VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

IMPLEMENTACE OSPFV3 V INET4

IMPLEMENTATION OF OSPFV3 FOR INET4

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE — **Bc. LUKÁŠ GALBIČKA**
AUTHOR

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

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2018/2019

Zadání diplomové práce

Student: **Galbička Lukáš, Bc.**
Program: Informační technologie Obor: Počítačové sítě a komunikace
Název: **Implementace OSPFv3 v INET4**
Implementation of OSPFv3 for INET4
Kategorie: Počítačové sítě
Zadání:

1. Analyzujte link-state směrovací protokoly OSPFv2 a OSPFv3 a prostudujte jejich chování na Cisco zařízeních.
2. Zjistěte stav implementace link-state protokolu OSPFv3 v OMNeT++.
3. Implementujte podporu OSPFv3 protokolu do frameworku ANSAINET v prostředí OMNeT++ se zaměřením na podporu multi address-family směrování.
4. Ověřte chování implementovaných simulačních modelu vůči reálné topologii a analyzujte výsledky.
5. Podle doporučení vedoucího integrujte vaše řešení do frameworku INET4.

Literatura:

- M. Ruprich, *Modeling of OSPFv3 Link-State Routing Protocol*. Brno, 2017. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Vladimír Veselý, Ph.D.
- J. Moy, *RFC 2328 - OSPF Version 2*, IETF, 1998.
- R. Coltun, *RFC 5340 - OSPF for IPv6*, IETF, 2008.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

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

Táto práca sa zaoberá tvorbou simulácie smerovacieho protokolu OSPF v simulačnom prostredí OMNeT++. OMNeT++ je diskrétny modulárny simulátor využívaný prednostne pre simuláciu počítačových sietí. Práca obsahuje teoretické základy pre pochopenie fungovania OSPFv2 a zmeny v OSPFv3 pre IPv6, na základe ktorých je implementovaný samotný model. Ďalej sa v práci nachádza postup konfigurácie OSPFv3 protokolu na topológiu zloženú zo zariadení s referenčnou implementáciou od firmy Cisco. Po nej pokračuje rozbor prebraných zdrojových súborov, stav súčasnej implementácie a popis jej ďalšieho rozšírenia. Práca je zakončená testovaním funkcionality a zhodnotením dosiahnutých výsledkov.

Abstract

This thesis deals with simulation of routing protocol OSPF in simulation software called OMNeT++. OMNeT++ is a discrete modular simulator mostly used for simulation of computer networks. This thesis includes theory needed for an understanding of the functionality of OSPFv2 and changes in OSPFv3 for IPv6, which are implemented in the model itself. Moreover, thesis contains the configuration of OSPFv3 protocol on topology created from Cisco devices following by analysis of previous source files, state of implementation and its further extension. Thesis is finished with functionality testing and evaluation of results.

Kľúčové slová

OSPF, OSPFv3, modelování sítí, OMNET++, ANSAINET, INET, směrování

Keywords

OSPF, OSPFv3, network modeling, OMNET++,ANSAINET, INET, routing

Citácia

GALBIČKA, Lukáš. *Implementace OSPFv3 v INET4*. Brno, 2019. Semestrální projekt. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vladimír Veselý, Ph.D.

Implementace OSPFv3 v INET4

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Vladimíra Veselého Ph.D. Dodatočné informácie mi poskytol Ing. Michal Ruprich. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Lukáš Galbička
20. mája 2019

Poďakovanie

Týmto by som chcel poďakovať môjmu vedúcemu práce Ing. Vladimírovi Veselému, Ph.D za jeho venovaný čas a nápomocné rady počas celej doby vedenia práce. Taktiež by som rád poďakoval Ing. Michalovi Ruprichovi, že bol aj po ukončení školy ochotný nájsť si čas na konzultácie ohľadom jeho odovzdanej implementácie. Rodine a priateľke za ich psychickú podporu a spolužiakom a kamarátom, ktorí mi boli v prípade potreby nápomocní.

Ako poďakovanie by som sa tiež rád podelil o recept k môjmu oblúbenému jedlu, Chilli sin carne. Sójový granulát zalejeme vriacim zeleninovým vývarom a necháme ho odstáť 10 minút. Zatiaľ na olivovom oleji na panvici opražíme nadrobno nakrájanú cibuľu, pretlačený cesnak a restujeme do sklovita. Do oleja pridáme na krúžky nakrájanú mrkvu. Keď trochu zmäkne, pridáme k nej na prúžky pripravenú papriku spolu s mletou a údenou paprikou, rímskou rascou, oreganom a čili. Varíme na miernom ohni a dávame pozor, aby sa suroviny nepripálili.

Odstáty sójový granulát zbavíme vody scedením a pridáme ho na panvicu. Trochu osolíme a pár minút miešame, aby mrkva s paprikou rozvoňali, zmäkli a ich chuť sa rozptýlila do granulátu. Pridáme passatu, prepláchnutú fazuľu, kukuricu a nasekáme na drobné čiastočky arašidy, ktoré v jedle očaria chrumkavosťou. Necháme prevariť a servírujeme.

Obsah

Kapitola 1

Úvod

V modernej dobe sa počítačové siete rozrástli do obrovských rozmerov. Je nemožné spravovať takéto siete len za pomoci statického smerovania s nulovou odolnosťou proti zlyhaniu a preto postupne vznikali komplexnejšie a efektívnejšie protokoly pre dynamické smerovanie. Jedným z najrozšírenejších je OSPF. Avšak zložitosť sietí neustále narastá a vytvárať takéto siete bez tvorby simulačných modelov môže byť veľmi náročné. V prípade chyby či nesprávneho postupu môže takéto hazardovanie viesť k zbytočnému nárastu ceny a celkového času realizácie siete.

Simulovanie nám teda mimo iné pomáha šetriť čas a zdroje, analyzovať komplexné problémy, rozvíjať nápady a hľadať problémy navrhnutého dizajnu. Jedným z takýchto simulačných prostredí, ktorý poskytuje potrebné modelovacie nástroje je OMNeT++. S týmto prostredím je silno previazaný framework INET, ktorý poskytuje modely so zameraním na TCP/IP model. Na rozšírenie ďalšej funkcionality INETu sa zameriava projekt ANSA na FIT VUT v Brně. INET aj ANSA sú bližšie popísané
v kapitole 5.

Táto práca sa zaoberá rozšírením nástrojov pre simulačný model protokolu OSPFv3. Tento simulačný model je súčasťou projektu ANSA. V kapitole 2 je poskytnutý úvod do dynamického smerovania a link-state protokolov, aby mohol byť následne detailne rozobraný protokol OSPF. Kapitola 3 sa zameriava na podporu IPv6, prechod na OSPFv3 a rozdiely oproti staršej verzii. Kapitola 4 obsahuje komentovaný postup konfigurácie OSPFv3 na Cisco zariadeniach. Následne kapitola 5 popisuje stav vtedy aktuálnej implementácie simulačných nástrojov, čo zahrňuje rozbor súborov, hierarchiu implementovaných tried, rozbor konfiguračných súborov a súpis problémov a chýb prebraného modelu. Taktiež je v kapitole krátke zhrnutie vykonaných úprav OSPFv3 modelu. V kapitole 6 je rozobrané testovanie, ktoré s pokrytím viacerých možných scenárov overí validitu implementovaného protokolu. V závere je uvedené celkové zhodnotenie vykonanej práce a plány pre ďalší postup. Na konci práce sa nachádza sekcia prílohy, ktorá obsahuje príklad konfiguračného súboru, obsah priloženého DVD a vizualizáciu konečných automatov OSPFv3 protokolu.

Kapitola 2

Smerovacie Protokoly

Táto kapitola poskytuje stručný náhľad do smerovacích protokolov a vkladá problematiku protokolu OSPF do širšieho merítka.

Poznáme dvojaký spôsob akým môžu pribúdať záznamy do smerovacej tabuľky smerovača. Statické smerovanie, kedy je každá cesta ručne nakonfigurovaná na každom smerovači a dynamické smerovanie, kedy túto činnosť preberá do rúk nejaký dynamický protokol. Hlavnými úlohami smerovacích protokolov sú:

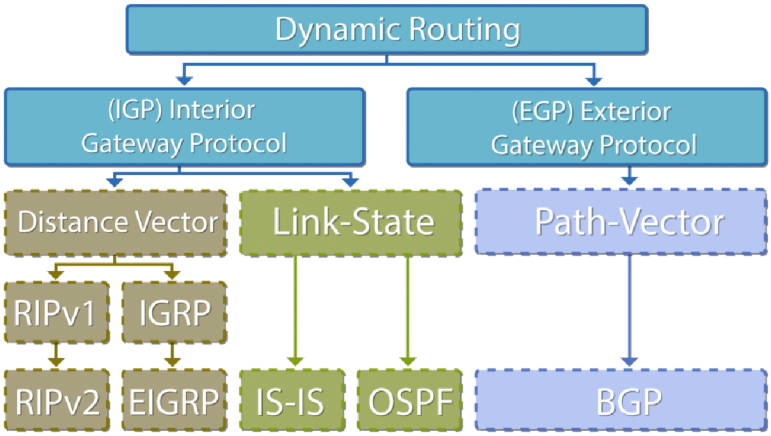
- objavovanie vzdialených sietí;
- udržiavanie aktuálnych informácií v smerovacej tabuľke;
- hľadanie najlepšej cesty k cieľovej sieti;
- zabránenie slučkám;
- reagovať na zmeny v sieti.

Existujú dynamické protokoly pre smerovanie v rámci autonómnych systémov (`AS`) a protokoly pre smerovanie medzi jednotlivými AS. Podľa toho sa protokoly delia na `Interior Gateway Protocols` (`IGP`) a `Exterior Gateway Protocols` (`EGP`). Obrázok 2.1 znázorňuje delenie smerovacích protokolov, ktoré bude popísané nižšie.

2.0.1 Interior Gateway Protocol

IGP je typ smerovacích protokolov používaných pre distribúciu smerovacích informácií v rámci jedného AS. Na základe metódy, ktorou si smerovače vymieňajú smerovacie informácie sa IGP delí na `Distance Vector` a `Link-state` protokoly.

`Distance-vector` protokoly sú jednoduché smerovacie protokoly, ktoré ako primárnu metriku pre určenie najlepšej cesty k cieľovému zariadeniu využívajú distance (vzdialenosť). Tá je meraná predovšetkým počtom hopov, ktoré musí paket k cieľovému zariadeniu vykonať [4]. Niektoré protokoly do výpočtu metriky zahrnú aj latenciu a iné faktory, ktoré ovplyvňujú kvalitu cesty. Smerovač s takýmto protokolom pravidelne zasiela susedným smerovačom kópiu svojej smerovacej tabuľky. Keďže smerovač obdrží smerovacie informácie vždy len od suseda, nemôže si vybudovať topológiu celej siete. To je hlavný dôvod, prečo sú distance vector protokoly používané len v malých a jednoduchých sieťach. K výpočtu cieľovej destinácie využívajú hlavne Bellman–Fordov algoritmus. Príkladmi takýchto protokolov sú RIP verzia 1 a 2, IGRP a EIGRP.



Obr. 2.1: Delenie smerovacích protokolov[1]

`Link-state` smerovanie je komplexná smerovacia technika, v ktorej každý smerovač zdieľa informácie s ostatnými smerovačmi o dostupnosti iných sietí a ich metrike k určeniu najlepšej cesty [3]. Link-state smerovanie funguje tak, že každý smerovač v sieti obdrží mapu prepojenia siete vo forme grafu, ukazujúc ktoré uzly sú medzi sebou prepojené. Každý smerovač potom nezávisle od seba vypočítava najlepší next hop pre každú možnú cieľovú destináciu v sieti len za použitia jeho lokálnej kópie topológie. Kolekcia najlepších next hopov sformuje smerovaciu tabuľku smerovača. Pre výpočet najlepšej cesty smerovače využívajú Dijkstrov algoritmus. Príkladmi link-state protokolov sú IS-IS a OSPF, ktorý je predmetom tejto práce.

2.0.2 Exterior gateway protocol

EGP je smerovací protokol používaný pre výmenu smerovacích informácii medzi smerovačmi z rôznych AS. Aj keď EGP je dynamický protokol, používa veľmi jednoduchý dizajn. Nevyžíva metriky a preto nemôže robiť rozumné smerovacie rozhodnutia. Jediný známy a využívaný EGP protokol je `Border Gateway Protocol` (`BGP`).

2.1 OSPF - Open Shortest Path First

Táto sekcia popisuje protokol OSPF, ktorý existuje vo dvoch verziách. OSPF pre IPv4 a neskoršia verzia OSPFv3, ktorá pridala podporu pre IPv6 smerovanie. OSPFv2 a OSPF ako také je rozoberané v RFC 2328[13]. RFC 5340[8] potom popisuje rozdiely oproti OSPFv2 . Keďže OSPF je rozpísané do dvoch RFC dokumentov, bude aj jeho rozbor robený týmto štýlom. Teda najskôr bude rozoberaný OSPFv2 a neskôr OSPFv3. V rámci nadväzujúcej práce nebolo potrebné meniť implementovaný prechod medzi stavmi v rámci nadväzovania susedstva. Preto táto práca neobsahuje stavový automat suseda (o susedoch: 2.8) ani stavový automat rozhrania (o rozhraniach: 2.7).

OSPF patrí medzi `TCP/IP` dynamické smerovacie protokoly. Slúži k distribúcii smerovacích informácii vo vnútri jedného autonómneho systému (IGP). Je to najpoužívanejší

[1]Obrázok prebraný z: *https : //www.talari.com/glossary_faq/what − are − router − protocols*

link-state smerovací protokol. OSPFv2 má číslo protokolu 89 a pre IPv4 multicast komunikáciu používa adresy:

- `224.0.0.5`, adresa všetkých OSPF smerovačov na danom segmente.
- `224.0.0.6`, adresa všetkých `Designated Router` (`DR`) a `Backup Designated Router` (`BDR`) smerovačov na danom segmente. (O DR/BDR smerovačoch je bližšie info v kapitole 2.5.)

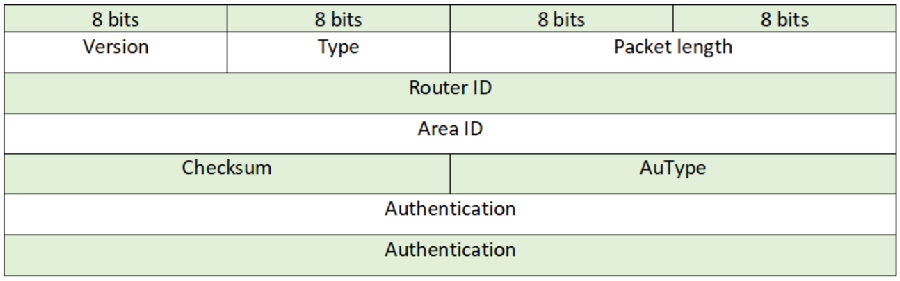
Predvolenú hodnotu `administrative distance` (t.j. hodnota pre výber preferovanej trasy pri smerovaní paketov) má OSPF nastavenú na 110, avšak možno ju predefinovať. Pre určenie cesty do cieľovej siete využíva znalosti stavu liniek v celej sieti. Presnejšie si vypočítava cenu každej linky z maximálnej šírky pásma, ktorou táto linka oplýva. Je založený na `Shortest path first` (`SPF`) technológii. SPF technológia je bližšie rozoberaná v sekcii 2.11. Oproti iným protokolom, OSPF svojou réžiou príliš nezaťažuje sieť a pri zmene sieti má pomerne malý čas konvergencie. Taktiež podporuje `Variable Length Subnet Mask (VLSM)` pre subnetovanie sietí a sumarizáciu ciest. Každý smerovač, na ktorom beží OSPF proces prechádza tromi základnými stavmi:

1. objavovanie susedov, takzvané `Neighbor discovery`;
2. výmena topológie;
3. výpočet ciest.

Po týchto troch stavoch, každý smerovač by mal mať v smerovacej tabuľke zostavené najlepšie trasy do odľahlých sietí.

2.2 OSPF pakety

Všetky OSPF pakety zdieľajú rovnaký formát hlavičky zasielaného paketu. Štruktúra paketu je znázornená na obrázku 2.2 [2].



Obr. 2.2: Hlavička OSPF paketu

Version - verzia, určuje či sa jedná o OSPFv2 alebo OSPFv3.

Type - typ, určuje jeden z piatich typov paketov. Popísané nižšie.

Packet Length - dĺžka paketu v bytoch, vrátane hlavičky.

Router ID - unikátny identifikátor smerovača, ktorý vytvoril túto správu.

[2] Obrázky formátu paketov, vrátane tohoto, sú prebrané z odpovedajúcich RFC dokumentov

Area ID - 32-bitoví identifikátor oblasti, do ktorej smerovač patrí.

Checksum - kontrolný súčet. Pole s autentifikáciou nie je do výpočtu tejto hodnoty zahrnuté.

AuType - typ autentifikačnej procedúry, ktorá bola pre tento paket použitá.

Authentication - 64-bitové autentifikačné pole.

Existuje päť rozdielnych typov OSPF paketov. Sú to:

- Hello
- Database Description
- Link State Request
- Link State Update
- Link State Acknowledgment

Bližšie rozoberanie paketov je v sekcii 2.6.1.

2.3 OSPF oblasti

Zariadenia, ktoré sú súčaťou OSPF domény a susedia spolu môžu vytvárať grupy. Takáto grupa sa nazýva oblasť alebo tiež `area`. V každej takejto oblasti beží samostatná kópia link-state smerovacieho algoritmu. To znamená, že každá takáto oblasť má vlastnú link-state databázu a odpovedajúci graf. [13] Na každom zariadení, ktoré je súčasťou rovnakej oblasti je udržiavaná rovnaká link-state databáza. Z nej možno vyčítať celú topológiu v rámci tejto oblasti. Avšak táto topológia nie je viditeľná z iných sietí, ktoré sú súčasťou inej oblasti alebo sú súčasťou iného AS. Toto delenie topológie do oblastí sa využíva pre pre jednoduchšiu správu siete a taktiež pre zníženie réžie a záťaže liniek medzi zariadeniami. Základné delenie oblastí je na:

- `Backbone` oblasť - označovaná ako `Area 0` alebo `Transit Area`
- Bežná (`Regular`) oblasť - označovaná ako `Non-backbone` oblasť. Sú to všetky ostatné oblasti mimo oblasti 0.

Každá bežná oblasť musí byť pripojená k backbone oblasti. Je to kvôli prevencii pred smerovacími slučkami. Rôzne oblasti od seba oddeľujú `Area-Border` smerovače (`ABR`) (sekcia 2.5).

Každé oblasť môže byť ďalej nakonfigurovaná ako jedna zo štyroch `stub` podtypov [7]:

- `Stub Area`
- `Totally Stubby Area`
- `Not-So-Stubby Area (NSSA)`
- `Totally Not-So-Stubby Area (Totally NSSA)`

Tieto typy oblastí sa líšia filtrovaním a úpravou `Link State Advertisements` (`LSA`)[3] správ prichádzajúcich z backbone oblasti do bežnej oblasti skrz ABR. Z toho vyplýva, že tieto špeciálne typy oblastí môžu byť nakonfigurované výhradne v bežných oblastiach a nie na backbone.

- **Stub Area** - LSA typu 4 a 5 nie sú rozposielané v rámci tejto oblasti. V stub oblasti teda nie sú žiadne `Autonomous System Border Router`(`ASBR`) a teda nevie nič o externých sieťach. Táto oblasť nemôže byť použitá ako medzičlánok pre virtuálne linky. Každá LSA typu 5 správa, ktorá je do stub oblasti zaslaná, je skonvertovaná na východziu cestu (`default route`) skrz LSA typu 3.
- **Totally Stubby Area** - LSA typu 3, 4 a 5 nie sú rozposielané v rámci tejto oblasti. Každá LSA správa typu 3 a 5, ktorá je do takejto oblasti zaslaná, je skonvertovaná na východziu cestu skrz LSA typu 3.
- **NSSA, Totally NSSA** - v niektorých prípadoch je žiadané, aby daná oblasť mala stub či totally stubby vlastnosti, ale s ASBR vo vnútri. Napríklad pre redistribúciu statických ciest či iných ciest z iných sieťových protokolov. Práve pre tento účel slúžia tieto typy oblastí. Majú rovnaké vlastnosti ako stub a totally stubby oblasti, avšak prijímajú aj ASBR a externé siete. Hlavný rozdiel teda je, že NSSA filtruje LSA typu 4 a 5, avšak ABR ich automaticky nekonvertuje na východziu cestu skrz LSA typu 3. V (totally) `NSSA` oblasti sú externé siete prenášané pomocou LSA správ typu 7, ktoré ABR konvertuje na LSA typu 5 a odosiela do ďalšej oblasti.

2.4 Typy sietí

OSPF definuje päť typov sietí[9]:

- `Point-to-point` (`P2P`) spojuje jeden pár smerovačov. Validní susedia na P2P sieťach vždy nadviažu spojenie. V týchto sieťach bude cieľová adresa OSPF paketov vždy adresa `224.0.0.5`.
- `Broadcast` sieť, ako je Ethernet, môže byť pre lepšie rozlíšenie od `NBMA` definovaná ako `broadcast multi-access` sieť. Multi-access sú skrz to, že dokáže spojovať viac ako dve zariadenia a broadcast preto, lebo všetky pripojené zariadenia môžu prijať jedenkrát od zdroja vyslaný paket. OSPF smerovače v broadcast sieťach si volia DR a BDR podľa pravidiel opísaných nižšie. `Hello` pakety a všetky OSPF pakety od DR a BDR sú zasielané všetkým OSPF smerovačom v danom segmente na `multicast 224.0.0.5`. Všetky ostatné smerovače zasielajú `LS update` a `LS acknowledgement` správy všetkým DR a BDR smerovačom v danom segmente na multicast `224.0.0.6`.
- `NBMA (Nonbroadcast Multiaccess)` siete, ako je napríklad `Frame Relay`, sú schopné prepojiť viac ako dva smerovače, nemajú však schopnosti broadcast sietí. Paket zaslaný jedným pripojeným smerovačom nemusia prijať všetky smerovače v danej sieti. Tým pádom môže byť potrebná dodatočná konfigurácia smerovačov pre komunikáciu s ich susedmi. V NBMA sieťach si OSPF smerovače volia DR a BDR, všetky pakety sú však typu unicast.

