídá, „Emulated Virtual Enviroment – Next Generation“ je emulační virtuální prostředí nové generace. Oficiálně bylo zpřístupněno veřejnosti v lednu

roku 2017. [9] Navazuje na předchozí projekt zvaný UnetLab, u kterého se vývoj zastavil a není nadále podporován. Veškeré aktualizace a záplaty jsou již implementovány pouze do EVE-NG.

EVE-NG má v roce 2019 k dispozici tři verze. První verze „Community“, která je zdarma. Pro potřeby této práce je dostačující. Druhá „Professional“ s sebou nese výhody jako vetší počet uzlů na laboratoř, možnost snadného importu a exportu konfigurace do lokálního PC, atd.. Třetí verze „Learning Center“ existuje pro studijní účely. Umožňuje vytvoření rolí studentům, kteří se mohou připojovat k laboratořím a nezávisle na sobě na nich pracovat. Administrátor/učitel může studentovi nastavit i časové omezení pro přístup.

EVE-NG poskytuje grafické uživatelské rozhraní prostřednictvím webového prohlížeče, viz Kap. 3.3.1. Uživatelé mohou vytvářet síťové uzly výběrem z knihovny šablon, propojit je a konfigurovat přes Putty či jiného protokolového klienta.

Webové prostředí tak umožňuje připojení k EVE-NG pomocí IP adresy z více počítačových stanic. Samotný EVE-NG může být nainstalován jako virtuální zařízení skrze hypervisor, např. VMware Workstation, Player nebo ESXi. Nebo může být nainstalován přímo na fyzický hardware bez hypervisoru, což je vývojáři nejvíce doporučovaná varianta, která zajistí větší výkon než varianta první. Ta je ale pro laboratorní prostředí dostačující. [12]

V případě, že EVE-NG funguje jako virtuální zařízení, může být nastaven na libovolném operačním systému (Windows, Linux nebo Mac OS). Při instalaci bez hypervisoru je nutné mít odpovídající hardware podporující Intel VT-x např. Intel Xeon CPU.

EVE-NG podporuje všechny emulátory zmíněné a popsané v předchozí kapitole.

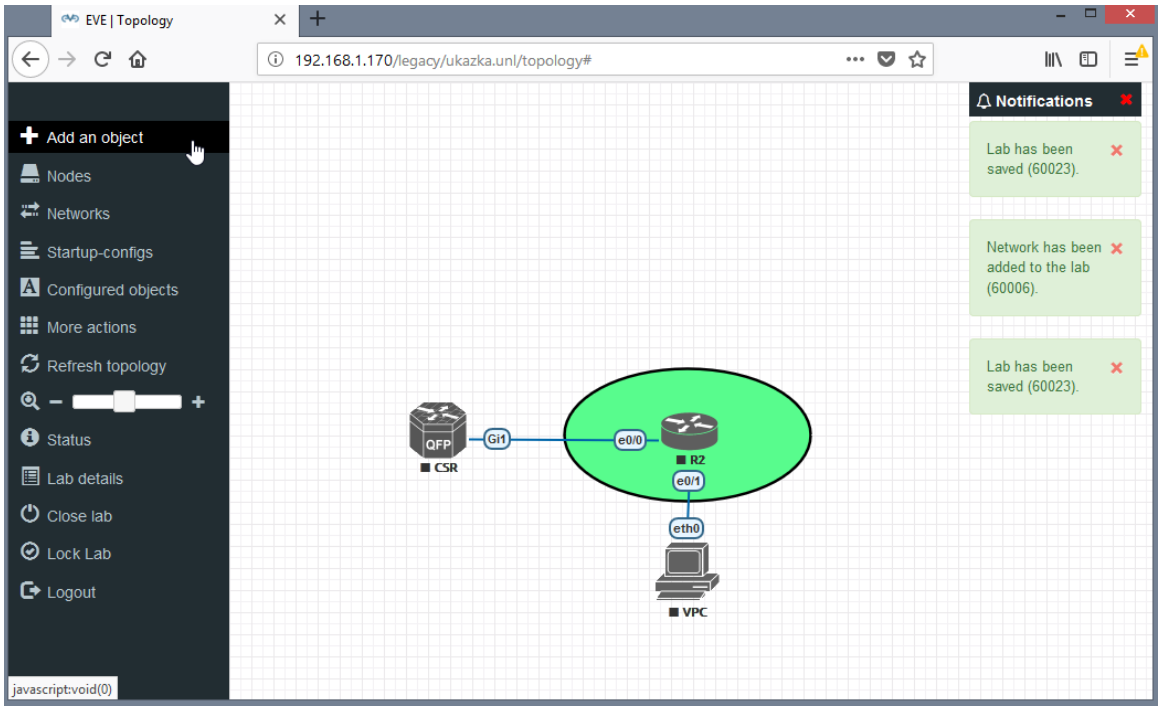
Z důvodu konkurování veřejnosti známějšímu GNS3, provozují vývojáři Live-Helpdesk, kde odpovídají 24 hodin denně na jakékoliv dotazy týkající se EVE-NG, od instalace až po výběr obrazů a konfiguraci.

3.3.1 Popis grafického prostředí

EVE-NG se ovládá přes webové prostředí. Příklad je možné vidět na Obr. 3.4.

Na levé straně se nachází rolovací lišta. Jsou zde možnosti jako:

- **Přidání objektů** - lze zde přidat uzel, síť (cloud, bridge nebo NAT), přidat obrázek, tvar nebo textové pole.
- **Uzly** - přehled uzlů použitých v laboratoři, možnost hromadného zapínání/vypínání uzlů.
- **Sítě** - případná správa sítě pokud je přidána.
- **Startup konfigurace** - v PRO verzi se ukáže „start-up“ konfigurace uzlů. V Community verzi není povolena.



Obr. 3.4: Grafické webové prostředí EVE-NG

- **Správa objektů** - přehled přidaných tvarů a obrázků.
- **Více možností** - umožňuje hromadné ovládání celé laboratoře.
- **Obnova topologie** - provede obnovu topologie - „refresh“.
- **Status** - udává zatíženost hardwaru, počet a typ běžících emulátorů a celkový přehled o laboratoři.
- **Zavřít laboratoř** - zavře pouze aktuální laboratoř a umožní jít do nabídky s jinými laboratořemi.
- **Uzamknout laboratoř** - uzamkne úpravu a laboratoř není možné editovat.
- **Odhlásit se** - odhlášení z celého EVE-NG.

V pravé části jsou upozornění, objevují se zde i chybová hlášení.

3.3.2 Vlastní instalace

EVE-NG v2.0.3-86 byl instalován na vytvořený ESXi v6.0 server přes vSphere klienta. Nejprve bylo nutné stáhnout instalační soubor ze stránek eve-ng.net a překopírovat jej na disk ESXi serveru. Při instalaci bylo vyhrazeno pro EVE-NG 16 GB RAM paměti a 300 GB ROM.

U prvního bootování EVE-NG bylo potřeba nastavit obecné informace (jazyk, login, heslo). Dalším krokem bylo přidělení IP adresy. Nastavení lze ponechat na DHCP serveru. V tomto případě byla síť nastavena ručně na IP adresu 192.168.1.170.

Po dokončení instalace je možné si otestovat konektivitu přihlášením se na zmíněnou adresu, zadat login a heslo.

Důležité je také nainstalovat „Windows Client integrated pack“ na PC, ze kterého chceme pracovat s laboratořemi v EVE-NG. Tento „balíček“ umožní připojení přes telnet, spojení s programem Wireshark atd. Balíček se nachází opět na stránkách eve-ng.net s možností varianty i pro Linux.

3.3.3 Nahrávání emulátorů

Nahrávání obrazů emulátorů je mírně komplikované. Nejideálnějším řešením se ukázalo být využití programu WinSCP. S tímto programem je nutné se připojit k EVE-NG přes SSH protokol a IP 192.168.1.170. Obrazy nejzákladnějších emulátorů se nacházejí v adresářích:

- IOL/IOS - /opt/unetlab/addons/iol/bin/
- Dynamips - /opt/unetlab/addons/dynamips
- Qemu - /opt/unetlab/addons/qemu

Uvnitř výše zmíněných adresářů je nutné správně pojmenovat složky a obrazy uvnitř. EVE-NG uvádí na stránkách[3] seznam pojmenování složek a obrazů.

Příklad pro Cisco směrovač CSR 1000v je následující:

- Název složky musí být „csr1000v-“ informace následující za „-“ již mohou být libovolné.
- Název obrazu uvnitř této složky musí být pro CSR 1000v „virtioa.qcow2“.
- Celková cesta pro CSR 1000v pak má podobu /opt/unetlab/addons/qemu/csr1000v-libovolnytext/virtioa.qcow2.

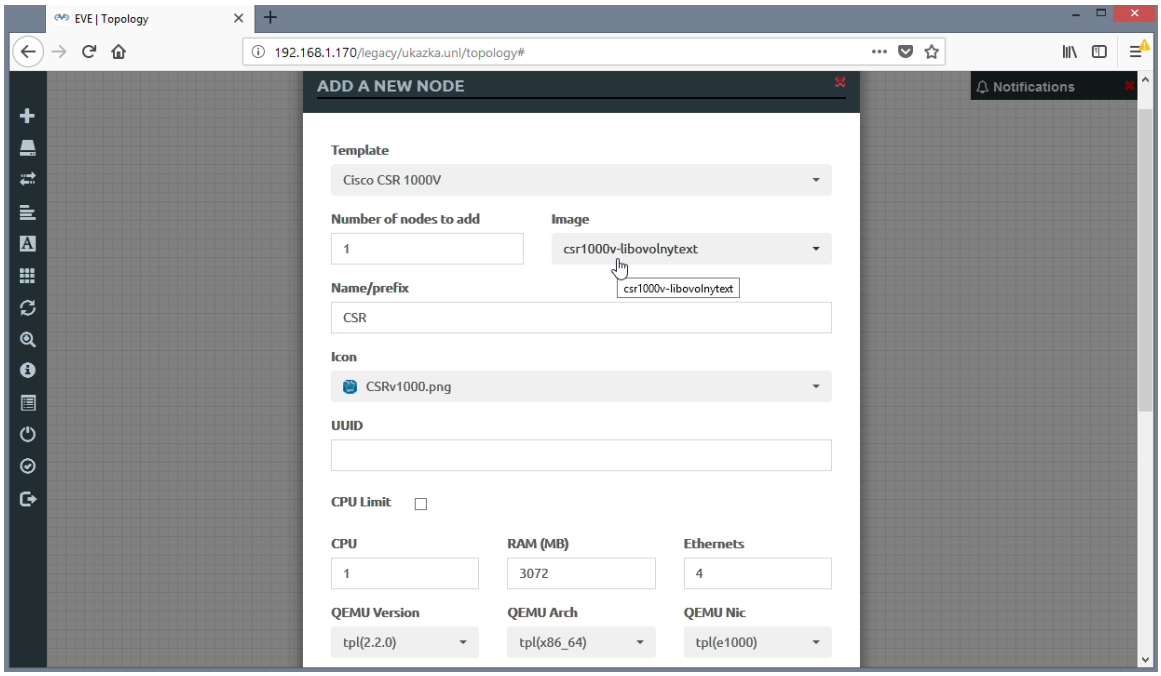
Po obnovení laboratoře v grafickém prostředí se obraz načte ve výběru pro přidání nových uzlů, viz Obr. 3.5.

3.4 GNS3 (Graphical Network Simulator-3)

„Graphical Network Simulator-3“, jinak také grafický síťový simulátor, je volně přístupný a poprvé byl spuštěn v roce 2008. [11] Jedná se o nejvíce používané emulační prostředí s velkou uživatelskou základnou. Aktivně je vyvíjen dobrovolníky a odborníky pohybujícími se v oblasti sítí.

GNS3 poskytuje grafické uživatelské prostředí pomocí softwaru GNS3-all-in-one. Vytvořená zařízení musí být hostována a spuštěna serverovým procesem. Nabízí se tři varianty. První je lokální GNS3 server, který běží na stejném PC, na kterém je nainstalován GNS3-all-in-one software. Všechny další procesy, jako je Dynamips,

[3] www.eve-ng.net/documentation/images-table



Obr. 3.5: Načtení nového obrazu v EVE-NG

jsou opět spuštěny na stejném PC. Druhá varianta je lokální GNS3 virtuální zařízení, které je nainstalováno na PC např. pomocí VMware Workstation nebo Player. Poskytuje více možností a dovoluje vytvořit složitější topologie s použitím QEMU zařízení. Třetí variantou je vzdálené GNS3 virtuální zařízení s využitím VMware ESXi nebo cloudu. [13] Třetí varianta je použita v této práci.

Grafické uživatelské rozhraní je podporováno Windows, Linuxem i MacOS.

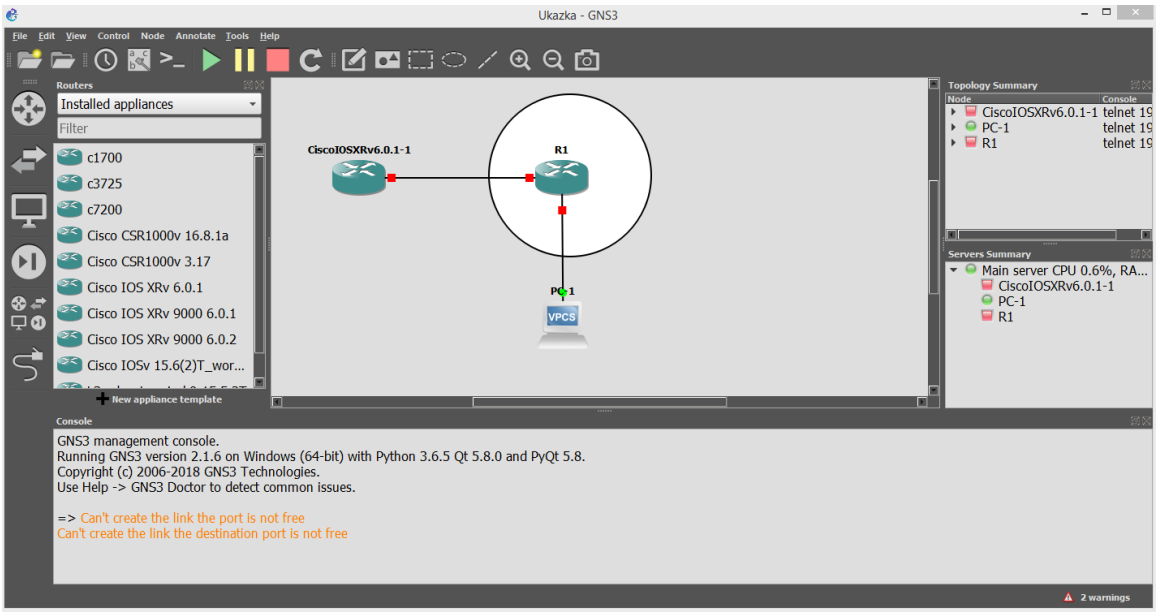
Stejně jako EVE-NG podporuje GNS3 všechny emulátory popsané v Kap. 3.1.

3.4.1 Popis grafického prostředí

GNS3 nemá webové grafické prostředí jako EVE-NG a je zapotřebí celý program nainstalovat na PC. Po spuštění a nastavení vypadá GUI (Graphical User Interface) viz Obr. 3.6.

V levé části se nachází panel pro přidávání zařízení (od směrovačů až po kabeláž). V horní části se nachází záložky:

- **File** - záložka pro vytváření nových projektů, ukládání, importování a exportování projektů, vkládání šablon atd.
- **Edit** - záložka slouží pro hromadnou úpravu projektu a nastavení GNS3, společně s vkládáním nových obrazů.
- **View** - správa zobrazení laboratoře v GNS3.
- **Control** - záložka pro hromadnou správu uzlů.
- **Node** - zobrazení veškerých možností nastavení pro zrovna označený uzel.



Obr. 3.6: Grafické prostředí u GNS3

- **Annote** - záložka umožňující vložení obrazců, čar nebo obrázků.
- **Tools** - záložka umožňující screenshot a import/export konfigurace uzlu.
- **Help** - založka s informacemi o verzi GNS3 a užitečnými odkazy.

Tlačítka pod záložkami jsou „Rychlou volbou“ pro některé funkce. Ve spodní části nalezneme konzoly, kde se objevují chybová hlášení společně s upozorněními. V pravém horním rohu se nachází souhrn topologie. Jsou zde vypsány všechny uzly s porty a IP adresami pro telnet připojení. V tomto okně lze ve výchozím nastavení vidět, kde se zařízení (směrovač, přepínač) nachází, zda na serveru nebo na lokálním PC. Zobrazují se i údaje o vytíženosti CPU a RAM paměti.

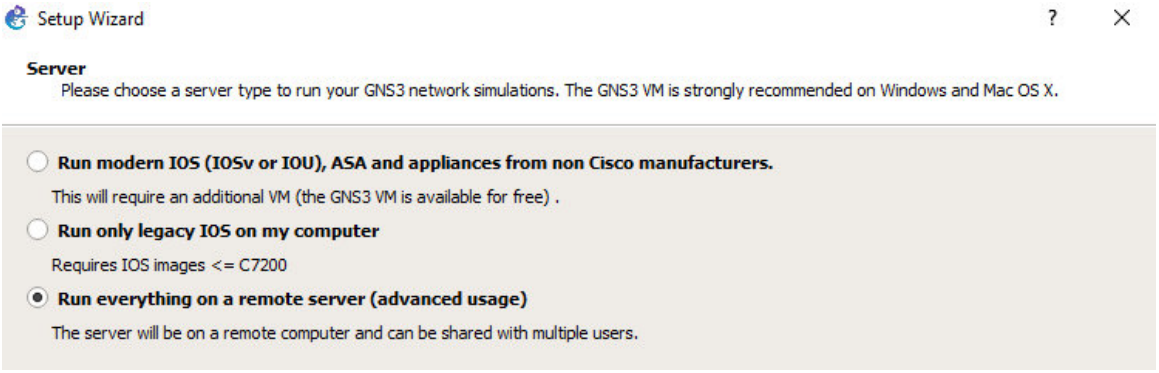
3.4.2 Vlastní instalace

Podobně jako EVE-NG byl GNS3 v2.1.6 nainstalován na server ESXi v6.0. Ze stránek gns3.com bylo nutné stáhnout instalační soubor určený pro ESXi. Soubor byl poté přes vSphere klienta nahrán a nainstalován na server přes možnost „Deploy OVF Template“. Po dokončení instalace a při prvotním spuštění bylo v konzolovém okně napsáno „KVM support available: False“. Toto hlášení značí, že virtualizační software, na kterém běží GNS3 virtuální zařízení nepodporuje vnořenou virtualizaci. Pokud by nedošlo k opravě, snížil by se výkon celého virtuálního zařízení a také funkčnost QEMU emulátoru. Bylo proto nutné vyhledat na disku serveru soubor GNS3.vmx a dopsat do souboru vhv.enable="TRUE". Tím došlo k povolení vnořené virtualizace.

Přes konzolu běžící v programu vSphere byla nastavena statická IP adresa

192.168.1.180 společně s login jménem a heslem. Tímto byla instalace GNS3 VM na serveru dokončena. Dále byl zprovozněn GNS3 GUI.

Pro spuštění GUI je vždy zapotřebí nainstalovat GNS3-all-in-one software na PC, ze kterého se připojujeme na vzdálené GNS3 VM běžící na serveru s ESXi. Instalace proběhla na PC bez problému. Po prvotním spuštění bylo pouze nutné vybrat třetí variantu spuštění na vzdáleném serveru viz Obr. 3.7.



Obr. 3.7: Volba vzdáleného serveru při instalaci GNS3 GUI

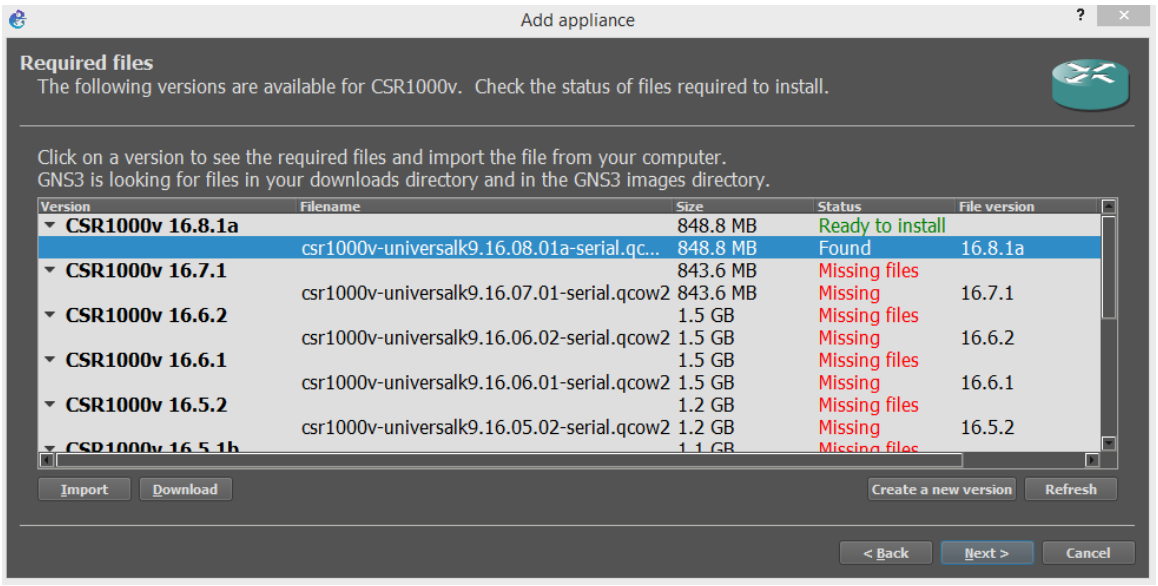
Jako host byla zadána IP adresa GNS3 virtuálního zařízení na ESXi serveru, tedy 192.168.1.180 a port byl zvolen 3080. Veškerá data a laboratoře se budou ukládat na server, a nikoliv na PC s GUI. Díky tomu k nim lze vzdáleně přistupovat z více zařízení.

3.4.3 Nahrávání emulátorů

Nahrávání obrazů směrovačů a dalších zařízení v GNS3 v2.1.6 může probíhat více způsoby. První způsob spočívá v importování přes záložku *Edit –> Preferences*. V levé části nově načteného okna jsou vypsány druhy emulátorů, které GNS3 podporuje. Dle typu se poté přidá obraz pomocí tlačítka *New*. Obraz je vybrán z adresáře na PC, kde je uložen a importován na disk serveru. Nově přidaný obraz je pak dostupný ve výběru na hlavní ploše.

Druhou možností nahrávání obrazů je přes záložku *File –> Import Appliances*. GNS3 na webových stránkách[4] poskytuje šablony pro importování obrazů. Po nahrání libovolné šablony se načte stránka s výběrem verze obrazu, kterou chceme pro danou šablonu nahrát. GNS3 dokáže automaticky najít v adresáři na PC soubor s obrazem. Pokud je obraz uložen v PC, pak se v poli status objeví hlášení „Ready to install - Found“. Pokud obraz není nalezen automaticky, je nutné jej importovat. V případě, že verze obrazu není mezi šablonami, lze si ji manuálně vytvořit tlačítkem *Create a new version*, viz Obr. 3.8.

[4] www.gns3.com/marketplace/appliances



Obr. 3.8: Nahrávání obrazu směrovače pomocí šablony u GNS3

3.5 Cisco IOS

V této práci jsou využity pouze obrazy od firmy Cisco, které využívají Cisco IOS. Jedná se o balíček směrovacích, přepínacích a telekomunikačních funkcí integrovaných do operačního sytému.

3.5.1 CLI – příkazový řádek

Cisco IOS využívá příkazový řádek CLI, založený na textových příkazech namísto grafického prostředí. Důvodem je jednoduší a rychlejší konfigurace. Má tři základní režimy popsané v Tab 3.1: uživatelský, privilegovaný a globální.

Tab. 3.1: Spuštění a ukončení příkazových režimů [15]

Příkazový režim	**Spuštění**	**Prompt**	**Ukončení**
Uživatelský	Log in.	Router>	Příkaz **logout**
Privilegovaný	Z uživatelského režimu pomocí příkazu **enable**	Router#	Pro navrácení do uživatelského režimu příkaz **disable**
Globální	Z privilegovaného režimu pomocí příkazu **configure terminal**	Router(config)#	Pro navrácení do uživatelského režimu příkaz **ctrl+z**. O řežim níže příkazem **exit**.

V uživatelském režimu je jen velice omezený počet příkazů, nelze zde nic měnit ani nastavovat. Pro přístup k příkazům je nutno vstoupit do privilegovaného režimu často vyžadujícího vložení hesla. Zde je již možné vkládat příkazy, kterými se nastavují obecné systémové funkce. Z tohoto režimu lze dále vstoupit do specifického konfiguračního módu (např. pro nastavení portu nebo konkrétního protokolu). Veškeré provedené změny je u Cisco IOS nutno ukládat do paměti Cisco zařízení příkazem **copy running-config startup-config**, čímž se uloží právě běžící konfigurace do startovací konfigurace. Bez provedení tohoto příkazu a případného restartu by došlo ke ztrátě neuložené konfigurace.

Výhodou Cisco IOS je funkce návrhu příkazů. Pomocí symbolu **?** se zobrazí seznam všech příkazů, které jsou dostupné v daném režimu. Přínosné je i dokončování slov použitím tabulátoru.

3.5.2 Verze Cisco IOS

Vzhledem k tomu, že Cisco IOS postupně procházelo vývojem, je na trhu je několik verzí. Jednotlivé verze jsou implementovány v rozdílných typech zařízení dle využití. Příklady verzí jsou vyznačeny v Tab. 3.2.

Tab. 3.2: Příklad Cisco IOS Softwaru ve spojení s hardwarem [16]

Cisco IOS Software	Hardware
12.2S	Cisco 10000
12.4	Cisco 7200
15	Cisco 7600
NX-OS	Cisco Nexus
XE	Cisco ASR 900, 1000
XR	Cisco ASR 9000, Cisco XR 12000

Cisco IOS XE

V této práci se využívá směrovačů s verzí Cisco IOS XE. Liší se od klasického IOS, který je monolitický. Operační systém a všechny procesy sdílí stejnou paměť a CPU. To představuje nevýhodu, neboť se může stát, že jeden proces odstaví celý systém. Stejně tak při upgradování IOS je nutné vyměnit celý soubor a poté je potřeba provést restart, nelze dělat změny po částech.

Cisco IOS XE nepoužívá IOS jako operační systém. Využívá Linux, nad kterým IOS běží jako oddělený proces (deamon). Umožňuje to rozložit zátěž mezi více CPU. Přestane-li jeden proces fungovat, neochromí to celý systém. Cisco IOS XE se skládá

z několika balíčků, nejedná se pouze o jeden velký soubor a je možné ho tak upgradovat po částech. V praxi to znamená, že je možné nainstalovat pouze potřebné balíčky a nezatěžovat paměť směrovače. [17]

Klasický IOS sdílí velkou část kódu s IOS XE a proto CLI příkazy zůstávají stejné.

IOS XE se používá u podnikových přepínačů (Catalyst 9xxx, 3xxx), agregačních/okrajových směrovačů (ASR 900, 1000) nebo virtuálních směrovačů (CSR 1000v). Díky podpoře virtualizace je možné CSR 1000v snadno spustit v emulačním prostředí a byl proto použit v této práci, konkrétně verze *csr1000v-universalk9.03.17.00.S.156-1.S-ext.qcow2*.

BDI (Bridge Domain Interface)

Cisco IOS XE podporuje funkci BDI. BDI je logické rozhraní, který umožňuje obousměrný tok mezi linkovou vrstvou L2 a síťovou vrstvou L3. Roli zde hraje i *bridge domain*, neboli doména mostu. Ta má stejný index jako BDI. Každá doména mostu reprezentuje L2 všesměrovou doménu.

Bridge Domain Interface je omezen počtem 4096 na systém. Podporuje QoS značkování a přidělování „policy map“, IPv6 unicast směrování, dynamické směrování jako OSPF, RIP, BGP atd. [34]

Lze ho přirovnat k rozhraní SVI (Switch Virtual Interface), které je uživatelsky rozšířenější.

3.6 Ostinato

Ostinato je open-source síťový generátor a analyzátor s uživatelsky vstřícným GUI. Je díky němu možné vytvořit a odeslat různými rychlostmi pakety s různými protokoly. Lze ho považovat za obrácený Wireshark.

Ostinato podporuje protokoly jako jsou Ethernet/802.3/LLC SNAP, VLAN, ARP, IPv4, IPv6, IP tunelování a další. Funguje na OS jako jsou Windows, Linux, BSD nebo Mac OS X.

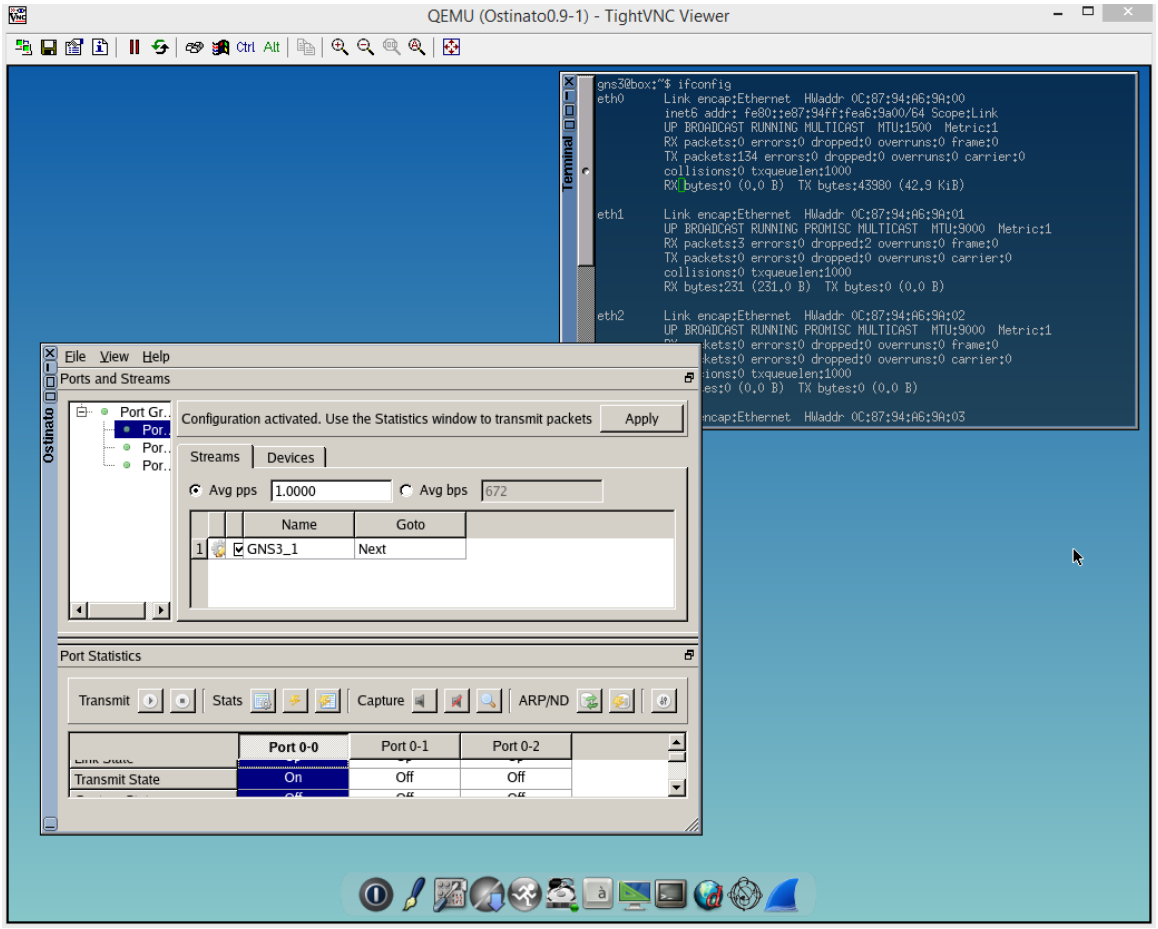
3.6.1 Instalace do GNS3

Ostinato bylo nainstalováno do GNS3 nahráním předchystané šablony ze stránek gns3.com. Pomocí tlačítka *File–>Import appliance* byla zvolena šablona z disku. Poté byla vybrána verze Ostinato 0.9, která byla opět uložená na disku. Instalace dále proběhla výchozím nastavením.

Porty od *eth1* výše se používají na data. Port *eth0* se doporučuje nevyužívat, protože je určený pro správu. Do topologie byly vloženy dva obrazy. První Ostinato

bylo připojeno portem *eth1* na port *g1* u SIAD směrovače. Druhé Ostinato bylo připojeno opět portem *eth1* na port *g11* na SIAD směrovači.

Výhodou u GNS3 je to, že je GUI importované přímo uvnitř VM a není tak potřeba připojovat další zařízení pro grafické prostředí, viz Obr. 3.9.



Obr. 3.9: Ukázka grafického prostředí Ostinato v GNS3

3.6.2 Instalace do EVE-NG

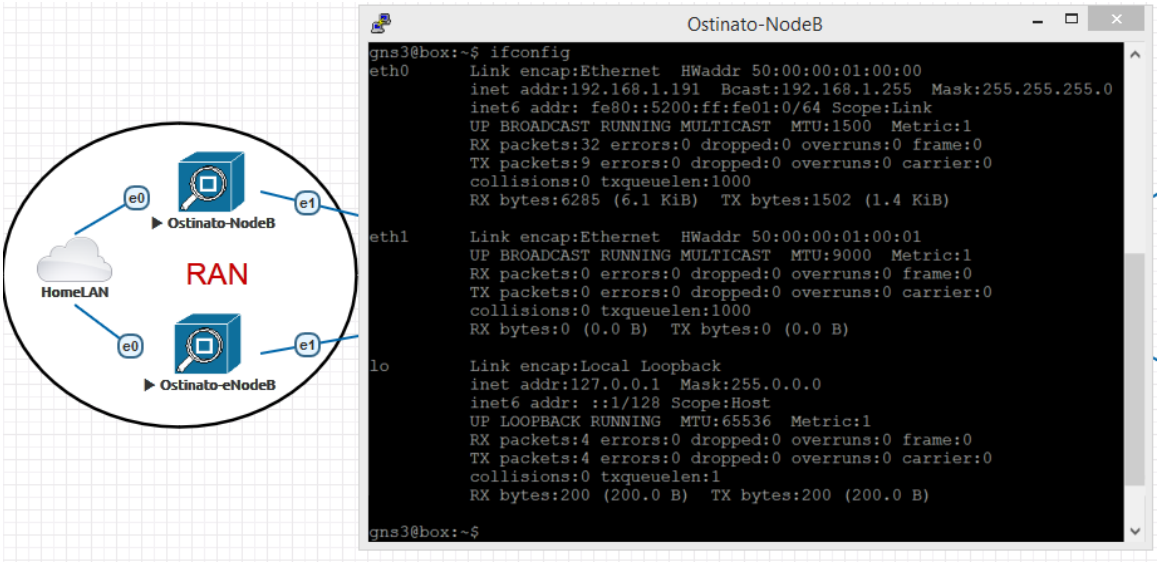
V případě EVE-NG je instalace Ostinato 0.9 komplikovanější. Ostinato nelze spustit jako celek, jak tomu bylo u GNS3. U EVE-NG je od sebe odděleno uživatelské prostředí a samotný obraz generátoru Ostinato. Na PC se musí tedy nainstalovat GUI programu Ostinato a v EVE-NG je nutné vytvořit obraz Ostinato, ke kterému lze přistupovat pouze přes konzolu.

Do konzoly EVE-NG v programu vSphere je nutné zadat příkazy `apt-get update` a následně `apt-get install eve-ng-addons-ostinato-drone`. Instalace tímto způsobem je možná díky dostupnosti obrazu Ostinato v repozitáři volně dostupných doplňků. Přidání uzlu se provede tlačítkem *Add an object –> Node* a v seznamu

se vybere šablona Ostinato. Důležité je přiřadit obrazu minimálně dva porty, jeden pro správu a jeden pro data.

Následně se musí vytvořit spojení s vnitřní sítí PC, kde je již nainstalovaná klientská část programu Ostinato, neboli GUI prostředí. Záložkou *Add an object –> Network* se vytvoří v EVE-NG síť např.pojmenovaná HomeLAN, s typem připojení *Management (Cloud0)*. Síť je připojená k obrazu Ostinato portem *eth0*.

V konzolovém okně programu Ostinato spuštěnému v EVE-NG je možné ověřit přidělení IP adresy DHCP serverem vnitřní sítě, viz Obr. 3.10. V tomto případě byla portu *eth0* u NodeB přidělena adresa 192.168.191, u eNodeB adresa 192.168.1.189. Na tyto adresy je nutné se připojit GUI na PC s portem 7878.



Obr. 3.10: Zapojení Ostinato v EVE-NG s přidělením IP adresy

3.7 Wireshark

Wireshark je open-source protokolový analyzátor a paketový *sniffer*.

Slouží k vyhledávání problémů, vytváření softwaru a komunikačních protokolů, ale také k výuce. Je podporován OS jako jsou Windows, Linux, BSD, Mac OS atd.

Wireshark využívá promiskuitního módu. Síťová rozhraní, která Wireshark podporují, jsou jím na něj přímo nastavena. Je tak možné sledovat provoz na těchto rozhraních, včetně broadcast i multicast komunikace, ne pouze provoz určený jedné adrese na portu. [18]

Wireshark umí rozložit strukturu zapouzdřeného paketu a rozlišit příslušnost informací k jednotlivým protokolům. K zachytávání paketů používá knihovnu *pcap*,

čímž je omezen jen na pakety podporované touto knihovnou. Data je možné analyzovat přímo z živé sítě nebo z již zachycených paketů. Lze je procházet pomocí GUI nebo v příkazovém řádku přes aplikaci TShark. Zachycené pakety lze i programově upravovat.

Výhodou Wireshark je jeho automatická instalace společně s instalací emulačních programů. U GNS3 je jeho instalace nabídnuta při instalování softwaru GNS3-all-in-one na uživatelský PC. V případě EVE-NG je jeho instalace nabídnuta při instalaci „Windows Client integrated pack“. Wireshark je poté spuštěn přes emulační programy. Díky tomu je možno zachytit pakety na přímo zvoleném portu a není nutné Wireshark nastavovat.

3.8 PowerSNMP manager

„PowerSNMP manager“ je volně přístupný program umožňující sledování zařízení v síti pomocí SNMP (Simple Network Managment Protocol) požadavků. Jedná se o plnohodnotnou aplikaci od společnosti Dart. Na trhu existuje velké množství podobných programů, pro tuto práci byl však zvolen tento, a to z důvodu jeho dostupnosti.

