VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

MODELOVÁNÍ A SIMULACE BGP

MODELING AND SIMULATION OF BGP

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. ADRIÁN NOVÁK
AUTHOR

VEDOUCÍ PRÁCE — Ing. VLADIMÍR VESELÝ, Ph.D.
SUPERVISOR

BRNO 2019

Ústav informačních systémů (UIFS) Akademický rok 2018/2019

Zadání diplomové práce

21560

Student: **Novák Adrián, Bc.**
Program: Informační technologie Obor: Počítačové sítě a komunikace
Název: **Modelování a simulace BGP**
Modeling and Simulation of BGP
Kategorie: Počítačové sítě
Zadání:

1. Analyzujte EGP směrovací protokol BGPv4 a prostudujte jeho chování na Cisco zařízeních.
2. Zjistěte stav implementace BGP v OMNeT++.
3. Dle doporučení vedoucího implementujte podporu BGP do frameworku ANSAINET v prostředí OMNeT++ se zaměřením na podporu multi address-family směrování.
4. Ověřte chování implementovaných simulačních modelů vůči odpovídající reálné topologii a analyzujte výsledky.

Literatura:

- Rekhter, Y., Li, T., & Hares, S. (2005). *A border gateway protocol 4 (BGP-4)* (No. RFC 4271).
- Bates, T., Chandra, R., Katz, D., & Rekhter, Y. (2007). *Multiprotocol extensions for BGP-4* (No. RFC 4760).

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Veselý Vladimír, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 30. října 2018

Abstrakt

Práca sa zaoberá modelovaním a simuláciou BGP protokolu v rámci prostredia OMNeT++. V úvodnej časti je priblížený BGP protokol, jeho základné dátové štruktúry a konečný automat nadviazania susedstva. Ďalej je predstavený spôsob konfigurácie protokolu na Cisco zariadeniach. Následne je uvedený popis aktuálneho stavu implementácie BGP protokolu v rámci simulačného prostredia OMNeT++, jeho nedostatky a zistené problémy. Druhá časť práce sa zaoberá návrhom, implementáciou a testovaním danej funcionality BGP protokolu a simulačných modelov. Na záver sú uvedené celkové dosiahnute výsledky a zhodnotenie práce.

Abstract

This Master's thesis deals with modeling and simulation of BGP protocol within the OMNeT++ environment. The BGP protocol is described with employed data structures and the finite state machine of BGP peering. Next, the basic configuration is outlined involving the setup of the BGP protocol on Cisco devices. Further, BGP for OMNeT++ state-of-the-art is investigated together with its lack of functionality and issues. The second part of this thesis deals with design, implementation, and testing of the new functionality of BGP protocol and simulation models. The last section describes the overall achieved results.

Kľúčové slová

BGP, Border gateway protocol, OMNeT++, ANSAINET, INET, modelovanie a simulácia sietí

Keywords

BGP, Border gateway protocol, OMNeT++, ANSAINET, INET, network modeling and simulation

Citácia

NOVÁK, Adrián. *Modelování a simulace BGP*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vladimír Veselý, Ph.D.

Modelování a simulace BGP

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pana inžiniera Vladimíra Veselého. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

. . . . . .
Adrián Novák
19. mája 2019

Poďakovanie

Týmto by som sa chcel poďakovať mojej mamine, za všetko čo pre mňa celý život robila a minimálne za morálnu a finančnú podporu počas celého štúdia. Taktiež veľké ďakujem patrí mojej skvelej snúbenici Lucke, za jej nekonečnú lásku a trpezlivosť s mojimi náladami a problémami. Taktiež ďakujem bratovi a všetkým mojím kamarátom, s ktorými som mohol zdielať svoj život a štúdium na fakulte.

V neposlednom rade sa chcem poďakovať vedúcemu práce pánovi Ing. Vladimírovi Veselému Ph.D., za jeho cenné rady a ochotu kedykoľvek konzultovať problémy.

Pridávam recept na Ham and Eggs, pretože raňajky sú základ dňa. Na prípravu potrebujeme:

- vajíčka – ideálne 3 ks veľkosti L alebo 4 ks veľkosti S, alebo M
- niekoľko plátkov šunky – ideálne, aby pokryli dno panvice
- plátkový syr
- trochu oleja, soli a prípadne korenia
- obľúbenú zeleninu napríklad uhorku a rajčinu
- chlieb alebo pečivo

Na troche oleja mierne osmažíme šunku, na ktorú pridáme vajíčka tak, aby sa žĺtka neroztiekli. Vajíčka osolíme a okoreníme podľa preferencie. Po pár minútach vajíčka so šunkou otočíme naberačkou alebo použijeme trik pomocou zápästia :-). Keď sú už vajíčka hotové dáme ich na tanier, kde na vrch poukladáme plátkový syr. Ham and Eggs ozdobíme zeleninou a priložíme pečivo.

Obsah

Kapitola 1

Úvod

V dnešnej modernej dobe je Internet neodmysliteľnou súčasťou života človeka. Internet je verejne dostupný celosvetový systém vzájomne prepojených počítačových sietí, ktoré prenášajú dáta pomocou preposielania paketov za použitia smerovacích protokolov alebo staticky nakonfigurovaných ciest. Tieto protokoly sú spustené na smerovačoch a zaisťujú výmenu informácií medzi susednými zariadeniami. Mimo iné, medzi tieto informácie patria dostupnosti jednotlivých sietí a liniek, z ktorých sa vytvárajú smerovacie tabuľky, ktoré slúžia pre vyhľadanie najvýhodnejšej trasy od zdrojovej k cieľovej stanici.

V prípade, že je požadovaná alebo plánovaná zmena topológie alebo podporovaného smerovacieho protokolu je potrebný zásah do konfigurácie sieťových prvkov. Takáto zmena môže priniesť určité komplikácie, kedy sa môže stať, že by koncoví klienti zostali bez konektivity k lokálnej sieti alebo Internetu. To v určitých prípadoch nie je žiadúce a je výhodné najskôr využiť možnosť práve modelovania a simulácie takejto zmeny v simulačných prostrediach a nástrojoch, aby sa overila správnosť konfigurácie poprípade či sa zmena konfigurácie oplatila a dosahuje očakávané výsledky. Jedným z takýchto nástrojov je práve OMNeT++. Práve modelovaním a simuláciou smerovacieho protokolu BGP v rámci simulačného prostredia OMNeT++ sa zaoberá táto práca.

V úvode do danej problematiky 2, sa nachádza popis smerovacích protokolov s rozborom BGP protokolu 2.2, ktorý je predmetom tejto práce. K BGP protokolu je uvedené aj rozšírenie o *multi address-family* smerovanie 2.3. V závere tejto kapitoly 2.4 sa nachádza návod so schémou vzorovej topológie, v ktorom je vysvetlené ako nakonfigurovať BGP protokol na Cisco zariadeniach.

V nadväzujúcej kapitole 3 je predstavené simulačné prostredie OMNeT++, framework ANSAINET a jeho začlenenie v rámci OMNeT++. V druhej podkapitole 3.2 je rozobraný aktuálny stav implementácie BGP protokolu v rámci frameworku INET, ktorý obsahuje základný popis zdrojových súborov, aktuálny stav konfiguračného súboru simulačných modelov, zhodnotený beh simulačných modelov so znázorneným diagramom základného nadviazania BGP susedstva medzi smerovačmi. Následne je uvedený zoznam zistených problémov a nekonzistencií implementácie. Tretia podkapitola 3.3 sa zaoberá návrhom opravy zistených chýb a návrhom implementácie rozšírenej funkcionality. V poslednej podkapitole 3.4 je podrobnejšie priblížená implementácia daného rozšírenia a opravy zistených problémov.

V neposlednom rade tvorí súčasť tejto práce kapitola 4, v ktorej je znázornený princíp testovania simulačných modelov voči reálnej topológii so zhodnotením výsledkov daných testov.

Záverečná kapitola 5 obsahuje celkové zhodnotenie práce a komentuje dosiahnuté výsledky.

Kapitola 2

Smerovacie protokoly

V tejto kapitole sa nachádza úvod do smerovacích protokolov [2] so zameraním na skupinu *Exterior Gateway Protocol* (EGP) protokolov s podrobným popisom BGP protokolu s následným rozšírením o multi addres-family smerovanie, ktoré poskytuje prenos viacerých protokolov sieťovej vrstvy. Ako ďalšia je znázornená konfigurácia BGP protokolu na Cisco zariadeniach s podrobným návodom ako sprevádzkovať BGP protokol aj s podporou IPv4 a IPv6 smerovania. Návod je doplnený o ukážky overenia konfigurácie.

2.1 Úvod do smerovacích protokolov

Smerovače preposieľajú pakety na základe informácií, ktoré sa nachádzajú v ich smerovacích tabuľkách. Trasy do vzdialených sietí sa smerovače môžu naučiť dvoma spôsobmi: buď sa jedná o trasy statické, alebo dynamické.

Dynamické smerovacie protokoly vypočítavajú najlepšiu trasu do každej siete. Táto trasa je následne pridaná do smerovacej tabuľky. Hlavnou výhodou dynamických smerovacích protokolov je tá, že smerovače si vymieňajú smerovacie informácie keď nastane zmena topológie. Táto výmena umožňuje smerovačom aby sa automaticky učili o nových sieťach a tiež aby reagovali na stratu konektivity vyhľadaním novej najlepšej trasy.

V porovnaní so statickým smerovaním, dynamické nevyžaduje toľko administratívneho úsila. Avšak dynamické smerovanie vyžaduje viac zdrojov pre svoju funkciu ako je procesorový čas a priepustnosť sieťovej linky. Napriek výhodám, ktoré dynamické smerovacie protokoly ponúkajú, statické smerovanie má v sieťach stále svoje miesto. Sú situácie kedy je vhodnejšie statické smerovanie a tak isto sú situácie kedy je vhodnejšie využiť smerovanie dynamické. Preto sa v moderných sieťach obe kombinujú.
Účel dynamických smerovacích protokolov zahŕňa:

- zisťovanie vzdialených sietí;
- zaistenie aktuálnosti smerovacích informácií;
- výber najlepšej trasy do cieľovej siete;
- schopnosť vyhľadania novej najlepšej trasy ak aktuálna trasa už nie je dostupná.

Hlavné komponenty dynamických smerovacích protokolov obsahujú:

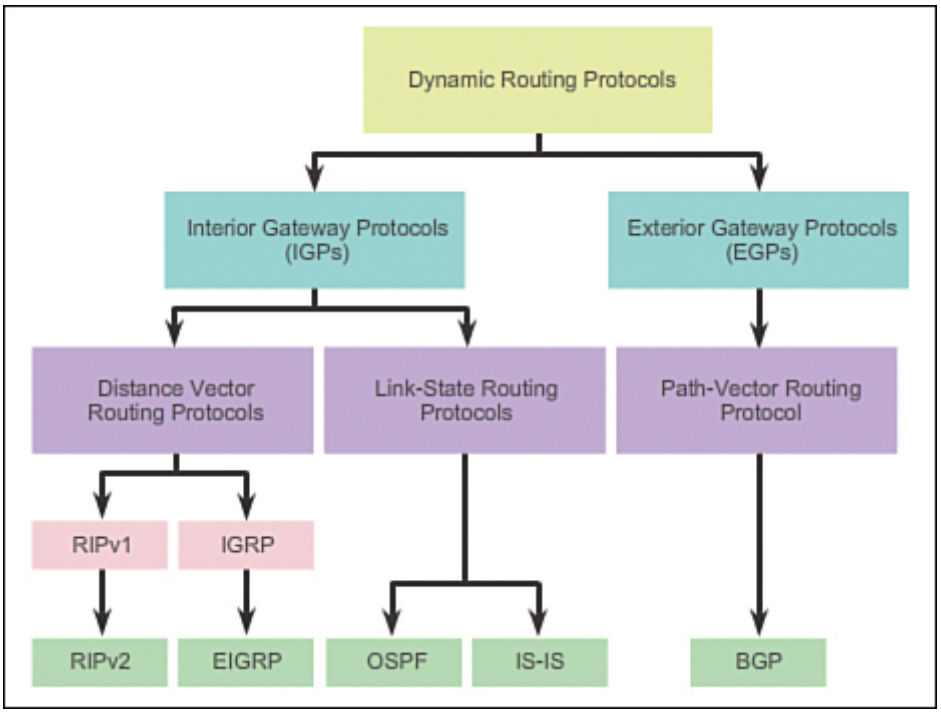
- **Dátové štruktúry**: Smerovacie protokoly pre svoju funkcionalitu typicky využívajú tabuľky alebo databázy, ktoré sú uložené v RAM.

- **Správy smerovacieho protokolu**: Smerovacie protokoly využívajú niekoľko druhov správ. Pre objavenie a nadviazanie spojenia so susednými smerovačmi, pre výmenu smerovacích informácií a na vykonanie úloh spojených s učením a správou informácií o topológií.
- **Algoritmus**: Konečný zoznam krokov, potrebný pre vykonanie danej úlohy. Smerovacie protokoly využívajú algoritmy pre spracovanie smerovacích informácií a pre výber najlepšej cesty.

2.1.1 Rozdelenie smerovacích protokolov

Na obrázku 2.1 je zobrazené ako môžu byť smerovacie protokoly rozdelené do jednotlivých skupín na základe viacerých charakteristík. Medzi tieto charakteristiky patria:

- **Účel**: Interior Gateway Protocol (IGP) alebo Exterior Gateway Protocol (EGP)
- **Princíp činnosti**: Distance vector protocol, link-state protocol alebo path-vector protocol
- **Správanie**: Classful protocol (triedne) alebo classless protocol (beztriedne)



Obr. 2.1: Rozdelenie smerovacích protokolov [2].

Internet je založený na koncepte autonómnych systémov. **Autonómny systém (AS)** je kolekcia smerovačov pod jednou administratívou ako je spoločnosť alebo organizácia. Typicky sa za AS považuje interná sieť spoločnosti a sieť poskytovateľa sieťových služieb (ISP). Pre tento účel sú vyžadované dva typy smerovacích protokolov:

- **Interior Gateway Protocol (IGP)**: používa sa na smerovanie v rámci daného AS. Tiež sa nazýva ako *intra-AS* smerovací protokol. Spoločnosti a tiež ISP používajú

IGP protokol v rámci svojich interných sietí. Do tejto skupiny patria protokoly: RIP, EIGRP, OSPF a IS-IS.

- **Exterior Gateway Protocol (EGP)**: používa sa na smerovanie medzi autonómnymi systémami. Tiež sa nazýva ako *inter-AS* smerovací protokol. Používa sa na prepojenie spoločností a sietí poskytovateľov služieb. Do tejto skupiny patrí práve *Border Gateway Protocol (BGP)* a je jediný protokol, ktorý sa využíva na smerovanie v rámci internetu.

IGP protokoly sa na základe princípu činnosti delia na dve podskupiny:

- **Distance vector protocol**: Smerovače oznamujú trasy na základe dvoch parametrov
  – Distance (vzdialenosť) udáva ako ďaleko je vzdialená sieť a je založená na metrike, ktorá závisí na počte skokov, cene, priepustnosti, oneskorení linky a iné.
  – Vector určuje smer, čiže susedný smerovač alebo výstupné rozhranie, ktoré vedie do cieľa.
  Distance vector smerovací protokol nemá povedomie o celej trase do cieľovej siete. Jediná informácia, ktorú smerovač pozná o vzdialenej sieti je vzdialenosť a susedný smerovač alebo výstupné rozhranie smerovača, ktoré vedie do vzdialenej siete. Patria sem: RIPv1, RIPv2, IGRP a EIGRP.

- **Link-state protocol**: Smerovač si vytvára kompletný pohľad na celú topológiu na základe informácií, ktoré sú poskytované všetkými ostatnými smerovačmi v rámci siete. Je dôležité aby všetky smerovače mali rovnakú mapu topológie v opačnom prípade by mohli vznikať nekonzistencie ako smerovacie slučky a pod. Smerovače si neposielajú periodické aktualizácie, ale posielajú si aktualizácie iba ak nastane zmena v rámci siete napríklad pád linky a pod. Do tejto skupiny patria: OSPF a IS-IS protokoly.

Pre skupinu EGP protokolov existuje jediná skupina na základe princípu činnosti a tou je skupina **Path-Vector Routing Protocol**. Path-vector funguje podobne ako distance vector, len miesto vzdialenosti sa využíva trasa. Trasa je zoznam autonómnych systémov, ktoré vedú k danej sieti. Path-vector zabezpečuje bezslučkové smerovanie a to takým štýlom, že ak daný hraničný smerovač v AS obdrží trasu od smerovača z iného AS a vidí v danej trase svoje vlastné číslo AS, tak túto trasu ignoruje. V opačnom prípade by vznikali smerovacie slučky. Ako bolo uvedené do tejto skupiny patrí práve protokol BGP.

Podľa správania sa delia protokoly na triedne [11] – využitie pevného adresného prefixu definovaného podľa triedy do ktorej patria dané sieťové adresy a beztriedne – možnosť využitia dĺžky adresného prefixu nezávislého na triednom adresovaní. Triedne protokoly ako RIPv1 a IGRP využívajú triedne IP adresovanie a sú uvádzané prevažne z historických dôvodov, lebo stoja za vznikom novších smerovacích protokolov. Protokol RIPv1 bol nahradený beztriednym protokolom RIPv2 a protokol IGRP bol nahradený rozšírenou verziou EIGRP. Všetky ostatné protokoly využívajú beztriedne adresovanie.

2.2 BGP – Border Gateway Protocol

Border Gateway Protocol (BGP) [10],[15] je smerovací protokol využívaný pre smerovanie medzi autonómnymi systémami. Jeho zaradenie podľa účelu je teda v skupine *Exterior Gateway Protocol – (EGP)* protokolov. BGP protokol sa podľa princípu činnosti zaraďuje medzi *Path-Vector Protocol*, ktorý ako smerovacie informácie využíva zoznam AS, ktoré

vedú k danej sieti. Na transportnej vrstve využíva BGP protokol TCP protokol, ktorý využíva port 179. Využitie TCP protokolu na transportnej vrstve eliminuje potrebu implementácie mechanizmov spojených so spoľahlivým doručením správ. TCP spojenie je vytvárané medzi smerovačmi v rámci susedstva. Každé BGP susedstvo využíva svoj vlastný konečný stavový automat, kde pre každú stranu existuje vlastná inštancia.

2.2.1 Správy využívané BGP protokolom

V tejto sekcii sú popísané formáty jednotlivých správ, ktoré využíva BGP protokol pre svoju funkčnosť.

BGP správy sú prenášané pomocou TCP spojenia a preto sú správy spracované, až keď sú kompletne prijaté. Maximálna veľkosť jednej BGP správy je stanovená na 4096 oktetov a preto je dané, že všetky korektné implementácie musia byť schopné spracovať správy o tejto maximálnej veľkosti. Najmenšia veľkosť správy, ktorá môže byť použitá je samotná BGP hlavička, ktorá má stanovenú veľkosť 19 oktetov.

Formát BGP hlavičky

Každá správa má hlavičku pevnej veľkosti. Na základe typu správy sa môžu za hlavičkou nachádzať dáta. Hlavička protokolu, viď obrázok č. 2.2, vyzerá nasledovne:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                                                               +
|                           Marker                              |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Length               |      Type     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Obr. 2.2: Formát BGP hlavičky [10].

- **Marker** – Značka, ktorú tvorí šestnásť oktetov, sú nastavené na hodnotu 1. Využíva sa z dôvodu kompatibility.
- **Length** – Dĺžka, tvorená dvomi celo-číselnými oktetmi určuje celkovú dĺžku správy, ktorá zahŕňa aj dĺžku hlavičky v oktetoch. Dĺžka musí byť v rozmedzí spomínaných 19 – 4096 oktetov.
- **Type** – jeden celo číselný oktet, ktorý reprezentuje typ správy. BGP podporuje nasledujúce typy BGP správ:

  1 – správa typu BGP Open

  2 – správa typu BGP Update

3 – správa typu BGP Notification

4 – správa typu BGP Keepalive

Formát správy BGP Open

Po nadviazaní TCP spojenia medzi smerovačmi je prvá správa zasielaná oboma smerovačmi práve správa *BGP Open*. Ak je táto správa akceptovaná ako potvrdenie je odoslaná správa typu *BGP Keepalive*.

Ako už bolo uvedené každá správa obsahuje hlavičku pevnej veľkosti. Za touto hlavičkou v prípade BGP Open správy nasledujú polia, viď obrázok č. 2.3:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+
|    Version    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     My Autonomous System      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Hold Time           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         BGP Identifier                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Opt Parm Len  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|             Optional Parameters (variable)                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Obr. 2.3: Formát BGP Open správy [10].

- **Version** – Verzia, veľkosť jedného celo-číselného oktetu, určuje verziu BGP protokolu tejto správy. Aktuálna verzia BGP protokolu je 4.
- **My Autonomous System** – Číslo Autonómneho Systému odosielateľa o veľkosti dvoch celo-číselných oktetov.
- **Hold Time** – Hodnota v sekundách o veľkosti dvoch celočíselných oktetov, určuje čas, po ktorý bude smerovač čakať na odpoveď od susedného smerovača ako reakciu na BGP Open správu.
- **BGP Identifier** – Identifikátor o veľkosti štyroch celočíselných oktetov. Identifikátor je reprezentovaný pomocou IP adresy, ktorá je priradená BGP smerovaču v rámci celého BGP procesu. Identifikátor je fixný od štartu a je zhodný pre všetky lokálne rozhrania smerovača a voči všetkým BGP susedom.
- **Optional Parameters Length** – jeden celo-číselný oktet udáva celkovú veľkosť voliteľných parametrov v oktetoch. Ak je tento parameter nastavený na 0, žiadne voliteľné parametre nie sú prenášané.

- **Optional Parameters** – Toto pole reprezentuje zoznam voliteľných parametrov, kde každý parameter je reprezentovaný ako trojica *<Parameter Type, Parameter Length, Parameter Value>*. Presný formát poľa voliteľných parametrov je uvedený na obrázku č. 2.4. **Parameter Type** je jeden oktet, ktorý jednoznačne identifikuje každý parameter. **Parameter Length** je jeden oktet reprezentujúci veľkosť údaju **Parameter Value** v oktetoch. Parameter Value je pole premennej veľkosti závislé na hodnote parametru Parameter Type.