[3] Bližší popis LSA správ sa nachádza v sekcii 2.9

- `P2MP (Point-To-Multipoint)` siete sú špeciálny typ NBMA sietí, v ktorých sú siete brané ako kolekcia `point-to-point` liniek. Smerovače si v týchto sieťach nevolia DR/BDR a OSPF pakety sú unicastové pre každého známeho suseda.
- Virtuálne linky, sú špeciálnou konfiguráciou ktorá je smerovačom interpretovaná ako P2P sieť, v ktorej sú OSPF pakety zasielané ako unicast.

2.5 Typy smerovačov v OSPF sieťach

Šírenie LSA správ v multi-access sieťach so sebou prináša pár problémov. Formovanie susedstiev medzi prepojenými smerovačmi by malo za následok vytváranie veľkého množstva zbytočných LSA správ čo by viedlo k chaotickému šíreniu nadbytočných správ, teda zbytočné zahltenie linky réžiou. Smerovač by šíril LSA všetkým susedným zariadeniam, ktoré by ich šírili ich susedom čo by vytváralo množstvo kópií rovnakých LSA správ na jednej sieti. Aby sa predišlo týmto problémom sú volené dve zariadenia v rámci každej podsiete[9]:

- `Designated Router DR` - smerovač v multi-access sieti, ktorý slúži ako centrálny bod pre výmenu smerovacích informácii. Taktiež reprezentuje multi-access sieť a do nej pripojené smerovače do zvyšku OSPF oblasti. Každý smerovač v danej podsieti nadväzuje susedstvo s týmto DR.
- `Backub Designated Router BDR` - slúži ako záložný DR pre prípad, že by aktuálny DR prestal správne pracovať.

Voľba DR a BDR prebieha na základe nakonfigurovanej *priority* na rozhraní, ktorým sa do danej podsiete pripája. Ak majú viaceré smerovače túto hodnotu rovnakú, volí sa na základe najvyššej hodnoty `router-id`.

Pre správne fungovanie medzioblastnej komunikácie či komunikácie s externými sieťami OSPF pozná ďalšie dva druhy smerovačov [7]:

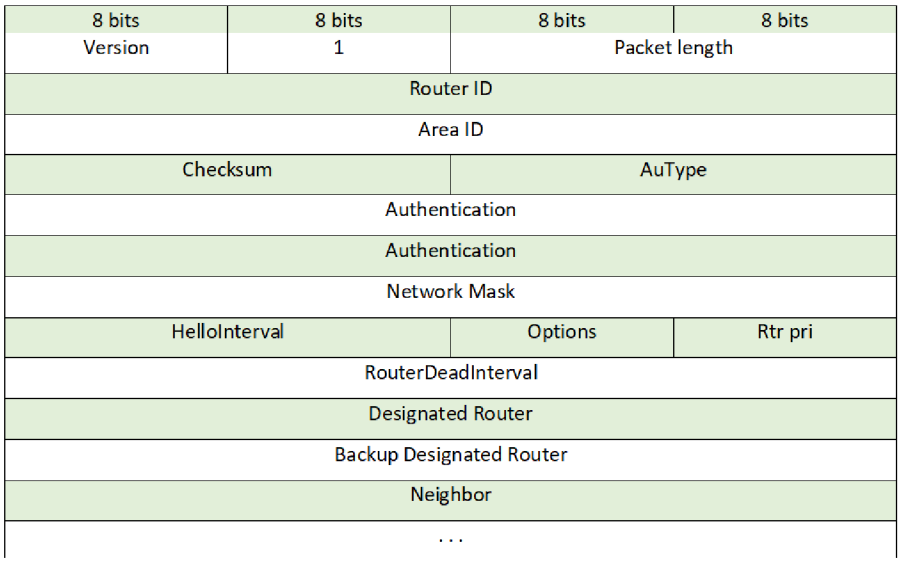
- `Area Border Router ABR` - Smerovač, ktorý je svojimi rozhraniami zapojený do dvoch a viac oblastí. V OSPF musí byť každý ABR zapojený aspoň jedným rozhraním do backbone oblasti. ABR sa chová ako centrálny bod sumarizácie, filtrovania či preposielania správ medzi oblasťami.
- `Autonomous System Boundary Router ASBR` - Smerovať na hranici medzi OSPF doménou a zvyškom siete, čo môže byť iný smerovací protokol alebo iný AS. ASBR sa chová ako hlavný uzol pre redistribúciu, filtrovanie či sumarizovanie smerovacích informácii mimo OSPF doménu.

2.6 Vytváranie susedstiev

OSPF nadväzuje susedstvo medzi susediacimi smerovačmi za účelom výmeny smerovacích informácií. Avšak nie každé dva susedné smerovače vytvoria susedstvo. [13]

2.6.1 Hello protokol

Hello protokol je zodpovedný za dynamické nadväzovanie a udržiavanie susedstva medzi smerovačmi. Zaisťuje tiež, že komunikácia medzi susedmi je obojsmerná. Pre svoje fungovanie využíva Hello pakety (obrázok 2.3). Hello pakety sú zasielané v pravidelných intervaloch (s predvolenou hodnotou 10 sekúnd) na všetkých rozhraniach smerovača. Obojsmerná



Obr. 2.3: Formát Hello paketu

komunikácia sa dá rozoznať podľa toho, že smerovač vidí samého seba medzi susedmi vo vnútri Hello paketu.

Hello protokol funguje rozdielne na rôznych typoch sietí. Na broadcast sieťach každý smerovač oznamuje seba samého pravidelným zasielaním Hello paketov ako mutlicast. To zariadeniam umožňuje, aby boli dynamicky objavovaní. Tieto Hello pakety obsahujú informácie o DR pre daný smerovač a zoznam smerovačov, ktorých Hello paket bol nedávno videný.

V P2P sieti Hello pakety na vytvorenie susedstva nestačia. Susedia sa musia vzájomne objaviť inou metódou, ako napríklad inverzné ARP, alebo musia byť nakonfigurované manuálne.

V non-broadcast sieťach Hello pakety nemôžu byť zasielané ako multicast. Preto sa pri NBMA sieťach rozosielajú každému susedovi po jednom. Pri `P2MP` je rozosielané riešené ako pri kolekcii P2P liniek.

Každý smerovač, ktorý sa potenciálne môže stať DR, si drží zoznam všetkých smerovačov pripojených do danej podsiete. Hneď ako sa rozhranie smerovača stane pre danú NBMA sieť aktívne, smerovač zašle Hello paket všetkým potenciálnym DR.

Aby zariadenia vytvorili susedstvo, musia sa niektoré ich hodnoty v Hello pakete zhodovať, sú to [14]:

Subnet - podsieť, v ktorej sa rozhrania smerovačov nachádzajú.

Hello a dead intervaly - hodnoty hello a dead časovačov musia byť identické. Viac o intervaloch v podsekcii 2.7

ID oblasti - oba smerovače musia byť v rovnakej oblasti

Typ oblasti - typy oblastí (stub alebo normálna) sa musia spolu zhodovať.

MTU - maximálna prenosová jednotka, oba smerovače musia mať na linke rovnakú hodnotu MTU.

nie Router ID - ID smerovačov musí byť pre danú smerovaciu doménu rozdielne.

2.7 Dátová štruktúra rozhrania

OSPF rozhranie je spoj medzi smerovačom a sieťou. Aj keď OSPF podporuje pripojenie viacero rozhraní jedného smerovača do jednej podsiete, v drvivej väčšine prípadov sa na pripojenie smerovača do jednej podsiete používa len jedno rozhranie. Smerovač však pochopiteľne môže byť svojimi rozhraniami pripojený do viacerých podsietí. Každé OSPF rozhranie je pripojené do siete v rámci nejakej oblasti a na základe toho nesie každý paket vyslaný týmto rozhraním ID oblasti, v ktorej sa rozhranie nachádza. [13]

Každé OSPF rozhranie má niekoľko dátových položiek, ktoré si smerovač musí pamätať a mať uložené. Sú to:

Typ - typ OSPF rozhrania udáva typ siete, do ktorej je toto rozhranie pripojené. Teda možné hodnoty sú broadcast, NBMA, P2P alebo P2MP.

Stav - udáva funkcionálnu úroveň rozhrania, teda rozhoduje či je možné na tomto rozhraní nadviazať plné susedstvo.

IP adresa a maska rozhrania - táto adresa bude braná ako zdrojová pre všetky OSPF správy, ktoré budú z tohto smerovača vyslané týmto rozhraním. Pre IPv4 platí, že každé rozhranie má len jednu unikátnu IP adresu.

ID oblasti - udáva ID oblasti, v ktorej sa toto rozhranie nachádza.

Hello, Dead a Rxmt interval - Hello interval udáva v sekundách v akej perióde bude smerovač zasielať Hello pakety. Dead interval tiež v sekundách špecifikuje, v ktorom čase musí sused prijať aspoň jeden Hello paket. Ak uplynie čas dead intervalu a žiaden Hello paket nebol prijatý, všetky susedné smerovače na danej linke sú považované za nedostupné. Dead interval je v predvolenej hodnote nastavený na štvornásobok hello intervalu. `Rxmt` interval vyjadruje čas medzi preposlaním LSA správ.

InfTransDelay - odhadovaný čas v sekundách kým smerovač bude schopný vyslať LSU paket týmto rozhraním. LSA správy obsiahnuté v tejto LSU správe budú mať pred vyslaním správy svoj vek zvýšený práve o túto hodnotu. Hodnota musí byť väčšia ako 0. Viac o LSA správach v sekcii 2.9.

Priorita smerovača - 8-bitový bezznamienkový integer. Keď dva smerovače vytvoria spojenie, oba sa pokúsia stať sa DR. Prednosť má však ten s vyššou hodnotou priority. Smerovač s hodnotou priority nastavenú na 0 sa nezúčastňuje voľby DR.

Hello a Wait časovač - Hello časovač spôsobuje pravidelné zasielanie Hello paketov. Spúšťa sa každých *hello interval* sekúnd. Wait časovač je jednorazový časovač, ktorý spôsobuje, aby rozhranie opustilo stav `Waiting`.

Zoznam susediacich smerovačov - zoznam smerovačov pripojených do rovnakej siete ako je rozhranie tohto smerovača.

DR a BDR smerovač - `DR` a `BDR` pre sieť, v ktorej sa toto rozhranie nachádza. Viac o `DR` a `BDR` v sekcii 2.5.

Cena rozhrania - cena zasielania paketov skrz toto rozhranie, vyjadrené v link state metrike. Hodnota musí byť vyššia ako 0.

Typ a kľúč autentifikácie - v rámci danej podsiete.

Počas behu protokolu prechádza každé rozhranie viacerými stavmi, v závislosti od aktuálnej situácie nadväzovania vzťahov a prijatých správ. Túto vlastnosť najlepšie popsiuje konečný automat dostupný v prílohe D.

2.8 Dátová štruktúra suseda

Každý OSPF smerovač si vedie záznamy o svojich susedoch a jeho vzťahu k nim vo svojej `Adjacency` databázy (databázy susedstiev). Každé takéto susedstvo je popísané v samostatnej dátovej štruktúre suseda. Každé susedstvo je viazané na rozhranie OSPF smerovača. Identifikuje sa podľa ID smerovača alebo susedovej IP adresy. Takže ak majú dve zariadenia viaceré spoločné podsiete, je zaistené, že každé susedstvo medzi týmito dvoma smerovačmi bude mať vlastnú dátovú štruktúru.

Dátová štruktúra suseda má niekoľko dátových položiek. Sú to :

Stav - jeden z ôsmych stavov, ktorý indikuje postup v nadväzovaní susedstva

Časovač nečinnosti - časovač s dĺžkou *dead interval* sekúnd. Indikuje, že od suseda nebol dlhšiu dobu prijatý žiaden Hello paket.

Master/Slave - keď si dva smerovače vymieňajú databáze, nadväzujú vzťah master-slave. Master zasiela DD paket ako prvý a je jediný kto môže v prípade potreby správu znovu preposlať. Slave môže jedine odpovedať na DD packet mastera. Master/slave vzťah sa dohaduje v stave `ExStart`.

DD sekvenčné číslo - sekvenčné číslo `DD` paketu, ktorý bol susedovi naposledy zaslaný.

Naposledy prijatý DD paket - slúži pre rozlíšenie či prijatý DD paket je duplikát.

ID suseda - ID susedného smerovača získané z prijatého Hello paketu.

Options suseda - voliteľné OSPF možnosti, ktoré sused podporuje.

Priorita suseda - priorita susedného smerovača získaná z Hello paketu.

IP adresa suseda - IP rozhrania susedného smerovača.

Susedov DR - ID smerovača, ktorého sused berie ako svojho DR.

Susedov BDR - ID smerovača, ktorého sused berie ako svojho BDR.

Tak ako rozhranie, aj sused prechádza počas nadväzovania susedstva viacerými stavmi. Opäť danú skutočnosť najlepšie popisuje konečný automat v prílohe C

2.9 Rozbor LSA správ

`Link-state advertisements` (`LSAs`) sú stavebné bloky OSPF topologickej databáze - `LSBD`. Samostatne sa chovajú ako záznamy a v kombinácii popisujú celú OSPF sieť. LSA je dátová štruktúra, ktorá obsahuje hlavičku s informáciami (pôvodca LSA, identifikátor LSA, sekvenčné číslo, typ a iné) a dáta. Každý smerovač generuje LSA s informáciami o sebe, priamo pripojených linkách s aktuálnym stavom a zoznam susedných smerovačov. Správy LSA sa delia do 11 skupín. Tabuľka 2.1 stručne popisuje toto rozdelenie. Bližšie rozoberanie väčšiny LSA správ je v kapitole 3.4. Informácie boli čerpané z [13]

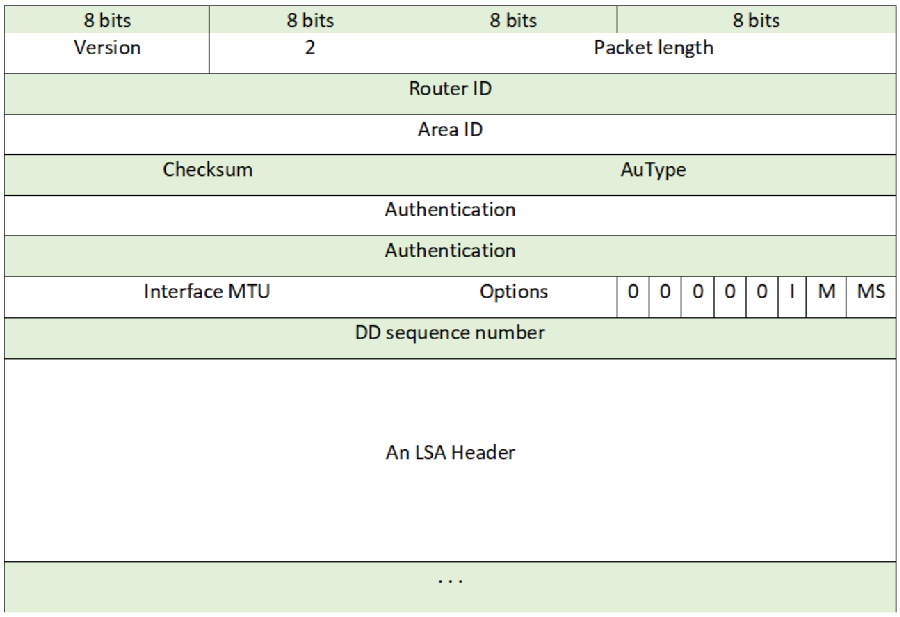
Typ	Názov	Popis
1	Router	Router LSA obsahuje stav a cenu rozhraní pripojených do oblastí a DR pre danú podsieť.
2	Network	Network LSA je generovaná DR smerovačom pre každú broadcast a NBMA sieť. Stručne popisuje všetky pripojené smerovače vrátane DR do danej sieťe.
3	Summary	Typ 3 summary-LSA správy su tvorené ABR smerovačmi. Popisujú cesty do sietí mimo danú oblasť.
4	ASBR Summary	Typ 4 LSA správy určujú cestu k ASBR hraničnému smerovaču.
5	External	External-LSA správy vytvára ASBR smerovač a oznamuje nimi smerovacie informácie redistribuované do OSPF z iného smerovacieho protokolu
6	Group Membership	Vytvorený pre multicast extension of OSPF (MOSPF)
7	NSSA External	Špeciálny typ LSA správ používaný v NSSA sieťach.
8	External Attributes	Typ 8 LSA správy sa používajú pre súčinnosť OSPF a BGP.
9-11	Opaque	Typ 9 a 11 LSA správy sú vytvorené pre budúce vylepšenia OSPF protokolu.

Tabuľka 2.1: LSA typy správ

2.10 Synchronizácia databáz

Po tom, ako smerovače nadviažu susedstvo, prvý ďalší krok predstavuje synchronizácia ich link-state databáz. Každý smerovač popíše svoju databázu zaslaním DD paketov svojmu susedovi (formát paketu je na obrázku 2.4).

Každý DD paket obsahuje skupinu LSA správ patriacich do databázy odosielajúceho smerovača. Keď sused vidí, že prijaté LSA je novšie ako to, ktoré má uložené v databázy, poznamená si, že má požadovať novšie LSA.



Obr. 2.4: Formát Database Description paketu

Toto zasielanie a prijímanie DD paketov sa nazýva *Database Exchange Process* (teda synchronizácia databáz). Počas tohoto procesu dva smerovače vytvoria `master/slave` vzťah. Master inkrementuje sekvenčné číslo a zasiela DD pakety slave smerovaču. Slave potvrdzuje prijaté správy zasielaním vlastných DD paketov s rovnakým sekvenčným číslom, aké prijal od smerovača master. Len master môže znovu zasielať nepotvrdené DD pakety. Štruktúra DD paketu je za hlavičkou nasledovná [13] [14]:

0 - toto sú rezervované polia a musia byť vždy nastavené na 0.

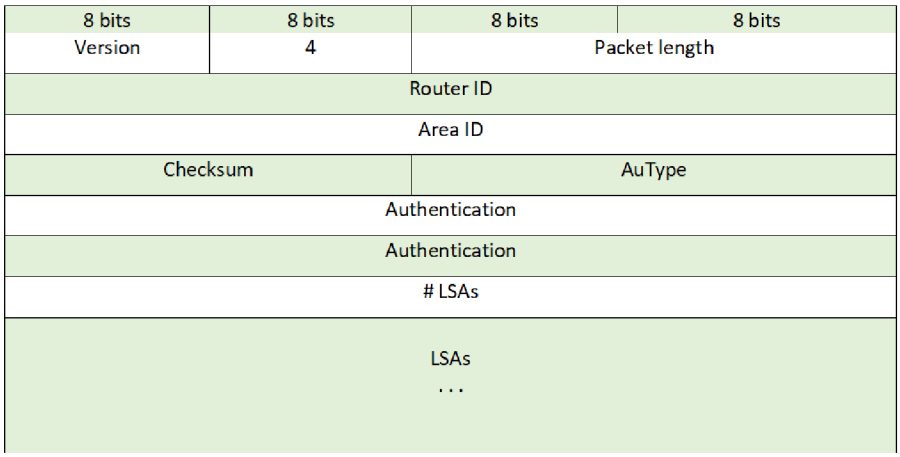
Options - dodatočné vlastnosti podporované smerovačmi. Niektoré nastavenia sú povinné, iné dobrovoľné. Avšak ak sú rozdiely v nastavení *options* medzi susednými smerovačmi, zvyčajne nie sú schopné nadviazať spojenie, alebo smerovač, ktorý vypočítava SPF nezahrnie do výpočtu smerovač s iným nastavením *options*.

I-bit - Inicializačný bit indikuje, že tento paket je prvý DD paket,

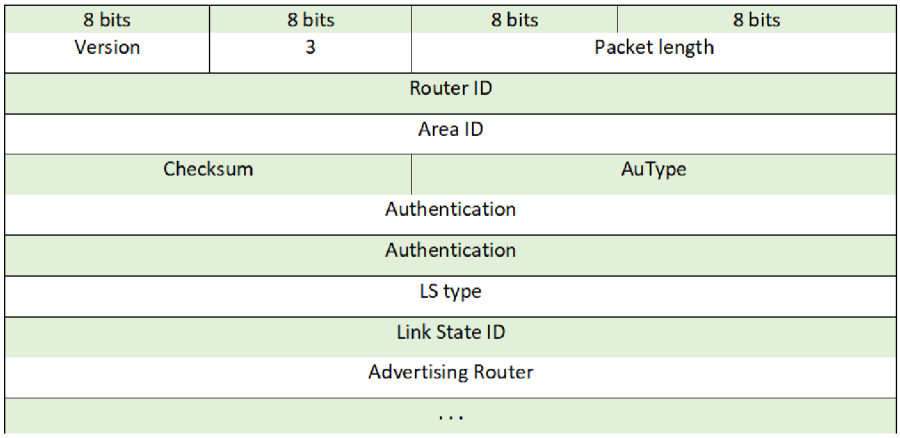
M-bit - *More* bit indikuje, že tento paket nie je posledný a ďalší DD paket bude ešte následovať.

MS-bit - Master/Slave bit určuje, ktorý smerovač je master, a ktorý slave. Nastavením tohto bitu na 1 smerovač indikuje, že on je master.

Po synchronizácii databáz má každý smerovač zoznam LSA správ, ktorých má sused novšiu inštanciu. Tieto LSA záznamy si smerovač vyžiada pomocou LSR správ (obrázok 2.5). Požadované LSA je špecifikované pomocou LS typu, LS ID a odosielajúceho smerovača. Táto unikátna špecifikácia jednotlivých LSA je chápaná ako požiadavka na najnovší LSA záznam. LSA prijaté od suseda po jeho vyžiadaní môže byť novšie, ako to, ktoré si smerovač poznačil z DD paketu. LSR správy, ktoré neboli odbavené sú znovu zaslané po vypršaní *RxmtIntervalu*



Obr. 2.6: Formát Link State Update paketu



Obr. 2.5: Formát Link State Request paketu

Na doručenie vyžiadaných LSA správ sa používajú typ 4 `Link State Update` (`LSU`) pakety (obrázok 2.6). LSU správy sa rozošlú do siete ako multicast, avšak len do vzdialenosti jeden hop od zdroja. Jedna LSU správa môže niesť jeden a viac LSA správ.

