

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2018

Bc. Adrián Bárdossy



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

MULTIPLATFORMNÍ APLIKACE PRO SPRÁVU SÍŤOVÝCH PRVKŮ MIKROTIK

MULTIPLATFORM APPLICATION FOR MIKROTIK NETWORK DEVICES MANAGEMENT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Adrián Bárdossy

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ondřej Krajsa, Ph.D.

BRNO 2018

Diplomová práce

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

Ústav telekomunikací

Student: Bc. Adrián Bárdossy

ID: 154674

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Multiplatformní aplikace pro správu síťových prvků Mikrotik

POKYNY PRO VYPRACOVÁNÍ:

Vytvořte interaktivní aplikaci pro hromadnou správu sítě založené na aktivních prvcích Mikrotik. Aplikace bude využívat Mikrotik API-SSL, uživatelské rozhraní bude realizováno v jazyce Python a přenositelné mezi různými operačními systémy.

DOPORUČENÁ LITERATURA:

[1] BURGESS, Dennis. Learn RouterOS. [Lexington]: Dennis Burgess, 2009, 391 s. : il. ISBN 978-0-557-09271-0.

[2] ROMANO, Fabrizio, Dusty PHILLIPS a Rick van HATTEM. Python. Birmingham: Packt Publishing, 2016.

Termín zadání: 5.2.2018

Termín odevzdání: 21.5.2018

Vedoucí práce: Ing. Ondřej Krajsa, Ph.D.

Konzultant:

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

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

KLÍČOVÉ SLOVÁ

ABSTRACT

KEYWORDS

BÁRDOSSY, Adrián *Multiplatformní aplikace pro správu síťových prvků Mikrotik*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2018. 85 s. Vedúci práce bol Ing. Ondřej Krajsa, Ph.D

PREHLÁSENIE

Prehlasujem, že som svoju diplomovou prácu na tému „Multiplatformní aplikace pro správu síťových prvků Mikrotik“ vypracoval(a) samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisiacich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi Ing. Ondřejovi Krajsovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno

.....

podpis autora(-ky)

POĎAKOVANIE

Výzkum popsaný v tejto diplomovej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno

.....
podpis autora(-ky)

OBSAH

1	Úvod do diplomovej práce	13
2	Mikrotik a RouterOS (SwitchOS)	14
2.1	Mikrotik API	14
2.1.1	Požiadavky na použitie API	14
2.1.2	Porty	14
2.1.3	Základný port 8728	14
2.1.4	SSL port 8729	15
2.2	API slová	16
2.3	Príkazové slová API	16
2.4	Použitie atribútov v príkaze a filtrovanie	16
2.5	Špeciálne slová API	17
3	Pripojenie na Mikrotik	18
3.1	Možnosti pripojenia	18
3.2	Pripojenie pomocou winboxu	18
3.3	Pripojenie pomocou webfigu	19
3.4	Mactelnet	19
3.5	Pripojenie pomocou telnet a SSH	20
3.5.1	Pripojenie cez telnet	20
3.5.2	Pripojenie pomocou ssh	20
4	Programovací jazyk Python	22
4.1	Python 2	22
4.2	Python 3	22
4.3	Prostredia na programovanie v jazyku Python	23
4.4	Pycharm	23
5	Použité knižnice v diplomovej práci	25
5.1	OS.SYSTEM	25
5.2	Telnetlib	26
5.3	Pxssh a pexpect	27
5.3.1	Inštalácia pexpect	28
5.4	TikApy	29
6	Konzolová časť aplikácie na správu mikrotikov	31
6.1	Popis naprogramovanej časti prihlasovania na mikrotik	31
6.1.1	Súbor centralControl	32

6.1.2	Súbor Constructors	34
6.1.3	Súbor dhcpClient	35
6.1.4	Súbor LoginManager	37
6.2	Rozbor hlavnej časti backendu	40
7	Hlavná časť backendu	43
7.1	Zložka bridge	43
7.1.1	Popis tried zložky	43
7.1.2	Vybraný analyzovaný súbor	44
7.2	Zložka capsman	45
7.2.1	Popis tried zložky	46
7.2.2	Vybraný analyzovaný súbor	48
7.3	Zložka Dude	48
7.3.1	Popis tried zložky	49
7.3.2	Analýza vybraného súboru	49
7.4	Zložka Interfaces	50
7.4.1	Popis tried zložky	50
7.4.2	Analýza vybraného súboru	52
7.5	Zložka IPv4	53
7.5.1	Popis tried zložky	54
7.5.2	Analýza vybraného súboru	58
7.6	Zložka IPv6	58
7.6.1	Popis tried zložky	58
7.6.2	Analýza vybraného súboru	60
7.7	Zložky KVM, log a makeSupportFile	61
7.7.1	Popis triedy zložky KVM	61
7.7.2	Popis triedy zložky log	62
7.7.3	Popis triedy makeSupportFile	62
7.8	Zložka Mesh	63
7.8.1	Popis tried zložky	63
7.8.2	Analýza vybraného súboru	63
7.9	Zložka MPLS	65
7.9.1	Popis tried zložky	65
7.9.2	Analýza vybraného súboru	67
7.10	Zložka PPP	67
7.10.1	Popis tried zložky	68
7.10.2	Analyzovaný súbor	70
7.11	Zložka Queues	71
7.11.1	Zoznam tried zložky	71

7.11.2	Analáza vybraného súboru	72
7.12	Zložka Radius	73
7.13	Zložka routing	75
7.13.1	Zoznam tried zložiek	75
7.13.2	Analýza vybraného súboru	76
7.14	Zložka switch	77
7.14.1	Zoznam tried zložky	77
7.14.2	Analýza vybraného súboru	78
7.15	Zložka System	79
7.15.1	Zoznam súborov zložky	79
7.15.2	Analýza vybraného súboru	80
Literatúra		82
Zoznam symbolov, veličín a skratiek		84
Zoznam príloh		85