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
|  Parm. Type   | Parm. Length  |  Parameter Value (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

Obr. 2.4: Formát voliteľných parametrov BGP Open správy [10].

Minimálna veľkosť **BGP Open** správy je 29 oktetov, táto veľkosť zahŕňa aj veľkosť BGP hlavičky.

Formát správy BGP Update

Správy typu *BGP Update* sú využívané pre prenos smerovacích informácií medzi BGP smerovačmi. Využívajú sa na oznamovanie dostupných trás, ktoré zdieľajú spoločnú trasu, danému susedovi alebo pre odstránenie niekoľkých neplatných trás, ktoré sa majú odstrániť zo smerovacieho procesu. V jednej BGP Update správe môžu byť umiestnené trasy, ktoré sa majú do smerovacieho procesu pridať aj tie, ktoré sa z neho majú odstrániť. BGP Update správa ako aj ostatné správy obsahuje BGP hlavičku pevnej veľkosti a okrem nej obsahuje nasledovné časti. Formát *BGP Update* správy je zobrazený na obrázku č. 2.5:

```
+-----------------------------------------------------+
|   Withdrawn Routes Length (2 octets)                |
+-----------------------------------------------------+
|   Withdrawn Routes (variable)                       |
+-----------------------------------------------------+
|   Total Path Attribute Length (2 octets)            |
+-----------------------------------------------------+
|   Path Attributes (variable)                        |
+-----------------------------------------------------+
|   Network Layer Reachability Information (variable) |
+-----------------------------------------------------+
```

Obr. 2.5: Formát BGP Update správy [10].

- **Withdrawn Routes Length** – Pole o veľkosti dvoch celo-číselných oktetov určuje celkovú dĺžku poľa *Withdrawn Routes* udávanú v oktetoch. Ak je táto hodnota 0 znamená to, že žiadne trasy sa nebudú z procesu odstraňovať a pole *Withdrawn Routes* nie je v správe uvedené.

- **Withdrawn Routes** – Pole o variabilnej veľkosti, ktoré obsahuje zoznam IP adries, ktoré majú byť odstránené zo smerovacieho procesu. Každý IP adresný prefix v zozname je uvedený ako dvojica *<Length, Prefix>*, ktorá je popísaná nižšie a jej formát je zobrazený na obrázku č. 2.6:

```
+---------------------------+
|   Length (1 octet)        |
+---------------------------+
|   Prefix (variable)       |
+---------------------------+
```

Obr. 2.6: Prvok udávajúci IP prefix v poli *Withdrawn Routes* v BGP Update správe [10].

  Význam prvkov dvojice:

  – **Length** – Dĺžka, určuje počet bitov IP adresného prefixu. Prefix dĺžky 0 znamená prefix, ktorý sa zhoduje so všetkými IP adresami.

  – **Prefix** – Obsahuje IP adresný prefix.

- **Total Path Attribute Length** – Pole o veľkosti dvoch celo-číselných oktetov určuje celkovú dĺžku poľa *Path Attributes* udávanú v oktetoch. Ak je táto hodnota 0 žiadna *Network Layer Reachability Information* ani *Path Attributes* nie je v danej BGP Update správe prenášaná.

- **Path Attributes** – Je sekvencia variabilnej dĺžky, ktorá je uvedená v každej BGP Update správe okrem tej, ktorá obsahuje iba položky z poľa *Withdrawn Routes*. Každý atribút zo sekvencie *Path Attributes* je reprezentovaný ako trojica *<Attribute Type, Attribute Length, Attribute Value>* variabilnej dĺžky.

  Parameter *Attribute Type* je dvoj oktetové pole, kde prvý oktet značí *Attribute Flags* a druhý *Attribute Type Code*. Parameter Attribute Type je uvedený na obrázku č 2.7.

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attr. Flags  |Attr. Type Code|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Obr. 2.7: Formát parametru *Attribute Type* BGP Update správy [10].

  Popis jednotlivých bitov *Attribute Flags* oktetu je nasledovný:

  0. bit – (voliteľný) ak je 0, atribut je povinný ak je 1, atribut je voliteľný

  1. bit – ak je 0, voliteľný atribút je netranzitívny. Ak je 1, je tranzitívny. Pre povinné atribúty tranzitívny bit musí byť nastavený na 1.

  2. bit – udáva, či informácia obsiahnutá vo voliteľnom tranzitívnom atribúte je čiastočná (ak je 1), alebo kompletná (ak je 0). Ak sa jedná o povinný alebo voliteľný netranzitívny atribút, musí byť tento bit nastavený na 0.

3. bit – bit rozšírenej dĺžky udáva, či *Attribute Length* je jedno oktetový (ak je 0), alebo dvoj oktetový (ak je 1).

Zvyšné bity sú nepoužité, teda musia byť nastavené na 0 a v prijatej správe musia byť ignorované.

Zvyšné oktety sekvencie *Path Attributes* určujú *Attribute Value* a sú interpretované na základe polí popísaných vyššie. Podporované typy atribútov a ich hodnoty sú nasledovné:

- **ORIGIN** – kód 1 – Povinný atribút, ktorý definuje pôvod trasy. Jeho dátový oktet nadobúda nasledujúce hodnoty:

  0 – **IGP** – NLRI je z daného autonómneho systému

  1 – **EGP** – NLRI je získaná pomocou EGP protokolu

  2 – **UNCOMPLETE** – NLRI získaná iným spôsobom

- **AS_PATH** – kód 2 – Povinný atribút, ktorý je zložený zo sekvencie segmentov AS path (trasy). Každý segment sa skladá z trojice *<Path Segment Type, Path Segment Length, Path Segment Value>*.

  Path Segment Type je jeden oktet dlhý a má definované nasledujúce hodnoty:

  1 – **AS_SET** – nezoradená postupnosť autonómnych systémov, ktorými prešla daná *BGP Update* správa

  2 – **AS_SEQUENCE** – daná postupnosť je zoradená

  Path Segment Length – jeden oktet, ktorý určuje počet AS v poli *Path Segment Value*.

  Path Segment Value – Obsahuje jedno alebo viacero čísel AS, každé zakódované ako dvoj oktetové pole.

- **NEXT_HOP** – kód 3 – Povinný atribút, ktorý definuje IP adresu smerovača, ktorý bude použitý ako adresa ďalšieho skoku na ceste do cieľovej siete definovanej v NLRI danej *BGP Update* správy.

- **MULTI_EXIT_DISC** – kód 4 – Voliteľný netranzitívny štvor-oktetový číselný atribút. Hodnota atribútu môže BGP smerovač použiť pri rozhodovaciom procese, ak vyberá z niekoľkých záznamov smerujúcich do susedného autonómneho systému.

- **LOCAL_PREF** – kód 5 – Povinný štvor-oktetový číselný atribút. BGP smerovač využíva tento atribút, aby informoval ostatných BGP susedov o preferencii danej trasy.

- **ATOMIC_AGGREGATE** – kód 6 – Je povinný atribút dĺžky nula.

- **AGGREGATOR** – kód 7 – Voliteľný tranzitívny atribút dĺžky šesť. Atribút obsahuje číslo posledného AS, ktorý vytváral agregovanú trasu (dva oktety), nasledovanú IP adresov BGP smerovača, ktorý vytváral agregovanú trasu (štyri oktety).

• **Network Layer Reachability Information** – Pole variabilnej veľkosti obsahuje zoznam IP adresných prefixov, ktoré sa majú pridať do BGP procesu. Jeho dĺžka je udávaná v oktetoch a nie je určená explicitne, ale musí sa vypočítať nasledovne:

celková dĺžka BGP Update správy - 23 - Total Path Attributes Length - Withdrawn Routes Length

kde celková dĺžka BGP Update správy je hodnota zakódovaná v hlavičke BGP správy, *Total Path Attributes Length* a *Withdrawn Routes Length* sú hodnoty zakódované vo variabilnej časti BGP Update správy. Hodnota 23 je kombinácia dĺžky BGP hlavičky, poľa *Total Path Attribute Length* a poľa *Withdrawn Routes Length*.

Pole IP adries, ktoré sa majú pridať do smerovacieho procesu je zakódované ako niekoľko dvojíc typu *<Length, Prefix>* a platí pre nich rovnaké pravidlá ako pre pole *Withdrawn Routes*, ktoré už bolo popísané.

Pre vytváranie správ typu **BGP Update** platí niekoľko pravidiel:

- Minimálna veľkosť správy typu BGP Update je 23 oktetov. 19 oktetov tvorí povinná hlavička BGP protokolu, dva oktety tvorí pole *Withdrawn Routes Length* a dva oktety *Total Path Attribute Length*. V takomto prípade sú obe polia nastavené na hodnotu 0.
- BGP Update správa môže oznamovať najviac jednu sekvenciu AS atribútov, ale viacero cieľových IP adries, kde však platí, že AS trasa je pre všetky IP adresy zhodná.
- BGP Update správa môže obsahovať viacero trás, ktoré majú byť odstránené zo smerovacieho procesu.
- Ak BGP Update správa oznamuje trasy, ktoré majú byť odstránené zo smerovacieho procesu, neobsahuje táto správa Path Attributes ani NLRI informácie a naopak ak oznamuje trasy ktoré majú byť pridané nemôže obsahovať trasy, ktoré sa majú odstrániť zo smerovacieho procesu.
- V BGP Update správe by nemal byť rovnaký IP prefix uvedený v zozname adries na odstránenie a na pridanie do BGP procesu.

Formát správy BGP Notification

Správy typu *BGP Notification* sú odosielané v prípade, že bola detekovaná nejaká chyba. V takom prípade je BGP spojenie okamžite ukončené.

Okrem povinnej BGP hlavičky obsahuje BGP Notification správa nasledujúce položky a formát je znázornený na obrázku č. 2.8:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Error code    | Error subcode |   Data (variable)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Obr. 2.8: Formát BGP Notification správy [10].

- **Error Code** – jedno oktetové celé číslo udáva typ BGP Notification správy. Protokol definuje nasledujúce chybové kódy:

  1 – **Message Header Error** – chyba v hlavičke BGP protokolu

  2 – **OPEN Message Error** – chyba v správe typu BGP Open

3 – **UPDATE Message Error** – chyba v správe typu BGP Update

4 – **Hold Timer Expired** – časovač dostupnosti vypršal

5 – **Finite State Machine Error** – chyba v konečnom automate BGP spojenia

6 – **Cease** – využíva sa v prípade, že BGP smerovač chce ukončiť spojenie so susedným smerovačom.

- **Error Subcode** – jedno oktetové celé číslo, ktoré bližšie špecifikuje vzniknutú chybu. Každy *Error Code* (chybový kód), môže mať jeden alebo viacero chybových subkódov. V prípade, že chybový kód nemá žiadny subkód, je v poli *Error Subcode* hodnota 0.
- **Message Header Error subcodes** - chybové subkódy BGP hlavičky:

  1 – **Connection Not Synchronized** – chyba, spojenie nie je synchronizované

  2 – **Bad Message Length** – chybná dĺžka BGP správy

  3 – **Bad Message Type** – chybný typ BGP správy

- **OPEN Message Error subcodes** - chybové subkódy BGP Open správy:

  1 – **Unsupported Version Number** – nepodporovaná verzia protokolu

  2 – **Bad Peer AS** – nesprávne číslo AS susedného smerovaču

  3 – **Bad BGP Identifier** – nesprávny BGP identifikátor

  4 – **Unsupported Optional Parameter** – nepodporovaný voliteľný parameter

  5 – **Deprecated** – tento parameter sa už nepoužíva

  6 – **Unacceptable Hold Time** – neakceptovaný čas *Hold Timer*-u

- **UPDATE Message Error subcodes** - chybové subkódy BGP Update správy:

  1 – **Malformed Attribute List** – poškodený zoznam atribútov

  2 – **Unrecognized Well-known Attribute** – nerozpoznaný povinný parameter

  3 – **Missing Well-known Attribute** – chýbajúci povinný parameter

  4 – **Attribute Flags Error** – nesprávny *Attribute Flags*

  5 – **Attribute Length Error** – nesprávna dĺzka atribútu

  6 – **Invalid ORIGIN Attribute** – nesprávna hodnota *ORIGIN* atribútu

  7 – **Deprecated** – tento parameter sa už nepoužíva

  8 – **Invalid NEXT_HOP Attribute** – chybný *NEXT_HOP* atribút

  9 – **Optional Attribute Error** – chyba voliteľného atribútu

  10 – **Invalid Network Field** – nesprávne pole *NLRI*

  11 – **Malformed AS_PATH** – poškodená trasa *AS_PATH*

- **Data** – Pole premenlivej veľkosti využité pre diagnostiku dôvodu vzniknutej správy typu *BGP Notification*. Pole je závislé na predchádzajúcich poliach *Error Code* a *Error Subcode*.
  Veľkosť dátového poľa je určená podľa nasledujúceho výpočtu:

Dĺžka dátového poľa = celková dĺžka BGP Notification správy - 21

Minimálna dĺžka správy **BGP Notification** správy je 21 oktetov. Táto veľkosť zahŕňa aj veľkosť BGP hlavičky.

Formát správy BGP Keepalive

BGP protokol nevyužíva žiadny mechanizmus založený na TCP protokole pre zistenie či sú susedia stále dostupní. Miesto toho si BGP smerovače vymieňajú správy typu *BGP Keepalive* a to tak často, aby nevypršal *Hold Timer* časovač, čo by značilo nedostupnosť susedného smerovača. Čas medzi zasielaním jednotlivých BGP Keepalive správ by mal byť maximálne jedna tretina z času *Hold Timeru* a BGP Keepalive správy by nemala byť zasielaná častejšie ako raz za sekundu.

V prípade, že dohodnutá hodnota *Hold Timeru* je 0, periodické *BGP Keepalive* správy nesmú byť zasielané.
Veľkosť správy **BGP Keepalive** je 19 oktetov a je tvorená iba BGP hlavičkou.

2.2.2 Dátové štruktúry BGP protokolu

Každý BGP smerovač prijíma smerovacie informácie od susedných BGP smerovačov. Prijaté informácie používa k lokálnemu spracovaniu a následne oznamuje vybrané smerovacie informácie ďalším susedným smerovačom. Pre možnosti vykonávania danej funkcionality BGP smerovača je využitá dátová štruktúra ktorá sa nazýva **BGP Routing Information Base (RIB)** [9] – báza smerovacích informácií.
BGP Routing Information Base sa skladá z troch hlavných častí:

- **Adj-RIBs-In** – V tejto dátovej štruktúre sú uložené smerovacie informácie pred samotným lokálnym spracovaním tak, ako boli prijaté od susedných BGP smerovačov a sú použité ako vstup pre BGP rozhodovací proces.

- **Local-RIB** – Lokálna databáza smerovacích informácií obsahuje smerovacie informácie, ktoré sú využité lokálne a sú výsledkom aplikovania filtrov, BGP postupov a rozhodovacieho procesu aplikovaného na smerovacie informácie uložené v *Adj-RIBs-In*.

- **Adj-RIBs-Out** – Posledná využívaná databáza obsahuje smerovacie informácie, ktoré boli vybrané lokálnym BGP smerovačom, aby boli odoslané jednotlivým susedným BGP smerovačom pomocou *BGP Update* správ. BGP smerovací protokol oznamuje iba najlepšie trasy, ak je povolené ich oznamovanie lokálnym procesom.

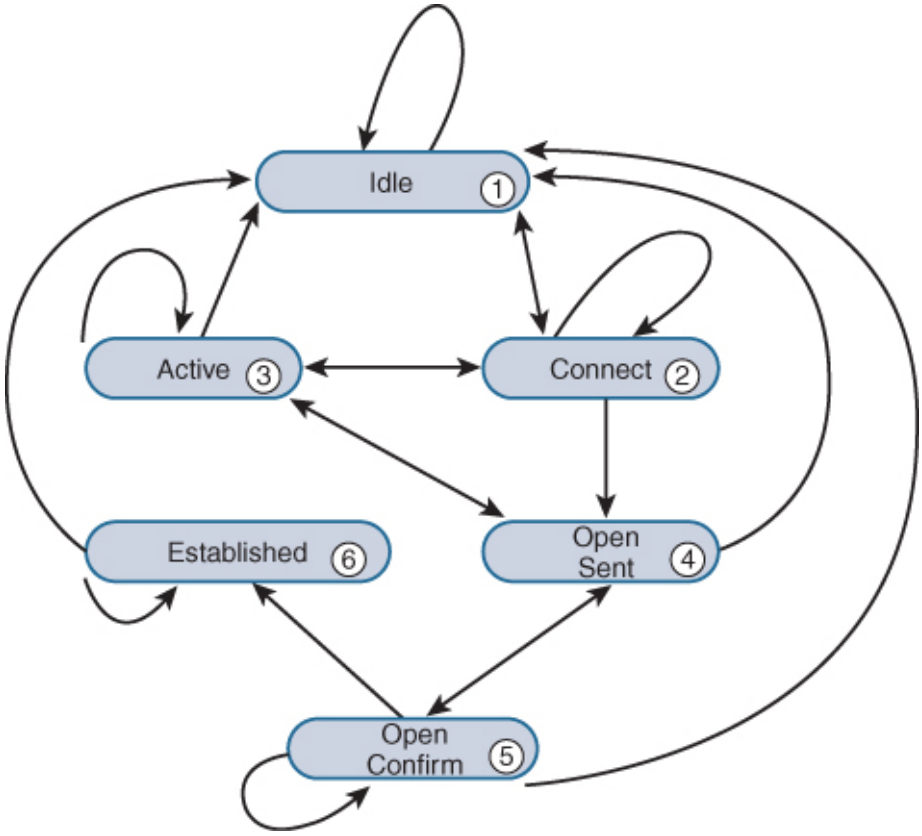
Dátové štruktúry sú využívané BGP procesom a nie sú využité pre preposielanie paketov. Iba trasy z *Local-RIB* sú následne vložené do smerovacej tabuľky daného smerovača, kde záleží na konkrétnych kritériách BGP smerovača tj. záleží na implementácií a preferenciách výrobcu pre daný smerovací protokol.

2.2.3 Stavový automat nadviazania spojenia so susednými smerovačmi

BGP smerovací protokol musí pre každý susedný BGP smerovač, nazývaný *BGP peer*, spravovať vlastný konečný stavový automat (*finite-state machine, FSM*) [16]. Každý BGP peer sa snaží nadviazať spojenie so svojimi susednými smerovačmi, ak nie je nakonfigurovaný tak, aby zostal v počiatočnom stave *Idle* alebo aby bol BGP smerovač pasívny.

BGP spojenie je nadviazané pomocou TCP spojenia na porte 179. To znamená, že smerovač, ktorý sa snaží nadviazať BGP spojenie so svojím susedom sa pokúsi pripojiť na port 179 a tak isto daný smerovač počúva na danom porte 179.

BGP spojenie sa môže nachádzať v jednom z nasledujúcich stavov a konečný stavový automat je zobrazený na obrázku č. 2.9:



Obr. 2.9: Konečný stavový automat BGP spojenia [16].

- **Idle** – Počiatočný stav konečného stavového automatu. V tomto stave začínajú všetky smerovače svoje BGP spojenia. BGP proces detekuje štartovaciu udalosť a snaží sa nadviazať TCP spojenie so svojimi susednými BGP smerovačmi a zároveň počúva na nadviazanie spojenia od susedných smerovačov.

  V prípade vzniknutej chyby, ktorá spôsobí, že sa BGP proces vráti do stavu *Idle*, musí smerovač vytrvať v tomto stave po dobu, kým nevyprší *ConnectRetryTimer*, ktorého hodnota je zvyčajne nastavená na dobu 60 sekúnd. Až po tejto dobe sa môže BGP smerovač znovu pokúsiť nadviazať BGP spojenie so susedným smerovačom.

- **Connect** – V tomto stave BGP proces iniciuje TCP spojenie. Keď prebehne 3-cestné TCP nadviazanie spojenia, tak BGP proces resetuje *ConnectRetryTimer* a odosiela správu *BGP Open* susednému smerovaču a prechádza do stavu *OpenSent*.

  Ak *ConnectRetryTimer* vyprší skôr ako sa dokončia operácie v danom stave, nové TCP nie je naviazané, *ConnectRetryTimer* je resetovaný na počiatočnú hodnotu a smerovač prechádza do stavu *Active*. V prípade, že je prijatá nežiadúca správa, smerovač prechádza do stavu *Idle*.

  V tomto stave smerovač s vyššou IP adresou nadväzuje spojenie. Ako zdrojový port smerovač použije ľubovoľný dynamický port a ako cieľový port musí byť použitý port 179.

- **Active** – V tomto stave BGP smerovač zahajuje nové 3-cestné TCP nadviazanie spojenia. Ak sa spojenie podarí nadviazať, smerovač odosiela správu *BGP Open*, na-

stavuje hodnotu *Hold Timer* časovaču na 4 minúty (doba po ktorú smerovač čaká na odpoveď na správu BGP Open) a prechádza do stavu *OpenSent*. Ak sa nadviazať spojenie znova nepodarí, smerovač prechádza späť do stavu *Connect* a resetuje *ConnectRetryTimer* na pôvodnú hodnotu.

- **OpenSent** – V danom stave smerovač odoslal *BGP Open* správu a čaká na príjem *BGP Open* správy od susedného smerovača. Po tom čo smerovač obdrží *BGP Open* správu od susedného smerovača je vykonaná jej kontrola na možné chyby a nekonzistencie. V priebehu kontroly sú sledované nasledujúce atribúty:
  - Musí sa zhodovať verzia BGP protokolu.
  - Zdrojová IP adresa musí byť zhodná s IP adresov susedného smerovača.
  - Číslo autonómneho systému v správe sa musí zhodovať s číslom autonómneho systému susedného smerovača.
  - BGP identifikátor (*RID*) musí byť unikátny.
  - Sú kontrolované bezpečnostné parametre ako heslá, TTL...