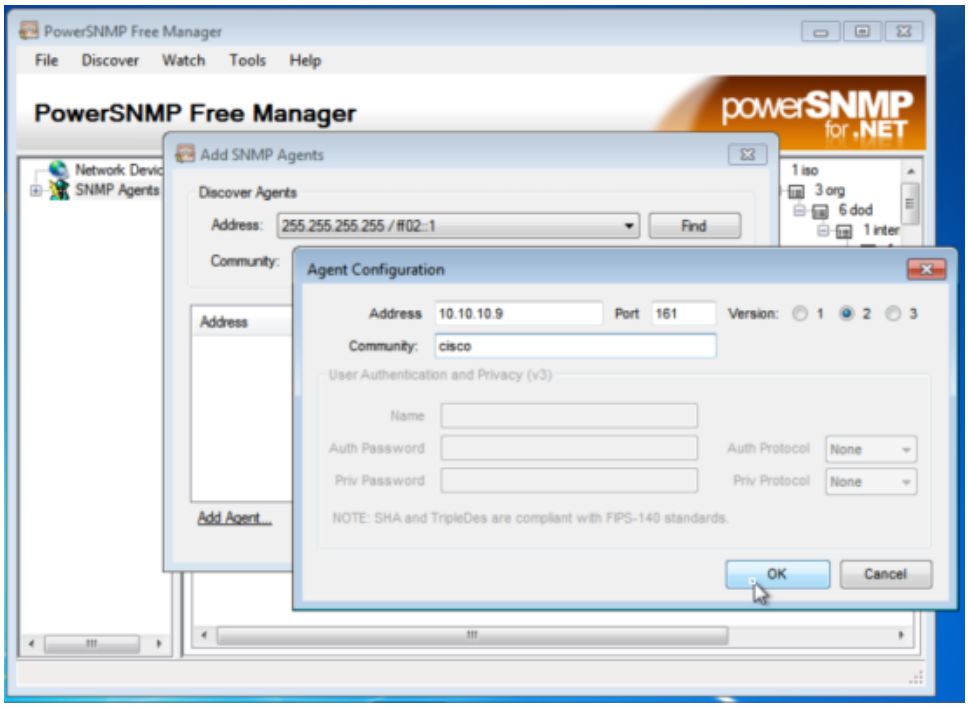
PowerSNMP zajišťuje veškeré potřeby pro dohled nad sítí. Dotazuje se a sleduje hodnoty od SNMP agentů, dohlíží nad zprávami „Trap“, zjišťuje funkčnost hostů (ping) a konfiguruje hlášení s možností zasílání emailových upozornění.

Existují dvě možnosti jak importovat SNMP manažera do emulačních programů. První je emulace PC klienta s Windows a následné nainstalování PowerSNMP. Druhou možností je propojení emulované sítě se sítí, ve které se nachází uživatelský PC přes tzv. cloud a následná instalace PowerSNMP. V této práci bylo využito první varianty, aby nebylo nutné pro každého nového uživatele instalovat a nastavovat na vlastním PC PowerSNMP.

3.8.1 Nastavení SNMP manažera

SNMP manažerovi byla přidělena statická IP adresa 10.10.10.10/24. Tato adresa byla nastavena na rozhraní *eth1* na PC připojeném ke směrovači. Jako výchozí brána byla nastavena IP adresa 10.10.10.9 na portu *g1* u MSN_A.

Následně je nutné v programu PowerSNMP přiřadit manažerovi agenty. Pomocí záložky na horní liště *Discover–>SNMP Agents...* se otevře okno, kde se v levém dolním rohu nachází *Add Agent...*. Dle zvolené verze se doplní informace včetně IP adresy agenta a stiskem tlačítka *Ok* je přidán, viz Obr. 3.11.



Obr. 3.11: Přidání agenta do programu PowerSNMP

4 Srovnání EVE-NG a GNS3

V následující kapitole se budeme věnovat srovnání obou emulačních programů. Z předchozí kapitoly 3 vyplývá, že se oba liší instalačním procesem. Velký rozdíl je také patrný při nahrávání obrazů emulátorů. Zatímco u GNS3 je postup nahrávání implementován přímo v GUI a je usnadněn i předem vytvořenými šablonami, u EVE-NG je uživatel nucen připojit se na server, nahrát obraz do přesně určeného adresáře a přidělit mu předem stanovený název.

EVE-NG má naopak výhodu ovládání přes webové prostředí, zatímco GNS3 vyžaduje pro spuštění GUI instalaci celého programu na PC.

Uvnitř samotných programů jsme dále zjistili, že u EVE-NG je v případě nutnosti přepojení kabelu do jiného portu nutno zařízení vypnout, zatímco GNS3 umožňuje změnu za provozu. Dostupná dokumentace je rozšířenější u GNS3 než u EVE-NG, což svědčí o jeho velké popularitě. Rozdíl je také patrný v připojení obrazu Ostinato, viz Kap.3.6.

Výchozí topologie, která je popsána níže v této kapitole, byla simulována v obou emulačních programech. Byly použity totožné obrazy Cisco CSR 1000v (*csr1000v-universalk9.03.17.00.S.156-1.S-ext.qcow2*) a ty samé verze programů Ostinato a PowerSNMP manager. Do směrovačů byla vložena identická výchozí konfigurace. Na serveru ESXi jim byly přiřazeny odpovídající zdroje.

Porovnány byly 3 parametry: zatížení CPU, doba načtení všech obrazů a zatížení paměti RAM.

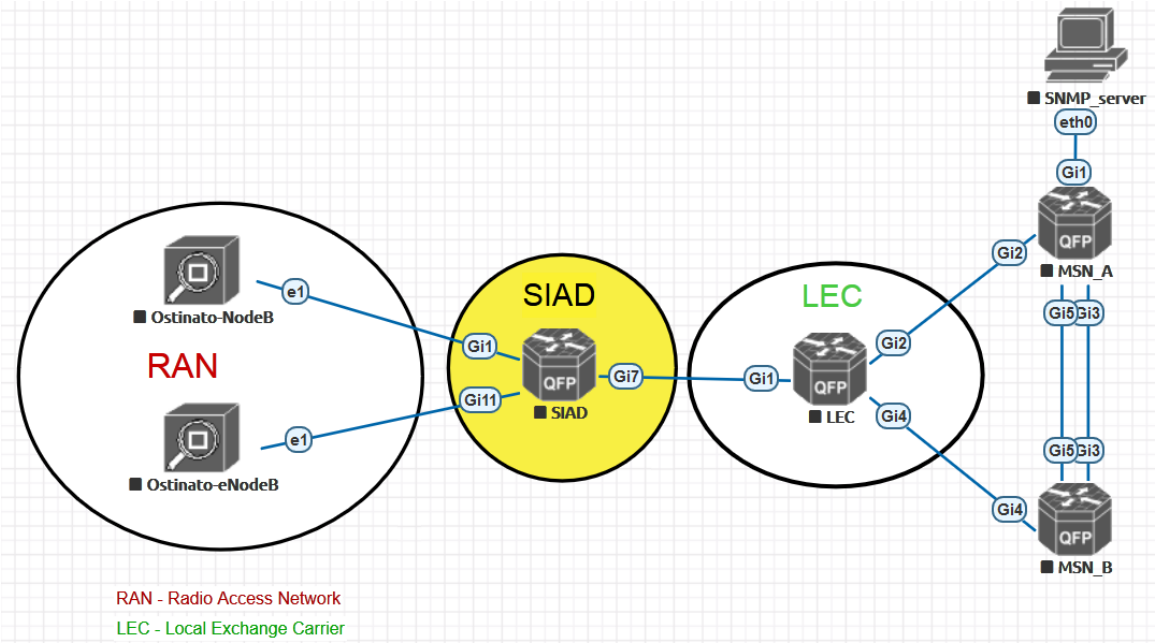
4.1 Testovaná výchozí transportní síť

Tato podkapitola nabízí přehled transportní sítě, která bude základem pro oba simulační scénáře v kapitole 5 a zároveň bude sloužit jako výchozí konfigurace při testování zátěže ESXi serveru. Veškerý popis je soustředěn na Obr. 4.1, kde je znázorněna celá výchozí topologie.

4.1.1 RAN část

RAN část je tvořena dvojicí QEMU emulátorů síťového generátoru Ostinato 0.9, pro simulaci NodeB a eNodeB. Na Obr. 4.2 lze vidět přidělený rozsah a IPv4 adresy pro NodeB a IPv6 pro eNodeB.

NodeB obsahuje VLAN 101 a 102. *Vl101* je nazývána „Bearer“ a je VLAN nosnou, přes kterou prochází veškerá data. *Vl102* je tzv. „OAM“ neboli „Operations,



Obr. 4.1: Topologie výchozí transportní sítě

administration and management“ a slouží k instalování, monitorování nebo k řešení problémů se zařízením. eNodeB má naopak VLAN 211 a 212. *Vl212* je Bearer a *Vl211* je OAM.

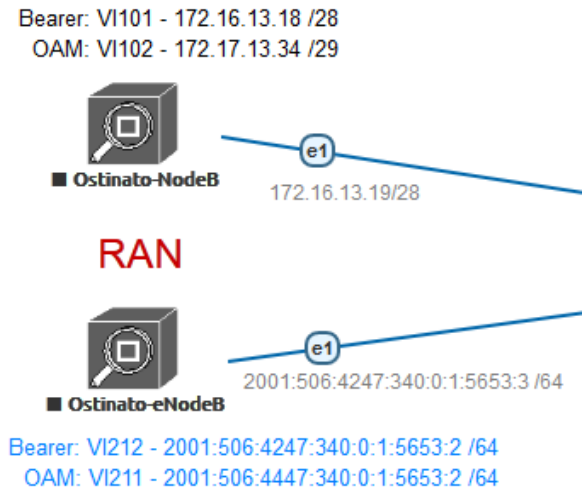
NodeB a eNodeB jsou přímo spojené portem *e1* se SIAD směrovačem.

4.1.2 SIAD směrovač

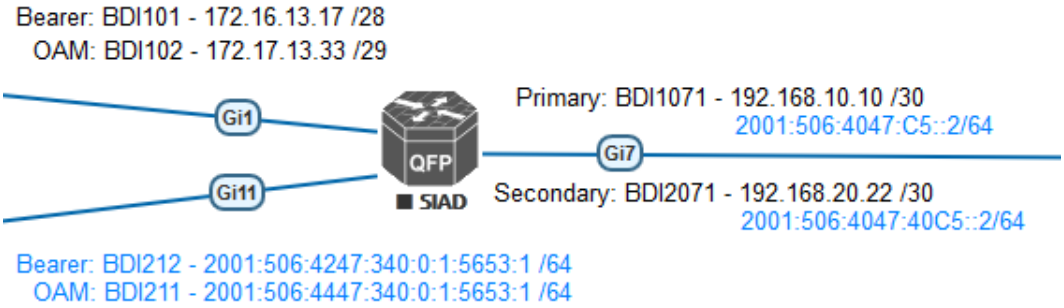
Pro emulaci SIAD směrovače byl použit QEMU emulátor s obrazem směrovače Cisco CSR 1000v (*csr1000v-universalk9.03.17.00.S.156-1.S-ext.qcow2*). Přiřazení IP adres je zobrazeno na Obr. 4.3. Na SIAD směrovači je využíváno BDI. Dochází zde k přemostění L2 toku dat na síťovou vrstvu. Dále je zde využito OSPFv2 ve spolupráci s protokolem BFD a statického směrování jak u IPv4, tak IPv6. SIAD směrovač je agentem SNMP protokolu.

Tyto porty jsou na směrovači klíčové:

- *g1* - *Shorthaul* vedoucí k NodeB a obsahující *BDI101* a *BDI102*
- *g11* - *Shorthaul* vedoucí k eNodeB a obsahující *BDI211* a *BDI212*
- *g7* - *Backhaul* port směřující k MSN směrovačům. Spadá pod něj primární *BDI1071* a sekundární *BDI2071*.



Obr. 4.2: Nastavení IP adresování u RAN části sítě



Obr. 4.3: Nastavení IP adresování u SIAD směrovače

4.1.3 LEC část

Simulace této části je pouze okrajová a není zahrnuta podrobněji v simulačních scénářích, neboť se netýká podstaty této práce.

Abychom předešli komplikacím, zvolili jsme pro emulaci opět směrovač Cisco CSR 1000v. Slouží však jen jako prostředník na L2 vrstvě. Z pohledu SIAD a MSN směrovačů není viditelný. Poté je nutné nastavit přemostění na správné porty, a to opět pomocí BDI.

4.1.4 MSN směrovače

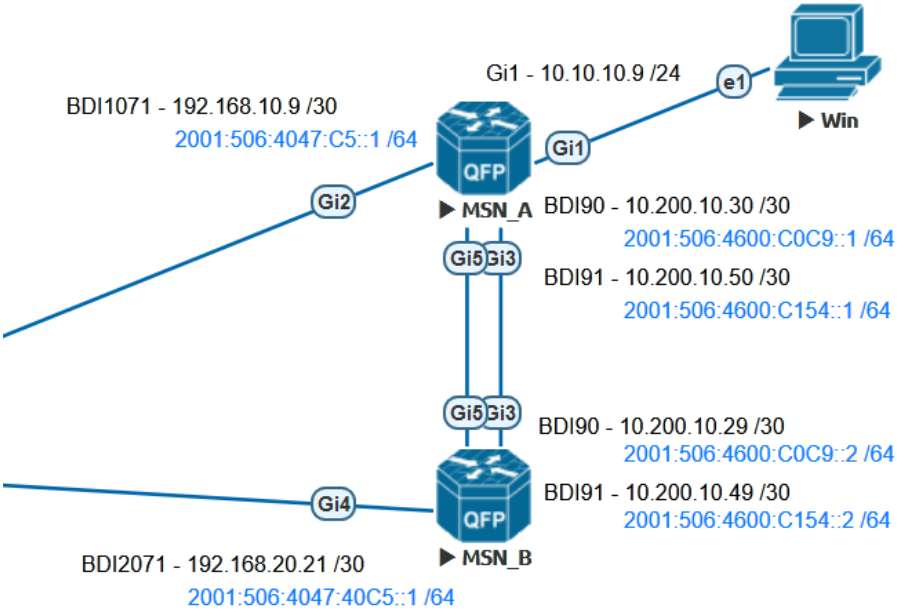
Data přichází od SIAD směrovače na MSN směrovače přes *Backhaul* linku. U MSN_A na port *g2*, který je *bridge* doménou spojen s *BDI1071*. U MSN_B na port *g4*, který je spojen s *BDI2071* a slouží jako záloha v případě výpadku. IPv4 je směrováno pomocí protokolu OSPFv2 a statických cest. IPv6 využívá pouze statické cesty mezi SIAD a MSN směrovači.

Pro větší redundanci je MSN pár spojen 2 linkami mezi porty *g3-g3* a *g5-g5*. Na směrovačích je ve výchozí topologii využito pouze protokolů OSPFv2 a OSPFv3, nikoliv BGP.

OSPFv2 i OSPFv3 jsou použity mezi MSN párem pro přenos informací a jako záloha v případě výpadku jednoho z nich. Dvojice fyzických linek je sloučena protokolem LACP (Link Aggregation Control Protocol) do jedné logické linky s využitím funkce *port-channel*.

MSN_A je dále připojeno portem *g1* k SNMP manažerovi, který dohlíží nad sítí.

Na Obr. 4.4 lze vidět IP adresy u MSN_A a MSN_B pro výchozí topologii.



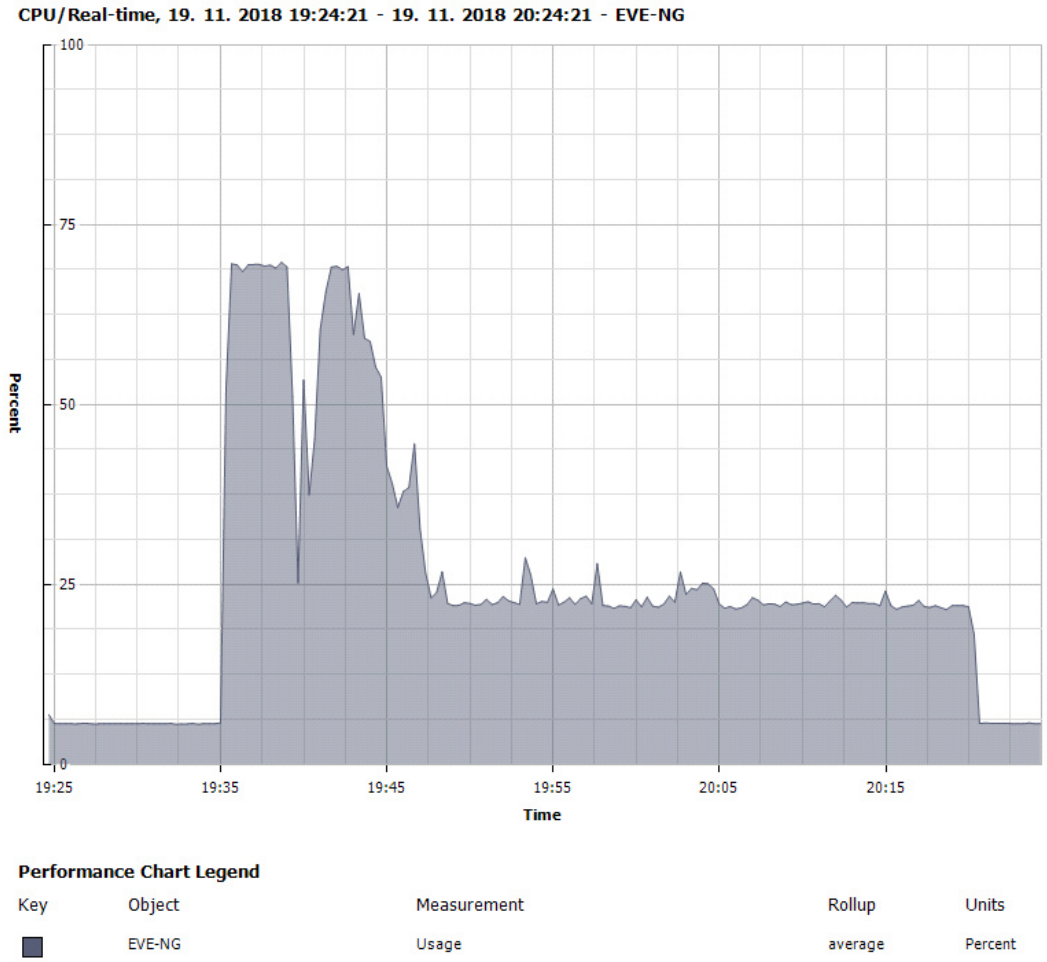
Obr. 4.4: Nastavení IP adresování u MSN směrovačů

4.2 Zatížení CPU

Prvním sledovaným parametrem při měření bylo zatížení CPU. Sledovali jsme jej od počátku hromadného spuštění do momentu ustálení, tedy načtení obrazů. Sledováno bylo procentuální zatížení. Naměřené údaje se vztahují především k obrazům směrovačů a to z důvodu velké hardwarové náročnosti.

Stav zatížení CPU EVE-NG lze pozorovat na Obr. 4.5. Dosažené hodnoty GNS3 jsou na Obr. 4.6.

V čase 19:35 došlo k hromadnému startu obrazů v EVE-NG, viz Obr. 4.5. Následně došlo k počítání SHA-1 heše pro ověření správnosti obrazů. Zátěž CPU byla okolo 65% po dobu 5 min. Poté došlo k poklesu téměř na hranici 20% a okamžitému



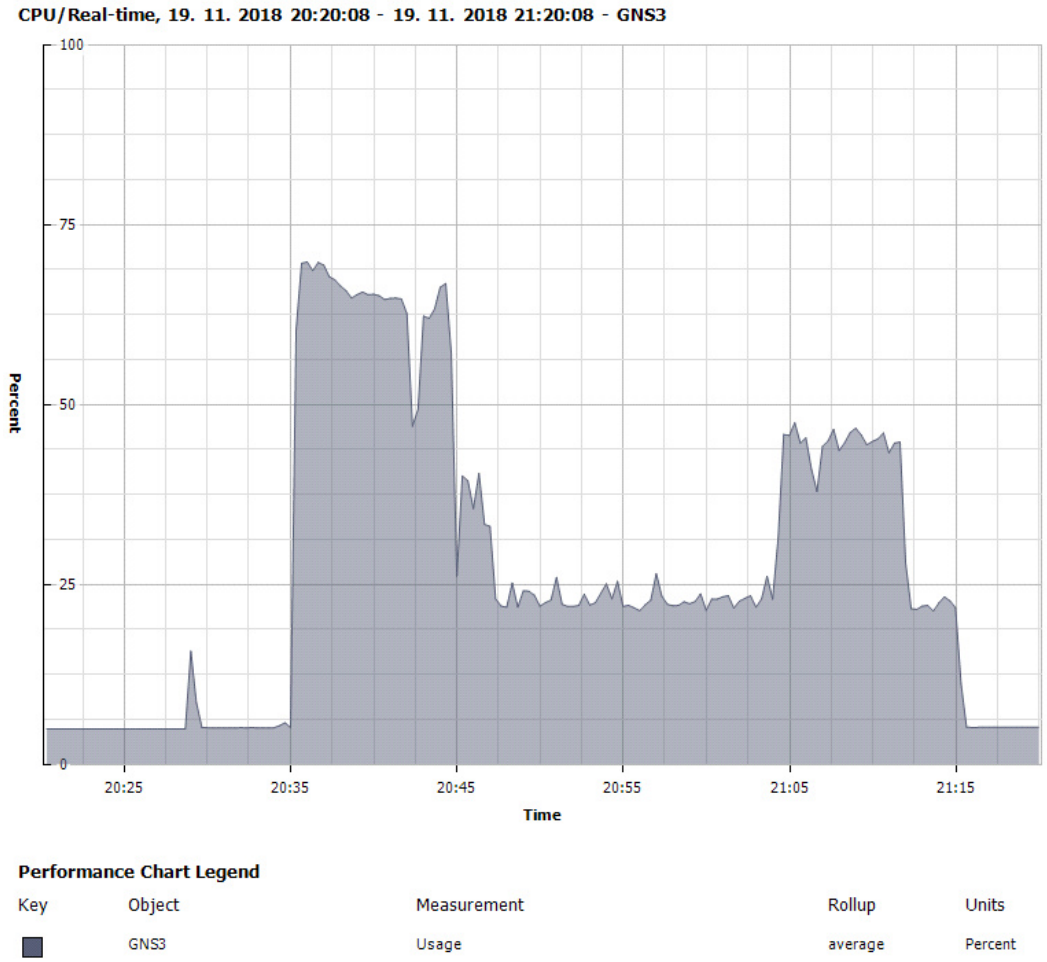
Obr. 4.5: Zatížení CPU při spuštění laboratoře viz Kap.4.1 v EVE-NG - procentuálně

nárůstu zpět na hranici zatížení CPU na 65%, kdy došlo k načítání jednotlivých funkcí IOS XE. Doba trvání byla opět okolo 5 min. Po načtení jednotlivých funkcí došlo v čase 19:45 k poklesu na přibližně 36%. Poté došlo k nahrávání start-up konfigurace na směrovačích po dobu 2,5 min. Maximální zátěž během tohoto intervalu byla 40%.

K celkovému ustálení došlo v čase 19:47. Během ustálení bylo zatížení CPU okolo 23%. Vypnutí celé laboratoře lze pozorovat v čase 20:20.

V případě GNS3 došlo k hromadnému startu v čase 20:35 viz Obr. 4.6. Počítání SHA-1 heše trvalo oproti EVE-NG o něco déle - 7 min. Procentuální zatížení na CPU bylo na 65%, během intervalu 7 min, ale postupně klesalo na 60%. Pokles na 42% nastal v čase 20:42. V ten stejný čas došlo k růstu zpět na 62%, způsobenému načítáním funkcí IOS XE.

K načítání start-up konfigurace došlo ve 20:45, kdy bylo bootování IOS XE ukončeno. Maximální zátěž během načítání, které trvalo také 2,5 min jako v případě



Obr. 4.6: Zatížení CPU při spuštění laboratoře viz Kap.4.1 v GNS3 - procentuálně

EVE-NG, bylo 35%. V čase 20:47 došlo k ustálení na 20%, což je méně než u EVE-NG.

Při měření v GNS3 došlo v čase 21:04 k navýšení zátěže CPU na hodnotu okolo 40% za dobu téměř 8 min. Nejpravděpodobněji došlo k výměně dat mezi klientskou a serverovou částí GNS3. V čase 21:15 byla laboratoř vypnuta.

Z toho vyplývá, že EVE-NG a GNS3 se z pohledu zátěže na CPU příliš neliší.

4.3 Čas potřebný pro načtení obrazů

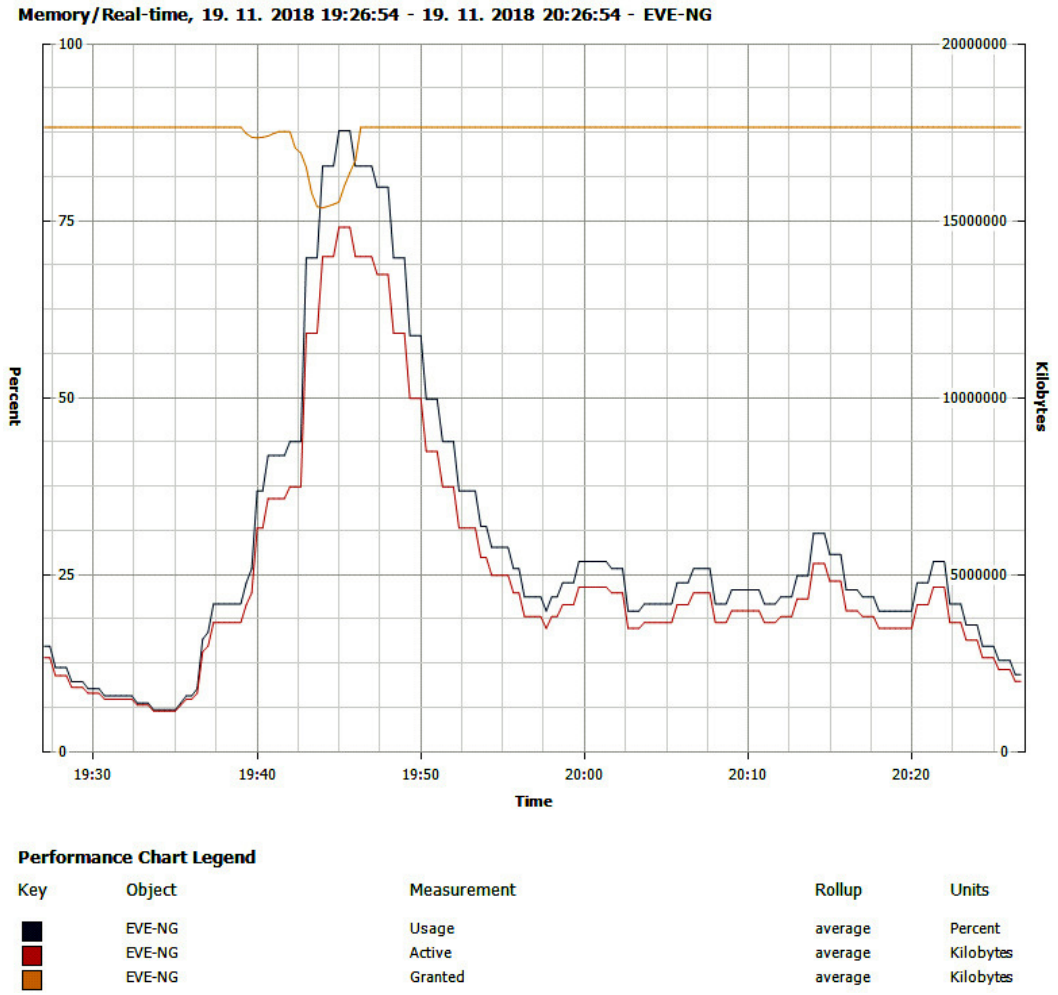
Dalším parametrem, který jsme zkoumali a který vychází z výsledků uvedených v Obr. 4.5 a 4.6, je doba potřebná pro načtení obrazů směrovačů s konfigurací popsanou v Kap.4.1. Abychom dosáhli přesnější naměřené hodnoty, byly spuštěny při hromadném startu stopky. Vypnuty byly v momentě načtení konfigurací na zařízeních, dle údajů v konzoli. Naměřené časy jsou uvedeny v Tab. 4.1. Jak je patrné, jsou časy téměř totožné.

Tab. 4.1: Naměřený čas potřebný pro načtení obrazů

	EVE-NG	GNS3
Doba nahrání obrazů	12 min 20 s	12 min 30 s

4.4 Zatížení paměti RAM

Třetím parametrem bylo zatížení paměti RAM. Oběma emulačním programům byla přidělena maximální možná paměť pro daný server - 16 GB. Opět došlo k měření v obou emulačních programech. Obr. 4.7 znázorňuje výsledky měření v EVE-NG a Obr. 4.8 v GNS3. Zde jsou již rozdíly mezi emulačními programy markantnější.



Obr. 4.7: Zatížení RAM paměti při spuštění laboratoře v EVE-NG viz Kap.4.1

Na Obr. 4.7 si lze povšimnout prudkého nárůstu spotřeby paměti v čase 19:35 až do času 19:45, a to vlivem počítání SHA-1 heše a bootování IOS XE. V 19:45 došlo k maximální spotřebě paměti okolo 15 GB, procentuálně je to 87,5%. V čase

19:48 došlo k prudšímu klesání zátěže na paměť RAM z důvodu ustálení a ukončení nahrávání start-up konfigurace. Při ustáleném stavu se paměť RAM v EVE-NG pohybovala okolo 25%, což odpovídá 4 GB.

Při testování emulačního programu GNS3 vidíme na Obr. 4.8 navíc i stav při spuštění samotného VM GNS3 ještě před hromadným startem. K hromadnému startu došlo v čase 20:35. I zde došlo k prudkému nárůstu zátěže ze stejných důvodů jako v EVE-NG. V čase 20:47 při tomto měření dosáhla paměť RAM maxima 13,75 GB, tedy 81,25%. Během této doby došlo k nahrávaní start-up konfigurace. Poté následovalo ustálení.

Performance Chart Legend

Key	Object	Measurement	Rollup	Units
	GNS3	Usage	average	Percent
	GNS3	Active	average	Kilobytes
	GNS3	Granted	average	Kilobytes

Obr. 4.8: Zatížení RAM paměti při spuštění laboratoře viz Kap.4.1 u GNS3

Při něm byla dle výsledků na Obr. 4.8 spotřeba RAM paměti okolo 6 GB, tedy 37,5%. A to i ještě před výkyvem, který nastal v čase 21:04.

Rozdíl mezi EVE-NG a GNS3 nastal při inicializaci a při spotřebě paměti RAM v ustáleném stavu. EVE-NG potřeboval pro úvodní načtení okolo 15 GB RAM,

naopak GNS3 potřeboval pouze 13,75 GB RAM. Nicméně, EVE-NG u topologie viz Kap.4.1 spotřebovává o 2 GB paměti RAM než GNS3 v ustáleném stavu.

Nabízí se několik možností pro vysvětlení těchto rozdílů. EVE-NG může odlišným způsobem spouštět vybrané emulátory, proto je náročnější při úvodním spuštění. GNS3 naopak může být hůře optimalizovaný pro zvolenou výchozí topologii. Je dále možné, že jinak zpracovává vybrané QEMU obrazy nebo odlišně zachází s jejich konfigurací. Je zde i možnost, že celý emulační program GNS3 je při provozu náročnější na pamět RAM serveru, na kterém je spuštěn.

5 Simulační scénáře

V předchozí kapitole jsme porovnali 2 emulační programy a nyní se zaměříme na konkrétní scénáře emulace transportní sítě. Transportní síť je část mobilní přístupové sítě (RAN), která přenáší data od uživatele do páteřní sítě a naopak. Není závislá na technologii použité v rádiové části a pracuje na protokolu IP.

Předmětem této kapitoly jsou dva komplexní scénáře, včetně detailního popisu a řešení, které jsme vymysleli pro emulaci transportní sítě.

Scénář 1 má název „Konfigurace transportní sítě“ a zaměřuje se na konfiguraci celé transportní sítě, včetně simulace základnových stanic NodeB a eNodeB z pohledu QoS. Scénář tvoří devět kroků, které jsou detailně popsány a význam jednotlivých kroků konfigurace je vysvětlen. U každého kroku je zdůrazněno, jaké změny způsobil. Díky sérii kontrolních otázek a samostatných úkolů, které jsme do scénáře zakomponovali, lze konfiguraci ověřit a lépe pochopit.

Scénář 2 vychází ze Scénáře 1, je však u něj zvolen rozdílný postup. Scénář 2 má název „Časté chyby v transportní síti“ a je založený na hledání chyb v předem vytvořených konfiguracích směrovačů. Jsou vytvořeny dva „TroubleTickets“, kde jsou nahlášeny nefunkční části a co je potřeba opravit. Úkolem uživatele je, aby sám přišel na chybu a opravil ji. Scénář 2 také obsahuje kontrolní otázky pro lepší pochopení problematiky. Na konci jsou pro každý „TroubleTicket“ uvedena vzorová řešení.

Oba scénáře využívají pro směrovače obraz Cisco CSR 1000v, konkrétně *csr1000v-universalk9.03.17.00.S.156-1.S-ext.qcow2*.

5.1 Scénář 1 - Konfigurace transportní sítě

Scénář 1 popisuje konfiguraci celé transportní sítě i simulaci základnových stanic NodeB a eNodeB především z hlediska QoS. Scénář 1 tvoří sedm základních kroků a dva dodatečné kroky, pro pokračování na Scénář 2 stačí pouze provedení sedmi základních kroků. Všechny kroky konfigurace jsou detailně popsány a význam jednotlivých kroků je vysvětlen.

Na některých místech se nachází symbol ✎, který signalizuje příkazy, kterými lze ověřit správnost konfigurace, případně objasnění některých otázek spojených s nastavením.

Ve scénáři jsou také „Samostatné úkoly“, které ověřují konfiguraci a zároveň mají za cíl vysvětlit význam provedeného nastavení.

Po sedmi základních krocích následuje podkapitola s kontrolními otázkami ke Scénáři 1. Zodpovězení těchto otázek napomůže čtenářům pochopit problematiku simulované transportní sítě. Odpovědi na tyto otázky se nachází v příloze C.1.

Scénář 1 vychází z celkové topologie vyobrazené na Obr. A.1 viz příloha. Scénář předpokládá, že řešitel je schopný zařízení pojmenovat a provést základní nastavení.

Konfigurací směrovače LEC nebude nutné se podrobněji zabývat, neboť jeho nastavení není podstatou simulace, jak již bylo uvedeno v Kap.1.2. Z pohledu SIAD směrovače a MSN páru není LEC viditelný. Je potřeba na něj však nahrát základní konfiguraci viz příloha B.2. Směrovač PE bude naopak konfigurován, pouze však z pohledu směrovacího protokolu BGP.

Správné řešení celé konfigurace je obsaženo v příloze B.

5.1.1 Krok 1. - nastavení IP adres na rozhraních

V mobilní transportní síti je využito protokolů IPv4 a IPv6 viz Kap.2. IP adresy byly přiděleny dle Tab. A.1 viz příloha. Krok 1. se zaměřuje na nastavení na SIAD, MSN_A, MSN_B a PE směrovačů, viz Obr. A.1. Tuto konfiguraci potřebujeme provést, protože zajišťuje celkové rozdělení adresního prostoru v simulované transportní síti. Nastavení IP adres u základnových stanic NodeB a eNodeB popisuje Krok 2.

1. **SIAD:**

   Jak již bylo zmíněno v úvodu této kapitoly, SIAD směrovač využívá obraz Cisco CSR 1000v, kde lze pracovat s L2 a L3 vrstvou pomocí rozhraní BDI.

   (a) **Rozhraní vedoucí k NodeB a eNodeB:**

   - Z pohledu *Shorthaul* se jedná o připojení k NodeB portem *Gi1* a eNodeB portem *Gi11*. Fyzicky jsou základnové stanice připojeny každá jedním kabelem. Daný provoz se dělí pomocí L2 vrstvy na správu a na samotná data. K provozu pro správu slouží směrem k NodeB *BDI102* a je nazýván OAM, veškerá data zákazníků prochází *BDI101*, který se nazývá Bearer. Opačná logika číslování je pak použita směrem k eNodeB. OAM je *BDI211*, Bearer *BDI212*. 3G architektura (NodeB) využívá pouze IPv4, 4G architektura (eNodeB) pracuje na IPv6.
   - Konfigurace BDI je specifická, samotné BDI rozhraní se musí nejprve vytvořit prostřednictvím `interface BDI<číslo>` a poté přiřadit k fyzickému rozhraní přes `service instance <číslo>, encapsulation dot1q <číslo>` a `bridge-domain <číslo>`.

```
interface BDI101
 description Link to NodeB Bearer
 ip address 172.16.13.17 255.255.255.240

interface BDI102
 description Link to NodeB OAM
 ip address 172.17.13.33 255.255.255.248

interface Gi1
 service instance 101 ethernet
  encapsulation dot1q 101
  bridge-domain 101
 service instance 102 ethernet
  encapsulation dot1q 102
  bridge-domain 102

interface BDI211
 description Link to eNodeB OAM
 ipv6 address 2001:506:4447:340:0:1:5653:1/64

interface BDI212
 description Link to eNodeB Bearer
 ipv6 address 2001:506:4247:340:0:1:5653:1/64

interface Gi11
 service instance 211 ethernet
  encapsulation dot1q 211
  bridge-domain 211
 service instance 212 ethernet
  encapsulation dot1q 212
  bridge-domain 212
```

Veškerá část základního nastavení portů *Shorthaul* linky je díky předchozí konfiguraci vytvořena a nyní se můžeme zaměřit na *Backhaul*.

(b) **Rozhraní vedoucí k MSN páru:**

Backhaul je tvořen kabelem připojeným na fyzický port *Gi7* u SIAD směrovače. Kabel vede přes síť LEC, kde se rozdvojí. Jedno připojení vede do MSN_A a druhé do MSN_B. Aby bylo možné rozlišit, kam budou data posílána, je opět využito BDI. Na SIAD směrovači je nutné vytvořit *BDI1071*, který představuje primární linku směřující k MSN_A a *BDI2071*, který vede k MSN_B a slouží jako sekundární linka pro případ výpadku. Oba BDI jsou přiřazeny k fyzickému portu *Gi7*.

```
interface BDI1071
 description - Primary Vlan to MSN_A
 ip address 192.168.10.10 255.255.255.252
 ipv6 address 2001:506:4047:C5::2/64
 encapsulation dot1Q 1071  //without it OSPF neighborship won't come UP
```

```
interface BDI2071
 description - Secondary Vlan to MSN_B
 ip address 192.168.20.22 255.255.255.252
 ipv6 address 2001:506:4047:40C5::2/64
 encapsulation dot1Q 2071  //without it OSPF neighborship won't come UP

interface GigabitEthernet7
 description Backhaul Link to MSN_A,MSN_B
 service instance 1071 ethernet
  encapsulation dot1q 1071
  bridge-domain 1071
 service instance 2071 ethernet
  encapsulation dot1q 2071
  bridge-domain 2071
```

(c) **Nastavení Loopbacků:**

Posledním úkolem Kroku 1. je vytvoření *Loopback0*, který slouží k identifikaci směrovače a k jejímu následnému použití ve scénáři.

```
interface Loopback0
 description OSPF RID
 ip address 192.168.0.99 255.255.255.255
```

Loopbacky budou později hrát ve scénáři důležitou roli, zejména u dynamických směrovacích protokolů. Díku nim nebudou tyto protokoly vázány na konkrétní fyzický port na směrovači, ale na logické rozhraní, tedy Loopback, které je - pokud tomu administrátor nechce jinak - vždy ve stavu UP.