ZOZNAM OBRÁZKOV

3.1	Winbox základné prihlasovacie rozhranie	18
3.2	Webfig základné prihlasovacie rozhranie	19
3.3	Výstup príkazu mactelnet	19
3.4	Prihlásenie na mikrotik pomocou príkazu SSH	21
4.1	Rozhranie IDE Pycharm Professional Edition	24
6.1	Zoznam základných konfiguračných súborov	31
6.2	Ukážka konštruktorov projektu	35
6.3	Štruktúra projektu konzolovej časti projektu	42
7.1	Zoznam súborov zložky bridge	43
7.2	UML diagram vybraného súboru bridgeArp	45
7.3	Zoznam súborov zložky capsman	46
7.4	UML diagram triedy configRates	48
7.5	Zoznam súborov adresáru Dude	49
7.6	UML diagram knižnice Devices	50
7.7	Zoznam súborov adresáru Interfaces	52
7.8	UML diagram greTunnelSet triedy	53
7.9	Štruktúra časti zložky IPv4	57
7.10	UML diagram triedy FirewallAddressist	58
7.11	Zoznam súborov zložky IPv6	60
7.12	UML diagram triedy FirewallMangle	61
7.13	Zoznam súborov zložky KVM	61
7.14	UML diagram triedy KVM	62
7.15	Zoznam súborov zložky log	62
7.16	UML diagram triedy log	62
7.17	UML diagram triedy makeSupport	63
7.18	Zoznam súborov zložky mesh	63
7.19	UML diagram triedy MeshInterfaces	65
7.20	Zoznam súborov zložky MPLS	66
7.21	UML diagram triedy MplsLocalBindings	67
7.22	Zoznam tried v zložke PPP	70
7.23	UML diagram triedy interfacePppoeGeneralSet	71
7.24	Zoznam súborov zložky Queues	72
7.25	UML diagram triedy QueueTree	73
7.26	UML diagram triedy Radius	75
7.27	Zoznam súborov zložky Routing	76
7.28	UML diagram súboru RipNetworks	77
7.29	Zoznamsúborov zložky switch	78

7.30	UML digram triedy switchPorts	78
7.31	Zoznam súborov triedy system	80
7.32	UML diagram triedy SystemMaintenance	81

ZOZNAM TABULIEK

7.1	Tabulka zoznamu metód triedy bridgeArp	44
7.2	Popis triedy configRates	48
7.3	Tabulka metód triedy Devices	50
7.4	Tabulka metód triedy greTunnelSet	53
7.5	Obsah triedy FirewallAdressist	58
7.6	Tabulka triedy FirewallMangle	60
7.7	Tabulka metód triedy KVM	61
7.8	Tabulka metód triedy log	62
7.9	Tabulka metód triedy makeSupport	63
7.10	Tabulka metód v triede MeshInterfaces	64
7.11	Tabulka zoznamu metód triedy MplsLocalBindings	67
7.12	Tabulka popisu metód triedy interfacePppoeGeneralSet	71
7.13	Tabulka zoznamu metód triedy QueueTree	72
7.14	Tabulka metód triedy Radius	74
7.15	Tabulka popisu metód triedy RipNetworks	77
7.16	Tabulka zoznamu metód triedy switchPort	78
7.17	Tabulka metód triedy SystemMaintenance	81

1 ÚVOD DO DIPLOMOVEJ PRÁCE

Diplomová práca na tému "Multiplatformní aplikace pro správu síťových prvků Mikrotik" sa bude primárne zaoberať samostatným mikrotikom. Primárne pomocou application programmable interface (API) vytvorenie jej konzolovej časti (backendu) a grafickej časti (frontendu). Tieto dve časti dajú celkovú aplikáciu dokopy ako celok.

V prvej časti práci bude definovanie Mikrotik API a jeho možností, porovnanie podobnosti s operačným systémom unix. Ďalej budú popísané možnosti zabezpečenia API pomocou secure socket layer (SSL). Budú tu tiež spomenuté použité porty, a ďalšie možnosti.

V druhej časti práce bude popis API a spôsoboch softvérového riešenia aplikácie pre správu Mikrotikov. Táto časť bude tiež obsahovať niečo ohľadom technológie git, popise, čo je git, princíp tzv. commitu a pushu. Rozdiely medzi vetvami, prepínanie medzi vetvami a pridávanie zmien. Taktiež tu bude spomenutý aj úvod do certifikátov a to konkrétne Single Sign-on metódy.

V ďalšej časti bude návrh riešenia softvérovej implementácie aplikácie. Bude obsahovať popis, princípy, diagramy, hlavne Unified modeling language (UML), popisy knižníc, jednotlivých tried a modulov. Každý modul bude popísaný svojou funkcionalitou, parametrami a výstupom s praktickými ukážkami.

V ďalšej časti bude použitá implementácia softvérového návrhu riešenia. Bude tu riešenie ako v konzolovej časti, jeho ukážky, test a výsledky.

V poslednej časti práce bude ukážka grafického spracovania konzolovej časti aplikácie a ich prepojenia do jednej aplikácie, spoločne s ukážkami kódov, testu a výsledkov.

2 MIKROTIK A ROUTEROS (SWITCHOS)

V dnešných malých a stredne veľkých firmách sa na správu siete používajú prevažne routre a switche typu Mikrotik. Mikrotik je firma vyvíjajúca routre a switche, prístupové body a ďalšie sieťové prvky v Litve.

Mikrotik zariadenia používajú operačný systém routerOS, prípadne switchOS. Rozdiel medzi nimi je na základe použitého zariadenia. Čo sa týka routrov, používa operačný systém routerOS, switch používa switchOS, v prípade prístupových bodov (AP) je to routerOS.

2.1 Mikrotik API

Za pomoci Mikrotik API môžeme programovať užívateľské programy a prostredia na riadenie a konfiguráciu Mikrotik zariadení. V dnešnej dobe existuje softvér na konfiguráciu mikrotik zariadení a to pod názvom **Winbox**. Winbox v dnešnej dobe existuje len na operačný systém Windows a Macintosh (MAC). Bohužiaľ na operačný systém Linux winbox samostatne neexistuje a musí sa simulovať pomocou emulátoru Windows aplikácií za pomoci programu Wine. Toto spôsobuje komplikácie pri použití niektorých funkcií winboxu ale aj iných programov operačného systému Windows. Výstupom práce bude práve Graphical User Interface (GUI).

2.1.1 Požiadavky na použitie API

- Verzia routerOS verzie *3.0.X* a vyššie [1]

2.1.2 Porty

Základné porty na použitie Mikrotik API [1] sú:

- **API port:** 8728
- **Application programable interface Secure Socket Layer (API-SSL)**
port: 8729

2.1.3 Základný port 8728

Na základné pripojenie k API aplikácii na prvku Mikrotik musí byť povolený port 8728, ktorý tiež nájdeme v IP-> Services spoločne s API-SSL.

Na základné pripojenie nie je potreba žiadneho transport layer security (TLS) certifikátu. Stačí jednoducho napísať kód a skompilovať ho.

2.1.4 SSL port 8729

Pre použitie portu 8729 tiež známeho ako API-SSL portu je potreba zabezpečenej komunikácie pomocou SSL protokolu.

Primárne musí byť nastavený port, základný port 8729 v IP -> Services. Môžeme ale definovať aj užívateľsky definovaný port.

Možnosti nastavenia API-SSL:

- prístup bez certifikátu TLS
- prístup pomocou certifikátu TLS

Prístup pomocou certifikátu TLS

Pre použitie certifikátu TLS je potrebné vygenerovať certifikát TLS, a to na certifikačnej autorite alebo na ľubovoľnej linux stanici ideálne, ale tiež to dokážeme spraviť aj na Windows stanici či MAC. Spôsoby vygenerovania certifikátov:

- openssl
- easy-rsa
- Windows Server Certificate Services

Openssl

Openssl [4] je softvér na generovanie certifikátov pre komunikáciu v počítačovej sieti. Koreňovo sa používa na prístup na web skrz protokol Hyper Trasfer Transport Protocol Secure (HTTPS). Pre vygenerovanie certifikátov sa musí vygenerovať:

- certifikát **.crt*
- certifikačný požiadavok **.csr*
- kľúč k certifikátu **.key*

Easy-rsa

Softvér easy-rsa [2] sa používa na vytvorenie open-source certifikačnej autority a užívateľských certifikátov napr. pre potreby HTTPS spojenia.