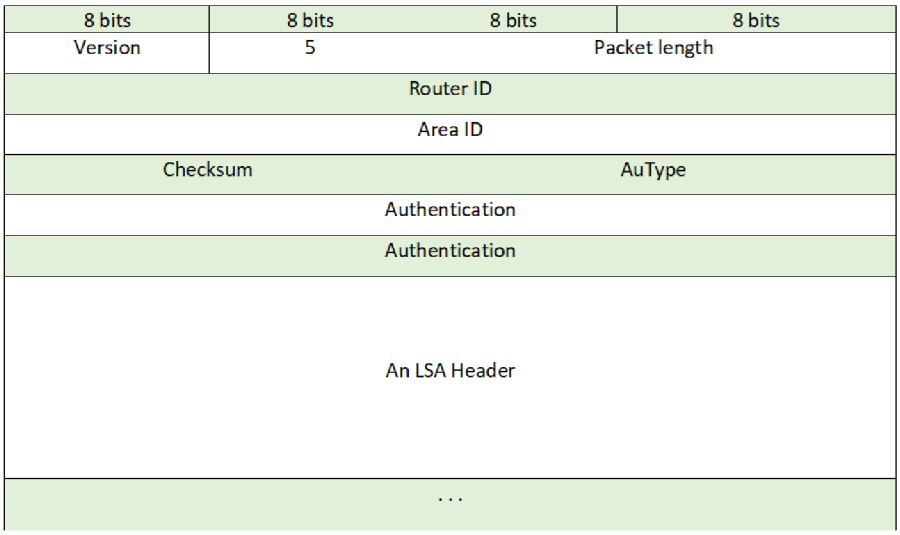
Prijaté a spracované LSA záznamy sa následne potvrdzujú `Link State Acknowledgement` (`LSAck`) paketmi s LS typom 5 (obrázok 2.7). Každá LSAck správa v sebe nesie hlavičku LSA správy, ktorú potvrdzuje. Viaceré LSA správy môžu byť potvrdené jednou LSAck správou. LSAck sa zasielajú ako multicast na adresu `AllSPFRouter` (`224.0.0.5`) alebo na adresu `AllDRouters` (`224.0.0.6`), alebo ako unicast.

Keď sú všetky LSR správy odbavené a databáze oboch smerovačov sú synchronizované, susedstvo medzi smerovačmi je nadviazané a plne funkčné.

2.11 Shortest Path First

V tejto sekcii bude predstavený spôsob budovania smerovacej tabuľky tak ako ho definuje OSPF protokol.

Informácie uložené v Link State databázy smerovač využíva k výpočtu ciest do vzdialených sietí. Link State databáza v podstate popisuje v rámci AS orientovaný graf s uzlami



Obr. 2.7: Formát Link State Acknowledgement paketu

pre smerovače a siete. Každý smerovač si generuje svoju smerovaciu tabuľku z grafu vypočítaného stromu metódou `shortest-path-first` (`SPF`), kde seba berie ako zdrojový uzol. Pre výpočet tohto stromu OSPF používa Dijkstrov algoritmus. Vypočítaný strom definuje celú a zároveň najkratšiu cestu do akejkoľvek cieľovej siete či smerovača. Avšak v smerovacom procese sa vyžíva len *next hop*. Pre spracovanie ciest do externých sietí v inej oblasti či mimo AS sa značí *next hop* a vzdialenosť k smerovaču , ktorý tieto externé siete oznamuje. V prípade P2P sietí má smerovač uloženú samostatnú cestu ku každému zariadeniu. .

2.11.1 Dijkstrov algoritmus

Na Dijktrov algoritmus možno pozerať ako na prehľadávanie do šírky, pri ktorom sa vlna nešíri na základe počtu hrán od zdroja, ale vzdialenosti od zdroja (v zmysle váhy hrán). Táto vlna preto spracováva len tie uzly, ku ktorým bola nájdená najkratšia cesta [11].

Dijkstrov algoritmus si uchováva všetky uzly v prioritnej fronte radené podľa vzdialenosti od zdroja. V prvej iterácii má iba zdroj vzdialenosť 0, všetky ostatné uzly nekonečno. Algoritmus v každom svojom kroku vyberie z fronty uzol s najvyššou prioritou (najnižšiu vzdialenosťou od už spracovanej časti) a zaradí ho medzi spracované uzly. Potom prejde všetkých jeho doteraz nespracovaných potomkov a ak nie sú už vo fronte zahrnuté, pridá ich a overí či sa ich vzdialenosť od zdroja neznížila. To znamená, že pre všetkých potomkov overuje:

$$d(v) + w(u, v) < d(u) \tag{2.1}$$

kde u a v značia uzly grafu, $d(v)$ značí vzdialenosť uzlu u od zdrojového uzla a $w(u, v)$ značí váhu hrany (u, v) respektíve vzdialenosť uzla u od uzlu v.

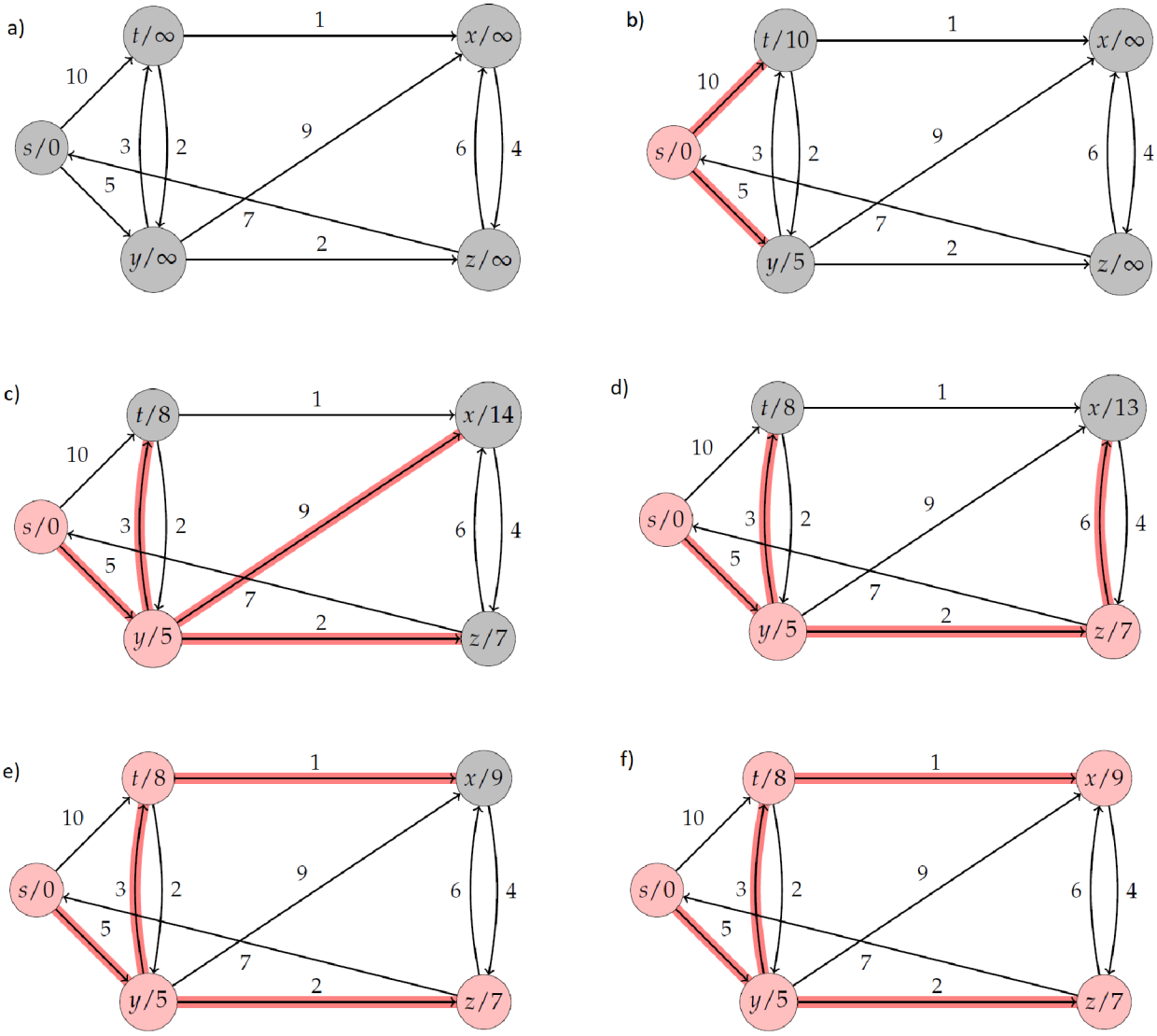
Ak nerovnosť platí, tak danému potomkovi nastaví novú vzdialenosť a označí za jeho predka spracovávaný uzol. Po priechode všetkých potomkov algoritmus vyberie z fronty uzol s najvyššou prioritou (najnižšou vzdialenosťou od zdrojového uzlu) a celý krok opakuje. Algoritmus končí v okamžiku, kedy sú spracované všetky uzly, respektive keď je prioritná fronta prázdna.

Dijkstrov algoritmus je použiteľný len v prípade, ak graf neobsahuje záporne ohodnotené hrany. V takom prípade je schopný garantovať, že pri spracovaní uzlu bola nájdená najkratšia možná cesta k uzlu.

Zložitosť Dijkstrovho algoritmu závisí na implementácii prioritnej fronty. V prípade, že je implmentovaná pomocou sekvenčného vyhľadávania, je zložitosť algoritmu $O|n|^2$, pri použití binárneho stromu je to $O(|n|log_2|m|)$, pričom n značí počet uzlov a m značí počet hrán.

2.11.2 Príklad výpočtu Dijkstrovho algoritmu

Predpokladajme graf s piatimi uzlami a desiatimi hranami poprepájaný tak, ako je znázornené na obrázku 2.8 bod *a)*. Zdrojový uzol s nadobudne hodnotu 0, ostatné uzly hodnotu nekonečno. Výpočet dijkstrovho algoritmu tak, ako je popísaný vyššie je následne znázornený bodmi b) až f), kde na konci pozná zdrojový uzol s najkratšiu cestu ku každému uzlu grafu.



Obr. 2.8: Príklad hľadania najkratšej cesty pomocou dijkstrovho algoritmu[17]

2.11.3 Štruktúra záznamov smerovacej tabuľky

Výpočtom SPF vznikajú záznamy ktoré sa ukladajú do smerovacej tabuľky. Takáto dátová štruktúra obsahuje všetky informácie potrebné k smerovaniu IP paketu do cieľovej destinácie. Pri smerovaní paketu sa hľadá najlepšia zhoda so záznamom v smerovacej tabuľke. Tento záznam potom poskytuje next-hop k cieľového zariadeniu. OSPF taktiež poskytuje *default* cestu, ktorá sa bude vo výsledku zhodovať s každou IP adresou.

Štruktúra záznamu smerovacej tabuľky sa skladá z niekoľkých polí. Sú to:

Cieľový typ - je buďto *network* (sieť), alebo *router* (smerovač). Pre smerovanie dát sa v podstate využíva len sieťových záznamov. Východzia cesta sa radí tiež medzi záznamy typu sieť. Záznamy typu smerovač sú cesty len k AB a ASB smerovačom, ktoré slúžia pre smerovanie do iných oblastí a iných AS [13].

Cieľové ID - závisí od cieľového typu. Pre sieťový typ je identifikátor IP adresa siete. Pre typ smerovač je identifikátor ID OSPF smerovača.

Maska siete - je definovaná len pre cieľ typu sieť.

Voliteľné pole - ak je cieľom smerovač, toto pole definuje volitelné nastavenia podporované cieľovým OSPF smerovačom.

Oblasť - indikuje oblasť cieľovej stanice či siete. Pre záznamy do externých sietí iných AS toto pole nie je definované.

Zvyšok záznamu smerovacej tabuľky popisuje množinu rôznych ciest do cieľovej

Typ cesty - sú štyri možné typy ciest. Intra-area, teda v rámci oblasti, inter-area, teda medzi-oblastný, a externý typ 1 a typ 2.

Cena - pre všetky cesty s výnimkou externých typu 2, táto hodnota popisuje celkovú cenu cesty do cieľovej siete/smerovača. Počíta sa ako suma jednotlivých cien naprieč celou trasou.

Cena typu 2 - je valídna len pre externé cesty typu 2. Táto hodnota je ohlasovaná ASB smerovačom.

Link state pôvod - toto pole indikuje LSA správu (typu 1 alebo 2), ktorá sa priamo odkazuje na cieľovú sieť/smerovač.

Ak existuje viacero ciest do cieľovej siete/smerovača s rovnakou cenou aj typom cesty, tak sú uložené v rámci jedného záznamu smerovacej tabuľky. Cesty sú v takom prípade rozdielne v kombinácii polí:

Next hop - výstupné rozhranie, ktoré má smerovač použiť pre dosiahnutie cieľovej stanice. V prípade broadcast, P2MP či NBMA sietí je v next hop uložená aj IP adresa nasledujúceho smerovača na ceste k cieľu.

Oznamujúci smerovač - valídne len pre medzi-oblastné a AS externé cesty. Toto pole indikuje ID smerovača, ktorý ohlasuje summary-LSA či AS-external LSA k cieľu.

Kapitola 3

OSPF pre IPv6

Táto kapitola poukazuje na rozdiely medzi OSPFv2, ktorý bol popisovaný doteraz a OSPFv3, tj. novším OSPF s podporou pre IPv6. Základné princípy OSPF protokolu zostávajú. Avšak niektoré zmeny boli potrebné či už kvôli rozdielom sémantiky IPv4 a IPv6 alebo jednoducho kvôli nárastu celkového počtu IP adries.

Napriek väčšiemu počtu IPv6 adries, väčšina OSPFv3 paketov je rovnako kompaktných ako pre OSPFv2. Obmedznia veľkosti paketov sa zvolnili a spracovanie *option* sa stalo viac flexibilné.

IPv6 používa pojem *linka* pre označenie média skrz ktoré uzly môžu komunikovať na linkovej vrstve. Rozhrania sa pripájajú na tieto linky. Viacero IPv6 podsietí môže byť priradených na jednu linku a dva uzly môžu komunikovať priamo skrz jednu linku aj v prípade, že sa ich rozhrania nenachádzajú v rovnakej IPv6 podsieti. Tieto informácie boli čerpané z [8].

3.1 Address family

Prechod z IPv4 na IPv6 nie je skokový a v rámci pomalého prechodu je potrebné, aby smerovače, respektíve smerovacie protokoly podporovali súčasný beh oboch IP protokolov. V prípade OSPF by bolo potrebné, aby bežali dva samostatné OSPF procesy - jeden pre IPv4, druhý pre IPv6. Práve pre zjednodušenie tohto problému sa správa oboch procesov stretáva pod jednou strechou.

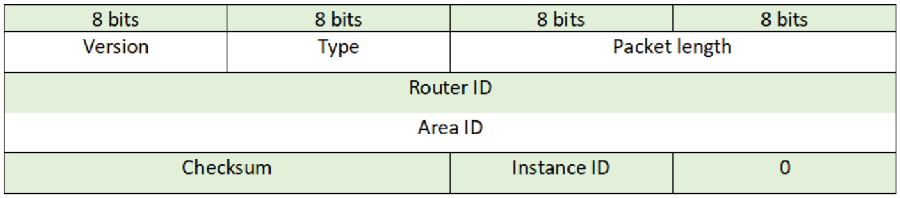
`Address family` (`AF`) pre OSPFv3 pridáva podporu komunikácie pre IPv4 aj IPv6 unicast aj multicast. S AF môžu na smerovači bežať na jednom rozhraní dva procesy, jeden pre AF IPv4, druhý pre AF IPv6. Nie je podporované aby pod jedným IPv4 alebo IPv6 OSPFv3 procesom bežalo viacero inštancií[1] [5].

Spravovanie IPv4 aj IPv6 pomocou AF zjednodušuje plánovanie implementácie, šetrí výpočtové cykly procesoru, ktoré by boli potrebné pre dva samostatne bežiace protokoly a zjednodušuje bezpečnostnú politiku [6].

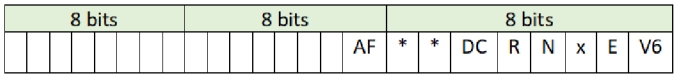
3.2 Zmeny vo formáte paketov

OSPFv3 beží priamo nad IPv6. Avšak všetka adresná sémantika bola z hlavičky paketu odstránená čím OSPF pakety robí zcela nezávislé od sieťového protokolu. Všetky adresné informácie sú teraz obsiahnuté v rôznych LSA typoch.

[1] toto platí pre OSPFv3 podľa definície Cisco Systems, Inc.



Obr. 3.1: Formát hlavičky OSPFv3 paketu



Obr. 3.2: OSPFv3 Options pole s podporou AF

Zmenou prešla hlavička OSPFv3 paketov. V OSPFv3 bola autentifikácia a všetky polia, ktoré pre ňu slúžili zcela odstránené. OSPFv3 sa zcela spolieha na `IP security`. Keďže OSPFv3 môže na jednej linke bežať vo viacerých inštanciách, nesie hlavička paketu v sebe ID inštancie, ktorej sa paket týka. Verzia udáva verziu OSPF protokolu, čiže pre OSPFv3 nesie hodnotu 3. Typ určuje o aký paket sa jedná (DD paket, Hello paket, LSR, LSU, LSA). Zmenenú OSPFv3 hlavičku možno vidieť na obrázku 3.1

Hello paket pre OSPFv3, zobrazený na obrázku 3.3, prešiel určitými úpravami. Keďže viaceré IPv6 podsiete môžu byť priradené na jednu linku, maska siete nie je viac potrebná a preto bola odstránená. Namiesto toho Hello paket využíva 32-bitové ID rozhrania, ktoré jednoznačne identifikuje rozhranie pripojené na danú linku. OSPFv3 `Option` pole sa z 8-bitov zväčšilo na 24-bitov a pole pre *dead interval* bolo z 32-bitov zmenšené na 16-bitov. Tieto informácie sú uvádzané na základe [1]. DD pakety, LSR a LSU pakety sa vo svojej podstate nezmenili. Zmenila sa len ich hlavička, ktorá bola popísaná vyššie. S prechodom na OSPFv3, prešlo *options* pole (viď. obrázok 3.2) taktiež určitými zmenami. Celkovo toto pole zaberá 24 bitov, avšak nie všetky sú využité a teda niektoré bývajú konštantne nastavené na 0. Zvyšné bity sú [8] [12]:

V6-bit - nastavením na nulu smerovač indikuje že nemá byť zahrnutý do výpočtu IPv6 smerovania.

E-bit - indikuje spôsob akým sa má šíriť AS-external LSA v rámci OSPFv3 domény.

x-bit - kedysi používaný pre MOSPF, má vždy niesť hodnotu 0, ponechaný pre budúce účely.

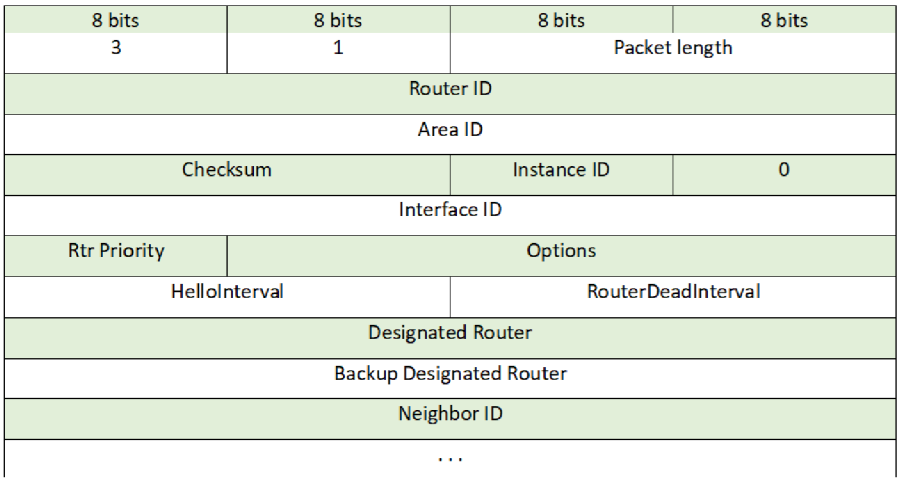
N-bit - indikuje či je smerovač pripojený do NSSA oblasti.

R-bit - indikuje či je smerovač aktívny v rámci zasielania informácii o okolitých sieťach a výpočte SPF.

DC-bit - tento bit súvisí so spracovaním *demand circuits*

***-bit -** rezervované pre migráciu OSPFv2 vlastností

AF-bit - indikuje podporu pre *address families*.



Obr. 3.3: Formát OSPFv3 Hello paketu

3.3 Flooding scope

`Flooding scope`, alebo rozsah šírenia paketov sa rozšíril o ďalšie typy a je explicitne daný pre každý typ LSA správy. Sú teda tri rozdielne rozsahy šírenia paketov:

- **Link-local scope**, alebo linkový rozsah šírenia znamená, že LSA správa je šírená len na lokálnej linke a nikam ďalej. Používa sa pre nové link-LSA správy.
- **Area scope**, alebo oblastný rozsah šírenia znamená, že LSA správa sa šíri len do rozsahu jednej OSPF oblasti. Používa sa pre router-LSA, network-LSA, inter-area-prefix-LSA, inter-area-router-LSA a intra-area-prefix-LSA správy.
- **AS scope**, alebo rozsah autonómneho systému znamená, že správa sa šíri celou doménou. Používa sa pre AS-external-LSA správy.

3.4 LSA správy

V tejto sekcii budú bližšie rozobrané niektoré LSA správy a ich zmeny oproti OSPFv2.

OSPFv3 využíva rovnaké správy ako OSPFv2, avšak niektoré správy boli premenované a tiež dve nové pribudli. Tabuľka 3.4 znázorňuje tieto zmeny.

Vzhľadom na to, že táto práca nadväzuje na minulú, bolo v prevzatej implementácii odhalených niekoľko chýb, ktoré sú bližšie popísané v kapitole 5. Kvôli týmto chybám je nižšie rozpísaný bližší rozbor niektorých LSA správ.

