VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

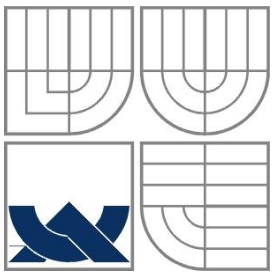
MODELOVÁNÍ MULTICASTOVÝCH DISTRIBUČNÍCH STROMŮ A KLIENTSKÝCH PROTOKOLŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

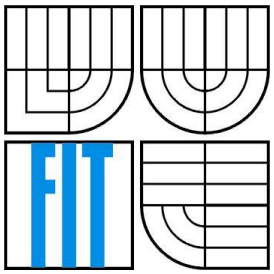
AUTOR PRÁCE
AUTHOR

BC. ADAM MALIK

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

MODELOVÁNÍ MULTICASTOVÝCH DISTRIBUČNÍCH STROMŮ A KLIENTSKÝCH PROTOKOLŮ

MULTICAST DISTRIBUTION TREES MODELLING IN OMNET++

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. ADAM MALIK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. VLADIMÍR VESELÝ

BRNO 2013

Abstrakt

Vďaka stále sa zvyšujúcemu dopytu po kvalitnom multimediálnom obsahu na internete, je potrebné zavádzať do aktuálnej sieťovej architektúry podporu pre multicastové smerovanie. Zavádzanie novej technológie do fungujúcej živej siete však môže byť problematické. Preto je vhodné danú zmenu najprv odskúšať v simulačnom prostredí, kde môžeme sieť podrobiť rôznym testom a až neskôr, pri priaznivých výsledkoch túto novú technológiu do našej siete implementovať. Cieľom tejto diplomovej práce je oboznámiť čitateľa s problematikou multicastového smerovania, popísať možnosti simulácie siete v simulátore OMNeT++ a implementácia nových modulov pre tento simulátor.

Abstract

Support of multicast routing and its implementation is one of the main goals in nowadays computer networks. Adapting new technology could be often challenging and connected with difficulties. For this reason its better to try it in some simulating enviroment and implement it only after successful results of tests and simulations. The aim of this diploma thesis is to familiarize the reader with the multicast routing, describe the possibilities of network testing in OMNeT++ and come up with new multicast routing framework for this discrete simulation tool.

Klíčová slova

OMNeT++, Multicastové smerovanie, IGMP, MLD, Cisco, INET, ANSA

Keywords

OMNeT++, Multicast routing, IGMP, MLD, Cisco, INET, ANSA

Citace

Malik Adam: Modelování multicastových distribučních stromů a klientských protokolů, diplomová práce, Brno, FIT VUT v Brně, 2013

Modelování multicastových distribučních stromů a klientských protokolů

Prehlásenie

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

……
Adam Malik
3.1.2012

Poďakovanie

Týmto by som chcel poďakovať svojmu vedúcemu práce Ing. Vladimírovi Veselému za poskytovanie cenných rád, podnetov, postrehov a taktiež za výdrž pri mojom nekonečnom množstve otázok. Ako odmenu by som mu rád venoval môj obľúbený recept, ktorý tvoril výraznú časť mojho jedálnička v posledné dni pred odovzdaním tejto práce. Recept je na špagety Carbonara. Toto jedlo je neuveriteľne chutné a jeho príprava nezaberie veľa času. Ako prvé je potrebné dať do hrnca variť väčšie množstvo vody, v ktorm uvaríme akékľvek cestoviny(áno viem, že som písal recept na špagety). Medzičasom si na panvici osmažíme slaninku. Čím viac tým lepšie, nakoľko sa riadim pravilom: „slaniny nieje nikdy dosť!“. Kým sa nám cestoviny dovaria, v miske si zmiešame žĺtka z troch vajíčok spolu s parmazánom a čiernym korením a pre krajšiu glazúru omáčky môžeme pridať trošku vody z cestovín. Po dovareni cestovín cestoviny scedíme a premiešame so slannkou. Túto zmes odstavíme z plamena a zalejeme pripravenou omáčkou v miske. Následne miešame až kým konzistencia cestovín nieje taká akú sme si predstavovali. V tomto okamihumi neostáv nič iné iba popriať dobrú chuť.

Obsah

1 Úvod

V dnešnej dobe lacného a rýchleho internetového pripojenia každodenne stúpa dopyt po kvalitnom multimediálnom obsahu. So zvyšujúcou sa rýchlosťou pripojenia narastá aj kvalita prenášaného obsahu. To však znamená aj väčšiu záťaž na prenosových linkách. Pre ich odľahčenie slúži multicastové vysielanie. Vďaka multicastu máme možnosť odosielať rovnaký obsah pre viacero koncových užívateľov bez zbytočného vyťažovania linky. Pred praktickým nasadením každej technológie je však vhodné otestovať jej vplyv na aktuálnu fungujúcu sieť. Vďaka modelovaniu a následnému testovaniu sa môžeme vyhnúť problémom, ktoré by mohli vzniknúť pri nasadzovaní novej technológie na neotestovanú živú sieť.

V tejto práci sa pokúsim oboznámiť čitateľa so základnými princípmi fungovania multicastových sietí. Stručne popíšem akým spôsobom sú smerované multicastové toky a taktiež akým spôsobom sa stanice prihlasujú k ich odberu. Cieľom tejto práce však bude navrhnúť možnosť implementácie protokolv IGMPv3 a oboch verzií MLD v simulačnom prostredí OMNeT++ a ich samotná implementácia. Implementované moduly budú otestované na referenčnej sieti, kde porovnám výsledky simulácie so správaním reálnych zariadení.

1.1 Stručný obsah kapitol

V druhej kapitole tejto práce sú popísané základné vlastnosti multicastového vysielania, princíp jeho fungovania. V kapitole sú ďalej podrobnejšie rozpísané podporné protokoly pre prihlasovanie užívateľov do multicastových skupín a taktiež popísaný princíp multicastového smerovania v sieti.

Tretia kapitola sa zaoberá podporou multicastového vysielania na smerovačoch Cisco. V kapitole sú popísané základné kroky potrebné k spusteniu multicastového smerovania na smerovači Cisco.

Štvrtá kapitola sa zaoberá simulačným nástrojom OMNeT++ spolu s jeho rozšírením INET. V kapitole je popísaná súčasná možnosť simulácie multicastového vysielania a popis návrhu implementácie nových modulov pre framework INET.

Piata kapitola je venovaná samotnému popisu implementácie jednotlivých protokolov. Sú v nej vysvetlené a opísané rôzne štruktúry a metódy. V kapitole je taktiež uvedený ClassDiagram prevytvorené protokoly. Ďalej je v kapitole popísaný spôsob možnosti nastavovania zariadení pomocou XML konfiguračného súboru a spôsob akým je toto nastavovanie zabezpečené. Poslednou časťou kapitoly je popis implementácie vizualizácie multicastových stromov v rámci IGMP a MLD.

Šiesta kapitola je venovaná simuláciám a testovaniu implementovaných protokolov voči reálnej sieti zostavenej zo zariadení Cisco. V kapitole budú popísané jednotlivé simulačné scenáre s ukázané jednotlivé testované parametre. Pre každý test bude na konci zhodnotenie dosiahnutia zhody v správani modelu voči reálnej sieti.

2 Multicast

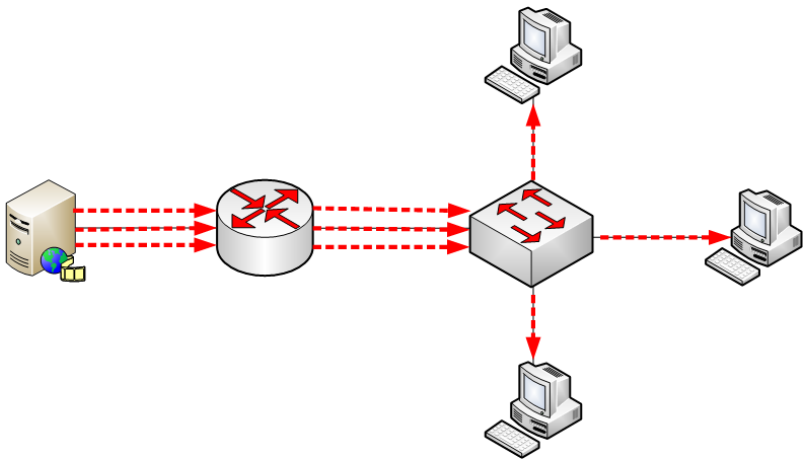
V tejto kapitole sú popísané základné princípy multicastového vysielania, popis adresovania na druhej a tretej vrstve modelu ISO/OSI. Ďalej je v kapitole detailne popísaný princíp smerovania multicastového vysielania a protokol IGMP pre prihlasovanie koncových staníc do multicastových skupín.

IP Multicast by sme mohli prirovnať k rádiu alebo televízii [1]. Televízny či rozhlasový signál môžu prijímať iba užívatelia so zapnutým prijímačom naladeným na presnú frekvenciou pre príjem určitého kanálu. Podobným spôsobom funguje aj prijímanie multicastového vysielania. Rozdiel je v tom, že užívateľ nemusí ladiť správnu frekvenciu ale stačí mu jednoducho prihlásiť sa do zvolenej multicastovej skupiny. Na rozdiel od televízie a rádia má užívateľ možnosť byť prihlásený do viacerých skupín súčasne a odoberať tak väčšie množstvo informácií.

2.1 Potrebujeme vôbec Multicast?

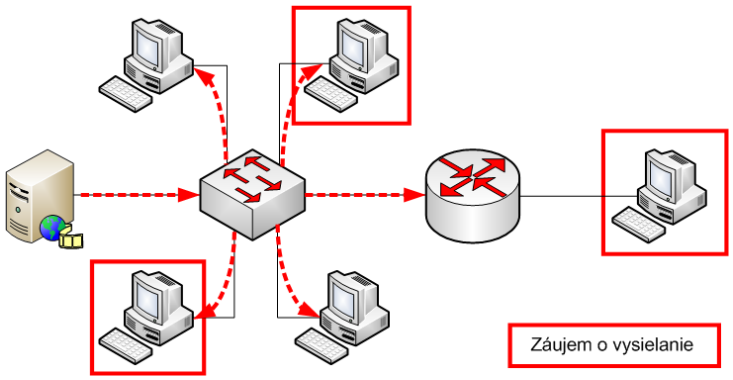
Internet je čoraz častejšie využívaný na prenos multimediálneho obsahu. Pre stále sa zvyšujúcu kvalitu tohto obsahu(hlavne u videa) je potrebné navyšovať kapacitu prenosového pásma. Predstavme si, že chceme vysielať multimediálne obsah pre tisíc koncových užívateľov. Ako k nim môžeme tento obsah dopraviť?

Jednou z volieb je unicastové vysielanie. To by však znamenalo posielať každému užívateľovi v separátnom toku požadovaný obsah. To však kvôli náročnosti na spracovávanie na strane serveru a náročnosti na prenosové pásmo bude zrejme pri vyššom počte užívateľov nemožné. Obrázok 2-1 popisuje vzniknutú situáciu.



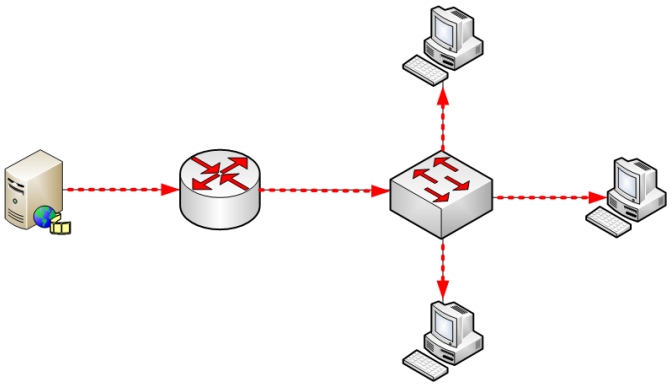
Obrázok 2-1 Unicastové vysielanie

Ďalšou z možností je použiť metódu broadcast vysielania. Na broadcastovú adresu odošleme obsah, ktorý bude doručený každému užívateľovi v sieti. Čo však s užívateľmi ktorý si obsah nevyžiadali? Obsah im bude zasielaný úplne zbytočne. Naopak, pokiaľ máme užívateľa mimo našej lokálnej siete LAN, obsah sa k nemu vôbec nedostane. Vzniknutú situáciu popisuje obrázok 2-2.



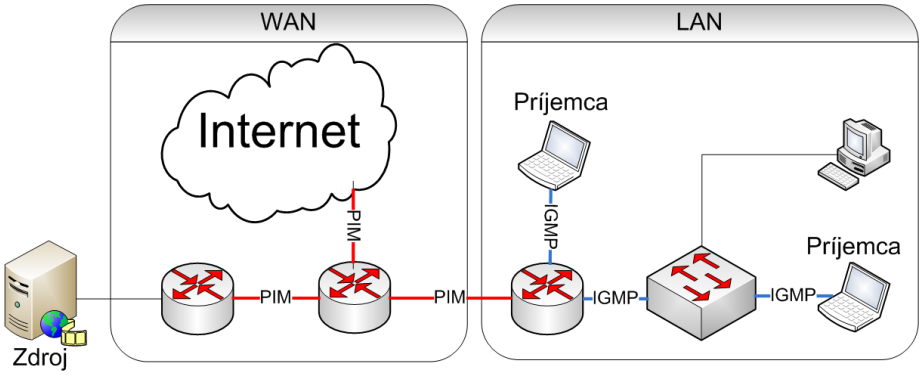
Obrázok 2-2 Broadcastové vysielanie

Vzniknutý problém môžeme vyriešiť práve použitím multicastového vysielania. Obsah bude odosielaný iba v jednom toku, čím ušetríme prenosové pásmo a taktiež bude obsah doručený všetkým užívateľom so záujmom o daný obsah a nebude zbytočne doručovaný ostatným.



Obrázok 2-3 Multicastové vysielanie

Keďže je potrebné dostať vysielanie od zdroja ku všetkým užívateľom so záujmom o vysielaný obsah, je potrebné zabezpečiť Smerovanie tohto obsahu k prihláseným užívateľom.



Obrázok 2-4 Smerovanie multicastového vysielania

2.2 Sieťová vrstva

2.2.1 Adresovanie v IPv4

Adresy pre IP multicast boli spoločnosťou IANA[1] zaradené do skupiny D IP adresného priestoru. Tieto adresy začínajú binárnym prefixom 1110 a spadajú tam všetky IP adresy v rozsahu 224.0.0.0 – 239.255.255.255.

Skupina	Prefix prvého oktetu	IP Rozsah
A	0xxxxxxx	0.0.0.0 – 127.255.255.255
B	10xxxxxx	128.0.0.0 – 191.255.255.255
C	110xxxxx	192.0.0.0 – 223.255.255.255
D (Multicast)	1110xxxx	224.0.0.0 – 239.255.255.255
E (Rezervované)	11110xxx	240.0.0.0 – 247.255.255.255

Tabuľka 2-1: Rozdelenie IP adresného rozsahu.

Tento rozsah je ešte rozdelený na tri časti [2]. Prvá časť je rozsah 224.0.0.0 – 224.0.0.255, ktorý sa používa iba v lokálnych sieťach. Používa TTL = 1 a nie je smerovačom preposielaný mimo lokálnej siete. Adresy z tohto rozsahu sú používané hlavne sieťovými protokolmi.
Pre podporu multicastu sú používane hlavne adresy 224.0.0.1, 224.0.0.2 a 224.0.0.22.

IP adresa	Použitie
224.0.0.1	Všetky koncové zariadenia v sieti
224.0.0.2	Všetky smerovače v sieti
224.0.0.5	Všetky OSPF smerovače v sieti
224.0.0.6	Všetky OSPF designated smerovače v sieti
224.0.0.9	Všetky RIPv2 smerovače v sieti
224.0.0.10	Všetky IGRP smerovače v sieti
224.0.0.13	Všetky PIM smerovače v sieti
224.0.0.22	Všetky IGMPv3 smerovače v sieti

Tabuľka 2-2: Príklad použitia multicastových adries [3].

Druhou časťou je rozsah adries 224.0.1.0 - 238.255.255.255, ktoré sú používané globálne v internete.

[1] IANA(Internet Assigned Numbers Authority) http://www.iana.org

Poslednou skupinou sú adresy v zvyšnom rozsahu 239.0.0.0 – 239.255.255.255, ktoré sú používane podobne ako privátne IP adresy, a to iba vo vnútorných sieťach a nie sú globálne smerované v internete.

2.2.2 Adresovanie v IPv6

Na rozdiel od adresovania v IPv4, v IPv6 už adresy nemáme rozdelené do skupín. multicastové adresy v IPv6 začínajú prefixom FFxx a nasledujúce dve hodnoty určujú dosah adresy. Formát IPv6 multicastovej adresy je popísaný na nasledujúcom obrázku:

Obrázok 2-5 Formátmulticastovej IPv6 adresy

Prvých 8 bitov v adrese nastavených na hodnotu 1 značí že sa jedná o multicastovú adresu. Ďalšie 4 bity adresy nám tvoria príznaky, kde:

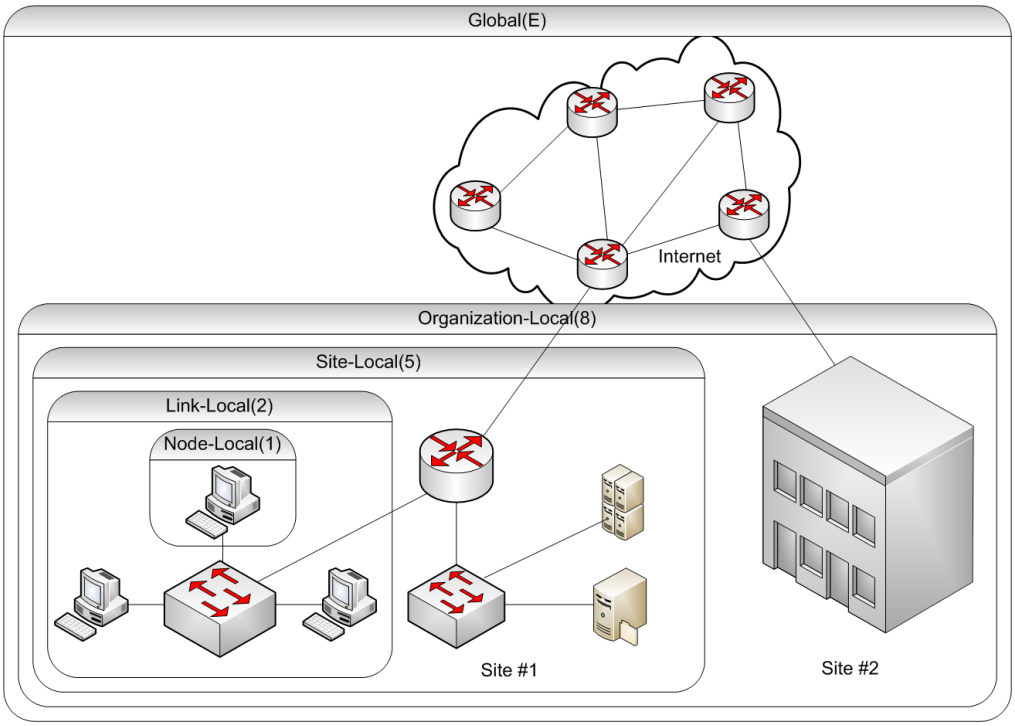
- Flag **O** (High-Order flag): je rezervovaný a musí byť nastavený na hodnotu 0.
- Flag **T**: v prípade, že je nastavený na hodnotu 0, označuje permanentne pridelenú adresu priradenú spoločnosťou IANA. V opačnom prípade, ak je príznak nastavený na hodnotu 0, adresa je dynamicky priradenou multicastovou adresou.
- Flag **P**: v prípade, že je nastavený na hodnotu 0, označuje multicastovú adresu, ktorá nie je priradená na základe sieťového prefixu. V prípade nastavenia bitu na 1, multicastová adresa je priradená na základe sieťového prefixu.
- Flag **R**: nastavený na hodnotu 1a značí multicastovú adresu, ktorá obsahuje adresu RP. To však znamená že na hodnotu 1 musia byť nastavené aj flagy **P** a **T.**

Pole scope nám označuje dosah danej adresy a jej hodnoty môžu byť [4] [5]:

Hodnota	Dosah	Adresa	TTL
2	Interfeace-Local	FF01:/8	0
3	Link-Local	FF02:/8	1
6 - 7	Site-Local	FF05:/8	<32
9 - D	Organization-Local	FF08:/8	≤255
F	Global	FF0E:/8	≤255

Tabuľka 2-3 Dosah IPv6 multicastových adries

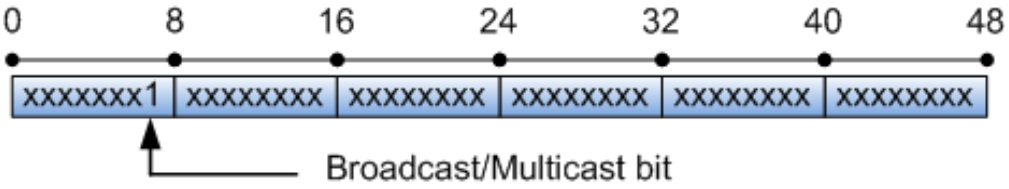
Dosah Interface-Local pokrýva iba samotné rozhranie a je teda používaný pre multicastové vysielanie na loopback rozhraní. Link-Local dosah pokrýva lokálnu sieť po najbližší smerovač. Site-Local dosah pokrýva celú pobočku, a spojenie viacerých pobočiek spadá pod dosah Organization-Local. Na najvyššej úrovni sa nachádza Global scope pokrývajúci celú globálnu sieť. Zanorenie týchto dosahov ilustruje doleuvedený obrázok [6]:



Obrázok 2-6 Dosah multicastových adries

2.3 Linková vrstva

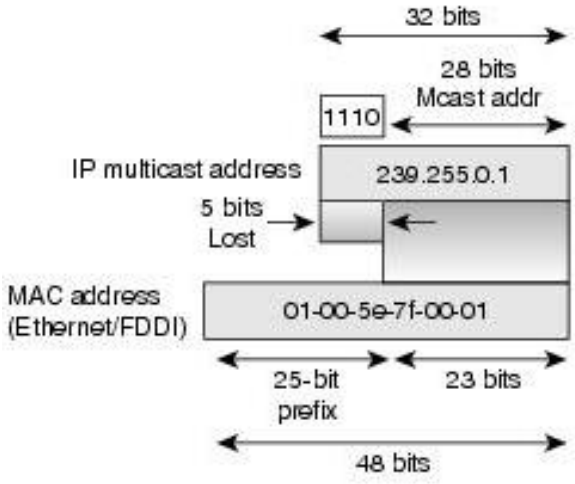
Každé sieťové rozhranie štandardne prijíma pakety určené MAC adresou daného rozhrania a taktiež pakety určené broadcastovou MAC adresou [7]. Pre príjem multicastu musia teda byť rozhrania schopné prijímať pakety označené multicastovou MAC adresou, avšak stále musia dokázať rozlišovať pakety určené pre rôzne multicastové skupiny. Rozpoznávanie broadcast a multicast paketov je zabezpečené pomocou posledného bytu prvého oktetu. Ten ukazuje či sa jedná o unicastový alebo broadcastový/multicastový paket.



Obrázok 2-7 Broadcast/Multicast bit

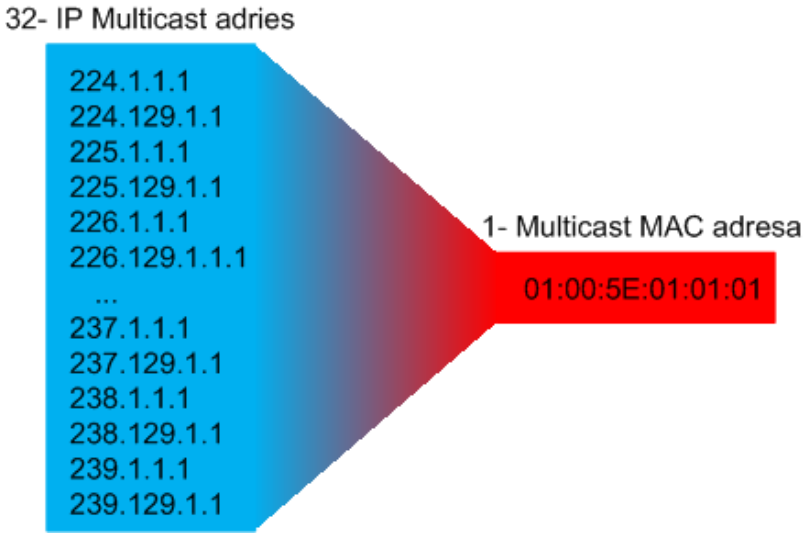
2.3.1 Mapovanie MAC adries pre multicastové IPv4 adresy

Pre multicastové MAC adresy bola spoločnosťou IANA vyhradená polovica bloku ethernetových MAC adries s prefixom 01:00:5E. Tento blok má rozsah 01:00:5E:00:00:00 – 01:00:5E:7F:FF:FF. Znamená to, že na priradenie multicastovej skupiny k MAC adrese môžeme použiť 23 bitov. Ako už vieme, multicastová skupina je identifikovaná pomocou 28 bitov a preto sme nútení vrchných 5 bitov zahodiť. Vďaka tomu multicastová adresa nebude unikátna.



Obrázok 2-8 Mapovanie IP adresy na MAC adresu [7]

Ako je na obrázku 2-9 vidieť, 32 rôznych multicastových skupín bude spadať pod práve jednu MAC adresu.