Po nainštalovaní easy-rsa napr. na Ubuntu príkazom *sudo apt install easy-rsa* sa musí spraviť nasledovné:

- Nakopírovanie konfiguračných súborov do zložky autority
- Vytvorenie šablóny na vygenerovanie certifikačnej autority
- Vytvorenie užívateľských certifikátov

Active Directory Certificate Services

Windows riešenie [18] pre generovanie certifikačnej autority je inštalácia roly servera Active Directory Certificate Services.

Pre použitie certifikačnej autority na Windows servery je potreba:

- Inštalácia role serveru
- Nadefinovanie certifikačnej autority
- Generovanie certifikátov

2.2 API slová

API slová [1] sú základnou časťou API "vety". API "veta" predstavuje príkaz v použití príkazu napr. `/ip/address/print`, `/ip/address/add address="10.1.1.1/24"interface=ether1`. Parametre na slová:

- každé slovo má svoju zakódovanú dĺžku t.j.
 - 0 - 127 bitov zaberá 1 Byte
 - 128 - 1023 bitov zaberá 2 Byty
 - 1024 bitov - 2097 kib zaberá 3 Byty
 - viac ako 2098 kib zaberá 4 Byty
- jednotlivé slová súzaraďené do viet
- maximum bztov na slovo sú 4 Byty
- kontrolné byty sa nepoužívajú

2.3 Príkazové slová API

Slová Mikrotik API sa zaraďujú do API viet použitím API slov, na ktoré platia požiadavky, ktoré sú spomenuté v kapitole 2.2. Na použitie API viet je potreba začínať znakom `/`. Napr. miesto `ip address print` sa použije `/ip/address/print`.

Pre úplnosť API viet musí platiť [1]

- zakódovaná dĺžka slova
- slovo musí začínať znakom `/`
- musí byť použitá správna syntax

2.4 Použitie atribútov v príkaze a filtrovanie

V prípade konfigurácie mikrotik zariadení sa pre nastavenie jednotlivých prvkov používajú tzv. atribúty [1] napr. ip adresa, číslo pravidla, meno rozhrania, nastavenie virtuálnej lokálnej siete (VLAN).

Použitie atribútov má špeciálnu syntax pre konfiguráciu prípadne zmenu prvku na mikrotiku, prípadne pridanie a zmazanie prvku. Na použitie atribútov sa použije špeciálny znak `=`. Napr. `/ip/address/add =address=10.1.1.1/24 =interface=ether1`. Pre filtrovanie prvkov v rámci mikrotik API syntaxe sa používa špeciálny atribút

parameter so znakom `?`. Napr. `/ip/address/print =?type=ether1` vyfiltruje len rozhranie ether1.

2.5 Špeciálne slová API

Mikrotik API má možnosť tzv. špeciálnych slov [1]. Špeciálne slová sú slová, ktoré sú rezervované a nesmú sa použiť pre iné použitie ako napríklad meno premennej, metódy, triedy, a iné. Medzi špeciálne slová patria:

- prihlásenie sa na zariadenie `/login`
- ukončenie spojenia na zariadenie `/cancel`
- odhlásenie sa zo zariadenie `/logout`
- získanie všetkých parametrov `/getall`

3 PRIPOJENIE NA MIKROTIK

3.1 Možnosti pripojenia

Pripojenie na mikrotik je realizované pomocou niekoľkých typov softvéru:

- **Winbox** - základný softvér na konfiguráciu mikrotiku
- **Webfig** - konfigurácia mikrotiku pomocou webového rozhrania štandardne na portoch 80 a 443
- Riadenie mikrotiku pripojením na mac adresu - **mactelnet**
- Pripojenie pomocou protokolu **SSH** - zabezpečené a šifrované spojenie
- Pripojenie pomocou protokolu **telnet** - nebezpečné v dnešnej dobe

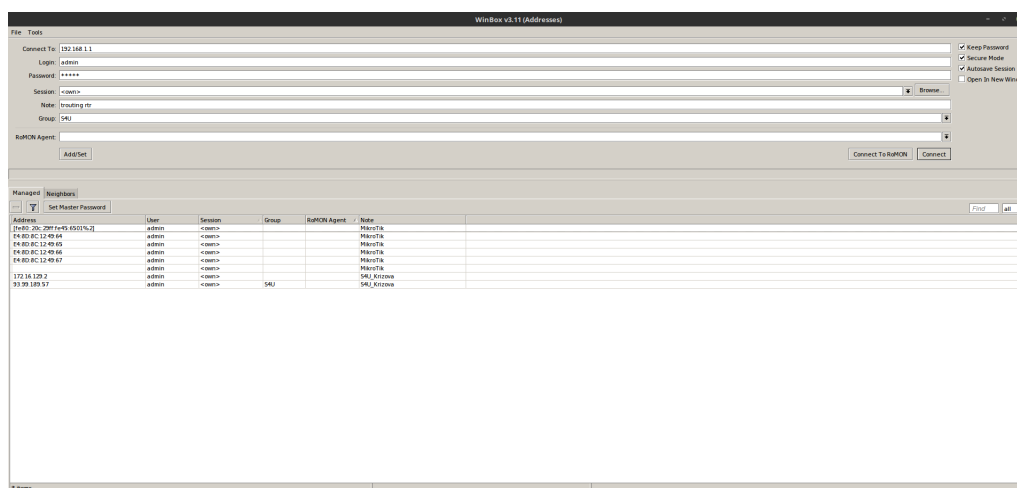
3.2 Pripojenie pomocou winboxu

Winbox[17] je nástroj na administráciu mikrotiku. Medzi jeho vlastnosti patrí:

- GUI nástroj (klikátko)
- rýchlosť
- spoľahlivosť

Winbox je prepis konzolovej aplikácie do grafickej. Obsahuje tiež nástroje ktoré sa v konzole nedajú odsimulovať napr. graphs, torch, netmon, scheduler,...

Niektoré funkcie nevieme meniť pomocou winboxu napr. Media Access Control (MAC) adresu rozhrania.