✎ Nastavení a stav portů ověříme následujícím příkazem, jehož výstup je taktéž ukázán níže:

show ip interface brief

```
SIAD#sh ip int br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       unassigned      YES unset  up                    up
GigabitEthernet2       unassigned      YES unset  administratively down down
GigabitEthernet3       unassigned      YES unset  administratively down down
GigabitEthernet4       unassigned      YES unset  administratively down down
GigabitEthernet5       unassigned      YES unset  administratively down down
GigabitEthernet6       unassigned      YES unset  administratively down down
GigabitEthernet7       unassigned      YES unset  up                    up
GigabitEthernet8       unassigned      YES unset  administratively down down
GigabitEthernet9       unassigned      YES unset  administratively down down
GigabitEthernet10      unassigned      YES unset  administratively down down
GigabitEthernet11      unassigned      YES unset  up                    up
GigabitEthernet12      unassigned      YES unset  administratively down down
BDI101                 172.16.13.17    YES manual up                    up
BDI102                 172.17.13.33    YES manual up                    up
BDI211                 unassigned      YES unset  up                    up
BDI212                 unassigned      YES unset  up                    up
BDI1071                192.168.10.10   YES manual up                    up
BDI2071                192.168.20.22   YES manual up                    up
Loopback0              192.168.0.99    YES manual up                    up
```

Samostatný úkol:

Proveďte kontrolu s Tab. A.1 viz příloha, zda nastavené IPv4 a IPv6 adresy odpovídají nastavení na zařízení. Využijte k tomu vhodné příkazy, nikoliv `show run`.

2. **MSN_A:**

Pro veškerý přenos dat z Cell site, kde se nachází SIAD směrovač, je nutné nastavit *Backhaul* i ze strany MSN.

(a) **Rozhraní vedoucí k SIAD směrovači:**

Rozhraní nastavíme stejným způsobem jako směrem od SIAD směrovače. Nejprve vytvoříme *BDI1071*, který přiřadíme v případě MSN_A k fyzickému portu *Gi2*. Přidělená IPv4 i IPv6 adresa má v posledním bytu o jedničku nižší hodnotu než adresy přiřazené ze strany SIAD směrovače.

```
interface BDI1071
 description Backhaul Primary interface to SIAD
 ip address 192.168.10.9 255.255.255.252
 ipv6 address 2001:506:4047:C5::1/64

interface GigabitEthernet2
 description Backhaul Primary interface to SIAD
 service instance 1071 ethernet
  encapsulation dot1q 1071
  bridge-domain 1071
 exit
```

(b) **Rozhraní mezi MSN_A a MSN_B:**

- Za účelem zvýšení redundance pracují MSN směrovače v páru. Tok dat mezi oběma zařízeními je značně vysoký a proto jsou propojeny fyzicky dvěma linkami: mezi porty *g3-g3* a *g5-g5*. Tyto fyzické linky jsou spojeny (agregovány) pomocí port-channelu do jedné logické linky protokolem LACP.
- Nejprve vytvoříme logická rozhraní *BDI90* a *BDI91*, která přidělíme do stejného `port-channel1`. Opět jsou jim přiděleny IPv4 a IPv6 adresy viz Tab. A.1.

```
interface BDI90
description OSPF interchassis link Area 0
 ip address 10.200.10.30 255.255.255.252
 ipv6 address 2001:506:4600:C0C9::1/64
 encapsulation dot1Q 90

interface BDI91
description OSPF Inter-chassis link Area 10
 ip address 10.200.10.50 255.255.255.252
 ipv6 address 2001:506:4600:C154::1/64
 encapsulation dot1Q 91
```

```
interface Port-channel1
 description LACP Link between MSN_A and MSN_B
 service instance 90 ethernet
  encapsulation dot1q 90
  bridge-domain 90
 service instance 91 ethernet
  encapsulation dot1q 91
  bridge-domain 91
```

- Následně aktivujeme port-channel na fyzickém rozhraní *g3* a *g5* příkazem `channel-group 1 mode active`. `1`, protože se jedná o port-channel 1. Důležitý je také příkaz `active` spouštějící stav vyjednávání, čímž port iniciuje komunikaci s jinými porty odesláním paketů LACP.

```
interface GigabitEthernet3
 channel-group 1 mode active

interface GigabitEthernet5
 channel-group 1 mode active
```

✎ Ověříme-li stav port-channel 1 příkazem `show etherchannel summary`, zjistíme, že **Po1** je RD - tedy ve stavu DOWN.

show etherchannel summary

```
MSN_A#sh etherchannel summary
Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling

Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1       Po1(RD)         LACP     Gi3(susp) Gi5(susp)

RU - L3 port-channel UP State
SU - L2 port-channel UP state
P/bndl -  Bundled
S/susp  - Suspended
```

(c) **Rozhraní mezi MSN_A a PE:**

- Také v případě spojení mezi MSN_A a PE použijeme dvě agregované linky a to mezi porty *g6-g1* a *g7-g3*. Opět z důvodu redundance. Nejsou zde použita logická rozhraní BDI, jako tomu bylo mezi MSN směrovači, protože mezi MSN a PE budou později v tomto Scénaři 1 implementovány odlišné směrovací protokoly. Naopak je zde konfigurace vložena na samotný port-channel.

- Mezi MSN_A a PE je `port-channel11`. Adresní rozsah je přidělen z Tab. A.1 viz příloha. Pokud by došlo k výpadku jedné z agregovaných linek, hrozil by vznik úzkého hrdla. Proto je přidán příkaz `lacp min-bundle 2`, který při výpadku vyřadí celý port-channel 11 a přepne na záložní MSN_B.

```
interface Port-channel11
 description LACP Link between MSN_A and PE
 ip address 77.66.55.46 255.255.255.252
 ipv6 address 2001:506:4600:8228::2/64
 lacp min-bundle 2
```

- Na závěr příkazem `channel- group 11 mode active` přiřadíme port-channel 11 k portům *g6* a *g7*, aby se spustil agregační protokol LACP. Navíc zde spustíme `lacp rate fast`, který způsobí odesílání LACP kontrolních paketů na porty podporující LACP protokol každou sekundu. Normální doba je přitom každých 30 sekund.

```
interface GigabitEthernet6
 description link to PE
 channel-group 11 mode active
 lacp rate fast

interface GigabitEthernet7
 description link to PE
 channel-group 11 mode active
 lacp rate fast
```

✎ Ověříme-li toto nastavení příkazem `show etherchannel summary` zjistíme, že **Po11** je RM - tedy „not in use“, protože není splněna podmínka minimálního počtu linek. PE v tomto stádiu ještě není nakonfigurováno.

show etherchannel summary

```
MSN_A#sh etherchannel summary
Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling

Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1       Po1(RD)         LACP     Gi3(susp) Gi5(susp)
11      Po11(RM)               LACP     Gi6(susp) Gi7(susp)

RU - L3 port-channel UP State
SU - L2 port-channel UP state
P/bndl -  Bundled
S/susp  - Suspended
```

(d) **Nastavení Loopbacků:**

Předposledním krokem pro nastavení rozhraní na MSN_A je vytvoření Loopbacků. Podobně jako tomu bylo u SIAD směrovače, vytvoříme *Loopback0* udávající ID směrovače. V případě MSN navíc vytvoříme i *Loopback20*, který poslouží k identifikaci při utvoření vztahů mezi MSN_A a MSN_B. *Loopback20* obsahuje i IPv6 adresu.

```
interface Loopback0
 description OSPF Router ID
 ip address 192.168.0.11 255.255.255.255

interface Loopback20
 description CORE-OSPF-ROUTER-ID
 ip address 192.168.200.11 255.255.255.255
 ipv6 address 2001:506:4600:8E1::2/128
```

(e) **Rozhraní vedoucí k SNMP serveru:**

Posledním krokem je v případě MSN_A nastavení IP adresy na portu *g1* spojeným se SNMP serverem, který budeme dále nastavovat v Kroku 8.

```
interface GigabitEthernet1
 ip address 10.10.10.9 255.255.255.0
```

Samostatný úkol:

Proveďte kontrolu s Tab. A.1 viz příloha, zda nastavené IPv4 a IPv6 adresy odpovídají nastavení na zařízení. Využijte k tomu vhodné příkazy, nikoliv `show run`.

3. **MSN_B:**

Backhaul směrem k SIAD směrovači nastavíme podobně jako tomu bylo u MSN_A.

(a) **Rozhraní vedoucí k SIAD směrovači:**

V SIAD směrovači je použit záložní *BDI2071* přiřazený k *Gi4* rozhraní. Přidělené IP adresy mají v posledním bytu o jedničku nižší hodnotu IP adresy než adresy přiřazené na SIAD směrovači.

```
interface BDI2071
 description Backhaul Secondary interface to SIAD
 ip address 192.168.20.21 255.255.255.252
 ipv6 address 2001:506:4047:40C5::1/64

interface GigabitEthernet4
 description Backhaul Secondary interface to SIAD
 service instance 2071 ethernet
  encapsulation dot1q 2071
  bridge-domain 2071
```

(b) **Rozhraní mezi MSN_B a MSN_A:**

Konfigurace spojení mezi MSN_B a MSN_A je totožná. Jsou zde logická rozhraní *BDI90* a *BDI91*, která také náleží do `port-channel1`. IPv4 mají v posledním bytu o jedničku nižší hodnotu IP adresy od IP adres přiřazených na MSN_A, IPv6 má naopak v posledním bytu o jedničku vyšší hodnotu od IP adres přiřazených na MSN_A. Port-channel 1 přiřadíme na fyzické porty *g3* a *g5*.

```
interface BDI90
 ip address 10.200.10.29 255.255.255.252
 ipv6 address 2001:506:4600:C0C9::2/64
 encapsulation dot1Q 90

interface BDI91
 ip address 10.200.10.49 255.255.255.252
 ipv6 address 2001:506:4600:C154::2/64
 encapsulation dot1Q 91
```

```
interface Port-channel1
 description LACP Link between MSN_B and MSN_A
 service instance 90 ethernet
  encapsulation dot1q 90
  bridge-domain 90
 service instance 91 ethernet
  encapsulation dot1q 91
  bridge-domain 91

interface GigabitEthernet3
 channel-group 1 mode active

interface GigabitEthernet5
 channel-group 1 mode active
```

(c) **Rozhraní mezi MSN_B a PE:**

V případě spojení mezi MSN_B a PE nepoužijeme logické BDI rozhraní, ale vytvoříme zde pouze `port-channel12`, který agreguje dvě linky mezi porty *g6-g2* a *g7-g4*. Ochrana před úzkým hrdlem příkazem `lacp min-bundle 2` je také zajištěna. Port-channel 12 přidělíme na porty *g6* a *g7* pomocí `channel-`

`group 12 mode active`. Je zde aktivována i rychlá výměna LACP paketů `lacp rate fast`.

```
interface Port-channel12
 description LACP Link between MSN_B and PE
 ip address 77.66.55.86 255.255.255.252
 ipv6 address 2001:506:4600:822A::2/64
 lacp min-bundle 2

 interface GigabitEthernet6
 description link to PE
 channel-group 12 mode active
 lacp rate fast

interface GigabitEthernet7
 description link to PE
 channel-group 12 mode active
 lacp rate fast
```

✎ Ověříme-li toto nastavení příkazem `show etherchannel summary` zjistíme, že **Po1** je již ve stavu RU - tedy „in use“, protože obě strany jsou již nakonfigurovány. **Po12** je ve stavu RM, neboť ani v tomto případě není ještě PE nakonfigurován.

show etherchannel summary

```
MSN_B#sh etherchannel summary
Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling

Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1       Po1(RU)         LACP     Gi3(bndl) Gi5(bndl)
12      Po12(RM)               LACP     Gi6(susp) Gi7(susp)

RU - L3 port-channel UP State
SU - L2 port-channel UP state
P/bndl -  Bundled
S/susp  - Suspended
```

(d) **Nastavení Loopbacků:**

Podobně jako tomu bylo u MSN_A jsou i zde jsou vytvořeny *Loopback0* pro ID směrovače a *Loopback20* pro komunikaci a identifikaci mezi MSN párem.

```
interface Loopback0
 description OSPF Router ID
 ip address 192.168.0.22 255.255.255.255

interface Loopback20
 description CORE-OSPF-ROUTER-ID
 ip address 192.168.200.22 255.255.255.255
 ipv6 address 2001:506:4600:8E9::2/128
```

✎ Nastavení a stav portů ověříme následujícím příkazem:

show ip interface brief

```
MSN_B#sh ip int br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       unassigned      YES unset  administratively down down
GigabitEthernet2       unassigned      YES unset  administratively down down
GigabitEthernet3       unassigned      YES unset  up                    up
GigabitEthernet4       unassigned      YES unset  up                    up
GigabitEthernet5       unassigned      YES unset  up                    up
GigabitEthernet6       unassigned      YES unset  up                    up
GigabitEthernet7       unassigned      YES unset  up                    up
BDI90                  10.200.10.29    YES manual up                    up
BDI91                  10.200.10.49    YES manual up                    up
BDI2071                192.168.20.21   YES manual up                    up
Loopback0              192.168.0.22    YES manual up                    up
Loopback20             192.168.200.22  YES manual up                    up
Port-channel1          unassigned      YES unset  up                    up
Port-channel12         77.66.55.86     YES manual down                  down
```

Samostatný úkol:

Proveďte kontrolu s Tab. A.1 viz příloha, zda nastavené IPv4 a IPv6 adresy odpovídají nastavení na zařízení. Využijte k tomu vhodné příkazy, ne `show run`.

4. **PE:**

Jak již bylo zmíněno v úvodu Scénáře 1, PE bude konfigurován pouze z pohledu směrovacího protokolu BGP. Jeho konfigurace se tím značně zjednoduší.

(a) **Rozhraní vedoucí k MSN_A a MSN_B:**

Na PE vytvoříme nejprve `port-channel11` pro spojení s MSN_A a poté `port-channel12`. Jejich konfigurace je stejná jako tomu bylo na MSN páru. IPv4 a IPv6 adresy mají v posledním bytu o jedničku nižší hodnotu než IP adresy přidělené na MSN_A a MSN_B.

```
interface Port-channel11
 description LACP Link between PE and MSN_A
 ip address 77.66.55.45 255.255.255.252
 ipv6 address 2001:506:4600:8228::1/64
 lacp min-bundle 2

interface Port-channel12
 description LACP Link between PE and MSN_B
 ip address 77.66.55.85 255.255.255.252
 ipv6 address 2001:506:4600:822A::1/64
 lacp min-bundle 2
```

Port-channel je nutné přiřadit k fyzickým portům. Příkaz `channel-group 11 mode active` přiřadíme na *g1* a *g3*. `Channel-group 12 mode active` zase přiřadíme na *g2* a *g4*. Je důležité neopomenout `lacp rate fast`.

```
interface GigabitEthernet1
 description link to MSN_A
 channel-group 11 mode active
 lacp rate fast
 no shutdown

interface GigabitEthernet3
 description link to MSN_A
 channel-group 11 mode active
 lacp rate fast
 no shutdown

interface GigabitEthernet2
 description link to MSN_B
 channel-group 12 mode active
 lacp rate fast
 no shutdown

interface GigabitEthernet4
 description link to MSN_B
 channel-group 12 mode active
 lacp rate fast
 no shutdown
```

✎ Nyní by již port-channel měl fungovat. Jeho funkčnost je možné ověřit příkazem uvedeným níže.

show etherchannel summary

```
PE#sh etherchannel summary
Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling

Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
11      Po11(RU)                LACP     Gi1(bndl) Gi3(bndl)
12      Po12(RU)                LACP     Gi2(bndl) Gi4(bndl)

RU - L3 port-channel UP State
SU - L2 port-channel UP state
P/bndl -  Bundled
S/susp  - Suspended
```

(b) **Nastavení Loopbacků:**

K identifikaci směrovače PE musí být vytvořen *Loopback0*.

```
interface Loopback0
 description OSPF Router ID
 ip address  192.168.0.77 255.255.255.255
```

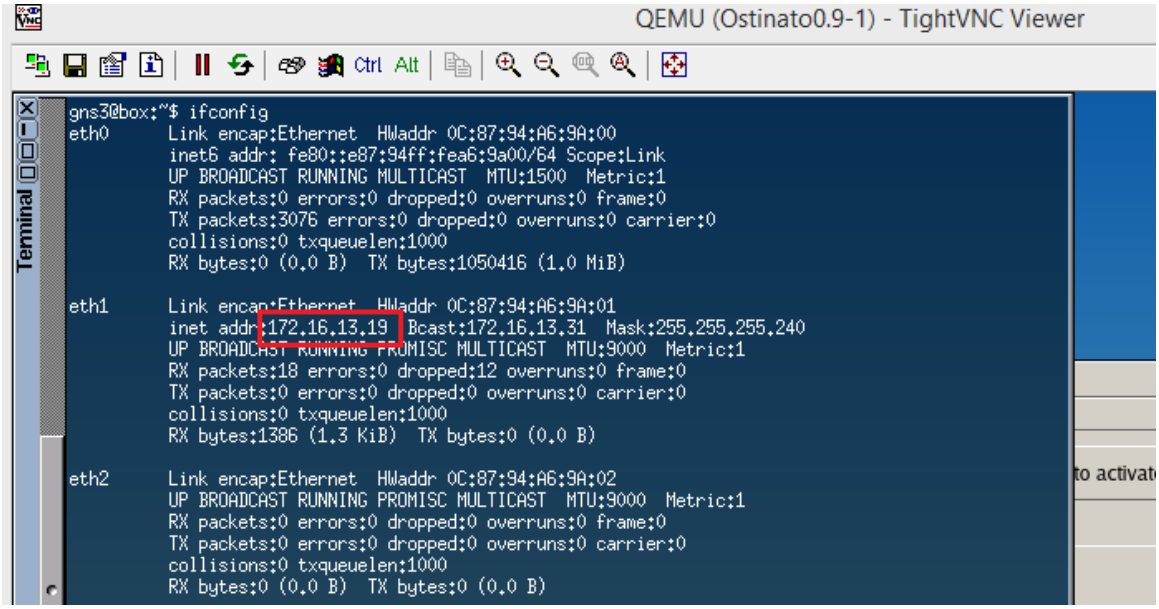
Samostatný úkol:

Ověřte zda je plně funkční ochrana před vznikem úzkého hrdla pomocí stanoveného minimálního počtu 2 funkčních linek tak, aby protokol LACP byl „in use“. Např. administrativně vypněte rozhraní *g1* na MSN_A směrovači a sledujte změnu na `port-channel 11`.

5.1.2 Krok 2. - nastavení simulátoru Ostinato

Ostinato je síťový generátor a je využit v tomto scénáři pro simulaci mobilních základnových stanic NodeB a eNodeB. Především je pak využit pro ověření nastavení QoS v Kroku 9. NodeB a eNodeB mají přidělené adresy viz Tab. A.1. Postup při jejich nastavování se příliš neliší, až na rozdíl v použití IPv4 a IPv6. Proto se vysvětlení a postup vztahuje pouze k NodeB, které využívá IPv4.

Před zapnutím samotného programu Ostinato nastavíme výchozí bránu v terminálu na *eth1*. Pro NodeB se bude jednat o příkaz `sudo ifconfig eth1 172.16.13.19 netmask 255.255.255.240`. Příklad nastavení je zobrazen na Obr. 5.1.

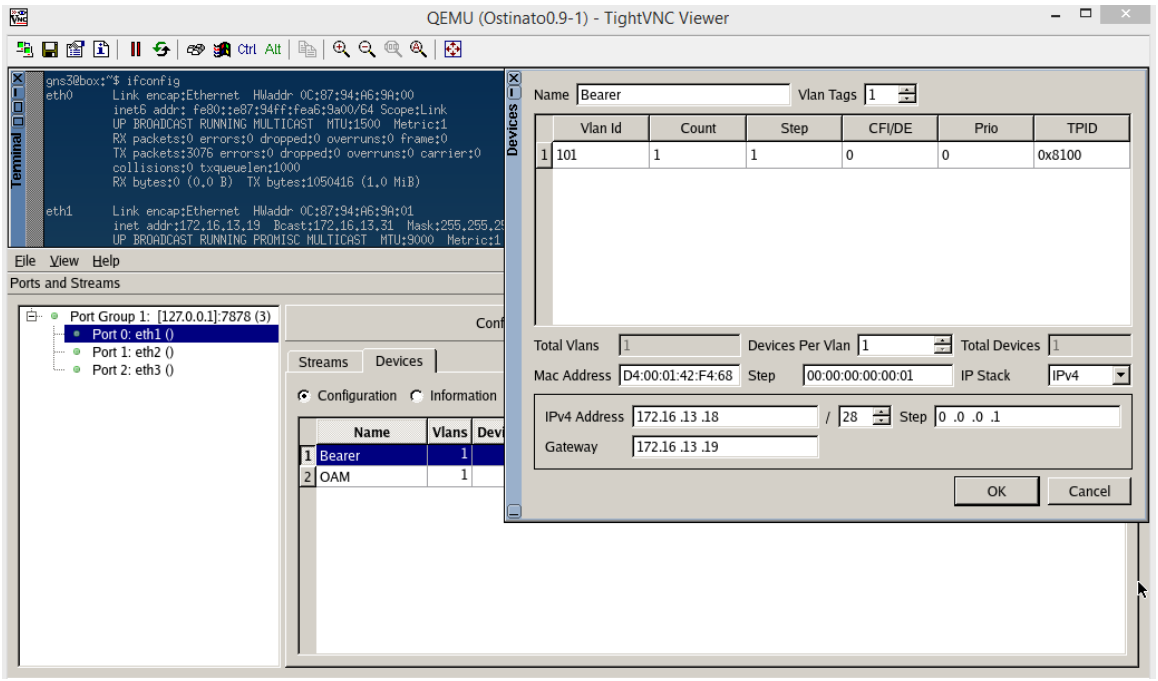


Obr. 5.1: IP adresy přidělené na port eht1

Poté spustíme program Ostinato a v levém rohu zvolíme *Port0: eth1*, klikneme na záložku *Devices*, vytvoříme zařízení s názvem „Bearer“ a přidělíme mu IPv4 dle Tab. A.1 s výchozí bránou na *eth1*. *Mac Address* ponecháme výchozí. Nastavení lze vidět na Obr. 5.2. To stejné provedeme pro „OAM“.

Po vytvoření „Bearer“ a „OAM“ nastavíme tok dat záložkou *Streams*. Pro ukázku zvolíme stream „payload_bearer“, který simuluje Real Time aplikace, např. „3G Voice“.

Pravým tlačítkem klikneme na *New Stream*. Pojmenujeme jej v kolonce *name*. V záložce *Protocol Selection* nastavíme následující: **L1** ponecháme výchozí, **VLAN** zvolíme „Tagged“, **L2** zaklikneme „Ethernet II“, **L3** v tomto případě bude „IPv4“ a u **L4** bude vybráno „UDP“. Zbylé nastavení v této záložce ponecháme ve výchozím stavu.

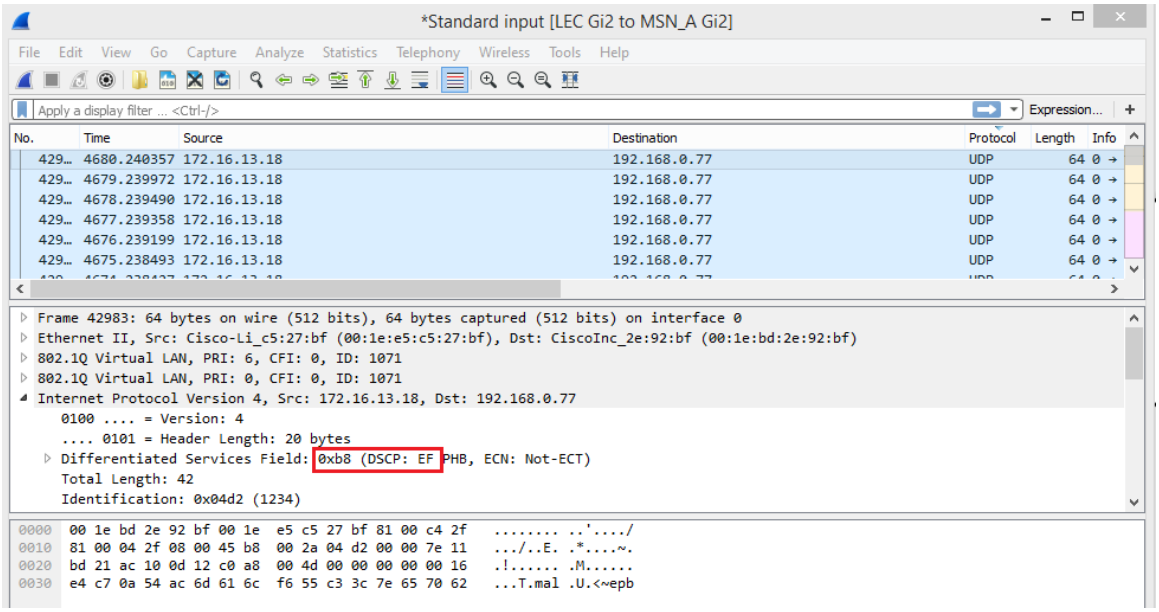


Obr. 5.2: Vytvoření a nastavení „Bearer“ v programu Ostinato

Poté se překlikne do záložky *Protocol Data*. V podzáložce **Media Access Protocol** jako destinaci zvolíme broadcast adresu „FF:FF:FF:FF:FF:FF“, aby docházelo k odesílání rámců na libovolná rozhraní na SIAD směrovači. Jako zdroj však musíme zvolit MAC adresu portu *eth1*, v tomto případě „0C:87:94:A6:9A:01“, aby rámce obsahovaly konkrétní zdrojovou adresu. V podzáložce **Vlan** nastavíme VLAN 101, protože nyní nastavujeme tok dat pro Bearer. V další podzáložce **Internet Protocol ver 4** v poli TOS/DSCP vložíme hodnotu „B8“, protože chceme simulovat 3G provoz, kterému náleží EF u DSCP. Jelikož v Ostinato se vkládá celé pole (DSCP: EF PHB, ECN: Not-ECT), musíme převzít hodnotu v hexadecimální podobě 0xb8 a nemůžeme vložit pouze EF. Jako zdroj zvolíme IPv4 adresu „172.16.13.18“ a jako destinaci adresu *Loopback0* na PE „192.168.0.77“. V poslední podzáložce **Payload Data** zvolíme Type „Random“. Zbytek ponecháme v původním nastavení.

Po dokončení nastavení klikneme na námi vytvořený stream, v pravém horním rohu stiskneme tlačítko *Apply* a nakonec v levém spodním rohu stiskneme tlačítko *Start Transmit*. Přidání Streamu pro „OAM“ by opakovalo stejný postup jako přidání Streamu pro „Bearer“.

✎ Pro ověření správnosti nastavení toku dat provedeme kontrolu pomocí programu WireShark např. na spojení mezi LEC a MSN_A viz Obr. 5.3:



Obr. 5.3: Zachycení paketu s DSCP značkou EF pro ověření správnosti nastavení toku dat

Samostatný úkol:

Nastavte obdobným způsobem provoz pro eNodeB. Ten využívá IPv6 adresní prostor. Nejprve musíte nastavit výchozí bránu pro *eth1* dle Tab. A.1 viz příloha a poté přes záložku *Devices* zařízení „Bearer“ a „OAM“. Nakonec se zaměřte na nastavení toku dat přes záložku *Streams*. Využijte k tomu znalosti z Kapitoly 2.

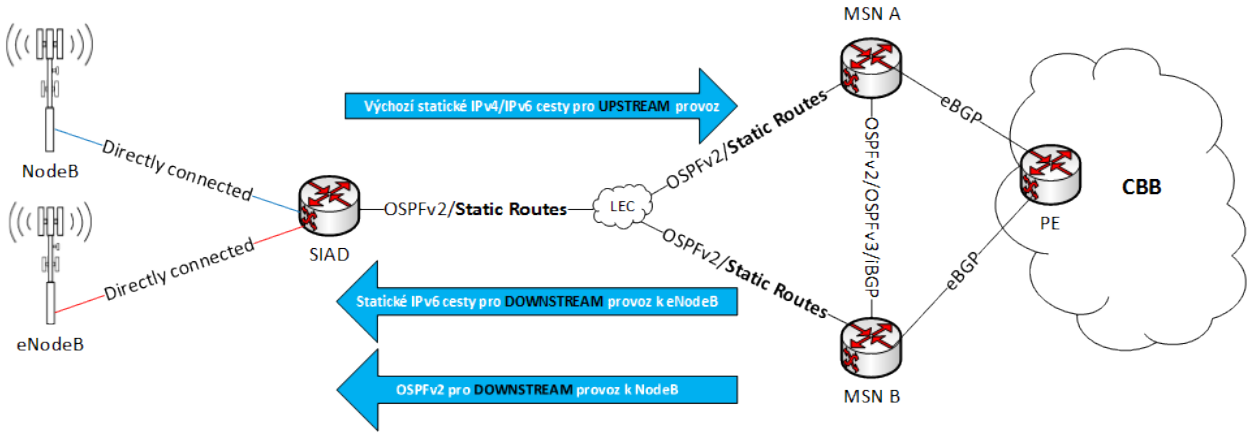
5.1.3 Krok 3. - nastavení statického směrování

Statické směrování hraje velkou roli v mobilní transportní síti. V případě *Shorthaul* linky je však jeho využití specifické. NodeB a eNodeB jsou k SIAD směrovači přímo připojeny („directly connected“). Statické směrování se používá k prevenci před nežádoucím provozem jako je například provoz z cizích zdrojů nebo vytváření smyček. Tento způsob využití statického směrování se nazývá „Black Hole Routing“. Mezi NodeB a SIAD směrovačem jsou IPv4 statické cesty s touto ochranou. Mezi eNodeB a SIAD směrovačem je tato ochrana použita pro IPv6 statické cesty.

V případě *Backhaul* linky se statické IPv4 a IPv6 cesty využívají pro „upstream“ přenos dat mezi SIAD směrovačem a MSN párem. Pro IPv6 je důležité nastavení statických cest i v opačném směru z MSN páru. Vytvoří to cestu zpět, tedy „downstream“ přenos dat pro eNodeB. Pro NodeB je „downstream“ přenos dat přenášen protokolem OSPFv2, viz Krok 4.

Důvodů, proč ani pro IPv6 „downstream“ provoz není využito dynamického protokolu je několik, především je to však z historického důvodu. 4G síť (eNodeB) pracující čistě na IPv6 protokolu byla vytvořena ke stávající 3G síti až dodatečně. Jelikož ale 3G síť používala protokol OSPFv2, nebylo možné tento protokol pro IPv6 použít. Při dodatečné implementaci OSPFv3 podporující IPv6 adresy by hrozila příliš velká zátěž na dané směrovače a proto je využíváno statického směrování.

Pro ujasnění uvádíme Obr. 5.4, na kterém lze vidět použití statických cest v kombinaci s dalšími protokoly. Ty budou podstatou Kroků 4.-6. Statické směrování z pohledu NodeB a eNodeB je nastaveno již v Kroku 2.



Obr. 5.4: Využití statických cest spolu s dalšími protokoly

1. **SIAD:**

   Implementace statických cest se rozlišuje podle zařízení, které se SIAD směrovačem sousedí.

   (a) **Spojení s NodeB:**

   NodeB je přímo připojen („directly connected“) se SIAD směrovačem. Spojení mezi nimi je přes *BDI101* - Bearer a *BDI102* - OAM, jak jsme viděli v Kroku 1. IP adresy na NodeB mají v posledním bytu o jedničku nižší hodnotu než na SIAD směrovači, patří však do stejné podsítě a jsou tedy obsaženy ve směrovací tabulce.
   Nastavení statických cest je specifické a nese název „Black Hole Routing“ - černé díry. Spočívá ve využití rozhraní `Null0` a slouží jako prevence před vytvářením smyček a nežádoucímu provozu. Konfigurace této cesty se zadává v tomto pořadí `ip route <prefix> <maska> <Null0> <libovolné specifické nastavení>`. Pomocí tohoto příkazu se vytvoří černá díra pro konkrétní IP adresy.
   Zjednodušeně řečeno, paket směrovaný do dané podsítě bude nejprve hledat specifickou cestu do cíle. Pokud však specifická cesta neexistuje, nebo ji paket nenajde, další nejlepší shoda bude `ip route` s `Null0` rozhraním a paket bude zahozen do černé díry.
   V tomto případě by měl paket směřující na NodeB najít shodu ve směrovací tabulce, kde je záznam o přímo připojených („directly connected“) cestách. Pokud by však dané NodeB nebylo funkční, záznam by chyběl a díky statické cestě s Null0 by byl paket zahozen a nezatěžoval by SIAD směrovač.
   Konkrétní záznam pro IP adresy připojené přes *BDI101* - Bearer a *BDI102* - OAM vypadají následovně.

```
ip route 172.16.13.16 255.255.255.240 Null0 name SIAD_NB_Interfaces
ip route 172.17.13.32 255.255.255.248 Null0 name SIAD_NB_Interfaces
```

   (b) **Spojení s eNodeB:**

   Pro eNodeB platí stejný princip jako u statických cest pro NodeB. Musí však pracovat s IPv6 adresami připojenými přes *BDI211* a *BDI212*. Příkaz pro nakonfigurování„Black Hole Routing“ je `ipv6 route <prefix/maska> <Null0>`. Jako <prefix/maska> je použita adresa IPv6 sítě a jako maska /61, která se používá pro Bearer a OAM podsíť. Maska /64, která je použita pro *BDI211* a *BDI212* se naopak používá pro point-to-point spojení mezi zařízeními, viz Tab. A.1.

```
ipv6 unicast-routing
ipv6 route 2001:506:4247:340::/61 Null0
ipv6 route 2001:506:4447:340::/61 Null0
```

(c) **Statické cesty k MSN páru:**

Připomeňme si, že veškerý „upstream“ provoz od SIAD směrovače je směrován pomocí statických cest. Pro určení výchozí statické cesty se nabízí dvě možnosti konfigurace `ip default-network x.x.x.x` nebo `ip route 0.0.0.0 0.0.0.0`. Z důvodu nutnosti bližší specifikace kvůli *BDI1071*, *BDI2071* a odlišné administrativní vzdálenosti je využita druhá možnost.
Přesná konfigurace použitá na IOS XE má následující formát:

```
ip route <prefix> <maska> <rozhraní> <cílová IP adresa> <adm.vzdálenost> <jméno>
```

Statické cesty je zapotřebí v tomto formátu vytvořit pro obě logická rozhraní *BDI1071* a *BDI2071*. V případě primární cesty směřující k MSN_A skrze *BDI1071* nenastavujeme `<adm.vzdálenost>`. Ponecháme zde výchozí hodnotu administrativní vzdálenosti, a to 1. Pro sekundární cestu směřující k MSN_B skrze *BDI2071* je administrativní vzdálenost nastavena na 10 tak, aby vždy byla upřednostňovaná primární cesta. Hodnotu 10 zvolíme i z toho důvodu, že je to méně než výchozí administrativní hodnota ostatních protokolů, které by mohly ovlivnit směrování na SIAD směrovači.

```
ip route 0.0.0.0 0.0.0.0 BDI1071 192.168.10.9 name PRIMARY_LINK
ip route 0.0.0.0 0.0.0.0 BDI2071 192.168.20.21 10 name SECONDARY_Link
```

V případě IPv6 je logika totožná.

```
ipv6 route ::/0 BDI1071 2001:506:4047:C5::1 name Primary_Link
ipv6 route ::/0 BDI2071 2001:506:4047:40C5::1 10 name Secondary_Link
```

✎ Zda byly statické cesty vloženy správně ověříme pomocí příkazu:

show run | i route

```
SIAD#sh run | i route
ip route 0.0.0.0 0.0.0.0 BDI1071 192.168.10.9 name PRIMARY_LINK
ip route 0.0.0.0 0.0.0.0 BDI2071 192.168.20.21 10 name SECONDARY_Link
ip route 172.16.13.16 255.255.255.240 Null0 name SIAD_NB_Interfaces
ip route 172.17.13.32 255.255.255.248 Null0 name SIAD_NB_Interfaces
ipv6 route 2001:506:4247:340::/61 Null0
ipv6 route 2001:506:4447:340::/61 Null0
ipv6 route ::/0 BDI2071 2001:506:4047:40C5::1 10 name Secondary_Link
ipv6 route ::/0 BDI1071 2001:506:4047:C5::1 name Primary_Link
```

2. **MSN_A:**

   Statické cesty z MSN_A směrem k SIAD směrovači se týkají pouze IPv6 cest, jak bylo zmíněno v úvodu Kroku 3. Je přes ně směrován veškerý provoz pro eNodeB. IPv4 využívá naopak protokol OSPFv2, který bude popsán v Kroku 4. Žádné jiné statické cesty MSN směrovače nevyužívají.
   IPv6 cesty jsou směrovány s prefixem /61 pro Bearer a OAM na konkrétní logické rozhraní na SIAD směrovači. Administrativní vzdálenost je zde ponechána výchozí.

```
ipv6 unicast-routing
ipv6 route 2001:506:4247:340::/61 BDI1071 2001:506:4047:C5::2
ipv6 route 2001:506:4447:340::/61 BDI1071 2001:506:4047:C5::2
```

   I v případě MSN se využívá „Black Hole Routing“. Opět platí, že pokud se nenajde lepší shoda, je paket přesměrován do rozhraní `Null0`, kde je zahozen. Daná `ipv6 route` obsahuje podsíť zahrnující všechny Bearer IPv6 adresy s maskou /50. Do této podsítě spadají všechny eNodeB pro daný MSN. Jelikož MSN směrovače mohou mít v reálném prostředí na sebe připojeno 200-400 SIAD směrovačů, může se jednat o 200-1000 eNodeB.

```
ipv6 route 2001:506:4247::/50 Null0
```