Obrázok 2-9 "32 to 1 overlaping problem"

2.3.2 Mapovanie MAC adries pre multicastové IPv6 adresy

V IPv6 multicastové MAC adresy začínajú prefixom 33:33. To teda ponecháva zvyšných 32 bitov MAC adresy pre adresovanie skupín. Nedochádza tu teda ku prekrývaniu IP adries tak ako tomu bolo u IPv4. Popis mapovania multicastovej IPv6 adresy na jej odpovedajúcu MAC adresu zobrazuje obrázok:

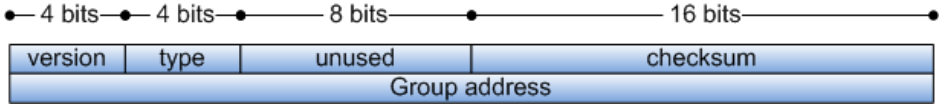
Obrázok 2-10 Mapovanie IPvž adresy na MAC adresu

2.4 Prihlasovanie do skupín v IPv4

Prihlasovanie a odhlasovanie užívateľov do skupín je zabezpečené výmenou IGMP správ. Práve vďaka týmto správam si môže každý smerovač udržiavať pre každé svoje rozhranie zoznam multicastových skupín, ku ktorým sú jeho užívatelia prihlásený.

2.4.1 IGMPv1

IGMP je asymetrický protokol používaný koncovými stanicami na oznamovanie príslušnosti v multicastových skupinách svojmu lokálnemu smerovaču [8]. Protokol IGMP je podobne ako protokol ICMP súčasťou protokolu IP. To znamená, že každá IGMP správa je zabalená do IP datagramu s protokolovým číslom 2. Obrázok 2-11 znázorňuje formát IGMP správy.



Obrázok 2-11 Formát IGMP správy

- **Version** – verzia protokolu (0x1)
- **Type** – typ IGMP správy: *Host Membership Query* (0x1)
  Host Membership Report (0x2)
- **Unused** – pole vyplnené nulami, pri prijatí je ignorované
- **Checksum** – kontrolný súčet
- **Group address** – toto pole je používané iba v *Host Membership Report* správach a obsahuje adresu oznamovanej skupiny.

2.4.1.1 Fungovanie protokolu

Lokálne smerovače posielajú *Host Membership Query* správy aby zistili, do ktorých skupín sú prihlásený užívatelia v ich lokálnej sieti. Tieto správy sú posielané na adresu 224.0.0.1 a majú nastavený parameter *TTL* na hodnotu 1. Koncové stanice na túto správu odpovedajú vygenerovaním a odoslaním *Host Membership Report* správy na rozhranie, z ktorého im prišla *Host Membership Query* správa. Pomocou týchto správ reportujú všetky skupiny svoj záujem prijímať multicastové vysielanie. Aby sa zamedzilo odoslaniu veľkého množstva správ naraz, používajú sa dve techniky, ktoré tento problém riešia:

- V okamihu prijatia *Host Membership Query* správy, spustí klient namiesto odoslania odpovede časovač s náhodne vygenerovaným časom v intervale $\langle 0 - D \rangle$ sekúnd pre každú svoju odpoveď. V okamihu vypršania časovača, klient vygeneruje *Host Membership Report* správu, ktorú následne odošle. Pomocou tohto mechanizmu je zabezpečené, že správy budú postupne doručené v časovom intervale D sekúnd a nebudú odoslané naraz.

- *Host Membership Report* je odoslaný na cieľovú IP adresu oznamovanej skupiny s hodnotou *TTL* nastavenou na 1. To znamená, že odoslaný report zaregistrujú všetky koncové stanice v danej skupine. Ak klientská stanica zaznamená report, zastaví svoj časovač a report nevygeneruje. To znamená, že pre každú oznamovanú skupinu bude vygenerovaný práve jeden report od klientskej stanice, ktoréj vyprší časovač ako prvému. Keďže multicastové smerovače prijímajú všetky multicastové datagramy, nie je potrebne adresovať report priamo smerovaču. Keďže smerovač nepotrebuje poznať všetky koncové stanice v danej skupine, stačí mu vedieť, že z danej časti siete má aspoň niekto z jeho klientov o multicastové vysielanie danej skupiny záujem.

Multicastové smerovače posielajú *Host Membership Query* správy pravidelne, aby zistili či záujem prijímať multicastové dáta stále trvá. V prípade, že smerovač neobdrží *Memership Report* správu na niekoľko svojich odoslaných *Membership Query* správ usúdi, že na danom segmente už o prijímanie dát pre danú skupinu nemá nikto záujem a prestane tam tieto dáta preposielať.
V prípade, že sa klient pripojí do novej skupiny a na danom sieťovom segmente je jediným so záujmom prijímať vysielanie tejto skupiny, odosiela *Membership Report* správu a nečaká na prijatie *Membership Query* správy od smerovača. Pre prípad straty alebo zničenia je táto správa odoslaná niekoľko krát po sebe v krátkych časových úsekoch. Stavový diagram na obrázku ilustruje tento postup.

Každé z koncových zariadení sa môže nachádzať v jednom z nasledujúcich stavov:

- ***Non-Member* stav -** počiatočný stav pred tým, než je klient prihlásený do skupiny
- ***Delaying Member* stav -** stav, v ktorom klientská stanica patrí do skupiny a má spustený časovač pre odoslanie *Membership Report* správy.
- ***Idle Member* stav -** stav, v ktorom koncová stanica patrí do skupiny a nemá spustený časovač pre odoslanie *Membership Report* správy.

V diagrame môže nastať päť udalostí, ktoré zapríčinia zmenu stavu koncovej stanice:

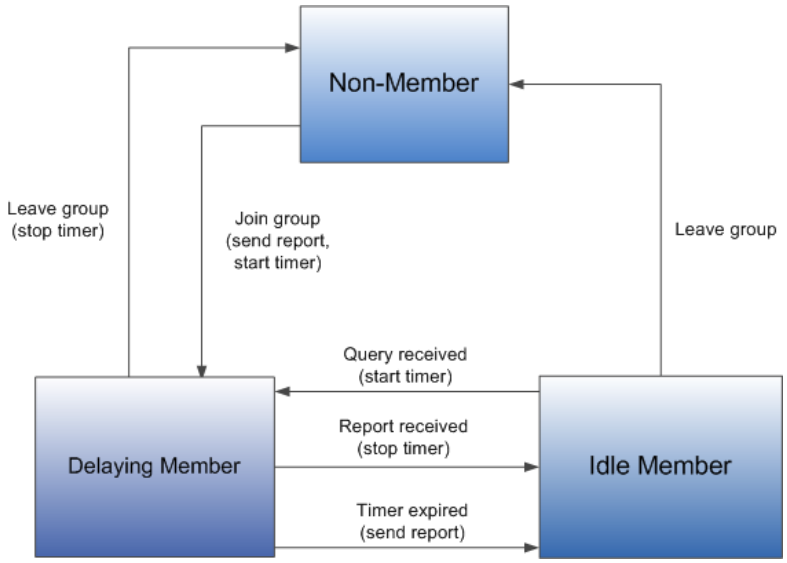
- **Prihlásenie do skupiny -** nastáva v okamihu, keď sa klient rozhodne stať odberateľom niektorej zo skupín. Táto udalosť môže nastať iba pokiaľ je klient v stave *Idle*.
- **Opustenie skupiny -** nastáva v okamihu, keď sa klient rozhodne, že už ďalej nechce odoberať vysielanie danej skupiny. Táto udalosť môže nastať iba pokiaľ sa klient nachádza v stave *Idle* alebo *Delaying Member*.
- **Prijatá *Membership Query* správa -** táto udalosť nastáva ak Klient obdrží platnú IGMP *Host Membership Query* správu. Aby bola správa platná, musí mať dĺžku aspoň 64 bitov, mať správny kontrolný súčet a mať správne nastavenú cieľovú IP adresu na 224.0.0.1. Správu obdržia všetky zariadenia na segmente, kde bola správa odoslaná avšak klienti, ktorý sú v stave *Non-Member* alebo *Delaying Member* túto správu ignorujú.
- **Prijatá *Membership Report* správa** - táto udalosť nastáva ak klient obdrží platnú *Membership Report* správu. Správa je platná vtedy ak je jej dĺžka aspoň 64 bitov, má správny kontrolný súčet, a ako cieľovú IP adresu má nastavenú adresu skupiny pre ktorú bola odosielaná. Táto správa bude doručená všetkým klientom na danom sieťovom segmente, ktorý sú členmi danej skupiny. Správa je ignorovaná koncovými stanicami v stavoch *Idle* a *Non-Member*.
- **Vypršanie časovača -** nastáva v okamihu, keď časovač pre oneskorené odoslanie *Membership Report* správy vyprší. Táto udalosť môže nastať iba pokiaľ je klient v stave *Delaying Member*.

Všetky ostatné udalosti ako napríklad prijatie neplatnej správy sú vo všetkých stavoch ignorované.

V diagrame môžu pre určité udalosti nastať 3 rôzne akcie:

- Odoslanie *Membership Report* správy pre skupinu na danom sieťovom segmente.
- Zapnutie časovača pre oneskorené odoslanie *Membership Report* správy.
- Vypnutie časovača pre oneskorené odoslanie *Membership Report* správy.

V nasledujúcom diagrame sú udalosti, ktoré spôsobili prechod zobrazené vedľa prechodu. V zátvorkách je následne uvedená prípadná akcia, ktorá sa pri prechode vykonáva.

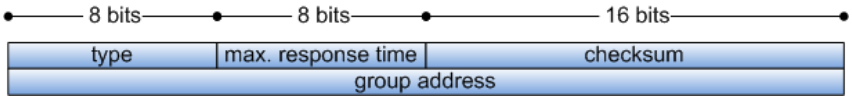


Obrázok 2-12 Stavový diagram IGMPv1 koncovej stanice

Jediná skupina, ktorá nepoužíva hore uvedený diagram je skupina pre všetkých klientov na adrese 224.0.0.1. Koncové stanice v nej začínajú v stave *Idle*, nikdy sa z neho nedostanú do iného stavu a nikdy neposielajú *Membership Report* správu pre skupinu 224.0.0.1.

2.4.2 IGMPv2

IGMP vo verzii 2 má správy vo formáte popísanom na obrázku nižšie [9].



Obrázok 2-13 Formát IGMPv2 správy

- **Type -** v IGMPv2 máme tri druhy správ:
  - *Membership Query* (0x11), ktorý rozdeľujeme na dve podtriedy:
    i) *General Query*, ktorá sa používa na zistenie všetkých skupín, ktoré majú na danom sieťovom segmente aktívnych odberateľov.
    ii) *Group-Specific Query*, ktorou zisťujeme či má daná konkrétna skupina na sieťovom segmente nejakých odberateľov.
  - *Version 2 Membership Report* (0x16)
  - *Leave Group* (0x17)
- **Max response time -** toto pole je používané iba v správach typ *Membership Query* a určuje nám maximálny možný čas, o ktorý sa môže zdržať odoslanie *Membership Report* správy. V ostaných typoch správ je nastavovaný na nulovú hodnotu a ignorovaný.
- **Checksum -** kontrolný súčet

- **Group address** - pri odosielaní *General Membership Query* je nastavená na 0. V prípade, že je správa odosielaná ako *Group-Specific Query*, je táto adresa nastavená na adresu danej skupiny. V správach typu *Membership Report* alebo *Leave Group* si pole ponecháva IP adresu skupiny, pre ktorú daná správa patrí.

Popis fungovania protokolu

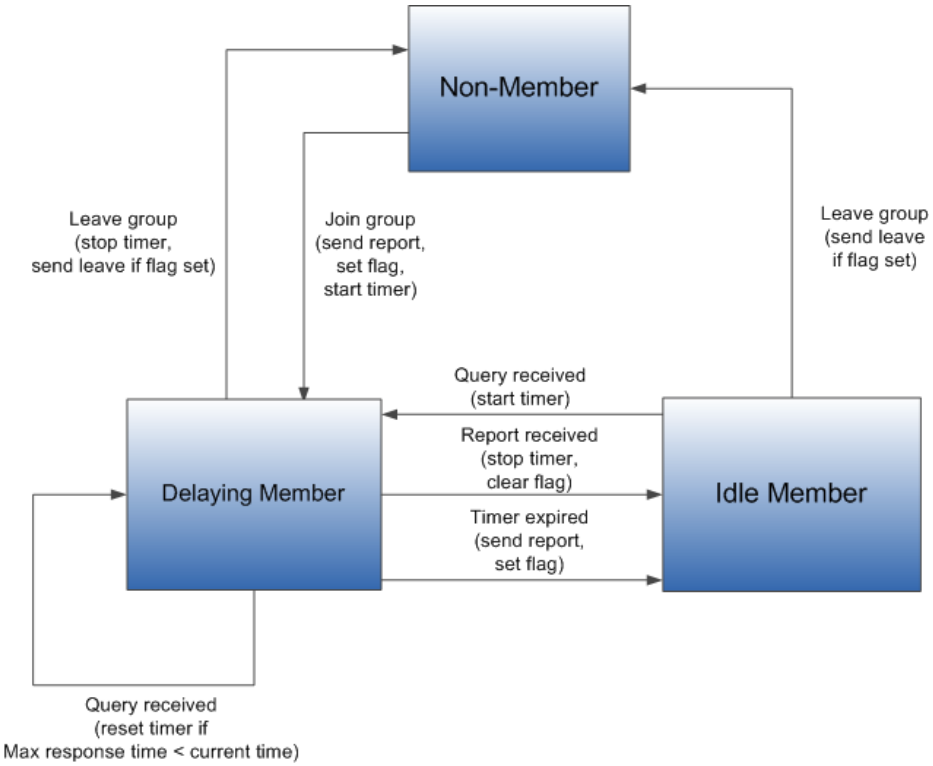
Každý multicastový smerovač si udržuje zoznam aktívnych skupín a časovač pre každú skupinu vo svojej lokálnej sieti. Voči každej multicastovej skupine môže byť jej lokálny smerovač v postavení tzv. *Querier* alebo *Non-Querier*. Štandardne je pre každú fyzickú sieť iba jeden *Querier* smerovač. Každý multicastový smerovače začína v roli *Querier*-a pre každú svoju pripojenú lokálnu podsieť. V prípade že smerovač zachytí *Membership Query* správu od smerovača s nižšou IP adresou, okamžite sa stane *Non-Querier* smerovačom pre danú sieť. Každý *Querier* smerovač potom pravidelne posiela *General Query* správy do podsiete pre ktorú je *Querier*, aby tak získal informácie o aktívnych skupinách.

Ak Host obdrží *General Query* správu, aktivuje pre dané sieťové rozhranie časovač pre každú skupinu do ktorej je prihlásený. Každý časovač je nastavený na náhodnú hodnotu s ohľadom na hodnotu *Max Response Time*. V prípade, že obdrží *Group-Specific Query*, spustí časovač iba pre danú skupinu. V okamihu vypršania časovača, Host odosiela *Version 2 Membership Report* s hodnotou *TTL* = 1 pre danú skupinu. V prípade, že klient obdrží *Membership Report* správu či už verzie 1 alebo 2, zastaví svoj časovač a nevykoná žiadnu akciu.

V okamihu keď smerovač prijme *Membership Report* správu, obnoví časovač (*Group Membership Interval*) pre danú skupinu. Pokiaľ časovač vyprší, smerovač predpokladá, že o vysielanie v danej multicastovej skupine už nemá nikto v podsieti záujem a prestane do danej podsiete vysielanie preposielať. Rovnako ako vo verzii 1, v okamihu prihlásenia Hosta do skupiny, bez vyzvania vyšle v krátkom časovom úseku niekoľko *Initial Version 2 Membership Report* správ pre prihlásenie k odberu skupinového vysielania.

Pokiaľ klient opustí skupinu a bol posledným, ktorý posielal odpoveď na *Membership Query* správu, musí odoslať na adresu všetkých multicast smerovačov 224.0.0.2 *Leave Group* správu. Po prijatí tejto správy *Querier* smerovačom(*Non-Querier* smerovače tieto správy ignorujú), smerovač odošle *Group-Specific Query* správy aby sa uistil, že na danom segmente nezostal nikto so záujmom o danú skupinu. V prípade, že sa smerovaču nevráti odpoveď, prestane na daný segment multicastové vysielanie danej skupiny preposielať.

Rozdiel medzi stavovým diagramom pre IGMP klienta verzie 1 a verzie 2, spočíva v pridaní odoslania *Leave Group* správy pri opúšťaní skupiny a nastavovaniu príznaku posledného Hosta odpovedajúceho na *Membership Query* správu. Vo verzii 2 bolo pridané obnovovanie časovača pri prijatí *Membership Query* správy s nižšou hodnotou *Max. response time*.



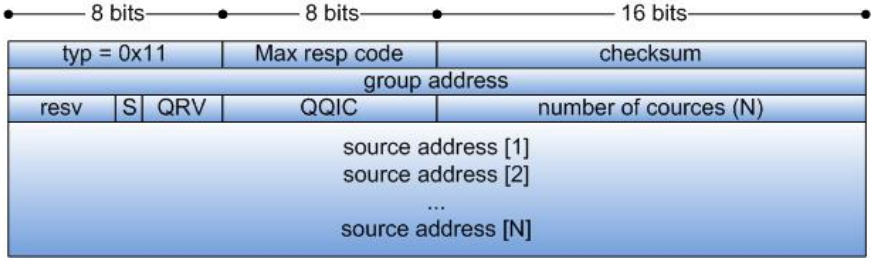
Obrázok 2-14 Stavový diagram IGMPv2 koncovej stanice

2.4.3 IGMPv3

IGMP verzie 3 používa opäť iba dva druhy správ, ktorými sú *Membership Query* a *Version 3 Membership Report.* Pre zachovanie spätnej kompatibility však IGMPv3 podporuje aj nasledovné tri staršie typy správ: *Version 1 Membership Report*, *Version 2 Membership Report* a *Version 2 Leave Group*.

Správa Membership Query

Formát *Membership Query* správy je nasledovný:



Obrázok 2-15 Formát IGMPv3 Membership Query správy

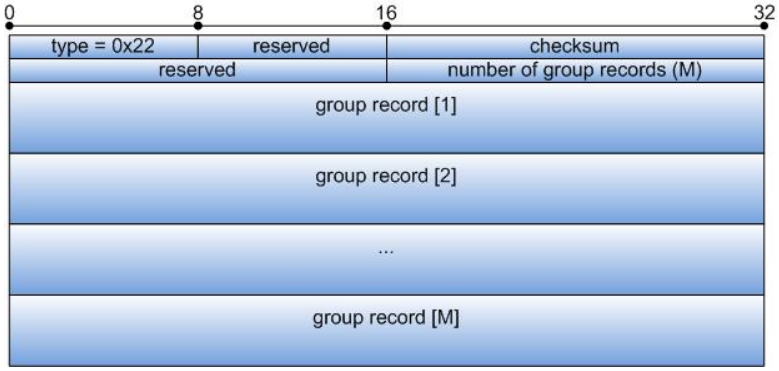
- **Max. resp code -** definuje maximálny čas pre zdržanie pred odoslaním odpovedajúcej *Membership Report* správy.
- **Group address -** podobne ako u IGMP verzie 2, je nastavená na 0 v prípade, že je odosielaná ako *GeneralQuery*. Ak je odosielaná ako *Group-Specific Query* alebo *Group-and-Source-Specific Query*, je táto hodnota nastavená na Multicastovú IP adresu skupiny.
- **Resv -** bity rezervované pre budúce použitie
- **S flag -** príznak pre potlačenie spracovania na strane smerovača
- **QRV**(*Querier's Robustness Variable*) - smerovače preberajú túto hodnotu z posledných prijatých *Membership Query* správ.
- **QQIC**(*Querier's Query Interval Code*) - časový interval pre posielanie Query správ. Smerovače ho preberajú z posledných prijatých *Membership Query* správ.
- **Number of sources** (N) - tento údaj nám prezrádza koľko zdrojových adries je uvedených v *Membership Query* správe. Hodnota tohto poľa je nastavená na 0 v prípade, že posielame *General Query* alebo *Group-Specific Query*. V prípade *Group-and-source-specific Query* je táto hodnota nastavená na nenulovú hodnotu.
- **Source address** [i] - je unicastová IP adresa zdroju dát, kde i označuje poradie adresy v zozname.

Membership Query správy sa podobne ako v predchádzajúcej verzii delia na tri podskupiny:

- *General Query* - tieto správy rozosielajú multicastové smerovače pre zistenie všetkých multicastových skupín na svojich sieťových rozhraniach. Tento druh správy je odosielaný na IP adresu 224.0.0.1 pre všetky systémy v sieti.
- *Group-Specific Query* - táto správa slúži smerovačom na zistenie údajov o konkrétnej multicastovej skupine. Cieľová IP adresa tejto správy je nastavená na multicastovú IP adresu požadovanej skupiny.
- *Group-and-Source-Specific Query* - správa, ktorú používajú smerovače pre zistenie informácii o konkrétnej multicastovej skupine a od konkrétneho zdroja vysielaných dát. Rovnako ako v predchádzajúcom prípade, je správa zasielaná na IP adresu skupiny.

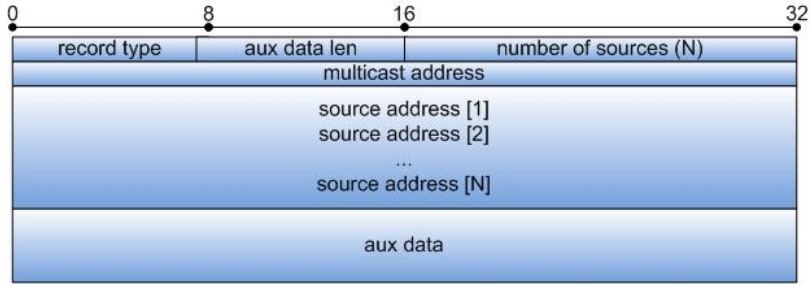
Version 3 Membership Report Správa

Tento druh správy odosielajú Hosti pre svoj lokálny smerovač a oznamujú v nej stav multicastovej skupiny a prípadné zmeny stavu skupiny. Správa má nasledujúci formát:



Obrázok 2-16 Formát IGMPv3 Membership Report správy

- **Reserved** - rezervované bity pre budúce použitie
- **Checksum** - kontrolný súčet
- **Number of group records**(M) - oznamuje koľko záznamov o skupinách bude nasledovať za hlavičkou
- **Group record** - je pre každú skupinu a má nasledovný formát:



Obrázok 2-17 Formát skupinového záznamu v IGMPv3 Membership Report správe

- **Record type** - pole určuje typ záznamu pre skupinu
  - *Current-State Record* – odpoveď Hosta na *Membership Request* správu, kde v odpovedi posiela aktuálny filtrovací mód a zoznam zdrojových IP adries.
    - MODE_IS_INCLUDE(0x1) – typ filtrovania zdrojov je INCLUDE
    - MODE_IS_EXCLUDE(0x2) – typ filtrovania zdrojov je EXCLUDE
  - *Filter-Mode-Change Record* – typ správy, ktorú Host odosiela pri zmene svojho filtrovacieho režimu:
    - CHANGE_TO_INCLUDE_MODE(0x3) – zmena filtrovacieho režimu na INCLUDE. V správe je odoslaný nový zoznam zdrojových adries pre danú Multicastovú adresu.
    - CHANGE_TO_EXCLUDE_MODE(0x4) – zmena filtrovacieho režimu na EXCLUDE. Pole zdrojových adries obsahuje nové záznamy pre danú Multicastovú adresu.

- *Source-List-Change Record* – Host odosiela túto správu v prípade, že nastala zmena v zozname zdrojových adries:
  - ALLOW_NEW_SIURCES(0x5) – v poli zdrojových adries budú nové záznamy adries, od ktorých chceme prijímať vysielanie. V prípade, že režim filtrovania bude nastavený na INCLUDE, budú tieto adresy pridané na zoznam zdrojov, v opačnom prípade pri filtrovaní EXCLUDE, budú z tohto zoznamu odstránené.
  - BLOCK_OLD_SOURCES(0x6) – v poli zdrojových adries budú adresy skupín, o ktoré už daný Host nemá záujem. V prípade nastavenia filtra na INCLUDE budú tieto adresy zmazané zo zoznamu zdrojov, naopak pri nastavenom filtri EXCLUDE, budú tieto adresy na zoznam zdrojov pridané.

- **Aux data len** - udáva aké množstvo pomocných údajov bude nasledovať po zdrojových adresách. Ak je toto pole nastavené na hodnotu nula, indikuje to absenciu pomocných údajov.
- **Number of sources** (N) - udáva koľko zdrojových adries bude nasledovať po hlavičke.
- **Multicast address** - je adresa, pre ktorú dané zdroje patria.
- **Source address** [i] - unicastová adresa zdroja dát, kde i udáva poradie záznamu.
- **Aux data** - V implementácii pre IGMPv3, sa tieto údaje nepoužívajú

Fungovanie protokolu

Správanie koncových staníc

Novinkou v IGMP verzie 3 je filtrovanie zdrojov. Ak sa na sieti nachádza viacero zdrojov vysielajúcich pre rovnakú multicastovú skupinu, majú klienti možnosť filtrovať dané zdroje. Pomocou filtru INCLUDE vymenujeme zdroje, od ktorých chceme pre danú multicastovú skupinu prijímať vysielanie. Naopak pomocou filtru EXCLUDE zabezpečíme príjem vysielania zo všetkých zdrojov okrem zdrojov uvedených v zozname.