Každý smerovač bude v rámci oblasti šíriť svoju Router-LSA správu. V tejto správe sú obsiahnuté všetky priamo pripojené siete skrz ID smerovača a ID rozhrania DR smerovača pre danú sieť. Teda ak je jedno zariadenie pripojené v rámci jednej oblasti do dvoch rôznych OSPF sietí, jeho router-LSA správa bude celkovo obsahovať informácie o dvoch DR smerovačoch, jeden pre každú sieť.

Pre broadcast siete je Network-LSA správa generovaná jedna v rámci každej kolíznej domény. Generuje ju DR a pod hlavičkou obsahuje zoznam pripojených zariadení vyjadrených skrz ich ID smerovača. V zozname je uvedené aj ID smerovača, ktorý túto správu generuje.

	LSA Function Code	LSA Type
Router-LSA	1	0x2001
Network-LSA	2	0x2002
Inter-Area-Prefix-LSA	3	0x2003
Inter-Area-Router-LSA	4	0x2004
AS-External-LSA	5	0x4005
Group-Membership-LSA	6	0x2006
Type-7-LSA	7	0x2007
Link-LSA	8	0x2008
Intra-Area-Prefix-LSA	9	0x2009

Renamed

New

Obr. 3.4: Tabuľka LSA správ pre OSPFv3 [16]

Inter-area-prefix-LSA je ekvivalent pre OSPFv2 typ 3 LSA. Vytvára ho ABR smerovač a popisuje cesty do sietí v rámci iných oblastí pomocou prefixov. Prefix je popísaný dĺžkou prefixu, *options* prefixu a samotná adresa prefixu.

Intra-area-router-LSA je ekvivalent pre OSPFv2 typ 4 LSA. Oznamuje cestu k ASBR smerovačom na okraji OSPF domény avšak v rámci rovnakého AS.

Link-LSA je nová LSA správa, ktorá popisuje pripojené fyzické linky smerovača a ich globálne a link-local IPv6 adresy. Každá linka je popísaná jednou Link-LSA správou. Úlohou Link-LSA správy je poskytovať tieto informácie o linkách ostatným smerovačom, ktoré sú priamo pripojené na zdrojový smerovač. Majú len linkový rozsah šírenia správy.

Keďže všetka adresná sémantika bola v OSPFv3 z LSA správ odstránená, využíva sa k tomuto účelu nová Intra-area-prefix-LSA správa. Táto správa oznamuje jeden alebo viac IPv6 adresných prefixov ktoré sú so smerovačom zviazané. Oproti Link-LSA má oblastný rozsah šírenia a preto slúži pre oznamovanie adresných prefixov vzdialenejším uzlom, nie len priamo pripojeným.

Kapitola 4

Konfigurácia OSPFv3 na Cisco zariadeniach

V nasledujúcej kapitole bude znázornená konfigurácia protokolu OSPFv3 na Cisco zariadeniach. Predstavené budú dve topológie. Jedna jednoduchšia, pre ľahšie pochopenie základnej konfigurácie zariadení, druhá pre zložitejšiu a detailnejšiu konfiguráciu. Verzia IOSu na všetkých smerovačoch je 15.4. Bude demonštrovaná konfigurácia a prediskutované jej následky. Informácie pre túto kapitolu boli prevažne čerpané z [7] a [16].

Ako prvá bude predstavená topológia o troch smerovačoch prepojených jedným prepínačom, umiestnených do spoločnej `area 0`. Znázornená je na obrázku č. 4.1 Prepínač je ponechaný s jeho predvolenou konfiguráciou a viac sa ním nebudeme zaoberať. Každý zo smerovačov má nakonfigurované rozhrania pre pripojenie smerovača do siete. Nakonfigurovaná je ako IPv4 tak IPv6 adresa. Smerovač `R1` má navyše nakonfigurovaný *loopback0*. Aby však bolo možné pracovať s IPv6 adresami, je potreba zadať príkaz:

Router(config)# ipv6 unicast-routing

ktorý na danom smerovači povolí IPv6 smerovanie.

4.1 Spustenie protokolu OSPFv3 Address-Family

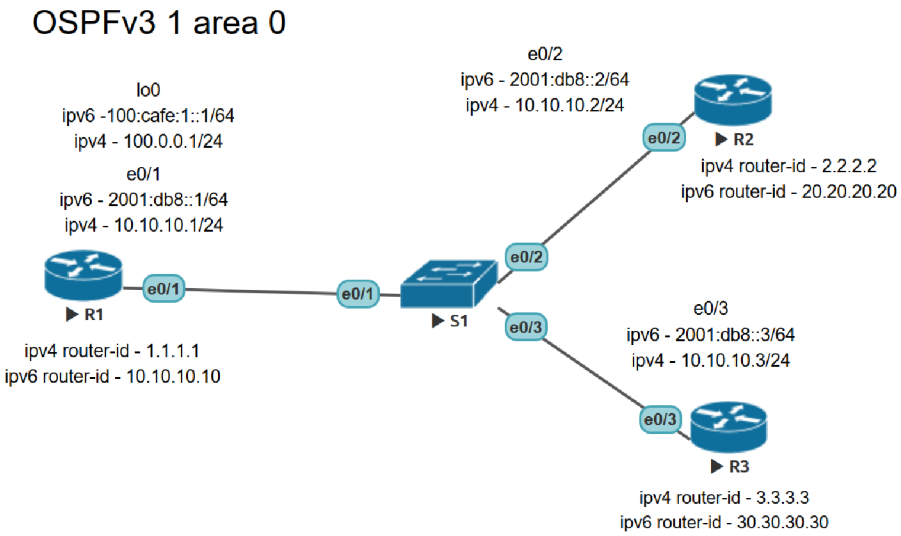
Ako prvé je potreba na každom smerovači aktivovať protokol OSPFv3. To dosiahneme príkazom:

Router(config)# router ospfv3 *Process-ID*

kde *Process-ID* predstavuje číslo OSPFv3 procesu, ktorý chceme vytvoriť. OSPFv3 AF umôžňuje vytvoriť pre každú AF samostatný proces avšak len s jednou inštanciou. Táto vlastnosť umôžňuje, aby smerovače vytvorili samostatné topológie pre IPv4 a IPv6 smerovanie. Pre tento príklad bude vytvorená jedna inštancia protokolu a teda príkaz:

Router(config)# router ospfv3 1

na smerovači `R1` vytvorí OSPFv3 proces s `ID 1`.



Obr. 4.1: Schéma topológie pre konfiguráciu OSPFv3

V tejto sekcii je možné konfigurovať či meniť vlastnosti protokolu, ktoré nadobudnú platnosť pre obe *address-family*. Ak by sme chceli, aby konfigurácia platila len pre jednu *address-family*, je potreba prejsť do sekcie danej *address-family* príkazom:

Router(config-router)# address-family *ipvX* **unicast**

a konfiguráciu zadať tam, kde *ipvX* značí voľbu pre IPv4 alebo IPv6. Konfigurovať môžeme pasívne rozhrania, autentifikáciu, ID smerovača a mnoho iných. Z topológie je vidieť, že každé zariadenie má rozdielne `router-Id` pre IPv4 a pre IPv6. Ak by sme teda chceli podľa topológie zmeniť `router-Id` smerovača `R1`, zadáme:

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

Router(config-router-af)# router-id 1.1.1.1

Router(config-router)# address-family ipv6 unicast 1

Router(config-router-af)# router-id 10.10.10.10

Ako možno vidieť, identifikátor smerovača sa zapisuje v tvare IPv4 adresy. Ak by tento identifikátor nebol explicitne nakonfigurovaný, je možné ho pre *address-family* IPv4 automaticky odvodiť z IP adresy jedného z rozhraní smerovača. Pre *address-family* IPv6 môže byť odvodený rovnako tak, avšak len v prípade že na danom smerovači sa nejaká IPv4 adresa nachádza. Ak nie, je potreba ju zadať ručne. Je tiež dobrým zvykom konfigurovať nečinné porty ako takzvané *passive-interface*. Tieto rozhrania sú následne brané ako stub. Na rozhrania s týmto nastavením nie sú posielané OSPF smerovacie informácie a teda predstavujú tiež určitú formu zabezpečenia a zníženia záťaže linky. Je možný dvojaký prístup

ku konfigurácii. Buďto sa vychádza z predvoleného nastavenia, kedy sú všetky linky aktívne a príkazom:

Router(config-router-af)# passive-interface *Interface-type Interface-id*

nastavíme konkrétne rozhranie ako pasívne, alebo príkazom:

Router(config-router-af)# passive-interface default

nastavíme všetky rozhrania ako pasívne a následne príkazom:

Router(config-router-af)# no passive-interface *Interface-type Interface-id*

aktivujeme jednotlivé rozhrania.

4.2 Pridanie rozhrania do OSPFv3 Address-Family

Po vytvorení procesu OSPFv3 je potreba nakonfigurovať tento protokol na konkrétnych rozhraniach. To sa robí príkazom:

Router(config-if)# ospfv3 *Process-ID ipvX* **area** *Area-ID*

kde *Process-ID* značí ID OSPF procesu, *ipvX* značí voľbu pre IPv4 alebo IPv6 a *Area-ID* značí ID pre oblasť v rámci daného OSPF procesu. Teda pre danú topológiu bude na smerovači `R1` potrebné na rozhranie Ethernet0/0 zadať príkazy:

Router(config-router-if)# ospfv3 1 ipv4 area 0

Router(config-router-if)# ospfv3 1 ipv6 area 0

čím sa toto rozhranie pridá do databázy OSPFv3 rozhraní jak pre IPv4 tak aj pre IPv6. Rovnaké príkazy je na `R1` potreba zadať pod rozhraním *loopback0*. S týmto rozhraním je však vždy zaobchádzané ako so stub sieťou.

Zcela obdobné príkazy je potrebné zadať na zvyšných dvoch smerovačoch. Následne všetky tri zariadenia vytvoria spojenie, určia si svojho DR a vymenia svoje LSA správy. Nadviazanie spojenia je možné overiť príkazom:

Router# show ospfv3 neighbor

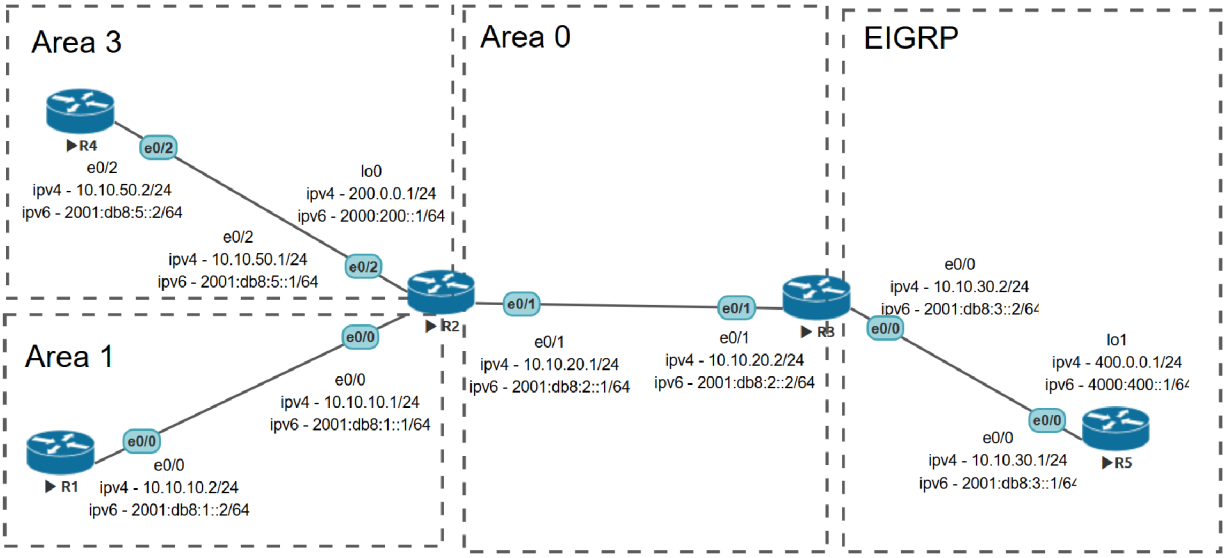
Link-state databázu, teda databázu LSA správ je možné overiť príkazom:

Router# show ospfv3 database

Túto databázu by po skonvergovaní mali mať všetky smerovače rovnakú.

4.3 Vytvorenie viacerých oblastí

V nasledujúcej topológii, ktorá je znázornená na obrázku č. 4.2. bude predstavená komunikácia medzi oblasťami, redistribúcia smerovacej tabuľky z externého smerovacieho protokolu, konkrétne protokolu EIGRP. Na obrázku č. 4.2 je vidieť prepojenie piatich smerovačov. Ich konfigurácia rozhraní odpovedá informáciám na obrázku. Na smerovačoch 1 - 4 je spustený protokol OSPFv3 s ID 1. Ako prvé bude predstavené prepojenie medzi oblasťami. Ako už bolo povedané, u OSPF je možné veľkú topológiu siete deliť do menších oblastí. Oblasť 0 je takzvaná bakcbone area, v okolí ktorej môžu byť nakonfigurované iné oblasti. Prepojenie jednotlivých oblastí je však možné len skrz oblasť 0 (výnimkou je použtie virtuálnych liniek). V topológii sú teda načrtnuté tri oblasti a to oblasť 0, 1 a 3. Ich konfigurácia je



Obr. 4.2: Schéma topológie pre konfiguráciu viacerých oblastí OSPFv3

v podstate obdobná ako pri konfigurácii jednej oblasti s rozdielom, že v rámci každej oblasti je potreba na príslušných rozhraniach uviesť odpovedajúcu oblasť. Teda napríklad pre smerovač R2, ktorý je súčasťou všetkých troch oblastí, je potrebné na rozhranie *Ethernet0/0* zadať príkaz:

Router(config-if)# ospfv3 1 ipv4 area 1

pre *Ethernet0/1*:

Router(config-if)# ospfv3 1 ipv4 area 0

pre *Ethernet0/2*:

Router(config-if)#ospfv3 1 ipv4 area 3

Obdobne pre IPv6. Rovnakú konfiguráciu vykonáme na všetkých rozhraniach, ktorými sú prepojené smerovače spadajúc pod oblasti OSPF protokolu. Podľa topológie smerovač `R1` v oblasti 1 má vytvorenú OSPFv3 *address-family* len pre IPv6. V takom prípade na rozhraní *Ethernet0/0* píšeme len príkaz:

Router(config-if)# ospfv3 1 ipv6 area 1

To má za následok, že smerovať `R1` nepríjme správy s cestou ku vzdialeným IPv4 sieťam. Rozdiel oproti tomu kedy sú všetky smerovače v jednej oblasti je možné vyčítať z link-state databázy, ktorá sa zobrazí už spomínaným príkazom:

Router# show ospfv3 database

V nej vidíme, že pribudol nový typ LSA správ, a to typ 3. Taktiež. po zadaní príkazu:

Router# show ipv6 route

na smerovači R4 sa vypíše smerovacia tabuľka, kde je možné všimnúť si značky `OI`, čo značí, že daná sieť sa nachádza v inej oblasti ako daný smerovač (obrázok 4.3).

```
R4#show ipv6 route
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, la - LISP alt
       lr - LISP site-registrations, ld - LISP dyn-eid, a - Application
OI  2001:DB8:1::/64 [110/20]
     via FE80::A8BB:CCFF:FE00:220, Ethernet0/2
OI  2001:DB8:2::/64 [110/20]
     via FE80::A8BB:CCFF:FE00:220, Ethernet0/2
C   2001:DB8:5::/64 [0/0]
     via Ethernet0/2, directly connected
L   2001:DB8:5::2/128 [0/0]
     via Ethernet0/2, receive
OE2 4000:400::/64 [110/40]
     via FE80::A8BB:CCFF:FE00:220, Ethernet0/2
L   FF00::/8 [0/0]
     via Null0, receive
```

Obr. 4.3: Výpis smerovacej tabuľky smerovača R4

Ďalej medzi smerovačom `R3` a `R5` je spustený protokol `EIGRP Named Mode`. Ten sa riadí podobným rozdelením do *address-family* ako OSPFv3. Aby sme mohli túto sieť spropagovať do OSPF siete, je potrebná redistribúcia smerovacej tabuľky. Pri redistribúcii je mimo iné možné zvoliť s akou metrikou sa majú dané cesty spropagovať do danej siete. Ak tento parameter nie je vyplnený siete sa spropagujú s predvolenou metrikou, čo pre OSPFv3 je hodnota 20. Redistribúcia sa vykoná na smerovači `R3`, ktorý je pripojený do oboch podsietí. Je teda potrebné prejsť do OSPFv3 správy protokolu pod konkrétny proces a konkrétnu *address-family*. Povedzme, že chceme spropagovať jak IPv4 tak aj IPv6 siete, avšak ipv6 chceme s metrikou 40. V takom prípade je potrebné v sekcii `address-family ipv4` zadať príkaz:

Router(config-router-af)# redistribute eigrp 1

a v sekcii `address-family ipv6` príkaz:

Router(config-router-af)# redistribute eigrp 1 metric 40

Príkazom:

Router#show [ipv4|ipv6] route

sa zobrazia nové záznamy v smerovacej tabuľke. Možno ich rozoznať tak, že majú pri sebe značku `O E2` (obrázok 4.4). V Link-state databázy zúčastnených smerovačov taktiež pribudli nové typy LSA správ a to typ 4 a typ 5. (Aby zariadenia z rôznych sieťových protokolov mohli komunikovať, je potrebné aby bola redistribúcia vykonaná aj opačne, teda OSPF podsietí do EIGRP siete.)

```
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
C        10.10.10.0/24 is directly connected, Ethernet0/0
L        10.10.10.1/32 is directly connected, Ethernet0/0
C        10.10.20.0/24 is directly connected, Ethernet0/1
L        10.10.20.1/32 is directly connected, Ethernet0/1
O E2     10.10.30.0/24 [110/20] via 10.10.10.2, 00:33:16, Ethernet0/1
C        10.10.50.0/24 is directly connected, Ethernet0/2
L        10.10.50.1/32 is directly connected, Ethernet0/2
      40.0.0.0/24 is subnetted, 1 subnets
O E2     40.0.0.0 [110/20] via 10.10.10.2, 00:33:16, Ethernet0/1
      200.0.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        200.0.0.0/24 is directly connected, Loopback0
L        200.0.0.1/32 is directly connected, Loopback0
```

Obr. 4.4: Výpis smerovacej tabuľky smerovača R2

Ako posledná bude predstavená konfigurácia stub oblastí. Vysvetlenie týchto typov oblastí je poskytnuté v kapitole 2.3. Povedzme, že chceme, aby oblasť 3, medzi smerovačmi 5 a 2 bola stub, avšak len pre IPv6 siete. Je teda potrebné na oboch zariadeniach v danej oblasti v nastaveniach OSPFv3 protokolu, pod `address-family ipv6` zadať príkaz:

Router(config-router-af)# area 3 stub

Výsledok je možné overiť na smerovači `R4` príkazom:

Router#show [ipv4|ipv6] route

kde je vidieť, že toto zariadenie pozná externé IPv4, avšak žiadnu IPv6 externú sieť. Taktiež mu medzi LSA typu 3 pribudol záznam na *default* cestu `::/0`.

Skúsme pre zmenu túto oblasť nakonfigurovať ako totally stubby oblasť. Keďže sa jedná o sprísnenie pravidiel pre danú sieť, nie je potrebné rušiť nastavenie stub oblasti, ale stačí zadať príkaz:

Router(config-router-af)# area 3 stub no-summary

Dodatok `no-summary` vraví, že si neželáme aby do tejto stub oblasti boli zasielané *summary* LSA správy (teda LSA správy typu 3). Tento príkaz stačí zadať na jednom smerovači, ktorý funguje ako ABR (teda `R2`). Po výpise link-state databázy pre IPv6 vidíme, že znovu chýba LSA typu 5 a všetky LSA typu 3, ktoré boli nahradené jednou *default* cestou na ABR pre oblasť 3.

Pozor, ak zadáme príkaz:

Router(config-router-af)# no area 3 stub no-summary

oblasť neprejde z totally stubby do normálneho, ale do stub nastavenia. Môžeme to, po zadaní vyššie spomenutého príkazu, overiť výpisom bežiacej konfigurácie:

Router# show running config | section ospfv3

Kapitola 5

Návrh a implementácia OSPFv3

V tejto kapitole je popísané simulačné prostredie OMNeT++, jeho rozšírenia INET a ANSA. Následne je popísaná hierarchia tried implementujúcich OSPFv3 protokol, zdrojové súbory a ich približný obsah a konfiguračný súbor. Za týmto rozborom sú vypísané všetky problémy, ktoré som počas testovania a rozširujúcej implementácie objavil.

5.1 Simulačné prostredie OMNeT++

5.1.1 OMNeT++

OMNeT++ [15] je rozšíriteľný, modulárny C++ simulátor, založený na komponentoch. Primárne sa používa pre tvorbu diskrétnych sieťových simulačných modelov. Vo svojej grafickej podobe, OMNeT++ ako *IDE* používa upravené prostredie Eclipse.

Správanie každého simulačného modulu je implementované v C++. OMNeT++ používa vlastný jazyk *NED* pre popis týchto modulov. Popisuje aké moduly budú v danom modely použité, ich rozmiestnenie a poprepájanie. Je možné teda prepojiť viaceré jednoduchšie moduly do komplexných modulov alebo celých sietí. Moduly potom komunikujú pomocou zasielania správ. Každá simulácia je popísaná samostatným *NED* súborom, konfiguračným *omnetpp.ini* súborom a externou konfiguráciou jazykom XML v samostatných súboroch či priamo v *.ini* súbore.

5.1.2 INET

INET [2] je *open-source* modelová knižnica pre OMNeT++ simulačné prostredie. Poskytuje protokoly, agentov a iné modely pre výskum, štúdium a prácu so sieťovou komunikáciou. Poskytuje modely pre protokoly ako `TCP`, `UDP`, `IP`, `RIP`, `OSPF`, `BGP`, `ARP`, `Ethernet`, `PPP` a iné.