3. **MSN_B:**

   Z MSN_B k SIAD směrovači vedou stejné statické cesty. IPv6 adresa prefixu i maska pro Bearer a OAM je totožná, pouze pro rozhraní je použit *BDI2071* s přidělenou adresou na SIAD směrovači. Podstatný je také fakt, že jsme zde změnili administrativní vzdálenost na 200 tak, aby byla upřednostňována primární statická cesta z MSN_A.

```
ipv6 unicast-routing
ipv6 route 2001:506:4247:340::/61 BDI2071 2001:506:4047:40C5::2 200
ipv6 route 2001:506:4447:340::/61 BDI2071 2001:506:4047:40C5::2 200
```

✎ Následujícím příkazem ověříme, zda jsou statické cesty zaznamenány ve směrovací tabulce:

show ipv6 route

```
MSN_B#show ipv6 route
IPv6 Routing Table - default - 12 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, RL - RPL, O - OSPF Intra, OI - OSPF Inter
       OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1
       ON2 - OSPF NSSA ext 2, la - LISP alt, lr - LISP site-registrations
       ld - LISP dyn-eid, a - Application
C   2001:506:4047:40C5::/64 [0/0]
     via BDI2071, directly connected
L   2001:506:4047:40C5::1/128 [0/0]
     via BDI2071, receive
S   2001:506:4247:340::/61 [200/0]
     via 2001:506:4047:40C5::2, BDI2071
S   2001:506:4447:340::/61 [200/0]
     via 2001:506:4047:40C5::2, BDI2071
LC  2001:506:4600:8E9::2/128 [0/0]
     via Loopback20, receive
C   2001:506:4600:822A::/64 [0/0]
     via Port-channel12, directly connected
L   2001:506:4600:822A::2/128 [0/0]
     via Port-channel12, receive
C   2001:506:4600:C0C9::/64 [0/0]
     via BDI90, directly connected
L   2001:506:4600:C0C9::2/128 [0/0]
     via BDI90, receive
C   2001:506:4600:C154::/64 [0/0]
     via BDI91, directly connected
L   2001:506:4600:C154::2/128 [0/0]
     via BDI91, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Na základě této konfigurace bude po nastavení zbytku sítě možné vidět, že provoz směřující zvenčí na eNodeB přes MSN_B bude směrován přes MSN_A, přes linku mezi MSN párem.

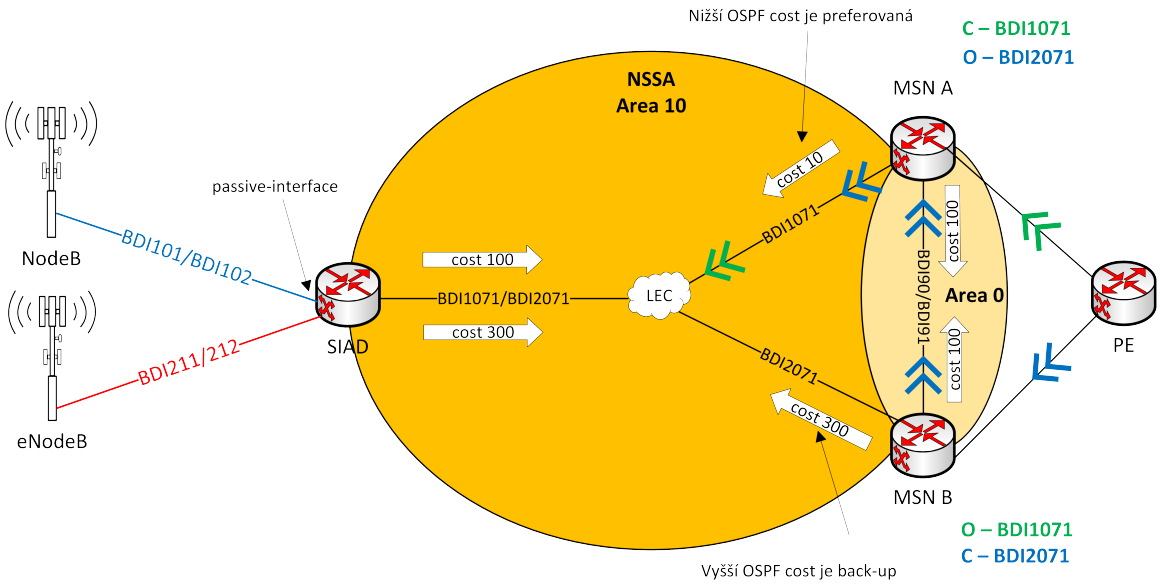
Samostatný úkol:

Proveďte testování pomocí příkazu `ping` ze SIAD směrovače na adresy portu Backhaul linky na MSN_A a MSN_B. Z MSN páru pak stejný test proveďte pro IPv6 adresu rozhraní *g7* na SIAD směrovači. Ping test by měl být úspěšný.

5.1.4 Krok 4. - nastavení OSPFv2

Kromě statických cest se mezi SIAD směrovačem a MSN párem nachází protokol OSPFv2. Je použit také mezi samotným MSN párem pro komunikaci přes IPv4.

Hlavní úlohou OSPFv2 není přenos dat ve směru ze SIAD směrovače k MSN páru - k tomu slouží právě statické cesty viz Obr. 5.4 - ale oznamování připojených sítí MSN páru. Přestože z MSN páru informace o připojených sítích k SIAD směrovači nejdou, je tímto směrem pomocí OPSFv2 posílán provoz pro NodeB (3G). Toho dosáhneme rozdělením transportní sítě do oblastí a také použitím více OSPF procesů na MSN směrovačích. Na Obr. 5.5 je znázorněno celkové shrnutí OSPFv2 pro Scénař 1.



Obr. 5.5: Směrovací schéma pro OSPFv2 - tok dat je vyjádřen dvojitou modrou/zelenou šipkou

SIAD směrovač patří do oblasti 10 (`router ospf 10`) a MSN do oblasti 10 (`router ospf 1`) a 0 (`router ospf 20`).

Oblast 10 je nastavená jako NSSA (Not-So-Stubby). NSSA oblast se používá v situaci, kdy má oblast z pohledu OSPF pouze jednu cestu ven, ale může mít cestu i přes jiný směrovací protokol. Tyto cesty vedoucí přes jiný směrovací protokol jsou posílány do OSPF pomocí LSA 7 zpráv.

Díky NodeB a eNodeB, které jsou přímo připojeny ("directly connected"), se na SIAD směrovači negenerují LSA zprávy typu 7. Na MSN u `router ospf 1` se naopak LSA zprávy typu 7 generují a to díky BGP, přes které MSN komunikuje s PE. Tyto zprávy jsou ale z bezpečnostního a zátěžového hlediska ve směru z MSN

k SIAD směrovači filtrovány. Zasílány jsou pouze LSA zprávy typu 1. Toho dosáhneme na MSN příkazem `area 10 nssa no-redistribution no-summary`.

Oblast 0 na MSN páru spadá pod `router ospf 20`. Obsahuje IP adresy Loopbacků směrovačů, které spadají pod oblast 0. OSPFv2 proces ID 20 zajišťuje spojení a komunikaci mezi MSN_A a MSN_B a přenos Loopbacků pro iBGP protokol, který bude popsán v Kroku 6.

1. **SIAD:**

   OSPFv2 nejprve spustíme na samotném zařízení. V tomto případě zadáme proces ID 10 (`router ospf 10`) a přidáme příslušné sítě. Detailnější nastavení OSPF se pak nakonfiguruje na samotném BDI.

   (a) **Nastavení samotného OSPFv2:**

   OSPFv2 na SIAD směrovači je přiřazeno pod proces ID 10, toto číslo zároveň i odpovídá číslu oblasti, tedy 10. Proces ID však může být libovolné číslo. Pro směrování nemá žádný význam, je to pouze vnitřní označení směrovače. Číslo oblasti je nastaveno příkazem `area 10 nssa`. NSSA označuje oblast jako Not-So-Stubby.
   Jako `router-id` je zvolena IP adresa *Loopback0* a to z toho důvodu, že tento port je vždy ve stavu UP. Protokol OSPFv2 navíc tuto adresu oznámí ostatním směrovačům, podobně jako sítě připojené směrem k NodeB a sítě připojené přes BDI u *Backhaul* linky.

```
router ospf 10
 router-id 192.168.0.99
 area 10 nssa
 network 172.16.13.16 0.0.0.15 area 10  //network connected through BDI101
 network 172.17.13.32 0.0.0.7 area 10   //network connected through BDI102
 network 192.168.0.99 0.0.0.0 area 10   //IPv4 address of Loopback0
 network 192.168.10.8 0.0.0.3 area 10   //network connected through BDI1071
 network 192.168.20.20 0.0.0.3 area 10  //network connected through BDI2071
```

   V globálním konfiguračním módu je nutné pod `router ospf 10` přidat také pasivní rozhraní. Pasivní rozhraní jsou ta, na která se neodesílají směrovací updaty. Je to především z důvodu bezpečnosti. Týká se to BDI101 a BDI102, tedy rozhraní směřujících k NodeB, který je připojen „Directly connected“.

```
router ospf 10
 passive-interface BDI101
 passive-interface BDI102
```

(b) **Nastavení OSPF na BDI rozhraních:**

OSPFv2 je možno konfigurovat na úrovni samotného logického rozhraní. Díky tomu se rozhraní účastní směrování OSPFv2 a zajistí to i větší možnost odlišného nastavení, například jako zde pro primární a sekundární linku. U SIAD směrovače se to týká *BDI1071* a *BDI2071* rozhraní na *Backhaul* lince. U OSPFv2 je pro vytvoření sousedství nutné, aby obě komunikující strany měly synchronizované časovače. Bude tedy důležité nastavit stejné parametry i na straně MSN páru. Rozlišení, který port a která linka budou mít přednost a stanou se primární u OSPF řeší metrika „cost". Čím nižší metrika, tím má cesta větší přednost. V tomto simulačním scénáři *cost* nastavíme ručně.
Rozhraní *BDI1071* vede k MSN_A a bylo zvoleno dle zadání jako primární, *cost* byla přidělena s hodnotou 100. *Hello interval* byl zvolen na 20 a *dead interval* na 70. Příkazem `ip ospf network point-to-point` je směrovač informován, že se jedná o spojení mezi dvěma zařízeními. Směrovač tak nebude mít snahu volit DR a BDR směrovač a ušetří se tím výpočetní zdroje.

```
interface BDI1071
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 100
```

U *BDI2071* bylo nastavení obdobné. Pouze metrika „cost" byla nastavena rozdílně, pro sekundární linku byla zvolena hodnota 300.

```
interface BDI2071
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 300
```

2. **MSN_A:**

Na MSN je OSPFv2 rozlišeno pomocí proces ID na `router ospf 1` a `router ospf 20`. Proces 1 patří do oblasti 10 a komunikuje se SIAD směrovačem, který je ve stejné oblasti. Proces 20 je využíván na komunikaci mezi MSN párem a spadá do oblasti 0.

(a) **Nastavení samotného OSPFv2:**

Pro vytvoření sousedství se SIAD směrovačem je zapotřebí nastavit `router ospf 1`. Jako `router-id` byl vybrán *Loopback0*. Oblast také musí být označena jako NSSA příkazem `area 10 nssa`. Jak již bylo zmíněno v úvodu Kroku 4, je nutné filtrovat LSA zprávy zasílané k SIAD směrovači. Použit je k tomu příkaz

`no-redistribution` zakazující zasílání LSA typu 7 a příkaz `no-summary`, který zakáže LSA typu 3. Nastaven je i pasivní port. Přidané sítě jsou zde dvě, z nichž první je síť nacházející se za *BDI91*. Ta zajišťuje spojení s MSN_B a jdou přes ni informace o všech připojených IPv4 sítích u SIAD směrovače (OAM,Bearer síť u NodeB, Loopback0 a Backhaul BDI). Druhá je síť, do které spadá *Backhaul* k danému SIAD směrovači. Tato síť má masku /16 a může obsahovat velké množství hostů. Je to proto, že na oblast 10 u MSN směrovače může být v reálu napojeno 200 až 400 SIAD směrovačů.

```
router ospf 1
 router-id 192.168.0.11
 area 10 nssa no-redistribution no-summary
 passive-interface GigabitEthernet1       //port to SNMP server
 network 10.200.10.48 0.0.0.3 area 10     //network connected through BDI91
 network 192.168.0.0 0.0.255.255 area 10  //network connected through BDI1071
```

Následuje nastavení `router ospf 20`. Jako `router-id` je použit *Loopback20*. Oblast je označená 0, takže se jedná o tzv. *Backbone*. Spadají do ní sítě patřící Loopbackům a *BDI90* rozhraní mezi MSN párem. Hlavním účelem OSPFv2 s proces ID 20 je informování sousedního směrovače v páru o jeho *Loopbacku0*. Informace o Loopbacku je dále použita protokolem iBGP, viz Krok 6.

```
router ospf 20
 router-id 192.168.200.11
 network 10.200.10.28 0.0.0.3 area 0    //network connected through BDI90
 network 192.168.0.11 0.0.0.0 area 0    //Loopback0
 network 192.168.200.11 0.0.0.0 area 0  //Loopbak20
```

(b) **Nastavení OSPF na BDI rozhraních:**

Na MSN směrovači musí taktéž dojít ke specifičtějšímu nastavení OSPFv2 na rozhraních. Především musíme nastavit správné a stejné hodnoty u *hello intervalu* a *dead intervalu*, aby došlo k vytvoření sousedství. Je důležité neopomenout ani nastavení `ip ospf network point-to-point`.
V čem se však nastavení primary linky liší, je hodnota metriky „cost“, v tomto případě je to pouze 10. Navíc přibude i příkaz `ip ospf database-filter all out`, který opět filtruje LSA zprávy, podobně jako `no-redistribution` a `no-summary` přímo u `router ospf 1`.

```
interface BDI1071
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 10
 ip ospf database-filter all out
```

Dalším portem, na kterém byl nastaven OSPFv2 je *BDI 90*. Tento port se nachází mezi MSN párem a byl zde využit opět příkaz `ip ospf network`

`point-to-point`. Byly zde stanoveny stejné intervaly jako pro celé OSPFv2. Metrika „cost“ má hodnotu 100. Na tomto portu již nedochází k filtraci LSA zpráv.

```
interface BDI90
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 100
```

Totožná konfigurace pro OSPFv2 platí i u *BDI 91*, které se také nachází mezi MSN párem.

```
interface BDI91
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 100
```

3. **MSN_B:**

OSPFv2 má obdobnou konfiguraci jako MSN_A. Ve skutečnosti může totiž být MSN_B pro některé SIAD směrovače primární. Je to dáno tím, že mezi MSN párem neexistuje „loadbalance“ protokol, zátěž je rozdělována ručně. Toho se dosáhne pomocí metriky „cost“, která je přidělena na BDI rozhraní vedoucí od MSN k danému SIAD směrovači. Pokud je „cost“ na BDI rozhraní 300, MSN se stává alternativním směrovačem. Pokud je „cost“ 10, stává se MSN primárním směrovačem pro konkrétní SIAD.

(a) **Nastavení samotného OSPFv2:**

Na MSN_B se nachází opět `router ospf 1`, u kterého je použit jako `router-id` *Loopback0*. Dále pak `router ospf 20`, který využívá *Loopback20*. Adresy přiřazených sítí jsou popsány přímo u konfigurace a odpovídají zrcadlově MSN_A.

```
router ospf 1
 router-id 192.168.0.22
 area 10 nssa no-redistribution no-summary
 network 10.200.10.48 0.0.0.3 area 10     //network connected through BDI91
 network 192.168.0.0 0.0.255.255 area 10  //network connected through BDI2071

router ospf 20
 router-id 192.168.200.22
 network 10.200.10.28 0.0.0.3 area 0    //network connected through BDI90
 network 192.168.0.22 0.0.0.0 area 0    //Loopback0
 network 192.168.200.22 0.0.0.0 area 0  //Loopbak20
```

(b) **Nastavení OSPF na BDI rozhraních:**

Zde nastává mezi MSN_A a MSN_B rozdíl v konfiguraci. Na rozhraní *BDI2071* bylo třeba nastavit rozhraní s „cost“ 300. Hodnota intervalů a ostatní konfigurace je stále stejná.

```
interface BDI2071
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 300
 ip ospf database-filter all out
```

Na závěr se musí nastavit *BDI90* a *BDI91*. Jejich konfigurace se nikterak neliší od konfigurace použité u MSN_A.

```
interface BDI90
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 100

interface BDI91
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 100
```

✎ Správnost nastavení OSPFv2 například na SIAD směrovači je možné ověřit pomocí příkazu:

show ip ospf neighbor

```
SIAD#    sh ip ospf nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.0.22      0   FULL/  -        00:00:56    192.168.20.21   BDI2071
192.168.0.11      0   FULL/  -        00:01:00    192.168.10.9    BDI1071
```

Samostatný úkol:

Tento samostatný úkol slouží pro lepší pochopení vyvažování zátěže v simulované transportní síti. Nejprve proveďte z MSN_A příkaz `traceroute` na adresu *BDI101* u SIAD směrovače. Poté proveďte stejný příkaz z MSN_B. Odůvodněte si, jak a proč MSN_B směruje provoz přes MSN_A. Pomůže Vám k tomu příkaz `show ip route <ip adresa BDI101>` na obou MSN směrovačích.

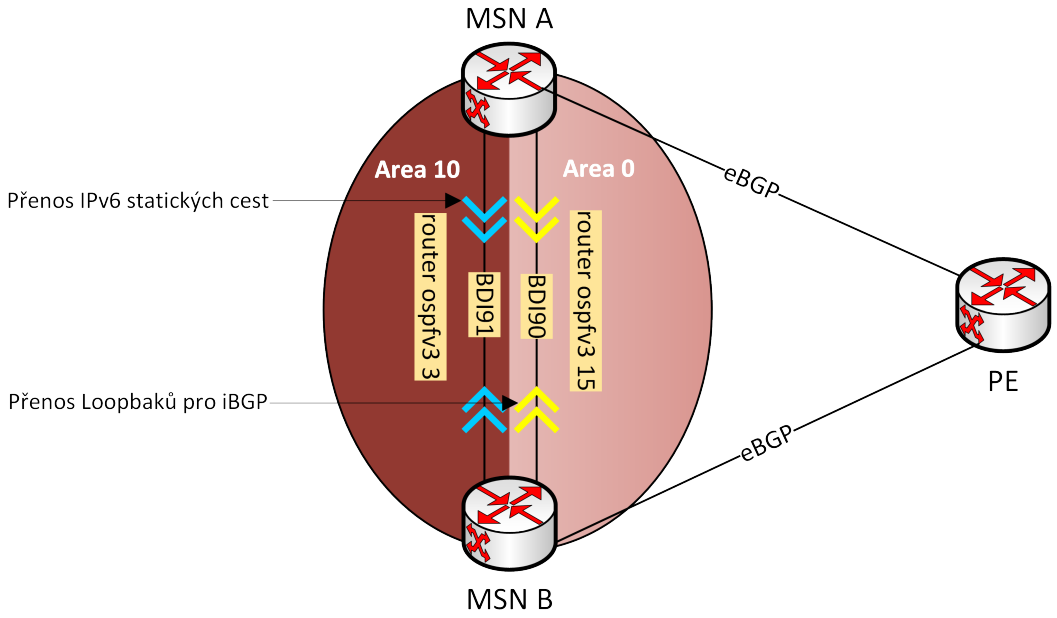
5.1.5 Krok 5. - nastavení OSPFv3

Jak je možné pozorovat na Obr. 5.4, mezi MSN párem se kromě OSPFv2 nachází i OSPFv3. Verze 2 a 3 na MSN spolu vzájemně nekomunikují. Verze 3 se nevyužívá pro „downstream“ provoz směrem k eNodeB, k tomu slouží statické cesty. Stará se ale o předávání IPv6 cest mezi MSN párem. V případě výpadku IPv6 statické primární cesty na MSN_A k SIAD směrovači bude využita záložní linka z MSN_B, která je oznamována MSN_A právě protokolem OSPFv3.

Protokol OSPFv3 je rozdělen podobně jako OSPFv2 pomocí proces ID na `router ospfv3 15` a `router ospfv3 3`.

Opět i v případě OSPFv3 je nastavení rozděleno na konfiguraci v globálním módu a na specifičtější konfiguraci přímo na rozhraní.

Konfigurace popsaná níže vychází z Obr. 5.6.



Obr. 5.6: Směrovací schéma pro OSPFv3

1. **Nastavení samotného OSPFv3:**

   Největší zodpovědnost má `router ospfv3 3`, který má na starost redistribuci IPv6 statických cest do OSPFv3. Jako `router-id` je zvolen *Loopback0*. Oblast, do které proces ID 3 spadá je 10. Oblast i proces ID jsou přiřazení v případě OSPFv3 až přímo na konkrétním portu. Pod adresní rodinou se nachází také příkaz `redistribute static metric-type 1`, který způsobí, že hodnota „cost“ se bude s průchodem přes jednotlivé linky měnit.

Platí pro MSN_A

```
router ospfv3 3
 router-id 192.168.0.11
 address-family ipv6 unicast
  redistribute static metric-type 1
```

Platí pro MSN_B

```
router ospfv3 3
 router-id 192.168.0.22
 address-family ipv6 unicast
  redistribute static metric-type 1
```

`Router ospfv3 15` má podobný význam jako `router ospf 20`; informuje ostatní připojené směrovače o dostupnosti *Loopback20* a tedy samotného MSN pro použití u protokolu iBGP, viz Krok 6. Jako `router-id` je zvolen *Loopback20*. Oblast, do které `router ospfv3 15` spadá je 0, t.j. *Backbone*. Toto přiřazení je v případě OSPFv3 až přímo na konkretním portu, viz níže.

Platí pro MSN_A

```
router ospfv3 15
 router-id 192.168.200.11
```

Platí pro MSN_B

```
router ospfv3 15
 router-id 192.168.200.22
```

2. **Nastavení OSPFv3 na BDI rozhraních:**

V případě verze 3 se specifické nastavení na rozhraních týká portů *BDI90* a *BDI91*. Aby bylo možné na portech zprovoznit IPv6 dynamický protokol, musí dojít k aktivování příkazem `ipv6 enable`. Nastavení OSPFv3 se od OSPFv2 příliš neliší. *Hello interval* má také hodnotu 20 a *dead interval* 70, je nakonfigurován i `network point-to-point`. Na *BDI90* a *BDI91* je nastavena metrika „cost“ s hodnotou 100. *BDI90* patří pod `router ospfv3 15`, který se na rozhraní přiřadí do oblasti 0. *BDI91* však spadá pod `router ospfv3 3` s oblastí 10.

Platí pro MSN_A i MSN_B

```
interface BDI90
 ipv6 enable
 ospfv3 network point-to-point
 ospfv3 15 ipv6 area 0
 ospfv3 15 ipv6 hello-interval 20
 ospfv3 15 ipv6 dead-interval 70
 ospfv3 15 ipv6 cost 100

interface BDI91
 ipv6 enable
 ospfv3 3 network point-to-point
 ospfv3 3 hello-interval 20
 ospfv3 3 dead-interval 70
 ospfv3 3 cost 100
 ospfv3 3 ipv6 area 10
```

Nesmíme také zapomenout přiřadit *Loopback20* pod proces 15, protože OSPFv3 se v tomto simulačním scénáři konfiguruje přímo na konkrétních rozhraních.

Platí pro MSN_A i MSN_B

```
interface Loopback20
ospfv3 15 ipv6 area 0
```

✎ Konfiguraci je možné ověřit příkazem:

show ospf neighbor

```
MSN_A#sh ospf nei

          OSPFv3 3 address-family ipv6 (router-id 192.168.0.11)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.168.0.22      0   FULL/  -        00:00:54    21              BDI91

          OSPFv3 15 address-family ipv6 (router-id 192.168.200.11)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.168.200.22    0   FULL/  -        00:00:52    20              BDI90
```

Samostatný úkol:

Pro lepší pochopení vyvažování zátěže v simulované transportní síti proveďte u OSPFv3 stejný úkol jako u OSPFv2. Z MSN_A se nejprve podívejte na výstup příkazu `traceroute` na adresu *BDI212* u SIAD směrovače. Poté proveďte ten stejný příkaz z MSN_B. Odůvodněte si, jak a proč MSN_B směruje provoz přes MSN_A. Pomůže Vám k tomu příkaz `show ipv6 route <ipv6 adresa BDI212>` na obou MSN směrovačích.

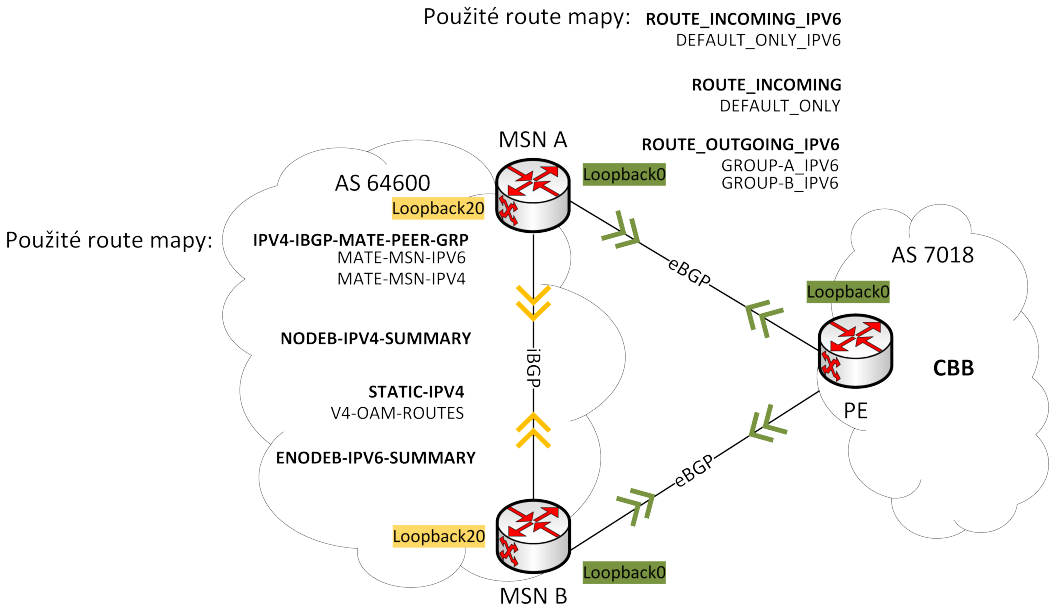
5.1.6 Krok 6. - nastavení BGP protokolu

I v mobilní transportní síti hraje protokol BGP svoji důležitou úlohu. Pomocí něj se přenáší veškeré IPv4 a IPv6 směrovací informace pro umožnění provozu dále z MSN směrem do CBB přes PE. Nachází se v podobě iBGP mezi MSN párem a pokračuje dále jako eBGP směrem k PE. iBGP protokol je implementován ze dvou důvodů: jednak protože OSPF databáze by byla příliš obsáhlá a protože BGP má více užitečných funkcí jako je například rozdělení provozu do skupin použitím route map nebo community listů.

OSPF protokol je však využíván protokolem iBGP k vytvoření tzv. full-mesh BGP, díky loopbackům, které jsou přenášeny pomocí OSPF, konkrétně *BDI90*. Full-mesh slouží jako prevence proti zasmyčkování.

V simulované mobilní transportní síti se pro nastavení směrování za pomoci BGP používá atributu community list. Tento atribut patří do skupiny tranzitních atributů a je tak přenášen za hranice AS. Pro přidělení *community* cestám se využívá prefix listů a route map. Jiný BGP atribut se v síti nenastavuje a jsou ponechána pouze výchozí nastavení např. u lokální preference nebo atributu *weight*.

Graficky je proces iBGP a eBGP znázorněn na obrázku Obr. 5.7. Simulovaná transportní síť patří pod AS64600 a je připojena do Core Backbone s hodnotou AS7018. Celé BGP v simulované transportní síti je postaveno na prefix listech, access listech, route mapách viz Obr. 5.7 a atributu community viz příloha Tab. A.2.



Obr. 5.7: Proces iBGP a eBGP spolu s route mapami a prefix listy

1. **Zprovoznění samotného BGP**

- První etapou při zprovoznění BGP na MSN směrovačích je jeho vytvoření příkazem `router bgp 64600`.
- Přiřadíme mu router-ID, které je jako u OSPF adresa *Loopback0*.
- Pro sledování změn v BGP se použije `bgp log-neighbor-changes`. Je zde použit i příkaz `no bgp default ipv4-unicast`, kterým se zakáže automatické přidávání IPv4 sousedů do adresní rodiny IPv4 a adresy budou muset být přidávány ručně. Důvodem tohoto řešení je větší bezpečnost sítě. Pro IPv6 je to již ve výchozím nastavení.

Platí pro MSN_A

```
router bgp 64600
 bgp router-id 192.168.0.11
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
```

Platí pro MSN_B

```
router bgp 64600
 bgp router-id 192.168.0.22
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
```

- Na směrovači PE zprovozníme podobným způsobem BGP, ale pro autonomní systém 7018.

Platí pro PE

```
router bgp 7018
 bgp router-id 192.168.0.77
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
```

2. **Zprovoznění iBGP mezi MSN párem**

- Následuje zprovoznění směrování mezi MSN párem pomocí BGP, tedy konkrétně iBGP. Konfigurace je vázána na route mapy `IPV4-IBGP-MATE-PEER-GRP` pro IPv4 a `IPV6-IBGP-MATE-PEER-GRP` pro IPv6.
- Tyto route mapy jsou použity při nastavení sousedství přiřazením jednotlivých obecných pravidel v globálním BGP módu AS64600 a shodují se pro MSN pár. Nejprve je nutné vytvořit `peer-group` pro zjednodušení konfigurace a zlepšení výkonu.
- K této peer-group určíme sousední AS `remote-as 64600`, tedy že patří do stejného AS jako zdroj, a spustí se tak iBGP. Příkazem `update-source Loopback20` se do směrovacích záznamů přidá *Loopback20*, nikoliv port, ze kterého byly pakety zaslány. Jelikož je Loopback vždy ve stavu UP - pokud není stanoveno administrátorem jinak - není BGP směrovací tabulka tolik zatěžována.

- Jako poslední jsou k route mapám přidány časovače `timers 30 90`, kde 30 s představuje čas pro zasílání keepalive zpráv a 90 s představuje „hold time“.

Platí pro MSN_A a MSN_B

```
router bgp 64600
 neighbor IPV4-IBGP-MATE-PEER-GRP peer-group
 neighbor IPV4-IBGP-MATE-PEER-GRP remote-as 64600
 neighbor IPV4-IBGP-MATE-PEER-GRP update-source Loopback20
 neighbor IPV4-IBGP-MATE-PEER-GRP timers 30 90

 neighbor IPV6-IBGP-MATE-PEER-GRP peer-group
 neighbor IPV6-IBGP-MATE-PEER-GRP remote-as 64600
 neighbor IPV6-IBGP-MATE-PEER-GRP update-source Loopback20
 neighbor IPV6-IBGP-MATE-PEER-GRP timers 30 90
```

- Dalším krokem pro zprovoznění iBGP je nastavení IP adres sousedního směrovače a přiřazení `peer-group` s názvem vytvořené route mapy. Jelikož příkaz `update-source Loopback20` je nastaven i na sousedním směrovači, jako IP adresy jsou použity adresy *Loopback20*.

Platí pro MSN_A

```
 neighbor 2001:506:4600:8E9::2 peer-group IPV6-IBGP-MATE-PEER-GRP //Loopback20 on MSN B
 neighbor 192.168.200.22 peer-group IPV4-IBGP-MATE-PEER-GRP //Loopback20 on MSN B
```

Platí pro MSN_B

```
 neighbor 2001:506:4600:8E1::2 peer-group IPV6-IBGP-MATE-PEER-GRP //Loopback20 on MSN A
 neighbor 192.168.200.11 peer-group IPV4-IBGP-MATE-PEER-GRP //Loopback20 on MSN A
```

- Další konfigurace se již musí přiřadit pod adresní rodiny IPv4 a IPv6. Přidání souseda pod konkrétní adresní rodinu znamená, že chceme vyměnit dané cesty s přiřazeným sousedem. V případě, že souseda pod konkrétní adresní rodinu nepřidáme znamená to, že žádné informace o cestách předávat nechceme. Souvisí to i s příkazem `no bgp default ipv4-unicast`, který je u BGP nastaven v bodě 1. Pro IPv6 je toto nastavení ve výchozím stavu.
  Pod adresní rodinu je přiřazen soused s adresami *Loopbacku20* u sousedního směrovače s příkazem `activate`. Tento příkaz povolí vyměňování informací s tímto sousedem. Bez tohoto příkazu by ke komunikaci nedošlo.

Platí pro MSN_A

```
router bgp 64600
 address-family ipv4
    neighbor 192.168.200.22 activate
 address-family ipv6
    neighbor 2001:506:4600:8E9::2 activate
```

Platí pro MSN_B

```
router bgp 64600
 address-family ipv4
  neighbor 192.168.200.11 activate
 address-family ipv6
    neighbor 2001:506:4600:8E1::2 activate
```

- Pod adresními rodinami IPv4 a IPv6 je poté již nastavení stejné. Konfigurace je vázaná na peer-group pro IPv4 a IPv6.
- Prvním nastavením je `send-community`, kterým se povolí odesílání community k sousedovi.
- Následuje `next-hop-self`, kterým je řešen problém, že iBGP nemění next hop IP adresu v BGP tabulce a ponechá adresu místa, odkud paket přišel. O této adrese nemá však směrovač připojený přes iBGP informaci a proto ji nepřidá do svojí směrovací tabulky. Příkaz `next-hop-self` změní v BGP tabulce jako zdrojovou adresu svoji vlastní a tím je směrovač připojený přes iBGP schopen vidět směrovač připojený přes eBGP a přidat jej do svojí směrovací tabulky.
- Posledním nastavením je přiřazení souseda přes peer-group a přiřazení `route map MATE-MSN/-IPV6 out`. Skrze ni jsou nastavena pravidla pro odchozí provoz ze směrovače. Tato route mapa bude dovysvětlena v bodě 4.

Platí pro MSN_A a MSN_B

```
router bgp 64600
 address-family ipv4
  neighbor IPV4-IBGP-MATE-PEER-GRP send-community
  neighbor IPV4-IBGP-MATE-PEER-GRP next-hop-self
  neighbor IPV4-IBGP-MATE-PEER-GRP route-map MATE-MSN out

 address-family ipv6
  neighbor IPV6-IBGP-MATE-PEER-GRP send-community
  neighbor IPV6-IBGP-MATE-PEER-GRP next-hop-self
  neighbor IPV6-IBGP-MATE-PEER-GRP route-map MATE-MSN-IPV6 out
```

3. **Nastavení Community-list naming**

   Nyní se dostáváme k nastavení atributu community-list. Na Cisco zařízeních je formát community jedno 32-bitové číslo, díky příkazu `ip bgp-community new-format` je toto číslo převedeno do člověku srozumitelného formátu AS:ČISLO.
   Poté se nastaví jednotlivým community-list jejich jmenovitý standard příkazem `ip community-list standard <jméno> permit <AS:ČISLO>`. Přiřazené jméno standardu ke community řetězci je zobrazeno v Tab. A.2 viz příloha. Tyto stadardy spolu s community řetězci pokrývají veškerý provoz přes BGP na MSN páru a budou postupně vysvětleny a přiřazeny v následujících krocích.

Platí pro MSN_A a MSN_B

```
ip bgp-community new-format
ip community-list standard SHORTHAUL-ALL permit 64600:1000
ip community-list standard OAM-IPV4-ALL permit 64600:3000
ip community-list standard PE-DEFAULT-V4 permit 64600:7000
ip community-list standard PE-DEFAULT-V6 permit 64600:7100
ip community-list standard CONNECTED-IPV4-ALL permit 64600:2000
```

4. **Route mapa MATE-MSN/-IPV6**

   Route mapa `MATE-MSN/-IPV6 out` představuje poslední filtr před odesláním paketů k sousednímu směrovači MSN. Jejím úkolem je hledání shod pomocí community standardů - pokud nalezne shodu, paket odešle. Příkaz `permit 10` je ponechán ve výchozím stavu.

Platí pro MSN_A a MSN_B

```
route-map MATE-MSN permit 10
 description permit all including specifics to mate
 match community SHORTHAUL-ALL CONNECTED-IPV4-ALL OAM-IPV4-ALL PE-DEFAULT-V4

route-map MATE-MSN-IPV6 permit 10
 description permit all including specifics to mate
 match community SHORTHAUL-ALL PE-DEFAULT-V6
```

   Na tuto route mapu naváže v následujících bodech několik dalších route map.

5. **Oznamování adres NodeB - Bearer**

- Jako první nastavíme statickou cestu pro IP adresu sítě, kam spadá adresa sítě Bearer u NodeB s rozhraním `Null0` pro zamezení zacyklení a zbytečnému přetěžování směrovačů.

Platí pro MSN_A a MSN_B

```
ip route 172.16.0.0  255.255.224.0 Null0 name NodeBBER
```

- K oznámení IP adresy sítě Bearer pro NodeB pomocí BGP je potřeba využít příkazu `network <adresa> mask <maska> route-map <název>` navíc ještě s podmínkou stanovenou route mapou. Příkaz je přiřazen pod adresní rodinu, aby mohl být oznamován sousednímu směrovači.

Platí pro MSN_A a MSN_B

```
router bgp 64600
 address-family ipv4
   network 172.16.0.0 mask 255.255.224.0 route-map NODEB-IPV4-SUMMARY
```

- Pro nově vytvořenou `route map NODEB-IPV4-SUMMARY` stanovíme popis a pravidla. Pokud síť spadá do této route mapy, je jí přidělen community řetězec 64600:1000. Příkaz `permit 10` je ponechán ve výchozím stavu.

Platí pro MSN_A a MSN_B

```
route-map NODEB-IPV4-SUMMARY permit 10
 description Summary NODEB address space
 set community 64600:1000
```

Díky přiřazené community 64600:1000 spadá takto označená síť pod community standard *SHORTHAUL-ALL* a projde route mapou `MATE-MSN` na sousední MSN pomocí iBGP.

6. **Oznamování adres NodeB - OAM**

- Pro oznamování OAM adresy u NodeB se používá jiných pravidel než u Bearer. Aby BGP bylo schopno oznamovat síť, musí najít shodu ve svojí směrovací tabulce s konkrétní adresou a maskou. OAM adresa na NodeB je známa pomocí OPSFv2 proces 1. Proto musíme použít příkaz `redistribute ospf 1`, který redistribuuje tento OSPF proces do BGP, je zde však podmínka v podobě route mapy `OAM-IPV4`.

Platí pro MSN_A a MSN_B

```
router bgp 64600
 address-family ipv4
   redistribute ospf 1 route-map OAM-IPV4
```

- Route mapa `OAM-IPV4` hledá shodu v access listu `V4-OAM-ROUTES` a přiřadí community 64600:3000.

Platí pro MSN_A a MSN_B

```
route-map OAM-IPV4 permit 10
 description Redistributed OAM IPv4 routes
 match ip address V4-OAM-ROUTES
 set community 64600:3000
```

- Nyní je zapotřebí nastavit access list `V4-OAM-ROUTES`. Nastavíme, které sítě má povolit a pustit je dále v cestě. V daném případě do access listu musí patří OAM adresa sítě pro NodeB.

Platí pro MSN_A a MSN_B

```
ip access-list standard V4-OAM-ROUTES
permit 172.17.8.0 0.0.7.255 //OAM on NodeB 172.17.13.32/29
```

Díky výše uvedenému nastavení bude OAM adresám přiřazena community 64600:3000 a tedy standard *OAM-IPV4-ALL*, kterého je zapotřebí k tomu, aby došlo k průchodu skrze `MATE-MSN`.

7. **Oznamování adres eNodeB - Bearer**

   Na eNodeB již pracujeme pouze s protokolem IPv6.

- Podobně jako tomu bylo u NodeB, i zde je potřeba vytvořit statickou cestu s portem *Null0*. V tomto případě se jedná o dva adresní rozsahy, které vysvětlíme v bodě 10. Kroku 6. Souvisí to s ručním nastavením vyvažování zátěže mezi MSN párem pomocí prefix listů.

  Platí pro MSN_A a MSN_B

```
ipv6 route 2001:506:4247::/50 Null0 name ENodeBBER
ipv6 route 2001:506:4247:4000::/50 Null0 name ENodeBBER
```

- V dalším bodě se adresy přidají pod protokol BGP a adresní rodinu IPv6 příkazem `network`, opět s pravidlem určeným tentokrát route mapou `ENODEB-IPV6-SUMMARY`

  Platí pro MSN_A a MSN_B

```
router bgp 64600
 address-family ipv6
  network 2001:506:4247::/50 route-map ENODEB-IPV6-SUMMARY
  network 2001:506:4247:4000::/50 route-map ENODEB-IPV6-SUMMARY
```

- Route mapa `ENODEB-IPV6-SUMMARY` přidá paketům community 64600:1000, podle community naming standard *SHORTHAUL-ALL*.

  Platí pro MSN_A a MSN_B

```
route-map ENODEB-IPV6-SUMMARY permit 10
 description Summary ENODEB IPV6 static /50
 set community 64600:1000
```

  Pakety patřící do *SHORTHAUL-ALL* pak projdou route mapou `MATE-MSN` a mezi MSN párem dojde k přeposílání informací přes iBGP.

8. **Oznamování adres eNodeB - OAM**

- I v případě OAM se vytvoří „Black Hole routing“ cesta přes rozhraní *Null0*. Také zde pracujeme se dvěma adresními rozsahy. Později se však na tyto adresy žádné prefix listy nevztahují.

  Platí pro MSN_A a MSN_B

```
ipv6 route 2001:506:4447::/50 Null0 name ENodeBOAM
ipv6 route 2001:506:4447:4000::/50 Null0 name ENodeBOAM
```

- Přidání sítě do BGP je totožné jako v případě přidání sítě Bearer.