V prípade, že sa klient pripojí do novej skupiny, bude mat filter nastavený na EXCLUDE a bude mať prázdny zoznam zdrojov, čo značí príjem od všetkých zdrojov v danej multicatovej skupine. Následne má klient možnosť tento zoznam upravovať pomocou zmien filtra a prípadným pridávaním alebo uberaním zdrojov zo zoznamu.

V prípade odhlásenia koncovej stanice zo skupiny, je nastavený jeho filter pre danú skupinu na INCLUDE a zoznam zdrojov je prázdny. To značí, že klient nechce prijímať multicastové vysielanie od žiadneho zdroja v danej skupine.

Všetky zmeny u klientov sú oznamované lokálnemu smerovaču pomocou *Version 3 Membership Report* správy, ktorá je odosielaná na multicastovú adresu 224.0.0.22 pre všetky IGMPv3 smerovače.

Tieto zmeny sú vyhodnocované porovnaním pôvodného stavu a stavu po zmene. V nasledujúcej tabuľke sú popísané všetky možné kombinácie stavov a akcie, ktoré sú vykonané.
Veľké písmeno označuje množinu zdrojových adries a akcia A-B označuje rozdiel množín.

Pôvodný stav	Nový stav	Odoslaný report
INCLUDE(A)	INCLUDE(B)	ALLOW(B-A), BLOCK(A-B)
EXCLUDE(A)	EXCLUDE(B)	ALLOW(A-B), BLOCK(B-A)
INCLUDE(A)	EXCLUDE(B)	TO_EX(B)
EXCLUDE(A)	INCLUDE(B)	TO_IN(B)

Tabuľka 2-4: Akcie pri zmene na rozhraní koncovej stanice.

Tak ako vo verzii 2, musia Koncové stanice odpovedať na odoslané Report správy. Odloženie odoslania odpovede zostalo nezmenené, avšak zmena nastala pri prijatom reporte od inej koncovej stanice. Vo verzii 3 si stanica svoj časovač nezruší. Namiesto toho, podľa nasledujúcich piatich pravidiel naplánuje svoju odpoveď.

1. Ak je naplánovaná odpoveď na predchádzajúcu *General Query*, ktorej časovač vyprší skôr ako zvolený čas pozdržania odpovede, nieje potrebné plánovať ďalšiu odpoveď.
2. Ak je prijatá správa *General Query,* časovač na rozhraní sa nastaví na zvolenú hodnotu pozdržania odpovede a akákoľvek predchádzajúca odpoveď na *General Query* je zrušená.
3. Ak je prijatá *Query* správa *Group Specific Query* alebo *Group-and-Source Specific Query* a nieje spustený časovač pre túto skupinu, tak použijeme skupinový časovač na naplánovanie *Report* správy. V prípade, že sa jedná o *Group-and-Source Specific Query,* je potrebné k časovaču pridať aj zoznam zdrojov, ktorý bude použitý pri generovaní odpovede.
4. V prípade, že pre skupinu beží časovač na predchádzajúcu *Query* správu a nová *Query* správa je typu *Group-Specific* alebo je zoznam zdrojových adries v *Query* prázdny, zoznam zdrojov priradených k bežiacemu časovaču sa zmaže a použije sa skupinový časovač na naplánovanie odpovede. Hodnota časovača je ponechaná v prípade, že nový naplánovaný čas vyprší neskôr ako aktuálny časovač. V opačnom prípade sa hodnota časovača nastaví na novú skoršiu hodnotu.
5. Ak pre skupinu beží časovač a zoznam zdrojov priradený k tomuto časovaču nieje prázdny, je zoznam zdrojov rozšírený o nové zdrojové adresy obdržané v *Query*. Rovnako ako v predchádzajúcom prípade je odoslaná iba jedna *Report* správa v najskoršom možnom čase.

Po vypršaní časovačov dochádza k odoslaniu odpovedí. Môžu teda nastať tri rôzne prípady.

1. Ak nám vypršal časovač na rozhraní je potrebné ako odpoveď odoslať *Current-State Record,* ktorý bude obsahovať všetky multicastové adresy, ktoré sú na danom rozhraní zaregistrované. Pre každú multicastovú adresu je zároveň odosielaný filtrovací mód spolu so zoznamom zrojovych adries.

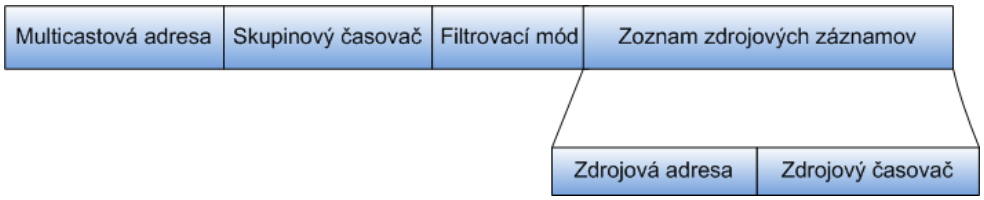
2. Ak vyoršal časovač pre skupinu a zoznam zdrojov priradených k skupine bol prázdny, je potrebné odoslať *Current-State Record* pre danú skupinu. Opäť ako v predchádzajúcom prípade je spolu so skupinovou adresou odosielaný aj jej fitrovací mód a zoznam zdrojových adries.
3. Posledným prípadom je vypršanie skupinového časovača s priradeným neprázdnym zoznamom zdrojov. V tomto prípade je odosielaný *Current-State Record* podľa nasledujúcej tabuľky.

Stav rozhrania	Zoznam zdrojov pre časovač	Current-State Recod
INCLUDE(A)	B	IS_IN(A*B)
EXCLUDE(A)	B	IS_EX(B-A)

Tabuľka 2-5: Akcie pri vytváraní *Current-State Record*-u.

Správanie smerovača

Dotazovanie smerovača pomocou *Membership Query* správ funguje podobne ako pri verzii 2. Novinkou však je pridanie podskupiny *Group-and-Source-Specific Query*, v ktorej smerovač zisťuje záujem o vysielanie pre danú skupinu z vybraného zoznamu zdrojov. Vďaka pridaniu možnosti filtrovať zdrojové adresy, bolo potrebné pridať nové štruktúry pre uchovávanie záznamov jednotlivých skupín. Každý IGMPv3 smerovač si teda udržuje pre každú svoju multicastovú skupinu na rozhraní novú štruktúru v tvare:



Obrázok 2-18 Formát štruktúry skupinového záznamu smerovača

Filtrovací mód smerovača je udržiavaný pre celú skupinu a je nastavený tak aby zohľadňoval požiadavky všetkých koncových staníc a zároveň nenarastala zložitosť štruktúry. Pre určovanie filtrovacieho módu platí jednoduché pravidlo: Akonáhle je prijatá zmena filtrovacieho módu na stav *EXCLUDE*, zmení filtrovací mód celej skupiny taktiež na *EXCLUDE* a v zozname zdrojových záznamov sú udržiavané dva typy záznamu. Prvým z nich sú záznamy s bežiacim zdrojovým časovačom. Druhú skupinu tvoria záznamy, ktorým už zdrojový časovač vypršal. V prípade, že je filtrovací mód nastavený do stavu *INCLUDE*, v zdrojových záznamoch sú udržiavané iba záznamy z bežiacim časovačom. Keďže prijatie skupinového zoznamu s filtrom *EXCLUDE* spôsobí prechod celého stavu skupiny do módu *EXCLUDE*, je potrebné zabezpečiť mechanizmus opätovného návratu do stavu *INCLUDE*. Pravé pre tento prípad je používaný skupinový časovač, ktorý je používaný iba v móde *EXCLUDE* a v okamihu jeho vypršania je filtrovací mód opäť nastavený do stavu INCLUDE a všetky zdrojové záznamy s vypršaným časovačom už nie je naďalej potrebné uchovávať a sú teda

zmazané. Samotné nastavovanie a obnovovanie týchto časovačov je zabezpečené pri spracovávaní *Report* správ popísaných nižšie.

Periodické plánovanie *General Query* správy zostalo nezmenené. Rozdiel však nastáva pri spracovávaní prijatých Report správ. Zmenou oproti predchádzajúcej verzii je fakt, že koncové zariadenia pri prijatí report správy od inej koncovej stanice nezrušia svoj časovač ale naplánujú odpoveď podľa pravidiel spomenutých v predchádzajúcej časti. Spracovanie reportov a ich výsledné akcie sú popísané v nasledujúcej tabuľke, kde veľké písmeno „A“ označuje množinu zdrojových záznamov, „ =“ označuje nastavenie časovača na zadanú hodnotu, „A-B“ označuje rozdiel množín, „A*B“ prienik množín, „A+B“ zjednotenie množín, GMI je hodnota *Group Membership Interval* a pri akcii „Odošli Query(G,A)“ alebo „Odošli Query(G)“ ,hodnota „ G“ označuje skupinovú adresu a množina „A“ zoznam zdrojov. Pri akcii „Zmaž(A)“ je zmazaná množina „A“ zdrojových záznamov. Pri stave EXLUDE(X,Y) je hodnotou „X“ označovaná množina zdrojových záznamov s bežiacim časovačom a hodnotou „Y“ množina s vypršaným časovačom.

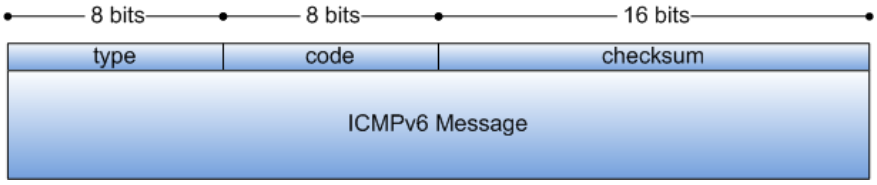
Stav smerovača	Prijatý report	Nový stav smerovača	Vykonané akcie
INCLUDE(A)	IS_IN(B)	INCLUDE(A+B)	B = GMI
INCLUDE(A)	IS_EX(B)	EXCLUDE(A*B, B-A)	(B-A) = 0 Zmazanie(A-B) Skupinový časovač = GMI
EXCLUDE(X,Y)	IS_IN(A)	EXCLUDE(X+A, Y-A)	(A) = GMI
EXCLUDE(X,Y)	IS_EX(A)	EXCLUDE(A-Y, Y*A)	(A-X-Y) = GMI Zmazanie(X-A) Zmazanie(Y-A) Skupinový časovač = GMI
INCLUDE(A)	ALLOW(B)	INCLUDE(A+B)	(B) = GMI
INCLUDE(A)	BLOCK(B)	INCLUDE(A)	Odošli Query(G,A*B)
INCLUDE(A)	TO_EX(B)	EXCLUDE(A*B, B-A)	(B-A) = 0 Zmazanie(A-B) Odošli Query(G,A*B) Skupinový časovač = GMI
INCLUDE(A)	TO_IN(B)	INCLUDE(A+B)	(B) = GMI Odošli Query(G,A-B)
EXCLUDE(X,Y)	ALLOW(A)	EXCLUDE(X+A, Y-A)	(A) = GMI
EXCLUDE(X,Y)	BLOCK(A)	EXCLUDE(X+(A-Y), Y)	(A-X-Y) = Skupinový časovač Odošli Query(G,A-Y)

Stav smerovača	Prijatý report	Nový stav smerovača	Vykonané akcie
EXCLUDE(X,Y)	TO_EX(A)	EXCLUDE(A-Y, Y*A)	(A-X-Y) = Skupinový časovač Zmazanie(X-A) Zmazanie(Y-A) Odošli Query(G,A-Y) Skupinový časovač = GMI
EXCLUDE(X,Y)	TO_IN(A)	EXCLUDE(X+A, Y-A)	(A) = GMI Odošli Query(G,X-A) Odošli Query(G)

Tabuľka 2-6: Akcie pri obdržaní report správy.

2.5 Prihlasovanie do skupín v IPv6

K zisťovaniu príjemcov vysielania slúžil v protokole IPv4 protokol IGMP [10]. Pre IPv6 však tento protokol zmenil názov na MLD. Jeho princípy však zostali nezmenené. Jeho autori prehlasujú, že sa jedná o preklad IGMP do IPv6. MLD je momentálne v dvoch verziách. MLDv1definovaný v RFC 2710 [11] je postavený na IGMPv2, a najnovšia verzia MLDv2 definovaná v RFC 3810 [12], ktorá je postavená na protokole IGMP verzie 3. Oba tieto protokoly pracujú analogicky ako ich predchodcovia v IPv4. Hlavný rozdiel však nastáva v samotnom postavení protokolu. IGMP v IPv4, bol samostatným protokolom zapúzdreným v protokole IP. MLD je však implementovaný ako subprotokol v ICMPv6. Formát správy teda vychádza práve z protokolu ICMPv6 a má tvar:



Obrázok 2-19 Formát ICMPv6 správy

Tieto správy sú odosielané z *Link-Local* adresy daného rozhrania a ich *TTL* je nastavený na hodnotu 1. Typy ICMPv6 správ pre MLD verzie 1 a 2 sú popísané v nasledujúcej tabuľke:

Typ správy	MLDv1	MLDv2
Query	130	130
Report	131	143
Done	132	-

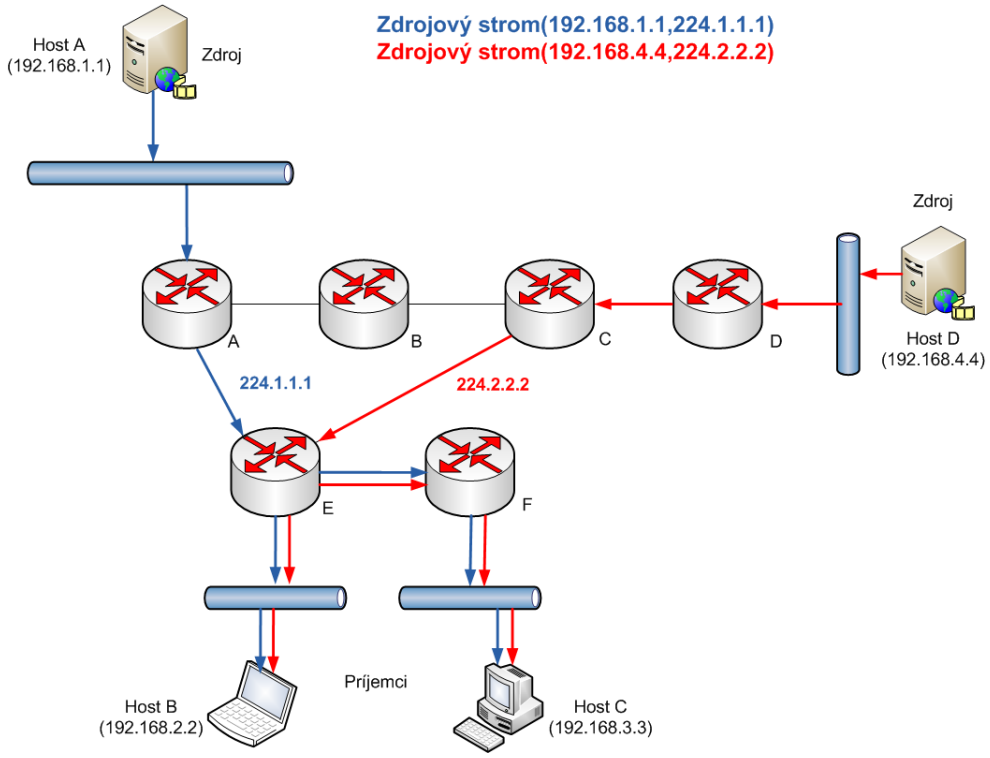
Tabuľka 2-4 Typy správ pre MLD

2.6 Distribučné stromy

Všetky multicastové smerovače vytvárajú distribučné stromy, vďaka ktorým môžu kontrolovať cestu, ktorou putujú pakety od zdroja k cieľu [13]. Tieto stromy delíme na dve kategórie: Zdrojové stromy a Zdieľané stromy.

2.6.1 Zdrojové stromy

Zdrojový strom sa často zvykne označovať ako strom s najkratšou cestou(shortest path tree). Je to teda najkratší strom od zdroju dát (koreň) po koncové stanice (listy). Príklad zdrojového stromu sa nachádza na obrázku 2-19.



Obrázok 2-20 Zdrojový strom

Notácia (**S**,**G**) v obrázku značí **S**:unicastovú adresu zdroju dát, a **G**:Multicastovú skupinu, ktorá dáta odoberá. Táto dvojica nám teda značí strom s najkratšou cestou a je použitá pre každý zdrojový strom v sieti. V prípade, že máme v sieti viacero zdrojov dát pre rôzne skupiny, pre každú z nich bude vytvorený vlastný zdrojový strom.

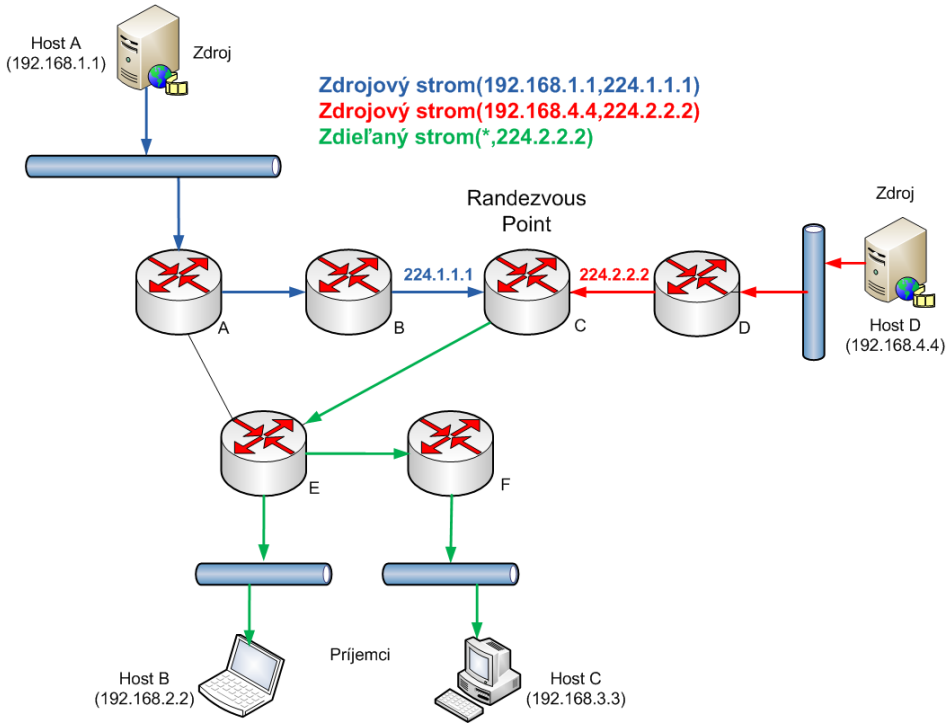
2.6.2 Zdieľané stromy

Na rozdiel od zdrojových stromov, v zdieľanom strome majú všetky multicastové skupiny jeden spoločný koreň nazývaný *randezvous point*(RP). Keďže pri zdieľanom strome nemáme presne určený zdroj dát, používame v notácii pre označenie stromu symbol „*“, ktorý nám označuje akýkoľvek

zdroj. Dáta sú teda posielané od zdroja do RP cez zdrojový strom a odtiaľ následne putujú cez zdieľaný strom k odberateľom. V prípade, že sa odberateľ nachádza niekde na ceste medzi zdrojom a RP, je obslúžený priamo pomocou zdrojového stromu.

V prípade že smerovač zistí kratšiu cestu ako je cesta zdieľaným stromom, vytvorí zdrojový strom a od zdieľaného stromu sa odpojí.

Obrázok 2-20 znázorňuje zdieľaný strom s RP na smerovači C.



Obrázok 2-21 Zdieľaný strom

Oba typy stromov nám spĺňajú podmienku bezslučkovosti, ktorá vyplýva z algoritmu pre nájdenie najkratšej cesty. Dáta sú posielané iba aktívnym vetvám stromu na základe prihlasovania a odhlasovania užívateľov zo skupín. Smerovače si držia v pamäti informácie o každom ich strome.

Práve preto je vo väčších sieťach s veľkým množstvom skupín výhodné používať práve zdieľaný strom, ktorý nie je pamäťovo až tak náročný.

2.7 Multicastové smerovanie

V unicastovom smerovaní preposielame správu od zdroja k cieľu [7]. Každému smerovaču preto stačí poznať iba cieľovú adresu správy. Túto adresu vyhľadá v svojej smerovacej tabuľke a správu prepošle správnym rozhraním ďalej k cieľu.

V multicastovom smerovaní je však situácia iná. Správu preposielame od zdroja skupine príjemcov reprezentovanej ich multicastovou skupinovou adresou. Multicastový smerovač teda musí

vedieť správne rozhodnúť, ktorá cesta je tzv. *Downstream*(smer od smerovača k cieľu), a ktorá naopak *Upstream*(od smerovača k zdroju dát). V prípade viacerých *Downstream* ciest potom smerovač rozosiela kópie správy smerom preč od zdroju dát. Táto metóda sa volá *Reverse Path Forwarding.*

2.7.1 Reverse Path Forwarding (RPF)

RPF je metóda používaná v multicastovom smerovaní pre doručovanie multicastových správ smerom dolu od zdroja k listom distribučného stromu. Po prijatí správy na smerovači je správa preposlaná ďalej iba v prípade, že bola prijatá z *Upstream* cesty. Pre zistenie *Upstream* a *Downsream* ciest používa smerovač unicastovú smerovaciu tabuľku, v ktorej vyhľadá zdroj dát a rozhranie, na ktorom sa daný zdroj nachádza. Ak bola správa prijatá z *Upstream* cesty, je preposlaná na *Downstream* cesty. V opačnom prípade ak bola správa prijatá na z *Downstream* cesty, bude zahodená. Vďaka tomuto mechanizmu je následne zabezpečená bezslučkovosť multicastového smerovania.

2.7.2 Protocol Independent Multicast (PIM)

Protocol Independent Multicast, ako už jeho názov napovedá, nie je závislý na zvolenom smerovacom protokole. PIM používa unicastové smerovacie informácie, na základe ktorých smeruje multicastové toky. PIM však na rozdiel od unicastových smerovacích protokolov nevytvára žiadne smerovacie tabuľky a taktiež neposiela žiadne smerovacie informácie ostatným smerovačom.

PIM Dense Mode

PIM DM používa tzv. *Push Model* pre šírenie multicastového vysielania do každej časti siete. PIM DM najprv zaplaví všetkých užívateľov siete multicastovým vysielaním. Smerovače, ktoré nemajú žiadnych susedov v *Downstream* ceste, toto vysielanie následne odrežú. Mechanizmus záplavy a odrezania je následne opakovaný každé tri minúty. V prípade, že na odrezanej vetve pribudne odberateľ, je možnosť vetvu opätovne pripojiť k odberu dát pomocou *Graft* správy. Tento spôsob smerovania však podporuje iba smerovanie cez zdrojové stromy.

PIM Sparse Mode

PIM SM používa pre doručovanie multicastového vysielania tzv. *Pull Model*. Pri tejto metóde je vysielanie smerované iba do sietí s aktívnymi odberateľmi, ktorý si tieto dáta vyžiadali. PIM SM prednostne využíva pre doručovanie multicastového vysielania zdieľané stromy. Vysielanie teda začne putovať od zdroju dát dolu zdieľaným stromom. V prípade že smerovač na ceste od zdroja k cieľu zistí lepšiu(kratšiu) cestu, designated smerovač odošle *Join* správu smerom k zdroju a zmení cestu na novú.

Nakoľko PIM SM prednostne využíva zdieľané stromy, v ktorých figuruje RP. Je potrebné tento bod administratívne nastaviť tak, aby sa nachádzal, čo možno najbližšie k zdrojom dát

a rovnako tak k cieľom. Pri smerovaní teda vysielanie smeruje od zdroja k RP a odtiaľ následne putuje k odberateľom.

V prípade, že cesta zdieľaným stromom cez RP nie je ideálna, smerovače ju môžu dynamicky zmeniť na odpovedajúci zdrojový strom.

Sparse Dense Mode

Spoločnosť Cisco[2] vyvinula alternatívu k používaniu iba Dense Mode alebo Sparse Mode. Dáva tak administrátorovi siete možnosť vybrať Sparse-Dense mode, kde fungovanie spočíva nasledovne. Ak má smerovač pre danú skupinu informácie o RP, bude táto skupina spracovaná ako pri SM a použije pre smerovanie tejto skupiny zdieľaný strom. V opačnom prípade, bude skupina smerovaná ako pri SM a teda cez zdrojový strom.

PIM Bi-Directional

Bi-Directional PIM je ďalšou možnosťou doručovania multicastového vysielania. Tento mód je odvodený od Sparse mode. Rozdiel však nastáva pri doručovaní od zdroja dát k RP, kde narozdiel od Sparse mode, Bi-Directional využíva zdieľaný strom, ktorý umožňuje odosielanie dát v oboch smeroch. Tento druh doručovania je využívaný hlavne v aplikáciách, kde je pre jednu multicastovú skupinu viacero zdrojov dát [14].

PIM Source Specific Multicast(SSM)

Poslednou možnosťou doručovania je metóda Source Specific Multicast, pri ktorej sú využívané iba zdrojové stromy. Mód je využívaný spolu s protokolom IGMP verzie 3, pri ktorej si užívateľ môže vybrať zdroj dát pre požadovanú multicastovú skupinu. Pre identifikáciu tokov je použitá dvojica (S,G) nazývana *channel*, kde S je zdrojová adresa a G je adresa multicastovej skupiny. Spoločnosť IANA pre SSM vyhradila rozsah adries 232.0.0.0/8 pre IPv4 a prefix FF3x::/32 pre IPv6. Výhodou pri využívaní SSM je fakt, že skupinová adresa môže byť použitá v kombinácii s rôznymi zdrojovými adresami, čo zabezpečuje unikátnosť *channel*-ov [15].