  Ak *BGP Open* správa neobsahuje žiadne chyby je odoslaná správa typu *BGP Keepalive*. Stav BGP procesu sa zmení na stav *OpenConfirm*.

  Ak sa v správe *BGP Open* nachádzajú chyby, je odoslaná správa typu *BGP Notification* a stav je zmenený na *Idle*.

  V prípade, že smerovač obdrží správu TCP disconnect, BGP proces ukončuje spojenie, resetuje *ConnectRetryTimer* a prechádza do stavu *Active*. Ľubovoľná iná správa prijatá v tomto stave spôsobí prechod do stavu *Idle*.

- **OpenConfirm** – V tomto stave, BGP smerovač čaká na príjem správy *BGP Keepalive* alebo *BGP Notification*.

  Ak smerovač obdrží správu *BGP Keepalive*, prechádza do stavu *Established*.

  Ak v priebehu čakania vyprší *Hold Timer* časovač, nastane udalosť *Stop Event* alebo je prijatá správa *BGP Notification* smení sa stav na *Idle*.

- **Established** – V tomto stave je nadviazané BGP spojenie. BGP smerovače si začínajú vymieňať smerovacie informácie pomocou správ typu *BGP Update*.
  S každým obdržaním správy *BGP Update* alebo *BGP Keepalive* je resetovaný *Hold Timer* časovač. V prípade, že tento časovač vyprší, je detekovaná chyba BGP spojenia a BGP smerovač prechádza späť do stavu *Idle*.

2.3 Rozšírenie protokolu BGP o podporu multi address-family smerovania

Rozšírenie protokolu BGP o možnosť prenosu smerovacích informácií pre viacero protokolov sieťovej vrstvy ako napríklad *IPv6, IPX, L3VPN* a pod [4]. Uvedené rozšírenie zaisťuje spätnú kompatibilitu, čo znamená, že smerovač, ktorý podporuje rozšírenie pre viacero protokolov musí byť schopný komunikovať so smerovačom, ktorý toto rozšírenie nepodporuje.

Pre rozšírenie BGP protokolu o podporu viacerých protokolov sieťovej vrstvy, musia byť pridané do protokolu dve funkcionality: 1. schopnosť priradiť príslušný protokol sieťovej

vrstvy danej informácií ďalšieho skoku (next hop information) a 2. schopnosť priradiť daný protokol príslušnému *NLRI*.

Pre poskytnutie spätnej kompatibility a podporu multiprotokolu sú pre BGP protokol využité dva nové atribúty:

- **Multiprotocol Reachable NLRI – MP_REACH_NLRI** – využitý pre prenos zoznamu dostupných cieľových sietí spolu s informáciou následného skoku (next hop), ktorý bude použitý pre dané cieľové siete.
- **Multiprotocol Unreachable NLRI – MP_UNREACH_NLRI** – využitý pre prenos zoznamu nedostupných cieľových sietí

Oba tieto atribúty sú voliteľné a netranzivívne. Preto BGP smerovač, ktorý nepodporuje toto rozšírenie jednoducho ignoruje informácie prenášané v týchto atribútoch a nebude ich preposielať ostatným BGP smerovačom.

Multiprotocol Reachable NLRI – MP_REACH_NLRI – kód 14

Ako už bolo spomenuté jedná sa o voliteľný netranzitívny atribút, ktorý sa využíva k nasledujúcim účelom: 1. oznamuje dostupné trasy susedným smerovačom a 2. umožňuje smerovaču oznamovať sieťovú adresu smerovača, ktorá má byť využitá ako next hop do cieľových staníc uvedených v poli *Network Layer Reachability Information MP_NLRI* atribútu. Štruktúra atribútu je zobrazená na obrázku č. 2.10 a vyzerá nasledovne:

```
+-------------------------------------------------------+
| Address Family Identifier (2 octets)                  |
+-------------------------------------------------------+
| Subsequent Address Family Identifier (1 octet)        |
+-------------------------------------------------------+
| Length of Next Hop Network Address (1 octet)          |
+-------------------------------------------------------+
| Network Address of Next Hop (variable)                |
+-------------------------------------------------------+
| Reserved (1 octet)                                    |
+-------------------------------------------------------+
| Network Layer Reachability Information (variable)     |
+-------------------------------------------------------+
```

Obr. 2.10: Formát atribútu *MP_REACH_NLRI* [4].

- **Address Family Identifier (AFI)** – v kombinácií s parametrom *SAFI* identifikuje množinu sieťových protokolov, do ktorých musí patriť sieťová adresa next hopu, spôsobom akým je zakódovaná táto adresa a sémantiku NLRI informácie. Hlavné AFI hodnoty sú zobrazené v tabuľke č. 2.1.
- **Subsequent Address Family Identifier (SAFI)** – využíva sa v kombinácií s *AFI* a preto pre tento parameter platia rovnaké charakteristiky.
- **Length of Next Hop Network Address** – jedno oktetové celé číslo, ktoré určuje dĺžku *Network Address of Next Hop* poľa v oktetoch.

Identifikátor skupiny	Popis skupiny
0	Reserved
1	IP (IP version 4)
2	IP6 (IP version 6)

Tabuľka 2.1: Základné hodnoty parametru AFI [3].

- **Network Address of Next Hop** – Pole premenlivej dĺžky, ktoré obsahuje sieťovú adresu next hop smerovača na ceste do cieľovej siete. Sieťový protokol, ktorý využíva adresa next hopu je identifikovaný dvojicou *<AFI, SAFI>*.
- **Reserved** – jeden oktet, ktorý je nastavený na hodnotu 0 a je ignorovaný príjemcom.
- **Network Layer Reachability Information** – zoznam premenlivej dĺžky obsahuje dostupné trasy, ktoré sú oznamované v tomto atribúte. Sémantika *NLRI* poľa je určená ako dvojica *<AFI, SAFI>*.

Next hop prenášaný v *MP_REACH_NLRI* definuje sieťovú adresu smerovača, ktorý môže byť použitý ako adresa skoku uvedená v *MP_NLRI* atribúte v správe typu *BGP Update*.

BGP Update správa, ktorá neprenáša iné *NLRI* ako to, ktoré je zakódované v *MP_REACH_NLRI* atribúte by nemala prenášať *NEXT_HOP* atribút. Ak takáto správa obsahuje atribút *NEXT_HOP*, BGP smerovač, ktorý obdrží danú správu tento atribút ignoruje.

BGP Update správa nesmie obsahovať rovnaký adresný prefix (*<AFI,SAFI>*) v nasledujúcich atribútoch: *WITHDRAWN ROUTES*, *Network Reachability Information*, *MP_REACH_NLRI* a *MP_UNREACH_NLRI*.

Multiprotocol Unreachable NLRI – MP_UNREACH_NLRI – kód 15

Jedná sa o voliteľný netranzitívny atribút, ktorý sa používa pre účel odstránenia viacero nedostupných trás zo smerovacieho procesu. Štruktúra daného atribútu je zobrazená na obrázku č. 2.11:

```
+-------------------------------------------------------+
| Address Family Identifier (2 octets)                  |
+-------------------------------------------------------+
| Subsequent Address Family Identifier (1 octet)        |
+-------------------------------------------------------+
| Withdrawn Routes (variable)                           |
+-------------------------------------------------------+
```

Obr. 2.11: Formát atribútu *MP_UNREACH_NLRI* [4].

- **Address Family Identifier (AFI)** – pre tento parameter platia rovnaké zásady ako pre parameter *AFI* z atribútu *MP_REACH_NLRI*.
- **Subsequent Address Family Identifier (SAFI)** – pre tento parameter platia tak isto rovnaké zásady ako pre parameter *SAFI* z atribútu *MP_REACH_NLRI* definovaného v predchádzajúcej podkapitole.

- **Withdrawn Routes** – zoznam premenlivej dĺžky obsahuje *NLRI* pre trasy, ktoré majú byť odstránené zo smerovacieho pocesu. Sémantika je rovnaká ako v prípade poľa *NLRI* v atribúte *MP_REACH_NLRI*, čiže je to dvojica *<AFI, SAFI>*.

BGP Update správa, ktorá obsahuje atribút *MP_UNREACH_NLRI* nemusí obsahovať žiadny iný *path* atribút.

Kódovanie NLRI

Atribút *NLRI* je pole niekoľkých dvojíc typu *<Length, Prefix>* a platia pre ne rovnaké pravidlá ako boli popísane v kapitole 2.2.1 pre atribúty *NLRI* a *Withdrawn Routes*. S tým rozdielom, že atribút *Length* určuje dĺžku podľa príslušnej adresnej rodiny.

Spracovanie chýb

Ak BGP smerovač obdrží od susedného BGP smerovača správu *BGP Update*, ktorá obsahuje *MP_REACH_NLRI* alebo *MP_UNREACH_NLRI* atribút a smerovač zistí, že atribút je nesprávny, smerovač odstráni všetky BGP trasy prijaté od daného susedného smerovača, ktorých *AFI/SAFI* je rovnaké ako to, ktoré bolo prijaté v danej správe a obsahuje chybu.

Smerovač by následne mal pre danú BGP reláciu ignorovať všetky trasy, ktoré obsahujú dané chybné atribúty *AFI/SAFI* prijaté od daného susedného smerovača. Prípadne môže smerovač ukončiť danú BGP reláciu a vygenerovať správu *BGP Notification* s chybovým kódom obsahujúcim chybu *BGP Update* správy/chybu voliteľného atribútu.

Využitie BGP Capability Advertisement

BGP Capability Advertisement [5] je procedúra, ktorú by mal využiť smerovač, ktorý chce využívať rozšírenie o multiprotokoly sieťovej vrstvy pre zistenie dostupnosti daného rozšírenia na susednom smerovači.

Capabilities – kód 2 je parameter, ktorý je definovaný trojicou *<Capabilities Code, Capabilities Length, Capabilities Value>* zobrazenou na obrázku č. 2.12:

```
+-------------------------------+
| Capability Code (1 octet)     |
+-------------------------------+
| Capability Length (1 octet)   |
+-------------------------------+
| Capability Value (variable)   |
+-------------------------------+
```

Obr. 2.12: Formát atribútu *Capabilities* [4].

```
0       7      15      23      31
+-------+-------+-------+-------+
|      AFI      | Res.  | SAFI  |
+-------+-------+-------+-------+
```

Obr. 2.13: Formát atribútu *Capability Value* [4].

- **Capabilitiy Code** – parameter je nastavený na hodnotu 1, čo značí rozšírenie o multiprotokol.
- **Capability Length** – dĺžka *Capability Value* atribútu je nastavená na hodnotu 4.
- **Capabilities Value** – hodnota atribútu je zobrazená na obrázku č. 2.13 a je definovaná nasledovne:
  - **Address Family Identifier (AFI)** – dva oktety, kódované rovnakým spôsobom ako atribút *AFI* popísaný v kapitole 2.3.
  - **Reserved** – rezervovaný oktet nastavený odosielateľom na hodnotu 0, ktorý je ignorovaný príjemcom.
  - **Subsequent Address Family Identifier (SAFI)** – jeden oktet, pre ktorý platí to isté ako pre atribút *AFI*, že je kódovaný rovnakým spôsobom ako atribút *SAFI* popísaný v kapitole 2.3.

BGP smerovač, ktorý podporuje viacero dvojíc <*AFI, SAFI*> ich zahŕňa ako viacero parametrov *Capabilities* vo voliteľnom parametri *Capabilities*.

Pre ustavenie obojstrannej výmeny smerovacích informácií pre dvojice <*AFI, SAFI*> medzi smerovačmi, musí každý zo smerovačov podporovať rovnaké dvojice <*AFI, SAFI*> pomocou popísaného *BGP Capability Advertisement* mechanizmu.

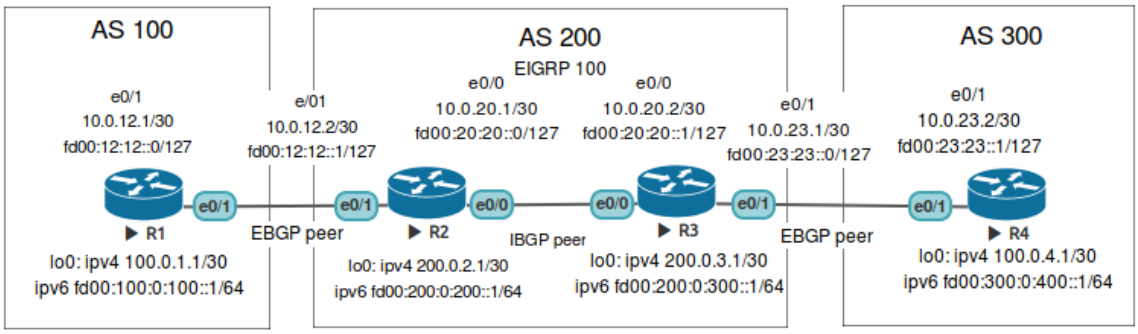
2.4 Konfigurácia BGP na Cisco zariadeniach

V nasledujúcej podkapitole je znázornené základné nastavenie protokolu BGP na Cisco zariadeniach. Cisco smerovače bežia na verzii operačného systému 15.4. Na obrázku č. 2.14 je zobrazená schéma topológie, na ktorej bude demonštrované spojazdnenie protokolu [7], [14], [6].

Topológia obsahuje štyri smerovače, z ktorých každý má nakonfigurované rozhranie typu *loopback* a rozhranie typu *ethernet*, slúžiace pre pripojenie s ostatnými zariadeniami v rámci topológie. Každé rozhranie má nakonfigurovanú *IPv4* a *IPv6* adresu podľa schémy.

Topológia obsahuje tri *autonómne systémy* – `AS 100`, `AS 200` a `AS 300`. V rámci `AS 200` je nakonfigurovaný *IGP* protokol *EIGRP* ako pre *IPv4*, tak aj pre *IPv6* konektivitu.

Medzi smerovačmi `R1 - (AS 100)` a `R2 - (AS 200)` bude nakonfigurované *externé BGP* susedstvo ako aj medzi smerovačmi `R3 - (AS 200)` a `R4 - (AS 300)`. Medzi smerovačmi `R2 a R3 (AS 200)` bude nakonfigurované *interné BGP* susedstvo.



Obr. 2.14: Schéma topológie pre popis konfigurácie protokolu BGP.

Pred samotnou konfiguráciou BGP susedstva je potreba zadať príkaz:

R(config)# ipv6 unicast-routing

na všetkých smerovačoch pre podporu `IPv6` smerovania.

Nadviazanie interného susedstva medzi smerovačmi

Nasleduje popis nadviazania susedstva v rámci `AS 200`, medzi smerovačmi `R2` a `R3`. Najskôr je potrebné spustiť samotný `BGP` proces na smerovači. To sa uskutoční zadaním príkazu v globálnom konfiguračnom móde:

R(config)# router bgp *autonomous-system-number*

Autonomous-system-number je číslo autonomného systému `AS`. V tomto prípade je AS 200.

R2(config)# router bgp 200

Následne je potrebné zadať príkaz pre definovanie susedov:

R(config-router)# neighbor *ip-address* **remote-as** *autonomous-system-number*

Kde *ip-address* je `IP` adresa susedného rozhrania pre smerovač `R2` je to adresa smerovača `R3` čiže `10.0.20.2`. Pomocou parametru *autonomous-system-number* v tomto prípade, je definované, či sa jedná o *interné* alebo *externé BGP* susedstvo:

R2(config-router)# neighbor 10.0.20.2 remote-as 200

V ideálnom prípade, sa pre nadviazanie susedstiev nevyužívajú fyzické rozhrania, ale rozhrani typu *loopback*, aby nebolo susedstvo medzi smerovačmi viazané na konkrétne fyzické rozhranie a to z dôvodu spoľahlivosti a možnosti pridania záložných a alternatívnych trás medzi *BGP* susedmi. V prípade použitia *loopback* rozhraní pre nadviazanie susedstva musia byť tieto rozhrania navzájom dostupné, a to buď pomocou statickej trasy, alebo musia byť zahrnuté do *IGP* smerovacieho protokolu. V tomto prípade sú *loopback* rozhrania v rámci `AS 200` zahrnuté v *EIGRP* protokole.

R(config-router)# neighbor *ip-address* **update-source** *interface*

Kde *ip-address* je `IP` adresa susedného `loopback` rozhrania, pre smerovač `R2` je to adresa smerovača `R3` čiže `200.0.3.1` a *interface* bude použitý `loopback0`.

R2(config-router)# neighbor 200.0.3.1 update-source loopback0

Overenie vytvoreného spojenia sa zaistí príkazom, viď obrázok č. 2.15:

R2# show ip bgp neighbors

```
R2#show ip bgp neighbors
BGP neighbor is 200.0.3.1,  remote AS 200, internal link
  BGP version 4, remote router ID 200.0.3.1
  BGP state = Established, up for 00:01:45
  Last read 00:00:04, last write 00:00:49, hold time is 180, keepalive interval
is 60 seconds
```

Obr. 2.15: Overenie nadviazania susedstva na smerovači R2.

Podpora multi-address family smerovania

Podpora *multi-address family* smerovania slúži na možnosť prenášania viacerých IP protokolov pod jedným *BGP* procesom. Mimo iného je možné nakonfigurovať podporu pre *IPv4* a *IPv6* smerovanie. Pred samotným nakonfigurovaním *multi-address family* smerovania je potrebné nakonfigurovať *IPv6* susedstvá pre *IPv6*. Konfigurácia je obdobná ako v prípade *IPv4*, popísaná v predchádzajúcej podkapitole s tým rozdielom, že sa využijú príslušné *IPv6* adresy dostupné podľa schémy topológie.

Do konfiguračného módu pre *address-family ipv4* je potrebné zadať nasledujúci príkaz v režime nastavovania *BGP* protokolu:

R2(config-router)# address-family ipv4 unicast

Následne je potrebné definovať, ktoré zariadenia s nadviazaným susedstvom majú byť aktívne v danej *address-family*.

R(config-router-af)# neighbor *ip-address* **activate**

Kde *ip-address* je analogicky adresa suseda. Pre daný príklad je to adresa *loopback* rozhrania smerovača `R3`:

R2(config-router-af)# neighbor 200.0.3.1 activate

Následne je potrebné definovať príkazom `network`, ktoré siete majú byť zahrnuté do *BGP* procesu:

R(config-router-af)# network *ip-address* **mask** *netmask*

Pre router `R2` je potrebné zadať nasledujúce príkazy.

R2(config-router-af)# network 200.0.2.1 mask 255.255.255.252

Obdobne je potrebná konfigurácia pre *address-family ipv6*

R2(config-router)# address-family ipv6 unicast

Analogicky je potrebné zadať príkazy `neighbor` *ipv6-address* `activate` a `network` *ipv6-*

address. Pre ukončenie sekcie *address-family* je potrebné zadať príkaz:

R2(config-router-af)# exit-address-family

Pre overenie správnosti nastavenia a výpis *BGP* smerovacej tabuľky pre *IPv4* a *IPv6* je potrebné zadať príkazy, viď obrázky 2.16 – 2.19:

R# show bgp [ipv4|ipv6] unicast summary
R# show ip bgp
R# show bgp ipv6 unicast

```
R2#show bgp ipv4 unicast summary
BGP router identifier 200.0.2.1, local AS number 200
BGP table version is 4, main routing table version 4
3 network entries using 420 bytes of memory
4 path entries using 320 bytes of memory
2/2 BGP path/bestpath attribute entries using 288 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 1028 total bytes of memory
BGP activity 4/0 prefixes, 5/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
200.0.3.1       4          200      15      15        4    0    0 00:09:34        2
```

Obr. 2.16: Overenie IPv4 konfigurácie na smerovači R2.

```
R2#show bgp ipv6 unicast summary
BGP router identifier 200.0.2.1, local AS number 200
BGP table version is 3, main routing table version 3
2 network entries using 328 bytes of memory
2 path entries using 208 bytes of memory
2/2 BGP path/bestpath attribute entries using 288 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 824 total bytes of memory
BGP activity 5/0 prefixes, 6/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
FD00:200:0:300::1
                4          200       7       7        3    0    0 00:02:11        1
```

Obr. 2.17: Overenie IPv6 konfigurácie na smerovači R2.

Nadviazanie externého susedstva medzi smerovačmi

Nadviazanie externého susedstva je analogické k internému susedstvu, ako bolo popísané v predchádzajúcich častiach, ale pre externé susedstvá je potrebné patrične doplniť konfiguráciu, aby proces fungoval korektne. Pretože sa vyžaduje, aby externé susedstvá boli priamo pripojené zariadenia a adresy, je žiadúce nadviazať externé susedstvo medzi *loopback* rozhraniami. To však vyžaduje nasledujúce doplnenie konfigurácie po špecifikovaní suseda príkazom `R(config-router)# neighbor` *ip-address* `remote-as` *autonomous-system-number*

```
R2#show ip bgp
BGP table version is 4, local router ID is 200.0.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 10.0.20.0/30     200.0.3.1                0    100      0 i
 *>                  0.0.0.0                  0         32768 i
 *>  200.0.2.0/30     0.0.0.0                  0         32768 i
 r>i 200.0.3.0/30     200.0.3.1                0    100      0 i
```

Obr. 2.18: Výpis BGP konfigurácie na smerovači R2 pre IPv4.

```
R2#show bgp ipv6 unicast
BGP table version is 9, local router ID is 200.0.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i FD00:20:20::/127 FD00:200:0:300::1
                                                0    100      0 i
 *>                  ::                       0         32768 i
 *>  FD00:200:0:200::/64
                      ::                        0         32768 i
 r>i FD00:200:0:300::/64
                      FD00:200:0:300::1
                                                0    100      0 i
```

Obr. 2.19: Výpis BGP konfigurácie na smerovači R2 pre IPv6.

a `R(config-router)#neighbor` *ip-address* `update-source` *interface* príkazom:

R2(config-router)# neighbor *ip-address* **ebgp-multihop** *number*

Kde *number* je počet skokov, na ktoré je daný sused vzdialený. V tomto prípade je počet skokov dva. Ako bolo popísané pri internom susedstve, adresy susedov musia byť známe v smerovacej tabuľke, inak sesedstvo nebude vytvorené. To platí aj v prípade externých susedstiev. Pre tento prípad je potrebné zadať statické cesty medzi *loopback* rozhraniami smerovačov, ktoré vyžadujú nadviazanie susedstva.