```
router bgp 64600
 address-family ipv6
  network 2001:506:4447::/50 route-map ENODEB-IPV6-SUMMARY
  network 2001:506:4447:4000::/50 route-map ENODEB-IPV6-SUMMARY
```

- Route mapa `ENODEB-IPV6-SUMMARY` je již nastavená z části pro Bearer. Pakety spadají do standardu *SHORTHAUL-ALL* a prochází route mapou `MATE-MSN`.

9. **Oznamování připojených adres SNMP serveru**

   K MSN_A je připojen virtuální PC se SNMP manažerem. Aby mohla i další zařízení v transportní síti komunikovat s tímto manažerem je potřeba zahrnout jej do směrovacích tabulek.

Platí pro MSN_A

```
router bgp 64600
 address-family ipv4
   redistribute static route-map CONNECTED-IPV4
```

   K tomuto účelu je vytvořena route mapa `CONNECTED-IPV4`, kde dochází k přiřazení community 64600:2000 a tedy přiřazení do standardu *CONNECTED-IPV4-ALL*, kterého je zapotřebí k tomu, aby došlo k průchodu skrze `MATE-MSN`.

Platí pro MSN_A

```
route-map CONNECTED-IPV4 permit 10
 description redistributed ipv4 connected interfaces
 set community 64600:2000
```

10. **Zprovoznění eBGP na MSN páru**

    Posledním krokem při nastavení BGP je sestavení komunikace se sousední AS7018, tedy zprovoznění eBGP.

- Do AS64600 se musí nejprve v globálním módu vložit informace o sousední AS7018, jak pro IPv4 tak pro IPv6.

Platí pro MSN_A

```
router bgp 64600
 neighbor 77.66.55.45 remote-as 7018           //IPv4 address on PE Port-channel11
 neighbor 77.66.55.45 timers 30 90

 neighbor 2001:506:4600:8228::1 remote-as 7018 //IPv6 address on PE Port-channel11
 neighbor 2001:506:4600:8228::1 timers 30 90
```

Platí pro MSN_B

```
router bgp 64600
 neighbor 77.66.55.85 remote-as 7018           //IPv4 address on PE Port-channel12
 neighbor 77.66.55.85 timers 30 90

 neighbor 2001:506:4600:822A::1 remote-as 7018 //IPv6 address on PE Port-channel12
 neighbor 2001:506:4600:822A::1 timers 30 90
```

- Pod adresní rodiny přidáme sítě tak, aby si nimi mohl směrovač vyměňovat informace. Nejprve je sousedství aktivováno příkazem `activate`. Poté je povoleno posílat atribut community na tohoto souseda příkazem `send-community`. Navíc zde platí podmínka pro přicházející provoz v podobě route map `ROUTE_INCOMING` a `ROUTE_INCOMING_IPV6`. Pro IPv6 platí dokonce i pro odchozí provoz z MSN v podobě `ROUTE_OUTGOING_IPV6`.

Platí pro MSN_A

```
router bgp 64600
 address-family ipv4
  neighbor 77.66.55.45 activate
  neighbor 77.66.55.45 send-community
  neighbor 77.66.55.45 route-map ROUTE_INCOMING in

 address-family ipv6
  neighbor 2001:506:4600:8228::1 activate
  neighbor 2001:506:4600:8228::1 send-community
  neighbor 2001:506:4600:8228::1 route-map ROUTE_INCOMING_IPV6 in
  neighbor 2001:506:4600:8228::1 route-map ROUTE_OUTGOING_IPV6 out
```

Platí pro MSN_B

```
router bgp 64600
 address-family ipv4
  neighbor 77.66.55.85 activate
  neighbor 77.66.55.85 send-community
  neighbor 77.66.55.85 route-map ROUTE_INCOMING in

 address-family ipv6
  neighbor 2001:506:4600:822A::1 activate
  neighbor 2001:506:4600:822A::1 send-community
  neighbor 2001:506:4600:822A::1 route-map ROUTE_INCOMING_IPV6 in
  neighbor 2001:506:4600:822A::1 route-map ROUTE_OUTGOING_IPV6 out
```

- Nejprve se zaměříme na route mapy filtrující přichozí provoz, tedy směr IN, `ROUTE_INCOMING` a `ROUTE_INCOMING_IPV6`. Hledají shodu v prefix listu `DEFAULT_ONLY` a `DEFAULT_ONLY_IPV6`. Následně nastavíme hodnotu community na 64600:7000 a 64600:7100.

Platí pro MSN_A a MSN_B

```
route-map ROUTE_INCOMING permit 10
 match ip address prefix-list DEFAULT_ONLY
 set community 64600:7000

route-map ROUTE_INCOMING_IPV6 permit 10
 match ipv6 address prefix-list DEFAULT_ONLY_IPV6
 set community 64600:7100
```

  Přiřazená community odpovídá standardu *PEDEFAULT-V4* pro 64600:7000 a *PE-DEFAULT-V6* pro 64600:7100. Tento community standard prochází skrze `MATE-MSN` a předává informace mezi MSN párem.
- Nyní je zapotřebí specifikovat použité prefix listy. Prefix listy mají stejnou podobu na MSN_A i MSN_B. Tvoří se příkazem `ip prefix-list <jméno> seq <číslo> <permit/deny> x.x.x.x/x`, kde `seq <číslo>` určuje pořadí prefix listů v seznamu. Směrovač porovná adresy s adresami uvedenými v prefix listech. Porovnává od prvního prefix listu v seznamu, kde jsou záznamy řazeny od nejnižších hodnot. Výchozí hodnota `seq <číslo>` je 10.
  Prefix listy `DEFAULT_ONLY/_IPV6` akceptují paket pouze pokud obsahuje jako zdroj adresu výchozí cesty. Stejné nastavení je pro IPv4 i IPv6.

Platí pro MSN_A i MSN_B

```
ip prefix-list DEFAULT_ONLY seq 10 permit 0.0.0.0/0
ipv6 prefix-list DEFAULT_ONLY_IPV6 seq 10 permit ::/0
```

- Pro odchozí provoz, tedy OUT pro IPv4 žádné stanovené podmínky nejsou. Naopak pro IPv6 je zde definovaná route mapa `ROUTE_OUTGOING_IPV6`. Je spjata s ručním vyvažováním připojených eNodeB, kde je IPv6 adresní prostor pro daný MSN pár rozdělen na dva rozsahy. Pro jeden je primární MSN_A, pro druhý MSN_B. Tato konfigurace souvisí s prefix listy.

Platí pro MSN_A

```
 route-map ROUTE_OUTGOING_IPV6 permit 10
 match ipv6 address prefix-list GROUP-A_IPV6
 set community 13979:2784

route-map ROUTE_OUTGOING_IPV6 permit 20
 match ipv6 address prefix-list GROUP-B_IPV6
```

Platí pro MSN_B

```
 route-map ROUTE_OUTGOING_IPV6 permit 10
 match ipv6 address prefix-list GROUP-A_IPV6

route-map ROUTE_OUTGOING_IPV6 permit 20
 match ipv6 address prefix-list GROUP-B_IPV6
 set community 13979:2784
```

Route mapa `ROUTE_OUTGOING_IPV6` je rozdělena na `GROUP-A_IPV6` a `GROUP-B_IPV6` z důvodu odlišného přiřazování community u MSN_A a MSN_B. U `GROUP-A_IPV6` na MSN_A je přiřazena community 13979:2784. Naopak u `GROUP-B_IPV6` je community 13979:2784 přiřazena na MSN_B. Tento postup je zvolen kvůli ručně řešenému vyvažování zátěže mezi směrovači. Nastavení prefix listů je u MSN páru stejné.

Platí pro MSN_A i MSN_B

```
ipv6 prefix-list GROUP-A_IPV6 seq 10 permit 2001:506:4247::/50  //1.address range Bearer
ipv6 prefix-list GROUP-A_IPV6 seq 20 permit 2001:506:4447::/50  //1.address range OAM

ipv6 prefix-list GROUP-B_IPV6 seq 10 permit 2001:506:4247:4000::/50  //2.address range
ipv6 prefix-list GROUP-B_IPV6 seq 20 permit 2001:506:4447:4000::/50  //2.address range OAM
```

11. Zprovoznění eBGP na PE

Na PE směrovači je konfigurace zjednodušená a nastavená podobným způsobem jako na MSN páru.
Jako první je do globálního módu nastaven soused směrem k MSN páru, společně s časovači.

Platí pro PE

```
router bgp 7018
neighbor 2001:506:4600:8228::2 remote-as 64600
 neighbor 2001:506:4600:8228::2 timers 30 90
 neighbor 2001:506:4600:822A::2 remote-as 64600
 neighbor 2001:506:4600:822A::2 timers 30 90
 neighbor 77.66.55.46 remote-as 64600
 neighbor 77.66.55.46 timers 30 90
 neighbor 77.66.55.86 remote-as 64600
 neighbor 77.66.55.86 timers 30 90
```

Následuje nastavení pod adresní rodiny, kde je nejprve přidána síť, poté je aktivován soused a nakonec je povoleno zasílání community.

Platí pro PE

```
router bgp 7018
 address-family ipv4
  network 77.66.55.44 mask 255.255.255.252
  network 77.66.55.84 mask 255.255.255.252
  neighbor 77.66.55.46 activate
  neighbor 77.66.55.46 send-community
  neighbor 77.66.55.86 activate
  neighbor 77.66.55.86 send-community
 exit-address-family

 address-family ipv6
  network 2001:506:4600:8228::/64
  network 2001:506:4600:822A::/64
  neighbor 2001:506:4600:8228::2 activate
  neighbor 2001:506:4600:8228::2 send-community
  neighbor 2001:506:4600:822A::2 activate
  neighbor 2001:506:4600:822A::2 send-community
 exit-address-family
```

✎ Správnost nastavení lze ověřit tím, že se přesvědčíme, zda je vytvořeno sousedství jak s druhým MSN, tak i s PE směrovačem:

show bgp summary

```
MSN_A#sh bgp sum
BGP router identifier 192.168.0.11, local AS number 64600
BGP table version is 40, main routing table version 40
11 network entries using 2728 bytes of memory
13 path entries using 1560 bytes of memory
2/2 BGP path/bestpath attribute entries using 512 bytes of memory
3 BGP community entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4872 total bytes of memory
BGP activity 33/13 prefixes, 62/37 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
77.66.55.45     4         7018     128     126       40    0    0 00:53:58        0
192.168.200.22  4        64600      10      12       40    0    0 00:03:31        6
```

✎ Příkazem `show ipv6 route` lze například zkontrolovat, že PE směrovač ví o NodeB adresách a že jsou pomocí BGP součástí směrovací tabulky.

show ip route

```
PE#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B        10.10.10.0/24 [20/0] via 77.66.55.46, 00:07:13
B        10.200.10.28/30 [20/0] via 77.66.55.46, 00:07:13
B        10.200.10.48/30 [20/0] via 77.66.55.46, 00:07:13
      77.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        77.66.55.44/30 is directly connected, Port-channel11
L        77.66.55.45/32 is directly connected, Port-channel11
C        77.66.55.84/30 is directly connected, Port-channel12
L        77.66.55.85/32 is directly connected, Port-channel12
      172.16.0.0/19 is subnetted, 1 subnets
B        172.16.0.0 [20/0] via 77.66.55.46, 00:07:13
      172.17.0.0/29 is subnetted, 1 subnets
B        172.17.13.32 [20/0] via 77.66.55.86, 00:00:58
      192.168.0.0/32 is subnetted, 3 subnets
B        192.168.0.11 [20/0] via 77.66.55.46, 00:07:13
B        192.168.0.22 [20/0] via 77.66.55.46, 00:07:13
C        192.168.0.77 is directly connected, Loopback0
      192.168.10.0/30 is subnetted, 1 subnets
B        192.168.10.8 [20/0] via 77.66.55.46, 00:07:13
      192.168.20.0/30 is subnetted, 1 subnets
B        192.168.20.20 [20/0] via 77.66.55.46, 00:07:13
      192.168.200.0/32 is subnetted, 2 subnets
B        192.168.200.11 [20/0] via 77.66.55.46, 00:07:13
B        192.168.200.22 [20/0] via 77.66.55.46, 00:07:13
```

Samostatný úkol:

Samostatný úkol pro BGP spočívá v zachytávání paketů pomocí programu Wireshark. Zachyťte BGP pakety na lince mezi rozhraními *g7* na MSN_A a *g3* na PE směrovači. Viditelné by měly být pouze KEEPALIVE zprávy. Proto proveďte příkaz `clear ip bgp summary *` na MSN_A a sledujte počet vyměněných zpráv. Zaměřte se na „UPDATE“ zprávy a především na záložku *Path Attribute - COMMUNITIES*, kde můžete sledovat přenos nastavených atributů v Kroku 6.

5.1.7 Krok 7. - nastavení BFD protokolu

Jak již bylo uvedeno v Kap. 2.4, slouží protokol BFD k rychlému rozpoznání výpadku na cestě. Jeho výhodou je, že jej dokáže rozeznat na jakémkoliv typu spojení nezávisle na médiu. Vždy se váže na směrovací protokol. V simulované transportní síti je navázán na statické cesty, protokol OSPFv2, port-channel mezi MSN párem a BGP.

Protokol BFD je vždy navázán pouze na primární linku. Je to logické, neboť při výpadku sekundární linky (záložní) nelze provoz nikam přesměrovat.

Konfigurace je řešena odlišně na SIAD směrovači a na MSN páru. Hodnoty časovače však mají stejné. Časovač je určen příkazem `bfd interval <milliseconds> min_rx <milliseconds> multiplier <interval-multiplier>`. Hodnota `interval` určuje hodnotu v milisekundách od 50 do 9999, kdy bude kontrolní paket zasílán BFD peer směrovači. Hodnota `min_rx` specifikuje dobu, kdy je očekáván příchod BFD kontrolního paketu od BFD peer směrovače, časový rozptyl je také od 50 do 9999. Poslední údaj `multiplier` představuje počet paketů, které musí chybět, tedy nebýt doručeny od BFD peer směrovače, aby došlo k oznámení o jeho nedostupnosti. Validní počet je od 3 do 50. V této simulované transportní síti byly zvoleny hodnoty 500 milisekund a počet chybějících kontrolních paketů 3.

Na SIAD směrovači je BFD protokol přiřazen k OSPFv2 a statickým cestám. U MSN páru je jeho použití více rozšířenější a využívá se tak „template“. Vytvoří se v globálním módu příkazem `bfd-template single-hop <název>`, poté se přidají hodnoty intervalů podobně jako tomu bylo v předchozím odstavci `interval min-tx <milliseconds> min-rx <milliseconds> multiplier <interval-multiplier>`. Echo mód je při použití „template“ ve výchozím stavu vypnutý.

Platí pro MSN_A MSN_B a PE

```
bfd-template single-hop msn-bfd-template
 interval min-tx 500 min-rx 500 multiplier 3
```

1. **Implementace BFD u statických cest**

   Protokol BFD je navázán pouze na primární linku, tedy na rozhraní *BDI1071*. Aby mohlo být BFD spojení skrze statické cesty zprovozněno, je nejprve nutné nastavit BFD na samotném BDI rozhraní. Intervaly jsou nastaveny na hodnoty 500 milisekund a BFD peer musí neobdržet 3 kontrolní pakety, aby byl prohlášen za nedostupný.

   Platí pro SIAD

   ```
   interface BDI1071
    bfd interval 500 min_rx 500 multiplier 3
    no bfd echo
   ```

Je zde použit příkaz `no bfd echo`, protože ze strany MSN směrovačů, kde se využívá „template“, je echo mód automaticky vypnutý. Na MSN_A je „template“ vložen také pod rozhraní *BDI1071*.

Platí pro MSN_A

```
interface BDI1071
 bfd template msn-bfd-template
```

Nyní je zapotřebí zprovoznit samotné BFD u statických cest. Syntaxe má následující podobu `ip/ipv6 route static bfd <rozhraní> <IP adresa>`. Tím se naváže protokol BFD na statické cesty. Na SIAD směrovači vložíme tento příkaz pro IPv4 a IPv6 statické cesty vedoucích k MSN_A. V opačném směru, tedy z MSN_A, příkaz zadáme pouze u IPv6 statických cest vedoucích k SIAD směrovači.

Platí pro SIAD

```
conf t
ip route static bfd BDI1071 192.168.10.9
ipv6 route static bfd BDI1071 2001:506:4047:C5::1
```

Platí pro MSN_A

```
conf t
ipv6 route static bfd BDI1071 2001:506:4047:C5::2
```

2. **Implementace BFD u protokolu OSPFv2**

Existují dva způsoby jak zprovoznit protokol BFD s OSPF. Prvním je hromadné zprovoznění pro všechny porty patřící do OSPF, kde se do konfigurace OSPF přidá příkaz `bfd all-interfaces`. Druhým způsobem je povolení BFD přímo na konkrétním portu patřícím do OSPF pomocí `ip ospf bfd`. V simulaci transportní sítě se využívá kombinace těchto možností.
Podobně jako tomu bylo u statických cest, musí být pro zprovoznění BFD skrze OSPF nakonfigurovány hodnoty intervalu na *BDI1071*. Ty jsou již nastaveny z předešlé konfigurace.
U OSPF se navíc využívá protokolu BFD ve striktním módu. Striktní mód zajistí, že OSPF soused bude ve stavu DOWN do té doby, než bude BFD protokol UP, a to tak, že protokol bude přímo závislý na stavu BFD. Striktní mód je přímo konfigurován na konkrétní port příkazem `ip ospf bfd strict-mode`.

Platí pro SIAD

```
router ospf 10
 bfd all-interfaces

interface BDI1071
 ip ospf bfd strict-mode
```

Platí pro MSN_A

```
router ospf 1
 bfd all-interfaces

interface BDI1071
 ip ospf bfd strict-mode
```

✎ Ověření lze provést příkazem:

show bfd neighbor

```
SIAD#sh bfd nei

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
192.168.10.9                            1/2          Up        Up        BD1071

IPv6 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
2001:506:4047:C5::1                     3/1          Up        Up        BD1071
```

3. **Implementace BFD u protokolu BGP**

BFD je také implementováno mezi MSN a PE, kde je navázáno na protokol BGP. Mezi MSN_A a PE je využito `Port-channel11`, mezi MSN_B a PE zase `Port-channel12` a platí zde stejná logika jako u statických cest a OSPF, a sice že pro funkčnost BFD je nutné nastavit hodnoty časovačů BFD na konkrétní rozhraní. Na port-channel je přiřazen „template“.

Platí pro MSN_A

```
interface Port-channel11
 bfd template msn-bfd-template
```

Platí pro MSN_B

```
interface Port-channel12
 bfd template msn-bfd-template
```

Platí pro PE

```
interface Port-channel11
 bfd template msn-bfd-template

interface Port-channel12
 bfd template msn-bfd-template
```

Následuje samotné nastavení BFD v konfiguraci BGP příkazem `fall-over bfd`. Tím se mezi sousedními směrovači vytvoří BFD peer spojení.

Platí pro MSN_A

```
router bgp 64600
 neighbor 77.66.55.45 fall-over bfd
 neighbor 2001:506:4600:8228::1 fall-over bfd
```

Platí pro MSN_B

```
router bgp 64600
 neighbor 77.66.55.85 fall-over bfd
 neighbor 2001:506:4600:822A::1 fall-over bfd
```

Platí pro PE

```
router bgp 7018
 neighbor 77.66.55.46 fall-over bfd
 neighbor 2001:506:4600:8228::2 fall-over bfd

 neighbor 77.66.55.86 fall-over bfd
 neighbor 2001:506:4600:822A::2 fall-over bfd
```

✎ Kontrola vytvoření BFD peer spojení by měla vypadat následovně a provedeme ji pomocí příkazu:

show bfd neighbors

```
PE#sh bfd neighbors

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
77.66.55.46                             1/3          Up        Up        Po11
77.66.55.86                             3/1          Up        Up        Po12

IPv6 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
2001:506:4600:8228::2                   2/4          Up        Up        Po11
2001:506:4600:822A::2                   4/2          Up        Up        Po12
```

```
MSN_A#sh bfd neighbors

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
77.66.55.45                             3/1          Up        Up        Po11
192.168.10.10                           1/1          Up        Up        BD1071

IPv6 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
2001:506:4047:C5::2                     2/2          Up        Up        BD1071
2001:506:4600:8228::1                   4/2          Up        Up        Po11
```

Samostatný úkol:

Tento samostatný úkol bude sloužit k ověření striktní závislosti protokolu OSPFv2 na BFD, která byla nastavena v Kroku 7. Pokud samotné BFD přestane fungovat, přeruší se i OSPFv2 spojení. Na MSN_A směrovači na rozhraní *BDI1071* odstraňte BFD nastavení příkazem `no bfd template msn-bfd-template`. Sledujte reakci, poté „template“ vložte zpět na rozhraní a opět sledujte změnu. Striktní nastavení závislosti na BFD protokolu není naopak použito u BGP ani statických cest, protože tam nejsou tak vysoké nároky na přepojení na záložní linku. Je to dáno architekturou sítě.

5.1.8 Kontrolní otázky ke Scénáři 1

Účelem následujících kontrolních otázek je objasnit zvolenou konfiguraci pro simulovanou transportní síť. Veškeré odpovědi se nachází v příloze C.1.

1. Co jsou *Shorthaul* a *Backhaul* linky a kde se nachází?
2. Jaký význam má *Loopback0* na SIAD směrovači?
3. Co je BDI rozhraní?
4. Jaký význam má `Null0` rozhraní u IPv4 a IPv6 statických cest na SIAD směrovači?
5. Jak je směrován IPv4 provoz ze SIAD směrovače k MSN páru neboli „upstream“?
6. Jak je směrován IPv6 provoz ze SIAD směrovače k MSN páru neboli „upstream“?
7. Jak je směrován IPv4 provoz z MSN páru k SIAD směrovači neboli „downstream“?
8. Jak je směrován IPv6 provoz z MSN páru k SIAD směrovači neboli „downstream“?
9. Na která spojení je navázán protokol BFD?
10. K čemu slouží OSPFv2 mezi MSN_A a MSN_B a jaký mají jednotlivé procesy význam?
11. K čemu slouží OSPFv3 mezi MSN_A a MSN_B a jaký mají jednotlivé procesy význam?
12. Které atributy jsou využívány při nastavení BGP?
13. Které hlavní route mapy mají na starost odchozí a příchozí provoz mezi MSN párem a PE u BGP?

5.1.9 Dodatečný krok 8. - nastavení SNMP protokolu

Protokol SNMP hraje v transportních sítích důležitou roli, protože dohlíží na celou síť. Dělí se na část manažera a na část týkající se agentů. Nastavení manažerovy části je popsáno viz Kap.3.8.1. Konfigurace popsaná níže se zabývá nastavením agentů využívajících „traps“ na SIAD, MSN_A a MSN_B směrovačích. Na všech zařízeních je nastavení stejné.

Nejprve nastavíme, kam budou „trap“ zprávy zasílány příkazem `snmp-server host <IP adresa> version 2c v2c`, přičemž adresa SNMP manažera v simulované transportní síti je 10.10.10.10. Na to navazuje povolení zasílání „trap“ zpráv spolu se specifikací, o jakých událostech bude agent manažera informovat.

Platí pro MSN_A MSN_B a SIAD

```
snmp-server host 10.10.10.10 version 2c v2c
snmp-server enable traps tty
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors bad-packet
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps config-copy
snmp-server enable traps syslog
snmp-server enable traps bfd
```

Dále je nastaveno posílání „linkUp/linkDown traps“ pomocí níže uvedeného příkazu.

Platí pro MSN_A MSN_B a SIAD

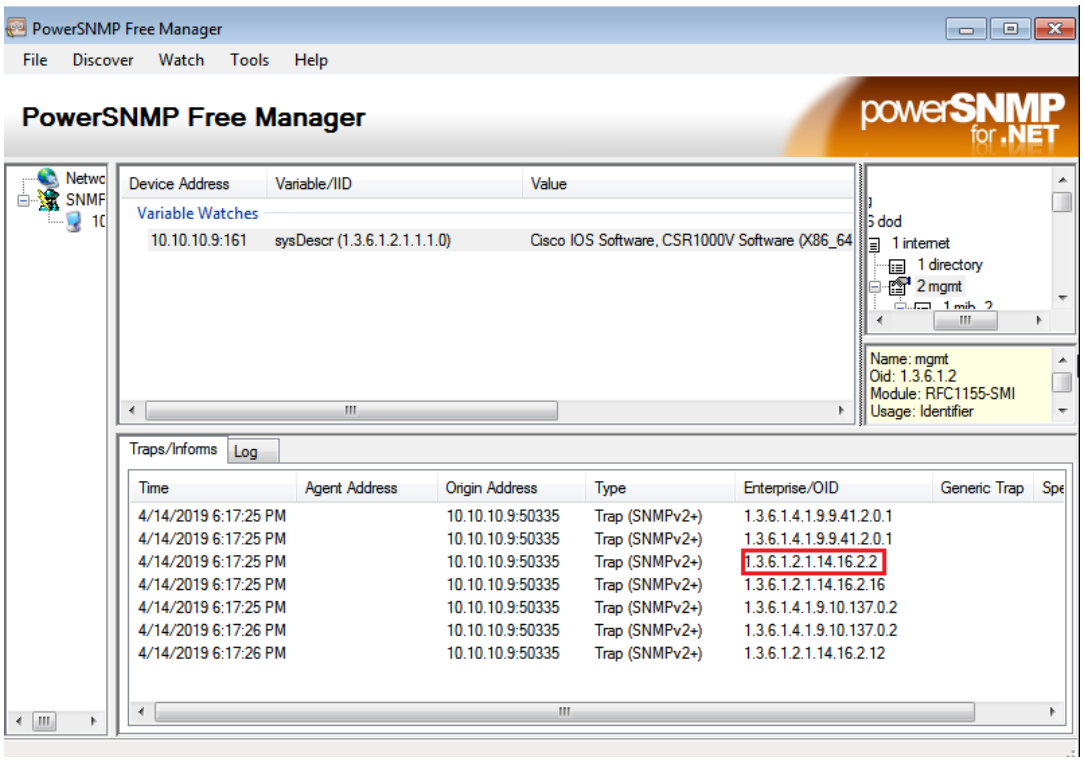
```
snmp-server trap link ietf
```

Zde je použit příkaz pro definování vztahu mezi manažerem a agentem. Řetězec, v tomto případě `cisco`, slouží jako heslo, které jim povoluje vzájemný přístup.

Platí pro MSN_A MSN_B a SIAD

```
snmp-server community cisco
```

✎ Správné fungování odesílání „trap“ zpráv např. pro MSN_A ověříme vypnutím portu *BDI1071*. V GUI SNMP manažera by mělo být vidět následující:



Obr. 5.8: Ověření funkčnosti odesílání trapů u MSN_A

Dle Enterprise/OID v Obr. 5.8 lze vyhledat, o jakou notifikaci se jedná. V červeném rámečku je to například *ospfNbrStateChange*.

5.1.10 Dodatečný krok 9. - nastavení QoS

Závěrečným 9. krokem je konfigurace QoS. Kvalita služeb je důležitou součástí transportní sítě. Každý operátor nebo poskytovatel sítě si dle nasmlouvaných hodnot či vlastní preference nastaví, které služby bude upřednostňovat před jinými a v jakém poměru. Jde o velmi komplexní záležitost.

Je třeba podotknout, že provoz přicházející na SIAD směrovač je již označkovaný díky síťovému generátoru Ostinato, který simuluje NodeB a eNodeB. Jeho nastavení je popsáno v Kroku 2.

Ve směrovačích s Cisco IOS XE se kvalita služeb nastavuje pomocí **class map** a **policy map**. Aby QoS bylo na směrovači funkční, je potřeba dodržet jasnou hierarchii: tedy nejprve vytvořit **class-map** a poté ji přiřadit do **policy-map** a tu pak nastavit na konkrétní rozhraní.

1. **Vytvoření Class map**

Nejprve na směrovači vytvoříme class mapy. Class mapy klasifikují příchozí nebo procházející síťovou komunikaci na základě shody se specifickými kritérii. Pomocí příkazu `match` vybereme provoz, o který máme zájem. V definici třídy lze určit, zda musí všechny výběry (match) souhlasit, tedy `match-all` nebo alespoň jeden, neboli `match-any`.

(a) **SIAD:**

Jako první vytvoříme class mapy pro příchozí data. Proto vytvoříme třídu Control. Třída hledá shodu s DSCP `cs4`, `cs5` a `cs7`, viz Tab. 2.4 v Kap.2.

```
class-map match-any Control
 description "Mobility signaling and Network Protocol traffic"
  match dscp cs6
  match dscp cs7
  match dscp cs4
```

Jelikož se jedná o mobilní transportní síť, jsou zde na třídy rozlišeny i Real Time aplikace (class mapa COS1), 3G data (class mapa COS2) a data u LTE (class mapa COS3). S Real Time aplikacemi se pojí hodnota dscp `cs5` a `ef`. 3G a 4G datový přenos se rozlišuje příkazem `match precedence <číslo>`, kde hodnota čísla nemá matematický význam (2 není významnější než 1), ale slouží pouze k označení. Číslo u IP precedence může nabývat hodnotu od 0 do 7. Tato čísla mají doporučená použití. Ve vlastní síti, kde řídíme prvotní značkování při vstupu do sítě, je však možné využít čísla libovolně. Např. `2` má doporučené použití pro značení datových aplikací, `3` je doporučeno používat pro signalizaci hovorů, což v případě Kroku 9. neodpovídá a značí se tím datový přenos u 3G. Číslo `4` je doporučeno pro videokonference a streaming, `6` a `7` pro značení paketů, pro kontrolu a dohled nad sítí.

```
class-map match-any COS1
 description "Real Time applications (3G Voice, VoLTE)"
  match dscp cs5
  match dscp ef

class-map match-any COS2
 description "UMTS R99, HS traffic"
  match precedence 3

class-map match-any COS3
 description "LTE data traffic"
  match precedence 2
```

Dále se využívá tzv. `qos-group`, a to především pro odchozí provoz. Je to interní značení hlavičky paketu uvnitř směrovače. Nemá vliv mimo daný směrovač.

Počet skupin, které IOS XE umí rozlišit, je od 0 do 99. Využití qos-group vyžaduje vytvoření nových class map. Jejich název je zvolen `qos-group-<název již existující třídy>`, kde existujícími třídami jsou Control, COS1, COS2 a COS3. Příkazem `match qos-group <číslo>` se hledá shoda, zda některý paket neobsahuje tuto značku. Žádný paket však zatím tuto značku obsahovat nemůže, protože nastavení tohoto vnitřního značkování proběhne až při konfiguraci **policy map**. Hodnota čísla u `match qos-group <číslo>` proto opět nemá matematický význam, jedná se pouze o značení.

```
class-map match-any qos-group-control
 match qos-group 6

class-map match-all qos-group-cos1
 match qos-group 5

class-map match-all qos-group-cos2
 match qos-group 3

class-map match-all qos-group-cos3
 match qos-group 2
```

✎ Správnost nastavení lze ověřit následujícím příkazem:

show class-map

```
SIAD#sh class-map
 Class Map match-any class-default (id 0)
   Match any

 Class Map match-any Control (id 1)
  Description: "Mobility signaling and Network Protocol traffic"
   Match   dscp cs6 (48)
   Match   dscp cs7 (56)
   Match   dscp cs4 (32)

 Class Map match-all qos-group-cos1 (id 2)
   Match qos-group 5

 Class Map match-all qos-group-cos3 (id 3)
   Match qos-group 2

 Class Map match-all qos-group-cos2 (id 4)
   Match qos-group 3

 Class Map match-any COS3 (id 5)
  Description: "LTE data traffic"
   Match   precedence 2

 Class Map match-any COS2 (id 6)
  Description: "UMTS R99, HS traffic"
   Match   precedence 3
```

```
Class Map match-any COS1 (id 7)
  Description: "Real Time applications (3G Voice, VoLTE)"
   Match   dscp cs5 (40)
   Match   dscp ef (46)

 Class Map match-any qos-group-control (id 8)
   Match qos-group 6
```

✎ Nabízí se otázka, zda má vytvoření těchto tříd nějaký dopad na provoz v dané topologii. Opak je pravdou. Aby měly nakonfigurované třídy nějaký dopad na provoz, musí se nejdříve přiřadit k policy mapám a poté ke konkrétním portům.

(b) **MSN:**

Na MSN jsou mapy tříd rozděleny stejným způsobem jako u SIAD směrovače, jedná se stále o stejná data. Rozdíl je ve zpracování class map policy mapami. Příkazy je nutné zadat na obou MSN směrovačích.

Platí pro MSN_A i MSN_B

```
class-map match-any Control
 description "Mobility signaling and Network Protocol traffic"
  match dscp cs6
  match dscp cs7
  match dscp cs4

class-map match-any COS3
 description "LTE data traffic"
  match precedence 2

class-map match-any COS2
 description "UMTS R99, HS traffic"
  match precedence 3

class-map match-any COS1
 description "Real Time applications (3G Voice, VoLTE)"
  match dscp cs5
  match dscp ef
```

Stejné zůstává i hledání shody v provozu dle stejných qos-group jako v případě SIAD směrovače. MSN směrovače používají stejné vnitřní značkování.

Platí pro MSN_A i MSN_B

```
class-map match-any qos-group-control
 match qos-group 6

class-map match-all qos-group-cos1
 match qos-group 5

class-map match-all qos-group-cos3
 match qos-group 2

class-map match-all qos-group-cos2
 match qos-group 3
```

✎ Pro ověření správnosti nastavení zadáme následující příkaz:

show run class-map

```
MSN_A#sh run class-map
Building configuration...

Current configuration : 647 bytes
!
class-map match-any Control
  description "Mobility signaling and Network Protocol traffic"
 match dscp cs6
 match dscp cs7
 match dscp cs4
class-map match-all qos-group-cos1
 match qos-group 5
class-map match-all qos-group-cos3
 match qos-group 2
class-map match-all qos-group-cos2
 match qos-group 3
class-map match-any COS3
  description "LTE data traffic"
 match precedence 2
class-map match-any COS2
  description "UMTS R99, HS traffic"
 match precedence 3
class-map match-any COS1
  description "Real Time applications (3G Voice, VoLTE)"
 match dscp cs5
 match dscp ef
class-map match-any qos-group-control
 match qos-group 6
end
```

✎ Nabízí se otázka, zda přidělujeme provozu ve výše uvedené konfiguraci nějaké značky a co způsobuje příkaz `match`? Značky není třeba přidělit, protože jsou přiděleny generátorem provozu Ostinato. Příkazem `match` systém hledá shodu s informací obsaženou v paketu a s nastavením na směrovači.

2. **Vytvoření Policy map**

Dalším krokem v nastavení QoS je vytvoření policy map. Policy mapy definují řadu funkcí aplikovaných na klasifikovaný přenos. Příkladem funkcí je označení provozu, nastavení IP precedence, DSCP nebo řešení front, omezení či vyhrazení pásma provozu, atd.
Na zařízení je vždy nutné nejprve stanovit pravidla pro danou policy mapu přiřazením jednotlivých class map a k nim námi zvolených pravidel.
Všechen ostatní provoz, který není zařazen do nějaké třídy, se automaticky zařadí do výchozí třídy, na kterou se neuplatní QoS. Použije se best-effort metoda. Je možné i se zbylým provozem pracovat vytvořením policy mapy a vložením `class class-default`.

Vytvoření policy mapy ještě neznamená, že se začne daná politika provádět. Je ještě nutné přidělit policy mapy portům příkazem `service-policy` a stanovit, zda se bude mapa týkat příchozích nebo odchozích paketů. Na porty lze přiřadit více policy map.

(a) **SIAD:**

U SIAD směrovače je nejprve nutné nastavit policy mapu na příchozí provoz jak na *Shorthaul*, tak i na *Backhaul*. Policy mapa přidělí dříve stanoveným třídám `Control,COS1,COS2,COS3` vnitřní značku `qos-group` příkazem `set qos-group <číslo>`. Tento příkaz lze použít pouze na příchozí provoz. Čísla jsou zvolena tak, aby odpovídala číslům používaným u IP precedence a také aby korespondovala s již nastavenými hodnotami u class map `qos-group-<control,cos1, cos2,cos3>`.