[2] Cisco Systems, Inc. http://www.cisco.com

3 Multicast na zariadeniach Cisco

Tretia kapitola sa zaoberá možnosťami multicastového smerovania na smerovačoch Cisco. Sú v nej vypísané príkazy pre CLI smerovača potrebné pre základné nastavenie mulicastového smerovania.

3.1 IPv4 Multicast

V tejto kapitole uvediem základné príkazy pre konfiguráciu IP multicastu na smerovačoch Cisco s verziou IOS 12.2 [14].

Ako prvé je potrebné v globálnom konfiguračnom režime povoliť multicastové smerovanie pomocou príkazu:

```
Router(config)# ip multicast-routing
```

Ďalej je potrebné povoliť na rozhraniach protokol PIM, ktorý nám súčasne povolí IGMPv2 pre dané rozhranie. Pri nastavovaní je potrebné vybrať režim, v ktorom chcem aby rozhranie pracovalo. Na výber máme *Dense-Mode*, *Sparse-Mode* Cisco proprietárny *Sparse-Dense-Mode, Bi-Directional* alebo *Source Specific Multicast.*

Dnese Mode:

```
Router(config-if)# ip pim dense-mode
```

Sparse Mode:

```
Router(config-if)# ip pim sparse-mode
```

Sparse-Dense Mode:

```
Router(config-if)# ip pim sparse-dense-mode
```

Bi-Directinal:

Pre spustenie Bi-Directional modu je potrebné zadať v globálnom konfiguračnom režime príkaz [15]:

```
Router(config-if)# ip pim sparse-dense-mode
```

Následne je potrebné vybrať metódu, ktorou sa bude vykonávať mapovanie skupín na RP.

Na výber máme tri možnosti:

```
Router(config)# ip pim rp-address
rp-address[access-list] [override]bidir
```

```
Router(config)# ip pim rp-candidate type
number[group-list access-list]bidir
```

```
Router(config)# ip pim send-rp-announce type number scope
ttl-value[group-list access-list] [interval seconds] bidir
```

Source Specific Multicast:

Nastavenie protokolu PIM pre použitie SSM je možné nastaviť v štyroch krokoch pomocou nasledujúcich príkazov [16]:

```
Router(config)# ip pim ssm [default |range access-list]
Router(config)#interface type number
Router(config-if)#ip pim {sparse-mode | sparse-dense-mode}
Router(config-if)#ip igmp version 3
```

Po aktivovaní protokolu PIM na rozhraniach, máme možnosť upraviť rôzne jeho nastavenia, ktoré sú popísané napríklad v [14] .

Nasledujúcim krokom je priradenie smerovača do multicastovej skupiny. Pre prihlásenie smerovača do skupiny je potrebné zadať na rozhraní príkaz:

```
Router(config-if)# ip igmp join-group group-address
```

Skupinu však môžeme pridať pre rozhranie aj staticky. Znamená to, že vysielanie pre danú skupinu bude na dané rozhranie vysielané stále bez ohľadu na to, či o toto vysielanie niekto záujem má alebo nie. Statickú skupinu pridáme na zvolenom rozhraní pomocou príkazu:

```
Router(config-if)# ip igmp static-group group-address
```

Pri povolení PIM na rozhraní je protokol IGMP spustený vo verzii 2. Pre zmenu verzie protokolu na rozhraní je potrebné zadať:

```
Router(config-if)# ip igmp version {3 | 2 | 1}
```

Ďalšou možnou voľbou je možnosť filtrovania preposielania jednotlivých multicastových skupín na rozhrania pomocou ACL. Pre ich aktivovanie je potrebné mať vytvorený Access-List a zadať na rozhraní príkaz:

```
Router(config-if)# ip igmp access-group access-list
```

Pre protokol IGMP je ešte možné nastaviť na rozhraní rôzne časovače ako napríklad:

```
Router(config-if)# ip igmp query-interval seconds
Router(config-if)# ip igmp querier-timeout seconds
Router(config-if)# ip igmp query-max-response-time seconds
Router(config-if)# ip igmp last-member-query-interval interval
```

Ďalšie rôzne nastavenia pre IP multicast je možné nájsť napríklad na [14].

3.2 IPv6 Multicast

V tejto kapitole sú uvedené základné konfiguračné kroky pre multicast na IPv6 na smerovačoch Cisco s IOS verzie 12.4 [15].

Podobne ako v predchádzajúcej podkapitole, je potrebné v globálnom konfiguračnom režime povoliť IPv6 multicastové smerovanie pomocou príkazu:

```
Router(config)# ipv6 multicast-routing
```

Spolu s IPv6 multicatovým smerovaním sa nám hore uvedeným príkazom spustí aj PIM-SM a protokol pre prihlasovanie do skupín MLD vo verzii 2.

Rovnako ako pri IPv4 je následne potrebné prihlásenie sa do skupiny pomocou príkazu:

```
Router(config-if)# ipv6 mld join-group [group-address]
[include | exclude] {source-address |source-list [acl]}
```

V IPv6 je zachovaná možnosť pridať skupinu na rozhranie staticky pomocou príkazu:

```
Router(config-if)# ipv6 mld static-group [group-address]
[include | exclude] {source-address |source-list [acl]}
```

Taktiež je možné filtrovať preposielanie multicastových skupín pomocou ACL. Príkaz pre filtrovanie na rozhraní je nasledovný:

```
Router(config-if)# ipv6 mld access-group access-list-name
```

Dalšou možnosťou je nastavenie rôznych časovačov pomocou príkazov na rozhraní:

```
Router(config-if)# ipv6 mld query-max-response-time seconds
Router(config-if)# ipv6 mld query-timeout seconds
Router(config-if)# ipv6 mld query-interval seconds
```

Ďalšie možnosti nastavení pre IPv6 multicast sú popísané napríklad na [16].

4 OMNeT++

Štvrtá kapitola tejto práce je venovaná simulačnému nástroju OMNeT++[3], jeho knižnici INET[4] a návrhu implementácie nových modulov pre túto knižnicu.

4.1 OMNeT++

OMNeT++ je objektovo orientovaný diskrétny simulátor počítačových sietí [17]. Vďaka svojej modularite je ho však možné použiť aj pri riešení rôznych iných problémov. OMNeT++ ako taký nie je konkrétnym simulátorom. Obsahuje však infraštruktúru, prostredie a nástroje pre vytváranie simulácií.

Modely sú vytvárané z menších komponentov, ktoré sú navzájom poprepájané do rôznych modulov. Následne sú tieto modely prepojené pomocou brán a komunikujú navzájom pomocou zasielania správ. Na najnižšej úrovni sú položené takzvané jednoduché modely, ktoré sú základným stavebným kameňom väčších modelov. Tieto komponenty sú naprogramovane v programovacom jazyku C++.

Simulácie v OMNeT++ je možné spúšťať v rôznych režimoch ako napríklad v grafickom, kde sú animované jednotlivé kroky simulácie. Simulácie je však taktiež možné spustiť aj v príkazovom riadku.

OMNeT++ je možné spustiť na takmer všetkých najpoužívanejších operačných systémoch ako Linux, Mac OS/X a Microsoft Windows.

Pre popis jednotlivých modelov je využívaný jazyk nazvaný NED. Pomocou tohto jazyka definujeme štruktúru jednotlivých modelov. Definujeme im rôzne parametre a taktiež ich pomocou brán navzájom prepájame.

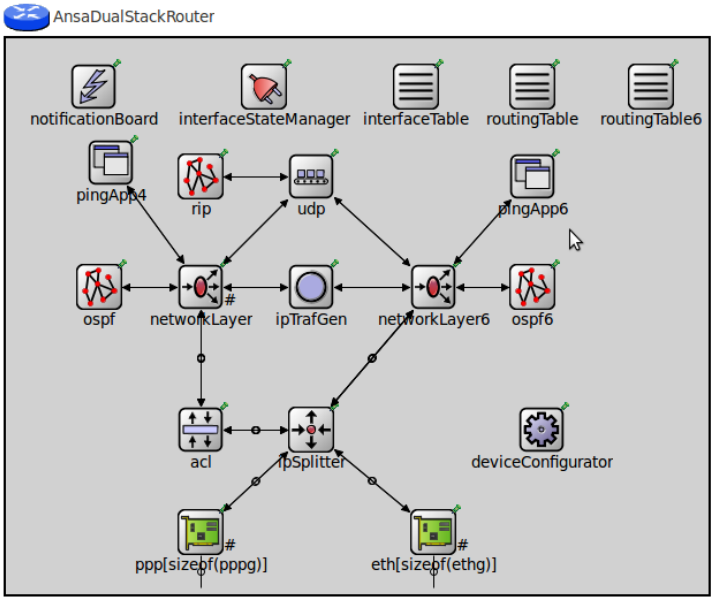
4.2 INET Framework

Framework INET(v aktuálnej verzii INET-2.1.0 pre OMNeT++ 4.2 a novšie) obsahuje implementácie protokolov ako napríklad IPv4, IPv6, TCP, SCTP, UDP a iné. Obsahuje taktiež MPLS model s LDP signalizáciou. Na linkovej vrstve obsahuje implementáciu modelov pre PPP, Ethernet a protokol 802.11. Pomocou autokonfigurátorov je pre modely možné nastaviť statické smerovanie, alebo je možné použiť niektorý z implementovaných smerovacích protokolov. Implementácia frameworku INET obsahuje implementáciu multicastového protokolu IGMPv2. V implemetácii však stále chýba

[3] OMNeT++ http://www.omnetpp.org
[4] INET http://inet.omnetpp.org

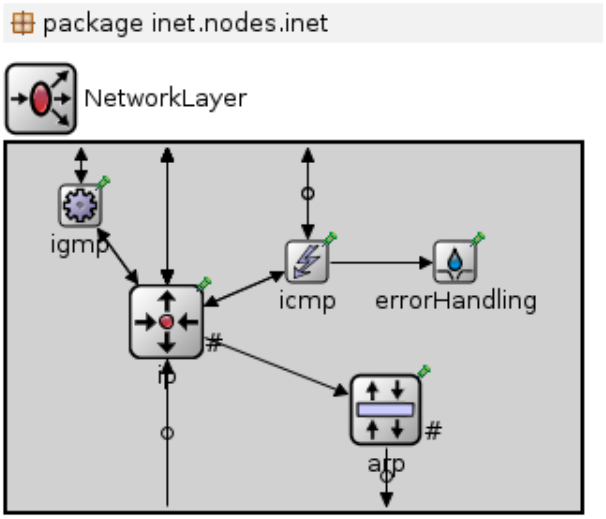
novšia verzia tohto protokolu s podporou filtrovania zdrojov a taktiež protokol MLD podporujúci multicast v IPv6 [21].

V rámci projektu ANSA[5] vznikli implementácie viacerých nových modulov, ktoré v pôvodnej verzii INET-u chýbali. Medzi nimi bol vytvorený aj nový model *ANSA dual stack smerovača* [22] zobrazeného na obrázku nižšie.



Obrázok 4-1 ANSA dual stack smerovač

Tento smerovač na svojej sieťovej vrstve(*NetworkLayer*) využíva implementáciu protokolu IGMPv2 pre framework INET. Samotná pôvodná a zjednodušená implementácia protokolu je popísaná v bakalárskej práci Petra Mateleška [21]. Nakoľko však táto verzia implementovala iba časť protokolu pre smerovač a logika klientských staníc bola vypustená z implementácie bola vývojármi INET-u vytvorená nová verzia protokolu IGMPv2.



Obrázok 4-2 IPv4 Network Layer

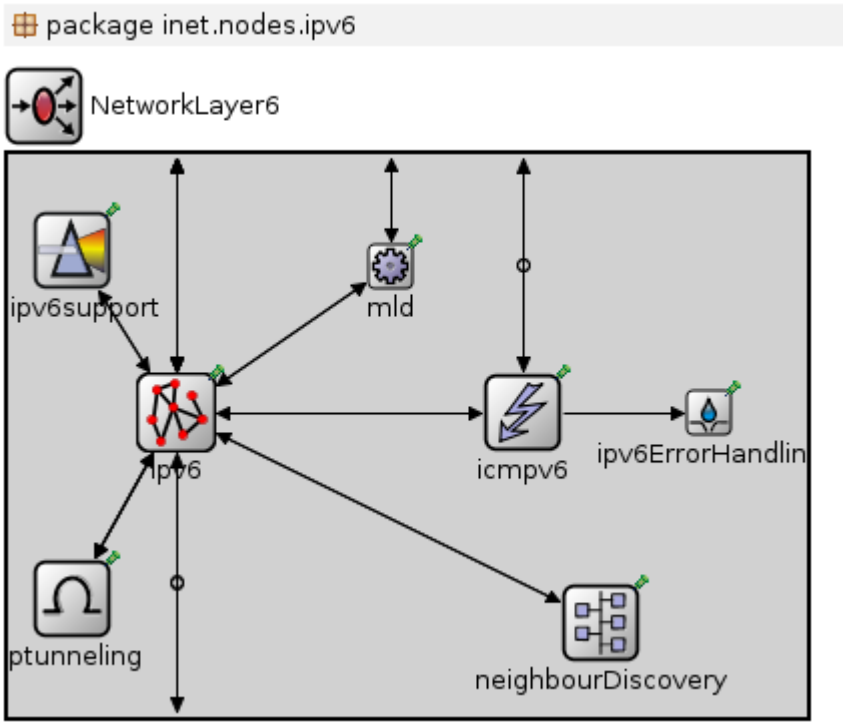
[5] ANSA Project http://nes.fit.vutbr.cz/ansa

4.3 Návrh implementácie

Prvou úlohou pred samotným návrhom nových verzií protokolov bolo otestovanie aktuálnej verzie IGMPv2. V prípade, že by obsahovala akékoľvek chyby je nutné ich opraviť pred implementáciou ostatných protokolov, ktoré budú pre prehľadnosť vychádzať práve z tejto verzie. Na otestovanie protokolu som použil testovaciu simuláciu základného fungovania protokolu, na ktorej sa dala overiť voľba *Querier*-a, odosielanie periodických *General Query* správ a prihlasovanie koncových staníc do skupín. Z výsledku testovania, ktoré je popísané v kapitole 6 a analýzy zdrojového kódu som zhodnotil, že protokol pracuje tak ako je popísaný v RFC2236 [9] a ja som sa mohol pustiť do návrhu nových protokolov.

4.3.1 MLDv1

Pri návrhu protokolu MLDv1, budem vo veľkej miere vychádzať z implementácie protokolu IGMPv2, nakoľko protokol MLD je jeho prekladom do IPv6. MLD je súčasťou ICMPv6, avšak pre lepšiu prehľadnosť a analógiu k IGMP som sa rozhodol modul MLD umiestniť priamo do sieťovej vrstvy NetworkLayer6, tak ako je to zobrazené na nasledujúcom obrázku. Modul bude cez brány pripojený k modulu Ipv6.



Obrázok 4-3 Sieťová vrstva IPv6

Jadrom modulu budú metódy pre prijatie, odoslanie a spracovanie *Query* a *Report* správ. Pre komunikáciu týchto modulov bude potrebné vytvoriť nový typ správy, ktorý bude obsahovať všetky potrebne polia zadefinovane v RFC2710. Druhou dôležitou súčasťou budú rôzne časovače,

ktoré budú zabezpečovať periodické dotazy v smere od smerovača ku koncovým zariadeniam, a taktiež spúšťať odložené odpovede na *Query* správy.

Keďže informácie si zariadenia udržujú pre každé svoje multicastové rozhranie, bude potrebné vytvoriť vhodnú štruktúru na uchovávanie všetkých dôležitých informácií o skupinách a ich stavoch.

Modul na prijímanie správ bude podľa typu rozlišovať druh prijatej správy a na základe toho správu prepošle správnemu modulu na spracovanie. V prípade prijatia akejkoľvek inej správy bude táto správa zahodená tak, ako je to v popise protokolu v jeho RFC [11].

Modul pre spracovanie *Query* správ bude podľa skupinovej IP adresy rozdeľovať *Query* na typ *General* alebo *Group specific query*. V prípade, že modul obdrží skupinovú adresu nastavenú na hodnotu " :: 0", bude sa jednať o *General query* a modul pre každú prihlásenú skupinu vytvorí vlastný časovač. V prípade, že je skupinová adresa nastavená na adresu prihlásenej multicastovej skupiny, modul vytvorí časovač pre túto skupinu. V prípade, že požadovaná skupina nieje na rozhraní prihlásená, modul bude túto správu ignorovať. Časovač je po vytvorení nastavený na náhodnú hodnotu z intervalu $\langle 0, Maximum\ Response\ Delay \rangle$. Prvá koncová stanica, ktorej časovač vyprší vytvorí Report správu, pre danú skupinu a tento report odošle. Následné si koncová stanica nastaví na rozhraní pre danú skupinu príznak *flag* na hodnotu *true*, čo bude znamenať, že koncová stanica ohlasovala skupinu ako posledná. Ostatné koncové stanice s doposiaľ nevypršaným časovačom po prijatí Report správy tieto časovače zrušia a nastavia si príznak *flag* na hodnotu *false*.

Ak je *Query* správa spracovávaná v module smerovača, dôjde k voľbe *Querier* smerovača. Modul porovná svoju IP adresu so zdrojovou IP adresou prijatej *Query* správy. V prípade, že adresa z prijatej správy je nižšia ako adresa smerovača, modul nastaví svoj stav na hodnotu *NonQuerier* a spustí časovač, po ktorého vypršaní opäť prejde do stavu *Querier*. V prípade prijatia nasledujúcej *Query* správy v sa časovač znovu obnoví na počiatočnú hodnotu.

Po prijatí reportu smerovač skontroluje skupinovú adresu obdržanú z reportu v zozname skupín, ktorý má uloženú pre dané rozhranie. Ak bola prijatá skupina, o ktorej smerovač doposiaľ nemal žiadne informácie, pridá si novú skupinu do svojho zoznamu a nastaví časovač. V prípade prijatia známej skupiny sa iba opäť nastaví časovač na počiatočnú hodnotu. Po vypršaní tohto časovača smerovač odošle *Group specific query* správu aby zistil záujem koncových staníc o danú skupinu a v prípade nezáujmu skupinu zmazal.

Poslednou možnosťou je prijatie *Done* správy. Túto správu vysiela koncové zariadenie v okamihu, kedy stratí o vysielanie skupiny záujem a má nastavený príznak *flag* na *true*. Smerovač po prijatí tejto správy opäť rozošle *Group Specific query* a nastaví časovač. Po vypršaní tohto časovača smerovač usúdi, že o vysielanie dotazovanej skupiny už nikto nemá záujem a skupinu zmaže. V prípade, že smerovač obdrží odpoveď na odoslanú *Query*, obnoví hodnotu svojho časovača a skupinu naďalej ponechá v zozname aktívnych multicastových adries.

4.3.2 IGMPv3

Protokol IGMP vo verzii 3 bude taktiež rovnako ako MLD v základe vychádzať z modulu IGMPv2. Kvôli odlišnému správaniu protokolov však budú zachované iba niektoré komponenty.

Verzia 3 používa dva úplne nové formáty správ. Hlavnou zmenou však je možnosť dotazovať sa na zdrojové adresy. Pre zvýšenie prehľadnosti správa *Version 3 Membership Report* pozostáva z typu správy, počtu skupinových záznamov a následne samotnými záznamami o jednotlivých skupinách. Každý skupinový záznam nesie potrebné informácie o skupine ako je jej multicastová adresa, typ záznamu a zoznam zdrojových adries. Vďaka zvýšenej zložitosti protokolu bude taktiež potrebné pridať nové časovače a upraviť štruktúry na rozhraniach tak, aby dokázali ukladať všetky potrebné údaje. Opäť budú potrebné moduly pre odosielanie, prijímanie a spracovanie správ.

Novinkou však bude modul spracovávajúci zmenu na rozhraní u koncovej stanice, ktorá si po prihlásení môže pomocou filtrov voliť zdroje, ktoré chce alebo nechce odoberať. Tento modul teda bude sledovať zmenu na rozhraní koncovej stanice a po vyhodnotení zmeny okamžite, bez nutnosti prijatia *Query*, odošle nový report, v ktorom oznámi zmenu. Ostatné moduly budú pracovať na podobnom princípe ako v predchádzajúcich verziách. Modul na prijímanie opäť podľa typu správu prepošle odpovedajúcemu modulu, ktorý správu spracuje.

Modul pre spracovanie *Query* správ skontroluje podľa multicastovej adresy, či sa jedná o *General Query*, *Group Specific Query* alebo *Group-and-Source Specific Query*. Hodnotu *Max Response Code* prijatú v správe využije na výpočet hodnoty *Max Response Time*, ktorá slúži nanastavenie časovača pre odpoveď na *Query*. Výpočet hodnoty sa vykoná podľa vzorca uvedeného v RFC3376. Následne podľa pravidiel uvedených v kapitole 2.4.3 rozhodne akú metódu odpovede zvolí. Konkrétne teda, či sa naplánuje oneskorená odpoveď na *General Query*, *Group Specific Query*, *Group-and-Source Specific Query* alebo sa využije odpoveď naplánovaná skôr. Tieto rozhodovacie pravidlá budú vyhodnotené na základe prijatej *Query* a bežiacich časovačov. V prípade ak *Query* obdrží smerovač, prebehne voľba *Querier* smerovača rovnakým spôsobom ako v predchádzajúcej verzii.

Ďalším dôležitým modulom bude modul spracúvajúci prijaté *Report* správy. Nakoľko v jednej report správe môže byť obsiahnutých viac skupinových záznamov, musí modul prejsť celou správou a spracovať jednotlivo každý skupinový záznam. Pri prijatí skupiny, ktorú smerovač doposiaľ nemá pre dané rozhranie vo svojom zozname, je pre túto skupinu vytvorený nový záznam. Záznamy budú ďalej spracované podľa pravidiel uvedených v kapitole 2.4.3. Každé pravidlo vychádza z aktuálneho stavu skupiny(filtrovací mód, a zoznam zdrojových záznamov) a následne vykonáva akcie na základe informácii uvedených v *Report* správe. Výslednými akciami môže byť nastavenie alebo upravenie časovačov, zmazanie zdrojových záznamov alebo odoslanie *Query*. Novinkou vo verzii 3 sú filtrovacie módy zdrojov. Vo verzii 3 je pridaný časovač pre zdrojový záznam. Tieto časovače sú obnovované a nastavované pri prijatí *Report* správ podľa určitých pravidiel. V prípade vypršania

časovača vo filtrovacom móde nastavenom na *Exclude*, sa zdroj ponecháva až do doby, kedy skupina opäť prejde do stavu *Include*. Ak je filtrovací mód skupiny v stave *Include*, zdroj sa zmaže.

Pri prijatí akejkoľvek správy s filtrovacím módom nastaveným do stavu *Exclude*, sa zmení filter pre celu skupinu taktiež na stav *Exclude*. Musí preto existovať mechanizmus na navrátenie skupiny opäť do stavu *Include*. Na návrat skupiny do stavu *Include* bude slúžiť skupinový časovač, kde po vypršaní časovača bude filtrovací mód skupiny opäť nastavený do stavu *Include*. Spolu s touto zmenou sa vykonajú aj zmeny v zdrojových záznamoch. Zdrojové adresy, ktoré budú mať po prechode do módu *Include* vypršaný časovač, nieje naďalej potrebné preposielať a budú teda zmazané. V prípade, že nezostane žiaden zdrojový záznam s bežiacim časovačom, bude na rozhraní zmazaná celá multicastová skupina

Pri prijatí reportu u koncového zariadenia sa naplánované časovače nerušia tak, ako tomu bolo u verzie 2. Keďže jednotlivé *Query* správy sú orientované na konkrétne skupiny a konkrétne zdroje bolo by toto správanie nežiadúce.

4.3.3 MLDv2

Ako už bolo spomenuté v kapitole 2.5, pri protokole MLDv2 sa jedná o preklad protokolu IGMPv3 do IPv6. Implementácia bude teda logicky zhodná s IGMPv3 a nieje potrebné opäť popisovať jej návrh. V tejto kapitole preto popíšem iba zmeny v názvoch správ používaných v MLDv2 a ich hodnôt. Oproti protokolu MLD bola z implementácie odstránená správa *Multicast Listener Done*. V implemntácii teda zostanú opäť iba dva druhy správ, ktorými sú *Multicast Listener Query* a *Multicast Listener Report.*

4.3.4 Vizualizácia distribučných stromov

Multicastový strom je možné rozdeliť na dve časti. Prvou z nich je časť od koncovej stanice, k jej smerovaču, cez ktorý má koncová stanica sprístupnený multicastový obsah. Druhou časťou stromu je následná cesta od smerovača za pomoci protokolu PIM k zdroju dát pomocou zdieľaného alebo zdrojového stromu. Nakoľko sa moja práca zaoberá protokolmi IGMP a MLD, budem vyzualizovať práve prvú časť. Táto časť stromu je vytváraná pri voľbe Querier smerovača.

Výstupom tejto vizualizácie teda bude orientovaný graf zobrazujúci koncové stanice a ich *Querier* smerovač. Pre možnosť vytvorenia grafu bude potrebné aby si každé rozhranie na koncovej stanici pamätalo IP adresu *Querier* smerovača. Pre dosiahnutie tohto stavu teda bude potrebné pridať do štruktúry pre rozhranie koncovej stanice atribút *Querier Router Address.*

4.3.5 Načítanie konfigurácie z XML

Rovnako dôležitá ako samotné modely protkolov IGMP A MLD je aj možnosť ich nastavenia. V rámci projektu ANSA sa pre nastavovanie zariadení v simulácii používa modul *DeviceConfigurator*. Tento modul parsuje dokument XML, z ktorého pre jednotlivé zariadenia identifikované pomocou parametru *id* získava počiatočnú konfiguráciu. Pomocou toho modulu je možné nastaviť zariadeniam napríklad smerovacie informácie, nastaviť pre rozhrania ich IP adresu, masku a bránu, a rovnako tak je možné nastavovať rôzne parametre napríklad smerovacím protokolom.