R(config)# ip route *ip-address mask next-hop*

Príklad pre smerovač `R1`:

R(config)# ip route 200.0.2.0 255.255.255.252 10.0.12.2
R(config)# ip route fd00:200:0:200::/64 fd00:12:12::3

Po nakonfigurovaní statických trás smerovače nadviažu externé susedstvo. Overenie konfigurácie je možné pomocou príkazu:

R# show ip bgp summary

Rozšírenie multi-address family konfigurácie pre externé susedstvá

Podpora *multi-address family* smerovania pre externé susedstvá sa konfiguruje zhodne ako pre interné susedstvá, ale keďže sa jedná o susedstvá medzi autonómnymi systémami je potrebné informovať susedné zariadenia o sieťach v rámci externého autonómneho systému. Toto je možné pomocou príkazu `network` alebo `redistribute` s príslušnými parametrami pre príslušné *address-family*. V tomto prípade bude konfigurácia pre `address-family [ipv4|ipv6] unicast` na smerovačoch `R2` a `R3` doplnená nasledujúcimi príkazmi:

R2(config-router-af)# redistribute eigrp 200

Príkaz `redistribute eigrp 200` zabezpečí, že smerovače `R1` a `R4` z externých *AS* sa dozvedia o sieťach z *EIGRP* procesu, ktorý beží v rámci `AS 200` a tým pádom bude nadviazaná konektivita `AS 100 - AS 200` a `AS 200 - AS 300`.

Aktualizovanie next-hop parametru pomocou príkazu next-hop-self

Parameter *next-hop-self* slúži pre zmenu `NEXT_HOP` parametru trasy, ktorá bola obdržaná z externého suseda na svoju vlastnú *IP* adresu. Tento príkaz je vhodné použiť, keď *IGP* protokol nepozná adresu *next-hop* externého suseda.

V tomto prípade smerovač `R2` obdržal od interného suseda smerovača `R3` informácie o autonomnom systéme `AS 300`. Smerovač `R2` nemá však `AS 300` dostupný, lebo nepozná adresu *next-hop*, ktorá sa nachádza na smerovači `R4`. Z toho dôvodu je potrebné doplniť konfiguráciu pre `address-family [ipv4|ipv6] unicast` o nasledujúce príkazy a analogicky pre smerovač R3.

R(config-router-af)# neighbor *ip-address* next-hop-self
R2(config-router-af)# neighbor 200.0.3.1 next-hop-self
R2(config-router-af)# neighbor fd00:200:0:300::1 next-hop-self

Po dokončení konfigurácie je dostupná konektivita v rámci celej topológie. Na obrázkoch 2.20 až 2.23, je zobrazený výstup overenia konektivity pomocou nástroju *ping* pre *IPv4* a *IPv6*. Ping je uskutočnený zo smerovača `R1` na smerovač `R4` a späť.

```
R1#ping 100.0.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.0.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Obr. 2.20: Overenie konektivity v rámci IPv4 topológie z R1 do R4.

```
R1#ping fd00:300:0:400::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FD00:300:0:400::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Obr. 2.21: Overenie konektivity v rámci IPv6 topológie z R1 do R4.

```
R4#ping 100.0.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.0.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Obr. 2.22: Overenie konektivity v rámci IPv4 topológie z R4 do R1.

```
R4#ping fd00:100:0:100::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FD00:100:0:100::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Obr. 2.23: Overenie konektivity v rámci IPv6 topológie z R4 do R1.

Kapitola 3

Návrh a implementácia BGP

V tejto kapitole je popísané simulačné prostredie OMNeT++, jeho rozšírenie o existujúce frameworky ANSAINET a INET. Následne je popísaný rozbor aktuálneho stavu implementácie v rámci INET frameworku. Je popísané aké obsahuje zdrojové súbory, rozobraný konfiguračný *.xml* súbor. Ako ďalšie je uvedený beh simulačných modelov v aktuálnom stave implementácie a na záver je spomínaný zoznam a popis zistených problémov.

3.1 Simulačné prostredie OMNeT++

3.1.1 OMNeT++

OMNeT++ [12],[13] je modulárne, diskrétne, objektovo orientovaná simulačná knižnica a framework slúžiaca primárne pre vytváranie sieťových simulácií. Vývojové prostredie OMNeT++ je vytvorené na základe vývojového prostredia Eclipse.

Sieťové simulácie sa skladajú z modulov, ktoré majú hierarchickú štruktúru. Jednotlivé moduly môžu byť súčasťou ďalších modulov. Moduly spolu komunikujú pomocou zasielania správ cez svoje vstupno-výstupné komunikačné kanály.

Model obsahuje niekoľko hlavných častí. Jednou z nich je *NED* súbor, ktorý popisuje aké moduly budu použité v danom modely, ich rozmiestnenie a vzájomné poprepájanie pomocou kanálov. Správanie a vlastnosti jednotlivých modulov je definované v samostatnom súbore implementovanom v jazyku C++.
Pre spustenie simulácie je v neposlednej rade potrebný konfiguračný súbor *omnetpp.ini*, špecifikuje parametre simulácie. Upresnenie týchto parametrov a externú konfiguráciu modulov podporuje aj pomocou jazyka *XML*, či už priamo v *.ini* súbore, alebo z externého *XML* súboru.

3.1.2 ANSAINET

Automated Network Simulation and Analysis (ANSA) [8] je projekt vyvýjaný na Fakulte informačních technológií Vysokého učení technického v Brne. ANSA rozširuje funkcionalitu frameworku *INET*, ktorý je taktiež súčasťou rozšírenia funkcionality prostredia OMNeT++.
INET [1] rozširuje funkcionalitu OMNeTu o simuláciu protokolov nad *TCP/IP*. Poskytuje podporu pre protokoly sieťovej vrstvy - *IPv4, IPv6, OSPF, BGP* a iné. Taktiež rozširuje funkcionalitu o drôtové a bezdrôtové protokoly linkovej vrstvy - *Ethernet, PPP, IEEE 802.11* a iné.

3.2 Aktuálny stav implementácie BGP protokolu

BGP protokol je z časti implementovaný vo frameworku *INET* vo verzii *BGPv4*. V tejto podkapitole je popísaný zistený, súčasný stav implementácie. Skúmaná verzia INET4 frameworku je **inet-4.0.0-576b2dc**.[1] Protokol sa snaží dodržiavať implementačné detaily v súlade s platným *RFC 4217* [10].

`BGP` je zahrnuté do frameworku INET v rámci balíku: **package inet.routing.bgpv4**, ktorý obsahuje zdrojové súbory implementácie.

- `bgpmessage/BgpHeader.msg` – súbor s popisom jednotlivých BGP správ, popis a podpora správy typu *BGP NOTIFICATION* celkom chýba.
- `Bgp.cc/h` – trieda, ktorá reprezentuje BGP protokol, zabezpečuje základnú funkčnosť, obsluhuje smerovaciu tabuľku ako aj BGP tabuľku, udržuje zoznam susedstiev a obsluhuje tabuľku rozhraní. Zaisťuje načítanie vstupnej konfigurácie z *.xml* súboru, ktorý je súčasťou simulačných modelov. Ďalej riadi prijímanie, rozhodovací proces a odosielanie BGP správ, na základe čoho riadi obsluhu konečných automatov jednotlivých BGP spojení.
- `BgpCommon.msg` – obsahuje informácie ohľadom BGP sessions, čiže ohľadom susedstiev.
- `BgpFsm.cc/h` – obsahuje implementáciu konečného automatu, ktorý reprezentuje proces nadviazania spojenia jednotlivých BGP smerovačov so svojími susedmi. Každý BGP smerovač si udržuje konečný automat pre každé BGP susedstvo. Ako už bolo spomenuté, v protokole nie je implementovaná podpora *notification* správ, preto nie je táto podpora implementovaná ani v konečnom automate a konečný automat nevykonáva prechody medzi stavmi v súdale s *RFC 4217 – sekcia 8.2.2. Finite State Machine*.
- `BgpRoutingTableEntry.h` – obsahuje deklarácie funkcií využitých pre prácu s BGP smerovaciou tabuľkou.
- `BgpSession.cc/h` – trieda reprezentujúca BGP spojenie, implementuje metódy, ktoré sa viažu na BGP susedstvo, tj. riadi časovače *(Hold, Connect Retry a Keep Alive)*, ustavuje spojenie so susedným BGP smerovačom, odosiela správy susedným smerovačom, udržuje štatistiky spojenia (koľko a akých správ bolo odoslaných a prijatých jednotlivým susedom).
- `Bgp.ned` – reprezentuje popis BGP modulu, jeho zaradenie v balíku v rámci **INETu** a popis jeho parametrov a rozhraní.

V protokole je implementovaná podpora pre povinné parametre BGP správ aj povinné reakcie na udalosti, ako aj samotné povinné udalosti s drobnými odchýlkami vzhľadom na *RFC 4217*. Podpora reakcie na chyby chýba. Tak isto chýba aj podpora IPv6.

[1] Dostupný na stiahnutie: https://github.com/inet-framework/inet/releases/download/v4.0.0/inet-4.0.0-src.tgz

Konfiguračný subor

Ako už bolo spomenuté, BGP protokol je konfigurovateľný pomocou *.xml* konfiguračného súboru, ktorý je súčasťou príkladov simulačných modelov. Názov súboru je špecifikovaný v inicializačnom súbore *.ini* pomocou parametru
****.bgpConfig = xmldoc("BGPConfig.xml")**, schéma konfiguračného súboru je zobrazená na obrázku č. 3.1 a má nasledujúcu štruktúru:
Na úvod musí byť špecifikovaný koreňový element **<BGPConfig />**. Nasledujú konfiguračné elementy:

- **<TimerParams> </TimerParams>** – element obsahuje v sebe zanorené elementy:
  - **<connectRetryTime> </connectRetryTime>** – hodnota *Connect Retry Timer* časovaču
  - **<holdTime> </holdTime>** – hodnota *Hold Timer* časovaču
  - **<keepAliveTime> </keepAliveTime>** – hodnota *Keep Alive* časovaču
  - **<startDelay> </startDelay>** – čas simulácie, v ktorom sa spúšťa BGP proces

V prípade, že nie je zadaný element **<TimerParams>**, spustenie simulácie končí s chybou. V prípade, že nie sú zadané parametre časovačov, simulácia sa spustí. Hoci časovače maju nastavené defaultné hodnoty v súbore **BgpSession.h**, tie sa prepíšu nulovými hodnotami a simulácia sa nespráva korektne.

Nasleduje element **<AS id="">** **</AS>**, ktorý obsahuje atribút *id*, ktorého hodnota označuje číslo autonómneho systému v rámci BGP protokolu. Ak je viac smerovačov nakonfigurovaných v jednom autonómnom systéme, bude sa medzi nimi nadväzovať interné BGP susedstvo. Element AS obsahuje zanorené elementy:
<Router interAddr="">/> – kde element *Router* označuje smerovač zahrnutý do BGP procesu v rámci daného autonómneho systému a atribút *interAddr* označuje *id* daného smerovača. Hodnota atribútu musí byť platná IP adresa nakonfigurovaná na jeho rozhraní, inak končí simulácia s chybovou hláškou pri inicializácií. Ďalej sa na tejto úrovni môžu nachádzať filtre pre obmedzenie BGP komunikácie. Tieto filtre sú:

- **<DenyRoute Address="" Netmask="">/>** – smerovač ignoruje trasu, ktorá je špecifikovaná atribútmi *Address* a *Netmask* smerom *IN* a *OUT*
- **<DenyRouteIN Address="" Netmask="">/>** – smerovač ignoruje trasu, ktorá je špecifikovaná atribútmi *Address* a *Netmask* smerom *IN*
- **<DenyRouteOUT Address="" Netmask="">/>** – smerovač ignoruje trasu, ktorá je špecifikovaná atribútmi *Address* a *Netmask* smerom *OUT*
- **<DenyAS> </DenyAS>** – smerovač ignoruje trasu naučenú z daného *AS* smerom *IN* a *OUT*, kde AS musí byť definované ako hodnota daného elementu.
- **<DenyASIN> </DenyASIN>** – smerovač ignoruje trasu naučenú z daného *AS* smerom *IN*, kde AS musí byť definované ako hodnota daného elementu.
- **<DenyASOUT> </DenyASOUT>** – smerovač ignoruje trasu naučenú z daného *AS* smerom *OUT*, kde AS musí byť definované ako hodnota daného elementu.

Smerovač vyhľadá svoj príslušný autonómny systém a v prípade, že sú nakonfigurované niektoré z uvedených filtrov, smerovač vyhľadá daný záznam vo svojej smerovacej tabuľke a vymaže príslušný záznam.

Ako ďalší element pre BGP konfiguráciu je v konfiguračnom súbore element **<Session id="">** **</Session>**, kde každý *Session* element označuje jedno externé BGP susedstvo. Preto v tomto elemente musia byť zanorené práve dva elementy **<Router exterAdd="" />**, kde hodnota atribútu *exterAddr* označuje IP adresu smerovača a tým identifikuje rozhranie, na ktorom má byť nadviazané externé BGP susedstvo. V prípade, že sa tu nenachádzajú práve dva elementy *Router*, simulácia nehlási chybu, ale nenadviaže sa dané susedstvo.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BGPConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="BGP.xsd">

    <TimerParams>
        <connectRetryTime> 120 </connectRetryTime>
        <holdTime> 180 </holdTime>
        <keepAliveTime> 60 </keepAliveTime>
        <startDelay> 15 </startDelay>
    </TimerParams>

    <AS id="100">
        <Router interAddr="100.0.1.1"/> <!--router R1-->
        <!-- DenyRoute: deny route in IN  and OUT traffic -->
        <!-- DenyRouteIN : deny route in IN traffic -->
        <!-- DenyRouteOUT: deny route in OUT traffic -->
        <!-- DenyAS: deny routes learned by AS in IN  and OUT traffic -->
        <!-- DenyASIN : deny routes learned by AS in IN traffic -->
        <!-- DenyASOUT: deny routes learned by AS in OUT traffic -->
    </AS>

    <AS id="200">
        <Router interAddr="10.0.20.1"/> <!--router R2-->
        <Router interAddr="10.0.20.2"/> <!--router R3-->
    </AS>

    <AS id="300">
        <Router interAddr="100.0.4.1"/> <!--router R4-->
    </AS>

   <Session id="1">
        <Router exterAddr = "10.0.12.1"/> <!--router R1-->
        <Router exterAddr = "10.0.12.2"/> <!--router R2-->
    </Session>

    <Session id="2">
        <Router exterAddr = "10.0.23.1"/> <!--router R3-->
        <Router exterAddr = "10.0.23.2"/> <!--router R4-->
    </Session>