```
policy-map SIAD_QOS_Policy_ingress
 description "SIAD ingress policy for short-haul and backhaul"
 class Control
  set qos-group 6
 class COS1
  set qos-group 5
 class COS2
  set qos-group 3
 class COS3
  set qos-group 2
```

Policy mapa SIAD_QOS_Policy_ingress je pak vložena na interface *Gi1* vedoucí k NodeB, *Gi11* k eNodeB a *Gi7* k MSN párům.

```
interface Gi1
service-policy input SIAD_QOS_Policy_ingress

interface Gi11
service-policy input SIAD_QOS_Policy_ingress

interface Gi7
service-policy input SIAD_QOS_Policy_ingress
```

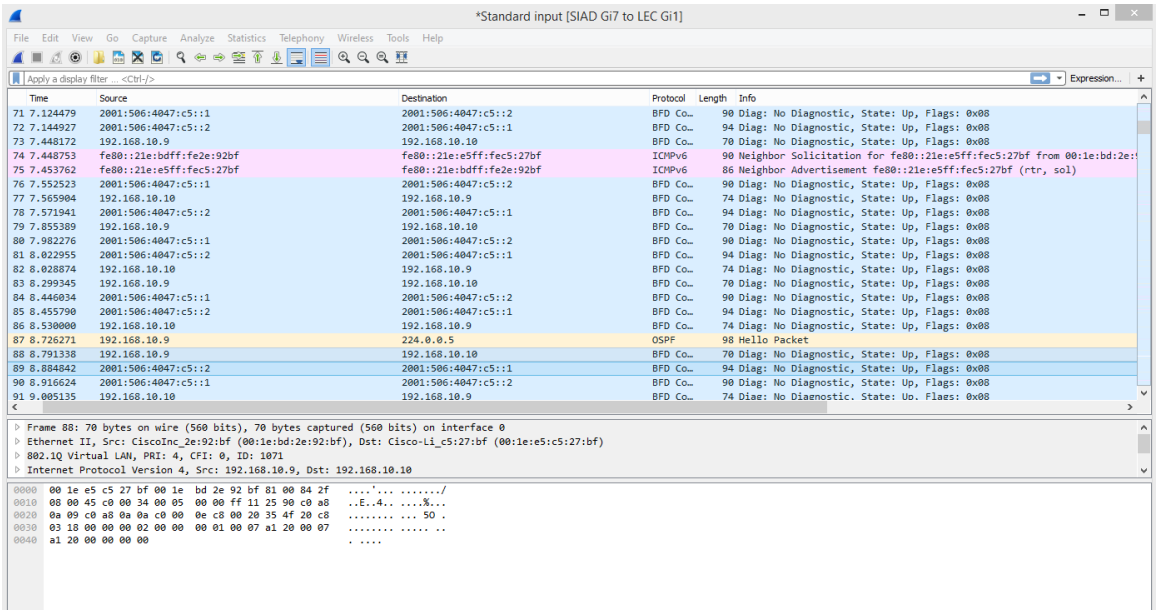
✎ Po nastavení příchozího provozu je možné sledovat přiřazení příchozího provozu do `class-map Control`. Dochází zde k výměně směrovacích informací a dalších údajů relevantních pro fungování sítě, viz Obr. 5.9. Na obrázku je možné pozorovat především velmi četnou výměnu BFD kontrolních paketů.

show policy-map int g7

```
SIAD#sh policy-map int g7
 GigabitEthernet7

  Service-policy input: SIAD_QOS_Policy_ingress

    Class-map: Control (match-any)
      62534 packets, 5027212 bytes
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match:  dscp cs6 (48)
      Match:  dscp cs7 (56)
      Match:  dscp cs4 (32)
      QoS Set
        qos-group 6
          Marker statistics: Disabled
```



Obr. 5.9: Ukázka výměny kontrolních paketů mezi SIAD směrovačem a MSN párem

Dalším krokem je stanovení policy map pro odchozí provoz. Provoz se rozlišuje dle použité technologie za portem (NodeB a eNodeB) a sjednané rychlosti na *Backhaul* lince. Policy mapy pro odchozí provoz jsou založeny na procentuálním rozdělení prostředků.

Jednoznačně největší přednost mají Real Time aplikace, tedy třída COS1, díky příkazu `priority percent <%> <burst>`. Procentuálně je jim vždy přiděleno 50% z dostupných prostředků. Velkou roli hraje hodnota „burst“. Tento údaj stanovuje velikost časového okna a je rozdílovým parametrem mezi nastavovanými policy mapami při nastavování různých rychlostí.

Zbývající šířka pásma je rozdělena mezi ostatní třídy příkazem `bandwidth remaining percent <%>`, kde 50% je vyhrazeno pro Control, 20% pro 3G data, 20%

procent pro 4G datový provoz a 10% připadá na zbývající provoz.
Pravidla stanovená v předchozích odstavcích se aplikují na vytvořené třídy `qos-group-<control,cos1,cos2,cos3>` a `class-default`.

```
policy-map SIAD_Output_Policy_Child_100M
 description "SIAD backhaul egress child policy for 100M"
 class qos-group-cos1
  priority percent 50 6250
 class qos-group-control
  bandwidth remaining percent 50
 class qos-group-cos2
  bandwidth remaining percent 20
 class qos-group-cos3
  bandwidth remaining percent 20
 class class-default
  bandwidth remaining percent 10

policy-map SIAD_Output_Policy_Child_1G
 description "SIAD backhaul egress child policy for 1G"
 class qos-group-cos1
  priority percent 50 62500
 class qos-group-control
  bandwidth remaining percent 50
 class qos-group-cos2
  bandwidth remaining percent 20
 class qos-group-cos3
  bandwidth remaining percent 20
 class class-default
  bandwidth remaining percent 10

policy-map SIAD_Output_Policy_Child_800M
 description "SIAD backhaul egress child policy for 800M"
 class qos-group-cos1
  priority percent 50 50000
 class qos-group-control
  bandwidth remaining percent 50
 class qos-group-cos2
  bandwidth remaining percent 20
 class qos-group-cos3
  bandwidth remaining percent 20
 class class-default
  bandwidth remaining percent 10
exit
```

Aby byly funkční, je opět třeba výše zmíněné policy mapy přiřadit na porty. Policy mapa u NodeB (3G) je vždy `SIAD_Output_Policy_Child_100M`. U eNodeB (4G) je využívána `SIAD_Output_Policy_Child_1G`. Je proto možné mapy ihned přiřadit na porty.

```
interface g1
service-policy output SIAD_Output_Policy_Child_100M

interface g11
service-policy output SIAD_Output_Policy_Child_1G
```

Pouze na portu *g7* dochází ke změnám rychlosti přenosu z důvodu měnících

se podmínek v LEC části. Jako výchozí byla pro tuto práci zvolena rychlost 800 Mb/s. Jelikož port *g7* je *Backhaul* linka, je zapotřebí detailněji zohlednit i všechen provoz, který by se mohl objevit na portu a který nepodléhá QoS. Řešením je rodičovská policy mapa `SIAD_Parent_Policy _Egress_800M`, kde se nastaví třída `class-default` příkazem `shape average <target bits rate> <burst bits per interval>` a přiřadí podřazené policy mapy `SIAD_Output_Policy_Child_800M`.

```
policy-map SIAD_Parent_Policy_Egress_800M
description "SIAD backhaul Egress parent policy for 800M"
class class-default
 shape average 760000000 400000
 service-policy SIAD_Output_Policy_Child_800M
exit
```

Opět je nutné přiřadit policy mapy (nyní pouze rodičovské) na port.

```
interface g7
 service-policy output SIAD_Parent_Policy_Egress_800M
```

✎ Správnost nastavení lze ověřit i pomocí příkazu `show policy-map`. Jelikož by ale takový výpis byl příliš obsáhlý, je z něj ukázána pouze část.

show policy-map | sec 800M

```
SIAD#sh policy-map | sec 800M
  Policy Map SIAD_Parent_Policy_Egress_800M
      Description: "SIAD backhaul Egress parent policy for 800M"
    Class class-default
      Average Rate Traffic Shaping
      cir 760000000 (bps) bc 3040000 (bits)

  Policy Map SIAD_Output_Policy_Child_800M
      Description: "SIAD backhaul egress child policy for 800M"
    Class qos-group-cos1
      priority 50 (%) 50000
    Class qos-group-control
      bandwidth remaining 50 (%)
    Class qos-group-cos2
      bandwidth remaining 20 (%)
    Class qos-group-cos3
      bandwidth remaining 20 (%)
    Class class-default
      bandwidth remaining 10 (%)
```

✎ Všimněme si, že je `qos-group-cos1` přiděleno 50% ze všech dostupných prostředků a `qos-group-cos2` a `qos-group-cos3` pouze 20% zbývajících prostředků. Je to z toho důvodu, že se jedná o mobilní síť, kde má třída `qos-group-cos1` na starost právě tento typ provozu. Na datový provoz je vyhrazeno méně prostředků.

(b) **MSN:**

U MSN směrovačů je nutné nastavit policy mapy pro příchozí provoz tak, aby došlo k přidělení vnitřní značky `qos-group` u dříve vytvořených tříd `Control,COS1, COS2,COS3`.

Platí pro MSN_A i MSN_B

```
policy-map MSN_QOS_Policy_ingress
 description "MSN ingress policy"
 class Control
  set qos-group 6
 class COS1
  set qos-group 5
 class COS2
  set qos-group 3
 class COS3
  set qos-group 2
```

Nově vytvořenou policy mapu přiřadíme na porty. Porty se liší dle použitého MSN směrovače.

Platí pro MSN_A

```
interface g2
service-policy input MSN_QOS_Policy_ingress
interface g3
service-policy input MSN_QOS_Policy_ingress
interface g5
service-policy input MSN_QOS_Policy_ingress
```

Platí pro MSN_B

```
interface g4
service-policy input MSN_QOS_Policy_ingress
interface g3
service-policy input MSN_QOS_Policy_ingress
interface g5
service-policy input MSN_QOS_Policy_ingress
```

Poté nastavíme mapy pro odchozí provoz. Na obou směrovačích vytvoříme policy mapu `MSN_Output_Policy` sloužící jako výchozí pro všechna rozhraní na MSN směrovači. Tato výchozí policy mapa přiděluje 50% z dostupných prostředků třídě `qos-group -cos1`. Velikost `<burst>`, která by měla následovat za hodnotou 50% není specifikována a je ponecháno na směrovači, aby si ji sám určil. Zbylé prostředky jsou rozděleny v poměru: 50% pro třídu `qos-group-control`, 20% pro `qos-group-cos2` a 20% pro `qos-group-cos3` a zbývajících 10% je přiřazeno třídě `class-default`.

Platí pro MSN_A i MSN_B

```
policy-map MSN_Output_Policy
 description Egress Policy for IP MSN
 class qos-group-cos1
   priority percent 50
 class qos-group-control
   bandwidth remaining percent 50
 class qos-group-cos2
   bandwidth remaining percent 20
 class qos-group-cos3
   bandwidth remaining percent 20
 class class-default
   bandwidth remaining percent 10
```

Policy mapu odchozího provozu `MSN_Output_Policy` v případě potřeby dále upravujeme přes rodičovskou policy mapu, která upravuje rychlost provozu na rozhraní. V případě tohoto scénáře se jedná o BDI rozhraní vedoucí k SIAD směrovači, kde se upraví odchozí provoz policy mapou `MSN_Shaper_Parent_Policy_egress_800M`.

Platí pro MSN_A i MSN_B

```
policy-map MSN_Shaper_Parent_Policy_egress_800M
 class class-default
  shape average 760000000 400000
   service-policy MSN_Output_Policy
```

Tato specifičtější policy mapa je přidělena na BDI rozhraní příslušící daným MSN směrovačům.

Platí pro MSN_A

```
interface BDI1071
 service-policy output MSN_Shaper_Parent_Policy_egress_800M
```

Platí pro MSN_B

```
interface BDI2071
 service-policy output MSN_Shaper_Parent_Policy_egress_800M
```

Na linkách mezi MSN směrovači a směrem do Core Backbone (CBB) je využito pouze `MSN_Output_Policy`.

✎ Následujícím příkazem lze sledovat rozdělení provozu na rozhraní dle nastavených policy map. Pro ukázku byl vybrán *int g6*, na kterém se nachází pouze `MSN_Output_Policy`. Byly vybrány třídy „Control“ a „Default“:

show policy-map int g6

```
MSN_A#show policy-map int g6
 GigabitEthernet6

  Service-policy output: MSN_Output_Policy

    Class-map: Control (match-any)
      41 packets, 4462 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match:  dscp cs6 (48)
      Match:  dscp cs7 (56)
      Match:  dscp cs4 (32)
      Queueing
      queue limit 2048 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 41/4462
      QoS Set
        cos 4
          Marker statistics: Disabled

   Class-map: class-default (match-any)
       27 packets, 2148 bytes
       30 second offered rate 0000 bps, drop rate 0000 bps
       Match: any
       Queueing
       queue limit 4096 packets
       (queue depth/total drops/no-buffer drops) 0/0/0
       (pkts output/bytes output) 27/2148
       QoS Set
        cos 4
          Marker statistics: Disabled
      bandwidth remaining 10%
```

✎ Za zmínku stojí, že policy mapa `MSN_Shaper_Parent_Policy_egress_800M` je přidělena k BDI rozhraní a nikoli k celému rozhraní.
Je to z toho důvodu, že k MSN lze připojit více SIAD směrovačů a pod jedním fyzickým portem tak může být více BDI. Pro každý z nich musí existovat možnost nastavit odlišnou rychlost.

5.2 Scénář 2 - Časté chyby v transportní síti

Jak již bylo zmíněno v úvodu této páté kapitoly, druhý scénář je založený na hledání chyb v konfiguraci. Cílem Scénáře 2 je upozornit na možné komplikace při konfiguraci jednotlivých protokolů a na chyby, které se v praxi často objevují.

Pro pochopení problematiky tohoto scénáře je nejprve nutné zcela pochopit Scénář 1.

Oba dva scénáře počítají s využitím dvou emulačních programů - GNS3 a EVE-NG. Ve Scénáři 2 nejprve popíšeme postup, jak konfiguraci s chybným nastavením nahrát do těchto programů.

Nefunkční části sítě budou popsány pomocí dvou „TroubleTickets“, které je zapotřebí opravit. Pro objasnění oprav jsou do Scénáře 2 zakomponovány kontrolní otázky, jejichž řešení se nachází v příloze C.2. Řešení dvou „TroubleTickets“ společně s postupem je uvedeno na konci této kapitoly. Jedná se o vzorová řešení, nicméně úlohy jsou řešitelné i jinými postupy. Opravenou konfiguraci je možné si ověřit s výpisem konfigurace ze Scénáře 1, která se nachází v příloze B.

Výpisy chybných konfigurací pro Scénář 2 jsou nahrány na přiloženém DVD.

5.2.1 Nahrání konfigurace s chybami

Nahrávání konfigurace se u emulačních programů GNS3 a EVE-NG liší. Rozdíly souvisí s exportem simulačních scénářů. EVE-NG je schopen exportovat pouze soubor o velikosti v řádech kilobytů, neobsahuje obrazy směrovačů a obsahuje pouze informace o topologii s uloženou konfigurací. Naopak GNS3 exportuje soubory o velikosti několika gigabytů a uchovává v sobě i nastavení uložená v paměti *bootflash*.

GNS3

Program GNS3 umožňuje obsáhlejší a komplexnější export.

V paměti směrovačů použitých v této práci je v adresáři *bootflash* uložena start-up konfigurace pro Scénář 2 (viz níže). Konfigurace se načte příkazem `copy Scenario2-TTX running-config` u SIAD, MSN_A a MSN_B směrovačů. Tato varianta počítá s importováním přenosného souboru vytvořeného v GNS3 *GNS3_Simulation_Scenarios1_2_TT1-2.gns3project* a uloženého na přiloženém DVD.

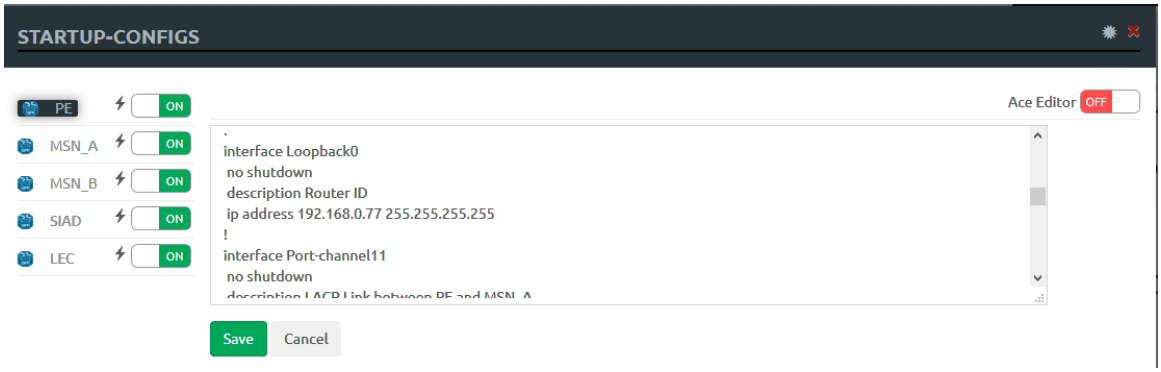
```
101      11109 Apr 19 2019 13:38:17 +00:00 /bootflash/Scenario1
102      10986 Apr 19 2019 16:32:26 +00:00 /bootflash/Scenario2-TT-1
103      11109 Apr 19 2019 22:46:45 +00:00 /bootflash/Scenario2-TT-2
```

EVE-NG

Ačkoliv EVE-NG není schopen přenášet laboratoře spolu s pamětí směrovačů, má tu výhodu, že obsahuje záložku *Startup-configs*. Tato záložka ukazuje start-up konfiguraci pro každý uzel v aktuální laboratoři. Chceme-li, aby uzel při spuštění nahrál tuto konfiguraci, zvolí se možnosti **ON**, viz Obr. 5.10.

Jedná se o textový soubor, takže je pro uživatele jednoduché s nahrávanou konfigurací manipulovat.

Pro přípravu Scénáře 2 se vychází z laboratoře vytvořené Scénářem 1 a tedy importováním souboru *EVE-NG_Simulation_Scenarios1_2_TT1-2.zip*. Do záložky *Startup-configs* se však vloží konfigurace pro SIAD, MSN_A a MSN_B uvedené v přiloženém DVD. Zvolí se možnost **ON** a jednotlivá zařízení se spustí.



Obr. 5.10: Ukázka záložky pro nahrávání start-up konfigurací

5.2.2 TroubleTicket-1 (TT-1)

V případě TT-1 byl nahlášen problém s Bearer portem *BDI212* na SIAD směrovači vedoucím k eNodeB. Nelze s ním komunikovat z vnější sítě, kterou představuje směrovač PE. Úkolem je zprovoznit tuto komunikaci tak, aby byl příkaz `ping 2001:506:4247:340:0:1:5653:1` úspěšný.

Součástí TT-1 je také informace, že z posledních dostupných logů proběhla komunikace přes záložní směrovač a nikoliv přes primární směrovač. Tento problém je zapotřebí ověřit a případně vyřešit.

5.2.3 TroubleTicket-2 (TT-2)

TT-2 nahlásil problém s OSPFv2 sousedstvím mezi SIAD směrovačem a MSN párem. V provozu je pouze sekundární linka. Úkolem je znovu obnovit běžný provoz se dvěma funkčními linkami.

Dále je zapotřebí opravit agregovanou linku mezi MSN párem, kde má fungovat protokol LACP spojující dvě fyzické linky do jediné logické.

Posledním nahlášeným problémem je absence sítě OAM pro NodeB na směrovači PE. Nedochází k redistribuci NodeB OAM adres do BGP a PE není schopen s touto sítí komunikovat.

5.2.4 Kontrolní otázky ke Scénáři 2

Účelem následujících kontrolních otázek je objasnit vyřešení TroubleTicket 1 a TroubleTicket 2. Ke každému „TroubleTicket“ je záměrně zvolena trojice otázek. Odpovědi se nachází v příloze C.2.

Otázky pro TroubleTicket-1

1. Pomocí jakých příkazů se BDI rozhraní naváže na fyzické rozhraní na směrovači?
2. Jakou administrativní vzdálenost mají statické cesty pro eNodeB vedoucí z MSN_A (primárního) směrovače k SIAD směrovači a jaká je tato administrativní vzdálenost v případě směrovače MSN_B (sekundárního)?
3. Na jaké hodnoty jsou nastaveny *hello interval* a *dead interval* pro OSPFv3 na MSN páru?

Otázky pro TroubleTicket-2

1. Jakým příkazem spustíme BFD protokol na rozhraní?
2. Jakým příkazem přiřadíme fyzické rozhraní k „port-channel“ a jak na „port-channel“ skrze fyzické rozhraní spustíme protokol LACP?
3. Jakým způsobem jsou redistribuovány OAM adresy NodeB do protokolu BGP na MSN páru?

5.2.5 Řešení: TroubleTicket-1

- Celý proces hledání chyb začíná kontrolou, zda je daná adresa opravdu nedostupná.

PE

```
PE#ping 2001:506:4247:340:0:1:5653:1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:506:4247:340:0:1:5653:1, timeout is 2 seconds:
XXXXX
Success rate is 0 percent (0/5)
```

- V dalším kroku je potřeba se zaměřit na zařízení, na kterém se problém nachází a zkontrolovat stav jeho rozhraní.

SIAD

```
SIAD#sh ip int br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       unassigned      YES unset  up                    up
GigabitEthernet2       unassigned      YES unset  administratively down down
GigabitEthernet3       unassigned      YES unset  administratively down down
GigabitEthernet4       unassigned      YES unset  administratively down down
GigabitEthernet5       unassigned      YES unset  administratively down down
GigabitEthernet6       unassigned      YES unset  administratively down down
GigabitEthernet7       unassigned      YES unset  up                    up
GigabitEthernet8       unassigned      YES unset  administratively down down
GigabitEthernet9       unassigned      YES unset  administratively down down
GigabitEthernet10      unassigned      YES unset  administratively down down
GigabitEthernet11      unassigned      YES unset  up                    up
GigabitEthernet12      unassigned      YES unset  administratively down down
BDI101                 172.16.13.17    YES TFTP   up                    up
BDI102                 172.17.13.33    YES TFTP   up                    up
BDI211                 unassigned      YES unset  up                    up
BDI212                 unassigned      YES unset  down                  down
BDI1071                192.168.10.10   YES TFTP   up                    up
BDI2071                192.168.20.22   YES TFTP   up                    up
Loopback0              192.168.0.99    YES TFTP   up                    up
```

Zde pozorujeme, že je port ve stavu *down*.

- Jelikož je BDI pouze logický port, je nutné zkontrolovat stav fyzického portu, což je v případě *BDI212* rozhraní *G11*. Ten se nachází ve stavu *up*.
- Následně se podíváme na konkretní konfiguraci těchto portů.

SIAD

```
SIAD#sh run int bdi212
Building configuration...

Current configuration : 155 bytes
!
interface BDI212
 description Link to eNodeB Bearer
 no ip address
 no ip redirects
 load-interval 30
 ipv6 address 2001:506:4247:340:0:1:5653:1/64
end
```

```
SIAD#sh run int g11
Building configuration...

Current configuration : 329 bytes
!
interface GigabitEthernet11
 description Link to eNodeB
 no ip address
 load-interval 30
 negotiation auto
 service-policy input SIAD_QOS_Policy_ingress
 service-policy output SIAD_Output_Policy_Child_1G
 service instance 211 ethernet
  encapsulation dot1q 211
  rewrite ingress tag pop 1 symmetric
  bridge-domain 211
```

Zde zjišťujeme problém: *BDI212* není přiřazen k rozhraní *G11*. Tuto chybu opravíme těmito příkazy:

SIAD

```
interface GigabitEthernet11
service instance 212 ethernet
 encapsulation dot1q 212
 rewrite ingress tag pop 1 symmetric
 bridge-domain 212
```

- Při kontrole funkčnosti ping příkazy ze směrovače PE opět dostáváme negativní výsledky. Proto je vhodné použít příkaz `traceroute`, abychom zjistili, kde se komunikace zastaví.

PE

```
PE#traceroute 2001:506:4247:340:0:1:5653:1
Type escape sequence to abort.
Tracing the route to 2001:506:4247:340:0:1:5653:1

  1 2001:506:4600:8228::2 3 msec 3 msec 2 msec
  2  *  *  *
  3  *  *  *
```

Z výpisu lze vyčíst, že z PE jsme schopni se dostat pouze ke směrovači MSN_A.

- Přesuneme se tedy na MSN_A, kde zkusíme ping test. Opět neúspěšně. Provedeme tedy kontrolu záznamů ve směrovací tabulce.

MSN_A

```
MSN_A#sh ipv6 route
IPv6 Routing Table - default - 15 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, RL - RPL, O - OSPF Intra, OI - OSPF Inter
       OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1
       ON2 - OSPF NSSA ext 2, la - LISP alt, lr - LISP site-registrations
       ld - LISP dyn-eid, a - Application
C   2001:506:4047:C5::/64 [0/0]
     via BDI1071, directly connected
L   2001:506:4047:C5::1/128 [0/0]
     via BDI1071, receive
S   2001:506:4247::/50 [1/0]
     via Null0, directly connected
S   2001:506:4247:4000::/50 [1/0]
     via Null0, directly connected
S   2001:506:4447::/50 [1/0]
     via Null0, directly connected
S   2001:506:4447:4000::/50 [1/0]
     via Null0, directly connected
LC  2001:506:4600:8E1::2/128 [0/0]
     via Loopback20, receive
O   2001:506:4600:8E9::2/128 [110/100]
     via FE80::21E:7AFF:FE3D:F1BF, BDI90
C   2001:506:4600:8228::/64 [0/0]
     via Port-channel11, directly connected
L   2001:506:4600:8228::2/128 [0/0]
     via Port-channel11, receive
C   2001:506:4600:C0C9::/64 [0/0]
     via BDI90, directly connected
L   2001:506:4600:C0C9::1/128 [0/0]
     via BDI90, receive
C   2001:506:4600:C154::/64 [0/0]
     via BDI91, directly connected
L   2001:506:4600:C154::1/128 [0/0]
     via BDI91, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Adresa portu se ve směrovací tabulce nenachází.

- Ze Scénáře 1 víme, že provoz z MSN páru je směrován statickými cestami pro eNodeB provoz, tedy na rozhraní *BDI212*. Statické cesty se ověří příkazem `sh run | i ipv6 route`. A jak lze vidět níže, na MSN_A uvedeny nejsou.

MSN_A

```
MSN_A#sh run | i ipv6 route
ipv6 route static bfd BDI1071 2001:506:4047:C5::2
ipv6 route 2001:506:4247::/50 Null0 name ENodeBBER
ipv6 route 2001:506:4247:4000::/50 Null0 name ENodeBBER
ipv6 route 2001:506:4447::/50 Null0 name ENodeBOAM
ipv6 route 2001:506:4447:4000::/50 Null0 name ENodeBOAM
```

- Oprava se provede vložením IPv6 statických cest pro OAM a Bearer směrem k eNodeB.

MSN_A

```
ipv6 route 2001:506:4247:340::/61 BDI1071 2001:506:4047:C5::2
ipv6 route 2001:506:4447:340::/61 BDI1071 2001:506:4047:C5::2
```

- Nyní je zapotřebí vyřešit druhou část nahlášené chyby a to tak, aby byl provoz směrován přes primární směrovač, tedy MSN_A. Z příkazu uvedeného níže vidíme, že provoz z MSN_B jde přímo na SIAD směrovač.

MSN_B

```
MSN_B#trace 2001:506:4247:340:0:1:5653:1
Type escape sequence to abort.
Tracing the route to 2001:506:4247:340:0:1:5653:1

  1 2001:506:4047:40C5::2 6 msec 6 msec 3 msec
```

- Ze Scénáře 1 víme, že spojení a předávání cest má mezi MSN párem na starost OPSFv3, konkrétně proces 3. Při výpisu ale vidíme, že funkčním je mezi MSN párem pouze proces 15.

MSN_B

```
MSN_B#sh ipv6 ospf nei
            OSPFv3 Router with ID (192.168.200.22) (Process ID 15)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.168.200.11    0   FULL/  -        00:00:58    20              BDI90
```

- Zkontrolujeme tedy nastavení samotného OSPF na směrovačích.

MSN_B

```
MSN_B#sh run | sec router ospfv3 3
router ospfv3 3
 router-id 192.168.0.22
 !
 address-family ipv6 unicast
  redistribute static metric-type 1
 exit-address-family
```

Stejný výsledek je i na MSN_A, pouze s jiným router-id, redistribuce statických cest je aktivována.

- Jelikož OSPFv3 proces 3 je konfigurován na portu *BDI91*, viz Obr. 5.6, je nutné provést kontrolu nastavení i zde.

MSN_A

```
MSN_A#sh run int bdi91 | i ospfv3
 ospfv3 3 network point-to-point
 ospfv3 3 hello-interval 20
 ospfv3 3 dead-interval 100
 ospfv3 3 cost 100
 ospfv3 3 ipv6 area 10
```

MSN_B

```
MSN_B#sh run int bdi91 | i ospfv3
 ospfv3 3 network point-to-point
 ospfv3 3 hello-interval 20
 ospfv3 3 dead-interval 70
 ospfv3 3 cost 100
 ospfv3 3 ipv6 area 10
end
```

Při porovnání jednotlivých nastavení na MSN směrovačích vidíme rozdílnou konfiguraci časů u časovačů. MSN_A má u *dead interval* hodnotu 100, MSN_B ale hodnotu 70. Jedná se o problém, který brání ve vytvoření sousedství.

- Oprava je provedena příkazem:

MSN_A

```
interface BDI91
no ospfv3 3 dead-interval 100
ospfv3 3 dead-interval 70
```

- Po vložení výše uvedeného příkazu se naváže spojení mezi MSN párem a jak lze pozorovat, provoz je již směrován přes MSN_A.

MSN_A

```
MSN_A#sh ipv6 ospf nei
            OSPFv3 Router with ID (192.168.0.11) (Process ID 3)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.168.0.22      0   FULL/  -        00:00:51    21              BDI91

            OSPFv3 Router with ID (192.168.200.11) (Process ID 15)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.168.200.22    0   FULL/  -        00:01:07    20              BDI90
```

MSN_B

```
MSN_B#trace 2001:506:4247:340:0:1:5653:1
Type escape sequence to abort.
Tracing the route to 2001:506:4247:340:0:1:5653:1

  1 2001:506:4600:C154::1 13 msec 4 msec 2 msec
  2 2001:506:4047:C5::2 4 msec 6 msec 4 msec
```

5.2.6 Řešení: TroubleTicket-2

- Nejprve se zaměříme na nefunkční primární linku mezi SIAD směrovačem a MSN_A.

MSN_B

```
SIAD#sh ip ospf nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.0.22      0   FULL/  -        00:01:09    192.168.20.21   BDI2071
192.168.0.11      0   DOWN/  -        00:00:56    192.168.10.9    BDI1071
```

- Na SIAD směrovači a MSN_A je třeba zkontrolovat, zda jsou všechny porty ve stavu *up*. To se provede příkazem `show ip interface brief`.
- Zkontrolujeme také nastavení samotného OSPFv2 na SIAD směrovači a MSN_A. Jedná se o proces 1.

SIAD

```
SIAD#sh run | sec ospf 10
router ospf 10
 router-id 192.168.0.99
 area 10 nssa
 passive-interface BDI101
 passive-interface BDI102
 network 172.16.13.16 0.0.0.15 area 10
 network 172.17.13.32 0.0.0.7 area 10
 network 192.168.0.99 0.0.0.0 area 10
 network 192.168.10.8 0.0.0.3 area 10
 network 192.168.20.20 0.0.0.3 area 10
 bfd all-interfaces
```

```
MSN_A#sh run | sec ospf 1
router ospf 1
 router-id 192.168.0.11
 area 10 nssa no-redistribution no-summary
 passive-interface GigabitEthernet1
 network 10.200.10.48 0.0.0.3 area 10
 network 192.168.0.0 0.0.255.255 area 10
 bfd all-interfaces
```

Zde je možné pozorovat, že nastavení odpovídá stejnému číslu a typu oblasti.

- Následně se provede kontrola nastavení na samotných rozhraních.

SIAD

```
SIAD#sh run int bdi1071
interface BDI1071
 description - Primary Vlan to MSN_A
 ip address 192.168.10.10 255.255.255.252
 no ip redirects
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf bfd strict-mode
 load-interval 30
 encapsulation dot1Q 1071
 ipv6 address 2001:506:4047:C5::2/64
```

MSN_A

```
MSN_A#sh run int bdi1071
interface BDI1071
 description Backhaul Primary interface to SIAD
 ip address 192.168.10.9 255.255.255.252
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf database-filter all out
 ip ospf bfd strict-mode
 ip ospf cost 10
 encapsulation dot1Q 1071
 ipv6 address 2001:506:4047:C5::1/64
 bfd template msn-bfd-template
```

OSPF intervaly nastavené u časovačů se rovnají. Na obou zařízením je přítomen BFD striktní mód. U SIAD směrovače však chybí nastavení intervalů pro BFD přenos.

- Ověříme stav protokolu BFD na SIAD směrovači a poté i na MSN_A.

SIAD

```
SIAD#show bfd neighbors
SIAD#
```

V případě SIAD směrovače nedostaneme žádný výstup, neboť BFD prokol není v provozu.

MSN_A

```
MSN_A#show bfd neighbors

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
77.66.55.45                            25/9          Up        Up        Po11
192.168.10.10                          11/1          AdminDown Down      BD1071

IPv6 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
2001:506:4047:C5::2                    28/2          AdminDown Down      BD1071
2001:506:4600:8228::1                  26/1          Up        Up        Po11
```

U MSN_A lze však vidět, že linky nejsou aktivní.

- Oprava spočívá v přidání časovačů na rozhraní *BDI1071*. Hodnoty intervalů musí odpovídat nastavení BFD protokolu u sousedního směrovače.

SIAD

```
 interface BDI1071
 bfd interval 500 min_rx 500 multiplier 3
*Apr 19 22:05:43.274: \%BFD-6-BFD_IF_CONFIGURE: BFD-SYSLOG: bfd config apply, idb:BDI1071
*Apr 19 22:05:43.275: \%BFD-6-BFD_SESS_CREATED: BFD-SYSLOG: bfd_session_created, neigh
     192.168.10.9 proc:OSPF, idb:BDI1071 handle:1 act
*Apr 19 22:05:43.333: \%BFDFSM-6-BFD_SESS_UP: BFD-SYSLOG: BFD session ld:4097 handle:1 is going UP
*Apr 19 22:05:43.440: \%OSPF-5-ADJCHG: Process 10, Nbr 192.168.0.11 on BDI1071 from LOADING to
     FULL, Loading Done
```

Při kontrole jsou již linky ve stavu *up*.

SIAD

```
SIAD#sh bfd neighbors

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
192.168.10.9                          4097/11        Up        Up        BD1071

IPv6 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
2001:506:4047:C5::1                     3/28         Up        Up        BD1071
```

- Nyní se zaměříme na agregaci linek mezi MSN párem. Prvotní kontrolu provedeme následujícím příkazem.

MSN_A

```
MSN_A#show etherchannel summary
Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1       Po1(RU)         LACP     Gi3(bndl) Gi5(susp)
11      Po11(RU)                LACP     Gi6(bndl) Gi7(bndl)

RU - L3 port-channel UP State
SU - L2 port-channel UP state
P/bndl -  Bundled
S/susp  - Suspended
```

Zjistili jsme, že port *Gi5* je ve stavu „suspended“ a veškerý provoz probíhá přes port *Gi3*.

- Provedeme kontrolu samotného portu.

MSN_A

```
MSN_A#sh run int gi5

interface GigabitEthernet5
 no ip address
 negotiation auto
 service-policy output MSN_Output_Policy
 channel-group 1 mode active
end
```

Zde se zdá být vše v pořádku. Stejnou kontrolu provedeme i u MSN_B.

MSN_B

```
MSN_B#sh run int gi5

interface GigabitEthernet5
 no ip address
 negotiation auto
 service-policy output MSN_Output_Policy
end
```

Zde vidíme, že chybí spojení s *Port-channel1*, a proto je protokol LACP ve stavu „suspended“.

- Oprava se provede následujícím způsobem:

MSN_B

```
interface GigabitEthernet5
channel-group 1 mode active
*Apr 19 22:35:05.575: \%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet5, changed
     state to up
*Apr 19 22:35:06.550: GigabitEthernet5 added as member-2 to port-channel1
```

```
MSN_B#show etherchannel summary
Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1       Po1(RU)         LACP     Gi3(bndl) Gi5(bndl)
12      Po12(RU)                LACP     Gi6(bndl) Gi7(bndl)

RU - L3 port-channel UP State
SU - L2 port-channel UP state
P/bndl -  Bundled
S/susp  - Suspended
```

- Na závěr zkontrolujeme problém redistribuce OAM IPv4 adres na PE směrovači. Příkazem `show ip route` zjistíme, že adresa sítě OAM pro NodeB (172.17.0.0/29) není obsažena ve směrovací tabulce.

PE

```
PE#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B        10.10.10.0/24 [20/0] via 77.66.55.46, 00:00:11
B        10.200.10.28/30 [20/0] via 77.66.55.46, 00:00:11
B        10.200.10.48/30 [20/0] via 77.66.55.46, 00:00:11
      77.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        77.66.55.44/30 is directly connected, Port-channel11
L        77.66.55.45/32 is directly connected, Port-channel11
C        77.66.55.84/30 is directly connected, Port-channel12
L        77.66.55.85/32 is directly connected, Port-channel12
      172.16.0.0/19 is subnetted, 1 subnets
B        172.16.0.0 [20/0] via 77.66.55.46, 00:00:11
      192.168.0.0/32 is subnetted, 3 subnets
B        192.168.0.11 [20/0] via 77.66.55.46, 00:00:11
B        192.168.0.22 [20/0] via 77.66.55.46, 00:00:11
C        192.168.0.77 is directly connected, Loopback0
      192.168.10.0/30 is subnetted, 1 subnets
B        192.168.10.8 [20/0] via 77.66.55.46, 00:00:11
      192.168.20.0/30 is subnetted, 1 subnets
B        192.168.20.20 [20/0] via 77.66.55.46, 00:00:11
      192.168.200.0/32 is subnetted, 2 subnets
B        192.168.200.11 [20/0] via 77.66.55.46, 00:00:11
B        192.168.200.22 [20/0] via 77.66.55.46, 00:00:11
```

- Tato adresa by měla být známa prostřednictvím protokolu BGP na PE směrovači. Zkontrolujeme tedy, zda má MSN pár ve směrovací tabulce údaj o této adrese.