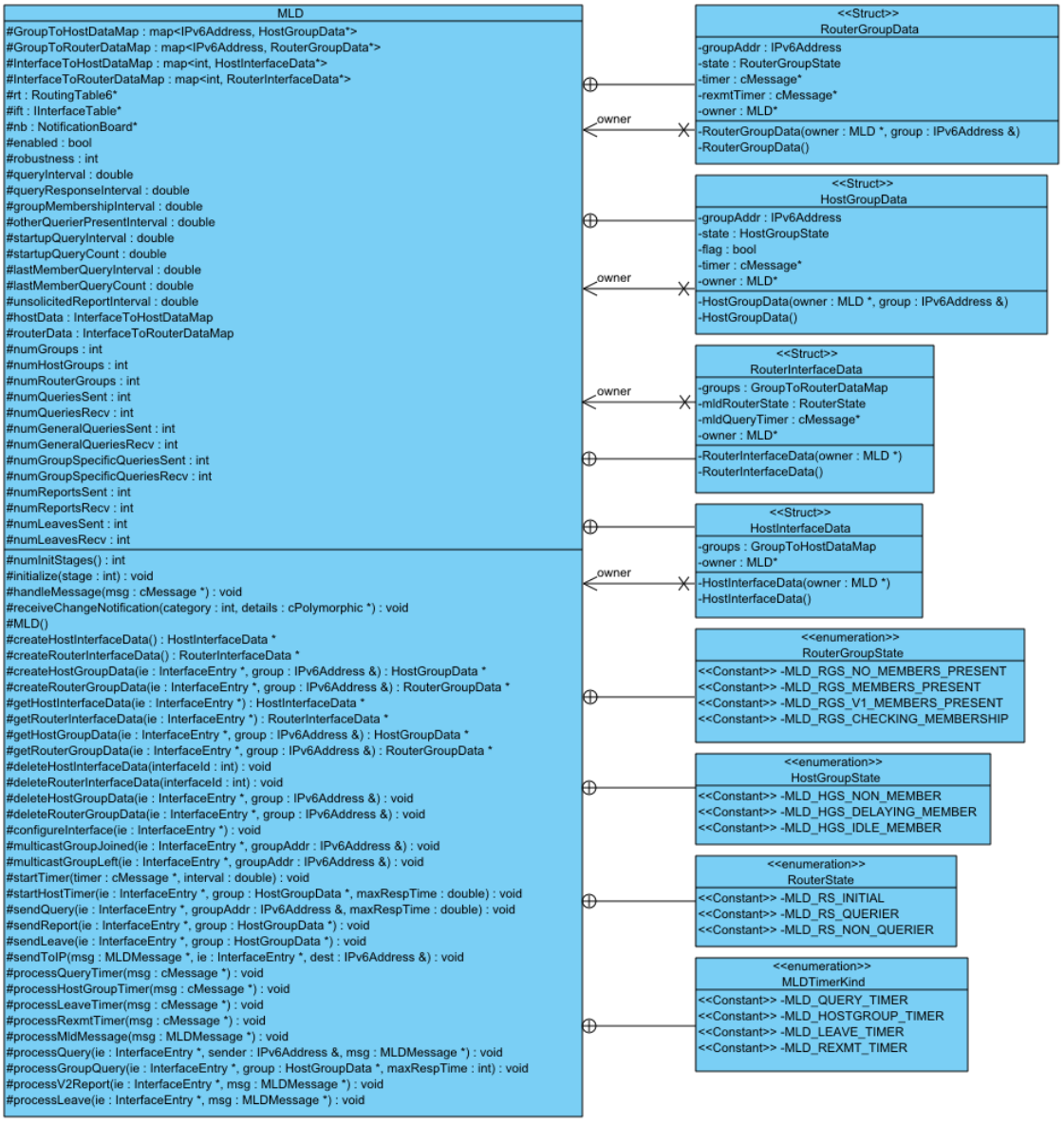
V tomto module som sa teda rozhodol vytvoriť novú položku, ktorá bude pre každé rozhranie umožňovať nastaviť multicastové adresy, ktoré chceme rozhraniu priradiť. V súbore XML s konfiguráciou teda bude potrebné pridať každému multicastovému rozhraniu nový obaľovcí tag s názvom MCastGroups, do ktorého budem vkladať jednotlivé multicastové adresy obalené tagmi $\langle MCastGroup \rangle multicastová\ adresa \langle /MCastGroup \rangle$. Týmto spôsobom bude možné pridávať multicastové adresy na rozhrania pre IP verzie 4. Keďže ale protokol MLD využíva IPv6, je potrebné doplniť možnosť načítavania multicastových skupín aj pre túto verziu. Bude preto potrebné vytvoriť druhý obaľovací tag nazvaný MCastGroups6, v ktorom budú jednotlivé záznamy multicastových adries obalené tagom $\langle MCastGroup6 \rangle multicastová\ adresa \langle /MCastGroup6 \rangle$. Tieto skupiny budú následne pri inicializácii simulácie pomocou XML Parseru prečítané z dokumentu XML s konfiguráciou. Všetky načítané skupiny budú pridané na určené rozhrania. a bude pre ne odoslaná *Report* správa *Join.*

5 Popis implementácie

V tejto kapitole popíšem najdôležitejšie časti samotnej implementácie, rozoberiem zložitejšie časti metód a popíšem jednotlivé štruktúry a tribúty. V kapitole taktiež budú zobrazené pri popisoch aj diagramy pre lepšiu vizuálnu orientáciu v moduloch.

5.1 MLDv1

V tejto časti sa budem zaoberať samotnou implementáciou navrhnutého modulu pre protokol MLD verzie 1. Popíšem typ správy používaný na komunikáciu medzi modulmi, štruktúry použité pre ukladanie dát, rôzne časovače a nakoniec aj funkcionalitu niektorých metód. Štruktúru implementovanej triedy zobrazuje obrazok s ClassDiagramom



Obrázok 5-1 ClassDiagram triedy MLD

5.1.1 Typy správ

Ako prvé je potrebné zaviesť nový typ správy potrebný pre korektnú komunikáciu jednotlivých modulov. Formát správy pre MLD sa pri rôznych typoch správ nemení. Jediná zmena nastáva pri čísle označujúcom typ správy. Správa bola preto navrhnutá nasledovne:

```
enum MLDType
{
    MLD_MULTICAST_LISTENER_QUERY = 130;
    MLD_MULTICAST_LISTENER_REPORT = 131;
    MLD_MULTICAST_LISTENER_DONE = 132;
}

packet MLDMessage
{
    int type enum(MLDType);
    int maxRespDelay;
    IPv6Address groupAddress;
}
```

Obrázok 5-2 definícia MLD správy

V atribúte *type* ju ukladaný typ správy, ktorý je vybraný z troch vymenovaných možností *MLDType*. Na základe tohto typu neskôr modul pre spracovanie správ dokáže rozhodnúť, ktorému modulu správu prepošle na spracovanie. Atribút *maxRespDelay* prenáša údaj, z ktorého je neskôr vypočítané pozdržanie odoslania odpovede na *Query* správu. Pole *groupAddress* obsahuje IPv6 adresu multicastovej skupiny, ku ktorej sa správa vzťahuje. Nakoľko v simulačnom prostredí nedochádza k chybám pri prenose, bolo z implemntácie vylúčené pole *Checksum* a taktiež nevyužívané polia *Code* a *Reserved*.

5.1.2 Časovače

Pre správne fungovanie protokolu je potrebné pre niektoré hodnoty použiť časovače. Časovače sú v prostredí OMNeT++ vytvárané pomocou správ, ktoré modul zasiela sám sebe. Pre zistenie tohto typu správy je možné použiť atribút správy s booleovskou hodnotu *true* alebo *false* s názvom *isSelfMessage*. Tento atribút je využívaný hlavne v metóde *handleMessage*, ktorá prijaté správy spracuje a prepošle. Časovače sú použité nielen pre udržovanie aktuálnych informácii ale slúžia aj na oneskorovanie odpovedí pri *Query* správach alebo periodické zasielanie *Query* správ. V module MLD využívam štyri druhy časovačov.

```
enum MLDTimerKind
{
    MLD_QUERY_TIMER,
    MLD_HOSTGROUP_TIMER,
    MLD_LEAVE_TIMER,
    MLD_REXMT_TIMER
};
```

Obrázok 5-3 MLD typy časovačov

- **MLD_QUERY_TIMER** je časovač, ktorý slúži na periodické odosielanie *General Query* správ. Tento časovač je po prvý krát nastavený v inicializácii modulu. Pri vypršaní časovača je správa prijatá metódou *handleMessage* a preposlaná do funkcie processQueryTimer, ktorá odošle General Query správu a opätovne naplánuje časovač *MLD_QUERY_TIMER* na hodnotu *queryResponseInterval*.
- **MLD_HOSTGROUP_TIMER** je využívaný pre oneskorovanie odoslania odpovde na strane koncových staníc. V metóde spracúvajúcej tento časovač sa však znovu neobnovuje a je obnovovaný v metóde spracovania *Query* správ.
- **MLD_LEAVE_TIMER** po svojom vypršaní spúšta metódu, v ktorej dôjde k odregistrovaniu multicastovej skupiny pre dané rozhranie a zmazaniu informácii o tejto skupine. Tento časovač je nastavený po obdržaní *Multicast Listener Done* správy.
- **MLD_REXMT_TIMER** je posledným zo štvorice časovačov. Jeho úlohou je udržiavať stav o jednotlivých skupinách. Tento časovač je pri obdržaní každej *Report* alebo *Done* správy pre danú multicastovú skupinu zastavený. V prípade, že dôjde k jeho vypršaniu, odošle smerovač *Group-Specific Query* pre danú skupinu opätovne nastaví tento časovač a zmení stav skupiny na *MLD_RGS_CHECKING_MEMBERSHIP*. Ak ani na túto *Query* správu neobdrží odpoveď usúdi, že o danú skupinu už nemá nikto záujem a skupinu zmaže.

5.1.3 Štruktúry

Pre udržiavanie všetkých informácii na každom rozhraní bolo potrebné vytvoriť štruktúru, ktorá by dokázala všetky tieto informácie uchovávať. Keďže smerovač aj koncové stanice potrebujú uchovávať rôzne informácie, bolo potrebné vytvoriť dve nové štruktúry pre každý typ zariadenia.

```
struct HostGroupData
{
    MLD *owner;
    IPv6Address groupAddr;
    HostGroupState state;
    bool flag;
    cMessage *timer;

    HostGroupData(MLD *owner, const IPv6Address &group);
    virtual ~HostGroupData();
};

struct RouterGroupData
{
    MLD *owner;
    IPv6Address groupAddr;
    RouterGroupState state;
    cMessage *timer;
    cMessage *rexmtTimer;
    //cMessage *v1HostTimer;

    RouterGroupData(MLD *owner, const IPv6Address &group);
    virtual ~RouterGroupData();
};
```

Obrázok 5-4 MLD Štruktúry na ukladani informácii o skupinách

Obe skupiny vo svojej štruktúre ukladajú adresu multicastovej skupiny, ktorá slúži na identifikáciu a pri odosielaní *Query* a *Report* správ. Obe zariadenia si v štruktúre udržiavajú informáciu o aktuálnom stave skupiny. Pri smerovačoch je to napríklad stav *MLD_RGS_MEMBERS_PRESENT*, ktorý značí, že medzi koncovými stanicami existuje záujemca o vysielanie tejto multicastovej skupiny. Názvy týchto stavov sú dostatočne zrozumiteľné a preto nieje potrebný ich opis.

Pri koncových staniciach je v štruktúre navyše príznak *flag*. Tento príznak označuje, že dané koncové zariadenie bolo posledné, ktoré odpovedalo na query. Tento príznak je dôležitý hlavne pri opúštaní skupiny. V prípade, že je nastavený na hodnotu *true*, musí koncová stanica pred opustením skupiny odoslať *Multicast Listener Done* správu. Časovač na koncovej stanici bol popísaný v sekcii o časovačoch a je použitý na pozdržanie odoslania správy.

Pri smerovačoch sú pre každú skupinu časovače dva. Konkrétne teda *Leave* a *Rexmt timer*. Aby však bolo možné tieto informácie o jednotlivých skupinách udržiavať pre každé rozhranie, museli byť vytvorené dalšie dve štruktúry.

```
struct HostInterfaceData
{
    MLD *owner;
    GroupToHostDataMap groups;

    HostInterfaceData(MLD *owner);
    virtual ~HostInterfaceData();
};
typedef std::map<int, HostInterfaceData*> InterfaceToHostDataMap;

struct RouterInterfaceData
{
    MLD *owner;
    GroupToRouterDataMap groups;
    RouterState mldRouterState;
    cMessage *mldQueryTimer;

    RouterInterfaceData(MLD *owner);
    virtual ~RouterInterfaceData();
};
typedef std::map<int, RouterInterfaceData*> InterfaceToRouterDataMap;
```

Obrázok 5-5 MLD Štruktúry na ukladanie informácii o rozhraniach

Obe štruktúry sú pomerne jednoduché a obsahujú mapu, v ktorej kľúčom je multicastová adresa a hodnotou štruktúra udržujúca informácie o danej skupine popísaná vyššie. Štruktúra pre rozhranie smerovača obsahuje navyše ešte časovač, ktorý slúži naperiodické odosielanie *General Query* správ.

5.1.4 Metódy

Samotný modul je zložený z väčšieho množstva metód. Väčšina z nich je však pomerne jednoduchá a po vzhliadnutí jej zdrojového kódu je jednoduché odhadnúť jej činnost. V tejto časti by som sa teda chcel zamerať na dve funkcie slúžiace na registrovanie a odregistrovanie multicastových adries na rozhraniach. Pre túto funkcionalitu obsahuje modul metódu *receiveChangeNotification*.

```
void MLD::receiveChangeNotification(int category, const cPolymorphic *details)
{
    Enter_Method_Silent();

    InterfaceEntry *ie;
    int interfaceId;
    IPv6MulticastGroupInfo *info;
    switch (category)
    {
        case NF_INTERFACE_CREATED:
            ie = check_and_cast<InterfaceEntry*>(details);
            if (ie->isMulticast())
                configureInterface(ie);
            break;
        case NF_INTERFACE_DELETED:
            ie = check_and_cast<InterfaceEntry*>(details);
            if (ie->isMulticast())
            {
                interfaceId = ie->getInterfaceId();
                deleteHostInterfaceData(interfaceId);
                deleteRouterInterfaceData(interfaceId);
            }
            break;
        case NF_IPv6_MCAST_JOIN:
            info = check_and_cast<IPv6MulticastGroupInfo*>(details);
            multicastGroupJoined(info->ie, info->groupAddress);
            break;
        case NF_IPv6_MCAST_LEAVE:
            info = check_and_cast<IPv6MulticastGroupInfo*>(details);
            multicastGroupLeft(info->ie, info->groupAddress);
            break;
    }
}
```

Obrázok 5-6 MLD metóda receiveChangeNotification

Túto funkciu vzdialene spúšťa modul *notificationBoard* v okamihu zaznamenania zmeny, ktorú sme chceli z nášho modulu sledovať. Konkrétne v module MLD sledujeme nasledovné zmeny:

- Vytvorenie noveho rozhrania
- Zmazanie rozhrania
- Pridanie do skupiny
- Opustenie skupiny

Pre prihlasovanie a odhlasovanie zo skupín sú však zaujímave práve posledné dve.Tieto zmeny sú vyvolávane v module *IPv6InterfaceData* funkciou *joinMulticasGroup* a *leaveMulticastGroup*. Obe tieto funkcie v implemntácii pre IPv6 chýbali a bolo teda nutné ich naimplementovať.

```
void IPv6InterfaceData::joinMulticastGroup(const IPv6Address& multicastAddress)
{
    if(!multicastAddress.isMulticast())
        throw cRuntimeError("IPv6InterfaceData::joinMulticastGroup(): multicast address

    IPv6AddressVector &multicastGroups = getHostData()->joinedMulticastGroups;

    std::vector<int> &refCounts = getHostData()->refCounts;
    for (int i = 0; i < (int)multicastGroups.size(); ++i)
    {
        if (multicastGroups[i] == multicastAddress)
        {
            refCounts[i]++;
            return;
        }
    }

    multicastGroups.push_back(multicastAddress);
    refCounts.push_back(1);


    changed1();

    if(!nbo)
        nbo = NotificationBoardAccess().get();
    IPv6MulticastGroupInfo info(ownerp, multicastAddress);
    nbo->fireChangeNotification(NF_IPv6_MCAST_JOIN, &info);
}
```

Obrázok 5-7 metóda joinMulticastGroup

Ako je možné vidieť, funkcia si drží referenciu o jednotlivých multicastových adresách. V prípade známej adresy si inkrementuje počítadlo a odošle do modulu *notificationBoard* notifikáciu o zmene. V prípade novej adresy, ktorú v zozname ešte nemá, pridá túto adresu do zoznamu a nastaví počítadlo na hodnotu 1.

Funkcia *leaveMulticastGroup* pracuje opačným spôsobom

```
void IPv6InterfaceData::leaveMulticastGroup(const IPv6Address& multicastAddress)
{
    if(!multicastAddress.isMulticast())
        throw cRuntimeError("IPv6InterfaceData::leaveMulticastGroup(): multicast address

    IPv6AddressVector &multicastGroups = getHostData()->joinedMulticastGroups;
    std::vector<int> &refCounts = getHostData()->refCounts;
    for (int i = 0; i < (int)multicastGroups.size(); ++i)
    {
        if (multicastGroups[i] == multicastAddress)
        {
            if (--refCounts[i] == 0)
            {
                multicastGroups.erase(multicastGroups.begin()+i);
                refCounts.erase(refCounts.begin()+i);

                changed1();

                if (!nbo)
                    nbo = NotificationBoardAccess().get();
                IPv6MulticastGroupInfo info(ownerp, multicastAddress);
                nbo->fireChangeNotification(NF_IPv6_MCAST_LEAVE, &info);
            }
        }
    }
}
```

Obrázok 5-8 Metóda leaveMulticastGroup

V prípade opustenia skupiny opäť skontroluje svoj zoznam a pre danú multicastovú skupinu dekrementuje počítadlo a odošle do modulu *notificationBoard* notifikáciu o zmene. Cez modul *notificationBoard* sa tieto zmeny propagujú vo forme notifikácií aj do modulu MLD, v ktorom sú vďaka nim spúšťané funkcie pre prihlásenie alebo odhlásenie zo skupiny.

V pripade obdržania notifikácie o pripojení novej skupiny, zavoláme funkciu *joinMulticastGroup*

```
void MLD::multicastGroupJoined(InterfaceEntry *ie, const IPv6Address& groupAddr)
{
    ASSERT(ie && ie->isMulticast());
    ASSERT(groupAddr.isMulticast());

    if (enabled && !groupAddr.isLinkLocal())
    {
        HostGroupData *groupData = createHostGroupData(ie, groupAddr);
        numGroups++;
        numHostGroups++;

        sendReport(ie, groupData);
        groupData->flag = true;
        startHostTimer(ie, groupData, unsolicitedReportInterval);
        groupData->state = MLD_HGS_DELAYING_MEMBER;
    }
}
```

Obrázok 5-9 Metóda multicastGrupJoined

V tejto funkcii je na koncovej stanici vytvorena nova struktura pre ulozenie informacii o skupine a nasledne odoslana report sprava join, a nastaveny priznak flag.

Pri obdrzani notifikacie o odhlaseni multicastovej skupiny z rozhrania je zavolana funkcia multicast group left. Tato funkcia je takpovediac opakom predchadzajucej.

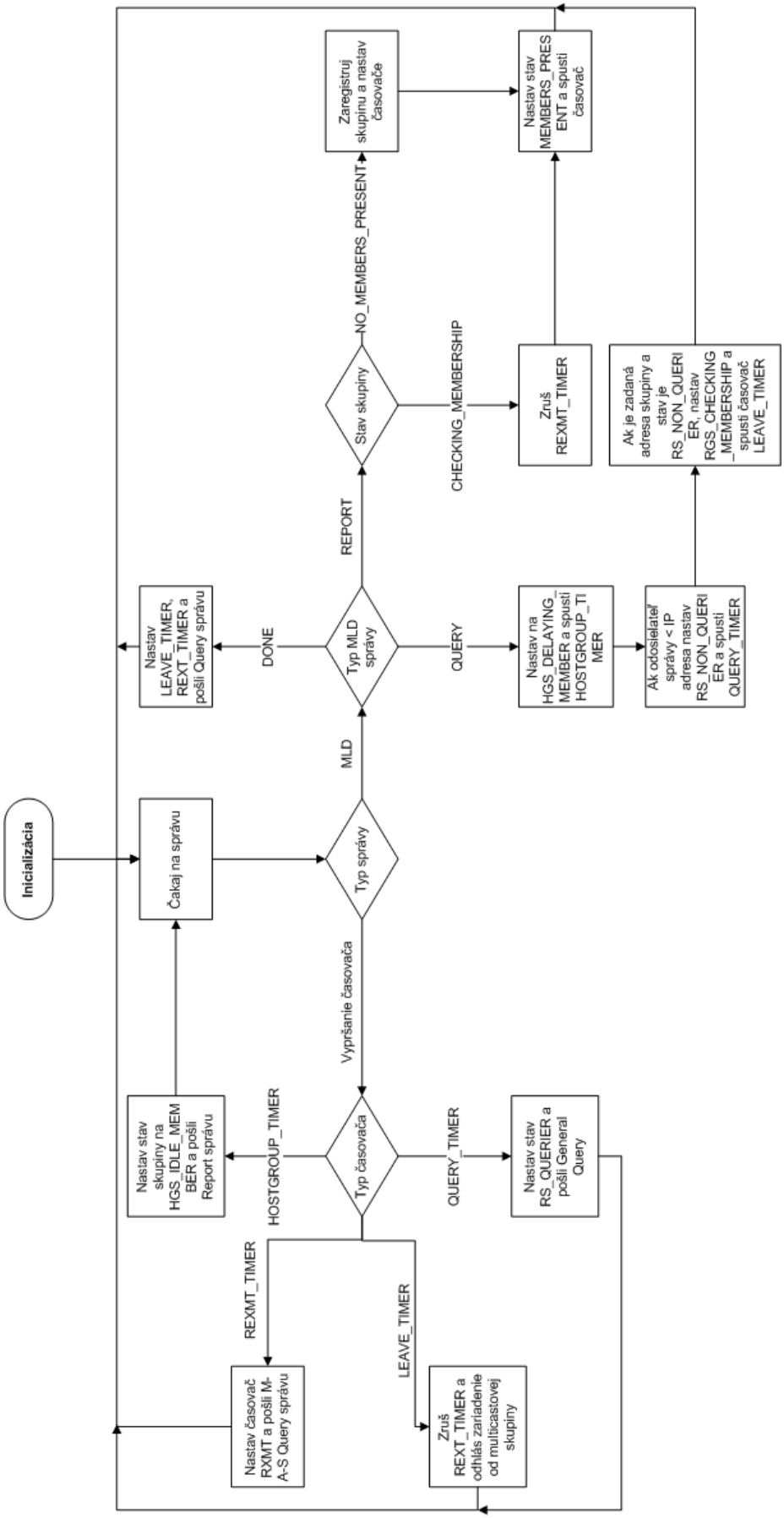
```
void MLD::multicastGroupLeft(InterfaceEntry *ie, const IPv6Address& groupAddr)
{
    ASSERT(ie && ie->isMulticast());
    ASSERT(groupAddr.isMulticast());

    if (enabled && !groupAddr.isLinkLocal())
    {
        HostGroupData *groupData = getHostGroupData(ie, groupAddr);
        if (groupData)
        {
            if (groupData->state == MLD_HGS_DELAYING_MEMBER)
                cancelEvent(groupData->timer);

            if (groupData->flag)
                sendLeave(ie, groupData);
        }

        deleteHostGroupData(ie, groupAddr);
        numHostGroups--;
        numGroups--;
    }
}
```

Obrázok 5-10 Metóda multicastGroupLeft

V tejto funkcii načítame aktuálne dáta o skupine. V prípade, že máme príznak *flag* nastavený na hodnotu *true*, odošleme *Done* správu aby sme o tejto zmene oboznámili aj smerovač a následne skupinu zmažeme.

5.1.5 Aktivita modulu

Nasledujúci diagram na obrázku 5-11 znázorňuje aktivitu naimplementovaného modulu. Jeho aktivita začína pri prvotnej inicializáci. Z inicializačnej časti sa program dostane do funkcie *handleMessage*, v ktorej čaká až do prijatia správy. Do tohto miesta sa následne dostáva program vždy po vykonaní príslušnej časti po spracovaní správy alebo časovača. Tento diagram nemá žiaden koncový bod, nakoľko je program vykonávaný v cykloch až do jeho vynúteného ukončenia.