</BGPConfig>
```

Obr. 3.1: Príklad BGP konfiguračného xml súboru.

Beh simulačných modelov

Aktuálne vytvorené simulačné modely sa nachádzajú v adresárovej štruktúre frameworku INET a to v: `inet4/examples/bgpv4`.

Zo spustených príkladov simulácií bolo vypozoravané nasledujúce správanie BGP procesu na smerovačoch. Správy, ktoré si medzi sebou smerovače vymieňajú počas nadviazania spojenia, sú zobrazené v prechodovom diagrame 3.2. BGP proces na smerovačoch je vždy plánovaný s patričným oneskorením a to z toho dôvodu, aby mali smerovače dostatočnú rezervu na skonvergovanie svojich smerovacích tabuliek. V prípade zložitejších príkladov sa jedná o oneskorenie 15 sekúnd.

Po nadviazaní BGP susedstva si smerovače odosielajú správy typu `BGP Update`. V týchto správach sa nachádzajú nekonzistencie, ktoré budú následne popísané v zozname zistených problémov a nekonzistencií. Smerovače automaticky oznamujú naučené siete cez BGP protokol svojím OSPF susedom, ktorí si pridajú danú trasu do svojej smerovacej tabuľky a označia ju ako `DIRECT OSPF`.

Siete, ktoré bude smerovač oznamovať pomocou BGP protokolu svojím susedom sú závislé na filtroch, ktoré sú pevne naimplementované v kóde. V daných simuláciách sú oznamované siete tie, ktoré sú získané pomocou `OSPF` protokolu a nie sú označené ako `External` preto aj najjednoduchší model, ktorý obsahuje v rámci AS iba jednu priamo pripojenú sieť vyžaduje konfiguráciu OSPF protokolu.

Zoznam zistených problémov a nekonzistencií implementácie:

- **Problém P1** – V prípade, že smerovače majú oznamovať susednému smerovaču iba jednu adresu/jeden prefix pomocou správy `BGP Update`, je všetko v poriadku, ak má smerovač oznamovať svojmu BGP susedovi viac sietí, odosiela smerovač iba jednu sieť čo je chybné správanie.

- **Problém P2** – V prípade, že má smerovač obdržať `BGP Update` správu, v ktorej, sa nachádza viac ako jedna adresa/prefix, smerovač je schopný spracovať iba jednu.

- **Problém P3** – *ConnectRetryTimer* sa neukončuje korektne ako má, čo má za následok, že ak sú aj susedné smerovače v korektnom stave a komunikujú spolu správne, po vypršaní ConnectRetryTimer-u sa ukončí TCP spojenie medzi smerovačmi a smerovače stratia TCP konektivitu.

- **Problém P4** – BgpNotificationMessage - tento typ BGP správy nie je vôbec podporovaný.

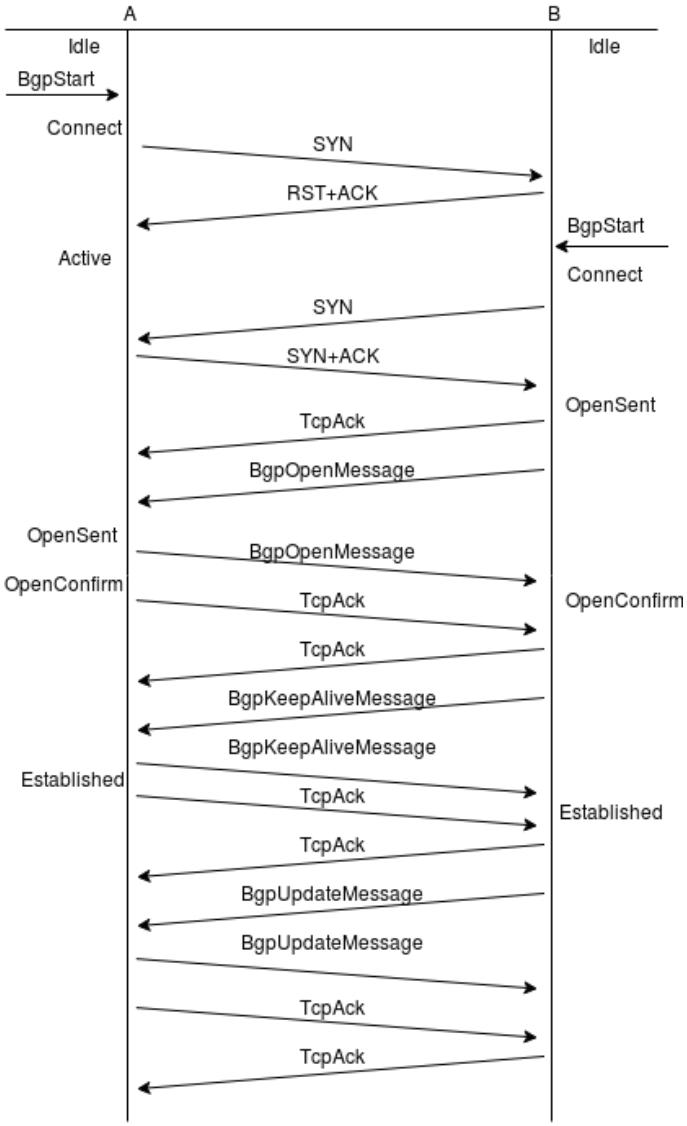
- **Problém P5** – Nekonzistencia niektorých stavov konečného automatu.

- **Problém P6** – Existujúca implementácia nepodporuje IPv6 protokol, je postavená iba na IPv4 sieťovom protokole.

- **Problém P7** – Implementácia simulačného modelu nepodporuje reakcie na výpadok spojenia medzi susednými smerovačmi.

- **Problém P8** – Implementácia simulačného modelu nie je schopná zotavenia sa po opätovnom obnovení linky.

- **Problém P9** – V prípade multipoint spojenia v rámci AS smerovač zahŕňa aj nesprávne siete do `BGP Update` správy.



Obr. 3.2: Diagram nadviazania BGP susedstva zo spusteného simulačného modelu.

- **Problém P10** – Smerovače pridávajú svoje číslo AS, aj keď sa odosiela trasa internému susedovi.
- **Problém P11** – V prípade, že neexistuje externé susedstvo v rámci smerovaču, nie je možné vytvoriť interné susedstvo s ľubovolným smerovačom.

3.3 Návrh

Návrh je zameraný na odstránenie problémov implementácie zistených zo spúšťania simulácií, ktoré vykazovali nekorektné správanie voči RFC a implementácií na Cisco smerovačoch.

Hlavnú časť bude tvoriť rozšírenie o *multi address-family* smerovanie. Po vzore implementácie popísanej v aktuálnom stave budú rozšírené triedy o metódy pracujúce s *IPv6*

adresami, a tak isto sa implementácia bude snažiť dodržať rozšírenie popísané v *RFC* [4] a konzistenciu voči implementácií na Cisco zariadeniach.

Týmto rozšírením bude snaha docieliť možnosť súčasne prenášať IPv4 a IPv6 cesty. V jednotlivých triedach bude potrebné doimplementovať podporu pre dátové štruktúry pracujúce s IPv6 adresami, čiže zahrnúť `IPv6` modul do modulu `BGP` smerovača, ktorý poskytuje `IPv6 smerovaciu tabuľku` a metódy pracujúce s touto tabuľkou. Vytvoriť `BGP smerovaciu tabuľku`, kde sa budú ukladať príslušné BGP záznamy pre IPv6 daného smerovača. Taktiež je potreba vymodelovať príslušné BGP správy – `BGP Open` a `BGP Update` pre podporu IPv6 protokolu a príslušných parametrov popísaných v kapitole 2.3. Ďalej je potrebné vytvoriť metódy, ktoré budú pracovať s týmito dátovými štruktúrami a logiku modelu tak, aby bola dosiahnutá daná funkcionalita.

V protokole BGP sa vytvárajú TCP spojenia. V implementácií BGP protokolu v prostredí INET je TCP spojenie súčasťou dátovej štruktúry, ktorá reprezentuje BGP spojenie `BGPSessions` a navyše zahŕňa informácie o IP adresách komunikujúcich smerovačov a portoch. Práve do tejto informačnej štruktúry, ktorá dopĺňa informácie daného spojenia bude pridaný príznak, či sa jedná o spojenie pomocou IPv4 alebo IPv6 sieťového protokolu. Týmto zmenám bude prispôsobená informačná dátová štruktúra, rozšírením o potrebné informácie. Operácie v rámci BGP protokolu sa vždy viažu ku konkrétnemu BGP spojeniu, to znamená na základe daného príznaku bude smerovač schopný rozlíšiť použité sieťové protokoly a teda či má odosielať, prijímať, spracovávať správy pomocou IPv4 alebo IPv6 protokolu a bude vedieť, ktoré smerovacie tabuľky má použiť.

Ako ďalšiu vec, ktorú bude treba odstrániť je závislosť implementácie na `OSPF` protokole. V aktuálnom stave je implementácia BGP protokolu závislá na OSPF protokole. OSPF protokol priamo určuje, ktoré siete majú byť oznamované susedným smerovačom pomocou protokolu BGP v správach `BGP Update`. Toto riešenie bude treba vylepšiť a odstrániť túto závislosť, a taktiež pridať možnosť užívateľsky nastaviť, ktoré siete majú byť zahrnuté do protokolu BGP pre konkrétne smerovače. Ďalší dôvod prečo odstrániť túto závislosť je to, že protokol OSPF zatiaľ nepodporuje *multi address-family* smerovanie, čo by tvorilo ďalšie limitácie hlavného cieľu tejto práce a tou je práve rozšírenie BGP implementácie o možnosť prenosu viacerých sieťových protokolov.

Podľa zhodnotenia aktuálneho stavu, je žiadané dosiahnuť lepšiu užívateľskú konfigurovateľnosť simulačných modelov. Preto je potrebné vytvoriť nový konfiguračný súbor, ktorý sa bude snažiť napodobniť konfiguráciu na Cisco zariadeniach a taktiež zjednotí niekoľko konfiguračných súborov do jedného. Nový konfiguračný súbor bude konfigurovať IPv4 a IPv6 adresy daných rozhraní smerovačov, statické trasy pre IPv4 a IPv6, ktoré nahradia vo funkcionalite OSPF protokol a samotnú konfiguráciu BGP protokolu pre `address-family` IPv4 a IPv6. V rámci konkrétnej `address-family` by malo byť možné zabezpečiť príslušnú konfigurácii susedných smerovačov a sietí, ktoré majú byť oznamované susedným BGP smerovačom. V neposlednej rade by mala byť zachovaná možnosť nakonfigurovať zakázanie konkrétnych IP adries alebo autonómnych systémov pre konkrétne smerovače v rámci adresnej rodiny.

Taktiež by bolo vhodné naimplementovať reakciu simulačného modelu na prípadný výpadok spojenia medzi susednými smerovačmi. V prípade, že smerovače zistia zlyhanie daného spojenia, odstránia príslušné trasy zo svojich smerovacích tabuliek a odošlú príslušnú `BGP Update` správu ostatným susedným smerovačom, v ktorej informujú smerovače o nedostupnosti daných sietí a aby odstránili záznamy zo svojích smerovacích tabuliek. K plnohodnotnému využitiu danej funkcionality je však vhodné doimplementovať podporu zotavenia

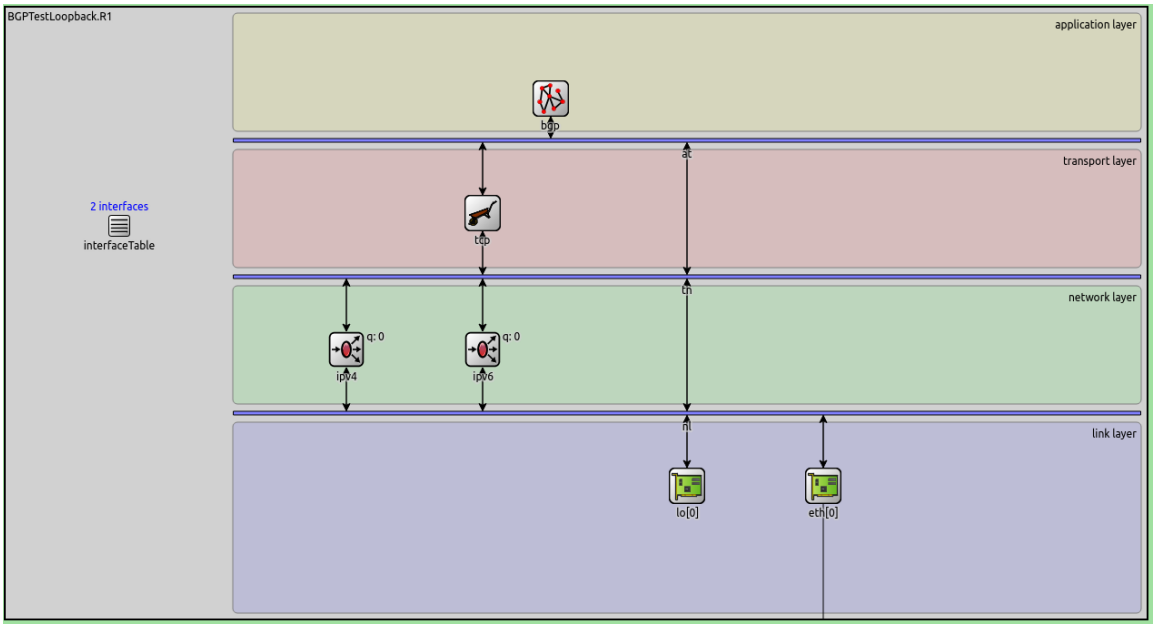
sa simulačného modelu po obnovení spojenia medzi smerovačmi. Funkcionalita by mala byť vo finále podporovaná ako pre IPv4 tak pre IPv6 sieťový protokol.

3.4 Implementácia

V rámci implementácie vznikol simulačný model, ktorý podporuje *multi address-family* smerovanie. Zdrojové kódy BGPv4 protokolu obohateného o dané rozšírenie sú dostupné vo verejnom repozitári[2]. Schéma takéhoto smerovača je zobrazená na obrázku č. 3.3. Pred samotnou implementáciou rozšírenia pre podporu IPv6 smerovania, bolo potrebné odstrániť problémy, ktoré boli zistené zo spúšťania simulačných modelov pre IPv4 sieťový protokol.

Prvý bol odstránený problém nesprávneho ukončovania `ConnectRetryTimer` časovača, pretože spôsoboval ukončenie TCP spojenia, aj keď bolo spojenie nadviazané korektne. `Problém P3` bol odstránený zmenou podmienky pre obnovenie tohoto časovača v triede `BgpSession` a v niektorých stavoch konečného automatu v triede `BgpFsm`.

Ako ďalšie prišlo na rad správne vytváranie a spracovávanie `BGP Update` správy. Problémy sú popísané v predchádzajúcej podkapitole ako `P1` a `P2`. Nesprávne vytváranie `BGP Update` správy spočívalo v neschopnosti správnej konfigurácie protokolu, ktorá pramenila zo závislosti na OSPF protokole a na trasách v smerovacej tabuľke získaných pomocou OSPF protokolu. Tento problém sa vyriešil odstránením závislosti na OSPF protokole a novým konfiguračným súborom, ktorý bude v tejto kapitole následne popísaný. Problém nesprávneho spracovávania `BGP Update` správy bol vyriešený cyklickým spracovaním prijatej správy na úrovni BGP protokolu v metóde `socketDataArrived` triedy `Bgp`.



Obr. 3.3: Schéma BGP smerovača v simulačnom prostredí OMNeT++.

[2]Dostupný na adrese: https://github.com/xnovak1j/DP-BGP.git

3.4.1 Konfiguračný súbor

Pre vyššiu prehľadnosť a schopnosť konfigurácie simulačných modelov bol vytvorený nový konfiguračný súbor. Schéma takéhoto súboru je zobrazená na obrázku č. 3.4. Z pôvodného súboru zostalo zachované nastavovanie časovačov pod elementom **<TimerParams> </TimerParams>** rovnako ako v pôvodnom znení. Nasleduje element **<Devices>**, ktorý obsahuje všetky smerovače v danom simulačnom modely. Pod elementom **<Router>** je obsiahnutá celková konfigurácia daného smerovača, ktorá sa skladá z niekoľko častí:

- **<Interfaces>** – zahŕňa konfiguráciu IPv4 a IPv6 adries na jednotlivé rozhrania, ktoré sú definované tagom **id**.
- **<Bgp>** – element zahŕňa konfiguráciu BGP protokolu, ktorá sa snaží čo najviac priblížiť konfigurácií na Cisco zariadeniach. Element **Bgp** obsahuje v sebe elementy **<Address-family>**, ktoré špecifikujú konfiguráciu pre daný sieťový protokol, čiže IPv4 alebo IPv6.

  V rámci **Address-family** sa podobne ako na Cisco zariadeniach konfigurujú elementom **<Neighbor>** susedné smerovače, na ktoré sa má nadviazať konkrétne BGP susedstvo. Elementom **<Network>** sa konfigurujú siete, ktoré sa majú daným susedným smerovačom oznamovať pomocou `BGP Update` správ. Tu je možno vidieť zlepšenie variability konfigurácie oproti pôvodnému konfiguračnému súboru, kde tieto siete nebolo možné nastaviť. V rámci danej `Address-family` je možno nakonfigurovať parametre s prefixom **<Deny.*>**. Tieto elementy slúžia pre zákaz šírenia daných adries smerom dnu alebo von a tak isto pre zákaz šírenia alebo spracovávania daných autonómnych systémov. Tieto parametre sa konfigurujú pre každú `address-family` samostatne.
- **<Route>** – element reprezentuje konfiguráciu statickej trasy pre IPv4 sieť.
- **<Route6>** – element reprezentuje konfiguráciu statickej trasy pre IPv6 sieť.

K novému konfiguračnému súboru bolo potrebné naimplementovať nový parser tohoto súboru, ktorý sa postará o správne naplnenie jednotlivých dátových štruktúr a o zahájenie príslušných susedstiev a procesov v rámci BGP protokolu. Konfiguračný súbor sa spracováva v zdrojovom súbore `Bgp.cc` v metóde `loadConfigFromXML`. Následne sa konfigurujú jednotlivé časti zvlášť. V metóde `loadTimerConfig` sa nastavia hodnoty časovačov. Metóda `routerIntfAndRouteConfig` má na starosti konfiguráciu príslušných IPv4 a IPv6 adries k daným rozhraniam. Konfigurácia adries je uložená v tabuľke rozhraní. Tieto adresy sa tak isto pridajú aj do príslušných smerovacích tabuliek pre IPv4 a IPv6 sieťové protokoly ako priamo pripojené linky. Poslednou časťou tejto metódy je spracovanie statických trás tým, že sa pridávajú do smerovacích tabuliek.

V metóde `loadBgpNodeConfig` sa spracováva konfigurácia spojená s BGP protokolom teda element `Bgp`. Najskôr sa naplnia vektory, ktoré obsahujú nepodporované adresy a čísla autonómnych systémov. Ako ďalší sa naplní vektor `_networksToAdvertise`, ktorý obsahuje príslušné adresy, ktoré sa majú oznamovať susedným smerovačom. V prípade, že sú tieto siete dostupné v smerovacej tabuľke, tak sa pridajú do BGP smerovacej tabuľky. Potom sa spracovávajú elementy `<Neighbor>`, ktoré sa na základe čísla AS delia na susedstvá interné a externé. Ak sa jedná o interné susedstvo, hodnota `remote-as` je zhodná s číslom AS daného smerovača, pridá sa adresa suseda do vektoru `_routerInSameASList`, ktorý je

```xml
<BGPConfig>
    <TimerParams>
        <connectRetryTime> 120 </connectRetryTime>
        <holdTime> 180 </holdTime>
        <keepAliveTime> 60 </keepAliveTime>
        <startDelay> 15 </startDelay>
    </TimerParams>
    <Devices>
        <Router name="R1" id="100.0.1.1">
            <Interfaces>
                <Interface id="lo0">
                    <Ipv4 address="100.0.1.1" netmask="255.255.255.252" />
                    <Ipv6 address="fd00:100:0:100::1/64" />
                </Interface>
                <Interface id="eth0">
                    <Ipv4 address="10.0.12.1" netmask="255.255.255.252" />
                    <Ipv6 address="fd00:12:12::0/127" />
                </Interface>
            </Interfaces>
            <Bgp as="100">
                <Address-family id="Ipv4">
                    <DenyRouteIN address="200.0.2.0" netmask="255.255.255.0"/>
                    <DenyRouteOUT address="172.10.8.0" netmask="255.255.255.0"/>
                    <DenyRoute address="172.10.8.0" netmask="255.255.255.0"/>
                    <DenyASIN as="200" />
                    <DenyASOUT as="100" />
                    <Neighbor address="10.0.12.2" remote-as="200" />
                    <Network address="100.0.1.0" />
                </Address-family>
                <Address-family id="Ipv6">
                    <DenyRouteIN address="fd00:12:12::0/127" />
                    <DenyRouteOUT address="fd00:12:12::0/127" />
                    <DenyRoute address="fd00:12:12::0/127" / >
                    <DenyAS as="300" />
                    <Neighbor address="fd00:12:12::1" remote-as="200" />
                    <Network address="fd00:100:0:100::" />
                </Address-family>
            </Bgp>
            <Route destination='200.0.3.0' netmask='255.255.255.252' interface='eth0' nexthop="10.0.12.2"/>
            <Route6 destination='fd00:200:0:300::/64' interface='eth0' nexthop="fd00:12:12::1"/>
        </Router>
    </Devices>
</BGPConfig>
```

Obr. 3.4: Schéma nového konfiguračného xml súboru pre BGP protokol.

potrebný pre vytvorenie všetkých interných susedstiev daného smerovača. V prípade, že sa jedná o externé susedstvo, toto susedstvo sa v danej metóde vytvorí a pridá sa do zoznamu všetkých susedstiev `_BGPSessions`. Táto časť funguje zhodne pre obe address-family s tým, že obsluhuje dátové štruktúry podľa protokolu.

Pre zachovanie určitej postupnosti akcií v rámci simulačných modelov sú vytvárané najskôr externé susedstvá. Po tom čo sú všetky externé susedstvá daného smerovača vytvorené, snaží sa smerovač vytvoriť svoje interné susedstvá. Interné BGP susedstvá sa vytvárajú ako posledná časť metódy `loadBgpNodeConfig`, po vykonaní predchádzajúcich operácií.

3.4.2 Odstránenie závislosti na OSPF protokole

Podľa predpokladov uvedených v návrhu je potreba sa zbaviť tejto závislosti a zachovať príslušnú funkcionalitu. Po odstránení modulu OSPF protokolu, bolo potrebné zabezpečiť funkcionalitu, ktorú poskytoval OSPF protokol iným spôsobom. Pre túto príčinu boli do implementácie BGP protokolu pridané statické trasy, ktoré sa pre konkrétny smerovač konfigurujú v konfiguračnom súbore spôsobom, ktorý bol popísaný v predchádzajúcej podkapitole. Taktiež bolo treba zaistiť oznamovanie správnych sietí susedným smerovačom. Na túto skutočnosť poslúži vektor `_networksToAdvertise`, ktorého prvky sa pri vytváraní

`BGP Update` správy vyhľadávajú v smerovacej tabuľke smerovača a pridajú sa do príslušného poľa `BGP Update` správy. Výber konkrétnych záznamov, ktoré sa majú odoslať susednému smerovaču sa vykonáva v stave `Established::entry` triedy `BgpFsm`. Samotné naplnenie príslušných polí správy `BGP Update` a zaradenie správy na odoslanie sa vykonáva v metóde `updateSendProcess` triedy `Bgp`.

Ďalej bol odstránený **problém P9**, kde v prípade multipoint spojenia v rámci AS, smerovač, ktorý mal odoslať smerovacie informácie susedovi v rámci interného susedstva, zahŕňal do týchto informácií aj tie, ktoré obdržal od iného interného suseda čo viedlo k nekonzistenciám v rámci topológie. Problém bol odstránený obmedzujúcou podmienkou pri preposielaní smerovacích informácií interným susedom v metóde `updateSendProcess` triedy `Bgp`.

3.4.3 Podpora multi address-family smerovania

Podľa požiadavok uvedených v podkapitole 3.3, ktoré sa opierajú o teoretické poznatky zhrnuté v podkapitole 2.3 bolo naimplementované dané rozšírenie BGP protokolu.

V prvom rade bolo potrebné naimplementovať dátové štruktúry pre prácu s IPv6 sieťovým protokolom. Implementácia začala vytvorením príslušných správ, ktoré sa líšili od pôvodných. Jedná sa o štruktúry `BGP Open` a `BGP Update`, ktoré sú vymodelované v súbore `BgpHeader.msg`.

Správa `BGP Open` bola rozšírená o voliteľný parameter `Multiprotocol_capability_extension`. Táto správa sa pre prenos IPv4 líši od prenosu IPv6 protokolu iba hodnotou uvedeného voliteľného parametru, preto stačí jedna štruktúra pre oba protokoly, ktoré sa odlišujú hodnotou `AFI` daného parametru.

`BGP Update` správa sa pre prenos IPv4 a IPv6 protokolu líši, a preto bola vytvorená nová správa pre prenos IPv6 protokolu. Nová štruktúra správy má v sebe zahrnutý voliteľný parameter `MP_REACH_NLRI`, v ktorom sú obsiahnuté údaje o `AFI` daného sieťového protokolu, teda protokolu IPv6. Ako ďalší parameter je `SAFI`, ktorý znázorňuje, že sa jedná o unicast. V neposlednom rade je obsiahnutá adresa následného skoku, v parameteri `nextHop` a siete oznamované susedným smerovačom v `NLRI` atribúte. Pre podporu reakcií na zmenu topológie bol do správy `BGP Update` taktiež pridaný ďalší voliteľný parameter `MP_UNREACH_NLRI`, ktorý obsahuje identifikátory `AFI` a `SAFI`. Hlavným atribútom tohoto parametru je atribút `Withdrawn Routes`, ktorý je kódovaný ako klasický atribút `NLRI` a obsahuje IPv6 adresu siete, ktorá sa má odstrániť z daného smerovacieho procesu. Správa `BGP Keepalive` je v oboch prípadoch zhodná a nebola potrebná jej úprava.

Po naimplementovaní správ je pridaná podpora `IPv6` modulu do BGP modulu. Nový BGP modul s podporou IPv6 protokolu rozširuje funkcionalitu existujúceho modulu `BgpRouter`. Modul BGP protokolu pre podporu *multi address-family* smerovania je nazvaný `BgpRouter6` a je súčasťou balíčka `inet.node.bgp`.

Protokol vyžaduje pre správnu funkcionalitu svoju pracovnú databázu, z toho dôvodu bola vytvorená BGP smerovacia tabuľka pre IPv6, ktorá je tvorená vektorom obsahujúcim jednotlivé záznamy tejto tabuľky. Záznam BGP tabuľky je implementovaný v súbore `BgpRoutingTableEntry6`.

Ako bolo uvedené v podkapitole 3.3, BGP spojenia v implementácií protokolu si udržujú informačnú štruktúru, ktorá presne špecifikuje dané susedstvo smerovačov. Každý z dvojice smerovačov si vytvára vlastné spojenia a vlastné informačné štruktúry. Pre rozšírenie funkčnosti daného spojenia o IPv6 protokol bola táto informačná štruktúra rozšírená, aby mohla byť využitá pre oba protokoly IPv4 a IPv6. Výsledná podoba štruktúry je zobrazená

na obrázku č. 3.5. BGP smerovač na základe príznaku `multiAddress` v tejto štruktúre vie rozoznať, ktorý protokol či IPv4 alebo IPv6 je v rámci daného spojenia použitý a vie na základe jeho hodnoty, ktoré smerovacie tabuľky má použiť. Do informačnej štruktúry boli taktiež pridané zoznamy IPv4 a IPv6 adries, `routesFromPeer` a `routesFromPeer6`, ktoré obdržal smerovač od daného suseda. Tieto dva vektory slúžia pre podporu reakcií na výpadok v topológii.