MSN_A

```
MSN_A#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C        10.10.10.0/24 is directly connected, GigabitEthernet1
L        10.10.10.9/32 is directly connected, GigabitEthernet1
C        10.200.10.28/30 is directly connected, BDI90
L        10.200.10.30/32 is directly connected, BDI90
C        10.200.10.48/30 is directly connected, BDI91
L        10.200.10.50/32 is directly connected, BDI91
      77.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        77.66.55.44/30 is directly connected, Port-channel11
L        77.66.55.46/32 is directly connected, Port-channel11
B        77.66.55.84/30 [200/0] via 192.168.200.22, 08:57:31
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
S        172.16.0.0/19 is directly connected, Null0
O        172.16.13.16/28 [110/401] via 10.200.10.49, 08:57:14, BDI91
      172.17.0.0/29 is subnetted, 1 subnets
O        172.17.13.32 [110/401] via 10.200.10.49, 08:57:14, BDI91
      192.168.0.0/32 is subnetted, 3 subnets
C        192.168.0.11 is directly connected, Loopback0
O        192.168.0.22 [110/101] via 10.200.10.29, 09:01:30, BDI90
O        192.168.0.99 [110/401] via 10.200.10.49, 08:57:14, BDI91
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.8/30 is directly connected, BDI1071
L        192.168.10.9/32 is directly connected, BDI1071
      192.168.20.0/30 is subnetted, 1 subnets
O        192.168.20.20 [110/400] via 10.200.10.49, 08:57:14, BDI91
      192.168.200.0/32 is subnetted, 2 subnets
C        192.168.200.11 is directly connected, Loopback20
O        192.168.200.22 [110/101] via 10.200.10.29, 09:01:30, BDI90
```

V tabulce lze vidět, že síť 172.17.0.0/29 je známa pomocí OPSF. Stejný výsledek dostaneme i u MSN_B. Směrovače síť poznají a z toho vyplývá, že problém spočívá v redistribuci a nikoliv na *Backhaul* lince.

- Je zapotřebí zjistit, jak celý proces redistribuce pro OAM IPv4 adresy funguje. Vyhledáme si tedy, kde je síť 172.17.0.0/29 použita.

MSN_A

```
MSN_A#sh run | i 172.17
 permit 172.17.8.0 0.0.7.255
```

Výsledkem je zjištění, že síť 172.17.0.0/29 je součástí `access-list V4-OAM-ROUTES`.

- Při dalším hledání pomocí `show run` lze vidět, že je součástí `route-map OAM-IPV4`

MSN_A

```
MSN_A# sh run
route-map OAM-IPV4 permit 10
 description Redistributed OAM IPv4 routes
 match ip address V4-OAM-ROUTES
 set community 64600:3000
```

- Zde se dostáváme do konfigurace samotného BGP, kde v adresní rodině pro IPv4 můžeme vidět nastavení redistribuce pro OAM-IPV4.

MSN_A

```
MSN_A#sh run | sec address-family ipv4
 address-family ipv4
  network 172.16.0.0 mask 255.255.224.0 route-map NODEB-IPV4-SUMMARY
  redistribute connected route-map CONNECTED-IPV4
  redistribute static route-map OAM-IPV4
  neighbor IPV4-IBGP-MATE-PEER-GRP send-community
  neighbor IPV4-IBGP-MATE-PEER-GRP next-hop-self
  neighbor IPV4-IBGP-MATE-PEER-GRP route-map MATE-MSN out
  neighbor 77.66.55.45 activate
  neighbor 77.66.55.45 send-community
  neighbor 77.66.55.45 route-map ROUTE_INCOMING in
  neighbor 192.168.200.22 activate
```

  V tomto výpisu je nutné si všimnout příkazu `redistribute static`. Z předchozích výpisů víme, že síť 172.17.0.0/29 není známa skrze statickou cestu, ale pomocí OSPFv2. Redistribuce tak nefunguje.

- Opravu provedeme tak, že ověříme, do jakého proces ID u OSPF síť 172.17.0.0/29 spadá a následně vložíme celý příkaz `redistribute ospf 1 route-map OAM-IPV4` do adresní rodiny u BGP. Tento krok je potřeba provést na obou MSN zařízeních.

MSN_A i MSN_B

```
router bgp 64600
address-family ipv4
no redistribute static route-map OAM-IPV4
no redistribute static
redistribute ospf 1 route-map OAM-IPV4
```

- Na PE je nutné aktualizovat protokol BGP příkazem `clear ip bgp *`. Síť 172.17.0.0/29 pak bude součástí směrovací tabulky.

PE

```
PE#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B        10.10.10.0/24 [20/0] via 77.66.55.46, 00:02:34
B        10.200.10.28/30 [20/0] via 77.66.55.46, 00:02:34
B        10.200.10.48/30 [20/0] via 77.66.55.46, 00:02:34
      77.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        77.66.55.44/30 is directly connected, Port-channel11
L        77.66.55.45/32 is directly connected, Port-channel11
C        77.66.55.84/30 is directly connected, Port-channel12
L        77.66.55.85/32 is directly connected, Port-channel12
      172.16.0.0/19 is subnetted, 1 subnets
B        172.16.0.0 [20/0] via 77.66.55.46, 00:02:34
      172.17.0.0/29 is subnetted, 1 subnets
B        172.17.13.32 [20/0] via 77.66.55.46, 00:02:34
      192.168.0.0/32 is subnetted, 3 subnets
B        192.168.0.11 [20/0] via 77.66.55.46, 00:02:34
B        192.168.0.22 [20/0] via 77.66.55.46, 00:02:34
C        192.168.0.77 is directly connected, Loopback0
      192.168.10.0/30 is subnetted, 1 subnets
B        192.168.10.8 [20/0] via 77.66.55.46, 00:02:34
      192.168.20.0/30 is subnetted, 1 subnets
B        192.168.20.20 [20/0] via 77.66.55.46, 00:02:34
      192.168.200.0/32 is subnetted, 2 subnets
B        192.168.200.11 [20/0] via 77.66.55.46, 00:02:34
B        192.168.200.22 [20/0] via 77.66.55.46, 00:02:34
```

6 Závěr

Cílem této práce bylo nastudovat problematiku protokolů sady TCP/IP, směrovacích protokolů a fungování transportní části mobilních sítí. Dále si práce kladla za cíl porovnat nejvhodnější možnosti emulace těchto sítí a vytvořit dva simulační scénáře. Veškeré cíle této diplomové práce byly splněny. Seznámili jsme se s potřebným teoretickým základem, prostudovali a vybrali vhodné emulační programy a vytvořili dva komplexní scénáře.

V první kapitole jsme uvedli základní informace o struktuře mobilních sítí 3G a 4G s důrazem na zařízení, ze kterých je síť tvořena. Cílem této úvodní kapitoly bylo seznámení se s významem transportní části v mobilních sítích.

Další část byla věnována protokolům běžně používaným v transportní síti. Byly zde uvedeny teoretické základy pro následující praktickou část. Rozebrali jsme podrobně protokoly IPv4 (Internet Protocol v4), IPv6 (Internet Protocol v6) a směrovací strategie. Mezi směrovací strategie jsme zahrnuli statické směrování, OSPFv2 (Open Shortest Path First v2), OSPFv3 (Open Shortest Path First v3) i BGP (Border Gateway Protocol). Dále byly popsány protokoly BFD (Bidirectional Forwarding Detection), SNMP (Simple Network Managment Protocol) a také QoS (Quality of service), s ohledem na jejich využití v transportní části mobilní sítě.

Ve třetí kapitole byla řešena otázka emulace transportní sítě z pohledu softwarových a hardwarových zdrojů. Uvedli jsme zde specifikace vlastního zapojení společně s instalací hyperviseru ESXi 6.0. Při instalaci došlo k problémům s rozpoznáním síťové karty použité na serveru. ESXi instalační obraz musel být ručně upraven tak, aby síťová karta byla rozeznána.

Na tento krok jsme navázali instalací emulačních programů EVE-NG (Emulated Virtual Environment – Next Generation) a GNS3 (Graphical Network Simulator-3), popisem grafického prostředí a postupem při nahrávání emulátorů. Kapitola byla doplněna o popis a instalaci softwarů Ostinato, Wireshark a PowerSNMP. Tato kapitola může sloužit jako návod pro tvorbu vlastní virtuální laboratoře.

V kapitole číslo čtyři jsme se věnovali srovnání emulačních programů EVE-NG a GNS3. Byly zjištěny rozdíly týkající se samotné instalace, nahrávání emulátorů, GUI (Graphical User Interface), atd. Nalezené rozdíly nebyly příliš výrazné a nebrání uživateli v preferování jednoho či druhého programu a co je důležitější, nebrání zprovoznění transportní sítě.

Provedli jsme proto test hardwarové náročnosti emulačních programů na serveru ESXi. Testováno bylo zatížení CPU, doba načtení obrazů a zatížení paměti RAM. Testy prokázaly, že se emulační programy neliší ani v náročnosti na zatížení CPU, ani v době načtení obrazů. Rozdíl nastal až v zatížení RAM paměti. Při úvodním počítání heše a načítání spotřeboval emulační program EVE-NG 15 GB paměti

RAM, zatímco GNS3 spotřeboval pouze 13,75 GB. V ustáleném stavu naopak EVE-NG spotřeboval o 2 GB méně paměti než GNS3.

Jedním z možných vysvětlení je například to, že EVE-NG spouští vybrané emulátory odlišným způsobem. Je také možné, že GNS3 je hůře optimalizovaný pro zvolenou výchozí topologii. Naměřené výsledky se však vztahují na konkrétní výchozí topologii a při odlišné topologii a s použitím jiných zařízení by výsledky mohly tyto hypotézy vyvrátit.

Námi provedené testy nicméně potvrdily, že oba emulační programy jsou vhodné pro simulaci mobilní transportní sítě a splňují potřebné parametry pro koncového uživatele.

Jak vyplývá z detailních popisů Scénáře 1 a 2 v páté kapitole, simulace v obou programech přesáhla svým rozsahem přibližný limit 2 hodin, jak bylo stanoveno v zadání. V případě potřeby je však možné scénáře na tuto dobu zkrátit, aniž by to mělo nějaký výrazný dopad.

Scénář 1 - „Konfigurace transportní sítě“ se skládá ze sedmi základních a dvou dodatečných kroků. I bez těchto dodatečných kroků je však zajištěna celková funkčnost transportní sítě. Scénář 1 byl vytvořen tak, aby byl uživatel schopen sám celou transportní síť nastavit. Během konfigurace a při ověřování nastavení je možné si uvědomit mnohé souvislosti v celé mobilní transportní síti, stejně jako při plnění samostatných úkolů a při zodpovídání kontrolních otázek. Uživatel tak jednoduše pochopí význam všech kroků, ze kterých se scénář skládá.

Scénář 2 - „Časté chyby v transportní síti“ navazuje na předchozí scénář a jeho podstata spočívá v hledání chyb v předem vytvořených chybných konfiguracích směrovačů. Zvolení daných chyb má konkrétní důvod, a sice že se jedná o často se vyskytující problémy v transportní síti. Díky tomuto scénáři je možné analyzovat dopad jednotlivých způsobů konfigurace na celou mobilní transportní síť. Vzorová řešení jsou uvedena na konci Kapitoly 5.2.

Díky námi vytvořeným scénářům se prokázalo, že je možné simulovat v omezeném rozsahu prostředí mobilní transportní sítě. Scénáře a celé laboratorní prostředí tak mohou napomoci odborníkům, studentům nebo zájemcům o problematiku mobilních transportních sítí v pochopení celkového fungování těchto sítí.

Je samozřejmě možné a žádoucí dále rozšiřovat stávající virtuální laboratoř a testovat nové způsoby konfigurace, aby simulační scénáře mobilní transportní sítě držely krok se stávajícími trendy. Zcela nové otázky pak otevírá přechod na mobilní sítě nové generace 5G, které spolu nesou příchod větší virtualizace a postupnou migraci na ještě výkonnější zařízení.

Dílčí výsledky této práce byly oceněny na konferenci EEICT (Electrical Engineering, Information and Communication Technologies) konané 25.dubna 2019 v Brně, kde v kategorii *Komunikační technologie a informační bezpečnost* získaly 3. místo.

Literatura

[1] *About 3GPP Home* [online]. [cit. 2016-22-08]. www.3gpp.org. Dostupné z URL: <http://www.3gpp.org/about-3gpp/about-3gpp>.

[2] PROKOPEC J. *Systémy mobilních komunikací: sítě pro mobilní datové služby.* [online]. [cit. 2018-08-10] Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2012. ISBN 978-80-214-4498-0.

[3] PRAVDA I. *Nové trendy v elektronických komunikacích Mobilní a bezdrátové sítě.* [online]. [cit. 2018-08-10] Praha: Českе vysoké učení technické v Praze. Dostupné z URL:<http://publi.cz/books/236/Cover.html>.

[4] *Digi.ctu.cz: Veřejné širokopásmové mobilní sítě.* [online]. [cit. 2018-08-14]. Dostupné z URL: <http://digi.ctu.cz/>.

[5] NOVOTNÝ V. *Mobilní komunikační sítě a služby v all-IP prostředí pro integrovanou výuku VUT a VŠB-TUO.* [online]. [cit. 2018-08-10]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. ISBN 978-80-214-5129-2.

[6] DeBalko, G. A. *Unbundling device and method for connecting a competing local exchange carrier (CLEC) to the subscriber loop of a local exchange carrier (LEC).* [online].[cit. 2019-03-16]. U.S. Patent č. 6,282,277. 28 Aug. 2001. <{https://patents.google.com/patent/US6282277B1/en}.>.

[7] *Emulátor.* [online]. San Francisco (CA): Wikimedia Foundation, 2001-[cit. 2018-07-26]. Wikipedia: the free encyclopedia. Dostupné z URL: <https://cs.wikipedia.org/wiki/Emul%C3%A1tor>.

[8] *README.hypervisor* [online]. San Francisco, 2007-[cit. 2018-07-26].Github.com Dostupné z URL: <https://github.com/GNS3/dynamips/blob/master/README.hypervisor>.

[9] *HowTo-add-cisco-iou-iol.* [online]. [cit. 2018-07-28].Eve-ng.net Dostupné z URL: <http://www.eve-ng.net/documentation/howto-s/62-howto-add-cisco-iou-iol>.

[10] *Main_page* [online]. [cit. 2018-07-28]. Wiki.qemu.org Dostupné z URL: <https://wiki.qemu.org/Main_Page>.

[11] NEUMAN, C. *The Book of GNS3: Build Virtual Network Labs Using Cisco, Juniper, and More.* No Starch Press, 2018. ISBN 978-15-932-7695-9.

[12] DZERKALS, U. a DOE M., ed., LIM, Ch. ed. *EVE-NG Professional Cookbook.* [online]. [cit. 2018-08-14]. 1. vydání. Dostupné z URL: <http://www.eve-ng.net/images/EVE-COOK-BOOK-1.0.pdf>.

[13] BOMBAL, D. a DUPONCHELLE, J. *Getting Started with GNS3.* [online]. [cit. 2018-07-31]. Gns3.com Dostupné z URL: <https://docs.gns3.com/1PvtRW5eAb8RJZ11maEYD9_aLY8kkdhgaMB0wPCz8a38/index.html>.

[14] GOLDBERG, R. *Architectural Principles for Virtual Computer Systems.* [online]. [cit. 2018-07-25]. Harvard University, Cambridge, 1973 Dostupné z URL: <http://www.dtic.mil/dtic/tr/fulltext/u2/772809.pdf.>.

[15] *Chapter: Using Cisco IOS Software.* [online]. [cit. 2018-08-01]. Cisco.com Dostupné z URL: <https://www.cisco.com/c/en/us/td/docs/ios/12_2/interface/configuration/guide/finter_c/icfusing.html#wp1005903.>.

[16] *White Paper: Cisco IOS and NX-OS Software Reference Guide.* [online]. [cit. 2018-08-01]. Cisco.com Dostupné z URL: <https://www.cisco.com/c/en/us/about/security-center/ios-nx-os-reference-guide.html.>.

[17] *Introduction to Cisco IOS XE.* [online]. [cit. 2018-08-01]. Networklessons.com Dostupné z URL: <hhttps://networklessons.com/cisco/ccie-routing-switching-written/introduction-cisco-ios-xe/.>.

[18] *Wireshark* [online]. [cit. 2018-07-29]. wikipedia.org Dostupné z URL: <https://cs.wikipedia.org/wiki/Wireshark>.

[19] OCHANG A., P. a IRVING J., P. *Evolutionary Analysis of GSM, UMTS and LTE Mobile Network Architectures.* [online]. [cit. 2018-08-28]. Federal University Lafia, Nasarawa State, Nigeria; University of Sunderland, Sunderland, United Kingdom EISSN 2392-2192 Dostupné z URL: <http://www.worldscientificnews.com/wp-content/uploads/2016/01/WSN-54-2016-27-39.pdf>.

[20] HALONEN, T., ROMERO, J., MELERO, J. *EGSM, GPRS and EDGE Performance.* Chichester: JohnWilley & sons, 2006. 407. ISBN 0-470-01684-9.

[21] HANUS S. *Nové technologie mobilních komunikací pro integrovanou výuku VUT a VŠB-TUO.* [online]. [cit. 2018-08-31] Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. ISBN 978-80-214-4824-7.

[22] *Jádro sítě GPRS.* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-09-23]. Wikipedia: the free encyclopedia. Dostupné z URL: <https://cs.wikipedia.org/wiki/J%C3%A1dro_s%C3%ADt%C4%9B_GPRS>.

[23] FOROUZAN, Behrouz A. *TCP/IP protocol suite.*.4th ed. Boston: McGraw-Hill Higher Education, 2010, xxxv, 979 s. ISBN 978-0-07-337604-2.

[24] JEŘÁBEK, J. *Komunikační technologie.* Skriptum FEKT Vysoké učení technické v Brně 2018, s. 1-172.

[25] *TCP/IP - adresy, masky, subnety a výpočty.* [online]. [cit. 2018-10-05]. samuraj-cz.com Dostupné z URL: <www.samuraj-cz.com/clanek/tcpip-adresy-masky-subnety-a-vypocty/.>.

[26] JEŘÁBEK, J. *Pokročilé komunikační techniky.* Skriptum FEKT Vysoké učení technické v Brně 2018, s. 1-174.

[27] MOLNÁR, k. *Hardware počítačových sítí.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. ISBN 978-80-214-4449-2.

[28] HALABI, S. *OSPF design guide.* Cisco Systems Network Supported Accounts, 1996.

[29] *OSPF Not-So-Stubby Area (NSSA).* [online]. [cit. 2018-10-16]. www.cisco.com, 2005 Dostupné z URL: <https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/6208-nssa.html/.>.

[30] COLTUN, R. *OSPF for IPv6.* [online]. et al. RFC 5340, IETF. July, 2008, 24. Dostupné z URL: <https://tools.ietf.org/html/rfc5340.>.

[31] ČEPA, L.; HÁJEK, J. *Směrovaní a směrovací protokoly v IP síti verze 6.* [online]. Téma: Aplikace, sítě a služby,2010. České vysoké učení technické v Praze, FEL <http://access.feld.cvut.cz/rservice.php?akce=tisk&cisloclanku=2010010002.>.

[32] GRYGÁREK P. *Směrovací protokol BGP.* [online]. [cit. 2018-11-10]. VŠB-TU Ostrava, Fakulta elektrotechniky a informatiky <http://www.cs.vsb.cz/grygarek/SPS/lect/BGP/BGP.html.>.

[33] KATZ, D.; WARD, D. *Bidirectional forwarding detection (BFD).* [online]. et al. RFC 5880, IETF. June 2010. Dostupné z URL: <https://www.rfc-editor.org/rfc/pdfrfc/rfc5880.txt.pdf.>.

[34] *Configuring Bridge Domain Interfaces.* [online]. [cit. 2018-11-04]. www.cisco.com,November 24, 2010 Dostupné z URL: <https://www.cisco.com/c/en/us/td/docs/routers/asr1000/configuration/guide/chassis/bdi.html.>.

[35] LEDVINA, J. *QoS v datových sítích, IntServ a DiffServ.* [online]. Přednáška předmětu Počítačové sítě, 2007. Západočeská univerzita v Plzni, Katedra informatiky a výpočetní techniky <http://www.kiv.zcu.cz/~ledvina/vyuka/PSI/lekce/QOS_text.pdf.>.

[36] *An Architecture for Differentiated Services.* [online]. IETF NETWORK WORKING GROUP, et al. RFC 2475 IETF, 1998. <https://www.ietf.org/rfc/rfc2475.txt.>.

Seznam symbolů a zkratek

3GPP	The 3rd Generation Partnership Project
ARIB	Association of Radio Industries and Businesses
ATIS	Alliance for Telecommunications Industry Solutions
CCSA	China Communications Standards Association
ETSI	European Telecommunications Standards Institute
TSDSI	Telecommunications Standards Development Society, India
TTA	Telecommunications Technology Association of Korea
TTC	Telecommunication Technology Committee
RAN	Radio Access Network
UE	User Equipment
DU	Digital Unit
RNC	Radio Network Controller
eNodeB	evolved NodeB
UMTS	Universal Mobile Telecommunications System
UTRAN	Universal Terrestrial Radio Access Network
LTE(-A)	The Long Term Evolution (- Advanced)
EUTRAN	Evolved Universal Terrestrial Radio Access Network
FDD	Frequency Division Duplex
TDD	Time Division Duplex
WCDMA	Wideband Code Division Multiple Access
ČTU	Český telekomunikační úřad
OFDMA	Orthogonal frequency division multiple access
OFDM	Orthogonal Frequency Division Multiplexing
SC-FDMA	Single-carrier frequency division multiple access
TDMA	Time-division multiple access
FDMA	Frequency-division multiple access
MIMO	Multiple Input Multiple Output
QoS	Quality of service
SIAD	Smart Integrated Access Device
LEC	Local Exchange Carrier
NTE	Network Terminal Endpoint
MME	Mobility Management Entity
MSC	Mobile Switching Centre
MTSO	Mobile Telephone Switching Office
EVE-NG	Emulated Virtual Environment – Next Generation
GNS3	Graphical Network Simulator-3
PE	Provider Edge

CS	Circuit Switched - přepínání okruhů
PS	Packet Switched - přepínání paketů
MSC	Mobile Switching Center
VLR	Visitor Location Register
HLR	Home Location Register
GMSC	Gateway Mobile Switching Center
PSTN	Public Switched Telephone Network
SGSN	Serving GPRS Support Node
GPRS	General Packet Radio Service
GGSN	Gateway GPRS Support Node
EPC	Evolved Packet Core
SGW	Serving Gateway
HSS	Home Subscriber Server
PGW	Packet Gateway
MSN	Multi-Service Node
HSDPA	High-Speed Downlink Packet Access
HSUPA	High-Speed Uplink Packet Access
CBB	Core Backbone - páteřní síť
TCP	Transmission Control Protocol
CIDR	Classless Inter–Domain Routing
UDP	User Datagram Protocol
NAT-PT	Network Address Translator - Protocol Translator
OSPF	Open Shortest Path First
IGP	Interior Gateway Protocol
ABR	Area Border Router
ASBR	Autonomous System Border Router
IR	Internal Router
DR	Designated Router
BDR	Backup Designated Router
AS	Autonomous System
BGP	Border Gateway Protocol
eBGP	External Border Gateway Protocol
iBGP	Internal Border Gateway Protocol
EGP	Exterior Gateway Protocol
NLRI	Network Layer Reachability Information
BFD	Bidirectional Forwarding Detection
SNMP	Simple Network Managment Protocol
SMI	Structure of Management Information
MIB	Management Information Base

PAT	Port Address Translation
GUI	Graphical User Interface
BDI	Bridge Domain Interface
SVI	Switch Virtual Interface
DSCP	Differentiated Services Codepoint
LACP	Link Aggregation Control Protocol

Seznam příloh

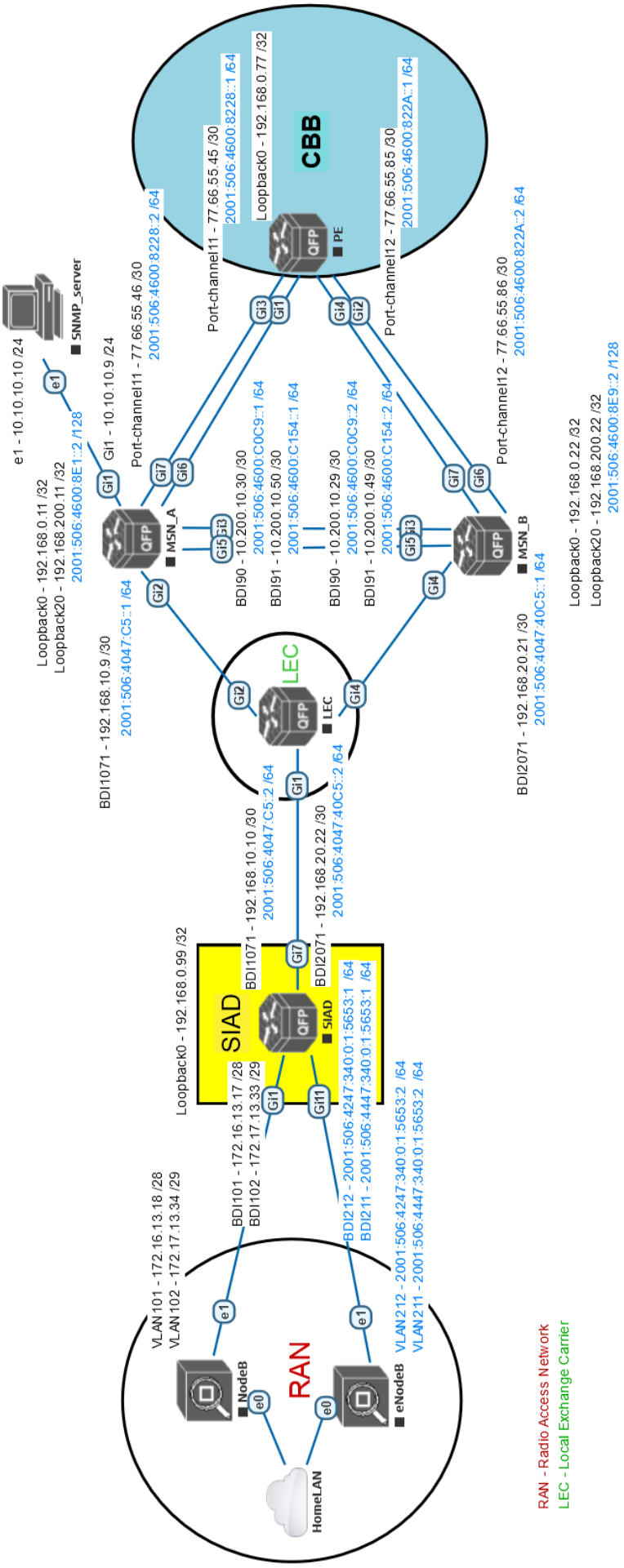
A Příloha

Tab. A.1: Seznam logických a fyzických rozhraní s přidělenými IP adresami

Zařízení	Logické rozhraní	Fyzické rozhraní	IPv4 adresa	IPv6 adresa	Popis
NodeB	VLAN101	eth1	172.16.13.18/28	-	Link to SIAD Bearer 3G
NodeB	VLAN102	eth1	172.17.13.34/29	-	Link to SIAD OAM 3G
eNodeB	VLAN211	eth1	-	2001:506:4447:340:0:1:5653:2/64	Link to SIAD OAM 4G
eNodeB	VLAN212	eth1	-	2001:506:4247:340:0:1:5653:2/64	Link to SIAD Bearer 4G
SIAD	Loopback0	-	192.168.0.99/32	-	OSPF RID
SIAD	BDI101	Gi1	172.16.13.17/28	-	Link to NodeB Bearer
SIAD	BDI102	Gi1	172.17.13.33/29	-	Link to NodeB OAM
SIAD	BDI211	Gi11	-	2001:506:4447:340:0:1:5653:1/64	Link to eNodeB OAM
SIAD	BDI212	Gi11	-	2001:506:4247:340:0:1:5653:1/64	Link to eNodeB Bearer
SIAD	BDI1071	Gi7	192.168.10.10/30	2001:506:4047:C5::2/64	Primary Vlan to MSN_A
SIAD	BDI2071	Gi7	192.168.20.22/30	2001:506:4047:40C5::2/64	Secondary Vlan to MSN_B
MSN_A	Loopback0	-	192.168.0.11/32	-	OSPF RID
MSN_A	Loopback20	-	192.168.200.11/32	2001:506:4600:8E1::2/128	CORE-OSPF-ROUTER-ID
MSN_A	BDI90/Port-channel1	Gi3,Gi5	10.200.10.30/30	2001:506:4600:C0C9::1/64	OSPF interchassis link Area 10+0
MSN_A	BDI91/Port-channel1	Gi3,Gi5	10.200.10.50/30	2001:506:4600:C154::1/64	OSPF Inter-chassis link Area 10+10
MSN_A	BDI1071	Gi2	192.168.10.9/30	2001:506:4047:C5::1/64	Backhaul Primary interface to SIAD
MSN_A	Port-channel11	Gi6,Gi7	77.66.55.46/30	2001:506:4600:8228::2/64	LACP Link between MSN_A and PE
MSN_B	Loopback0	-	192.168.0.22/32	-	OSPF RID
MSN_B	Loopback20	-	192.168.200.22/32	2001:506:4600:8E9::2/128	CORE-OSPF-ROUTER-ID
MSN_B	BDI90/Port-channel1	Gi3,Gi5	10.200.10.29/30	2001:506:4600:C0C9::2/64	OSPF interchassis link Area 0
MSN_B	BDI91/Port-channel1	Gi3,Gi5	10.200.10.49/30	2001:506:4600:C154::2/64	OSPF Inter-chassis link Area 10
MSN_B	BDI2071	Gi4	192.168.20.21/30	2001:506:4047:40C5::1/64	Backhaul Secondary interface to SIAD
MSN_B	Port-channel12	Gi6,Gi7	77.66.55.86/30	2001:506:4600:822A::2/64	LACP Link between MSN_B and PE
PE	Loopback0	-	192.168.0.77/32	-	Router ID
PE	Port-channel11	Gi1,Gi3	77.66.55.45/30	2001:506:4600:8228::1/64	LACP Link between PE and MSN_A
PE	Port-channel12	Gi2,Gi4	77.66.55.85/30	2001:506:4600:822A::1/64	LACP Link between PE and MSN_B

Tab. A.2: *BGP community* odpovídající přiřazeným jmenným standardům

Community-list standard (jméno)	Community řetězec
SHORTHAUL-ALL	64600:1000
OAM-IPV4-ALL	64600:3000
PE-DEFAULT-V4	64600:7000
PE-DEFAULT-V6	64600:7100
CONNECTED-IPV4-ALL	64600:2000



Obr. A.1: Kompletní síť s popisem portů a přiřazených IPv4/IPv6 adres pro účely Scénářů 1 a 2

B Konfigurace zařízení pro Scénář 1

Níže jsou uvedeny finální podoby konfigurací jednotlivých zařízení pro Scénář 1. Zároveň tato konfigurace může posloužit jako kontrola pro Scénář 2, který ze Scénáře 1 vychází. Konfigurace Scénáře 2 se nachází na přiloženém DVD.

B.1 SIAD

```
SIAD#sh run
Building configuration...

Current configuration : 7470 bytes
!
! Last configuration change at 12:54:23 UTC Fri Apr 19 2019
!
version 15.6
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname SIAD
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
!
!
ipv6 unicast-routing
!
subscriber templating
!
!
!
multilink bundle-name authenticated
!
!
license udi pid CSR1000V sn 9HAD6S7WETH
!
spanning-tree extend system-id
!
!
redundancy
!
!
class-map match-any Control
  description "Mobility␣signaling␣and␣Network␣Protocol␣traffic"
 match dscp cs6
 match dscp cs7
 match dscp cs4
class-map match-all qos-group-cos1
 match qos-group 5
```

```
class-map match-all qos-group-cos3
 match qos-group 2
class-map match-all qos-group-cos2
 match qos-group 3
class-map match-any COS3
  description "LTE data traffic"
 match precedence 2
class-map match-any COS2
  description "UMTS R99, HS traffic"
 match precedence 3
class-map match-any COS1
  description "Real Time applications (3G Voice, VoLTE)"
 match dscp cs5
 match dscp ef
class-map match-any qos-group-control
 match qos-group 6
!
policy-map SIAD_Output_Policy_Child_1G
 description "SIAD backhaul egress child policy for 1G"
 class qos-group-cos1
  priority percent 50 62500
 class qos-group-control
  bandwidth remaining percent 50
 class qos-group-cos2
  bandwidth remaining percent 20
 class qos-group-cos3
  bandwidth remaining percent 20
 class class-default
  bandwidth remaining percent 10
policy-map SIAD_Output_Policy_Child_100M
 description "SIAD backhaul egress child policy for 100M"
 class qos-group-cos1
  priority percent 50 6250
 class qos-group-control
  bandwidth remaining percent 50
 class qos-group-cos2
  bandwidth remaining percent 20
 class qos-group-cos3
  bandwidth remaining percent 20
 class class-default
  bandwidth remaining percent 10
policy-map SIAD_QOS_Policy_ingress
 description "SIAD ingress policy for short-haul and backhaul"
 class Control
  set qos-group 6
 class COS1
  set qos-group 5
 class COS2
  set qos-group 3
 class COS3
  set qos-group 2
policy-map SIAD_Parent_Policy_Egress_800M
 description "SIAD backhaul Egress parent policy for 800M"
 class class-default
  shape average 760000000 3040000
policy-map SIAD_Output_Policy_Child_800M
 description "SIAD backhaul egress child policy for 800M"
 class qos-group-cos1
  priority percent 50 50000
```

```
 class qos-group-control
  bandwidth remaining percent 50
 class qos-group-cos2
  bandwidth remaining percent 20
 class qos-group-cos3
  bandwidth remaining percent 20
 class class-default
  bandwidth remaining percent 10
!
!
interface Loopback0
 description OSPF RID
 ip address 192.168.0.99 255.255.255.255
!
interface GigabitEthernet1
 no ip address
 negotiation auto
 service-policy input SIAD_QOS_Policy_ingress
 service-policy output SIAD_Output_Policy_Child_100M
 service instance 101 ethernet
  encapsulation dot1q 101
  rewrite ingress tag pop 1 symmetric
  bridge-domain 101
 !
 service instance 102 ethernet
  encapsulation dot1q 102
  rewrite ingress tag pop 1 symmetric
  bridge-domain 102
 !
!
interface GigabitEthernet2
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet3
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet4
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet5
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet6
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet7
 description Backhaul Link to MSN_A,MSN_B
 no ip address
 negotiation auto
```

```
 service-policy input SIAD_QOS_Policy_ingress
 service-policy output SIAD_Parent_Policy_Egress_800M
 service instance 1071 ethernet
  encapsulation dot1q 1071
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1071
 !
 service instance 2071 ethernet
  encapsulation dot1q 2071
  rewrite ingress tag pop 1 symmetric
  bridge-domain 2071
 !
!
interface GigabitEthernet8
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet9
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet10
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet11
 description Link to eNodeB
 no ip address
 load-interval 30
 negotiation auto
 service-policy input SIAD_QOS_Policy_ingress
 service-policy output SIAD_Output_Policy_Child_1G
 service instance 211 ethernet
  encapsulation dot1q 211
  rewrite ingress tag pop 1 symmetric
  bridge-domain 211
 !
 service instance 212 ethernet
  encapsulation dot1q 212
  rewrite ingress tag pop 1 symmetric
  bridge-domain 212
 !
!
interface GigabitEthernet12
 no ip address
 shutdown
 negotiation auto
!
interface BDI101
 description Link to NodeB Bearer
 ip address 172.16.13.17 255.255.255.240
!
interface BDI102
 description Link to NodeB OAM
 ip address 172.17.13.33 255.255.255.248
!
```

```
interface BDI211
 description Link to eNodeB OAM
 no ip address
 no ip redirects
 load-interval 30
 ipv6 address 2001:506:4447:340:0:1:5653:1/64
!
interface BDI212
 description Link to eNodeB Bearer
 no ip address
 no ip redirects
 load-interval 30
 ipv6 address 2001:506:4247:340:0:1:5653:1/64
!
interface BDI1071
 description - Primary Vlan to MSN_A
 ip address 192.168.10.10 255.255.255.252
 no ip redirects
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf bfd strict-mode
 load-interval 30
 encapsulation dot1Q 1071
 ipv6 address 2001:506:4047:C5::2/64
 bfd interval 500 min_rx 500 multiplier 3
 no bfd echo
!
interface BDI2071
 description - Secondary Vlan to MSN_B
 ip address 192.168.20.22 255.255.255.252
 no ip redirects
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 300
 load-interval 30
 encapsulation dot1Q 2071
 ipv6 address 2001:506:4047:40C5::2/64
!
router ospf 10
 router-id 192.168.0.99
 area 10 nssa
 passive-interface BDI101
 passive-interface BDI102
 network 172.16.13.16 0.0.0.15 area 10
 network 172.17.13.32 0.0.0.7 area 10
 network 192.168.0.99 0.0.0.0 area 10
 network 192.168.10.8 0.0.0.3 area 10
 network 192.168.20.20 0.0.0.3 area 10
 bfd all-interfaces
!
!
virtual-service csr_mgmt
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
```

```
ip route static bfd BDI1071 192.168.10.9
ip route 0.0.0.0 0.0.0.0 BDI1071 192.168.10.9 name PRIMARY_LINK
ip route 0.0.0.0 0.0.0.0 BDI2071 192.168.20.21 10 name SECONDARY_Link
ip route 172.16.13.16 255.255.255.240 Null0 name SIAD_NB_Interfaces
ip route 172.17.13.32 255.255.255.248 Null0 name SIAD_NB_Interfaces
!
!
ipv6 route static bfd BDI1071 2001:506:4047:C5::1
ipv6 route 2001:506:4247:340::/61 Null0
ipv6 route 2001:506:4447:340::/61 Null0
ipv6 route ::/0 BDI2071 2001:506:4047:40C5::1 10 name Secondary_Link
ipv6 route ::/0 BDI1071 2001:506:4047:C5::1 name Primary_Link
!
snmp-server enable traps tty
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors bad-packet
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps config-copy
snmp-server enable traps syslog
snmp-server enable traps bfd
snmp-server host 10.10.10.10 version 2c v2c
!
!
control-plane
!
!
line con 0
 stopbits 1
line vty 0
 login
line vty 1
 login
 length 0
line vty 2 4
 login
!
!
end
```