5.1.3 ANSAINET

Autonamed Networ Simulation and Analysis (ANSA) [10] je framework rozširujúci funkcionalitu INET (preto zdrojový kód nesie názov ANSAINET). Projekt je vyvíjaný na Fakulte informačných technológií Vysokého učení technického v Brne. Zameriava sa na vývoj a rôznych simulačných modelov kompatibilných s RFC špecifikáciami. Jeho cieľom je vytvoriť nástroje, ktoré umožnia formálny analýzu reálnych sietí a ich konfigurácií.

5.2 Stav implementácie OSPFv3

Vzhľadom na to, že táto práca nadväzuje na minulú, implementácia protokolu bola do určitej miery zvládnutá. Jeho implementácia sa nachádza vo frameworku ANSAINET, konkrétne verzii `ansainet-3.4.0`. Aktuálna implementácia OSPFv3 sa nachádza v balíku: `package ansa.routing.ospfv3`. Implementovaný model zachováva štruktúru, ktoré bola pôvodne uvedená v [14].

5.2.1 OSPFv3 moduly a triedy

`OSPFv3Routing` je zložený z dvoch jednoduchších modulov a to `OSPFv3Splitter` a `OSPFv3Process`. Modul je súčasťou `ANSA_Router` a keďže operuje na sieťovej vrstve, je priamo napojený na `ANSA_MultiNetworkLayer`. `OSPFv3Splitter` modul je zodpovedný za parsovanie konfiguračných súborov a vytváranie potrebných štruktúr a objektov. Preskúmava každý prijatý `L3` paket a na základe vstupného rozhrania ho posúva na správny `OSPFv3Process`.

RFC5340 uvádza, že na jednej linke môže existovať viacero inštancií protokolu. Cisco smerovače však umožňujú tvorbu len dvoch procesov, jeden pre *address family* IPv4, druhý pre IPv6 a len s jednou inštanciou na proces. Ako kompromis, tento model umožňuje tvorbu dvoch procesov na rozhranie, pričom každý proces môže mať viacero inštancií. Každá inštancia je reprezentovaná triedou `OSPFv3Instance` a má svoj `integer ID`. Toto ID sa objavuje v každom OSPFv3 pakete a slúži pre rozlíšenie či má byť daný paket spracovaný. Keďže pakety nenesú žiadnu informáciu o tom, ktorému procesu náležia, `OSPFv3Splitter` zduplikuje každý prijatý paket a odošle ho obom procesov. Proces potom rozhodne, či pod ním beží inštancia s takým ID, aké je v pakete zapísané. Inštancie sa ďalej delia na `OSPFv3Area` triedy, ktoré predstavujú OSPF oblasti, tak ako sú popísané v kapitole 2.3. Každé rozhranie, ktoré patrí do oblasti je implementované ako `OSPFv3Interface` trieda. Každé rozhranie môže mať viacerých susedov reprezentovaných triedou `OSPFv3Neighbor`.

Samotná implementácia je potom rozdelená do viacerých priečinkov a zdrojových súborov. Hierarchia je nasledovná:

- `interface` - priečinok obsahujúci zdrojové súbory `OSPFv3Interface.cc/h` popisujúci správanie každého rozhrania, ktoré je súčasťou OSPFv3. Tvorba Link-LSA, spracovanie prijatých paketov, šírenie LSA správ na danú linku, a iné. Ďalej priečinok obsahuje súbory, v ktorých sú definované triedy pre každý stav, ktorý môže OSPF rozhranie nadobudnúť. Názov súborov je `OSPFv3InterfaceState*.cc/h`, kde `*` značí každý stav OSPFv3 Rozhrania. Ako posledný obsahuje súbory `OSPFv3InsterfaceState.cc/h`, kde sú v rámci triedy definované metódy pre zmenu medzi týmito stavmi na danom rozhraní.

- `neighbor` - v podobnom duchu ako `interface`, priečinok, obsahujúci zdrojové súbory `OSPFv3Neighbor.cc/h` popisujúci správanie daného zariadenia ako jedného zo susedov v rámci danej OSPF siete. Jedná sa napríklad o metódy pre zistenie DR a BDR, vytvorenie susedstva a iné. `OSPFv3NeighborState*.cc/h` predstavuje zase súbory s triedami pre každý stav ktorý môže zariadenie ako sused nadobudnúť. `OSPFv3NeighborState.cc/h` slúži na opäť pre zmenu medzi týmito stavmi.

- `process` - priečinok obsahujúci súbory:

  - `OSPFv3Area.cc/h` - súbor s implementáciou procesov ktoré sa musia vykonávať na danom zariadení v rámci jednej oblasti. Popisuje teda tvorbu a zapisovanie

area-scope LSA správ, šírenie LSA správ do celej oblasti, a prototypy metód pre výpočet SPF a *next-hop* a iné.

  - `OSPFv3LSA.cc/h` - triedy pre rozšírenie LSA správ o ďalšie vlastnosti ako zvyšovanie (*installTime*) a vzdialenosť ako (*distance*) či *nextHops*.
  - `OSPFInstance.cc/h` pre tvorbu oblastí a `OSPFv3Process.cc/h/ned` pre tvorbu inštancií a informácii spoločné pre každú inštanciu na jednom zariadení, ako napríklad `router-id` či smerovacej tabuľky.
  - `OSPFv3RoutingTableEntry.cc/h` - súbory s triedou pre popis záznamu smerovacej tabuľke.

- `OSPfv3Splitter.cc/h` - najvyššia trieda v rámci hierarchie tried OSPFv3. Rozdeľuje protokol na maximálne dva procesy a to pre *address-family* IPv4 a IPv6.
- `OSPV3Common.h` a `OSPFv3Timers.h` obsahujú predovšetkým inicializáciu a definíciu makier a časovačov potrebných pre správne fungovanie protokolu.
- `OSPFv3Packet.msg` obsahuje definíciu všetkých paketov ktoré OSPFv3 pre svoje fungovanie používa. Tento súbor je však čitateľný len pre OMNeT++ a preto aby bol preložiteľný, sú z neho generované súbory `OSPFv3Packet_m.cc/h`.
- `OSPFv3Routing.ned` reprezentuje popis OSPFv3 modulu. Inými slovami, jedná sa o OMNeT++ súbor slúžiaci pre napojenie vyššie opísaného zdrojového kódu na simulované zariadenia.

5.2.2 Konfigurácia

Ako bolo povedané, pre konfiguráciu simulovaných zariadení sa používa súbor `config.xml`. Bežná konfigurácia OSPFv3 sa skladá z dvoch krokov:

1. Nastavenie OSPFv3 procesu s `address family` a `router-id` v globálnej konfigurácii;
2. Nastavenie OSPFv3 konfigurácie inštancií, oblastí a `adress family` na každom rozhraní ktoré do protokolu chceme zapojiť.

Štruktúra `config.xml` súboru sa snaží priblížiť tejto konfigurácii s rovnakým zanorením ako prebieha pri konfigurácii reálnych Cisco zariadení a preto má podobný formát. Štruktúra takéhoto zápisu je znázornená na obrázku 5.1 (XML DTD schéma, viď obrázok 5.2). Príklad konfigurácie dvoch smerovačov je potom možné vidieť v prílohe A.

```xml
<Devices>
    <Router>
        <Routing6>
            <OSPFv3>
                <Process id="">
                    <RouterID>...</RouterID>
                </Process>
            </OSPFv3>
        </Routing6>
        <Interfaces>
            <Interface name="">
                <Process id="">
                    <Instance AF="">
                        <InterfaceType>...</InterfaceType>
                        <Area>...</Area>
                    </Instance>
                </Process>
                <IPv6Address>...</IPv6Address>
            </Interface>
        </Interfaces>
    </Router>
</Devices>
```

Obr. 5.1: štruktúra zápisu konfigurácie simulovaného smerovača

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Devices [
  <!ELEMENT Devices     (Router*)>
  <!ELEMENT Router      (Routing6, Interfaces)>
  <!ELEMENT Routing6    (OSPFv3)>
  <!ELEMENT OSPFv3      (Process, (Process?))>
  <!ELEMENT Process     (RouterID|Instance)>

  <!ELEMENT Interfaces  (Interface+)>
  <!ELEMENT Interface   (Process)>
  <!ELEMENT Instance    (InterfaceType, Area, IPv6Address+)>

  <!ATTLIST Router
            id            CDATA   #REQUIRED>
  <!ATTLIST Process
            id            CDATA   #REQUIRED>
  <!ATTLIST Interface
            name          CDATA   #REQUIRED>
  <!ATTLIST Instance
            AF            CDATA   #REQUIRED>
  <!ELEMENT RouterID                (#PCDATA)>
  <!ELEMENT InterfaceType           (#PCDATA)>
  <!ELEMENT Area                    (#PCDATA)>
  <!ELEMENT IPv6Address             (#PCDATA)>
]>
```

Obr. 5.2: XML DTD schéma pre config.xml

5.2.3 Aktuálny stav

Podľa vyjadrenia v predchádzajúcej práci, na ktorú táto práca nadväzuje [14], by aktuálny stav mal zvládať všetko až na výpočet SPF a smerovanie ako také. Pri implementácii tejto poslednej chýbajúcej časti protokolu som však postupne narážal na nedostatky, niektoré viac, niektoré menej kritické. To, prečo niektoré vážnejšie chyby v implementácii testovanie neodhalilo je z časti pochopiteľné, pretože väčšina z nich sa začne vynárať vo chvíli, keď sa začne implementovať práve chýbajúci výpočet SPF. Ak má byť tento výpočet vedený tak ako je opísaný v RFC 5340 [8] a RFC 2328 [13], je potrebné pracovať s bezchybnými LSA správami. Nasleduje teda opis stavu v akom som túto prácu prebral. Jednotlivé body majú priradenú anotáciu ako **problém P*** pre jednoduchšie rozlíšenie vykonaných zmien a budúcich potrebných úprav. Problémy nie sú zoradené podľa závažnosti chýb.[1]

- **P0** - Chýba výpočet SPF a smerovanie ako také.
  - **P0.1** - smerovanie v sieti typu broadcast
  - **P0.2** - smerovanie v sieti typu P2P
  - **P0.3** - smerovanie v sieti typu NBMA
  - **P0.4** - smerovanie v sieti typu P2MP
  - **P0.5** - medzi-oblastné smerovanie
  - **P0.6** - smerovanie medzi AS
  - **P0.7** - smerovanie cez virtuálne linky
- **P1** - Router-LSA správy, nie sú tvorené v správnom formáte. Router-LSA správy sú stavebným kameňom OSPFv3 protokolu. Bez nich nie je možné vypočítať SPF. Každá router-LSA správa každého uzlu v sieti nesie zoznam liniek, ktorý je tvorený z `router-ID` pre DR danej siete a `interface ID`, ktorý vyjadruje ID rozhrania, ktorým je daný smerovač do siete s týmto DR pripojený. V aktuálnej implementácii sa do poľa `links` router-LSA správ vkladá `Router-ID` všetkých susedných zariadení na danej linke. V `link state ID` má byť pre router-LSA uložená hodnota `router-ID` smerovača, ktorý túto správu vytvára, čo v aktuálnej implementácii neplatí. `Link Count` vo výpise link-state databázy vyjadruje počet liniek, ktorými smerovač, ktorý danú router-LSA správu vytvoril, je pripojený do danej oblasti. V aktuálnej implementácii router-LSA je natvrdo počet 1, čo pochopiteľne nie je správne a tento počet je potrebné dopočítať.
- **P2** - Link-LSA správa by mala prenášať `link-local` IPv6 adresu a prefixy adries a ich dĺžku, ktoré sú na danej linke. V aktuálnej implementácii Link-LSA prenáša `link-local` tak ako má, avšak prefixy sú vždy vyplnené 4 hodnotou link local IPv6 adresy s prefixom 128.
- **P3** - Ak je správa Intra-Area-Prefix-LSA zostavovaná niekým iným ako budúcim DR, v prefixe uvedená cela adresa rozhrania, nie len jeho prefix. Taktiež je dĺžka prefixu natvrdo nastavená na 56, čo je nesprávne.

[1] Niektoré minoritné chyby či chýbajúce časti protokolu sú uvedené priamo v implementácii s anotáciou *TODO*

- **P4** - Spracovanie LSU nepracuje vždy správne.
  - **P4.1** Ak je prijatá daná LSA správa, ktorá sa už v databázy nachádza a nejedná sa o *self-originated* LSA správu, proces ju zahadzuje bez toho, aby zvážil *InstallTime* pre danú správu. Je to mimo iné tiež preto, lebo *InstallTime* je súčasťou triedy *LSATrackingInfo*, ktorá nie je zahrnutá v aktuálnej implementácii.
  - **P4.2** - Chýba spracovanie situácie, kedy LSA správa je duplikát.
  - **P4.3** - LSA sa nachádza v *request* zozname
  - **P4.4** - kópia LSA správy v databázy je novšia.
  - **P4.5** - prijatá LSA správa je staršia ako jej novšia verzia v databázy.
- **P5** - Všetky LSA správy sú implementované v zjednodušenom formáte. Neposkytujú premenné a metódy pre výpočet SPF a ani sledovanie veku správ v databázy.
- **P6** - Nie je zakomponované starnutie link-state databázy. Čas LSA správ strávený v databázy sa vôbec nesleduje, čo spôsobuje nesprávne fungovanie viacerých častí protokolu.
- **P7** - Zmena stavu rozhrania nezohľadňuje mnohé prípady, kedy by bolo potrebné prevolať pretvorenie určitých LSA správ či inkrementovať *Link State Age*, alebo rozoslať LSA na linku.
- **P8** - Chýba nastavovanie *options* pre všetky LSA správy, ktoré toto pole majú.
- **P9** - Aktualizácia router-LSA a network-LSA správ nie je úplná.
- **P10** - Tvorba inter-area-prefix-LSA nie je úplná. Je tvorená len z prvého prefixu v intra-area-prefix-LSA. Teda ak jedna LSA typu 9 obsahuje viacero prefixov, LSA typu 3 sa pre ne nevytvorí.
- **P11** - Chýba parsovanie prefixov z *config.xml* súboru. `IPv6` adresy sú rozpoznané a uložené, avšak dĺžka ich prefixu sa neukladá.
- **P12** - Pre LSR chýba výpočet
  - **P12.1** - *TTL*
  - **P12.2** -*Checksum*
- **P13** - Protokol nevie znovu preposlať stratené LSA správy typu:
  - **P13.1** - 3
  - **P13.2** - 4
  - **P13.3** - 5
  - **P13.4** - 9
- **P14** - V prípade, že oblasť bola tvorená z viac ako jednej podsiete, smerovač nevie šíriť LSA správy a program padá s chybou `segmentation fault`. To v podstate znamená, že oblasť mohla byť tvorená len jednou broadcastovou doménou.
- **P15** - Chýba spracovanie LSAck, čo mimo iné spôsobuje neustále sa opakujúcu výmenu LSU správ.

- **P16** :
  - **P16.1** - LSA správy neprenášajú metriku.
  - **P16.2** - Chýba aj parsovanie tejto hodnoty z *config.xml* súboru.
- **P17** - Šírenie a spracovanie LSA správ typu 9 nie je úplne správne. Smerovač si neuloží takéto LSA ak príde do DR.
- **P18** - Šírenie Link-LSA správ presahuje linkový rozsah a šíri sa ďalej v rámci oblasti.
- **P19** - Do štruktúry suseda sa ukladá nesprávne ID rozhrania
- **P20** - Po dosiahnutí maximálneho veku LSA správ v LSDB sa správy vymažú a žiadne ďalšie sa neposielajú.
- **P21** - Inter-Area-Prefix LSA správy sa nešíria ďalej ako na jeden skok od ABR.
- **P22** - Link State ID pre Inter-Area-Prefix LSA správy nie sú nastavované správne.
- **P23** - Protokol len čiastočne reaguje na zmenu topológie. Spojenie v susedstvách sa zruší, avšak neprebehne pretvorenie LSA správ, ktorých sa zmena dotkla.
- **P24** - V prípade P2P spojenia sa smerovače vo vytváraní susedstva zastavia na EXSTART a nezasielajú si žiadne DD pakety
- **P25** - Pre pripojené siete s koncovými stanicami (*host networks*) by sa mala generovať intra-area-prefix-LSA správa. Aktuálne sa pre túto sieť tvorí záznam v router-LSA správe daného smerovača, čo je nesprávne.
- **P26** - Nadväzovanie spojenia trvá príliš dlho, Zasielanie nadbytočných Hello paketov.
- **P27** - Kontrola na validnosť vstupného konfiguračného súboru nie je úplná a v prípade omylu môže spôsobiť chybný beh simulácie.
- **P28** - Výpočet metriky pre medzi-oblastné smerovanie neprebieha vždy správne.
- **P29** - Schopnosť protokolu reagovať na zmeny v sieti je veľmi slabá. Po odpojení linky sused zmizne z tabuľky susedov, avšak žiadne ďalšie zmeny sa neprejavia.
- **P30** - Chýba OSPFv3 *Load Balancing*. Teda v prípade že má smerovač viacero rovnako výhodných ciest do cieľovej siete, smerovač nevie zasielať dáta na všetky next hopy rovnomerne[2]

[2]Túto funkcionalitu však neumožňuje ani samotný INET, pretože záznamy v smerovacej tabuľke aktuálne umožňujú len jeden next hop na záznam.

5.3 Návrh a Implementácia

Z výpisu chýb je zrejmé, že protokol má omnoho viac nedostatkov, ako je len chýbajúci výpočet ciest do smerovacej tabuľky. Samotný návrh protokolu zostáva nezmenený a drží sa teda opisu aký bol poskytnutý v sekcii 5.2, rozšírené sú skôr len samotné moduly o veľkú dávku nových funkcií. Cieľom je teda odstrániť čo najviac zistených problémov a priblížiť sa simuláciou reálnemu protokolu OSPFv3 podľa dokumentov RFC 2328 a 5340. Prioritou je tiež zaistiť podporu pre IPv6 a IPv4 *address-family* tak, ako je definovaná v dokumente RFC 5838.

Po ukončení samotnej implementácie protokolu je zdrojový kód zmigrovaný na novšiu verziu INETu, konkrétne INET4.0. Nejedná sa o úplne triviálnu záležitosť, pretože nový INET používa iný prístup k spracovaniu a tvorbe paketov, než ako tomu bolo v starších verziách. Mimo zmeny v implementácii samotného modulu je potrebné vykonať aj niekoľko úprav v moduloch samotného INETu. Model simulovaného smerovača totižto tento nový protokol nepozná a preto je potrebné k nemu pridať ďalšie rozšírenie, ktoré pri spustení simulácie načíta všetky nové a potrebné moduly. Protokol sa taktiež musí pridať do skupiny podporovaných protokolov v triede `Protocols` a pre podporu OSPFv3 multicast adries je potrebná menšia úprava triedy `Ipv6Address`.

Kapitola 6

Testovanie

V tejto kapitole bude popísané testovanie implementovaných zmien. Testovanie prebehlo štýlom porovnávania simulovanej siete voči reálnej, zloženej z Cisco zariadení. Teda pre oba prípady bola vytvorená zhodná topológia, ktorá sa snaží pokryť čo najviac situácií na testovnie. Simulovaná topológia bola vytvorená v simulačnom prostredí OMNeT++. K vytvoreniu reálnej topológie poslúžil školský virtualizovaný hardvér `Ciscolab`.

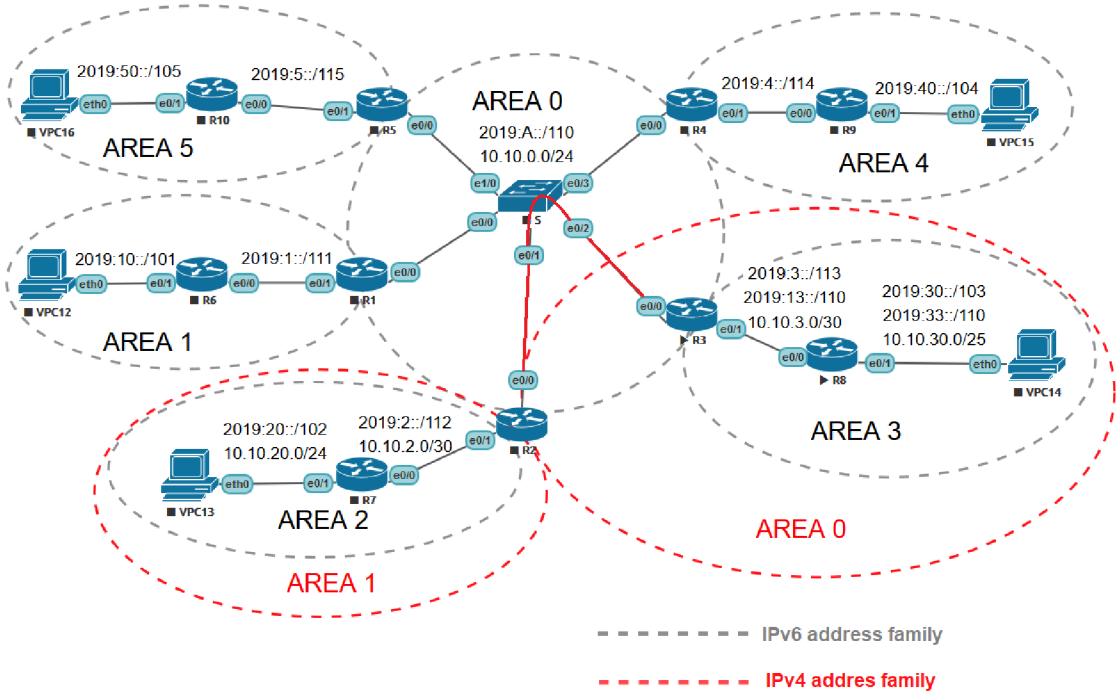
Testovanie sa skladá z troch hlavných častí. Keďže vo výmene správ v rámci nadväzovania spojenia medzi smerovačmi nebol žiadny veľký problém a testovanie prebehlo v minulej práci, na ktorú táto nadväzuje, nie je potrebné túto časť znovu testovať . Teda prvá časť testovania sa priamo zameriava na porovnanie tabuľky susedov a LSDB smerovačov v simulácii a z reálnej topológie. Druhá časť ukáže korektnosť výpočtu SPF a teda správne naplnenie smerovacích tabuliek všetkých smerovačov, vrátane medzi-oblastnej komunikácie. V tretej časti bude simulovaný výpadok linky a po určitej dobe jej opätovné nabehnutie kde sa ukáže, ako sa protokol dokáže vysporiadať s týmito udalosťami.

6.1 Testovacia topológia

Testovacia topológia je znázornená na obrázku 6.1. Pozostáva z desiatich smerovačov, jedného prepínača a piatich koncových zariadení. Na všetkých smerovačoch je nakonfigurovaný OSPFv3 protokol. Každý smerovač má nastavené `Router-ID` na 4-krát zopakovanú hodnotu čísla smerovača (teda `R1` má `Router-ID 1.1.1.1`, `R2` zase `2.2.2.2` atď). Obrázok znázorňuje konfiguráciu IPv4 aj IPv6 sietí a rozdelenie domény do niekoľkých oblastí. Medzi zariadeniami `R3` a `R8` sú na linke nakonfigurované viaceré IPv6 adresy, aby sa preukázalo správne spracovanie takejto konfigurácie protokolom. Pre ukážku flexibilnosti protokolu OSPFv3 je zámerne IPv4 doména menšia a má iné rozdelenie oblastí ako IPv6 doména. Každá z IPv6 sietí oplýva inou hodnotou prefixu, aby sa testovaním preukázalo ich funkčné spracovanie, ktoré v implementácii doposiaľ chýbalo. Smerovač `R2` má nakonfigurovanú `RouterPriority` na hodnotu 10, teda zariadenie na najvyššiu prioritu v priľahlých sieťach stať sa DR.

6.2 Nadviazanie spojenia a výmena topológie

Po nadviazaní spojenia sa smerovače pomocou DD paketov vzájomne informujú o stave svojej LSDB a následne si pomocou LSU správ tieto záznamy vymieňajú. Ako ukážkový príklad poslúži najlepšie smerovač `R2`, ktorý zasahuje do najviac častí testovacej topológie.



Obr. 6.1: Schéma testovacej topológie

Pri porovnávaní výpisov reálneho a simulovaného smerovača možno naraziť na rozdiely niektorých informácii, ktoré sú vo výpisoch obsiahnuté avšak nijak nemenia správnosť simulovaného výpisu. Sú to:

- **Interface/Interface ID** - rozdielne názvy a označenia pre rozhrania.
- **Age** - teda vek. Na reálnom zariadení sa táto hodnota každú sekundu mení a výpis sa nedá získať vždy v požadovanom čase. V simulácii sú časovače a intervaly nastavené na rovnaké hodnoty ako na reálnom zariadení a výpočty teda prebiehajú v približne rovnakom čase.
- **Seq** - sekvenčné číslo. Pri každom generovaní novej LSA správy sa zvýši hodnota sekvenčného čísla. Niektoré kroky výpočtu sú riešené heuristikou a preto v niektorých prípadoch môže simulovaný protokol generovať viac či menej LSA správ.
- **Link ID/LSID** - Link-State ID je 32-bitový identifikátor, ktorý u LSA rôzneho typu môže predstavovať inú hodnotu. V prípade Link-LSA a Network-LSA sa jedná o označenie výstupného rozhrania. A v simulácii sú rozhrania značené inak ako na reálnom zariadení. Druhý rozdiel je v prípade Intra-Area-Prefix-LSA, kde Link ID je len identifikátor, ktorý sa mení s každou novou Intra-Area-Prefix-LSA správou, ktorú smerovač vytvorí. V simulácii pre jednoduchosť táto hodnota začína nulou a s každou novou LSA typu 9 správou sa o jednu zvýši.