Obr. 3.1: Winbox základné prihlasovacie rozhranie

Režimy winboxu:

- jednoduchý režim - obsahuje na pripojenie len užívateľské meno, heslo a adresu mikrotiku

- pokročilý režim - možnosť pridania skupiny mikrotikov, popisky a názov spojenia

3.3 Pripojenie pomocou webfigu

Webfig[14] je webová aplikácia RouterOS a umožňuje konfiguráciu, minitoring a údržbu prvkov RouterOS. Medzi hlavné tasky webfigu patrí:

- konfigurácia mikrotiku
- mnotring mikrotiku
- riešenie problémov na mikrotiku za pomoci webového rozhrania



Obr. 3.2: Webfig základné prihlasovacie rozhranie

3.4 Mactelnet

Mactelnet[3] predstavuje aplikačný protokol riadený na druhej vsrte referenčného modelu. Tiež predtavuje kombináciu winboxu a telnetu v jednom protokole. Riadi prístup na napr. nový mikrotik, ktorý ešte neobsahuje žiadnu konfiguráciu. Pracuje absolútne rovnakým spôsobom ako telnet. Je možné sa pripojiť len na fyzicky pripojený mikrotik pomocou mactelnet, vzdialený prístup pomocou mactelnet nie je možný.

```
adrian@adrian-Lenovo-Z50-75 ~/Desktop/diplomka2016/bakalaris $ mactelnet -l
Searching for MikroTik routers... Abort with CTRL+C.

IP           MAC-Address  Identity (platform version hardware) uptime
192.168.1.1  64:d1:54:53:59:72 MikroTik (MikroTik 6.41.2 (stable) R8750r2) up 10 days 23 hours GMB8-QCUB ether2
```

Obr. 3.3: Výstup príkazu mactelnet

Po pripojení na mikrotik pomocou mactelnet sa nastaví základná konektivita a pripájame sa potom na základe Internet Protocol (IP) adresy.

3.5 Pripojenie pomocou telnet a SSH

Ďalšou možnosťou pripojenia na mikrotik je prihlásenie sa pomocou telnetu[12] prípadne SSH[11] na konzolu mikrotiku. Napríklad na nastavenie fronty, firewallu,... .

3.5.1 Pripojenie cet telnet

Telnet predstavuje protokol, ktorý umožňuje pripojenie na vzdialené servery. Jeho štandardným portom je port 23.

Na povolenie pripojenia pomocou telnetu je potrebné povoliť službu telnet na mikrotiku v IP -> Services. Pre bezpečnostné účely by sa telnet nemal používať, je terčom útokov nakoľko je nešifrovaný. Pokiaľ chceme povoliť telnet na pripojenie na mikrotik, by sa mal minimálne zmeniť štandardný port z 23 na užívateľsky definovaný port.

Príklad príkazu na pripojenie na zariadenie pomocou telnetu: *telnet <IP adresa> <port>*

3.5.2 Pripojenie pomocou ssh

SSH predstavuje protokol, ktorý umožňuje vzdialené pripojenie pomocou tohoto protokolu. Používa štandardný port 22. Tak isto ako u telnetu, pre SSH platí to isté, je potrebné ho povoliť v IP -> Services. SSH na rozdiel od telnetu je ale šifrovaný a zabezpečený protokol. SSH predstavuje bezpečnú verziu telnetu. Je možné si zabezpečiť SSH prístup na bezpečnejší, a to tak, že sa budú porovnávať verejný a súkromný kľúč certifikátu TKIP. Vstupom pripojenia SSH na mikrotik je na obrázku 3.4.

```

adrian@adrian-Lenovo-Z50-75 ~/Desktop/diplomka2016/bakalaris $ ssh admin@192.168.1.1
The authenticity of host '192.168.1.1 (192.168.1.1)' can't be established.
RSA key fingerprint is SHA256:SRyPppD9fkv88z2HmUqZoCZ4UNreKZtehIpxusH0rFg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.1' (RSA) to the list of known hosts.
admin@192.168.1.1's password:

MMM      MMM      KKK      TTTTTTTTTT      KKK
MMMMM    MMMMM    KKK      TTTTTTTTTT      KKK
MMM MMMM MMM III KKK KKK RRRRRR 000000 TTT III KKK KKK
MMM MM MMM III KKKKK RRR RRR 000 000 TTT III KKKKK
MMM MMM III KKK KKK RRRRRR 000 000 TTT III KKK KKK
MMM      MMM III KKK KKK RRR RRR 000000 TTT III KKK KKK

MikroTik RouterOS 6.41.2 (c) 1999-2018 http://www.mikrotik.com/

[?] Gives the list of available commands
command [?] Gives help on the command and list of arguments

[Tab] Completes the command/word. If the input is ambiguous,
a second [Tab] gives possible options

/ Move up to base level
.. Move up one level
/command Use command at the base level

[admin@MikroTik] > █

```

Obr. 3.4: Prihlásenie na mikrotik pomocou príkazu SSH

4 PROGRAMOVACÍ JAZYK PYTHON

Python je interpretovaný, interaktívny, objektovo-orientovaný a vysoko-úrovňový programovací jazyk. Jazyk Python bol vytvorený pánom Guido van Rossum v Wiskundskom centre informatiky v 80-tych rokoch.

Medzi jeho vlastnosti patrí:

- dynamické typovanie
- konzolové aplikácie
- objektové aplikácie
- všetko v pythone je objekt
- jednoduchá syntax
- biele znaky sú súčasťou jazyka
- dynamické typy premenných
- široká škála knižníc
- dokumentácia na vysokej úrovni
- používaný na webové aplikácie, strojové učenie, teórie zložitosti,...

Verzie jazyku Python:

- Python verzia 2
- Python verzia 3

4.1 Python 2

Vlastnosti jazyku Python 2[10]

- automatická spáva pamäti (garbage collector)
- podporuje viac vstupných paradigiem
- Volanie niektorých príkazov je odlišné od Python 3
- referenčný interpret sa nazýva CPython a spravuje ho organizácia Python Software Foundation
- Súčasne sa používa Python vo verzii 2.7.2

4.2 Python 3

Vlastnosti jazyku Python 3[15]

- V niektorých častiach syntaxe v porovnaní s jazykom Python 2 je trochu odlišná (napr. príkaz print,...)
- Od verzie 3.6 má premenná typu slovník interné zachovávané poradie vkladá-ných prvkov
- Pridanie anotácií cez metatriedy

- deklarácia nelokálnej premennej vonku z funkcie
- Slová typu True, False a None sú rezervované slová
- Mnoho vlastností ma rovnakých ako Python 2
- Miesto <> sa voverzii 3 používa relačný operátor !=
- od Júla 2018 by mala výjsť verzia Python 3.7 s ďalšími novinkami

4.3 Prostredia na programovanie v jazyku Python

Na realizovanie python programu je nutnosť mať nainštalovaný softvér na kompiláciu softvéru napísaného v jazyku Python. Na tieto účely slúži tzv. intergrated developement environroment (IDE). Ecistuje niekoľko ciet aj mimo IDE ako spustiť kód napísaný v jazyku python.