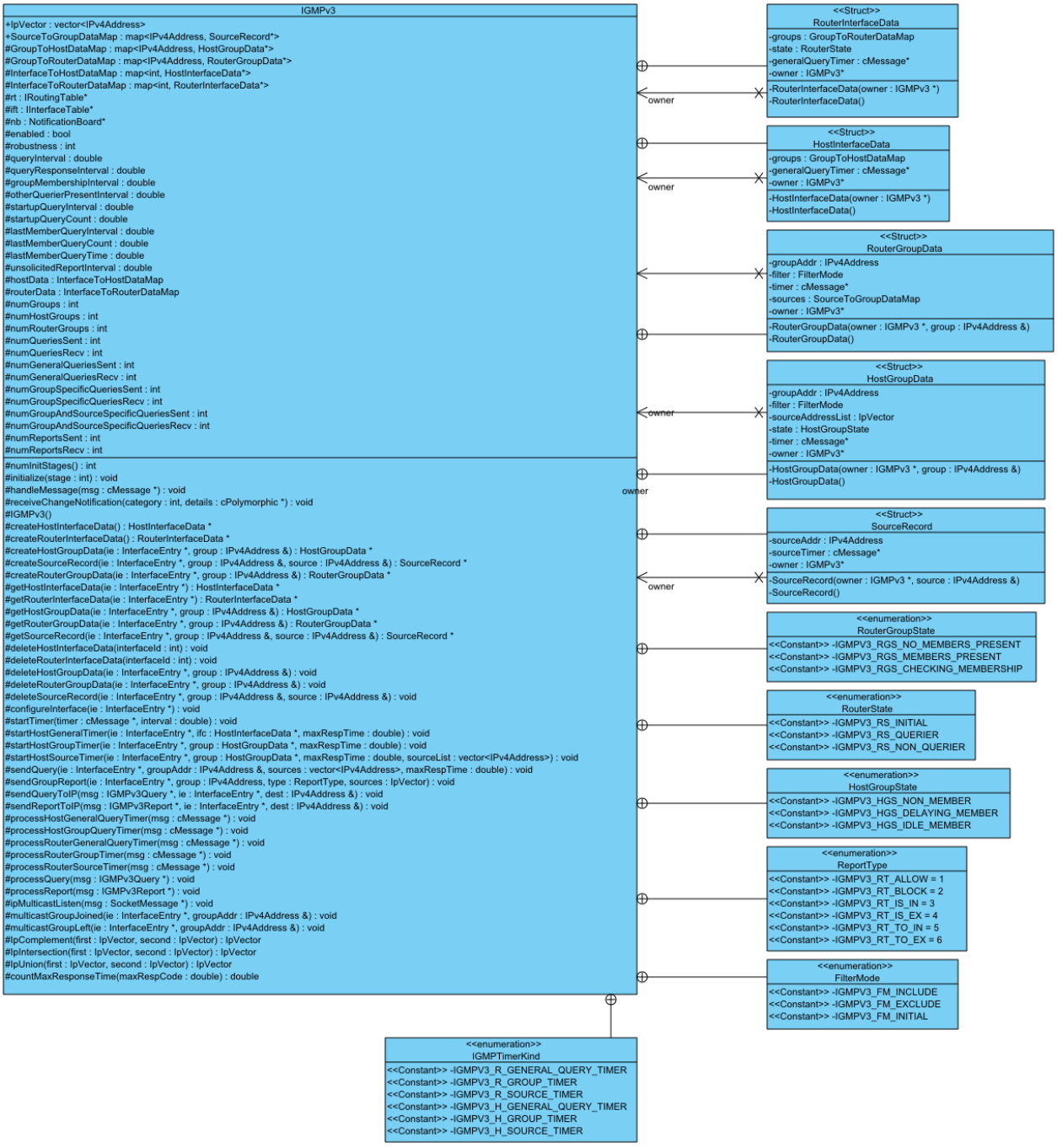


Obrázok 5-11 Diagram aktivity modulu MLD

5.2 IGMPv3

V tejto časti práce popíšem implementáciu modulu IGMPv3. Budú v nej popísane novo vytvorené štruktúry, časovače, atribúty a vysvetlím aj zložitejžie a zaujímavé časti použitých metód.

Nasledujúci ClassDiagram zobrazuje štruktúru triedy.



Obrázok 5-12 IGMPv3 ClassDiagram

5.2.1 Typy správ

Protokol IGMPv2 používa pre svoje fungovanie dva druhy správ. Keďže sú tieto správy odlišné od pôvodnej správy použitej vo verzii 2, bolo potrebne navrhnúť nové typy správ, ktoré budu spĺňať požiadavky IGMPv3. Boli preto vytvorené dva nové druhy správy. Prvou z nich je *IGMPv3Query*.

```
packet IGMPv3Query
{
    int type enum(IGMPType) = IGMP_MEMBERSHIP_QUERY;
    int maxRespCode;
    IPv4Address groupAddress;
    int numOfSources;
    IPVector sourceList;
}
```

Obrázok 5-13 IGMPv3 definícia Query správy

Atribút *type* je v tejto správe priradený na stálo, nakoľko v IGMPv3 neexistuje iný typ správy s rovnakou štruktúrou. Atribút *maxRespCode* v sebe nesie hodnotu využívanú pre výpočet časovača oneskorenia reportu. *groupAddres* je atribút, ktorý obsahuje adresu multicastovej skupiny dotazovanej v Query správe. Novinkou verzie 3 je dotazovanie sa na konkrétne zdrojové adresy pre multicastovú skupinu. V atribúte *numOfSources* je uvedený počet týchto zdrojových adries ktoré budú v správe obsiahnuté. Ako sa teda dalo očakávať, v poli sourceList sú uložené všetky IPv4 zdrojové adresy pre danú skupinu.

Druhým zložitejsím typom správy je typ IGMPv3Report. Zložitejším je preto, že správa nemá podobne jednoduchú štruktúru ako prechádzajúce správy ale je zložená z hlavičkových informácií a následne sú v nej uložené štruktúry pre skupinové záznamy.

Štruktúa pre skupinový záznam má nasledujúci formát:

```
struct GroupRecord
{
    int recordType;
    int numOfSources;
    IPv4Address groupAddress;
    IPVector SourceList;
};
```

Obrázok 5-14 IGMPv3 definícia skupinového záznamu

V atribute *record type* prenášame typ skupinového záznamu. Tieto typy boli podrobnejšie popísané v kapitole 2.4.3. Druhou položkou štruktúry je počet zdrojových adries pre danú skupinu, koré budu v zázname prenášané. V poli *groupAddres* je multicastová adresa slúžiaca na identifikáciu záznamu. Posledné pole obsahuje vektor všetkých zdrojových adries oznamovaných v skupinovom zázname.

Samotná správa teda vyzerá takto:

```
packet IGMPv3Report
{
    int type enum(IGMPType) = IGMPV3_MEMBERSHIP_REPORT;
    int numGroupRecords;
    GroupRecord groupRecord[];
}
```

Obrázok 5-15 IGMPv3 definícia Report správy

Ako v každej správe, v prvom poli je prenášaný typ správy slúžiaci na identifikáciu a následne preposlanie správnemu modulu na spracovanie. Zvyšné polia tvoria informácie o počte prenášaných skupinových záznamov a samotný vektor obsahujúci všetky zdrojové záznamy. Oba druhy správy majú v reálnej implementácii niektoré ďalšie polia. Nakoľko však niesu v simulačnom prostredí využívané, z implementácie správ som tieto polia vyradil.

5.2.2 Časovače

Nová verzia protokolu so sebou priniesla aj nové druhy časovačov. Niektoré ako naríklad *Query* a *Group* timer zostali s rovnakým využitím aj v novšej verzii. V tejto časti teda popíšem iba novo vzniknuté časovače.

```
enum IGMPTimerKind
{
    IGMPV3_R_GENERAL_QUERY_TIMER,
    IGMPV3_R_GROUP_TIMER,
    IGMPV3_R_SOURCE_TIMER,
    IGMPV3_H_GENERAL_QUERY_TIMER,
    IGMPV3_H_GROUP_TIMER,
    IGMPV3_H_SOURCE_TIMER,
};
```

Obrázok 5-16 IGMPv3 Časovače

Na prechod rozhrania zo stavu *EXCLUDE* do opätovného stavu *INCLUDE* bol na rozhraniach smerovačov vytvorený nový časovač s názvom *RouterGroupTimer*. Tento časovač je aktívny iba v prípade, že filtrovací mód skupiny je v stave *EXCLUDE*. Po jeho vypršaní je stav tohto módu prepnutý znovu do stavu *INCLUDE*. Vďaka novo vzniknutej štruktúre pre zdrojové záznamy, vznikol aj nový časovač. Tento časovač udržuje aktuálnosť zdrojových záznamov na smerovačoch. Posledným z novo vzniknutých časovačov je časovač na naplánovanie odpovedí pre prijatú *Genral Query* správu. Po vypršaní tohto časovača je z koncovej stanice odoslaná *Report* správa, ktorá zahŕňa všetky skupiny, do ktorých je koncová stanica prihlásená

5.2.3 Štruktúry

Ako už bolo spomenuté vyššie, v implementácii bola pridaná nová štruktúra pre zdrojový záznam. Táto štruktúra obsahuje IPv4 adresu záznamu a jeho časovač. Štruktúry pre informácie o skupinách pre koncové stanice a smerovač sa logicky taktiež mierne pozmenili.

```
struct HostGroupData
{
    IGMPv3 *owner;
    IPv4Address groupAddr;
    FilterMode filter;
    IpVector sourceAddressList;
    HostGroupState state;
    cMessage *timer;

    HostGroupData(IGMPv3 *owner,const IPv4Address &group);
    virtual ~HostGroupData();
};
typedef std::map<IPv4Address,HostGroupData*> GroupToHostDataMap;
```

Obrázok 5-17 IGMPv3 Štruktúra pre informácie o skupine koncového zariadenia

Štruktúra pre informácie o skupine u koncovej stanice bola obohatená a dve položky, ktorými sú filtrovací mód pre danú skupinu a zoznam zdrojových adries. Z impelentácie bol vypustený stav skupiny, ktorý vo verzii 3 nieje potrebný nakoľko všetky stanice odosielajú report bez ohľadu na ostatné koncové stanice.

```
struct RouterGroupData
{
    IGMPv3 *owner;
    IPv4Address groupAddr;
    FilterMode filter;
    cMessage *timer;
    SourceToGroupDataMap sources;

    RouterGroupData(IGMPv3 *owner, const IPv4Address &group);
    virtual ~RouterGroupData();
};
typedef std::map<IPv4Address, RouterGroupData*> GroupToRouterDataMap;
```

Obrázok 5-18 Štruktúra pre informácie o skupine na smerovači

Podobnou zmenou prešla aj štruktúra obsahujúca informácie o skupinách na smerovači. Pribudol v nej taktiež fitrovací mód a zoznam zdrojových záznamov. Do štruktúry pribudol aj časovač pre prechod zo stavu *EXCLUDE* popísaný v sekcii o časovačoch.

Štruktúra pre informácie o rozhraní smerovačov ostala nezmenená. Miernou zmenou však prešla štruktúra uchovávajúca záznamy o rozhraní koncvej stanice. Tejto štruktúre pribudol časovač odpovedí pre *General Query* správ spomentý vyššie.

5.2.4 Metódy

Modul IGMPv3 opäť obsahuje vačšie množstvo metód použitých pre spracovanie prijatých časovačov, vytváranie a napĺňanie správ a rôzne výpočty ako napríklad výpočet hodnoty *MaxResponseTime* z poľa *MaxResponseCode* prijatého v správe. Väčšina týchto metód je však pomerne triviálna a nieje teda potrebný ich ďalší popis.

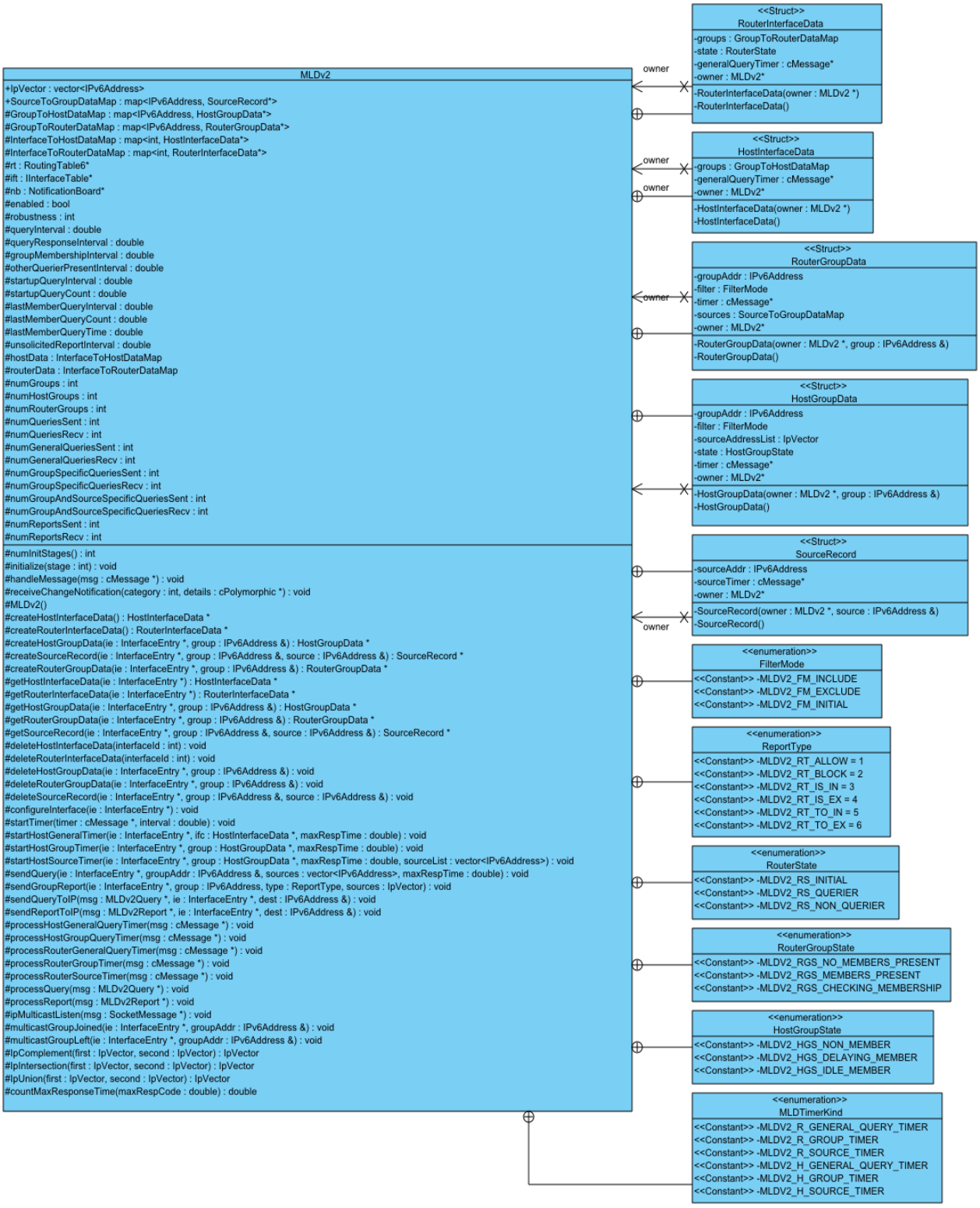
Medzi zujímavé metódy parí metóda *ipMulticastListen*, ktorá slúži na zisťovanie zmien na rozhraní koncových staníc a pri zmene stavu rozhrania odosiela report pokrývajúci danú zmenu.
Táto metóda spracúva nový typ správy nazvaný socketMessage. Táto správa však zatial nieje využívana kvôli absencii aplikácie pracujúcej s multicastovým vysielaním. Typ správy bol teda vytvorený iba ako referncia pre hodnoty použité v tejto metóde a bude slúžiť pre budúce možné rozšírenie tohto modulu. Samotné telo aplikácie prijme správu a porovná hodnoty na danom rozhraní s informáciami v obdržanej správe. V prípade, že po obdržaní a spracovaní tejto správy zostane stav rozhrania nezmenený neudeje sa nič. V opačnom prípade, ak došlo k zmene fitrovacieho režimu alebo zmene pri zozname zdrojových adries, je odoslaný report oznamujúci túto zmenu. Tento report je odosielaný smerom k smerovaču bez nutnosti obdržania *Query* správy.

Ostatné dve zujímavejšie metódy sú *procesQuery* a metóda procesReport. Pri metóde *procesQuery* dochádza k spracovaniu obdržanej *IGMPv3Query* správy. Nakoľko je potrebné naplánovať odpovď rovnakým spôsobom ako je to popísané v RFC3376 [23], prejdú informácie obdržané v správe niekoľkými podmienkami , ktoré zahŕňajú všetkých päť pravidiel na plánovanie typu odpovďe. Na základe vyhodnotenia týchto podmienok naplánujemodul typ odpoveďe a nastaví príslušný časovač na jej odoslanie.

Metóda *process Report* patrí medzi obsahovo najobjemenšie metódy. To je spôsobené tým, že pri spracovávaní reportov dochádza k veľkému množstvu možných kombinácii aktuálneho stavu skupiny a obdržanej odpovede. Všetky tieto kombinácie sú vyhodnotené a spracované akciami popísanými v tabuľke 2-6 .

5.3 MLDv2

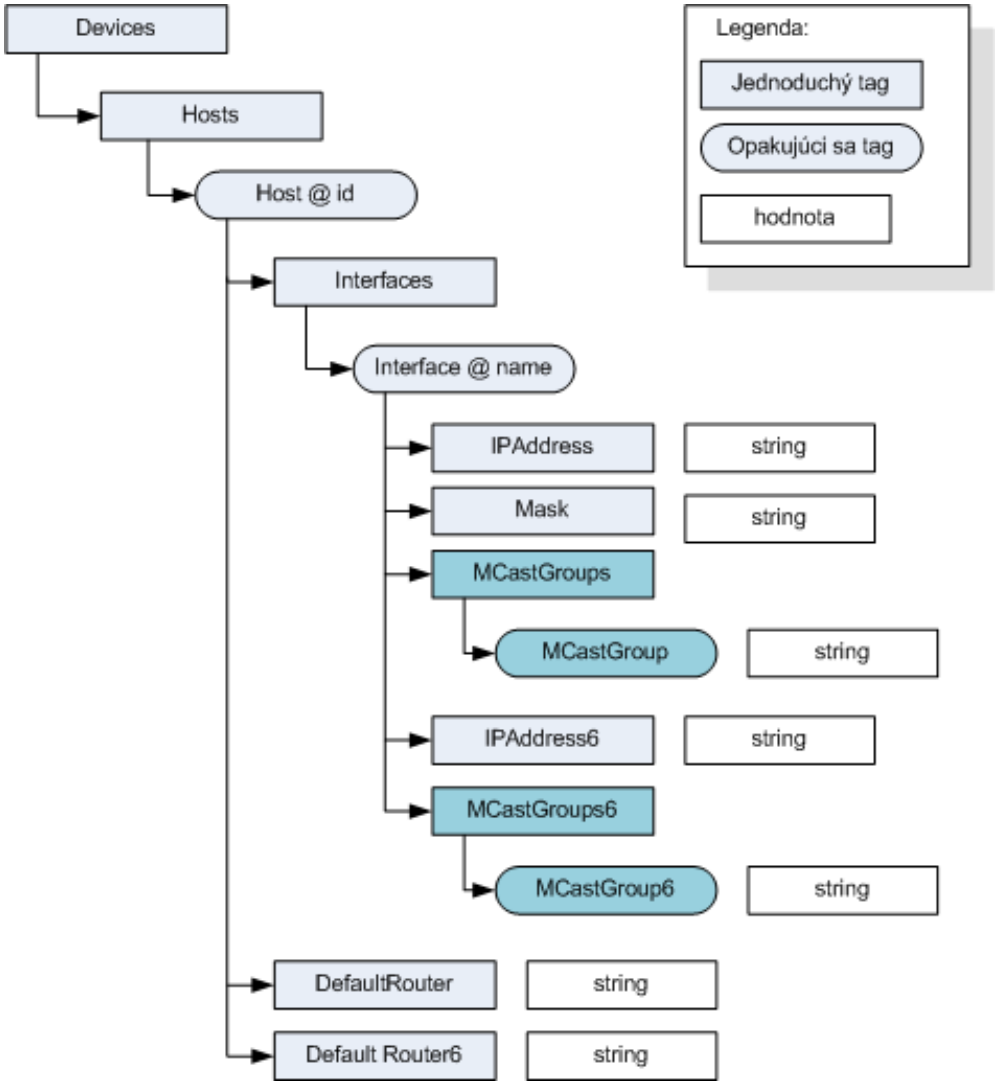
Pri implementácii protokolu MLDv2 došlo iba k prekladu protokolu IGMPv3 do IP verzie 6. Logicky teda tieto protokoly pracujú totožným spôsobom. Nakoľko bol pred protokolom MLDv2 naimplementovaný protokol MLD vo verzii 1. Všetky potrebné funkcie a metódy, ktoré pre IPv6 chýbali, boli naimplementované práve vo verzii jedna. Nasledujúci diagram ukazuje štruktúru triedy.



Obrázok 5-19 MLDv2 ClassDiagram

5.4 Načítanie konfigurácie z XML

Framework INET doposiaľ neobsahuje modul pre aplikáciu, ktorá by dokázala pracovať s multicastovým vysielaním. Nieje preto možné pomocou aplikácie prihlasovať či odhlasovať rozhrania do multicastových skupín. Jedinou možnosťou teda zostáva prihlásenie skupín priamo z konfiguračného súboru simulácie. Na toto načítavanie konfigurácie slúži trieda *deviceConfigurator*, ktorá pomocou parsovania získava jednotlivé údaje, ktoré následne aplikuje. Existujúci formát konfiguračného súboru som rozšíril o nové tagy(sfarbené do modra) zobrazené na obrázku



Obrázok 5-20 Štruktúra konfiguračného XML dokumentu

Popis jednotlivých tagov:

- **Devices** – tag Devices je použitý ako obal celého dokumentu a všetko ostatné musí byť umiestnené práve v tomto tagu.
- **Hosts –** v ramci tagu Host su umiestnené údaje o všetkých koncových zariadeniach obsiahnutých v konfiguračnom súbore
- **Host** – Tag host reprezentuje jednotlivé koncové stanice a pre ich identifikáciu používa atribút *id*. Na základe tohto atribútu sú nastavenia korektne priradené konkrétnej koncovej stanici. Tento tag je v konfiguračnom súbore použitý viac krát, nakoľko každé koncové zariadenie je obalené vlastným tagom s konkrétnim *id*.
- **Interfaces** – Tag interfaces obaľuje všetky rozhrania koncového zariadenia.
- **Interface** – pomocou tagu interface a jeho atribútu *name* je možné nastaviť konkrétnemu rozhraniu atribúty popísané v jeho vnorených tagoch.
- **IPAddress** – tento tag obsahuje IP adresu, ktorá bude nastavená na rozhraní, v rámci našej štruktúry sa jedná o prvý tag, u ktorého sa nastavuje priamo hodnota. Hodnotou je v tomto prípade reťazec, ktorý je v zdrojovom kóde použitý pre vytvorenie a nastavenie adresy.
- **Mask** – Podobne ako v predchádzajúcom prípade, tag v sebe nesie reťazec, ktorý je použitý pre nastavenie sieťovej masky rozhrania
- **MCastGroups** – tag pridaný pre podporu prihlasovania multicastových skupín. Ako v predchádzajúcich prípadoch je pre lepšiu orientáciu v XML použitý ako obaľovací tag.
- **MCastGroup** – je tag nesúci hodnotu konkrétnej adresy multicastovej skupiny, prezentovanej ako reťazec. Tento tag je použitý pre každú skupinu, ktorú chceme na dané rozhranie pripojiť.
- **IPAddress6** – je tag používaný pre nastavenie IPv6 adresy, narozdiel od tagu pre IPv4 tento tag obsahuje aj sieťovú masku, ktorá po adrese nasleduje za lomítkom.
- **MCastGroups6** – je rovnako ako pre IPv4 použitý pre obalenie všetkých skupín, ktoré chceme pomocou protokolu MLD pripjiť.
- **MCastGroup6** – analogicky k tagu MCastGroup je tento tag nositeľom informácie o konkrétnej IPv6 multicastovej adrese.
- **DefaultRouter** – je pri koncových zariadeniach atribút nastavujúci východziu bránu. Jeho hodnotou je reťazec predstavujúci IPv4 adresu brány.
- **DefaultRouter6** – rovnako ako v predchádzajúcom prípade tag v sebe prenáša informáciu a vychodzej bráne pre IPv6.

Popis konfiguračného XML súboru máme za sebou a tak sa môžeme pustiť do nastavovania. Pre načítanie hodnôt XML tagov, je použitá metóda *addIPv4MulticastGroups* pre IPv4 a *addIPv6MulticastAddress* pre IP verzie 6. Obe tieto metódy sú volané pred samotným spustením

simulácie v poslednom kroku inicializácie modulu *deviceConfigurator*. Obe metódy pracujú na rovnakom princípe takže popíšem fungovanie metódy pre IPv4.