```
struct SessionInfo
{
    //support for multi address-family
    bool multiAddress = false;
    SessionId sessionID = 0;
    BgpSessionType sessionType = INCOMPLETE;
    AsId ASValue = 0;
    Ipv4Address routerID;
    Ipv4Address localAddr;
    Ipv4Address peerAddr;
    std::vector<Ipv4Address> routesFromPeer;
    std::vector<Ipv6Address> routesFromPeer6;
    Ipv6Address localAddr6;
    Ipv6Address peerAddr6;
    InterfaceEntry *linkIntf = nullptr;
    TcpSocket *socket = nullptr;
    TcpSocket *socketListen = nullptr;
    bool sessionEstablished = false;
};
```

Obr. 3.5: Informačná štruktúra, ktorá bližšie špecifikuje BGP susedstvo.

Po opravení zistených chýb IPv4 implementácie BGP protokolu, vytvorení nového konfiguračného súboru a vytvorení všetkých potrebných dátových štruktúr, ktoré pracujú s daným rozšírením, prichádza na rad popis implementácie funkčnosti rozšírenia o *multi address-family* smerovanie.

Po spracovaní konfiguračného súboru, z ktorého sa nakonfigurujú všetky potrebné informácie, ktoré boli popísané, sa prechádza k nadväzovaniu naplánovaných BGP susedstiev. Všetky susedstvá, ktoré bude smerovač nadväzovať sú uložené v mape `_BGPSessions`. Smerovače majú pre svoje susedstvá konečný stavový automat, ako bol popísaný v podkapitole 2.2.3. Tento automat je naimplementovaný v triede `BgpFsm`. Konečný automat pre daný smerovač je riadený z hlavnej triedy a tou je trieda `Bgp`.

Implementácia prenosu IPv4 a IPv6 protokolu pod jedným BGP procesom je možná na základe možnosti vytvoriť TCP spojenie, ktoré je nadviazané na konkrétnu IP adresu a dá sa využiť rozdelenie toku pre IPv4 a IPv6 protokoly na základe IP adries, ktoré su použité v rámci daného TCP prenosu. Smerovač na základe uvedeného príznaku `multiAddress` vie, na ktoré adresy sa má spojenie vytvoriť. Po vytvorení TCP spojenia sa odošlú `BGP Open` správy s príslušným nastavením `Capability` parametru. Po ich spracovaní sa odosielajú `BGP Keepalive` správy. Obe z nich sú vytvorené a odoslané z triedy `BgpSession`. Dané správy nevyžadujú prácu so žiadnou smerovacou tabuľkou, preto môžu byť vytvárané v príslušných metódach tejto triedy. Po prechode konečných automatov do stavu `Established` sú vytvorené a odoslané správy `BGP Update` pre príslušnú rodinu adries a odoslané susednému smerovaču na spracovanie. `BGP Update` štruktúry sú naplnené v metóde `updateSendProcess` a `updateSendProcess6` triedy `Bgp`, pretože potrebujú pracovať so smerovacími tabuľkami BGP protokolu. Po obdržaní `BGP Update` správy sa musí smerovač rozhodnúť ako s ňou

naloží. Preto tak ako je popísané v RFC [10] obsahuje trieda `Bgp` metódy, ktoré odpovedajú rozhodovaciemu procesu. V prípade, že sa obdržané siete od susedného smerovača pridávajú do smerovacej tabuľky a do tabuľky BGP procesu, tak sa pridajú aj do príslušného vektoru, ktorý obsahuje všetky siete obdržané od daného smerovača.

Po skonvergovaní topológie, teda po nadviazaní všetkých spojení, výmene všetkých `BGP Update` správ a aktualizovaní smerovacích tabuliek sa periodicky odosielajú `BGP Keepalive` správy, ktoré potvrdzujú živosť jednotlivých spojení.

Vo fáze zisťovaní aktuálneho stavu BGP implementácie bola skúmaná funkčnosť daných simulačných modelov a zisťovaná náväznosť na metódy v jednotlivých triedach. K jednotlivým metódam, ktoré pracovali výlučne s IPv4 sieťovým protokolom, boli vytvorené metódy podobnej funkcionality, ale s tým rozdielom, že využívali IPv6 protokol a dátové štruktúry pracujúce pre tento protokol. Novo vytvorené metódy kopírujú názov pôvodných metód s pridaným sufixom `6` do ich názvu. K tomu všetkému bola vytvorená obslužná logika, ktorá na základe už popísaného príznaku `multiAddress` rozhoduje, ktoré metódy sa majú použiť pre konkrétne susedstvo.

Vytvorený simulačný model je schopný reagovať na výpadky v topológii. V prípade, že smerovače zistia, že došlo k výpadku spojenia, tj. vyprší `Connect Retry Timer` alebo `Hold Down Timer` časovač. Smerovač následne skontroluje daný vektor sietí od suseda, s ktorým bolo prerušené spojenie a z príslušných stavov triedy `BgpFsm` sa zavolá metóda `updateSendProcess` triedy `Bgp` s príznakom odstránenia trasy. V tejto metóde sa naplnia odpovedajúce polia správy `BGP Update` a správa sa odosiela zvyšným susedným smerovačom. Tieto polia sú pre IPv4 `withdrawnRoutes` a pre IPv6 je to parameter `MP_UNREACH_NLRI`, ktorý už bol popísaný. Po odoslaní správ susedným smerovačom sa tieto siete odstránia z príslušných smerovacích tabuliek.

V prípade, že smerovač obdrží správu typu `BGP Update` v metóde `processMessage` zistí, či je veľkosť sietí na odstránenie rôzna od nuly. V kladnom prípade nastáva rovnaký proces ako v prípade predchádzajúcom, a to je odoslanie tejto informácie ďalším smerovačom a odstránenie príslušných záznamov zo smerovacej tabuľky.

Konečný automat susedných smerovačov, ktorých susedstvo bolo zrušené, prechádzajú svojimi stavmi do počiatočného stavu `Idle::init`. V prípade, že sa jedná o opätovný výskyt smerovaču v tomto stave, tj. `ConnectRetryCounter` nie je nuľový. Smerovače naplánujú opätovný pokus o nadviazanie spojenia o 15 sekúnd. Po uplynutí daného času sa smerovače pokúšajú znova nadviazať susedstvo a dosiahnuť skonvergovaný stav topológie.

Kapitola 4

Testovanie a porovnanie s reálnou topológiou

Po vykonaní implementácie prichádza na rad testovanie vzniknutého simulačného modelu. Pre účel testovania simulačného modelu voči reálnej topológii museli byť vytvorené zhodné topológie, ako v simulačnom prostredí OMNeT++, tak aj na reálnych zariadeniach. K vytvoreniu reálnej siete bude využitý školský virtualizovaný hardware `Ciscolab`. Testovanie a porovnávanie simulačného modelu sa skladá z dvoch hlavných častí:

- Porovnanie poradia a obsahu prenášaných správ v simulačnom modely a v reálnej topológii. Pre docielenie možnosti zobrazenia poradia a obsahu správ prenášaných v reálnej topológii musí byť využitý nástroj schopný zaznamenávať komunikáciu na danom rozhraní. V nástroji `Ciscolab` je možné využiť software `Wireshark`, ktorým bude odchytená komunikácia medzi smerovačmi. Poradie a obsah prenášaných správ v simulácií je súčasťou bežiacej simulácie prostredia `OMNeT++`, preto nie je potrebný iný nástroj pre odchytávanie komunikácie. Zachytená komunikácia obsahuje správy v poradí v akom prišli na dané rozhranie. To znamená, že poradie a obsah správ v simulácií by malo byť zhodné s poradím a obsahom správ v reálnej topológii.

- Porovnanie obsahu jednotlivých dátových štruktúr simulačného modelu a reálnej topológie. Pre získanie referenčných obsahov smerovacích tabuliek budú na reálnej topológii použité príkazy `show`, ktorých výstupy budú môcť byť porovnané s dátovými štruktúrami smerovačov simulačného modelu. Údaje v jednotlivých tabuľkách smerovačov simulačného modelu by mali byť veľmi podobné tomu, čo bude zobrazené v tabuľkách smerovačov reálnej topológie.

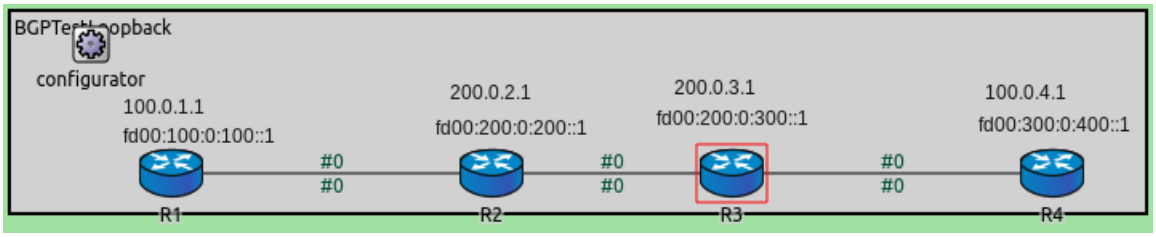
Predchádzajúce metódy budú v priebehu testovania kombinované za účelom overenia funkčnosti danej implementácie. K tomuto účelu vznikli dve testovacie topológie, v ktorých bude overovaná danú funkčnosť. Prvý model obsahuje topológiu, ktorá bola popísaná v kapitole 2.4 ale s tým rozdielom, že v rámci autonómneho systému 200 nebude použitý smerovací protokol `EIGRP`, ale jeho funkcionalita bola nahradená statickými trasami. V druhej topológii je AS 200 rozšírený o ďalší smerovač R5. V rámci AS 200 tým vzniká multipoint spojenie. K smerovaču R5 je pridaný externý susedný smerovač R6.

Obe topológie budú použité k testovaniu troch rôznych scenárov, ktoré majú za úlohu overiť funkčnosť implementácie vo viacerých prípadoch použitia ako sú: nadviazanie spojenia, strata spojenia a obnovenie spojenia po výpadku.

4.1 Testovacia topológia 1

Testovacia topológia sa skladá zo štyroch smerovačov R1–R4, ktoré sú pospájané za sebou R1–R2–R3–R4. Smerovač R1 sa nachádza v autonómnom systéme 100. Smerovače R2 a R3 sú v autonómnom systéme 200 a bude medzi nimi nadväzované interné susedstvo. Posledný smerovač R4 sa nachádza v autonómnom systéme 300. Medzi smerovačmi R1 – R2 a R3 – R4 je plánované externé susedstvo. Smerovače R1 a R4 budú svojim susedným smerovačom oznamovať jednu sieť a tou je sieť ich `Loopback 0` rozhraní. Smerovače R2 a R3 budú susedným externým smerovačom oznamovať sieť svojich `Loopback 0` rozhraní, ale aj siete `Loopback 0` rozhraní svojich interných susedov. Ako bolo spomenuté táto topológia bola zjednodušená a v rámci autonómneho systému 200 nie je aplikovaný protokol `EIGPR`, ale na smerovačoch sú nakonfigurované statické trasy. Smerovač R2 má nakonfigurovanú statickú trasu pre IPv4 a aj IPv6 protokol do siete smerovača R3 do siete, ktorú má smerovač R3 pripojenú na rozhranie `Loopback 0`. Smerovač R3 má statickú trasu nakonfigurovanú presne opačne, teda na sieť smerovača R2. Schéma upravenej topológie je zobrazená na obrázku č. 4.1. Testovacie scenáre overujú funkčnosť ako IPv4 tak aj IPv6 komunikácie v rámci BGP protokolu teda testuje funkčnosť *multi address-family* smerovania.

V schéme sú zobrazené IP adresy `Loopback 0` rozhraní jednotlivých smerovačov. Test je umiestnený v adresári `inet4/examples/bgpv4/BgpEx1Loopback` a simulácia obsahuje tri možné varianty, z ktorých je potrebné zvoliť si jednu pri spustení simulácie v prostredí OMNeT++.



Obr. 4.1: Topológia 1. simulačného príkladu v prostredí OMNeT++.

Počiatočný stav prvej topológie

Najskôr je vhodné uviesť počiatočný stav smerovacích tabuliek jednotlivých smerovačov, pred zahájením procesu nadviazania spojenia so susednými smerovačmi, v rámci BGP smerovacieho protokolu. Je žiadúce, uviesť tieto tabuľky z toho dôvodu, aby bolo dokázané, že smerovače simulačného modelu začínajú s rovnakým stavom svojích smerovacích tabuliek, ako smerovače reálnej referenčnej topológie a tým mohla byť overená správna funkčnosť simulačného modelu.

Stav smerovacích tabuliek smerovačov v topológií je uvedený v obrázkoch 4.2 až 4.9. Pre stručnosť a názornosť sú tu uvedené iba tabuľky smerovača R1. Je možno vidieť, že v reálnych smerovacích tabuľkách trasy označené ako `C` – priamo pripojené, odpovedajú tým, ktoré sú výstupom zo simulátora.

```
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.12.0/30 is directly connected, Ethernet0/1
L        10.0.12.1/32 is directly connected, Ethernet0/1
      100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        100.0.1.0/30 is directly connected, Loopback0
L        100.0.1.1/32 is directly connected, Loopback0
```

Obr. 4.2: Počiatočná smerovacia tabuľka R1 smerovača.

```
BGPTestLoopback.R1.ipv4.routingTable.routes (vector<Ipv4Route *>) size=2
elements[2] (inet::Ipv4Route *)
    [0] dest:10.0.12.0  gw:*  mask:255.255.255.252  metric:21 if:eth0(10.0.12.1)  DIRECT IFACENETMASK
    [1] dest:100.0.1.0  gw:*  mask:255.255.255.252  metric:21 if:lo0(100.0.1.1)  DIRECT IFACENETMASK
```

Obr. 4.3: Počiatočná smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

```
C   FD00:12:12::/127 [0/0]
     via Ethernet0/1, directly connected
L   FD00:12:12::/128 [0/0]
     via Ethernet0/1, receive
C   FD00:100:0:100::/64 [0/0]
     via Loopback0, directly connected
L   FD00:100:0:100::1/128 [0/0]
     via Loopback0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Obr. 4.4: Počiatočná IPv6 smerovacia tabuľka R1 smerovača.

```
BGPTestLoopback.R1.ipv6.routingTable.routeList (vector<Ipv6Route *>) size=3
elements[3] (inet::Ipv6Route *)
    [0] fd00:12:12::/127 --> if:eth0 next hop:<unspec> IFACENETMASK  DIRECT IFACENETMASK
    [1] fd00:100:0:100::/64 --> if:lo0 next hop:<unspec> IFACENETMASK  DIRECT IFACENETMASK
    [2] fe80::/10 --> if:eth0 next hop:<unspec> MANUAL  DIRECT MANUAL
```

Obr. 4.5: Počiatočná IPv6 smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

```
     Network          Next Hop            Metric LocPrf Weight Path
*    100.0.1.0/30     0.0.0.0                  0         32768 i
```

Obr. 4.6: Počiatočná BGP smerovacia tabuľka R1 smerovača.

```
BGPTestLoopback.R1.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=1
elements[1] (inet::bgp::RoutingTableEntry *)
    [0] BGP - Destination: 100.0.1.0/255.255.255.252 , PathType: IGP , NextHops: <unspec> , AS:
```

Obr. 4.7: Počiatočná BGP smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

```
    Network          Next Hop            Metric LocPrf Weight Path
*   FD00:100:0:100::/64
                      ::                      0         32768 i
```

Obr. 4.8: Počiatočná BGP IPv6 smerovacia tabuľka R1 smerovača.

```
▼ BGPTestLoopback.R1.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=1
  ▼ elements[1] (inet::bgp::RoutingTableEntry6 *)
      [0] BGP - Destination: fd00:100:0:100::/64 , PathType: IGP , NextHops: <unspec> , AS:
```

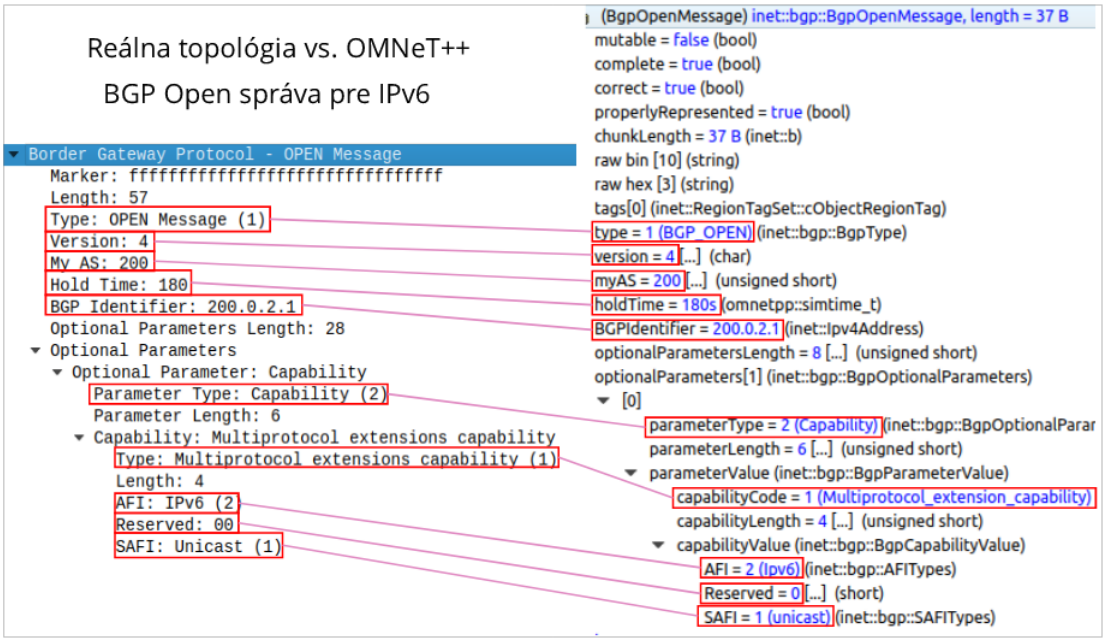
Obr. 4.9: Počiatočná BGP IPv6 smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

4.2 Topológia 1 – nadviazanie spojenia

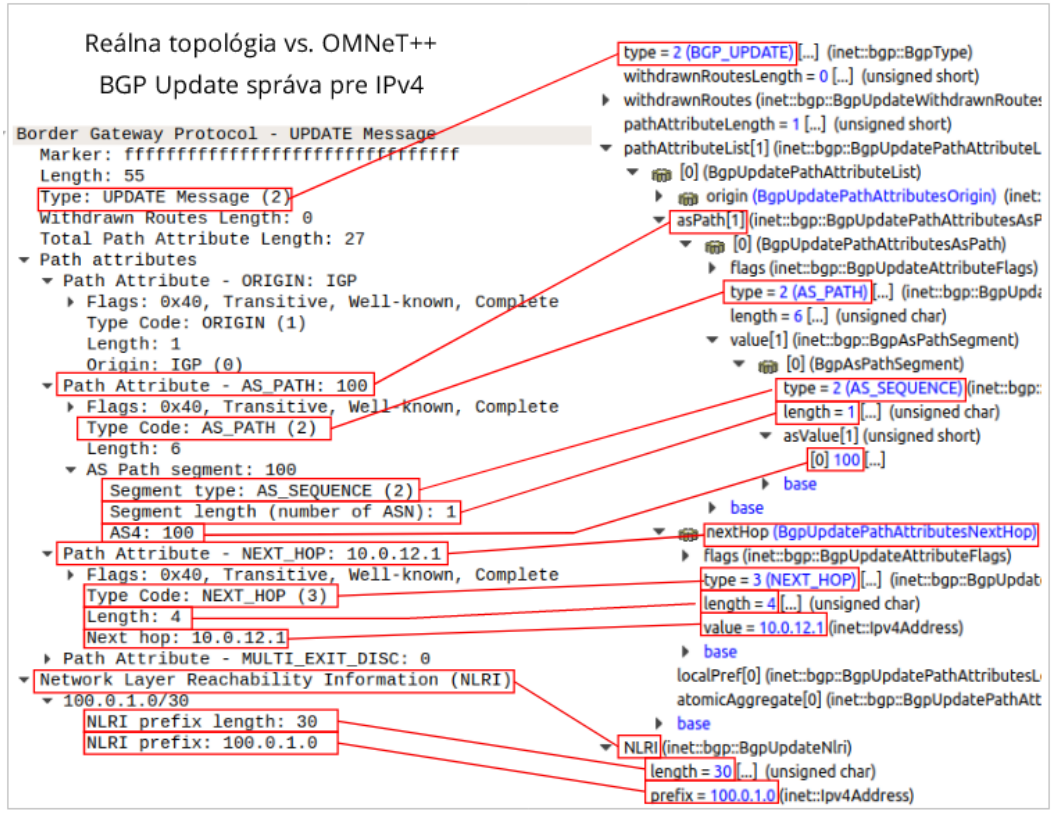
V prvom testovacom scenári bude sledované korektné nadviazanie susedstiev, výmena smerovacích informácií a aktualizácia smerovacích tabuliek smerovačov.

Po nadviazaní TCP spojenia si smerovače ako prvé odosielajú správu `BGP Open`. Tieto správy sú zobrazené na obrázku 4.10 pre názornosť sú zobrazené iba správy pre IPv6 protokol. Pre IPv4 protokol vyzerajú obdobne so zmenou hodnoty `AFI` parametru ako bolo popísané v predošlých kapitolách.