- Napísanie kódu napr. v textovom editore typu nano, vim, gedit ale aj windows riešenie ako napr. notepad
- nainštalovaný python kompilátor
- spustenie programu príkazom python <názov.py>

4.4 Pycharm

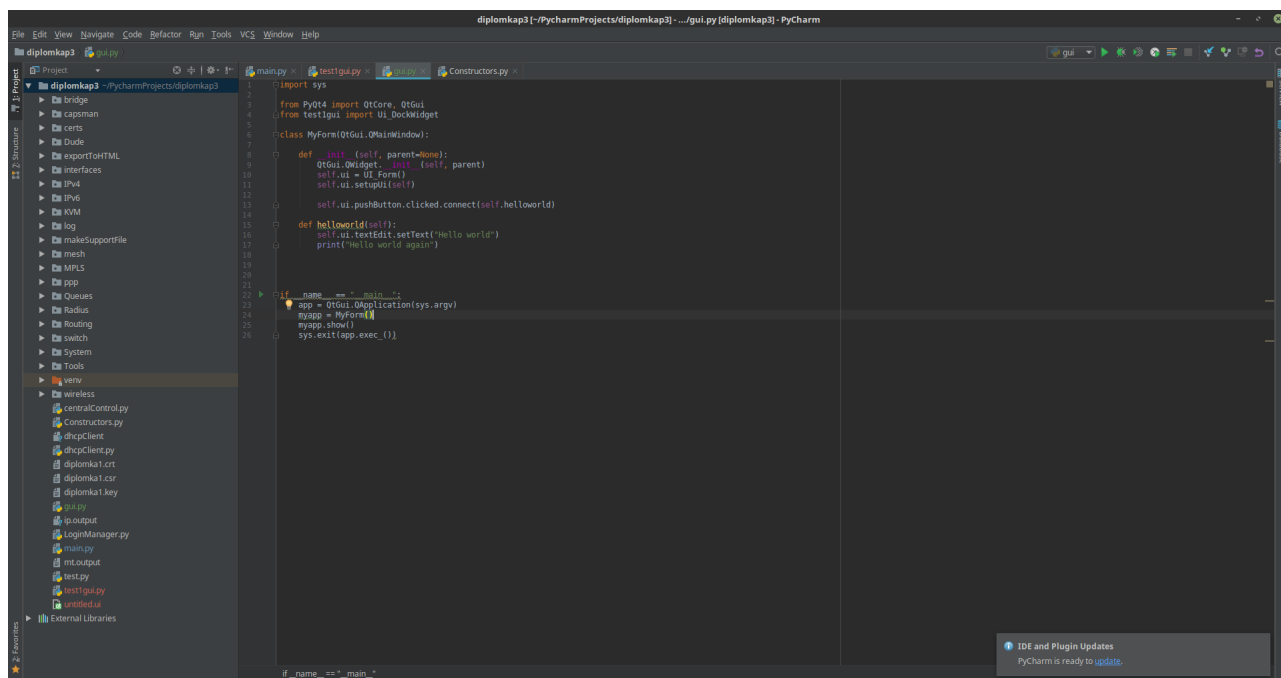
Pycharm predsatvuje IDE na pokročilé aplikácie napísané v jazyku Python. Exustuje v dcoch verziách:

- Pycharm Community Edition - voľne dostupné, nelicencované, neobsahuje niektoré doplnky professional verzie
- Pycharm Professional Edition - licencované, voľne dostupné na 30 dní, licencované, plný prístup ku všetkým doplnkom

Na nainštalovanie pycharm ľubovoľnej verzie je potreba:

- Stiahnutie bin respektíve exe súboru inštalátoru
- Napojenie na pracovný adresár projektu
- Napojenie na tzv. environroment, to je použitie cesty k volaniu príkazu python respektíve python3

Po spustení programu Pycharm sa vytvorí projekt, kde po jeho inicializácii nájdeme podobný výstup.



Obr. 4.1: Rozhranie IDE Pycharm Professional Edition

Jednou z najväčších výhod je generovanie UML diagramov z kódu.

5 POUŽITÉ KNIŽNICE V DIPLOMOVEJ PRÁCI

5.1 OS.SYSTEM

Modul operačný systém (OS) [5] je zahrnutý v rámci štandardných knižníc jazyku python. Jeho hlavnou výhodou je použitie príkazov operačného systému na ktorom beží python. Najčastejšie sa volá príkazom `os.system('príkaz')`. Použitie modulu aplikujem v rámci diplomovej práce. Používa sa hlavne pri vyhľadávaní a pripájaní sa na mikrotiky pomocou protokolu mactelnet. Jeho výstup môžeme aplikovať na:

- Štandardný výstup do konzoly
- Výstup do osobitného súboru, ktorý sa potom ďalej spracuje

V rámci práce som tento modul použil v rámci knižnice `loginManager`, kde v metódach pre mactelnet vyhľadáva mikrotiky za pomoci protokolu mactelnet a ukladá to do textového súboru `mt.output`.

Nie je potreba inštalácie modulu OS, pretože je zahrnutý v rámci štandardných knižníc jazyku python. V kóde vidíme ukážku metódy, ktorá vylistuje zoznam mikrotik zariadení za pomoci funkcie mactelnet a jej návratovou hodnotou je list zoznamu mikrotik zariadení

Listing 1: Použitie knižnice `os.system`

```
def listMikrotikDevices(self):
    deviceList = []
    loadMacAddress = False
    os.system("mactelnet_l-t_20_2>&l_>_mt.output")
    with open( "mt.output", "r" ) as file:
        for line in file:
            if loadMacAddress:
                macAddress = line.split( )[1]
                deviceList.append( macAddress )
            else:
                header = line.split( )
                if len( header ) > 1:
                    if "IP" in header[0] and "MAC-Address" in header[1]:
                        loadMacAddress = True
    return deviceList
```

5.2 Telnetlib

Telnetlib [13] obdobne ako OS knižnica je vstavaná knižnicaprogramovacieho jazyku python. Knižnice *telnetlib* implementuje protokol telnet do pythonu, definovaného referenčným modelom RFC 854. V rámci definície modulu telnetlib sa používa hlavíkový súbor *telnet.h* s odstráneným záhlavým obsahujúcim *TELOPT_*.

Modul telnetlib predstavuje jednoduchého telnet klienta pripájajúceho sa na telnet server. Na vytvorenie spojenia je potreba nasledujúcich krokov:

- vytvorenie objektu telnetlib s parametrami *host*, čo predstavuje IP adresu telnet serveru, *port*, štandardne 23, nepovinným parametrom je *timeout*.
- Je nutné otvoriť spojenie metódou *open()*
- ďalej metódami *readuntil()* a *write()* vyžadujeme očakávaný výstup a odoslanie dát na server (príkazov)
- potom ukončíme spojenie metódou *close()*

Medzi najpoužívanéjšie metódy v rámci diplomovej práce sú použité:

- *Telnet()*
- *readuntil()* - očakávanie výstupu serveru
- *write()* - zápis príkazov
- *sleep()* - doba trvania odoslania príkazu v sekundách
- *close()* - ukončenie spojenia

V rámci výstupu kódu je ukážka metódy na pripojenie na telnet server pomocou knižnice telnetlib.