```
          OSPFv3 1 address-family ipv6 (router-id 2.2.2.2)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
1.1.1.1           1   FULL/DROTHER    00:00:33    3               Ethernet0/0
3.3.3.3           1   FULL/DROTHER    00:00:39    3               Ethernet0/0
4.4.4.4           1   FULL/DROTHER    00:00:33    3               Ethernet0/0
5.5.5.5           1   FULL/BDR        00:00:36    3               Ethernet0/0
7.7.7.7           1   FULL/DR         00:00:37    3               Ethernet0/1
```

Obr. 6.2: Výpis tabuľky susedov reálneho smerovača R2 pre AF IPv6

```
          OSPFv3 100 address-family IPv6 (router-id 2.2.2.2)

Neighbor ID     Pri     State           Dead Time       Interface ID    Interface
1.1.1.1         1       FULL/DROTHER    38              101             eth0
4.4.4.4         1       FULL/DROTHER    38              101             eth0
3.3.3.3         1       FULL/DROTHER    38              101             eth0
5.5.5.5         1       FULL/BDR        38              101             eth0
7.7.7.7         1       FULL/DR         38              101             eth1
```

Obr. 6.3: Výpis tabuľky susedov simulovaného smerovača R2 pre AF IPv6

6.2.1 Vytvorenie susedstva

Správnosť nadviazania susedstva v rámci IPv6 AF je doložená dvojicami obrázkov 6.2 a 6.3, kde prvý obrázok predstavuje výpis terminálu z reálneho zariadenia a druhý je výpis zo simulácie[1].

Z obrázku 6.3 možno vyčítať, že v rámci IPv6 AF sa `R2` stal DR pre oblasť 0 a BDR pre oblasť 2. Keďže žiadny ďalší smerovač nemá nakonfigurovanú hodnotu `RouterPriority`, BDR pre oblasť 0 sa stane smerovač s najvyššou hodnotou `Router-ID`, teda `R5`. Túto prednostnú prioritu má však nastavenú len na rozhraní *Ethernet0/0*, takže v sieti, do ktorej je `R2` pripojený cez *Ethernet0/1* prebieha voľba DR/BDR podľa hodnoty `Router-ID`, kde teda za DR je zvolený smerovač `R7`.

Obrázky 6.4 a 6.5 zase dokladajú určenie DR/BDR pre IPv4 AF. Na jednej strane je opäť skrz vyššiu prioritu vybraný smerovač `R2` a na druhej skrz vyššie `Router-ID` smerovač `R7`.

```
          OSPFv3 1 address-family ipv4 (router-id 2.2.2.2)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
3.3.3.3           1   FULL/BDR        00:00:30    3               Ethernet0/0
7.7.7.7           1   FULL/DR         00:00:35    3               Ethernet0/1
```

Obr. 6.4: Výpis tabuľky susedov reálneho smerovača R2 pre AF IPv4

[1] V novšej verzii OMNeT++ nie je možné získať výpis so zalamovaním riadkov a teda vytvoriť snímku obrazovky s výpisom priamo z aplikácie nie je tak úplne možné. Preto bol tento výpis prenesený do textového dokumentu a snímok spravený z neho

```
                OSPFv3 101 address-family IPv4 (router-id 2.2.2.2)

Neighbor ID     Pri     State           Dead Time       Interface ID    Interface
3.3.3.3         1       FULL/BDR        38              101             eth0
7.7.7.7         1       FULL/DR         38              101             eth1
```

Obr. 6.5: Výpis tabuľky susedov simulovaného smerovača R2 pre AF IPv4

6.2.2 Výmena topológie

Ďalej sa porovnajú LSDB smerovačov, čím sa dokáže, že zariadenie má všetky potrebné informácie pre výpočet SPF. V rámci IPv6 AF je doména na zariadení `R2` rozdelená do dvoch oblastí. Prvou dvojicou sú obrázky 6.6 a 6.7. Jedná sa o výpis LSDB z oblasti 0. Každý smerovač v rámci oblasti 0 generuje jednu Router-LSA správu, ktorá má hodnotu *Link count* u všetkých smerovačov nastavenú na jedna, pretože každý smerovač je v rámci tejto oblasti pripojený len do jednej broadcast siete a to siete 2019:a::/110, kde je jeden DR, na ktorý sa tento záznam odkazuje. Tým sa opravuje chyba z minulej implementácie, kde táto hodnota nesprávne predstavovala počet pripojených smerovačov do danej siete. Keďže každý smerovač je zároveň súčasťou ďalšej oblasti, všetky smerovače sú klasifikované ako ABR a majú nastavený bit B. Oblasť je tvorená jednou sieťou a teda `R2` ako jediné DR v tejto oblasti generuje Network-LSA.

V tabuľke je mnoho Inter-Area-Prefix-LSA záznamov z rôznych okolitých oblastí vrátane štyroch od smerovača `R3`. Za smerovačom `R3` sa však nachádzajú len dve linky, pre ktoré sa generujú dve Intra-Area-Prefix LSA správy. Na oboch linkách sú však vytvorené dve IPv6 siete s rôznymi IPv6 adresami. Všetky tieto siete z oblasti 3 sa premietnú do oblasti 0 ako samostatné Inter-Area-Prefix-LSA správy. To opravuje predchádzajúcu implementáciu, kde protokol nebol schopný generovať viacero Inter-Area-Prefix-LSA z jednej Intra-Area-Prefix-LSA správy.

Druhú dvojicu tvoria obrázky 6.8 a 6.9. Jedná sa o výpis LSDB smerovača `R2` pre oblasť 2. Pre všetky ostatné smerovače v oblasti 2, teda v tomto prípade len pre smerovač `R7` sa smerovač `R2` javí ako ABR do všetkých sietí mimo tejto oblasti. To rozširuje pôvodnú funkcionalitu, kedy smerovač nevedel informovať ostatné smerovače v rámci svojej non-backbone oblasti o iných non-backbone oblastiach.

Ako možno vidieť, generované sú tiež dve Intra-Area-Prefix-LSA správy. Obe sú generované smerovačom DR `R7`, kde jedna s `Ref-lstype 0x2002` je generovaná pre spoločnú broadcast sieť so smerovačom `R2` a druhá typu `0x2001` je pre priľahlú sieť s koncovými stanicami. Sieť s koncovými stanicami, takzvaná *host network* sa teda nijak nepremietne do generovanej Router-LSA správy, nezvyšuje hodnotu *Link count*, ale je pre ňu generovaná samostatná Intra-Area-Prefix-LSA správa a to aj v prípade, že rozhranie pripojené do takejto siete nemá nastavenú vlastnosť `PassiveInterface`. Tým je opravená ďalšia nezrovnalosť z minulej implementácie.

V rovnakom čase ako je spustený OSPFv3 pre IPv6 beží pod iným procesom OSPFv3 pre IPv4 so značne inou logickou topológiou. Ako vidieť na obrázku topológie 6.1 protokol sa dokáže vysporiadať aj so situáciou, kedy má na sieťovej vrstve cez rozhranie pripojených viacero smerovačov, ktoré však nie sú súčasťou rovnakej AF. `R2` opäť predstavuje ABR, tentokrát medzi oblasťami 0 a 1. Na dvojici výpisov 6.10 a 6.11 vidieť túto rozdielnu topológiu oblasti 0 tak, ako jej rozumie samotný smerovač `R2`. Z Inter-Area-Prefix-LSA správ možno vyčítať že sa jedná o IPv4 siete. V tejto oblasti je tiež Router-LSA správa generovaná

```
          OSPFv3 1 address-family ipv6 (router-id 2.2.2.2)

               Router Link States (Area 0)

ADV Router       Age         Seq#       Fragment ID  Link count  Bits
 1.1.1.1         28          0x80000002  0            1           B
 2.2.2.2         29          0x80000002  0            1           B
 3.3.3.3         29          0x80000002  0            1           B
 4.4.4.4         30          0x80000002  0            1           B
 5.5.5.5         30          0x80000002  0            1           B

               Net Link States (Area 0)

ADV Router       Age         Seq#       Link ID    Rtr count
 2.2.2.2         24          0x80000002  3          5

               Inter Area Prefix Link States (Area 0)

ADV Router       Age         Seq#       Prefix
 1.1.1.1         62          0x80000001  2019:1::/111
 1.1.1.1         27          0x80000001  2019:10::/101
 2.2.2.2         62          0x80000001  2019:2::/112
 2.2.2.2         27          0x80000001  2019:20::/102
 3.3.3.3         63          0x80000001  2019:3::/113
 3.3.3.3         63          0x80000001  2019:13::/110
 3.3.3.3         29          0x80000001  2019:30::/103
 3.3.3.3         29          0x80000001  2019:33::/110
 4.4.4.4         64          0x80000001  2019:4::/114
 4.4.4.4         29          0x80000001  2019:40::/104
 5.5.5.5         65          0x80000001  2019:5::/115
 5.5.5.5         30          0x80000001  2019:50::/105

               Link (Type-8) Link States (Area 0)

ADV Router       Age         Seq#       Link ID    Interface
 1.1.1.1         67          0x80000002  3          Et0/0
 2.2.2.2         67          0x80000002  3          Et0/0
 3.3.3.3         68          0x80000002  3          Et0/0
 4.4.4.4         69          0x80000002  3          Et0/0
 5.5.5.5         70          0x80000002  3          Et0/0

               Intra Area Prefix Link States (Area 0)

ADV Router       Age         Seq#       Link ID    Ref-lstype  Ref-LSID
 2.2.2.2         29          0x80000001  3072       0x2002      3
```

Obr. 6.6: Výpis LSDB reálneho smerovača R2 pre AF IPv6, oblasť 0

```
OSPFv3 1 address-family ipv6 (router-id 2.2.2.2)

Router Link States (Area 0.0.0.0)
ADV Router      Age     Seq#           Fragment ID     Link count      Bits
1.1.1.1         28      0x80000008     0               1               B
3.3.3.3         28      0x80000008     0               1               B
4.4.4.4         28      0x80000008     0               1               B
5.5.5.5         32      0x80000006     0               1               B
2.2.2.2         22      0x80000008     0               1               B

Net Link States (Area 0.0.0.0)
ADV Router      Age     Seq#           Link State ID   Rtr count
2.2.2.2         22      0x80000002     0.0.0.101       5

Inter Area Prefix Link States (Area 0.0.0.0)
ADV Router      Age     Seq#           Prefix
2.2.2.2         73      0x80000001     2019:2::/112
2.2.2.2         62      0x80000002     2019:20::/102
1.1.1.1         62      0x80000002     2019:10::/101
1.1.1.1         73      0x80000001     2019:1::/111
3.3.3.3         73      0x80000001     2019:3::/113
3.3.3.3         73      0x80000002     2019:13::/113
3.3.3.3         62      0x80000003     2019:30::/103
3.3.3.3         62      0x80000004     2019:33::/103
4.4.4.4         73      0x80000001     2019:4::/114
4.4.4.4         62      0x80000002     2019:40::/104
5.5.5.5         73      0x80000001     2019:5::/115
5.5.5.5         62      0x80000002     2019:50::/105

Link (Type-8) Link States (Area 0.0.0.0)
ADV Router      Age     Seq#           Link State ID   Interface
2.2.2.2         73      0x80000001     0.0.0.101       eth0
1.1.1.1         73      0x80000001     0.0.0.101       eth0
3.3.3.3         73      0x80000001     0.0.0.101       eth0
4.4.4.4         73      0x80000001     0.0.0.101       eth0
5.5.5.5         73      0x80000001     0.0.0.101       eth0

Intra Area Prefix Link States (Area0.0.0.0)
ADV Router      Age     Seq#           Link ID         Ref-lstype      Ref-LSID
2.2.2.2         33      0x80000002     0.0.0.2         0x2002          0.0.0.101
```

Obr. 6.7: Výpis LSDB simulovaného smerovača R2 pre AF IPv6, oblasť 0

```
                Router Link States (Area 2)

ADV Router       Age         Seq#        Fragment ID  Link count  Bits
 2.2.2.2         32          0x80000002  0            1           B
 7.7.7.7         32          0x80000002  0            1           None

                Net Link States (Area 2)

ADV Router       Age         Seq#        Link ID    Rtr count
 7.7.7.7         32          0x80000001  3          2

                Inter Area Prefix Link States (Area 2)

ADV Router       Age         Seq#        Prefix
 2.2.2.2         62          0x80000001  2019:A::/110
 2.2.2.2         27          0x80000001  2019:50::/105
 2.2.2.2         27          0x80000001  2019:5::/115
 2.2.2.2         22          0x80000001  2019:40::/104
 2.2.2.2         22          0x80000001  2019:4::/114
 2.2.2.2         22          0x80000001  2019:33::/110
 2.2.2.2         22          0x80000001  2019:30::/103
 2.2.2.2         22          0x80000001  2019:13::/110
 2.2.2.2         22          0x80000001  2019:3::/113
 2.2.2.2         22          0x80000001  2019:10::/101
 2.2.2.2         22          0x80000001  2019:1::/111

                Link (Type-8) Link States (Area 2)

ADV Router       Age         Seq#        Link ID    Interface
 2.2.2.2         67          0x80000002  4          Et0/1
 7.7.7.7         72          0x80000002  3          Et0/1

                Intra Area Prefix Link States (Area 2)

ADV Router       Age         Seq#        Link ID    Ref-lstype  Ref-LSID
 7.7.7.7         32          0x80000002  0          0x2001      0
 7.7.7.7         32          0x80000001  3072       0x2002      3
```

Obr. 6.8: Výpis LSDB reálneho smerovača R2 pre AF IPv6, oblasť 2

```
OSPFv3 1 address-family ipv6 (router-id 2.2.2.2)

Router Link States (Area 0.0.0.2)
ADV Router      Age     Seq#           Fragment ID     Link count      Bits
7.7.7.7         22      0x80000009     0               1               None
2.2.2.2         33      0x80000004     0               1               B

Net Link States (Area 0.0.0.2)
ADV Router      Age     Seq#           Link State ID   Rtr count
7.7.7.7         32      0x80000001     0.0.0.101       2

Inter Area Prefix Link States (Area 0.0.0.2)
ADV Router      Age     Seq#           Prefix
2.2.2.2         73      0x80000001     2019:a::/110
2.2.2.2         62      0x80000003     2019:10::/101
2.2.2.2         62      0x80000004     2019:1::/111
2.2.2.2         62      0x80000005     2019:3::/113
2.2.2.2         62      0x80000006     2019:13::/113
2.2.2.2         62      0x80000007     2019:30::/103
2.2.2.2         62      0x80000008     2019:33::/103
2.2.2.2         62      0x80000009     2019:4::/114
2.2.2.2         62      0x8000000a     2019:40::/104
2.2.2.2         62      0x8000000b     2019:5::/115
2.2.2.2         62      0x8000000c     2019:50::/105

Link (Type-8) Link States (Area 0.0.0.2)
ADV Router      Age     Seq#           Link State ID   Interface
2.2.2.2         73      0x80000001     0.0.0.102       eth1
7.7.7.7         73      0x80000001     0.0.0.101       eth1

Intra Area Prefix Link States (Area0.0.0.2)
ADV Router      Age     Seq#           Link ID         Ref-lstype      Ref-LSID
7.7.7.7         32      0x80000002     0.0.0.2         0x2002          0.0.0.101

7.7.7.7         32      0x80000004     0.0.0.4         0x2001          0
```

Obr. 6.9: Výpis LSDB simulovaného smerovača R2 pre AF IPv6, oblasť 2

```
OSPFv3 1 address-family ipv4 (router-id 2.2.2.2)

               Router Link States (Area 0)

ADV Router      Age         Seq#       Fragment ID  Link count  Bits
 2.2.2.2        28          0x80000002 0            1           B
 3.3.3.3        28          0x80000003 0            2           None
 8.8.8.8        34          0x80000002 0            1           None

               Net Link States (Area 0)

ADV Router      Age         Seq#       Link ID    Rtr count
 2.2.2.2        28          0x80000001 3          2
 8.8.8.8        34          0x80000001 3          2

               Inter Area Prefix Link States (Area 0)

ADV Router      Age         Seq#       Prefix
 2.2.2.2        62          0x80000001 10.10.2.0/30
 2.2.2.2        27          0x80000001 10.10.20.0/24

               Link (Type-8) Link States (Area 0)

ADV Router      Age         Seq#       Link ID    Interface
 2.2.2.2        72          0x80000001 3          Et0/0
 3.3.3.3        73          0x80000001 3          Et0/0

               Intra Area Prefix Link States (Area 0)

ADV Router      Age         Seq#       Link ID    Ref-lstype  Ref-LSID
 2.2.2.2        28          0x80000001 3072       0x2002      3
 8.8.8.8        34          0x80000002 0          0x2001      0
 8.8.8.8        34          0x80000001 3072       0x2002      3
```

Obr. 6.10: Výpis LSDB reálneho smerovača R2 pre AF IPv4, oblasť 0

smerovačom s `ADV-Router 3.3.3.3`, ktorá nesie *Link count* s hodnotou dva. To preto, lebo smerovač `R3` je v rámci oblasti medzi dvoma ďalšími smerovačmi a teda svojimi rozhraniami je pripojený do dvoch broadcast sietí, pre ktoré sa, ako vidieť, generujú dve Network-LSA správy. Správne vybudovanie LSDB pre IPv4 AF oblasť ukazujú výpisy 6.12 a 6.13.

6.3 Naplnenie smerovacích tabuliek

Po tom ako si smerovače zosynchronizujú svoje LSDB si každý smerovač lokálne vypočíta najkratšiu cestu ku každému smerovaču v rámci danej smerovacej domény. Najskôr prebieha výpočet v rámci vlastnej oblasti, následne výpočet ciest k smerovačom a sieťam v iných oblastiach. Správnosť tohto výpočtu pre smerovač `R2` dokladá výpis reálnej (vľavo) a simulovanej (vpravo) smerovacej tabuľky na obrázku 6.14. Ako vidieť, smerovač pozná cestu ku každému smerovaču v topológii. Rovnako možno vidieť správny výpočet pre smerovač `R4`. Tabuľka simulovaného zariadenia uvádza navyše aj cesty do lokálnych sietí, ktoré sú pripojené *directly*, teda priamo. Do tabuľky sa pridávajú na začiatku, pri konfigurácii zariadenia. Aktuálna verzia INETu žiaľ pre IPv6 smerovacie tabuľky neumožňuje ukladanie dodatočných informácii k vypočítanej ceste, ako je *Administrative Distance* či metrika.