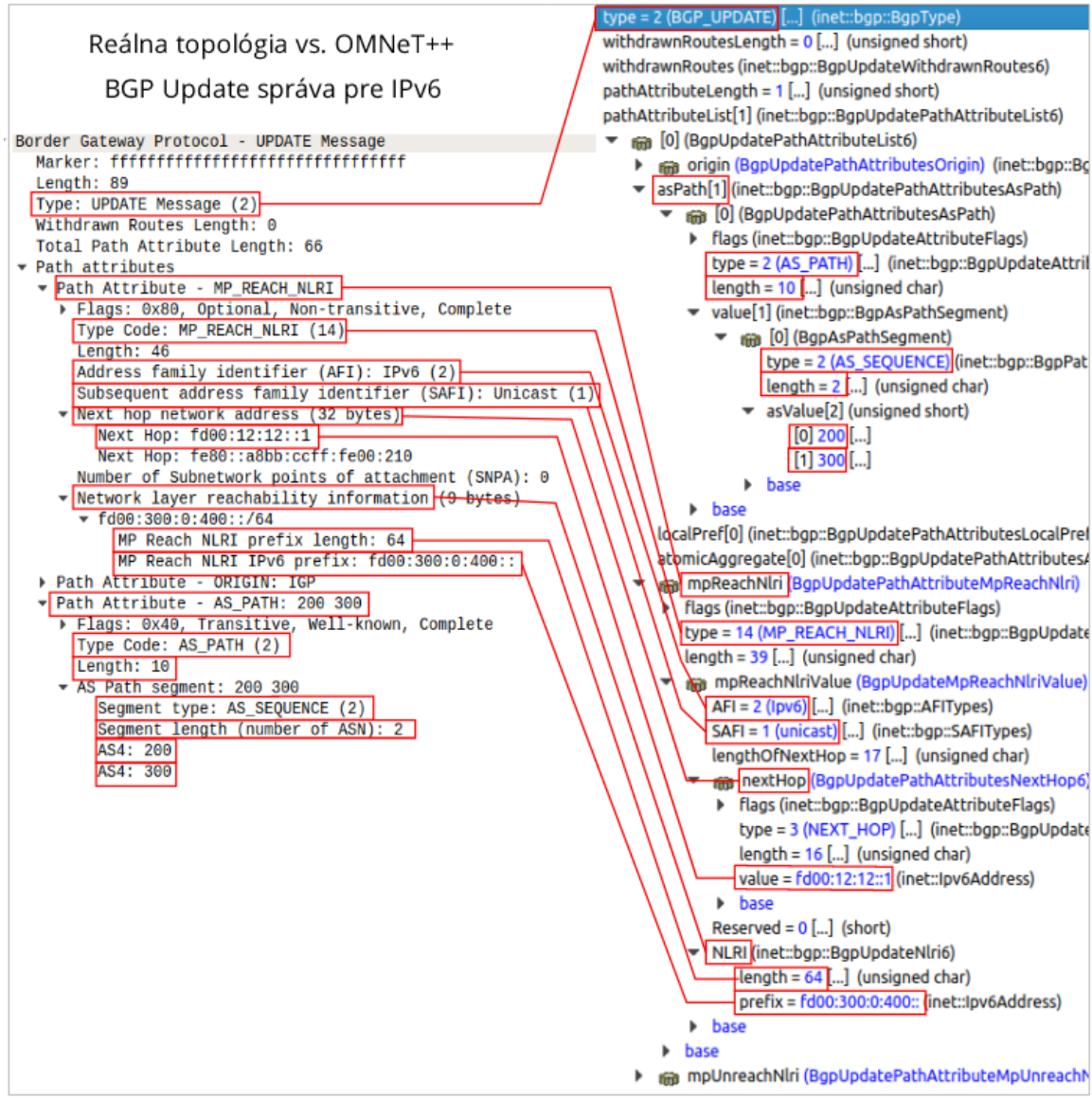
Následne si smerovače vymenia správy `BGP Keepalive`, ktoré ale obsahujú iba hlavičku BGP protokolu, preto tu nebudú uvedené. Po prechode do stavu `Established` si vymenia `BGP Update` správy a aktualizujú svoje smerovacie tabuľky. V teste sú uvedené `BGP Update` správy pre IPv4 zo smerovača R1, ktorý odosiela smerovaču R2 informácie a pre IPv6 je uvedená správa, ktorá bola odoslaná zo zariadenia R2 zariadeniu R1 s informáciou o sieti, ktorú propaguje R4. Každá správa obsahuje aj svoj ekvivalent z reálnej topológie. Dvojice správ sú zobrazené na obrázku 4.11 pre IPv4 sieťový protokol a na obrázku 4.12 pre sieťový protokol IPv6. Dané obsahy jednotlivých správ sú totožné, čo značí správnosť implementácie. Po obdržaní týchto správ nasleduje fáza spracovania nových smerovacích informácií a v prípade potreby aktualizácia príslušných tabuliek. Keďže každý smerovač pracuje so štyrmi tabuľkami, bolo by zobrazenie všetkých tabuliek v tejto kapitole veľmi neprehľadné, preto budú uvedené iba tabuľky zariadenia R1 zo simulačného prostredia s ekvivalentami zobrazenými v reálnej sieti voči ktorej sa daný test vykonáva. Smerovacie tabuľky skonvergovanej topológie sú uvedené na obrázkoch 4.13 až 4.20. Tabuľky pre zariadenie R1 obsahujú zhodné informácie zobrazené z reálnej topológie v porovnaní s tabuľkami získanými zo simulačného prostredia OMNeT++. Preto je možné vyvodiť záver, že smerovacie informácie v tabuľkách zo simulátoru sú validné. Test nadviazania spojenia a zaručenie konvergencie v sieti je dokončený úspešne pre obe adresné rodiny.



Obr. 4.10: Správa BGP Open pre prenos IPv6 protokolu z reálnej siete a zo simulačného prostredia OMNeT++.



Obr. 4.11: Správa BGP Update pre prenos IPv4 protokolu z reálnej siete a zo simulačného prostredia OMNeT++.



Obr. 4.12: Správa BGP Update pre prenos IPv6 protokolu z reálnej siete a zo simulačného prostredia OMNeT++.

```
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.12.0/30 is directly connected, Ethernet0/1
L        10.0.12.1/32 is directly connected, Ethernet0/1
      100.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        100.0.1.0/30 is directly connected, Loopback0
L        100.0.1.1/32 is directly connected, Loopback0
B        100.0.4.0/30 [20/0] via 10.0.12.2, 04:33:13
      200.0.2.0/30 is subnetted, 1 subnets
B        200.0.2.0 [20/0] via 10.0.12.2, 04:33:13
      200.0.3.0/30 is subnetted, 1 subnets
B  _     200.0.3.0 [20/0] via 10.0.12.2, 04:33:13
```

Obr. 4.13: IPv4 smerovacia tabuľka R1 smerovača.

```
BGPTestLoopback.R1.ipv4.routingTable.routes (vector<Ipv4Route *>) size=5
elements[5] (inet::Ipv4Route *)
    [0] dest:10.0.12.0 gw:* mask:255.255.255.252 metric:21 if:eth0(10.0.12.1) DIRECT IFACENETMASK
    [1] dest:100.0.1.0 gw:* mask:255.255.255.252 metric:21 if:lo0(100.0.1.1) DIRECT IFACENETMASK
    [2] dest:100.0.4.0 gw:10.0.12.2 mask:255.255.255.252 metric:1 if:eth0(10.0.12.1) REMOTE BGP
    [3] dest:200.0.2.0 gw:10.0.12.2 mask:255.255.255.252 metric:1 if:eth0(10.0.12.1) REMOTE BGP
    [4] dest:200.0.3.0 gw:10.0.12.2 mask:255.255.255.252 metric:1 if:eth0(10.0.12.1) REMOTE BGP
```

Obr. 4.14: IPv4 smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

```
C   FD00:12:12::/127 [0/0]
     via Ethernet0/1, directly connected
L   FD00:12:12::/128 [0/0]
     via Ethernet0/1, receive
C   FD00:100:0:100::/64 [0/0]
     via Loopback0, directly connected
L   FD00:100:0:100::1/128 [0/0]
     via Loopback0, receive
B   FD00:200:0:200::/64 [20/0]
     via FE80::A8BB:CCFF:FE00:210, Ethernet0/1
B   FD00:200:0:300::/64 [20/0]
     via FE80::A8BB:CCFF:FE00:210, Ethernet0/1
B   FD00:300:0:400::/64 [20/0]
     via FE80::A8BB:CCFF:FE00:210, Ethernet0/1
L   FF00::/8 [0/0]
     via Null0, receive
```

Obr. 4.15: IPv6 smerovacia tabuľka R1 smerovača.

```
BGPTestLoopback.R1.ipv6.routingTable.routeList (vector<Ipv6Route *>) size=6
elements[6] (inet::Ipv6Route *)
    [0] fd00:12:12::/127 --> if:eth0 next hop:<unspec> OWN_ADV_PREFIX DIRECT OWN_ADV_PREFIX
    [1] fd00:100:0:100::/64 --> if:lo0 next hop:<unspec> OWN_ADV_PREFIX DIRECT OWN_ADV_PREFIX
    [2] fd00:200:0:200::/64 --> if:eth0 next hop:fd00:12:12::1 BGP REMOTE BGP
    [3] fd00:200:0:300::/64 --> if:eth0 next hop:fd00:12:12::1 BGP REMOTE BGP
    [4] fd00:300:0:400::/64 --> if:eth0 next hop:fd00:12:12::1 BGP REMOTE BGP
    [5] fe80::/10 --> if:eth0 next hop:<unspec> MANUAL DIRECT MANUAL
```

Obr. 4.16: IPv6 smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   100.0.1.0/30     0.0.0.0                  0         32768 i
*>   100.0.4.0/30     10.0.12.2                                0 200 300 i
*>   200.0.2.0/30     10.0.12.2                0             0 200 i
*>   200.0.3.0/30     10.0.12.2                0             0 200 i
```

Obr. 4.17: BGP IPv4 smerovacia tabuľka R1 smerovača.

```
BGPTestLoopback.R1.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=4
  elements[4] (inet::bgp::RoutingTableEntry *)
    [0] BGP - Destination: 100.0.1.0/255.255.255.252 , PathType: IGP , NextHops: <unspec> , AS:
    [1] BGP - Destination: 200.0.2.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.12.2 , AS: 200
    [2] BGP - Destination: 200.0.3.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.12.2 , AS: 200
    [3] BGP - Destination: 100.0.4.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.12.2 , AS: 200 300
```

Obr. 4.18: BGP IPv4 smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   FD00:100:0:100::/64
                       ::                       0         32768 i
*>   FD00:200:0:200::/64
                       FD00:12:12::1            0             0 200 i
*>   FD00:200:0:300::/64
                       FD00:12:12::1            0             0 200 i
*>   FD00:300:0:400::/64
                       FD00:12:12::1                          0 200 300 i
```

Obr. 4.19: BGP IPv6 smerovacia tabuľka R1 smerovača.

```
BGPTestLoopback.R1.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=4
  elements[4] (inet::bgp::RoutingTableEntry6 *)
    [0] BGP - Destination: fd00:100:0:100::/64 , PathType: IGP , NextHops: <unspec> , AS:
    [1] BGP - Destination: fd00:200:0:200::/64 , PathType: EGP , NextHops: fd00:12:12::1 , AS: 200
    [2] BGP - Destination: fd00:200:0:300::/64 , PathType: EGP , NextHops: fd00:12:12::1 , AS: 200
    [3] BGP - Destination: fd00:300:0:400::/64 , PathType: EGP , NextHops: fd00:12:12::1 , AS: 200 300
```

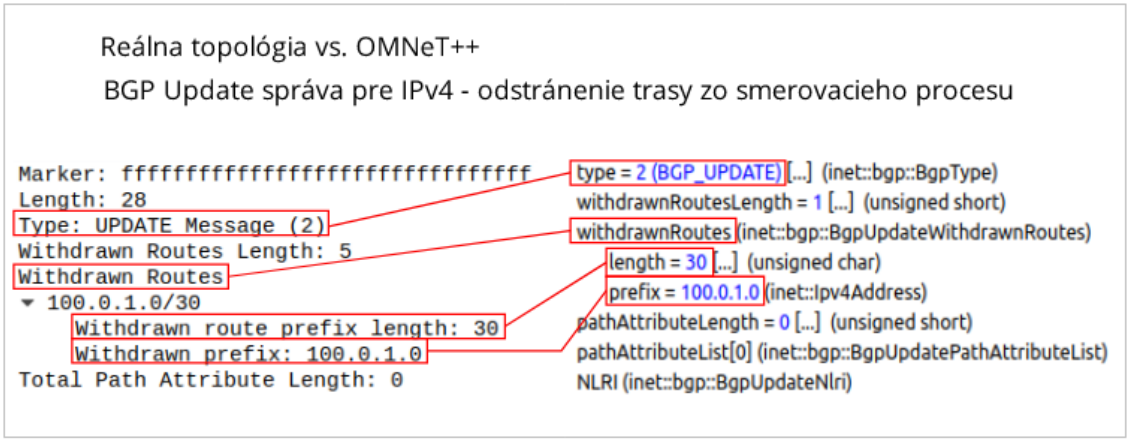
Obr. 4.20: BGP IPv6 smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

4.3 Topológia 1 – výpadok spojenia

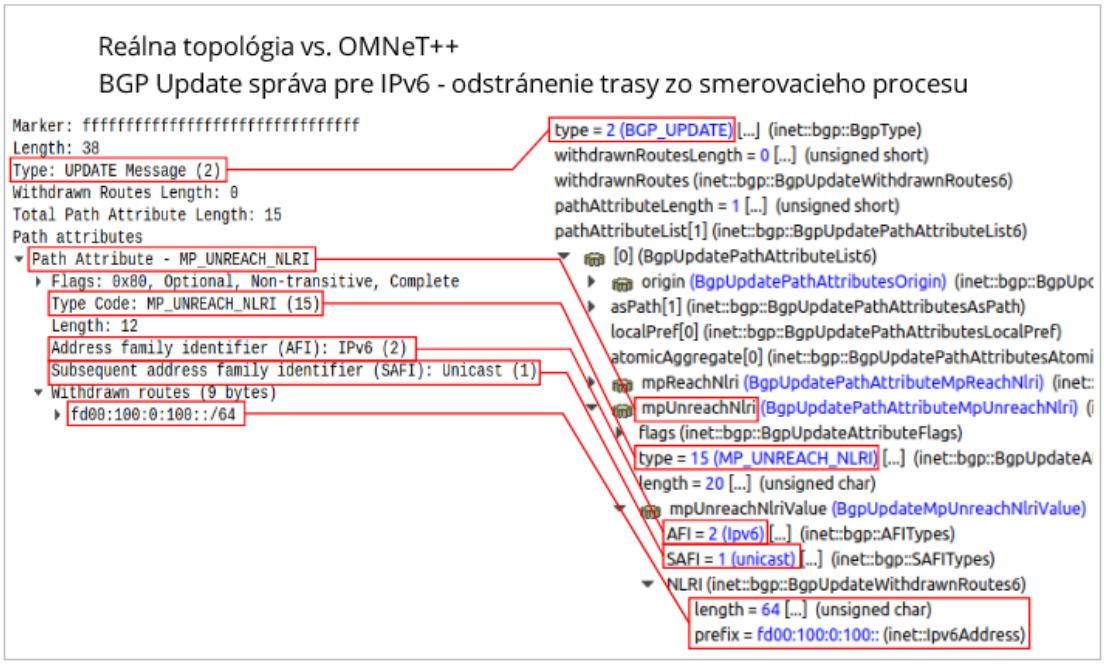
V tomto scenári bude testovaný výpadok spojenia medzi smerovačmi v rámci topológie. Daný scenár vykonáva kroky predchádzajúceho scenára až po dosiahnutie konvergencie siete. Smerovače, medzi ktorými je vytvorené BGP susedstvo, si pravidelne vymieňajú správu typu `BGP Keepalive` pre overenie živosti daného spojenia. Štandardne je interval odosielania týchto správ nastavený na 60 sekúnd. Prípadný výpadok spojenia je overovaný `Hold Down` časovačom, ktorý je doporučené mať nastavené na hodnotu trojnásobku `Keepalive` časovača. V prípade, že po dobu uplynutia `Hold Down` časovača neobdržal smerovač od svojho suseda žiadnu správu, považuje smerovač toto spojenie za prerušené a smerovače sa tejto zmene v topológii musia prispôsobiť. Smerovače, ktoré zistili výpadok spojenia odosielajú ostatným susedným smerovačom správu `BGP Update` s daným obsahom, ktorý udáva odstránenie príslušných sietí zo smerovacich tabuliek a taktiež odstránia trasy zo svojích smerovacích tabuliek. V rámci testovacieho scenára je naplánovaný výpadok linky mezi smerovačmi R1 a R2 v simulačnom čase **t=50**. Keďže v testovanej topológii sú nadväzované susedstvá pre obe adresné rodiny, teda IPv4 a IPv6, je treba zabezpečiť korektné reakcie protokolu aj v prípade IPv4 aj IPv6 sieťového protokolu. Po zistení výpadku smerovač R2 odosiela smerovaču R3 príslušnú správu `BGP Update`. Správy pre obe adresné rodiny

z reálnej topológie a simulačného prostredia OMNeT++ sú zobrazené na obrázkoch 4.21 a 4.22. Tieto zmeny topológie si vzájomne smerovače preposielajú a výsledkom je validný stav topológie. Zmeny je možné vidieť na obrázkoch 4.23 až 4.26, kde sú zobrazené BGP smerovacie tabuľky smerovača R4, ku ktorému sa propagovala zmena v topológii. Následne na túto zmenu reagoval úpravou svojich pracovných databáz vymazaním trasy do siete, ktorú propaguje smerovač R1, pretože táto trasa už nie je dostupná.

Z doložených smerovacích tabuliek je možné vidieť, že aj v tomto prípade sa simulačný model správa zhodne s reálnou topológiou. Jediným rozdielom je obsah `BGP Update` správ kde v prípade Cisco zariadení obsahujú správy iba polia potrebné pre informovanie susedných smerovačov o odstránení trás zo smerovacieho procesu. V prípade implementácie v simulátore v `BGP Update` správe, zostávajú aj ostatné polia. To však nie je chybné správanie, keďže v RFC sa udáva, že ostatné polia tam v takomto prípade byť nemusia ale ani nie sú zakázané.



Obr. 4.21: Správa BGP Update zo smerovača R2 pre IPv4 adresnú rodinu z reálnej topológie a simulačného prostredia OMNeT++.



Obr. 4.22: Správa BGP Update zo smerovača R2 pre IPv6 adresnú rodinu z reálnej topológie a simulačného prostredia OMNeT++.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   100.0.4.0/30     0.0.0.0                  0         32768 i
*>   200.0.2.0/30     10.0.23.1                0             0 200 i
*>_  200.0.3.0/30     10.0.23.1                0             0 200 i
```

Obr. 4.23: BGP IPv4 smerovacia tabuľka R4 smerovača.

```
BGPTestLoopback.R4.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=3
elements[3] (inet::bgp::RoutingTableEntry *)
    [0] BGP - Destination: 100.0.4.0/255.255.255.252 , PathType: IGP , NextHops: <unspec> , AS:
    [1] BGP - Destination: 200.0.2.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.23.1 , AS: 200
    [2] BGP - Destination: 200.0.3.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.23.1 , AS: 200
```

Obr. 4.24: BGP IPv4 smerovacia tabuľka R4 smerovača v prostredí OMNeT++.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   FD00:200:0:200::/64
                      FD00:23:23::             0             0 200 i
*>   FD00:200:0:300::/64
                      FD00:23:23::             0             0 200 i
*>   FD00:300:0:400::/64
                      ::                       0         32768 i
```

Obr. 4.25: BGP IPv6 smerovacia tabuľka R4 smerovača.

```
BGPTestLoopback.R4.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=3
elements[3] (inet::bgp::RoutingTableEntry6 *)
    [0] BGP - Destination: fd00:300:0:400::/64 , PathType: IGP , NextHops: <unspec> , AS:
    [1] BGP - Destination: fd00:200:0:200::/64 , PathType: EGP , NextHops: fd00:23:23:: , AS: 200
    [2] BGP - Destination: fd00:200:0:300::/64 , PathType: EGP , NextHops: fd00:23:23:: , AS: 200
```

Obr. 4.26: BGP IPv6 smerovacia tabuľka R4 smerovača v prostredí OMNeT++.

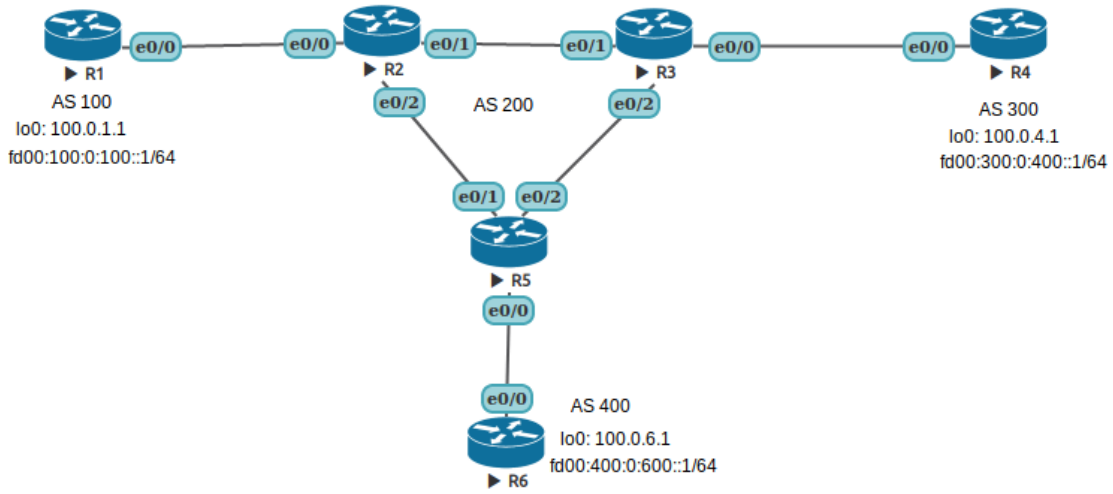
4.4 Topológia 1 – obnovenie spojenia po výpadku

Tento test overuje funkčnosť obnovenie spojenia po výpadku simulačného modelu. V prípade výpadku spojenia smerovače prechádzajú vo svojich konečných automatoch do stavu `Idle`, ktorý reprezentuje počiatočný stav konečného automatu. Smerovače sa opäť pokúsia nadviazať spojenie so susedným smerovačom. V prípade, že sa výpadok odstráni, tj. spojenie medzi smerovačmi je obnovené, smerovače nadväzujú znova svoje susedstvo ako v testovacom scenári 4.2. Výsledok tohoto scenára je zhodný s výsledkom scenára 4.2. Teda topológia sa znova dostáva do konvergentného stavu. Na konci simulácie obsahujú smerovače vo svojich smerovacích tabuľkách rovnaké trasy ako na konci prvého testovacieho scenára. Výpadok linky medzi smerovačmi R1 a R2 je naplánovaný v simulačnom čase **t=50** a obnovenie linky v čase **t=215**.