Metóda prechádza celým XML súborom a číta hodnoty v tagoch MCastGroup. Keďže sa toto načítavanie deje pre každé koncové zariadenie a každé rozhranie je možné pre každú z týchto skupín zavolať metódu *joinMulticastGroup*, čím zabezpečíme prihlásenie skupiny pre konkrétne zariadenie a jeho konkrétne rozhranie.

```
InterfaceEntry *ie = ift->getInterfaceByName(iface->getAttribute("name"));
bool empty = true;
for (cXMLElement *mcastNode=MCastGroupsNode->getFirstChild(); mcastNode; mcastNode = mcastNode->getNextSibling())
{
    const char *mcastAddress = mcastNode->getNodeValue();
    ie->ipv4Data()->joinMulticastGroup((IPv4Address)mcastAddress);
    empty = false;
}
```

Obrázok 5-21 volanie metódy joinMulticastGroup

5.5 Vizualizácia multicastových stromov

Implementácia vizualizácie stromov je pomerne jednoduchá po pridaní atribútu pre adresu Querier smerovača, stačí prejsť v cykle cez všetky rozhrania koncových staníc. Následne je do koznoly vytváraný výpis v tvare:

$$IPAdresa\ Koncovej\ stanice \rightarrow IP\ adresa\ Querier\ smerovača;$$

Všetky tieto výpisy sú pred samotným cyklom ešte obalené v štruktúre:

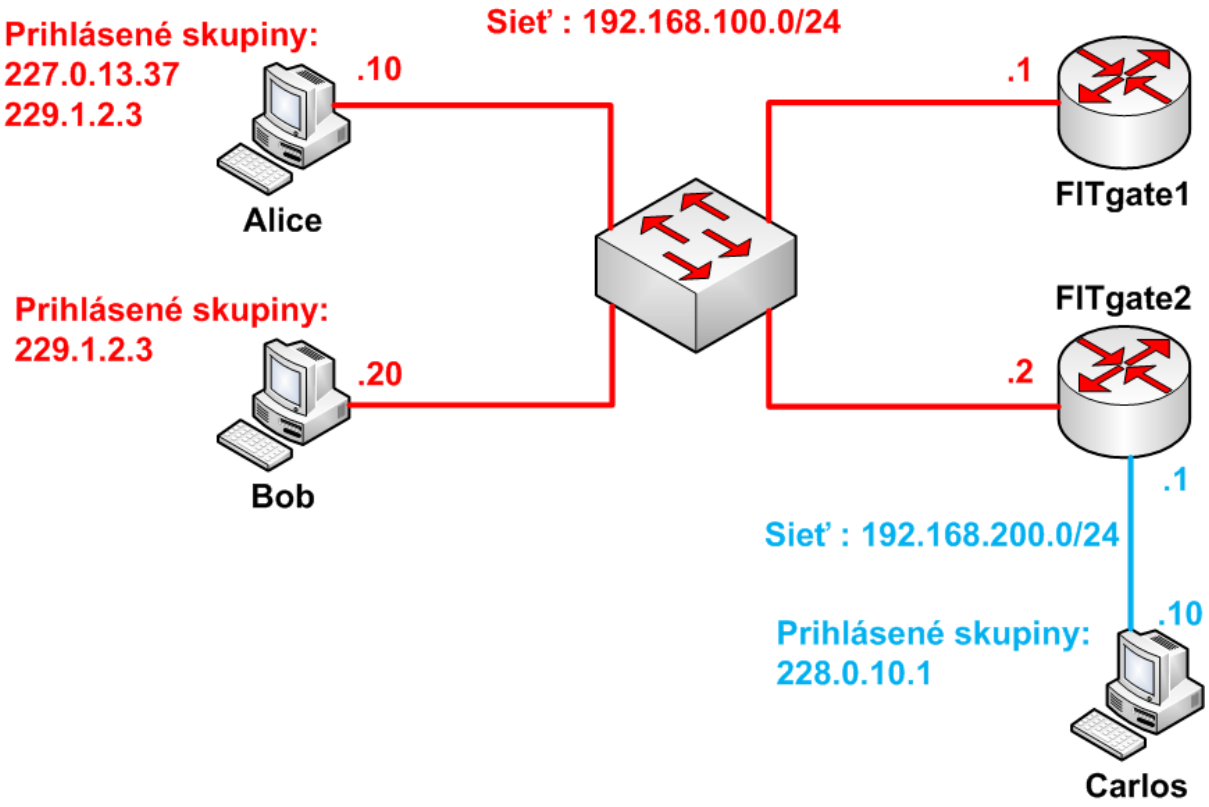
$$diagraph\ Name\ \{\ \}$$

Takto vzniknutý textový súbor je následne možné vložiť ako parameter pre program GraphViz[6].

[6] www.graphviz.org

6 Porovnanie simulácie so správaním reálnej Cisco siete

Pre otestovanie implementovaných modulov som zvoli univerzálnu topológiu na obrazku 6-1. Táto topológia pozostáva z troch koncových staníc nazvaných *Alice*, *Bob* a *Carlos* a dvoch smerovačov nazvaných *FITgate1* a *FITgate2*.



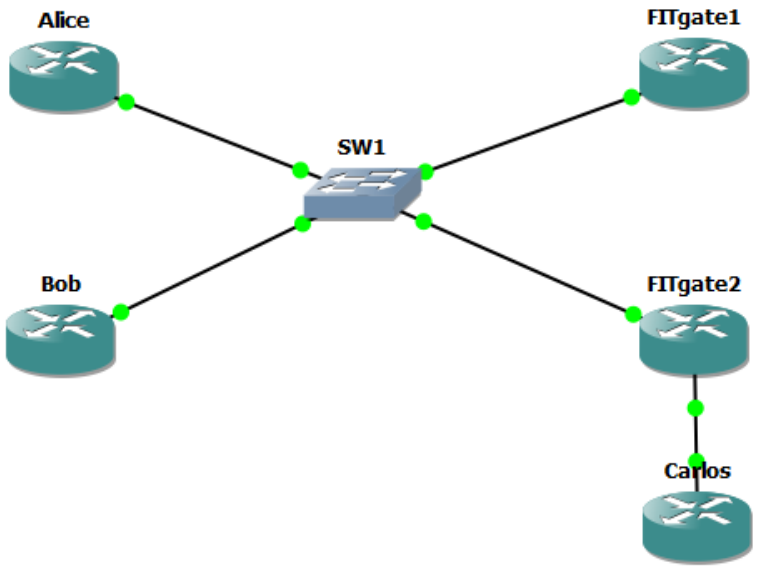
Obrázok 6-1 Referenčná topológia

V tejto topológii budú stanice *Alice* a *Bob* pripojené pomocou prepínača k obom smerovačom. Číslo tejto siete bude 192.168.100.0/24. Druhá sieť 192.168.200.0/24 bude tvorená iba smerovačom *FITgate2* a koncovou stanicou *Carlos*. Toto zapojenie je pre potreby nášho testovania optimálne. Na zapojení je totiž možné pozorovať periodické dotazovanie sa od *Querier* smerovača a rovnako tak odpovede jednotlivých koncových staníc pre obe verzie IGMP. Je taktiež možné vidiet voľbu *Querier* smerovača pre sieť 192.168.100.0/24. Druhá sieť bude slúžiť na ukážku toho, že smerovač môže byť pre jednu sieť v stave *NonQuerier* avšak pre ďalšiu pripojenú sieť môže zastávať úlohu *Querier* smerovača. V simulácii bude taktiž možnosť vidieť načítanie mutlticastových skupín z konfiguračného súboru a ich následné prihlásenie pomcou *Join* správ. Nakoľko framework INET doposiaľ neobsahuje aplikáciu, pre prácu s multicastovým vysielaním, neje momentálne možné

otestovať niektoré ďalšie funkcie ako napríklad odoslanie *Leave* správ alebo nastavovanie zdrojovych adries a ich filtrovanie pri IGMPv3.

Pre testovanie na reálnej sieti Cisco je použitý sieťový emulátor GNS3[7]. Ako referenčné zariadenie som použil smerovač Cisco 2961 s ios verziou *c2691-advsecurityk9-mz.124-6.t*. Pre emulovanie koncových staníc som taktiež využil tieto smerovače. Pomocou príkazu *ip igmp join-group* som simuloval prihlásenie koncovej stanice do zvolenej multicastovej skupiny. Pre odchytávanie komunikácie som využíval aplikáciu Wireshark[8].

Na nasledujúcom obrázku je zobrazená vytvorená topológia v emulatore GNS3. Všetky konfigračné súbory pre oba testy sú umiestnené v prílohách.



Obrázok 6-2 GNS3 topológia

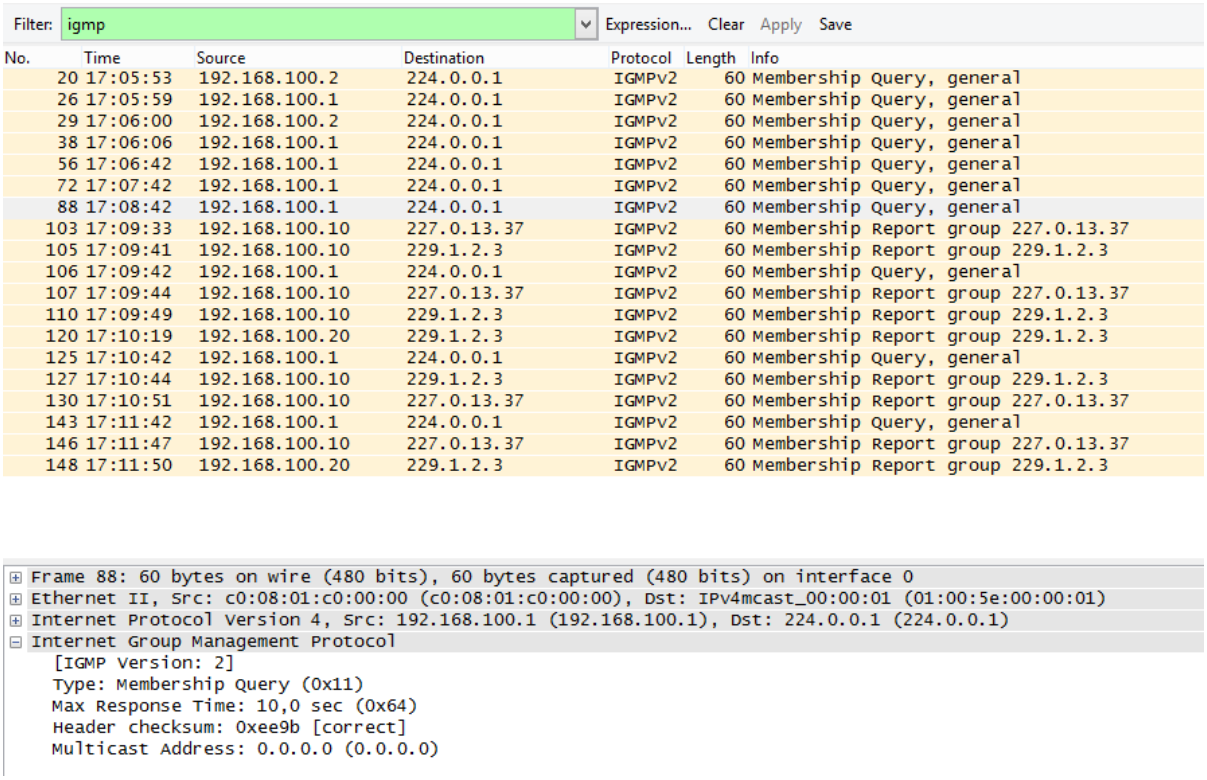
6.1 Test IGMPv2

Reálna sieť

Topológia v emulátore je zapojená a nastavená presne podla referenčnej topológie na obrazku 6-1.

Na multicastovych rozhraniach je spustený protocol PIM vo verzii *dense-mode*. Verzia IGMP je na všetkých rozhraniach nastavená na verziu 2. Odchytávanie komunikácie je spustené na linke medzi prepínačom a koncovou stanicou *Alice*. Kedže sa jedná o prepínanú sieť, v odchytenej komunikácii budeme vidieť všetky IGMP správy v sieti 192.168.100.0/24.

[7] www.gns3.net
[8] www.wireshark.org



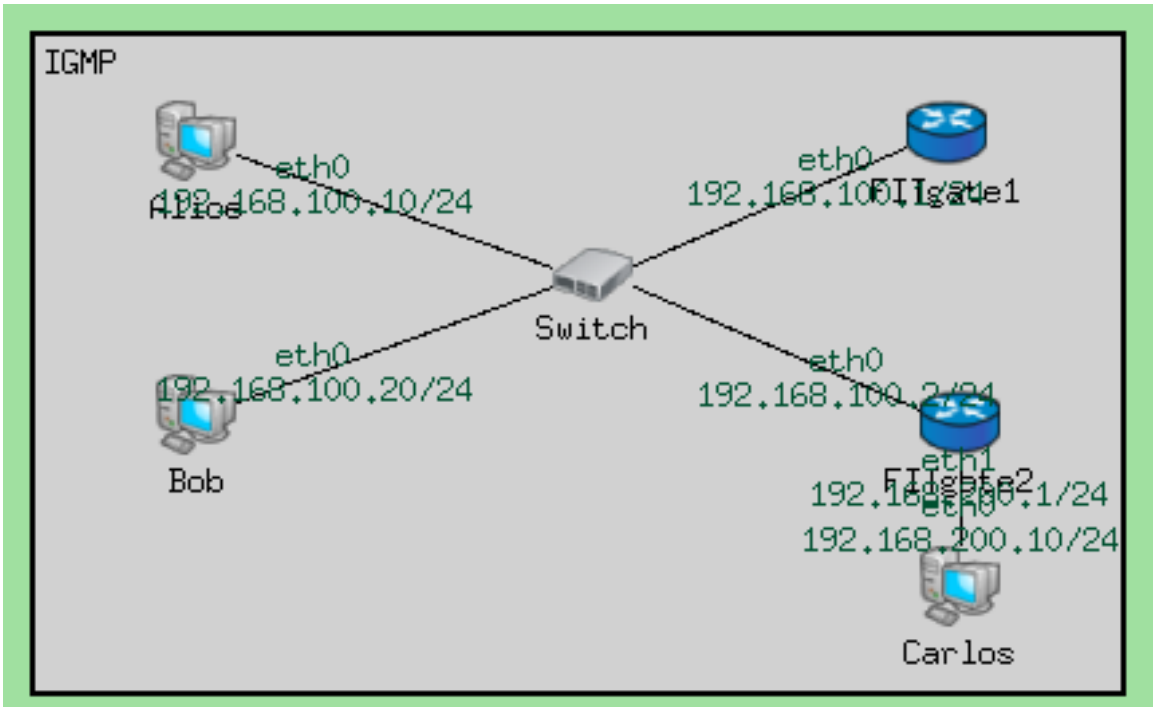
Obrázok 6-3 Odchytená IGMPv2 komunikácia na reálnej sieti

Z výstupu môžme vidieť, že ako prvá prebehne voľba *Querier* smerovača. Na obrázku 6-3, smerovač *FITgate2* odosiela *General Query* správu na adresu 224.0.0.1, následne po ňom rovnakú správu odosiela aj smerovač *FITgate1*. Smerovač *FITgate2* po obdržaní *General Query* od smerovača *FITgate1* zmení svoj stav pre sieť 192.168.100.0/24 na stav *NonQuerier* a ďalej pre túto sieť *Query* správy neodosiela. Ako je možné vidieť vo fáze tri už odosiela *General Query* iba smerovač *FITgate1* v pravidelnom intervale 60 sekúnd, čo je prednastavená hodnota na zariadeniach Cisco. Pre každú multicastovú skupinu prichádza do 10 sekúnd(*Max Response Time*) od obdržania *Query Report* správa. Na obrázku 6-3 je taktiež možné sledovať, že pre každú skupinu je *Report* správa odosielaná iba z jednej koncovej stanice, nakoľko druhá stanica po obrdžaní reportu svoj časovač zruší a odpoveď neodosiela. Toto správanie sa periodicky opakuje každých 60 sekúnd.

Pozrime sa teraz na sieť 192.168.200.0/24. V tejto sieti máme iba jediný smerovač, ktorým je *FITgate2*. Ten je pre sieť 192.168.100.0/24 v stave *NonQuerier*. Pre sieť 192.168.200.0/24 však odosiela *Query* správy a je v tejto sieti v stave Querier. Rovnako ako *Querier* smerovač v druhej sieti odosiela pravidelne v intervale 60 sekúnd *General Query* správy a následne do ďalších 10 sekúnd dostáva odpoveď v podobe reportu.

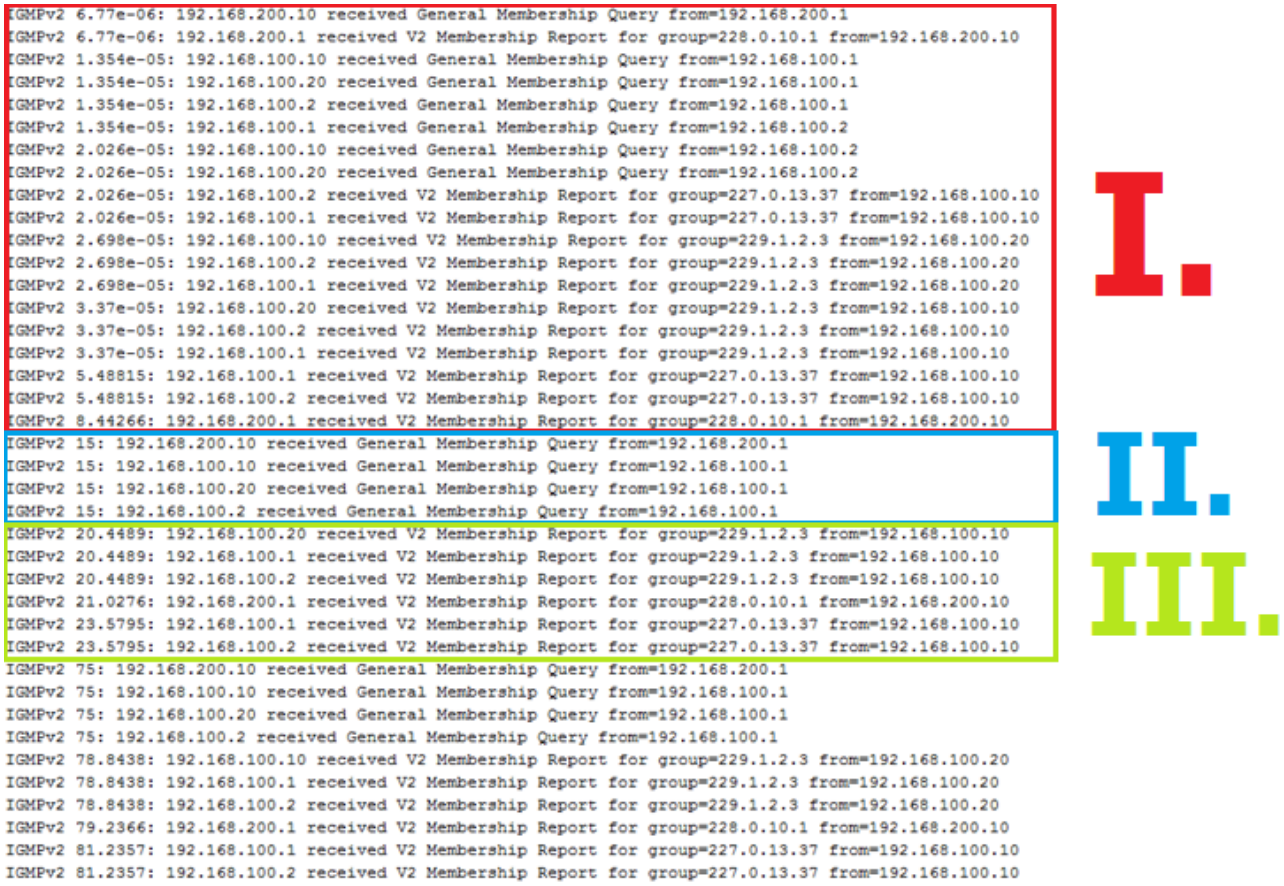
OMNeT++ simulácia

Pre simuláciu v prostredí OMNeT++ som si namodeloval referenčnú topológiu z úvodu kapitoly. Na zariadeniach boli pomocou konfiguracneho XML súboru nastavené IP adresy a multicastové skupiny, ktoré majú byť v inicializácii simulácie nastavené. Po spustení simulácie som pri prijatých správach robil konzolový výpis, ktorý obsahuje čas prijatia správy, jej typ, IP adresu zariadenia, ktoré správu obdržalo, IP adresu zariadenia, ktoré správu odoslalo a pokiaľ sa jednlo o *Group-Specific* správu tak aj skupinu, ktorej sa správa týka. Nakoľko modul IGMP je naimplemntovaný podľa RFC2236 [9], bolo pred samotnou simuláciou potrebné pomocou inicializčného *.ini* súboru nastaviť *Query Interval* z pôvodnej hodnoty 125 sekúnd na hodnotu 60 sekúnd tak, aby sa zhodoval s hodnotou na zariadeniach Cisco. Výsledna namodelovaná simulacia je znázornená na nasledujúcom obrázku.



Obrázok 6-4 OMNeT++ topológia

Po spustení simulácie je vidieť prvotnú výmenu správ, ktoré boli naplánované v inicializacii pred samotným spustením simulácie. Tu sú odosielané prvotné *Query* správy od smerovačov a *Join Report*správy od koncových zariadení, pomocou, ktorych sa prihlásia k odberu multicastových kupín.



Obrázok 6-5 Odchytená IGMPv2 komunikácia OMNeT++

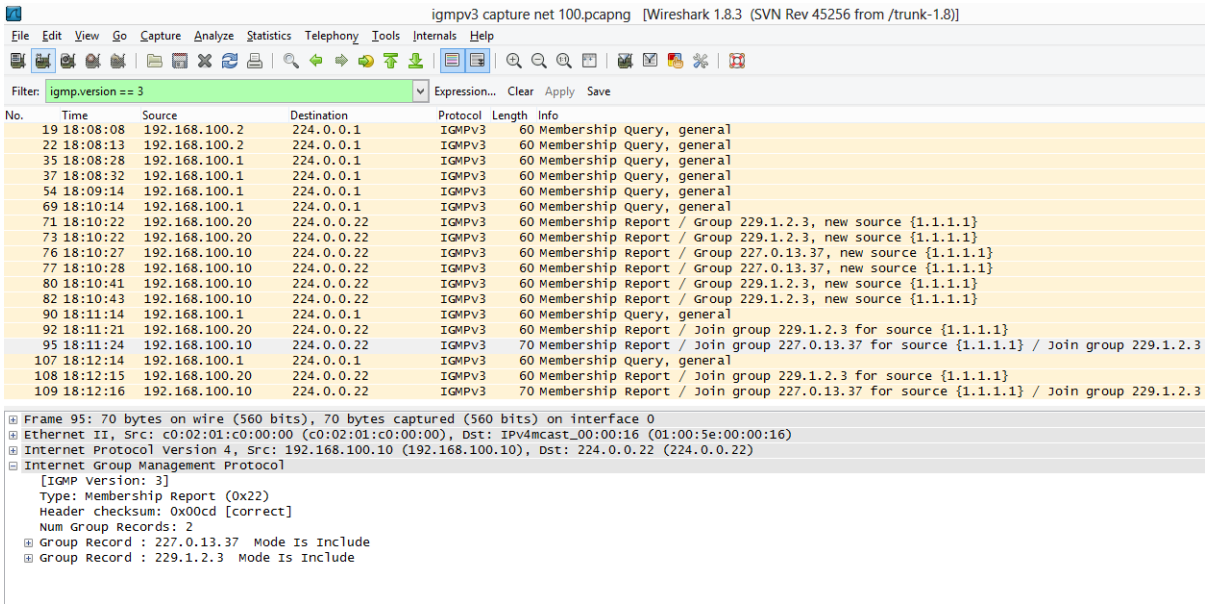
Po prvotnej výmene je vo fáze II. vidieť, že *General Query* správy su odosielané už len smerovačom *FITgate1* a sú odosielané periodicky každých 60 sekúnd. Rovnako ako v reálnej sieti, odpovede na *Query* správy dorazia k smerovaču vo fáze III. V čase do 10 sekúnd od obdržania *Query*. Pri sieti 192.168.200.0/24 je situácia opäť obdobná ako pri reálnej sieti a smerovač *FITgate2* v úlohe *Querier*- a odosiela *Query* správy, na ktoré dostáva odpovede od koncovej stanice *Carlos*.

Zhodnotenie

Pri porovnaní správania reálnej siete postavenej na smerovačoch Cisco a spravania simulácie pomocou naimplementovaných modulov v prostredi OMNeT++ sme dosiahli zhodu. Drobné odchýlky v časoch sú spôsobené generovaním náhodných časov na pozdržanie odpovede.

6.2 Test IGMPv3

Pre testovanie správania protokolu IGMPv3 v reálnej sieti som použil rovnakú topológiu ako pri predchádzajúcom teste. Rozdiel však nastal na rozhraniach, kde pre všetky IGMP rozhrania bola verzia nastavená na hodnotu 3. Rovnako ako v prvej simulácii je na začiatku vo fáze I. vidieť voľba *Querier* smerovača. Keďže algoritmus voľby *Querier* smerovača zostal nezmenený stane sa *Querier* smerovačom pre sieť 192.168.100.0/24 smerovač *FITgate1*.



Obrázok 6-6 Odchytená IGMPv3 komunikácia na reálnej sieti

Vo fáze II. výstupu odchytenej komunikácie je ďalej vidieť prihlasovanie jednotlivých skupín na rozhraniach. Následne už môžme sledovať periodické odosielanie *General Query* správy od smerovača *FITgate1*.