Listing 2: Použitie knižnice telnetlib

```
def loginTelnet(self, password, login="admin"):
    import telnetlib
    central = centralControl(login, password)
    server_list = central.listMikrotikDevices()
    print(server_list)
    for server in server_list:
        try:
            telnetcon = telnetlib.Telnet( host=server, port=23 )
            telnetcon.read_until( b"Login:_" )
            telnetcon.write( login.encode( ) + "\n" )
            telnetcon.read_until( b"Password:_" )
            telnetcon.write( password.encode( ) + b"\n" )
            time.sleep( 10 )
            telnetcon.close( )
        except:
            print( "Cannot_connect_to_router_via_telnet" )
```

5.3 Pxssh a pexpect

Moduly pexpect[7] a jeho submodul pxssh[9] sú knižnice, ktoré slúžia na vyžadovanie určitého výstupu zariadenia, na to slúži knižnica *pexpect* a na pripojenie sa na server z pythonu pomocou protokolu SSH slúži knižnica *pxssh*

Pexpect je čistá python modulna kontrola a riadenie aplikácií. Pozostáva z dvoch krokov:

- Vyžadovanie výstupu
- Odoslanie požadovaného výstupu

Pexpect môže byť použitý na interaktívne aplikácie, ktoré používajú protokoly SSH, File transfer protocol (FTP), telnet, atď. Pre implementáciu pexpect nie je potreba importovania knižníc z jazyka C na skompilovanie do jadra. Pracujú na všetkých platformách, a to v podobe štandardného vstupu a výstupu v príkazovom riadku operačného systému, či už serveru ale aj klienta. Pexpect je jednoduchýna implementáciu.

Pxssh predstavuje submodul modulu pexpect. Na zavolanie submodulu pxssh je nutné prednostne zavolať metódu *spawn()*. Po vytvorení spojenia metódou *spawn()* je nutné použiť metódy *login()*, *spawn()* a *logout()*.

5.3.1 Inštalácia pexpect

Pexpect je súčasťou sady nástrojov Pypi. Aktuálna verzia modulu pexpect je verzia

4.4. Požiadavky na softvér:

- Python vo verzii 2.7 alebo 3.3 a vyššie
- pre windows je potreba inštalácie modulu POSIX pre jeho funkčnosť

Na nainštalovanie pexpect [8] na linuxe, sa v príkazovom riadku zadá:

Listing 3: Inštalácia Pexpect

```
pip install pexpect
pip3 install pexpect
```

Pre inštaláciu na operačnom systéme Windows je potreba mať nainštalovaný program python prípadne python3 externe, pretože nie je štandardne zahrnutý v balíčkoch operačného systému.

Modul pexpect zahŕňa modul pxssh a nie je potreba ho potom extra inštalovať.

Nižšie vidíme ukážku použitia kombinácie modulov pexpect a pxssh na pripojenie na mikrotik.

Listing 4: Inštalácia Pexpect

```
def loginSSH(self, server, login, password):
    from pexpect import pxssh, spawn, expect
    import getpass
    try:
        connect = pxssh.pxssh( )
        server = '172.16.49.2'
        login = 'admin'
        password = 'admin'
        port = 22
        connect.login( server, login, password )
        commands = pxssh.spawn( )
        time.sleep( 10 )
    except pxssh.ExceptionPxssh as e:
        print( "Error" )
        print( str( e ) )
```

5.4 TikApy

Na správu mikrotik smerovačov je potreba implentovať do pythonu modul tikapy[16]. Modul tikapy funguje voverzii pythonu 3 a vyššie. Podobne ako bolo spomenuté v kapitole 2.2, API pracuje na základe tzv. "slov". Slová predstavujú jednotlivé príkazy na mikrotiku. Tieto príkazy budú popísané v ďalších kapitolách diplomovej práce. Modul tikapy ako celkovo mikrotik API komunikuje nezabezpečene na porte 8728 a pomocou API-SSL na porte 8729. Na pripojenie sa na mikrotik pomocou modulu tikapy, ktorýmá v sebe zahrnutých mnoho knižníc na komunikáciu s mikrotikom. Medzi najčastejšie patria:

- vytvorenie objektu TikapyClient prípadne TikapySSLClient
- *login()* - prihlásenie sa na mikrotik pomocou API
- *talk()* - odosielanie príkazov na mikrotik
- *close()* - ukončenie spojenia

Na nainštalovanie tikapy do pythonu je potreba to nainštalovať nasledovne, v príkazovom riadku zadáme príkaz:

Listing 5: Inštalácia tikapy

```
pip install tikapy
```

V rámci vytvorenia objektu sú dve možnosti:

- vytvorenie klienta a to buď SSL klienta prípadne štandardného klienta TikapyClient(adresa,port), pre SSL klienta to je port 8729, pre štandardného klienta port 8728
- vytvorenie metódy login(užívateľ,heslo)
- odoslanie príkazu metódou *talk(['príkaz'])*

Nižšie vidíme príklad metódy spoločne s konštruktorom triedy na pridanie IP adresy s príkladom použitia API slov.

Listing 6: Príklad použitia tikapy

```
from tikapy import TikapyClient
from tikapy import TikapySslClient
class Addresses:
    def __init__(self,address,username,password):
        self.client = TikapySSLClient( address, 8729 )
        self.client.login( username,password)
    def addAddress(self,address,interface):
        """
        """
    @staticmethod
    def will_add_address(
        param_address:
        param_interface:
        return:
        """
        """
        ipv4 = self.client.talk(['/ip/address/add',
            '=address='+address,'=interface='+interface])
        return ipv4
```

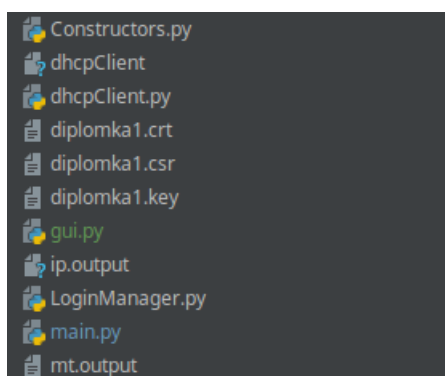
6 KONZOLOVÁ ČASŤ APLIKÁCIE NA SPRÁVU MIKROTIKOV

V tejto kapitole si popíšeme fungovanie naprogramovanej aplikácie. Celkovo je konzolová časť aplikácie napísaná za pomoci knižnice *tikapy* popísanej v kapitole 5.4. Kapitola bude rozdelená do niekoľkých častí:

- časť 1: popis naprogramovanej časti pre vyhľadávanie mikrotikov, pripojenie sa na mikrotik cez python pomocou protokolov telnet, SSH, mactelnet a napojenie na metódy
- časť 2: Popis infraštruktúry backendu - zložky, ich vysvetlenie, zoznam súborov na konfiguráciu mikrotku, vysvetlenie rozdelenia, vysvetlenie tried, metód daných tried a volanie funkcií
- časť 3 - prítidanie tabuliek jednotlivých tried a ich metód v každej zložke, krátka sumarizácia, ich niektoré vybrané UML diagramy, ostatné budú zahrnuté v prílohe

6.1 Popis naprogramovanej časti prihlasovania na mikrotik

V tejto časti si zobrazíme rozbor časti prihlasovania na mikrotik a základné funkcie. Toto je riadené v rámci projektu nazvaného *diplomkap3* v ktorom je súbor *loginManager.py*. V rámci login managera tu nachádzame ďalšie súbory, ktoré sú zobrazené na obrázku 6.1.