4.5 Testovacia topológia 2

Testovacia topológia je zložená zo štyroch autonómnych systémov a dokopy obsahuje šesť smerovačov `R1-R6`. Stratégia zostáva rovnaká ako v predchádzajúcom príklade len s tým rozdielom, že AS 200 teraz obsahuje tri smerovače `R2, R3` a `R5`. K smerovaču `R5` je pripojený smerovač `R6`, ktorý sa nachádza v AS 400. V topológii budú oznamované siete `loopback 0` rozhraní smerovačov `R1, R4` a `R6`. Schéma topológie je zobrazená na obrázku 4.27. V schéme sú uvedené IP adresy, ktoré budú smerovače oznamovať svojím susedom. Ako aj predtým, testovacia topológia ma za úlohu overiť funkčnosť *multi address-family* smerovania. Testovacie scenáre sú zhodné ako v predchádzajúcom prípade, preto nie je potrebný ich podrobný popis. V jednotlivých scenároch nebudú zobrazované správy BGP procesu, ale ako dôkaz funkčnosti budú slúžiť výpisy zo smerovacích tabuliek zo simulačného prostredia, ktoré budú obsahovať vždy svoj ekvivalent z reálnej topológie. Pre prehľadnosť sú v testoch zobrazené iba niektoré smerovacie tabuľky.

Test je umiestnený v adresári `inet4/examples/bgpv4/BgpEx3_3routers` a simulácia obsahuje tri možné varianty, z ktorých je potrebné zvoliť si jednu pri spustení simulácie v prostredí OMNeT++ presne ako v predchádzajúcom prípade.



Obr. 4.27: Schéma 2. referenčnej topológie z prostredia Ciscolab.

4.6 Topológia 2 – nadviazanie spojenia

Smerovače `R1`, `R4` a `R6` oznamujú susedným smerovačom svoje siete, ktoré majú pripojené na svoje rozhrania typu `loopback`. Po spustení testovacieho scenáru je overované nadviazanie spojenia medzi smerovačmi a výmena potrebných informácií. Smerovače v AS 200 implicitne neoznamujú žiadnu sieť. Ich úlohou je korektne preposielať informácie ostatným smerovačom. Ako dôkaz sú uvedené smerovacie tabuľky smerovačov `R1`, `R3` a `R6`. Pre smerovač `R1` sú zobrazené BGP tabuľky na obrázkoch 4.28 až 4.31. Pre smerovač `R3` sú príslušné tabuľky zobrazené na obrázkoch 4.32 až 4.35 a pre posledný smerovač `R6` sú zobrazené na obrázkoch 4.36 až 4.39. Z doložených výpisov je možné vidieť, že smerovače v simulácií obsahujú po skonvergovaní topológie rovnaké informácie ako v reálnej topológii.

```
     Network          Next Hop            Metric LocPrf Weight Path
 *>  100.0.1.0/30     0.0.0.0                  0         32768 i
 *>  100.0.4.0/30     10.0.12.2                              0 200 300 i
 *>  100.0.6.0/30     10.0.12.2                              0 200 400 i
```

Obr. 4.28: BGP IPv4 smerovacia tabuľka R1 smerovača.

```
▼ BGPTest3.R1.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=3
  ▼ elements[3] (inet::bgp::RoutingTableEntry *)
      [0] BGP - Destination: 100.0.1.0/255.255.255.252 , PathType: IGP , NextHops: <unspec> , AS:
      [1] BGP - Destination: 100.0.4.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.12.2 , AS: 200 300
      [2] BGP - Destination: 100.0.6.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.12.2 , AS: 200 400
```

Obr. 4.29: BGP IPv4 smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

```
    Network          Next Hop            Metric LocPrf Weight Path
*>  FD00:100:0:100::/64
                      ::                      0         32768 i
*>  FD00:300:0:400::/64
                      FD00:12:12::1                         0 200 300 i
*>  FD00:400:0:600::/64
                      FD00:12:12::1                         0 200 400 i
```

Obr. 4.30: BGP IPv6 smerovacia tabuľka R1 smerovača.

```
BGPTest3.R1.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=3
  elements[3] (inet::bgp::RoutingTableEntry6 *)
    [0] BGP - Destination: fd00:100:0:100::/64 , PathType: IGP , NextHops: <unspec> , AS:
    [1] BGP - Destination: fd00:300:0:400::/64 , PathType: EGP , NextHops: fd00:12:12::1 , AS: 200 300
    [2] BGP - Destination: fd00:400:0:600::/64 , PathType: EGP , NextHops: fd00:12:12::1 , AS: 200 400
```

Obr. 4.31: BGP IPv6 smerovacia tabuľka R1 smerovača v prostredí OMNeT++.

```
    Network          Next Hop            Metric LocPrf Weight Path
*>i 100.0.1.0/30     10.0.23.1                0    100      0 100 i
*>  100.0.4.0/30     10.0.34.2                0             0 300 i
*>i 100.0.6.0/30     10.0.35.2                0    100      0 400 i
```

Obr. 4.32: BGP IPv4 smerovacia tabuľka R3 smerovača.

```
BGPTest3.R3.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=3
  elements[3] (inet::bgp::RoutingTableEntry *)
    [0] BGP - Destination: 100.0.4.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.34.2 , AS: 300
    [1] BGP - Destination: 100.0.1.0/255.255.255.252 , PathType: IGP , NextHops: 10.0.23.1 , AS: 100
    [2] BGP - Destination: 100.0.6.0/255.255.255.252 , PathType: IGP , NextHops: 10.0.35.2 , AS: 400
```

Obr. 4.33: BGP IPv4 smerovacia tabuľka R3 smerovača v prostredí OMNeT++.

```
    Network          Next Hop            Metric LocPrf Weight Path
*>i FD00:100:0:100::/64
                      FD00:23:23::            0    100      0 100 i
*>  FD00:300:0:400::/64
                      FD00:34:34::1           0             0 300 i
*>i FD00:400:0:600::/64
                      FD00:35:35::1           0    100      0 400 i
```

Obr. 4.34: BGP IPv6 smerovacia tabuľka R3 smerovača.

```
BGPTest3.R3.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=3
  elements[3] (inet::bgp::RoutingTableEntry6 *)
    [0] BGP - Destination: fd00:300:0:400::/64 , PathType: EGP , NextHops: fd00:34:34::1 , AS: 300
    [1] BGP - Destination: fd00:100:0:100::/64 , PathType: IGP , NextHops: fd00:23:23:: , AS: 100
    [2] BGP - Destination: fd00:400:0:600::/64 , PathType: IGP , NextHops: fd00:35:35::1 , AS: 400
```

Obr. 4.35: BGP IPv6 smerovacia tabuľka R3 smerovača v prostredí OMNeT++.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   100.0.1.0/30     10.0.56.1                              0 200 100 i
*>   100.0.4.0/30     10.0.56.1                              0 200 300 i
*>_  100.0.6.0/30     0.0.0.0                  0         32768 i
```

Obr. 4.36: BGP IPv4 smerovacia tabuľka R6 smerovača.

```
BGPTest3.R6.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=3
  elements[3] (inet::bgp::RoutingTableEntry *)
    [0] BGP - Destination: 100.0.6.0/255.255.255.252 , PathType: IGP , NextHops: <unspec> , AS:
    [1] BGP - Destination: 100.0.1.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.56.1 , AS: 200 100
    [2] BGP - Destination: 100.0.4.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.56.1 , AS: 200 300
```

Obr. 4.37: BGP IPv4 smerovacia tabuľka R6 smerovača v prostredí OMNeT++.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   FD00:100:0:100::/64
                      FD00:56:56::                           0 200 100 i
*>   FD00:300:0:400::/64
                      FD00:56:56::                           0 200 300 i
*>   FD00:400:0:600::/64
                      ::                       0         32768 i
```

Obr. 4.38: BGP IPv6 smerovacia tabuľka R6 smerovača.

```
BGPTest3.R6.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=3
  elements[3] (inet::bgp::RoutingTableEntry6 *)
    [0] BGP - Destination: fd00:400:0:600::/64 , PathType: IGP , NextHops: <unspec> , AS:
    [1] BGP - Destination: fd00:100:0:100::/64 , PathType: EGP , NextHops: fd00:56:56:: , AS: 200 100
    [2] BGP - Destination: fd00:300:0:400::/64 , PathType: EGP , NextHops: fd00:56:56:: , AS: 200 300
```

Obr. 4.39: BGP IPv6 smerovacia tabuľka R6 smerovača v prostredí OMNeT++.

4.7 Topológia 2 – výpadok spojenia

V testovacom scenári je naplánovaný výpadok spojenia medzi smerovačmi `R1` a `R2` v simulačnom čase **t=50**. Pre overenie správnosti sú doložené výpisy smerovacích BGP tabuliek smerovačov `R1`, `R2` a `R4`, ktoré obsahujú aktualizované informácie s aplikovanými reakciami na výpadok spojenia. Pre smerovač `R1` sú zobrazené BGP tabuľky na obrázkoch 4.40 až

4.43. Pre smerovač R2 sú príslušné tabuľky zobrazené na obrázkoch 4.44 až 4.47 a pre posledný smerovač R4 sú zobrazené na obrázkoch 4.48 až 4.51. Smerovače obsahujú zhodné informácie ako zo simulácie tak z referenčnej reálnej topológie.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   100.0.1.0/30     0.0.0.0                  0         32768 i
```

Obr. 4.40: BGP IPv4 smerovacia tabuľka R1 smerovača po výpadku spojenia.

```
BGPTest3.R1.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=1
  elements[1] (inet::bgp::RoutingTableEntry *)
    [0] BGP - Destination: 100.0.1.0/255.255.255.252 , PathType: IGP , NextHops: <unspec> , AS:
```

Obr. 4.41: BGP IPv4 smerovacia tabuľka R1 v prostredí OMNeT++ po výpadku spojenia.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   FD00:100:0:100::/64
                      ::                       0         32768 i
```

Obr. 4.42: BGP IPv6 smerovacia tabuľka R1 smerovača po výpadku spojenia.

```
BGPTest3.R1.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=1
  elements[1] (inet::bgp::RoutingTableEntry6 *)
    [0] BGP - Destination: fd00:100:0:100::/64 , PathType: IGP , NextHops: <unspec> , AS:
```

Obr. 4.43: BGP IPv6 smerovacia tabuľka R1 v prostredí OMNeT++ po výpadku spojenia.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>i  100.0.4.0/30     10.0.23.2                0    100      0 300 i
*>i  100.0.6.0/30     10.0.25.2                0    100      0 400 i
```

Obr. 4.44: BGP IPv4 smerovacia tabuľka R2 smerovača po výpadku spojenia.

```
BGPTest3.R2.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=2
  elements[2] (inet::bgp::RoutingTableEntry *)
    [0] BGP - Destination: 100.0.4.0/255.255.255.252 , PathType: IGP , NextHops: 10.0.23.2 , AS: 300
    [1] BGP - Destination: 100.0.6.0/255.255.255.252 , PathType: IGP , NextHops: 10.0.25.2 , AS: 400
```

Obr. 4.45: BGP IPv4 smerovacia tabuľka R2 v prostredí OMNeT++ po výpadku spojenia.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>i  FD00:300:0:400::/64
                       FD00:23:23::1            0    100      0 300 i
*>i  FD00:400:0:600::/64
                       FD00:25:25::1            0    100      0 400 i
```

Obr. 4.46: BGP IPv6 smerovacia tabuľka R2 smerovača po výpadku spojenia.

```
BGPTest3.R2.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=2
  elements[2] (inet::bgp::RoutingTableEntry6 *)
    [0] BGP - Destination: fd00:300:0:400::/64 , PathType: IGP , NextHops: fd00:23:23::1 , AS: 300
    [1] BGP - Destination: fd00:400:0:600::/64 , PathType: IGP , NextHops: fd00:25:25::1 , AS: 400
```

Obr. 4.47: BGP IPv6 smerovacia tabuľka R2 v prostredí OMNeT++ po výpadku spojenia.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   100.0.4.0/30     0.0.0.0                  0         32768 i
*>   100.0.6.0/30     10.0.34.1                                  0 200 400 i
```

Obr. 4.48: BGP IPv4 smerovacia tabuľka R4 smerovača po výpadku spojenia.

```
BGPTest3.R4.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=2
  elements[2] (inet::bgp::RoutingTableEntry *)
    [0] BGP - Destination: 100.0.4.0/255.255.255.252 , PathType: IGP , NextHops: <unspec> , AS:
    [1] BGP - Destination: 100.0.6.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.34.1 , AS: 200 400
```

Obr. 4.49: BGP IPv4 smerovacia tabuľka R4 smerovača v prostredí OMNeT++ po výpadku spojenia.

```
     Network          Next Hop            Metric LocPrf Weight Path
*>   FD00:300:0:400::/64
                        ::                      0         32768 i
*>   FD00:400:0:600::/64
                       FD00:34:34::                                0 200 400 i
```

Obr. 4.50: BGP IPv6 smerovacia tabuľka R4 smerovača po výpadku spojenia.

```
BGPTest3.R4.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=2
  elements[2] (inet::bgp::RoutingTableEntry6 *)
    [0] BGP - Destination: fd00:300:0:400::/64 , PathType: IGP , NextHops: <unspec> , AS:
    [1] BGP - Destination: fd00:400:0:600::/64 , PathType: EGP , NextHops: fd00:34:34:: , AS: 200 400
```

Obr. 4.51: BGP IPv6 smerovacia tabuľka R4 smerovača v prostredí OMNeT++ po výpadku spojenia.

4.8 Topológia 2 – obnovenie spojenia po výpadku

Posledný testovací scenár overuje schopnosť topológie vysporiadať sa s výpadkom a uviesť sieť do konvergentného stavu. Keďže sú smerovacie informácie v rámci simulácie v BGP tabuľkách smerovačov pridávané v poradí v akom boli prijaté, je možné ukázať, že smerovač `R4` reagoval na smeny v topológii pridaním trasy, ktorú propaguje smerovač `R1`. Smerovacie tabuľky sú zobrazené na obrázkoch 4.52 až 4.55 a z nich možno vidieť, že nadväzujú na tabuľky predchádzajúceho scenára a súčasne sú v nich validné informácie v porovnaní s reálnou topológiou. Výpadok linky medzi smerovačmi R1 a R2 je naplánovaný v simulačnom čase **t=50** a obnovenie linky v čase **t=225**.

```
     Network          Next Hop            Metric LocPrf Weight Path
 *>  100.0.1.0/30     10.0.34.1                              0 200 100 i
 *>  100.0.4.0/30     0.0.0.0                  0         32768 i
 *>_ 100.0.6.0/30     10.0.34.1                              0 200 400 i
```

Obr. 4.52: BGP IPv4 smerovacia tabuľka R4 smerovača po obnovení spojenia.

```
▼ BGPTest3.R4.bgp._BGPRoutingTable (vector<RoutingTableEntry *>) size=3
  ▼ elements[3] (inet::bgp::RoutingTableEntry *)
      [0] BGP - Destination: 100.0.4.0/255.255.255.252 , PathType: IGP , NextHops: <unspec> , AS:
      [1] BGP - Destination: 100.0.6.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.34.1 , AS: 200 400
      [2] BGP - Destination: 100.0.1.0/255.255.255.252 , PathType: EGP , NextHops: 10.0.34.1 , AS: 200 100
```

Obr. 4.53: BGP IPv4 smerovacia tabuľka R4 smerovača v prostredí OMNeT++ po obnovení spojenia.

```
     Network          Next Hop            Metric LocPrf Weight Path
 *>  FD00:100:0:100::/64
                       FD00:34:34::                            0 200 100 i
 *>  FD00:300:0:400::/64
                       ::                       0         32768 i
 *>  FD00:400:0:600::/64
   _                   FD00:34:34::                            0 200 400 i
```

Obr. 4.54: BGP IPv6 smerovacia tabuľka R4 smerovača po obnovení spojenia.

```
▼ BGPTest3.R4.bgp._BGPRoutingTable6 (vector<RoutingTableEntry6 *>) size=3
  ▼ elements[3] (inet::bgp::RoutingTableEntry6 *)
      [0] BGP - Destination: fd00:300:0:400::/64 , PathType: IGP , NextHops: <unspec> , AS:
      [1] BGP - Destination: fd00:400:0:600::/64 , PathType: EGP , NextHops: fd00:34:34:: , AS: 200 400
      [2] BGP - Destination: fd00:100:0:100::/64 , PathType: EGP , NextHops: fd00:34:34:: , AS: 200 100
```

Obr. 4.55: BGP IPv6 smerovacia tabuľka R4 smerovača v prostredí OMNeT++ po obnovení spojenia.

Kapitola 5

Záver

V rámci diplomovej práce vznikol simulačný model BGPv4 protokolu vo frameworku INET4.0 v prostredí OMNeT++ 5.4.1. Model je schopný súčasne prenášať IPv4 a IPv6 sieťové protokoly a dokáže reagovať na výpadky kolektivity v sieti a po obnovení spojenia zvláda dostať sieť do skonvergovaného stavu. Diplomová práca bola prezentovaná formou plagátu na študentskej konferencii Excel@FIT a dané rozšírenie BGPv4 protokolu bude zahrnuté do žiadosti o pridanie do oficiálneho frameworku INET.

Z toho dôvodu bol naštudovaný úvod do smerovacích protokolov, ktorý je spísaný v kapitole 2 so zameraním na protokol `BGP`, ktorému sa venuje podkapitola 2.2, v ktorej je priblížený základný princíp smerovacieho protokolu s popisom využívaných správ, ktoré sa v danom protokole prenášajú. Následne je znázornený konečný automat nadviazania spojenia a sú charakterizované dátové štruktúry využívané daným protokolom. Ďalej je uvedené rozšírenie o *multi address-family* smerovanie. Poslednú teoretickú podkapitolu tvorí konfigurácia BGP smerovacieho protokolu na Cisco zariadeniach 2.4.

V nadväzujúcej kapitole 3 je priblížené simulačné prostredie `OMNeT++` s následným popisom aktuálneho stavu BGP protokolu v rámci frameworku `INET` a to vrátane zistených problémov a návrhom vylepšení. Na záver tejto kapitoly je uvedená podkapitola 3.4, ktorá sa venuje popisu implementácie vzniknutého simulačného modelu od nového konfiguračného súboru cez odstránenie chýb implementácie až po samotné rozšírenie o *multi address-family* smerovanie. V rámci implementácie boli vyriešené všetky uvedené problémy okrem problému P4. Správy typu BGP Notification by sa v rámci simulačného prostredia nedali korektne modelovať a v rámci simulačného modelu pre funkcionalitu protokolu nie sú potrebné.

Ďalšou dôležitou kapitolou je kapitola 4, ktorá overuje funkčnosť vzniknutého simulačného modelu. V testovacej fáze vznikli štyri topológie, na ktorých bola skúmaná funkčnosť implementácie. V kapitole venovanej testovaniu boli popísané dve. K simulačným príkladom boli vytvorené reálne siete, ktoré slúžili ako referenčné na zariadeniach firmy `Cisco`. Uvedené topológie obsahujú tri scenáre prípadného použitia simulačného modelu.

Literatúra

[1] INET Framework. [Online; navštívené 12.10.2018].
URL https://inet.omnetpp.org/

[2] Academy, C. N.: *Routing Protocols Companion Guide*. Cisco Press, 2014, ISBN 978-1-58713-323-7.

[3] Authority, I. A. N.: Address Family Numbers. [Online; navštívené 25.1.2019].
URL https://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml

[4] Bates, T. J.; Chandra, R.; Rekhter, Y.; aj.: Multiprotocol Extensions for BGP-4. *RFC 4760*, Jan 2007.

[5] Chandra, R.; Scudder, J. G.: Capabilities Advertisement with BGP-4. *RFC 3392*, Nov 2002.

[6] Cisco Systems, I.: *IP Routing: BGP Configuration Guide, Cisco IOS Release 15M&T*. [Online; navštívené 10.10.2018].
URL https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/15-mt/irg-15-mt-book/irg-overview.html

[7] Doyle, J.; Carroll, J. D.: *Routing TCP/IP, Volume II (CCIE Professional Development)*. Cisco Press, 2001, ISBN 1-57870-089-2.

[8] Faculty of Information Technology, B. U. o. T.: Project ANSA. [Online; navštívené 12.10.2018].
URL https://nes.fit.vutbr.cz/ansa/

[9] Osama, W.: BGP Routing Information Base (RIB). [Online; navštívené 20.12.2018].
URL http://www.networkers-online.com/blog/2010/03/bgp-routing-information-base-rib/

[10] Rekhter, Y.; Hares, S.; Li, T.: A Border Gateway Protocol 4 (BGP-4). *RFC 4271*, Jan 2006.

[11] Tutorialspoint: IPv4 - Address Classes. [Online; navštívené 25.1.2019].
URL https://www.tutorialspoint.com/ipv4/ipv4_address_classes.htm

[12] Varga, A.: OMNeT++ Discrete Event Simulator. [Online; navštívené 18.10.2018].
URL https://omnetpp.org/

[13] Varga, A.: OMNeT++ Simulation Manual. [Online; navštívené 20.10.2018].
URL https://omnetpp.org/doc/omnetpp/manual/

[14] Veselý, V.; Palúch, P.: Border Gateway Protocol (ROUTE Module 7). [Online; navštívené 10.10.2018].
URL https://netacad.fit.vutbr.cz/ccnp/route/ROUTE_M7_ENG_v7.pdf

[15] Vinit Jain, B. E.: *Troubleshooting BGP: A Practical Guide to Understanding and Troubleshooting BGP*. Cisco Press, 2016, ISBN 1-58714-464-6.

[16] Vinit Jain, B. E.: BGP Fundamentals. *Cisco Press*, Jan 2018.

Príloha A

Obsah priloženého média

Súčasťou práce je DVD, ktoré obsahuje nasledujúcu adresárovú štruktúru:

```
.
├── BgpEx1Loopback/ ..... Priečinok obsahujúci súbory, ktoré boli
│                         vytvorené a použité pri overovaní funkci-
│                         onality simulačného modelu.
├── BgpEx3_3routers/ .... Priečinok obsahujúci súbory, ktoré boli
│                         vytvorené a použité pri overovaní funkci-
│                         onality simulačného modelu.
├── inet4/ ............. Priečinok obsahujúci zdrojové súbory fra-
│                         meworku INET4, ktorého súčasťou je táto
│                         práca.
├── latex/ ............. Priečinok obsahujúci zdrojové súbory
│                         (LaTeX) práce.
├── readme.txt ......... Odkaz na verejný Git repozitár, ktorý ob-
│                         sahuje zdrojové súbory tejto práce.
└── thesis.pdf ......... Text práce vo formáte .pdf.
```