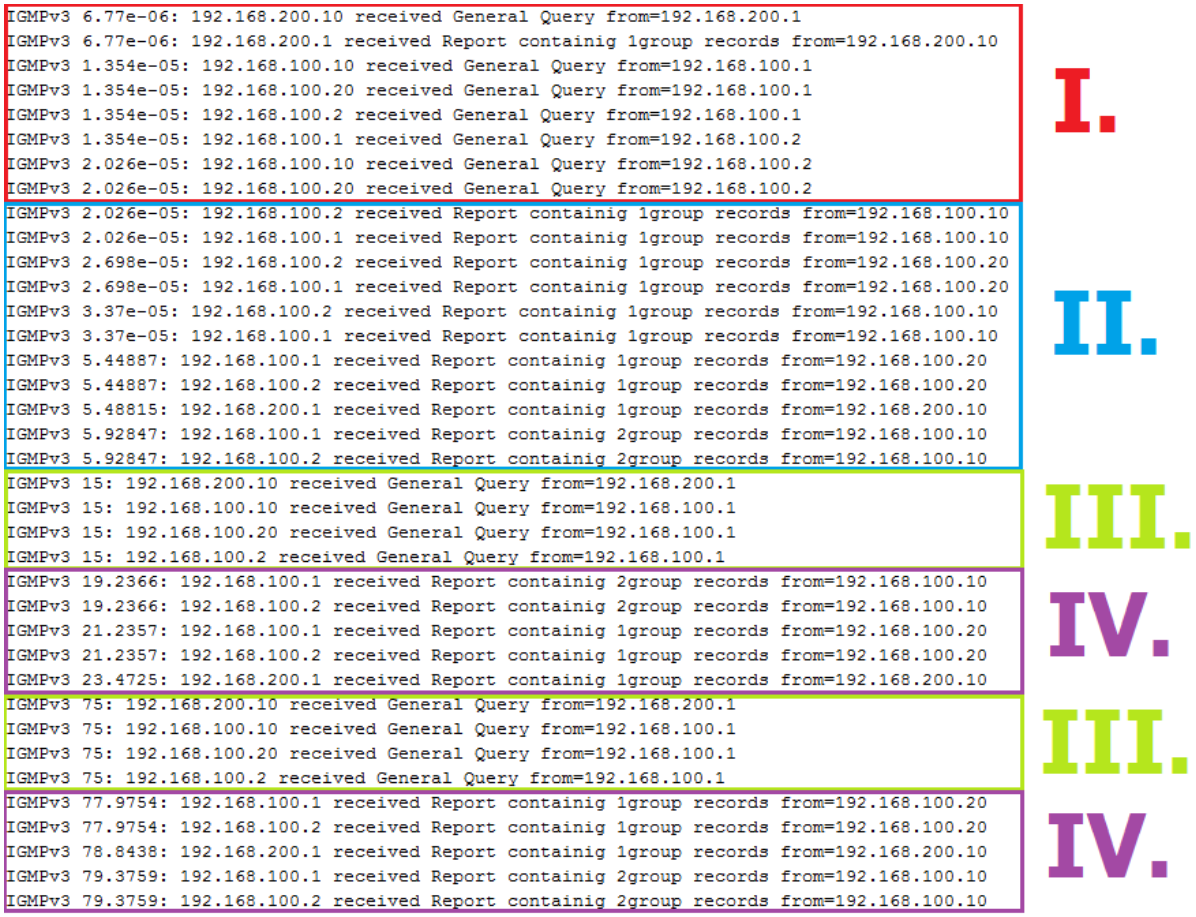
Zmena voči verzii 2 však nastáva pri odosielaní reportov. Tie, ako je vidno z obrázku 6-6 vo fáze IV. neodosielajú reporty na skupinovú adresu ale používajú ako destináciu multicastovú adresu všetkých IGMPv3 smerovačov 224.0.0.22. To teda znamená, že nenastáva rušenie odoslania reportu pri príchode reportu od iného koncového zariadenia a preto vidime odoslané *Report* správy od každého koncového zariadenia. Rozdielne je aj odosielanie reportu samotného. Vo verzii 2 sa odosielal report pre každú skupinu zvlášť a vo verzii 3 vidíme, že *Report* správa obsahuje viacero skupinových záznamov. Pozdržanie odoslania reportov ostalo nastavené na hodnotu 10 sekúnd. Ako je vidieť z obrázku, obe zariadenia odpovedajú svojim reportom v náhodne zvolenom čase v rozmedzí 0 až 10 sekúnd.

Pri sieti 192.168.200.0/24 sa rovnako ako v predchádzajúcej verzii stal *Querier*-om smerovač *FITgate 2*. Ten teda odosiela opäť pravidelne v intervale 60 sekúnd *General Query* správy, na ktoré mu ako je z obrázku vidiet chodia do 10 sekúnd odpovede od koncovej stanice *Carlos*.

OMNeT++ simulácia

V simulačnom prostredí využijeme opäť rovnakú topológiu ako pre verziu 2. Jediným rozdielom je, že v inicializačnom súbore nastavíme verziu IGMP na hodnotu 3.

Ako je po odchytení komunikácie vidieť. V prvom kroku po inicializácii su odoslané prvé *General Query* správy od všetkých smerovačov.



Obrázok 6-7 Odchytená IGMPv3 komunikávia OMNeT++

Následne na to sú vo fáze 2 odoslané *Report* správy pre prvotné prihlásenie koncových staníc do zvolených multicastových skupín, a v zápätí na to sú odoslané odpovede na prvé *Query* správy. V kroku číslo III. už môžme zaznamenať, že po 60 sekundách už *Query* odosiela iba *Querier* smerovač, ktorým sa stal *FITgate1*. V kroku číslo IV. môžme opäť sledovať odpovede koncových staníc. Každá z koncových staníc odosiela práve jednu odpoveď, v ktorej odosiela informácie o všetkých svojich pripojených skupinách. Vo výpise môžme vidieť koľko skupinových záznamov je prenášaných ktorou správou. Všetky tieto odpovede suú odosielané v náhodnom intervale do 10 sekúnd od obdržania *Query*. Od tohto okamihu sa začína opakovať správanie popísané vo fázach III. a IV.

Pre sieť 192.168.200.0/24 sa stal *Querier*-om smerovač *FITgate2* a v kroku III. A IV. pracujú spolu s koncovou stanicou *Carlos* rovnakým spôsobom ako zariadenia v sieti 192.168.100.0/24.

Zhodnotenie

Tak ako v predchádzajúcom teste, správanie odsimulované v simulačnom nástroji OMNeT++ zodpovedalo správaniu sa reálneho smerovača. Jedným rozdielom boli opäť mierne odlišné časy, ktoré sú spôsobené náhodným nastavením časovača v rozmedzí 0 – *max response time*.

Druhým rozdielom je obsah *Join* správ. V simulačnom modeli sú *Join* správy vytvárané ako reporty s filtrom nastaveným na stav *EXCLUDE* a prázdnym zoznamom zrojov. Na smerovači Cisco, však správu *Join* pre IGMP verzie 3 bez zadanej zdrojovej adresy nejde odoslať. Po zadaní zdrojovej adresy už však má správa nastavený filter na stav*INCLUDE*. V teste bolo preto popísane iba správanie protokolov a skontrolované hodnoty časovačov a nie samotný obsah *Report* správ.

7 Záver

Cieľom tejto práce bolo obznámenie sa s multicastovým smerovaním a všeobecným fungovaním multicastu v prostredí IPv4 a IPv6. Rozobral som v nej podrobnejšie multicastový smerovací protokol PIM a jeho päť módov. Ku každému z nich som pripísal jeho základnú charakteristiku. Neodlúčiteľnou súčasťou multicastového vysielania sú aj jeho klientské protokoly, bez, ktorých by nebolo možné prejavovať o multicastové vysielanie záujem. Protokoly IGMP a MLD som následne podrobne popísal a vysvetlil rozdiel medzi staršími a novšími verziami. Analógiou k ptorokolu IGMP som popísal obe verzie protokolu MLD. Pri multicastovom smerovaní cesta od zdroja k cieľu prechádza stromovou štruktúrou. Funkciu a význam tchto stromov ako aj dva rôzne prístupy k prenosu dát cez tieto štruktúri som popísal v poslednej podkapitole o multicaste.

V nasledujúcej kapitole som sa zameral na IP Multicast na zariadeniach Cisco. V tejto kapitole som uviedol všetky základné kroky a predpoklady pre nastavenie multicastu na smerovačoch Cisco. Poznatky z tejto kapitoly som využil aj pri nastavovaní simulácii v reálnej sieti.

Cieľom tejto práce však nebolo iba pochopenie teórie a preto som sa v kapitole štyri pustil do návrhu jednotlivých modulov pre simulátor OMNeT++. Ako prvý som otestoval už implementovaný protokol IGMP verzie 2. Po dôkladnej analýze zdrojových kódov a otestovaniu protokolu voči reálnemu zariadeniu som použil protokol IGMPv2 ako základ pre implemntáciu zvyšných troch. Nakoľko protokol MLD je prekladom IGMP voľba prvého modulu bola jasná. Preložil som teda protokol IGMP do IPv6. Také jednouché to však nebolo. Pre IPv6 nebolo naimplementovaných veľké množstvo menších pomocných štruktúr a metód, ktoré v IPv4 naimplementované sú. Pustil som sa teda do implementácie týchto drobností a po krátkom čase bol na svete protokol MLD. Logickou voľbou pokračovania implemntácie bol protokol IGMPv3. Popis implemntácie protokola IGMPv3 je popísaný v kapitole 5. Sú v nej uvedené nutné zmeny a pridané novinky, ktoré sa v staršej verzii protokolu nenachádzali. Ako posledný prišiel na rad MLD vo verzii 2. Kedže sa opäť jedná na logickej úrovni o rovnaký protokol ako IGMP, stačilo tento protokol naučiť hovorit jazykom IPv6. Všetky záludnosti, na ktoré som pri implemntácii prvej verzie MLD natrafil mi uľahčili výslednu implementáciu protokolu vo verzii 2.

Tieto moduly by však neboli na nič keby nebolo možné nastaviť v konfigurácii o ktoré skupiny máme záujem. Nakoľko je prekladač pre načítavanie konfigurácii zo zariadení Cisco nefunkčný, po dohode s vedúcim práce som nčítanie konfigurácie naimplementoval do modulu deviceConfigurátor, ktorý je používaný pre nastavovanie viacerých protokolov vytvorených v rámci projektu ANSA. Pre absenciu aplikácie schopnej využívať multicastové moduly ako PIM, IGMP alebo MLD, nebolo vo finálnom testovaní modulov voči reálnej sieti možné otestovať všetky prípady. Do testovania som teda zahrnul všetky situácie, ktoré pomocou nastavenia z konfiguračného režimu alebo vyplývajúc zo samotného správania protokolu ukázali zhodu takmer vo všetkých bodoch

testovania voči reálnemu zariadeniu. Modul IGMPv3 a MLDv2 v sebe obsahujú prepripravenú funkciu spolu so správou, ktorú bude pri dalšom rozširovaní modulu možné použiť a napojiť na aplikáciu. V budúcnosti by táto aplikácia mohla prepájať moduly IGMP a PIM a ukázať tak silu týchto protokolov v simulačných scenároch, ktoré vdaka nej vyuzžijú všetky možnosti, ktoré tieto protokoly ponúkajú. Ako posledný bod implemntácie je funkcia na zobrazovanie a grafickú reprezentáciu mutlicastových stromov. Keďže sa moja práca zaoberá klientskými protolmi, vyzualizujú sa v nej práve počiatočné kroky od koncových staníc k ich Querier smerovačom. Pre splnenie tejto funkcionality bolo nutné pridať ku každému rozhraniu klienta parameter, v ktorom si pamätá adresu svojho Queriera. Tento textový výstup mal automaticky generovať graf. Nakoniec sa mi však nepodarilo prepojiť prostredie OMNeT++ s programom GaphViz. Aj napriek tomuto problému som s dosiahnutým výsledkom spokojný. Vďaka tejto práci som opäť prehĺbil svoje vedomosti o sieťach a verím, že ich do budúcna budem môcť využiť.

Literatúra

1. **Goyenech, J.** Multicast over TCP/IP HOWTO. *tldp.org.* [Online] 20. Marec 1998. [Dátum: 6. Január 2012.] http://tldp.org/HOWTO/Multicast-HOWTO.html.

2. **Williamson, Beau.** Developing IP Multicast Networks Volume I. *Developing IP Multicast Networks Volume I.* Fifth Printing. Indianapolis : Cisco Press, 2000, 2.

3. **IANA.** IPv4 Multicast Address Space Registry. *www.iana.org.* [Online] 22. December 2011. [Dátum: 6. Január 2012.] http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xml.

4. **Hinden, R. a Deering, S.** IP Version 6 Adressing Architecture. *www.ietf.org.* [Online] Február 2006. [Dátum: 10. Január 2012.] http://www.ietf.org/rfc/rfc4291.txt. RFC 4291.

5. **Veselý, V.** Multicast. *www.fit.vutbr.cz.* [Online] 2010. [Dátum: 10. Január 2012.] https://www.fit.vutbr.cz/study/courses/PDS/private/pdf/pds-05-multicast.pdf.

6. **Kozierok, Charles M.** The TCP/IP Guide. *www.thetcpipguide.com.* [Online] 20. September 2005. [Dátum: 10. Január 2012.] http://www.tcpipguide.com/free/t_IPv6MulticastandAnycastAddressing-2.htm.

7. **Cisco.** Internet Protocol Multicast. *docwiki.cisco.com.* [Online] [Dátum: 6. Január 2012.] http://docwiki.cisco.com/wiki/Internet_Protocol_Multicast.

8. **Deering, Steve.** Host Extensions for IP Multicasting. *www.ietf.org.* [Online] August 1989. [Dátum: 7. Január 2012.] http://www.ietf.org/rfc/rfc1112.txt. RFC 1112.

9. **Fenner, W.** Internet Group MAnagement Protocol, Version 2. *www.ietf.org.* [Online] November 1997. [Dátum: 8. Január 2012.] http://www.ietf.org/rfc/rfc2236.txt. RFC: 2236.

10. **Satrapa, P.** IPv6. [aut. knihy] Pavel Satrapa. *IPv6.* Tretie vydanie. Praha : CZ.NIC, z. s. p. o., 2011.

11. **Deering, S., Fenner, W. a Haberman, B.** Multicast Listener Discovery (MLD) for IPv6. *www.ietf.org.* [Online] Október 1999. [Dátum: 10. Január 2012.] http://www.ietf.org/rfc/rfc2710.txt. RFC 2710.

12. **Vida, R. a Costa, L.** Multicast Listener Discovery Version 2 (MLDv2) for IPv6. *www.ietf.org.* [Online] jún 2004. [Dátum: 10. Január 2012.] http://www.ietf.org/rfc/rfc3810.txt. RFC 3810.

13. **Solie, K. a Lynch, L.** CCIE PRactical Studies Volume II. [aut. knihy] Karl Solie a Leah Lynch. *CCIE Practical Studies Volume II.* Indianapolis : Cisco Press, 2003, 3.

14. **Metaswitch Networks.** PIM Overview. *network-technologies.metaswitch.com.* [Online] [Dátum: 10. Marec 2013.] http://network-technologies.metaswitch.com/multicast/what-is-pim.aspx.

15. **Stretch, J.** Source Specific Multicast (PIM-SSM). *packetlife.net.* [Online] 27. Júl 2010. [Dátum: 10. Marec 2013.] http://packetlife.net/blog/2010/jul/27/source-specific-multicast-pim-ssm/.

16. **Cisco.** Configuring IP Multicat Routing. *www.cisco.com.* [Online] [Dátum: 9. Január 2012.] http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfmulti.html.
17. —. Configuring Bidirectional PIM. *www.cisco.com.* [Online] [Dátum: 12. Marec 2013.] http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfbipim.html#wp1001390.
18. —. Configuring Source Specific Multicast. *www.cisco.com.* [Online] [Dátum: 12. Marec 2013.] http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfssm.html.
19. —. Implementing IPv6 Multicast. *www.cisco.com.* [Online] [Dátum: 9. Január 2012.] http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-multicast.html.
20. **OMNeT++.** OMNeT++ User Manual. *www.omnetpp.prg.* [Online] [Dátum: 10. Január 2012.] http://www.omnetpp.org/doc/omnetpp/manual/usman.html.
21. **Mateleško, P.** *Simulování multicastových přenosů v simulátoru OMNeT++.* bakalářská práce. Brno : FIT VUT v Brně, 2010.
22. **Černý, M.** *Modelování IPv6 v prostředí OMNeT++.* Brno : FIT VUT v Brně, 2011. diplomová práce.
23. **Cain, B., a iní, a iní.** Internet Group Management Protocol, Version 3. *tools.ietf.org.* [Online] Október 2002. [Dátum: 2013. Mája 13.] http://tools.ietf.org/html/rfc3376. RFC 3376.

Zoznam Skratiek

ACL – Access Control List

ANSA – Automated Network-wide Security Analysis

CLI – Command-line Interface

DM – Dense Mode

IANA – Internet Assigned Numbers Authority

ICMP – Internet Control Message Protocol

IGMP – Internet Group Management Protocol

IOS – Internetwork Operating System

IP – Internet Protocol

ISO/OSI – International Organization for Standardization/Open Systems Interconnection

LAN – Local Area Network

LDP – Label Distribution Protocol

MAC – Media Access Control

MLD – Multicast Listener Discovery

MPLS – Multiprotocol Label Switching

PIM – Protocol Independent Multicast

PPP – Point-to-Point Protocol

QRV – Querier's Robustness Variable

QQIC – Querier's Query Interval Code

RFC – Request for Comments

RP – Randezvous Point

RPF – reverse Path Forwarding

SCTP – Stream Control Transmission Protocol

SM – Sparse Mode

SSM – Source Specific Multicast

TCP – Transmission Control Protocol

TTL – Time to Live

UDP – User Datagram Protocol

Zoznam obrázkov

Prílohy

Konfiguračné súbory pre test IGMPv2

```
Building configuration...

Current configuration : 799 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Alice
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource policy
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name lab.local
!
interface FastEthernet0/0
 ip address 192.168.100.10 255.255.255.0
 ip igmp join-group 227.0.13.37
 ip igmp join-group 229.1.2.3
 duplex auto
 speed auto
!
interface FastEthernet0/1
```

```
 no ip address
 shutdown
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end

Building configuration...

Current configuration : 765 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Bob
!
boot-start-marker
boot-end-marker
```

```
!
no aaa new-model
!
resource policy
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name lab.local
!
interface FastEthernet0/0
 ip address 192.168.100.20 255.255.255.0
 ip igmp join-group 229.1.2.3
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
```

```
 login
!
End

Building configuration...
Current configuration : 738 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Carlos
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource policy
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name lab.local
!
interface FastEthernet0/0
 ip address 192.168.200.10 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
```

```
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
End

Building configuration...
Current configuration : 781 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname FITgate1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource policy
!
memory-size iomem 5
```

```
ip cef
!
no ip domain lookup
ip domain name lab.local
ip multicast-routing
!
interface FastEthernet0/0
 ip address 192.168.100.1 255.255.255.0
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
End

Building configuration...
```

```
Current configuration : 816 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname FITgate2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource policy
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name lab.local
ip multicast-routing
!
interface FastEthernet0/0
 ip address 192.168.100.2 255.255.255.0
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.200.1 255.255.255.0
 ip pim sparse-mode
 duplex auto
 speed auto
!
no ip http server
```

```
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
End
```

Konfiguračné súbory pre test IGMPv3

```
Building configuration...

Current configuration : 799 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Alice
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
```

```
resource policy
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name lab.local
!
interface FastEthernet0/0
 ip address 192.168.100.10 255.255.255.0
 ip igmp join-group 227.0.13.37 source 1.1.1.1
 ip igmp join-group 229.1.2.3 source 1.1.1.1
 ip igmp version 3
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
```

```
!
!
end

Building configuration...

Current configuration : 765 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Bob
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource policy
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name lab.local
!
interface FastEthernet0/0
 ip address 192.168.100.20 255.255.255.0
 ip igmp join-group 229.1.2.3 source 1.1.1.1
 ip igmp version 3
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
```

```
 shutdown
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
End

Building configuration...
Current configuration : 738 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Carlos
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
```

```
resource policy
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name lab.local
!
interface FastEthernet0/0
 ip address 192.168.200.10 255.255.255.0
 ip igmp join-group 229.1.2.3 source 1.1.1.1
 ip igmp version 3
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
```

```
End

Building configuration...
Current configuration : 781 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname FITgate1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource policy
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name lab.local
ip multicast-routing
!
interface FastEthernet0/0
 ip address 192.168.100.1 255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
```

```
 speed auto
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
End

Building configuration...

Current configuration : 816 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname FITgate2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource policy
```

```
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name lab.local
ip multicast-routing
!
interface FastEthernet0/0
 ip address 192.168.100.2 255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.200.1 255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
```

!

end

Konfiguračné súbory pre test OMNeT++ IGMPv2

Konfiguračný súbor pre deviceConfigurator

```
<Devices>
	<Router id="FITgate1">
		<Routing>
			<Multicast enable="1"></Multicast>
			<Static>
				<Route>
					<NetworkAddress>192.168.100.0</NetworkAddress>
					<NetworkMask>255.255.255.0</NetworkMask>
					<NextHopAddress>192.168.100.2</NextHopAddress>
				</Route>
			</Static>
		</Routing>
		<Interfaces>
			<Interface name="eth0">
		<IPAddress>192.168.100.1</IPAddress>
		<Mask>255.255.255.0</Mask>
				<Pim>
					<Mode>dense-mode</Mode>
				</Pim>
		</Interface>
	</Interfaces>
	</Router>
	<Router id="FITgate2">
		<Routing>
			<Multicast enable="1"></Multicast>
			<Static>
				<Route>
					<NetworkAddress>192.168.100.0</NetworkAddress>
					<NetworkMask>255.255.255.0</NetworkMask>
					<NextHopAddress>192.168.100.1</NextHopAddress>
				</Route>
			</Static>
		</Routing>
		<Interfaces>
			<Interface name="eth0">
		<IPAddress>192.168.100.2</IPAddress>
		<Mask>255.255.255.0</Mask>
				<Pim>
					<Mode>dense-mode</Mode>
				</Pim>
		</Interface>
		<Interface name="eth1">
		<IPAddress>192.168.200.1</IPAddress>
		<Mask>255.255.255.0</Mask>
				<Pim>
					<Mode>dense-mode</Mode>
				</Pim>
```

```
            </Interface>
        </Interfaces>
        </Router>
        <Host id="Alice">
            <Interfaces>
                <Interface name="eth0">
                    <IPAddress>192.168.100.10</IPAddress>
          <Mask>255.255.255.0</Mask>
          <MCastGroups>
                        <MCastGroup>227.0.13.37</MCastGroup>
                        <MCastGroup>229.1.2.3</MCastGroup>
                    </MCastGroups>
                </Interface>
            </Interfaces>
            <DefaultRouter>192.168.100.1</DefaultRouter>
        </Host>
        <Host id="Bob">
            <Interfaces>
                <Interface name="eth0">
                    <IPAddress>192.168.100.20</IPAddress>
          <Mask>255.255.255.0</Mask>
          <MCastGroups>
                        <MCastGroup>229.1.2.3</MCastGroup>
                    </MCastGroups>
                </Interface>
            </Interfaces>
            <DefaultRouter>192.168.100.1</DefaultRouter>
        </Host>
        <Host id="Carlos">
            <Interfaces>
                <Interface name="eth0">
                    <IPAddress>192.168.200.10</IPAddress>
          <Mask>255.255.255.0</Mask>
          <MCastGroups>
                        <MCastGroup>228.0.10.1</MCastGroup>
                    </MCastGroups>
                </Interface>
            </Interfaces>
            <DefaultRouter>192.168.200.1</DefaultRouter>
        </Host>
</Devices>
```

.ini súbor pre IGMPv2

```
[General]
network = inet.examples.ansa.IGMPv2test.IGMP
debug-on-errors = true

**.configFile = "config.xml"
IGMP.FITgate1.routingTable.forwardMulticast = true
IGMP.FITgate1.hostname = "FITgate1"
IGMP.FITgate1.deviceId = "FITgate1"
IGMP.FITgate2.routingTable.forwardMulticast = true
IGMP.FITgate2.hostname = "FITgate2"
IGMP.FITgate2.deviceId = "FITgate2"
IGMP.Alice.deviceId = "Alice"
IGMP.Bob.deviceId = "Bob"
IGMP.Alice.networkLayer.igmpType = "IGMPv2"
IGMP.Bob.networkLayer.igmpType = "IGMPv2"
IGMP.FITgate1.networkLayer.igmpType = "IGMPv2"
IGMP.FITgate2.networkLayer.igmpType = "IGMPv2"
IGMP.Carlos.networkLayer.igmpType = "IGMPv2"
IGMP.Carlos.deviceId = "Carlos"
IGMP.FITgate1.networkLayer.igmp.queryInterval = 60s
IGMP.FITgate2.networkLayer.igmp.queryInterval = 60s
```

.ini súbor pre IGMPv3

```
[General]
network = inet.examples.ansa.IGMPv2test.IGMP
debug-on-errors = true

**.configFile = "config.xml"
IGMP.FITgate1.routingTable.forwardMulticast = true
IGMP.FITgate1.hostname = "FITgate1"
IGMP.FITgate1.deviceId = "FITgate1"
IGMP.FITgate2.routingTable.forwardMulticast = true
IGMP.FITgate2.hostname = "FITgate2"
IGMP.FITgate2.deviceId = "FITgate2"
IGMP.Alice.deviceId = "Alice"
IGMP.Bob.deviceId = "Bob"
IGMP.Alice.networkLayer.igmpType = "IGMPv3"
IGMP.Bob.networkLayer.igmpType = "IGMPv3"
IGMP.FITgate1.networkLayer.igmpType = "IGMPv3"
IGMP.FITgate2.networkLayer.igmpType = "IGMPv3"
IGMP.Carlos.networkLayer.igmpType = "IGMPv3"
IGMP.Carlos.deviceId = "Carlos"
IGMP.FITgate1.networkLayer.igmp.queryInterval = 60s
IGMP.FITgate2.networkLayer.igmp.queryInterval = 60s
```