B.2 LEC

```
LEC#sh run
Building configuration...

Current configuration : 1760 bytes
!
! Last configuration change at 18:18:59 UTC Tue Apr 16 2019
!
version 15.6
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname LEC
!
boot-start-marker
```

```
boot-end-marker
!
!
no aaa new-model
!
!
no ip domain lookup
!
!
subscriber templating
!
!
multilink bundle-name authenticated
!
!
license udi pid CSR1000V sn 9MIEMBEH981
!
spanning-tree extend system-id
!
!
redundancy
!
!
interface GigabitEthernet1
 no ip address
 negotiation auto
 service instance 1071 ethernet
  encapsulation dot1q 1071
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1071
 !
 service instance 2071 ethernet
  encapsulation dot1q 2071
  rewrite ingress tag pop 1 symmetric
  bridge-domain 2071
 !
!
interface GigabitEthernet2
 no ip address
 negotiation auto
 cdp enable
 service instance 1071 ethernet
  encapsulation dot1q 1071
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1071
 !
!
interface GigabitEthernet3
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet4
 no ip address
 negotiation auto
 service instance 2071 ethernet
  encapsulation dot1q 2071
  rewrite ingress tag pop 1 symmetric
  bridge-domain 2071
```

```
 !
!
interface GigabitEthernet5
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet6
 no ip address
 shutdown
 negotiation auto
!
!
virtual-service csr_mgmt
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
control-plane
!
line con 0
 logging synchronous
 stopbits 1
line vty 0
 login
line vty 1
 login
 length 0
line vty 2 4
 login
!
!
end
```

B.3 MSN A

```
MSN_A#sh run
Building configuration...

Current configuration : 11109 bytes
!
! Last configuration change at 13:36:55 UTC Fri Apr 19 2019
!
version 15.6
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname MSN_A
!
boot-start-marker
boot-end-marker
!
!
```

```
no aaa new-model
!
ipv6 unicast-routing
ipv6 multicast rpf use-bgp
!
!
subscriber templating
!
!
multilink bundle-name authenticated
!
!
license udi pid CSR1000V sn 9FDBC8JZELD
!
spanning-tree extend system-id
!
!
redundancy
bfd-template single-hop msn-bfd-template
 interval min-tx 500 min-rx 500 multiplier 3
!
!
class-map match-any Control
  description "Mobility␣signaling␣and␣Network␣Protocol␣traffic"
 match dscp cs6
 match dscp cs7
 match dscp cs4
class-map match-all qos-group-cos1
 match qos-group 5
class-map match-all qos-group-cos3
 match qos-group 2
class-map match-all qos-group-cos2
 match qos-group 3
class-map match-any COS3
  description "LTE␣data␣traffic"
 match precedence 2
class-map match-any COS2
  description "UMTS␣R99,␣HS␣traffic"
 match precedence 3
class-map match-any COS1
  description "Real␣Time␣applications␣(3G␣Voice,␣VoLTE)"
 match dscp cs5
 match dscp ef
class-map match-any qos-group-control
 match qos-group 6
!
policy-map MSN_Output_Policy
 description Egress Policy for IP MSN, revised QOS
 class COS1
  set cos 5
  police cir percent 40
  priority level 1
 class Control
  set cos 4
  bandwidth remaining percent 50
  queue-limit 2048 packets
 class COS2
  set cos 4
  bandwidth remaining percent 20
```

```
  queue-limit 4096 packets
 class COS3
  set cos 4
  bandwidth remaining percent 20
  queue-limit 4096 packets
 class class-default
  set cos 4
  bandwidth remaining percent 10
  queue-limit 4096 packets
policy-map MSN_Shaper_Parent_Policy_egress_800M
 class class-default
  shape average 760000000 3040000
   service-policy MSN_Output_Policy
!
!
interface Loopback0
 description OSPF Router ID
 ip address 192.168.0.11 255.255.255.255
!
interface Loopback20
 description CORE-OSPF-ROUTER-ID
 ip address 192.168.200.11 255.255.255.255
 ipv6 address 2001:506:4600:8E1::2/128
ospfv3 15 ipv6 area 0
!
interface Port-channel1
 description LACP Link between MSN_A and MSN_B
 no ip address
 no negotiation auto
 service instance 90 ethernet
  encapsulation dot1q 90
  bridge-domain 90
 !
 service instance 91 ethernet
  encapsulation dot1q 91
  bridge-domain 91
 !
!
interface Port-channel11
 description LACP Link between MSN_A and PE
 ip address 77.66.55.46 255.255.255.252
 no negotiation auto
 ipv6 address 2001:506:4600:8228::2/64
 bfd template msn-bfd-template
 lacp min-bundle 2
!
interface GigabitEthernet1
 ip address 10.10.10.9 255.255.255.0
 negotiation auto
!
interface GigabitEthernet2
 description Backhaul Primary interface to SIAD
 no ip address
 negotiation auto
 service-policy output MSN_Shaper_Parent_Policy_egress_800M
 service instance 1071 ethernet
  encapsulation dot1q 1071
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1071
```

```
 !
!
interface GigabitEthernet3
 no ip address
 negotiation auto
 service-policy output MSN_Output_Policy
 channel-group 1 mode active
!
interface GigabitEthernet4
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet5
 no ip address
 negotiation auto
 service-policy output MSN_Output_Policy
 channel-group 1 mode active
!
interface GigabitEthernet6
 description link to PE
 no ip address
 load-interval 30
 negotiation auto
 lacp rate fast
 service-policy output MSN_Output_Policy
 channel-group 11 mode active
!
interface GigabitEthernet7
 description link to PE
 no ip address
 load-interval 30
 negotiation auto
 lacp rate fast
 service-policy output MSN_Output_Policy
 channel-group 11 mode active
!
interface GigabitEthernet8
 no ip address
 shutdown
 negotiation auto
!
interface BDI90
 description OSPF interchassis link Area 10+0
 ip address 10.200.10.30 255.255.255.252
 ip ospf network point-to-point
 ip ospf resync-timeout 120
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 100
 encapsulation dot1Q 90
 ipv6 address 2001:506:4600:C0C9::1/64
 ipv6 enable
 ospfv3 network point-to-point
 ospfv3 15 ipv6 area 0
 ospfv3 15 ipv6 hello-interval 20
 ospfv3 15 ipv6 dead-interval 70
 ospfv3 15 ipv6 cost 100
!
```

```
interface BDI91
 description OSPF Inter-chassis link Area 10+10
 ip address 10.200.10.50 255.255.255.252
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 100
 encapsulation dot1Q 91
 ipv6 address 2001:506:4600:C154::1/64
 ipv6 enable
 ospfv3 3 network point-to-point
 ospfv3 3 hello-interval 20
 ospfv3 3 dead-interval 70
 ospfv3 3 cost 100
 ospfv3 3 ipv6 area 10
!
interface BDI1071
 description Backhaul Primary interface to SIAD
 ip address 192.168.10.9 255.255.255.252
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf database-filter all out
 ip ospf bfd strict-mode
 ip ospf cost 10
  encapsulation dot1Q 1071
 ipv6 address 2001:506:4047:C5::1/64
 bfd template msn-bfd-template
!
router ospfv3 15
 router-id 192.168.200.11
 !
 address-family ipv6 unicast
 exit-address-family
!
router ospfv3 3
 router-id 192.168.0.11
 !
 address-family ipv6 unicast
  redistribute static metric-type 1
 exit-address-family
!
router ospf 1
 router-id 192.168.0.11
 area 10 nssa no-redistribution no-summary
 passive-interface GigabitEthernet1
 network 10.200.10.48 0.0.0.3 area 10
 network 192.168.0.0 0.0.255.255 area 10
 bfd all-interfaces
!
router ospf 20
 router-id 192.168.200.11
 network 10.200.10.28 0.0.0.3 area 0
 network 192.168.0.11 0.0.0.0 area 0
 network 192.168.200.11 0.0.0.0 area 0
!
router bgp 64600
 bgp router-id 192.168.0.11
 bgp log-neighbor-changes
```

```
 no bgp default ipv4-unicast
 neighbor IPV4-IBGP-MATE-PEER-GRP peer-group
 neighbor IPV4-IBGP-MATE-PEER-GRP remote-as 64600
 neighbor IPV4-IBGP-MATE-PEER-GRP update-source Loopback20
 neighbor IPV4-IBGP-MATE-PEER-GRP timers 30 90
 neighbor IPV6-IBGP-MATE-PEER-GRP peer-group
 neighbor IPV6-IBGP-MATE-PEER-GRP remote-as 64600
 neighbor IPV6-IBGP-MATE-PEER-GRP update-source Loopback20
 neighbor IPV6-IBGP-MATE-PEER-GRP timers 30 90
 neighbor 2001:506:4600:8E9::2 peer-group IPV6-IBGP-MATE-PEER-GRP
 neighbor 2001:506:4600:8228::1 remote-as 7018
 neighbor 2001:506:4600:8228::1 timers 30 90
 neighbor 2001:506:4600:8228::1 fall-over bfd
 neighbor 77.66.55.45 remote-as 7018
 neighbor 77.66.55.45 timers 30 90
 neighbor 77.66.55.45 fall-over bfd
 neighbor 192.168.200.22 peer-group IPV4-IBGP-MATE-PEER-GRP
 !
 address-family ipv4
  network 172.16.0.0 mask 255.255.224.0 route-map NODEB-IPV4-SUMMARY
  redistribute connected route-map CONNECTED-IPV4
  redistribute ospf 1 route-map OAM-IPV4
  neighbor IPV4-IBGP-MATE-PEER-GRP send-community
  neighbor IPV4-IBGP-MATE-PEER-GRP next-hop-self
  neighbor IPV4-IBGP-MATE-PEER-GRP route-map MATE-MSN out
  neighbor 77.66.55.45 activate
  neighbor 77.66.55.45 send-community
  neighbor 77.66.55.45 route-map ROUTE_INCOMING in
  neighbor 192.168.200.22 activate
 exit-address-family
 !
 address-family ipv6
  network 2001:506:4247::/50 route-map ENODEB-IPV6-SUMMARY
  network 2001:506:4247:4000::/50 route-map ENODEB-IPV6-SUMMARY
  network 2001:506:4447::/50 route-map ENODEB-IPV6-SUMMARY
  network 2001:506:4447:4000::/50 route-map ENODEB-IPV6-SUMMARY
  neighbor IPV6-IBGP-MATE-PEER-GRP send-community
  neighbor IPV6-IBGP-MATE-PEER-GRP next-hop-self
  neighbor IPV6-IBGP-MATE-PEER-GRP route-map MATE-MSN-IPV6 out
  neighbor 2001:506:4600:8E9::2 activate
  neighbor 2001:506:4600:8228::1 activate
  neighbor 2001:506:4600:8228::1 send-community
  neighbor 2001:506:4600:8228::1 route-map ROUTE_INCOMING_IPV6 in
  neighbor 2001:506:4600:8228::1 route-map ROUTE_OUTGOING_IPV6 out
 exit-address-family
!
!
virtual-service csr_mgmt
!
ip forward-protocol nd
!
ip bgp-community new-format
ip community-list standard SHORTHAUL-ALL permit 64600:1000
ip community-list standard CONNECTED-IPV4-ALL permit 64600:2000
ip community-list standard PE-DEFAULT-V4 permit 64600:7000
ip community-list standard PE-DEFAULT-V6 permit 64600:7100
ip community-list standard OAM-IPV4-ALL permit 64600:3000
no ip http server
no ip http secure-server
```

```
ip route 172.16.0.0 255.255.224.0 Null0 name NodeBBER
!
ip access-list standard V4-OAM-ROUTES
 permit 172.17.8.0 0.0.7.255
!
!
!
ip prefix-list DEFAULT_ONLY seq 10 permit 0.0.0.0/0
ipv6 route static bfd BDI1071 2001:506:4047:C5::2
ipv6 route 2001:506:4247:340::/61 BDI1071 2001:506:4047:C5::2
ipv6 route 2001:506:4247::/50 Null0 name ENodeBBER
ipv6 route 2001:506:4247:4000::/50 Null0 name ENodeBBER
ipv6 route 2001:506:4447:340::/61 BDI1071 2001:506:4047:C5::2
ipv6 route 2001:506:4447::/50 Null0 name ENodeBOAM
ipv6 route 2001:506:4447:4000::/50 Null0 name ENodeBOAM
!
!
ipv6 prefix-list DEFAULT_ONLY_IPV6 seq 10 permit ::/0
!
ipv6 prefix-list GROUP-A_IPV6 seq 10 permit 2001:506:4247::/50
ipv6 prefix-list GROUP-A_IPV6 seq 20 permit 2001:506:4447::/50
!
ipv6 prefix-list GROUP-B_IPV6 seq 10 permit 2001:506:4247:4000::/50
ipv6 prefix-list GROUP-B_IPV6 seq 20 permit 2001:506:4447:4000::/50
route-map ROUTE_INCOMING permit 10
 match ip address prefix-list DEFAULT_ONLY
 set community 64600:7000
!
route-map CONNECTED-IPV4 permit 10
 description redistributed ipv4 connected interfaces
 set community 64600:2000
!
route-map OAM-IPV4 permit 10
 description Redistributed OAM IPv4 routes
 match ip address V4-OAM-ROUTES
 set community 64600:3000
!
route-map ROUTE_INCOMING_IPV6 permit 10
 match ipv6 address prefix-list DEFAULT_ONLY_IPV6
 set community 64600:7100
!
route-map ROUTE_OUTGOING_IPV6 permit 10
 match ipv6 address prefix-list GROUP-A_IPV6
 set community 13979:2784
!
route-map ROUTE_OUTGOING_IPV6 permit 20
 match ipv6 address prefix-list GROUP-B_IPV6
!
route-map ENODEB-IPV6-SUMMARY permit 10
 description Summary ENODEB IPV6 static /50
 set community 64600:1000
!
route-map MATE-MSN-IPV6 permit 10
 description permit all including specifics to mate
 match community SHORTHAUL-ALL PE-DEFAULT-V6
!
route-map MATE-MSN permit 10
 description permit all including specifics to mate
 match community SHORTHAUL-ALL CONNECTED-IPV4-ALL OAM-IPV4-ALL PE-DEFAULT-V4
```

```
!
route-map NODEB-IPV4-SUMMARY permit 10
 description Summary NODEB address space
 set community 64600:1000
!
snmp-server enable traps tty
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors bad-packet
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps config-copy
snmp-server enable traps syslog
snmp-server enable traps bfd
snmp-server host 10.10.10.10 version 2c v2c
!
!
control-plane
!
!
line con 0
 stopbits 1
line vty 0
 login
line vty 1
 login
 length 0
line vty 2 4
 login
!
end
```

B.4 MSN B

```
MSN_B#sh run
Building configuration...

Current configuration : 10959 bytes
!
! Last configuration change at 13:37:38 UTC Fri Apr 19 2019
!
version 15.6
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname MSN_B
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
ipv6 unicast-routing
ipv6 multicast rpf use-bgp
!
```

```
subscriber templating
!
!
multilink bundle-name authenticated
!
!
license udi pid CSR1000V sn 9VQTSAPVJS7
!
spanning-tree extend system-id
!
!
redundancy
bfd-template single-hop msn-bfd-template
 interval min-tx 500 min-rx 500 multiplier 3
!
!
class-map match-any Control
  description "Mobility␣signaling␣and␣Network␣Protocol␣traffic"
 match dscp cs6
 match dscp cs7
 match dscp cs4
class-map match-all qos-group-cos1
 match qos-group 5
class-map match-all qos-group-cos3
 match qos-group 2
class-map match-all qos-group-cos2
 match qos-group 3
class-map match-any COS3
  description "LTE␣data␣traffic"
 match precedence 2
class-map match-any COS2
  description "UMTS␣R99,␣HS␣traffic"
 match precedence 3
class-map match-any COS1
  description "Real␣Time␣applications␣(3G␣Voice,␣VoLTE)"
 match dscp cs5
 match dscp ef
class-map match-any qos-group-control
 match qos-group 6
!
policy-map MSN_Output_Policy
 description Egress Policy for IP MSN, revised QOS
 class COS1
  set cos 5
  police cir percent 40
  priority level 1
 class Control
  set cos 4
  bandwidth remaining percent 50
  queue-limit 2048 packets
 class COS2
  set cos 4
  bandwidth remaining percent 20
  queue-limit 4096 packets
 class COS3
  set cos 4
  bandwidth remaining percent 20
  queue-limit 4096 packets
 class class-default
```

```
  set cos 4
  bandwidth remaining percent 10
  queue-limit 4096 packets
policy-map MSN_Shaper_Parent_Policy_egress_800M
 class class-default
  shape average 760000000 3040000
   service-policy MSN_Output_Policy

!
interface Loopback0
 description OSPF Router ID
 ip address 192.168.0.22 255.255.255.255
!
interface Loopback20
 description CORE-OSPF-ROUTER-ID
 ip address 192.168.200.22 255.255.255.255
 ipv6 address 2001:506:4600:8E9::2/128
 ospfv3 15 ipv6 area 0
!
interface Port-channel1
 description LACP Link between MSN_B and MSN_A
 no ip address
 no negotiation auto
 service instance 90 ethernet
  encapsulation dot1q 90
  bridge-domain 90
 !
 service instance 91 ethernet
  encapsulation dot1q 91
  bridge-domain 91
 !
!
interface Port-channel12
 description LACP Link between MSN_B and PE
 ip address 77.66.55.86 255.255.255.252
 no negotiation auto
 ipv6 address 2001:506:4600:822A::2/64
 bfd template msn-bfd-template
 lacp min-bundle 2
!
interface GigabitEthernet1
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet2
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet3
 no ip address
 negotiation auto
 service-policy output MSN_Output_Policy
 channel-group 1 mode active
!
interface GigabitEthernet4
 description Backhaul Secondary interface to SIAD
 no ip address
```

```
 negotiation auto
 service-policy output MSN_Shaper_Parent_Policy_egress_800M
 service instance 2071 ethernet
  encapsulation dot1q 2071
  rewrite ingress tag pop 1 symmetric
  bridge-domain 2071
!
interface GigabitEthernet5
 no ip address
 negotiation auto
 service-policy output MSN_Output_Policy
 channel-group 1 mode active
!
interface GigabitEthernet6
 description link to PE
 no ip address
 load-interval 30
 negotiation auto
 lacp rate fast
 service-policy output MSN_Output_Policy
 channel-group 12 mode active
!
interface GigabitEthernet7
 description link to PE
 no ip address
 load-interval 30
 negotiation auto
 lacp rate fast
 service-policy output MSN_Output_Policy
 channel-group 12 mode active
!
interface GigabitEthernet8
 no ip address
 shutdown
 negotiation auto
!
interface BDI90
 description OSPF interchassis link Area 0
 ip address 10.200.10.29 255.255.255.252
 ip ospf network point-to-point
 ip ospf resync-timeout 120
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf cost 100
 encapsulation dot1Q 90
 ipv6 address 2001:506:4600:C0C9::2/64
 ipv6 enable
 ospfv3 15 network point-to-point
 ospfv3 15 hello-interval 20
 ospfv3 15 dead-interval 70
 ospfv3 15 cost 100
 ospfv3 15 ipv6 area 0
!
interface BDI91
 description OSPF Inter-chassis link Area 10
 ip address 10.200.10.49 255.255.255.252
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
```

```
 ip ospf cost 100
 encapsulation dot1Q 91
 ipv6 address 2001:506:4600:C154::2/64
 ipv6 enable
 ospfv3 3 network point-to-point
 ospfv3 3 hello-interval 20
 ospfv3 3 dead-interval 70
 ospfv3 3 cost 100
 ospfv3 3 ipv6 area 10
!
interface BDI2071
 description Backhaul Secondary interface to SIAD
 ip address 192.168.20.21 255.255.255.252
 ip ospf network point-to-point
 ip ospf dead-interval 70
 ip ospf hello-interval 20
 ip ospf database-filter all out
 ip ospf cost 300
 encapsulation dot1Q 2071
 ipv6 address 2001:506:4047:40C5::1/64
!
router ospfv3 15
 router-id 192.168.200.22
 !
 address-family ipv6 unicast
 exit-address-family
!
router ospfv3 3
 router-id 192.168.0.22
 !
 address-family ipv6 unicast
  redistribute static metric-type 1
 exit-address-family
!
router ospf 1
 router-id 192.168.0.22
 area 10 nssa no-redistribution no-summary
 network 10.200.10.48 0.0.0.3 area 10
 network 192.168.0.0 0.0.255.255 area 10
 bfd all-interfaces
!
router ospf 20
 router-id 192.168.200.22
 network 10.200.10.28 0.0.0.3 area 0
 network 192.168.0.22 0.0.0.0 area 0
 network 192.168.200.22 0.0.0.0 area 0
!
router bgp 64600
 bgp router-id 192.168.0.22
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor IPV4-IBGP-MATE-PEER-GRP peer-group
 neighbor IPV4-IBGP-MATE-PEER-GRP remote-as 64600
 neighbor IPV4-IBGP-MATE-PEER-GRP update-source Loopback20
 neighbor IPV4-IBGP-MATE-PEER-GRP timers 30 90
 neighbor IPV6-IBGP-MATE-PEER-GRP peer-group
 neighbor IPV6-IBGP-MATE-PEER-GRP remote-as 64600
 neighbor IPV6-IBGP-MATE-PEER-GRP update-source Loopback20
 neighbor IPV6-IBGP-MATE-PEER-GRP timers 30 90
```

```
 neighbor 2001:506:4600:8E1::2 peer-group IPV6-IBGP-MATE-PEER-GRP
 neighbor 2001:506:4600:822A::1 remote-as 7018
 neighbor 2001:506:4600:822A::1 timers 30 90
 neighbor 2001:506:4600:822A::1 fall-over bfd
 neighbor 77.66.55.85 remote-as 7018
 neighbor 77.66.55.85 timers 30 90
 neighbor 77.66.55.85 fall-over bfd
 neighbor 192.168.200.11 peer-group IPV4-IBGP-MATE-PEER-GRP
 !
 address-family ipv4
  network 172.16.0.0 mask 255.255.224.0 route-map NODEB-IPV4-SUMMARY
  redistribute connected route-map CONNECTED-IPV4
  redistribute ospf 1 route-map OAM-IPV4
  neighbor IPV4-IBGP-MATE-PEER-GRP send-community
  neighbor IPV4-IBGP-MATE-PEER-GRP next-hop-self
  neighbor IPV4-IBGP-MATE-PEER-GRP route-map MATE-MSN out
  neighbor 77.66.55.85 activate
  neighbor 77.66.55.85 send-community
  neighbor 77.66.55.85 route-map ROUTE_INCOMING in
  neighbor 192.168.200.11 activate
 exit-address-family
 !
 address-family ipv6
  network 2001:506:4247::/50 route-map ENODEB-IPV6-SUMMARY
  network 2001:506:4247:4000::/50 route-map ENODEB-IPV6-SUMMARY
  network 2001:506:4447::/50 route-map ENODEB-IPV6-SUMMARY
  network 2001:506:4447:4000::/50 route-map ENODEB-IPV6-SUMMARY
  neighbor IPV6-IBGP-MATE-PEER-GRP send-community
  neighbor IPV6-IBGP-MATE-PEER-GRP next-hop-self
  neighbor IPV6-IBGP-MATE-PEER-GRP route-map MATE-MSN-IPV6 out
  neighbor 2001:506:4600:8E1::2 activate
  neighbor 2001:506:4600:822A::1 activate
  neighbor 2001:506:4600:822A::1 send-community
  neighbor 2001:506:4600:822A::1 route-map ROUTE_INCOMING_IPV6 in
  neighbor 2001:506:4600:822A::1 route-map ROUTE_OUTGOING_IPV6 out
 exit-address-family
!
!
virtual-service csr_mgmt
!
ip forward-protocol nd
!
ip bgp-community new-format
ip community-list standard SHORTHAUL-ALL permit 64600:1000
ip community-list standard CONNECTED-IPV4-ALL permit 64600:2000
ip community-list standard PE-DEFAULT-V4 permit 64600:7000
ip community-list standard PE-DEFAULT-V6 permit 64600:7100
ip community-list standard OAM-IPV4-ALL permit 64600:3000
no ip http server
no ip http secure-server
ip route 172.16.0.0 255.255.224.0 Null0 name NodeBBER
!
ip access-list standard V4-OAM-ROUTES
 permit 172.17.8.0 0.0.7.255
!
!
ip prefix-list DEFAULT_ONLY seq 10 permit 0.0.0.0/0
ipv6 route 2001:506:4247:340::/61 BDI2071 2001:506:4047:40C5::2 200
ipv6 route 2001:506:4247::/50 Null0 name ENodeBBER
```

```
ipv6 route 2001:506:4247:4000::/50 Null0 name ENodeBBER
ipv6 route 2001:506:4447:340::/61 BDI2071 2001:506:4047:40C5::2 200
ipv6 route 2001:506:4447::/50 Null0 name ENodeBOAM
ipv6 route 2001:506:4447:4000::/50 Null0 name ENodeBOAM
!
!
ipv6 prefix-list DEFAULT_ONLY_IPV6 seq 10 permit ::/0
!
ipv6 prefix-list GROUP-A_IPV6 seq 10 permit 2001:506:4247::/50
ipv6 prefix-list GROUP-A_IPV6 seq 20 permit 2001:506:4447::/50
!
ipv6 prefix-list GROUP-B_IPV6 seq 10 permit 2001:506:4247:4000::/50
ipv6 prefix-list GROUP-B_IPV6 seq 20 permit 2001:506:4447:4000::/50
route-map ROUTE_INCOMING permit 10
 match ip address prefix-list DEFAULT_ONLY
 set community 64600:7000
!
route-map CONNECTED-IPV4 permit 10
 description redistributed ipv4 connected interfaces
 set community 64600:2000
!
route-map OAM-IPV4 permit 10
 description Redistributed OAM IPv4 routes
 match ip address V4-OAM-ROUTES
 set community 64600:3000
!
route-map ROUTE_INCOMING_IPV6 permit 10
 match ipv6 address prefix-list DEFAULT_ONLY_IPV6
 set community 64600:7100
!
route-map ROUTE_OUTGOING_IPV6 permit 10
 match ipv6 address prefix-list GROUP-A_IPV6
!
route-map ROUTE_OUTGOING_IPV6 permit 20
 match ipv6 address prefix-list GROUP-B_IPV6
 set community 13979:2784
!
route-map ENODEB-IPV6-SUMMARY permit 10
 description Summary ENODEB IPV6 static /50
 set community 64600:1000
!
route-map MATE-MSN-IPV6 permit 10
 description permit all including specifics to mate
 match community SHORTHAUL-ALL PE-DEFAULT-V6
!
route-map MATE-MSN permit 10
 description permit all including specifics to mate
 match community SHORTHAUL-ALL CONNECTED-IPV4-ALL OAM-IPV4-ALL PE-DEFAULT-V4
!
route-map NODEB-IPV4-SUMMARY permit 10
 description Summary NODEB address space
 set community 64600:1000
!
snmp-server enable traps tty
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors bad-packet
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps config-copy
```

```
snmp-server enable traps syslog
snmp-server enable traps bfd
!
control-plane
!
!
line con 0
 logging synchronous
 stopbits 1
line vty 0
 login
line vty 1
 login
 length 0
line vty 2 4
 login
!
!
end
```

B.5 PE

```
PE#sh run
Building configuration...

Current configuration : 3546 bytes
!
! Last configuration change at 11:37:42 UTC Fri Apr 19 2019
!
version 15.6
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname PE
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
ipv6 unicast-routing
!
!
subscriber templating
!
!
multilink bundle-name authenticated
!
!
license udi pid CSR1000V sn 95V4PNL3KC1
!
spanning-tree extend system-id
!
!
```

```
redundancy
bfd-template single-hop msn-bfd-template
 interval min-tx 500 min-rx 500 multiplier 3
!
!
interface Loopback0
 description Router ID
 ip address 192.168.0.77 255.255.255.255
!
interface Port-channel11
 description LACP Link between PE and MSN_A
 ip address 77.66.55.45 255.255.255.252
 no negotiation auto
 ipv6 address 2001:506:4600:8228::1/64
 bfd template msn-bfd-template
 lacp min-bundle 2
!
interface Port-channel12
 description LACP Link between PE and MSN_B
 ip address 77.66.55.85 255.255.255.252
 no negotiation auto
 ipv6 address 2001:506:4600:822A::1/64
 bfd template msn-bfd-template
 lacp min-bundle 2
!
interface GigabitEthernet1
 description link to MSN_A
 no ip address
 load-interval 30
 negotiation auto
 lacp rate fast
 channel-group 11 mode active
!
interface GigabitEthernet2
 description link to MSN_B
 no ip address
 load-interval 30
 negotiation auto
 lacp rate fast
 channel-group 12 mode active
!
interface GigabitEthernet3
 description link to MSN_A
 no ip address
 load-interval 30
 negotiation auto
 lacp rate fast
 channel-group 11 mode active
!
interface GigabitEthernet4
 description link to MSN_A
 no ip address
 load-interval 30
 negotiation auto
 lacp rate fast
 channel-group 12 mode active
!
interface GigabitEthernet5
 no ip address
```

```
 shutdown
 negotiation auto
!
interface GigabitEthernet6
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet7
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet8
 no ip address
 shutdown
 negotiation auto
!
router bgp 7018
 bgp router-id 192.168.0.77
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 2001:506:4600:8228::2 remote-as 64600
 neighbor 2001:506:4600:8228::2 timers 30 90
 neighbor 2001:506:4600:8228::2 fall-over bfd
 neighbor 2001:506:4600:822A::2 remote-as 64600
 neighbor 2001:506:4600:822A::2 timers 30 90
 neighbor 2001:506:4600:822A::2 fall-over bfd
 neighbor 77.66.55.46 remote-as 64600
 neighbor 77.66.55.46 timers 30 90
 neighbor 77.66.55.46 fall-over bfd
 neighbor 77.66.55.86 remote-as 64600
 neighbor 77.66.55.86 timers 30 90
 neighbor 77.66.55.86 fall-over bfd
 !
 address-family ipv4
  network 77.66.55.44 mask 255.255.255.252
  network 77.66.55.84 mask 255.255.255.252
  neighbor 77.66.55.46 activate
  neighbor 77.66.55.46 send-community
  neighbor 77.66.55.86 activate
  neighbor 77.66.55.86 send-community
 exit-address-family
 !
 address-family ipv6
  network 2001:506:4600:8228::/64
  network 2001:506:4600:822A::/64
  neighbor 2001:506:4600:8228::2 activate
  neighbor 2001:506:4600:8228::2 send-community
  neighbor 2001:506:4600:822A::2 activate
  neighbor 2001:506:4600:822A::2 send-community
 exit-address-family
!
!
virtual-service csr_mgmt
!
ip forward-protocol nd
!
ip bgp-community new-format
```

```
no ip http server
no ip http secure-server
!
!
control-plane
!
!
line con 0
 stopbits 1
line vty 0
 login
line vty 1
 login
 length 0
line vty 2 4
 login
!
!
end
```

C Odpovědi na kontrolní otázky

C.1 Pro Scénář 1

1. **Co jsou *Shorthaul* a *Backhaul* linky a kde se nachází?**
   Shorthaul je spojení (linka) mezi NodeB/eNodeB a SIAD směrovačem. *Backhaul* je spojení (linka) mezi SIAD směrovačem a MSN párem.
2. **Jaký význam má *Loopback0* na SIAD směrovači?**
   Loopback0 slouží k identifikaci směrovače a k jejímu následnému použití u směrovacích protokolů.
3. **Co je BDI rozhraní?**
   BDI neboli Bridge Domain Interface je logické rozhraní, které umožňuje obousměrný tok mezi linkovou vrstvou L2 a síťovou vrstvou L3. Musí být přiřazeno k fyzickému rozhraní.
4. **Jaký význam má `Null0` rozhraní u IPv4 a IPv6 statických cest nastavených na SIAD směrovači?**
   `Null0` rozhraní se zde využívá k prevenci proti vytváření smyček a nežádoucímu provozu. Jedná se o tzv. „Black Hole Routing"- černé díry.
5. **Jak je směrován IPv4 provoz ze SIAD směrovače k MSN páru neboli „upstream“?**
   IPv4 „upstream“ provoz je směrován ze SIAD směrovače pomocí statických cest. Primární linka má výchozí administrativní vzdálenost 1. Sekundární má administrativní vzdálenost ručně nastavenou na hodnotu 10.
6. **Jak je směrován IPv6 provoz ze SIAD směrovače k MSN páru neboli „upstream“?**
   IPv6 „upstream“ provoz je směrován ze SIAD směrovače pomocí statických cest. Primární linka má výchozí administrativní vzdálenost 1. Sekundární má administrativní vzdálenost ručně nastavenou na hodnotu 10.
7. **Jak je směrován IPv4 provoz z MSN páru k SIAD směrovači neboli „downstream“?**
   IPv4 „downstream“ provoz je z MSN páru směrován pomocí OSPFv2. Z MSN_A je hodnota „cost“ nastavena na 10. Z MSN_B je to hodnota 300. Primární cesta vedoucí přes MSN_A je tak upřednostňována.
8. **Jak je směrován IPv6 provoz z MSN páru k SIAD směrovači neboli „downstream“?**
   IPv6 „downstream“ provoz je z MSN páru na SIAD směrovač směrován pomocí statických cest. Primární linka má výchozí administrativní vzdálenost 1. Sekundární má administrativní vzdálenost ručně nastavenou na hodnotu 200.

9. **Na která spojení je navázán protokol BFD?**
   BFD protokol je navázán na primární statické cesty na *Backhaul* lince, na OSPFv2 u primárního spojení mezi SIAD směrovačem a MSN_A a u BGP protokolu na linkách mezi MSN párem a PE.
10. **K čemu slouží OSPFv2 mezi MSN_A a MSN_B a jaký mají jednotlivé procesy význam?**
    OSPFv2 slouží k předávání IPv4 cest mezi MSN párem. Využívá procesy `router ospf 1` a `router ospf 20`. `Router ospf 1` se stará o komunikaci se SIAD směrovačem, pro komunikaci mezi MSN párem je využíván `router ospf 20`.
11. **K čemu slouží OSPFv3 mezi MSN_A a MSN_B a jaký mají jednotlivé procesy význam?**
    OSPFv3 slouží k předávání IPv6 cest mezi MSN párem. V případě výpadku IPv6 statické primární cesty na MSN_A směrem k SIAD směrovači bude využita záložní linka z MSN_B, kterou MSN_A oznámí protokol OSPFv3. Protokol OSPFv3 je rozdělen na `router ospfv3 15` a `router ospfv3 3`, kde `router ospfv3 3` má na starost redistribuci IPv6 statických cest do OSPFv3 a `router ospfv3 15` informuje ostatní připojené směrovače o dostupnosti Loopback20 a tedy samotného MSN pro použití u protokolu iBGP.
12. **Které atributy jsou využívány při nastavení BGP?**
    V simulované mobilní transportní síti se pro nastavení směrování za pomoci BGP používá atributu community list. Jiný BGP atribut se v síti nenastavuje a jsou ponechána pouze výchozí nastavení např. u lokální preference nebo atributu weight.
13. **Které hlavní route mapy mají na starost odchozí a příchozí provoz mezi MSN párem a PE u BGP?**
    Hlavními route mapami jsou `ROUTE_INCOMING`, `ROUTE_INCOMING_IPV6` a `ROUTE_OUTGOING_IPV6`.

C.2 Pro Scénář 2

Odpovědi pro TroubleTicket-1

1. **Pomocí jakých příkazů se BDI rozhraní naváže na fyzické rozhraní na směrovači?**
   BDI rozhraní se přiřadí k fyzickému rozhraní přes `service instance <číslo>`, `encapsulation dot1q <číslo>` a `bridge-domain <číslo>`.
2. **Jakou administrativní vzdálenost mají statické cesty pro eNodeB vedoucí z MSN_A (primárního) směrovače k SIAD směrovači a jaká je tato administrativní vzdálenost v případě směrovače MSN_B (sekundárního)?**
   Statické cesty pro eNodeB vedoucí z MSN_A mají výchozí administrativní vzdálenost 1. Z MSN_B mají statické cesty administrativní vzdálenost ručně nastavenou na 200.
3. **Na jaké hodnoty jsou nastaveny *hello interval* a *dead interval* pro OSPFv3 na MSN páru?**
   Hodnoty intervalů jsou v simulované transportní síti nastaveny na 20 sekund pro *hello interval* a 70 sekund pro *dead interval* u OSPFv3.

Odpovědi pro TroubleTicket-2

1. **Jakým příkazem spustíme BFD protokol na rozhraní?**
   BFD protokol se spustí na rozhraní vložením intervalů, tedy příkazem `bfd interval <milliseconds> min-rx <milliseconds> multiplier <interval-multiplier>`, nebo pomocí „template“. Ten se vytvoří v globálním módu příkazem `bfd-template single-hop <název>` a poté se přidají hodnoty intervalů `interval min-tx <milliseconds> min-rx <milliseconds> multiplier <interval-multiplier>`.
2. **Jakým příkazem přiřadíme fyzické rozhraní k „port-channel“ a jak na „port-channel“ skrze fyzické rozhraní spustíme protokol LACP?**
   „Port-channelX“ již musí být předem vytvořený. Fyzické rozhraní do něj pak vložíme příkazem `channel-group X mode active`, který vložíme pod samotné fyzické rozhraní. Částí příkazu `mode active` spouštíme protokol LACP.
3. **Jakým způsobem jsou redistribuovány OAM adresy NodeB do protokolu BGP na MSN páru?**
   MSN pár posílá veškerá data směrem k NodeB pomocí OSPFv2. Údaje o těchto sítích jsou ve směrovací tabulce MSN páru pod protokolem OSPFv2. Redistribuce musí probíhat z OSPFv2 do BGP protokolu.

D Obsah DVD

```
ROOT
├── Simulační scénáře pro analýzu chování transportních sítí_xkolac15.pdf
├── GNS3_Simulation_Scenarios1_2_TT1-2.gns3project
├── EVE-NG_Simulation_Scenarios1_2_TT1-2.zip
├── csr1000v-universalk9.03.17.00.S.156-1.S-ext.qcow2
├── Scénář 1 - Konfigurace transportní sítě
│   ├── SIAD.txt
│   ├── LEC.txt
│   ├── MSN A.txt
│   ├── MSN B.txt
│   └── PE.txt
├── Scénář 2 - Časté chyby v transportní síti
│   ├── TroubleTicket - 1
│   │   ├── SIAD.txt
│   │   ├── MSN A.txt
│   │   └── MSN B.txt
│   └── TroubleTicket - 2
│       ├── SIAD.txt
│       ├── MSN A.txt
│       └── MSN B.txt
└── Měření na serveru ESXi
    ├── EVE-NG
    │   ├── eve-cpu.jpg
    │   └── eve-ram.jpg
    └── GNS3
        ├── gns3-cpu.jpg
        └── gns3-ram.jpg
```

Obsah DVD je dostupný také zde:

https://vutbr-my.sharepoint.com/:f:/g/personal/xkolac15_vutbr_cz/ErsCahrOhYNJm_od9B-m53ABsMCdk2RaQQQNedq9k5sx_A?e=qbTIOV