Obr. 6.1: Zoznam základných konfiguračných súborov

6.1.1 Súbor centralControl

V súbore centralControl sa popisuje spôsob hromadnej obsluhy mikrotikov na základe protokolu mactelnet. Pozostáva z metód:

- *konštruktor* - pozostáva zo užívateľských mien a hesiel, heslá v premennej credentials sú uložené ako slovník v podobe IP adresa: heslo
- *listMikrotikDevices()* - metóda vráti zoznam MAC a IP adries nájdených mikrotikov, uloží ich do súboru, a finálny výstup predstavuje list MAC adries
- *addCredentials()* - metóda pridáva heslo k užívateľskému účtu do slovníku, štandardnéužívateľské meno sa používa admin, ale tiež sa môže použiť aj iné užívateľské meno pri volaní metódy
- *loginSSH()* - metóda je použitá na hromadné prihlásenie pomocou protokolu SSH na mikrotiky, používa sa tu pritom knižnica pexpect a jej subknížnica pxssh, jej vstupné parametre sú IP adresa serveru, užívateľské meno a heslo

Listing 7: Konštruktor súboru

```
def __init__(self, login):
    self.username = login
    self.credentials = {
        "192.168.1.1": "admin",
        "192.168.2.1": ""
    }
```

Listing 8: Metóda zobrazenia mikrotikov

```
def listMikrotikDevices(self):
    deviceList = []
    loadAddress = False
    os.system("mactelnet_l-t_20_2>&1_>_mt.output")
    with open( "mt.output", "r" ) as file:
        for line in file:
            if loadAddress:
                address = line.split( )[0]
                deviceList.append( address )
            else:
                header = line.split( )
                if len( header ) > 1:
                    if "IP" in header[0]:
                        loadAddress = True
                for i in deviceList:
                    print(i)
    return deviceList
```

Listing 9: Metóda pridania užívateľských mien a hesiel

```
def addCredentials(self, login="admin"):
    server_list = self.listMikrotikDevices()
    print( server_list )
    for server in server_list:
        try:
            password = self.credentials[server]
        except KeyError:
            password = input( "Please_enter_the
            _password_for_" + server + ":" )
            self.credentials[server] = password
    return server_list
```

Listing 10: Metóda hromadného prihlásenia pomocou protokolu SSH

```
def loginSSH(self, server, login, password):
    from pexpect import pxssh, spawn, expect
    import getpass
    for server in self.credentials:
        try:
            connect = pxssh.pxssh( )
            server = self.credentials
            login = 'admin'
            password = self.credentials[server]
            port = 22
            connect.login( server, login, password )
            commands = pxssh.spawn( )
            time.sleep( 10 )
        except pxssh.ExceptionPxssh as e:
            print( "Error" )
            print( str( e ) )
```

6.1.2 Súbor Constructors

Súbor predstavuje zoznam konštruktorov pre konkrétne naprogramované API moduly pomocou knižnice tikapy. V úvode konštruktoru sú popísané importy jednotlivých modulov a submodulov pre konfiguráciu mikrotiku za pomoci API.

Následne je vytvorená trieda Mikrotik, ktorá zahŕňa všetky konštruktory spoločne s ich vstupnými parametrami, ktoré sú adresa, užívateľské meno a heslo.

```

from wireless import interfaceVirtualNstreamDualSlave, interfaceVirtualStation, interfaceVirtualStationBridge
from wireless import interfaceVirtualStationPseudobridge, interfaceVirtualStationPseudoBridgeClone, interfaceVirtualWds
from wireless import interfaceVirtualWds, interfaceVirtualWdsSlave
from interfaces import interfaces

class Mikrotik:
    def __init__(self, username, password, address):
        self.username = "admin"
        self.password = 'admin'
        self.login = LoginManager.LoginManager( username, password )
        self.address = "192.168.1.1"
        #self.interface = Interfaces.InterfaceManager( address, username, password )
        self.users = Users.Users( address, username, password )
        self.services = Services.Services( address, username, password )
        self.filesmanager = Files.Files( address, username, password )
        self.packages = PackageManager.PackageManager( address, username, password )
        self.system = SystemMaintenance.SystemMaintenance( address, username, password )
        self.clock = SystemClock.SystemClock( address, username, password )
        self.certs = Certificates.Certificates( address, username, password )
        self.host = Identity.Identity( address, username, password )
        self.update = AutoUpdate.AutoUpdate( address, username, password )
        self.console = Console.Console( address, username, password )
        self.health = Health.Health( address, username, password )
        self.history = History.History( address, username, password )
        self.LCD = LCD.LCD( address, username, password )
        self.led = LED.LED( address, username, password )

```

Obr. 6.2: Ukážka konštruktorov projektu

6.1.3 Súbor dhcpClient

V tomto súbore sa nachádza základná konfigurácia mikrotiku po pripojení naň. Obsahuje triedu `basicConfig`, ktorá pozostáva z dvoch metód.

V konštruktoze sa nastaví rozhranie, na ktorom sa má adresa nastaviť, IP adresa-/subnet a MAC adresa na pripojenie na mikrotik pomocou protokolu mactelnet.

Listing 11: Trieda `basicConfig`

```

class basicCOnfig:
    def __init__(self, interface, mac, ip):
        self.interface = interface
        self.mac = mac
        self.ip = ip

```

Prvá metóda `dhcp()`, ktorej vstupné parametre sú užívateľské meno a heslo. Pozostáva z prihlásenia na mikrotik, a nastavenia Dynamic Host Client Protocol (DHCP) klienta na rozhraní, ktoré sa definuje pri volaní objektu v rámci konštruktoru.

Listing 12: Metóda dhcp

```
def dhcp(self, username, password):
    child = pexpect.spawn('mactelnet_' + self.mac)
    child.expect('Username:')
    child.sendline(username)
    child.expect('Password:')
    child.sendline(password)
    child.sendline('\r')
    try:
        child.expect('>_')
        child.sendline('ip_dhcp-client_add
=====interface='+self.interface+"\r")
        child.expect('>_')
        child.close()
    except:
        print("error")
        child.close()
    time.sleep(1)
```

Druhá metóda *setAddress()*, ktorá bere ako vstupné parametre užívateľské meno a heslo nastaví statickú IP adresu na rozhraní definovanom v rámci konštruktoru.