```
OSPFv3 1 address-family ipv4 (router-id 2.2.2.2)

Router Link States (Area 0.0.0.0)
ADV Router      Age     Seq#           Fragment ID     Link count      Bits
3.3.3.3         28      0x80000007     0               2               None
8.8.8.8         22      0x80000009     0               1               None
2.2.2.2         22      0x8000000a     0               1               B

Net Link States (Area 0.0.0.0)
ADV Router      Age     Seq#           Link State ID   Rtr count
2.2.2.2         22      0x80000002     0.0.0.101       2
8.8.8.8         32      0x80000001     0.0.0.101       2

Inter Area Prefix Link States (Area 0.0.0.0)
ADV Router      Age     Seq#           Prefix
2.2.2.2         73      0x80000001     10.10.2.0/30
2.2.2.2         62      0x80000002     10.10.20.0/24

Link (Type-8) Link States (Area 0.0.0.0)
ADV Router      Age     Seq#           Link State ID   Interface
2.2.2.2         73      0x80000001     0.0.0.101       eth0
3.3.3.3         73      0x80000001     0.0.0.101       eth0

Intra Area Prefix Link States (Area0.0.0.0)
ADV Router      Age     Seq#           Link ID         Ref-lstype      Ref-LSID
2.2.2.2         33      0x80000002     0.0.0.2         0x2002          0.0.0.101

8.8.8.8         32      0x80000002     0.0.0.2         0x2002          0.0.0.101

8.8.8.8         32      0x80000004     0.0.0.4         0x2001          0
```

Obr. 6.11: Výpis LSDB simulovaného smerovača R2 pre AF IPv4, oblasť 0

```
                Router Link States (Area 1)

ADV Router       Age         Seq#       Fragment ID  Link count  Bits
 2.2.2.2         32          0x80000002 0            1           B
 7.7.7.7         32          0x80000002 0            1           None

                Net Link States (Area 1)

ADV Router       Age         Seq#       Link ID    Rtr count
 7.7.7.7         32          0x80000001 3          2

                Inter Area Prefix Link States (Area 1)

ADV Router       Age         Seq#       Prefix
 2.2.2.2         62          0x80000001 10.10.0.0/24
 2.2.2.2         27          0x80000001 10.10.30.0/25
 2.2.2.2         27          0x80000001 10.10.3.0/30

                Link (Type-8) Link States (Area 1)

ADV Router       Age         Seq#       Link ID    Interface
 2.2.2.2         72          0x80000001 4          Et0/1
 7.7.7.7         77          0x80000001 3          Et0/1

                Intra Area Prefix Link States (Area 1)

ADV Router       Age         Seq#       Link ID    Ref-lstype  Ref-LSID
 7.7.7.7         32          0x80000002 0          0x2001      0
 7.7.7.7         32          0x80000001 3072       0x2002      3
```

Obr. 6.12: Výpis LSDB reálneho smerovača R2 pre AF IPv4, oblasť 1

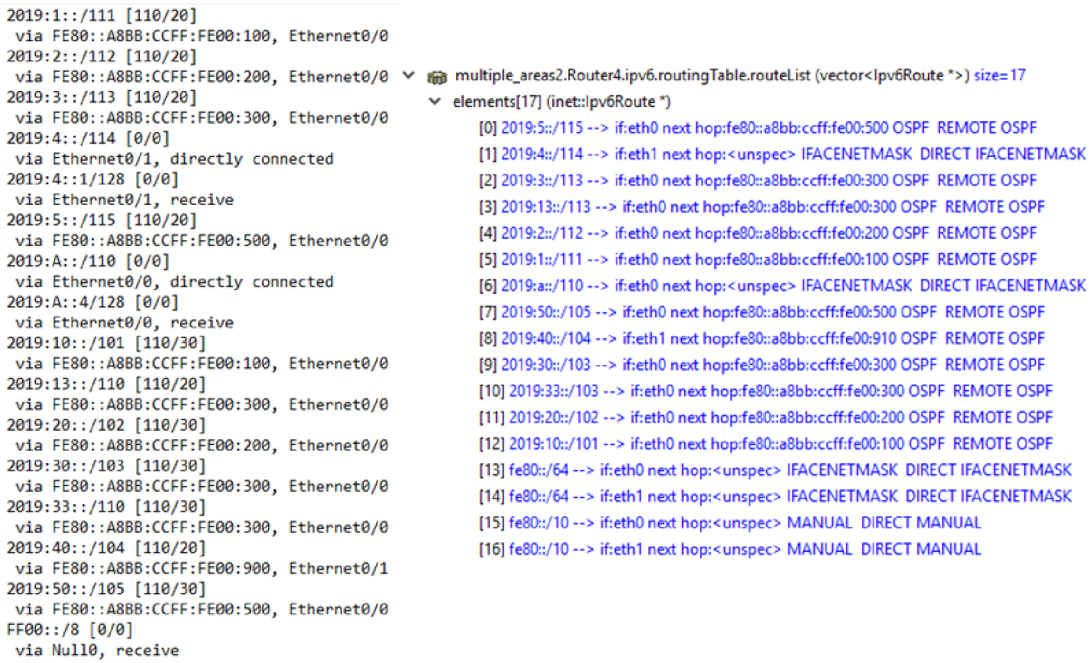
```
OSPFv3 1 address-family ipv4 (router-id 2.2.2.2)

Router Link States (Area 0.0.0.1)
ADV Router      Age     Seq#           Fragment ID     Link count      Bits
7.7.7.7         22      0x80000009     0               1               None
2.2.2.2         33      0x80000004     0               1               B

Net Link States (Area 0.0.0.1)
ADV Router      Age     Seq#           Link State ID   Rtr count
7.7.7.7         32      0x80000001     0.0.0.101       2

Inter Area Prefix Link States (Area 0.0.0.1)
ADV Router      Age     Seq#           Prefix
2.2.2.2         73      0x80000001     10.10.0.0/24
2.2.2.2         62      0x80000002     10.10.3.0/30
2.2.2.2         62      0x80000003     10.10.30.0/25

Link (Type-8) Link States (Area 0.0.0.1)
ADV Router      Age     Seq#           Link State ID   Interface
2.2.2.2         73      0x80000001     0.0.0.102       eth1
7.7.7.7         73      0x80000001     0.0.0.101       eth1

Intra Area Prefix Link States (Area0.0.0.1)
ADV Router      Age     Seq#           Link ID         Ref-lstype      Ref-LSID
7.7.7.7         32      0x80000002     0.0.0.2         0x2002          0.0.0.101

7.7.7.7         32      0x80000004     0.0.0.4         0x2001          0
```

Obr. 6.13: Výpis LSDB simulovaného smerovača R2 pre AF IPv4, oblasť 1

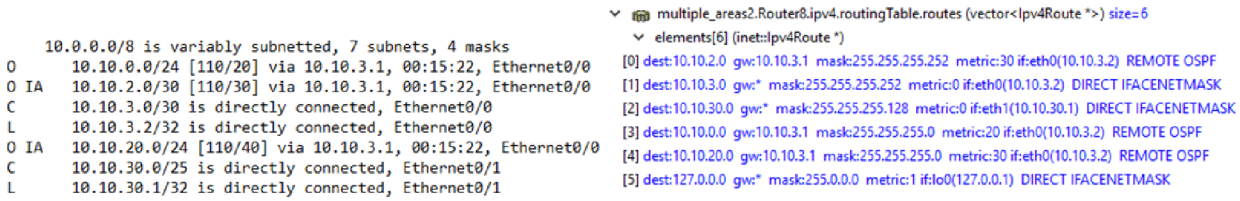
```
OI  2019:1::/111 [110/20]
     via FE80::A8BB:CCFF:FE00:100, Ethernet0/0
C   2019:2::/112 [0/0]
     via Ethernet0/1, directly connected
L   2019:2::1/128 [0/0]
     via Ethernet0/1, receive
OI  2019:3::/113 [110/20]
     via FE80::A8BB:CCFF:FE00:300, Ethernet0/0
OI  2019:4::/114 [110/20]
     via FE80::A8BB:CCFF:FE00:400, Ethernet0/0
OI  2019:5::/115 [110/20]
     via FE80::A8BB:CCFF:FE00:500, Ethernet0/0
C   2019:A::/110 [0/0]
     via Ethernet0/0, directly connected
L   2019:A::2/128 [0/0]
     via Ethernet0/0, receive
OI  2019:10::/101 [110/30]
     via FE80::A8BB:CCFF:FE00:100, Ethernet0/0
OI  2019:13::/110 [110/20]
     via FE80::A8BB:CCFF:FE00:300, Ethernet0/0
O   2019:20::/102 [110/20]
     via FE80::A8BB:CCFF:FE00:700, Ethernet0/1
OI  2019:30::/103 [110/30]
     via FE80::A8BB:CCFF:FE00:300, Ethernet0/0
OI  2019:33::/110 [110/30]
     via FE80::A8BB:CCFF:FE00:300, Ethernet0/0
OI  2019:40::/104 [110/30]
     via FE80::A8BB:CCFF:FE00:400, Ethernet0/0
OI  2019:50::/105 [110/30]
     via FE80::A8BB:CCFF:FE00:500, Ethernet0/0
L   FF00::/8 [0/0]
     via Null0, receive
```

```
multiple_areas2.Router2.ipv6.routingTable.routeList (vector<Ipv6Route *>) size=17
  elements[17] (inet::Ipv6Route *)
    [0] 2019:5::/115 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:500 OSPF  REMOTE OSPF
    [1] 2019:4::/114 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:400 OSPF  REMOTE OSPF
    [2] 2019:3::/113 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:300 OSPF  REMOTE OSPF
    [3] 2019:13::/113 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:300 OSPF  REMOTE OSPF
    [4] 2019:2::/112 --> if:eth1 next hop:<unspec> IFACENETMASK  DIRECT IFACENETMASK
    [5] 2019:1::/111 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:100 OSPF  REMOTE OSPF
    [6] 2019:a::/110 --> if:eth0 next hop:<unspec> IFACENETMASK  DIRECT IFACENETMASK
    [7] 2019:50::/105 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:500 OSPF  REMOTE OSPF
    [8] 2019:40::/104 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:400 OSPF  REMOTE OSPF
    [9] 2019:30::/103 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:300 OSPF  REMOTE OSPF
    [10] 2019:33::/103 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:300 OSPF  REMOTE OSPF
    [11] 2019:20::/102 --> if:eth1 next hop:fe80::a8bb:ccff:fe00:710 OSPF  REMOTE OSPF
    [12] 2019:10::/101 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:100 OSPF  REMOTE OSPF
    [13] fe80::/64 --> if:eth0 next hop:<unspec> IFACENETMASK  DIRECT IFACENETMASK
    [14] fe80::/64 --> if:eth1 next hop:<unspec> IFACENETMASK  DIRECT IFACENETMASK
    [15] fe80::/10 --> if:eth0 next hop:<unspec> MANUAL  DIRECT MANUAL
    [16] fe80::/10 --> if:eth1 next hop:<unspec> MANUAL  DIRECT MANUAL
```

Obr. 6.14: Výpis smerovacej tabuľky reálneho (vľavo) a simulovaného (vpravo) smerovača R2 pre AF IPv6



Obr. 6.15: Výpis smerovacej tabuľky reálneho (vľavo) a simulovaného (vpravo) smerovača R4 pre AF IPv6



Obr. 6.16: Výpis smerovacej tabuľky reálneho (vľavo) a simulovaného (vpravo) smerovača R8 pre AF IPv4

`R2` nie je jediný smerovač, na ktorom výpočet prebieha. Pre ukážku správneho výpočtu je poskytnuté porovnanie reálnej a simulovanej tabuľky ďalšieho zariadenia, smerovača **R4** na obrázku 6.15.

Popri výpočte SPF pre IPv6 AF na všetkých zúčastnených smerovačoch prebieha podobný výpočet na zariadeniach, ktoré sú dodatočne súčasťou IPv4 AF. Zariadení v tejto AF je menej a teda je menej aj cieľových sietí s IPv4 adresou. Pre túto AF je poskytnutý náhľad do vypočítanej smerovacej tabuľky zariadenia `R8`, na obrázku 6.16.

6.4 Výpadok a obnovenie linky

Ďalej bude simulovaná situácia, kedy v čase `t=90` sekúnd behu simulácie vypadne linka medzi zariadeniami `R3` a `R8` a následne sa v čase `t=300` sekúnd linka znovu obnoví. Sledovaná bude reakcia smerovačov v blízkom aj vzdialenom epicentre udalosti.

```xml
<scenario>
    <at t="90">
        <disconnect src-module="Router3" src-gate="ethg$o[1]" />
        <disconnect src-module="Router8" src-gate="ethg$o[0]" />
    </at>
    <at t="300">
        <connect src-module="Router3" src-gate="ethg[1]"
                 dest-module="Router8" dest-gate="ethg[0]"
                 channel-type="inet.common.misc.ThruputMeteringChannel">
            <param name="delay" value="0.1us" />
            <param name="datarate" value="100Mbps" />
            <param name="thruputDisplayFormat" value='"#N"' />
        </connect>
    </at>
</scenario>
```

Obr. 6.17: Scenár pre výpadok a obnovenie linky medzi smerovačmi R3 a R8[2].

```
                OSPFv3 100 address-family IPv6 (router-id 3.3.3.3)

Neighbor ID     Pri     State           Dead Time       Interface ID    Interface
1.1.1.1         1       2WAY/DROTHER    36              101             eth0
2.2.2.2         10      FULL/DR         36              101             eth0
4.4.4.4         1       2WAY/DROTHER    36              101             eth0
5.5.5.5         1       FULL/BDR        36              101             eth0
8.8.8.8         1       FULL/DR         36              101             eth1

                OSPFv3 101 address-family IPv4 (router-id 3.3.3.3)

Neighbor ID     Pri     State           Dead Time       Interface ID    Interface
2.2.2.2         10      FULL/DR         36              101             eth0
8.8.8.8         1       FULL/DR         36              101             eth1
```

Obr. 6.18: Tabuľka susedov pred výpadkom na zariadení R3 pre IPv6 a IPv4 AF

Možnosť vytvorenia takéhoto scenára je ďalšou z kľúčových vlastností prostredia OMNeT++. Scenár je možné vytvoriť pomocou modulu `ScenarioManager`. Ten využíva súbor s xml konfiguráciou, ktorá je znázornená na obrázku 6.17

6.4.1 Výpadok linky

Vychádzame zo stavu topológie, aká bola doposiaľ predstavená. Smerovače si udržujú susedstvo, pretože pravidelne dostávajú Hello pakety od svojich susedov. Tabuľka susedov pre smerovač `R3` je zobrazená na obrázku 6.18. Na obrázku je vidieť, že smerovač si drží rozdielne tabuľky pre IPv4 a IPv6 AF. Keď v čase `t=90` dôjde k výpadku linky, smerovače o danej udalosti nevedia až kým nevyprší `DeadTimer`, ktorý má predvolenú hodnotu 4-krát hodnotu `HelloInterval`, teda 40 sekúnd.

V čase približne 130 sekúnd dôjde na danom rozhraní k udalosti, kedy vyprší čakanie na Hello paket a `R3` ako aj `R8` zrušia svoje susedstvo a odstránia záznam z tabuľky susedov, ako je ukázané na obrázku 6.19.

To spustí sériu udalostí na dotknutých zariadeniach, ktoré musia spropagovať túto zmenu do celej siete. Predošlá verzia implementácie sa k takejto udalosti zachovala len tak, že daného suseda vymazala z tabuľky susedov. To samozrejme nestačí a je potrebné zneplatniť kľúčové LSA správy, ktoré pochádzajú od smerovača, s ktorým spojenie prerušené. V tomto scenári ostal od zvyšku siete odstrihnutý `R8`. `R3` teraz musí zneplatniť všetky Intra-Area-Prefix-LSA správy, ktoré od `R8` obdržal. Keďže `R3` je taktie ABR, musí v ostatných oblastiach, v ktorých je pripojený, zneplatniť všetky Inter-Area-Prefix-LSA správy,

```
                OSPFv3 100 address-family IPv6 (router-id 3.3.3.3)

Neighbor ID     Pri     State           Dead Time       Interface ID    Interface
1.1.1.1         1       2WAY/DROTHER    36              101             eth0
2.2.2.2         10      FULL/DR         36              101             eth0
4.4.4.4         1       2WAY/DROTHER    36              101             eth0
5.5.5.5         1       FULL/BDR        36              101             eth0

               OSPFv3 101 address-family IPv4 (router-id 3.3.3.3)

Neighbor ID     Pri     State           Dead Time       Interface ID    Interface
2.2.2.2         10      FULL/DR         36              101             eth0
```

Obr. 6.19: Tabuľka susedov po výpadku linky na zariadení R3 pre IPv6 a IPv4 AF

```
multiple_areas2.Router10.ipv6.routingTable.routeList (vector<Ipv6Route *>) size=17
elements[17] (inet::Ipv6Route *)
[0] 2019:5::/115 --> if:eth0 next hop:<unspec> IFACENETMASK DIRECT IFACENETMASK
[1] 2019:4::/114 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[2] 2019:3::/113 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[3] 2019:13::/113 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[4] 2019:2::/112 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[5] 2019:1::/111 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[6] 2019:a::/110 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[7] 2019:50::/105 --> if:eth1 next hop:<unspec> IFACENETMASK DIRECT IFACENETMASK
[8] 2019:40::/104 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[9] 2019:30::/103 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[10] 2019:33::/103 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[11] 2019:20::/102 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[12] 2019:10::/101 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[13] fe80::/64 --> if:eth0 next hop:<unspec> IFACENETMASK DIRECT IFACENETMASK
[14] fe80::/64 --> if:eth1 next hop:<unspec> IFACENETMASK DIRECT IFACENETMASK
[15] fe80::/10 --> if:eth0 next hop:<unspec> MANUAL DIRECT MANUAL
[16] fe80::/10 --> if:eth1 next hop:<unspec> MANUAL DIRECT MANUAL
```

```
multiple_areas2.Router10.ipv6.routingTable.routeList (vector<Ipv6Route *>) size=15
elements[15] (inet::Ipv6Route *)
[0] 2019:5::/115 --> if:eth0 next hop:<unspec> IFACENETMASK DIRECT IFACENETMASK
[1] 2019:4::/114 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[2] 2019:3::/113 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[3] 2019:13::/113 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[4] 2019:2::/112 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[5] 2019:1::/111 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[6] 2019:a::/110 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[7] 2019:50::/105 --> if:eth1 next hop:<unspec> IFACENETMASK DIRECT IFACENETMASK
[8] 2019:40::/104 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[9] 2019:20::/102 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[10] 2019:10::/101 --> if:eth0 next hop:fe80::a8bb:ccff:fe00:510 OSPF REMOTE OSPF
[11] fe80::/64 --> if:eth0 next hop:<unspec> IFACENETMASK DIRECT IFACENETMASK
[12] fe80::/64 --> if:eth1 next hop:<unspec> IFACENETMASK DIRECT IFACENETMASK
[13] fe80::/10 --> if:eth0 next hop:<unspec> MANUAL DIRECT MANUAL
[14] fe80::/10 --> if:eth1 next hop:<unspec> MANUAL DIRECT MANUAL
```

Obr. 6.20: Výpis smerovacej tabuľky simulovaného smerovača R10 pre AF IPv6 pred (vľavo) a po (vpravo) výpadku linky medzi R3 a R8

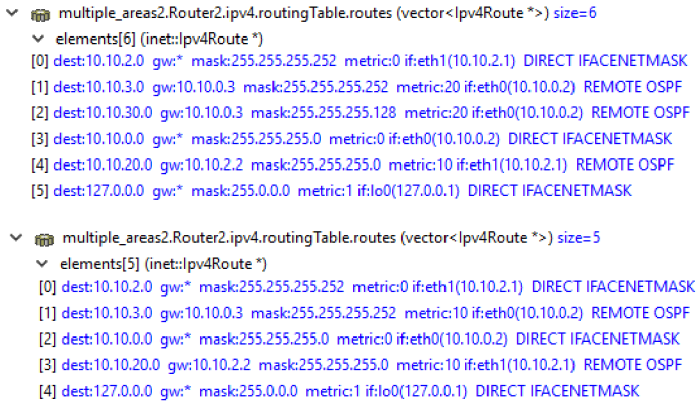
ktoré sám vytvoril a informovali zvyšok topológie o sieťach, s ktorými práve stratil spojenie. Zneplatnenie LSA správy prebieha tak, že sa pre ňu nastaví LS Age na hodnotu MAX_-AGE, čo pre OSPFv3 znamená 3600 a rozošle sa do zvyšku oblasti. Smerovače príjmu toto LSU, aktualizujú si svoju LSDB a v momente kontroly veku jednotlivých LSA odstránia túto LSA správu zo svojej LSDB čo spustí nový výpočet SPF.

Ak sa zmena udeje v non-backbone oblasti, ako sa naozaj udialo, je potrebné túto zmenu spropagovať cez oblasť 0 aj do ďalších non-backbone oblastí, ktoré majú vlastné ABR smerovače s generovanými Inter-Area-Prefix-LSA správami pre tieto už nedosiahnuteľné siete. To že sa zmena skutočne takto spropagovala dokladá výpis smerovacej tabuľky na obrázku 6.20 zariadenia `R10`, ktorý je na opačnej strane topológie. Je vidieť, že v jeho smerovacej tabuľke sa nenachádzajú IPv6 siete, ktoré sú za `R8`.

Výpadok linky samozrejme ovplyvnil aj smerovače v IPv4 AF. Ako ukazuje porovnanie smerovacej tabuľky pred a po výpadku na obrázku 6.21 pre smerovač `R2`.

6.4.2 Obnovenie linky

Teraz bude otestované, ako sa protokol zachová po obnove vypadnutého spojenia. Keď čas simulácie dosiahne hodnotu `t=300`, linka sa medzi `R3` a `R8` obnoví. Smerovače si po obnovenej

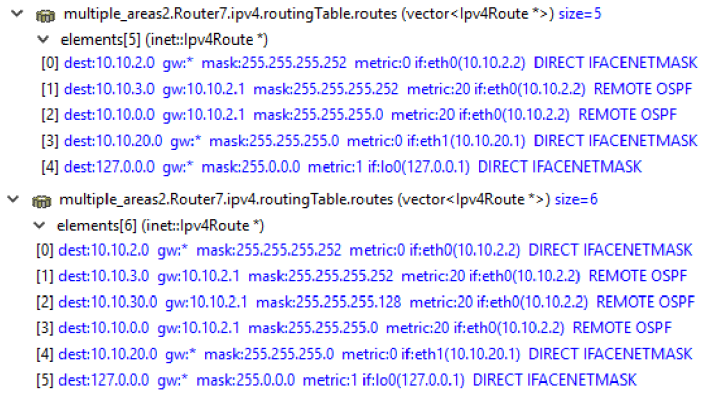


Obr. 6.21: Výpis smerovacej tabuľky simulovaného smerovača R2 pre AF IPv4 pred (hore) a po (dole) výpadku linky medzi R3 a R8

Obr. 6.22: Výpis smerovacej tabuľky simulovaného smerovača R6 pre AF IPv6 pred (vľavo) a po (vpravo) obnovení vypadnutej linky medzi R3 a R8

linke začnú vzájomne zasielať Hello pakety, čím začne proces nového nadväzovania susedstva ako pri štarte simulácie. Po vytvorení spojenia si vymenia LSDB a prepíšu si vytvorené Intra-Area-Prefix-LSA správy, ktoré boli po výpadku pre tieto rozhrania bez pripojenia vytvorené. Táto zmena sa v IPv6 AF taktiež prejaví znovu-vytvorením Inter-Area-Prefix-LSA správy pre siete za smerovačom `R8`. Opäť je túto zmenu potrebné spropagovať do zvyšku topológie. Príkladom úspešného spropagovania do celej topológie dokladá výpis smerovacej tabuľky smerovača `R6` pred a po obnovení vypadnutej linky na obrázku 6.22. Na výpise je vidieť, že do smerovacej tabuľky pribudli siete, ktoré sú za smerovačom `R8`

Rovnako tak sa spojenie nadviaže aj v rámci IPv4 AF. To dokazuje výpis smerovacej tabuľky pre `R7` na obrázku 6.23, kde do smerovacej tabuľky opäť pribudla sieť s koncovými stanicami.



Obr. 6.23: Výpis smerovacej tabuľky simulovaného smerovača R7 pre AF IPv4 pred (hore) a po (dole) obnovení vypadnutej linky medzi R3 a R8

6.5 Zhodnotenie

Testovanie preukázalo, že pridaná funkcionalita, ako je smerovanie, reakcia na zmeny a celková oprava majoritných problémov opäť posunuli model OSPFv3 protokolu bližšie k reálnemu správaniu. Implementované modely sú v súlade so štandardmi tohto protokolu a jeho referenčným správaním nad reálnou topológiou do takej miery, ako to len prostredie simulácie aktuálne umožňuje. Riešenie teda možno považovať za úspešne implementované.

Kapitola 7

Záver

Cieľom mojej práce bolo oboznámiť sa s problematikou OSPFv3 protokolu, preskúmať aktuálny stav implementovaného riešenia a posunúť ho bližšie k reálnemu správaniu tohto protokolu tak, ako je spísaný v RFC 2328, 5340 a s dodatkom *address-family* z RFC 5838. Napriek detailnému popisu existujú prvky protokolu, ktoré sa nedajú zistiť inak, ako ich odpozorovať zo správania nad reálnou topológiou. A presne toho bolo aj počas práce využité. Cenným zdrojom informácii bola tiež implementácia staršej verzie OSPFv2, ktorá je už dlhšiu dobu súčasťou INETu. Mnohé časti logiky protokolu, ktoré sa medzi verziami nelíšia boli prebrané a patrične upravené.

Z pôvodne malého množstva predpokladaných úprav sa ukázala potreba značného prepracovania viacerých častí protokolu. Mimo plánovaného SPF výpočtu, značné úpravy vyžadovala tvorba a zasielanie rôznych LSA správ, starnutie správ v LSDB a celkový koncept *address-family*. Veľké množstvo práce vyžiadalo zaistenie, že protokol správne reaguje na zmeny v topológii, predovšetkým v prípade medzi-oblastnej komunikácie. Oblasť, kam výsledná implementácia nesiaha je podpora iných typov sietí ako broadcast, komunikácia cez virtuálne linky a podpora komunikácie s inými autonómnymi systémami. Tieto veci boli zhodnotené ako nad rámec tejto práce. Kľúčovým prínosom je aj toľko potrebné zhodnotenie aktuálneho stavu implementácie. To totižto môže značne zjednodušiť pochopenie problematiky pri ďalšom postupe.

Po uzavretí implementácie nasledovala migrácia celého modelu OSPFv3 do novšej verzie INETu, INET4.0. Opäť sa nejednalo o triviálnu záležitosť, pretože INET4.0 oproti starším verziám využíva iné API pre prácu s paketmi a pochopiteľne nemá v sebe žiadny z ANSA modelov, na ktoré bol OSPFv3 v staršej verzii viazaný.

Pri spätnom zhodnotení vykonanej práce môžem prehlásiť, že môj prínos k tejto práci považujem za úspech. Dokladá to aj fakt, že som sa so svojou prácou zúčastnil na študentskej konferencii Excel@FIT, kde bola práca prijatá a prezentovaná formou plagátu. Ako bolo spomenuté, ďalší postup práce sa môže zamerať na podporu ostatných typov sietí a následne na komunikáciu medzi autonómnymi systémami. Popri tom má protokol množstvo minoritných nedostatkov, ktoré taktiež vyžadujú pozornosť. Avšak v stave akom je protokol teraz, ja, spolu s mojim vedúcim považujeme implementáciu dostatočnú na to, aby sme ju predstavili na oficiálnom OMNeT++ stretnutí a zažiadali o trvalé pripojenie do INETu.

Literatúra

[1] *Cisco Community: Comparing OSPFv3 OSPFv2 Routing Protocol*. [Online; navštívené 20.12.2018].
URL https://community.cisco.com/t5/networking-documents/comparing-ospfv3-amp-ospfv2-routing-protocol/ta-p/3109370

[2] *INET Framework*. [Online; navštívené 15.11.2018].
URL https://inet.omnetpp.org/

[3] *LINFO: Link State Routing Definition*. [Online; navštívené 12.12.2018].
URL http://www.linfo.org/link_state_routing.html

[4] *OmniSecu: Introduction to Distance Vector Routing Protocols*. [Online; navštívené 12.12.2018].
URL http://www.omnisecu.com/cisco-certified-network-associate-ccna/introduction-to-distance-vector-routing-protocols.php

[5] *OSPFv3 Address Families*. [Online; navštívené 8.1.2019].
URL https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-3s/iro-xe-3s-book/ip6-route-ospfv3-add-fam-xe.html

[6] Bass, J.: OSPFv3 Address Families: How They're Used and Why. 2015, [Online; navštívené 8.1.2019].
URL https://www.globalknowledge.com/blog/2015/01/15/ospfv3-address-families-how-theyre-used-and-why/

[7] Cisco Systems, I.: *IP Routing: OSPF Configuration Guide, Cisco IOS Release 15MT*. [Online; navštívené 15.11.2018].
URL https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/15-mt/iro-15-mt-book.html

[8] Coltun, R.; Ferguson, D.; Moy, J.; aj.: RFC 5340, OSPF for IPv6. *IETF. July*, ročník 24, 2008.

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

[10] Faculty of Information Technology, B. U. o. T.: *Project ANSA*. [Online; navštívené 15.11.2018].
URL https://nes.fit.vutbr.cz/ansa/

[11] KOLÁŘ, J.: *Teoretická informatika. 2. vyd.)*. Praha: Česká informatická společnost, 2004, ISBN 80-900853-8-5.

[12] Lindem, A.; Mirtorabi, S.; Barnes, M.; aj.: Support of address families in OSPFv3. 2010.

[13] Moy, J.: RFC 2328. *OSPF version 2*, 1998.

[14] RUPRICH, M.: *Modeling of OSPFv3 Link-State Routing Protocol*. Diplomová práce, Brno University of Technology, Faculty of Information Technology, 2017.

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

[16] Veselý, V.; Palúch, P.: *Open Shortest Path First (ROUTE Module 3)*. [Online; navštívené 15.11.2018].
URL https://netacad.fit.vutbr.cz/ccnp/route/ROUTE_M3_ENG_v7.pdf

[17] Zbyněk Křivka, T. M.: Grafové Algoritmy. 2017, [Online; navštívené 23.12.2018].
URL https://www.fit.vutbr.cz/study/courses/GAL/public/gal-text.pdft

Prílohy

Príloha A

Príklad konfiguračného súboru

```
<Devices>
    <Router id="Router1">
        <Routing6>
            <OSPFv3>
                <Process id="100">
                    <RouterID>10.10.10.1</RouterID>
                </Process>
            </OSPFv3>
        </Routing6>

        <Interfaces>
            <Interface name="eth0">
                <Process id="100">
                    <Instance AF="IPv6">
                        <InterfaceType>Broadcast</InterfaceType>
                        <Area>0.0.0.0</Area>
                    </Instance>
                </Process>
                <IPv6Address>fe80::a8bb:ccff:fe00:100/64</IPv6Address>
                <IPv6Address>2001:db8:a::1/64</IPv6Address>
            </Interface>
        </Interfaces>
    </Router>

    <Router id="Router2">
        <Routing6>
            <OSPFv3>
                <Process id="100">
                    <RouterID>10.10.10.2</RouterID>
                </Process>
            </OSPFv3>
        </Routing6>

        <Interfaces>
            <Interface name="eth0">
                <Process id="100">
                    <Instance AF="IPv6">
                        <InterfaceType>Broadcast</InterfaceType>
                        <Area>0.0.0.0</Area>
                    </Instance>
                </Process>
                <IPv6Address>fe80::a8bb:ccff:fe00:200/64</IPv6Address>
                <IPv6Address>2001:db8:a::2/64</IPv6Address>
            </Interface>
        </Interfaces>
    </Router>
</Devices>
```

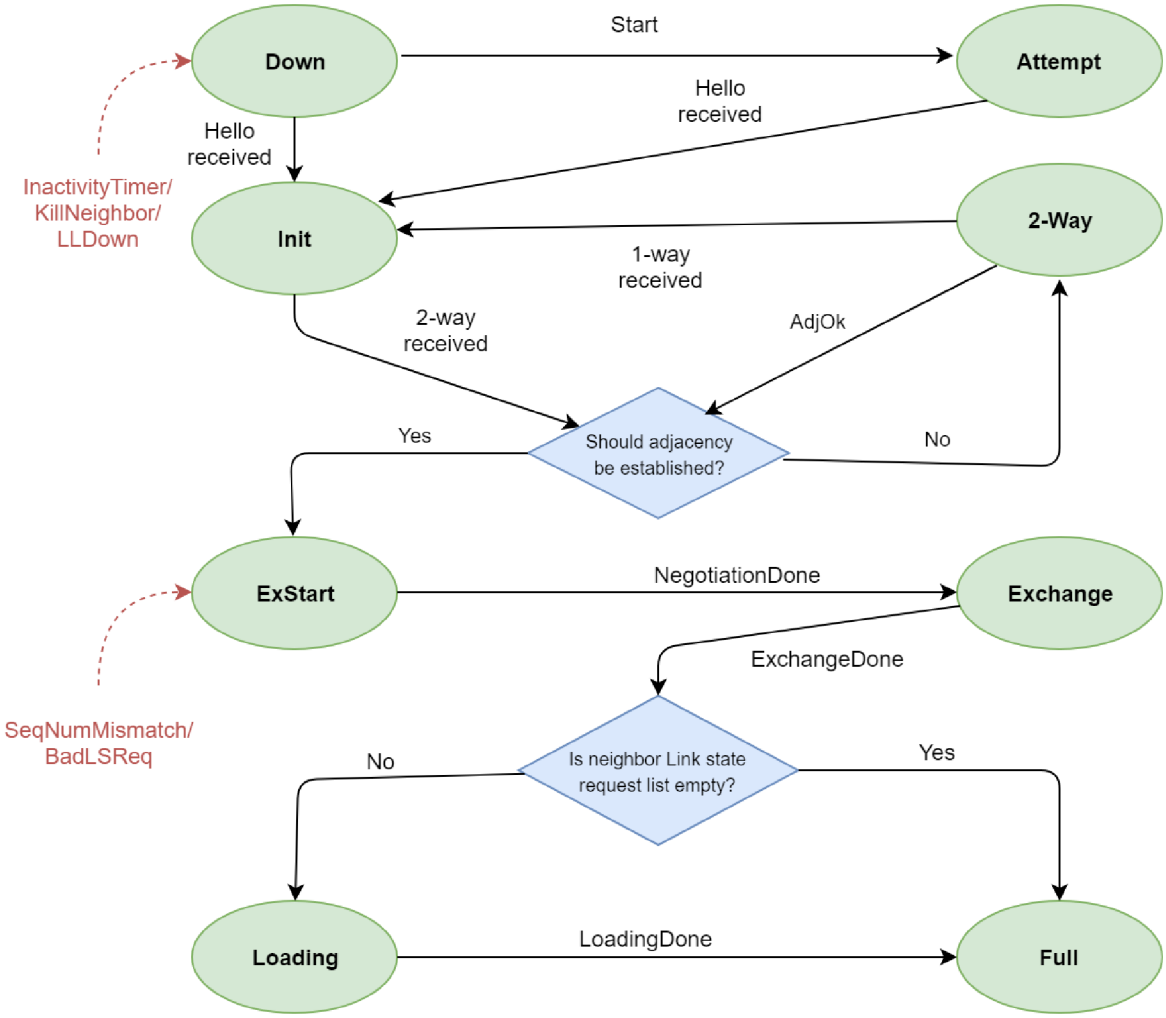
Príloha B

Obsah priloženého DVD

/xgalbi01.pdf	Elektronická podoba diplomovej práce v PDF
/readme.txt	Manuál popisujúci postup spustenia projektu
/install/*	Súbory potrebné pre inštaláciu OMNeT++
/tex/*	Zdrojový text diplomovej práce v .tex formáte.
/src/*	Zdrojové súbory projektu.

Príloha C

Konečný automat suseda



Stavy suseda reflektujú progres v nadväzovaní spojenia so susedom na danom rozrhaní. Stavy Down, Attempt, 2-Way a ExStart nastávajú v závislosti od prijatých a zaslaných Hello paketov. Ostatné stavy reprezentujú výmenu LSDB.

Down - počiatočný stav suseda. Znamená, že od tohto suseda neboli prijaté žiadne nedávne informácie.

Attempt - tento stav je valídny len na NBMA sieťach. Rozhranie zasiela susedovi Hello pakety za účelom skontaktovania sa s ním

Init - v tomto stave sa sused ocitne, keď od neho bol v nedávnej dobe zaznamenaný Hello paket avšak obojsmerná komunikácia ešte stále nadviazaná nebola.

2-Way - v tomto stave je vďaka Hello protokolu už komunikácia medzi susedmi obojsmerná. Voľba DR/BDR prebiha v tomto a vyšších stavoch.

ExStart - toto je prvý krok v nadväzovaní spojenia susedov. Cieľom tohto stavu je rozhodnúť, ktorý so susedov bude *master*, a ktorý *slave*. Od tohto stavu hovoríme o konverzácii susedov ako o spoejní

ExChange - smerovače si vymieňajú LSDB. Bližší popis v kapitole 2, sekcii 2.10.

Loading - LSR pakety sú zaslané susedovi, pretože boli vyvtorené aktuálnejšie LSA správy, ktoré ešte neboli prijaté.

Full - v tomto stave je spojenie medzi susedmi plne nadviazané. Tieto spojenia sa teraz objavia v router-LSA a network-LSA správach.

Ďalej následuje popis jednotlivých prechodov. Čiarkované šípky indikujú udalosti, ktoré sa môžu stať vo viac ako jednom stave, avšak výsledkom bude stav, do ktorého táto šípka ukazuje.

Start - Tento prechod má význam len pre NBMA siete. Vraví, že susedovi by mal byť zaslaný Hello packet.

Hello received - prijatie Hello paketu

2-way received - smerovač si je vodomí svojho suseda, pretože našiel samého seba v prijatom Hello pakete. Môže byť nadviazaná obojsmerná komunikácia.

1-way received - smerovač obdržal Hello paket, avšak seba v ňom už viac nevidí.

AdjOK - rozhodnutie o tom či má byť so susedom nadvizané spojenie.

NegotiationDone - úvodná výmena informácii prebehla, môže sa postúpiť k výmene LSDB.

ExchangeDone - výmena LSDB informácií prebehla úspešne.

LoadingDone - všetky staré LSA správy boli aktulaizované. Oba smerovače majú teraz rovnaké LSA správy.

InactivityTimer - po dobu *DeadInterval* sekúnd nebol prijatý žiaden Hello paket. Dosihanutie tohto časovaču zapríčiní prechod z akéhokoľvek stavu do *Down* stavu.

KillNeighbor - komunikácia so susedom nie je možná.To zapríčiní prechod z akéhokoľvek stavu do *Down* stavu.

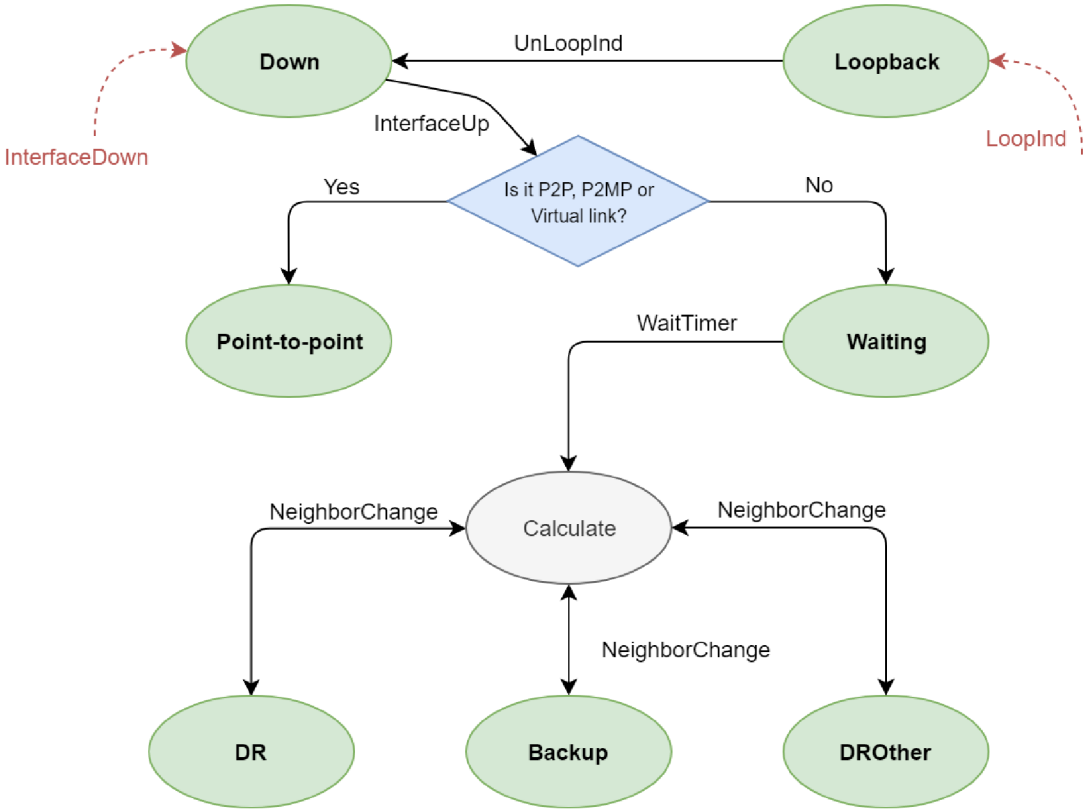
LLDwon - protokol z nižšej vrstvy indikuje, že sused je nedostiahnuteľný. To zapríčiní prechod z akéhokoľvek stavu do *Down* stavu.

SeqNumMismatch - nesprávne DD sekvenčné číslo či nezhoda v *options* v prijatom DD pakete. To zapríčiní prechod z *Exchange* a vyššieho stavu do stavu *ExStart*.

BadLSReq - Smerovač obdržal LSR pre LSA správu, ktorú nemá v LSDB. To zapríčiní prechod z *Exchange* a vyššieho stavu do stavu *ExStart*.

Príloha D

Konečný automat rozhrania



Štruktúra rozhrania obsahuje mnohé informácie pre valídny beh OSPF protokolu. Rôzne stavy rozhrania odpovedajú rôznej úrovni funkcionality rozhrania. Stavy, ktoré môže rozhranie nadobudnúť sú:

Down - počiatočný stav rozhrania. Rozhraním neprechádzajú žiadne pakety.

Loopback - rozhranie je nakonfigurované ako *Loopback*.

Point-to-point - rozhranie je pripojené do fyzického point-to-point rozhrania, alebo sa jedná o virtuálnu linku.

Waiting - v tomto stave smerovač monitoruje prijaté Hello pakety a snaží sa určit identitu DR/BDR pre danú sieť. Smerovač nemôže zvoliť konkrétny DR/BDR pokým neopustí tento stav.

DR - tento stav indikuje, že smerovač bol vybraný ako DR pre danú sieť.

Backup - tento stav indikuje, že smerovač bol vybraný ako BDR pre danú sieť.

DRother - rozhranie je na sieťovom segmente, kde DR a BDR sú zvolení.

Tak ako pri stavovom automate suseda, tak aj pri rozhraní sú rôzne prechody, ktoré potrebujú bližší popis. Čiarkované šípky opäť indikujú udalosti, ktoré sa môžu stať vo viac ako jednom stave, avšak výsledkom bude stav, do ktorého táto šípka ukazuje.

InterfaceUp - rozrhanie je funkčné a pripravené na použitie. Ďalší stav závisí od konfigurácie siete, do ktorej je rozhranie pripojené.

LoopInd - rozhranie sa zmenilo na *loopback* rozhranie. To spôsobí prechod z akéhokoľvek stavu do *Loopback* stavu.

UnLoopInd - rozhranie už nie je *loopback* rozhraním

WaitTimer - časovač *Wait timer* uplynul. Následuje voľba DR/BDR.

BackupSeen - bol obdržaný Hello paket, ktorý indikuje (ne)existenciu BDR. Táto udalosť oznameuje smerovaču, že nastáva prechod zo stavu *Waiting*.

NeighborChange - v sieti nastala zmena a DR/BDR musia byť určený na novo.

InterfaceDown - infromácia od protokolu z nižšej vrstvy, že rozhranie už nie je funkčné. To spôsobí prechod z akéhokoľvek stavu do stavu *Down*.