Listing 13: Metóda setAddress

```
def setAddress(self, username, password):
    child = pexpect.spawn( 'mactelnet_' +
        self.mac )
    child.expect( 'Username:' )
    child.sendline( username )
    child.expect( 'Password:' )
    child.sendline( password )
    child.sendline( '\r' )
    try:
        child.expect( '>_' )
        child.sendline( 'ip_address_add_address='
            +self.ip + "_interface="
            +self.interface+"\r" )
        child.expect( '>_' )
        child.close()
    except:
        print( "error" )
        child.close()
    time.sleep( 1 )
```

6.1.4 Súbor LoginManager

Súbor LoginManager pozostáva z niekoľkých metód, tieto metódy majú podobnú štruktúru ako súbor centralConrol popisujúci v kapitole 6.1.1.

Ako prvá popísaná časť je konštruktor, ktorý prijíma vstupné parametre užívateľské meno a heslo.

Listing 14: Konštruktor súboru

```
def __init__(self, login, password):
    self.username = login
    self.pwd = password
```

Druhá metóda je metóda *loginTelnet()*, v rámci tejto metódy sa rieši prihlásenie na mikrotik pomocou protokolu telnet za použitia knižnice telnetlib. Vo vstupe metódy sa definuje premenná *server_ip*. Táto premenná je naplnená IP adresou mikrotikovej vrámky sú

Ďalej sa tu nachádza metóda loginSSH(), táto metóda pracujúca podobne ako metóda loginTelnet() pracuje na základe protokolu SSH, na vstupe má server IP adresu, užívateľské meno a heslo.

Listing 16: Metóda loginSSH

```
def loginSSH(self, server, login, password):
    from pexpect import pxssh, spawn, expect
    import getpass
    try:
        connect = pxssh.pxssh( )
        server = '172.16.49.2'
        login = 'admin'
        password = 'admin'
        port = 22
        connect.login( server, login,
            password )
        commands = pxssh.spawn( )
        time.sleep( 10 )
    except pxssh.ExceptionPxssh as e:
        print( "Error" )
        print( str( e ) )
```

Ďalšou metódou je metóda na vylistovanie všetkých mikrotikov, táto metóda je bez vstupného parametru. Ako výstup je súbor mikrtik.output naplnený MAC adresami mikrotikov.

Listing 17: Metóda listMikrotikDevices

```
def listMikrotikDevices(self):
    deviceList = []
    loadMacAddress = False
    os.system("mactelnet_l-l-t_20
    .....2>&l_>_mt.output")
    with open( "mt.output", "r" )
    as file:
        for line in file:
            if loadMacAddress:
                macAddress = line.split( )[1]
                deviceList.append( macAddress )
            else:
                header = line.split( )
                if len( header ) > 1:
                    if "IP" in header[0]
                    and "MAC-Address"
                    in header[1]:
                        loadMacAddress = True

    return deviceList
```

Poslednou metódou je metóda *mactelnetLoginToSingleDevice()*, vďaka ktorej sa pripája pomocou protokolu mactelnet na jedno mikrotik zariadenie pomocou macadresy získanej z výstupu metódy *listMikrotikDevices()* *mikrotik.output*.

Listing 18: Metóda mactelnetLoginToSingleDevice

```
def mactelnetLoginToSingleDevice(self, username,
password, address=None):
    deviceList = self.listMikrotikDevices()
    print( deviceList )
    if address:
        print('mactelnet_{}_-u_{}_-p_{}'.format
( address, username, password ))
        os.system( 'mactelnet_{}_-u_{}_-p_{}'.format
( address, username, password ) )
    elif deviceList:
        print( 'mactelnet_{}_-u_{}_-p_{}'.format
( deviceList[0], username, password ) )
        os.system( 'mactelnet_{}_-u_{}_-p_{}'.format
( deviceList[0], username, password ) )
    else:
        print("No_device_was_found")
```

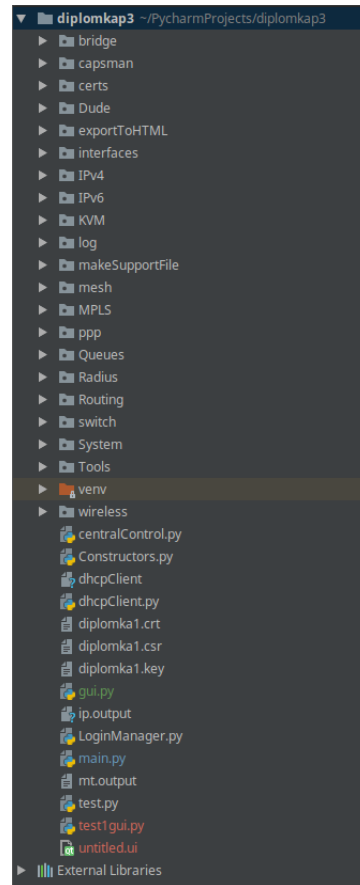
6.2 Rozbor hlavnej časti backendu

V rámci hlavnej konfiguračnej časti diplomovej práce, pre konfiguráciu backendu mikrotiku za pomoci programovacieho jazyka python som projekt rozdelil do niekoľkých častí:

- **bridge** - táto časť obsahuje prvky konfigurácie, pridania, odstránenia, zapnutia, vypnutia možnosti bridgu na mikrotiku, konfigurácia existujúceho bridgu, zobrazenie zoznamu bridgov
- **capsman** - táto časť obsahuje konfiguráciu hromadnej obsluhy mikrotik úrstupových bodov a WiFi, profily, bezpečnosť, konfigurácie, povolené rýchlosti, zobrazenie zoznamu pripojených prvkov a ďalšie funkcie
- **certs** - obsahuje certifikáty na pripojenie sa na mikrotik pomocou protokolu api-ssl
- **Dude** - obsahuje popis konfigurácie ako nastaviť nástroj Dude klienta, ako nakonfigurovať Dude na vzdialený monitoring na Dude serveri, taktiež Dude server, a ďalšie možnosti
- **exportToHtml** - časť predstavuje generovanie súboru na analýzu v podobe webovej stránky

- **interfaces** - časť predstavuje konfiguráciu rozhraní na mikrotiku, tieto časti sú tiež popísané aj v iných zložkách ako napr. bridge. 4asť popisuje pridanie, odstránenie, zapnutie, vypnutie, konfiguráciu existujúcich rozhraní.
- **IPv4** - rozsiahla časť, obsahuje konfiguráciu IP adries, firewallu, monitoringu, smerovania a ďalších nástrojov spadajúcich pod IP zložku na mikrotiku.
- **IPv6** - pre zložku IPv6 platí to isté čo pre zložku IPv4, ale platí pre konfiguráciu na základe IPv6 adresného rozsahu
- **KVM** - sekcia bude popisovať možnosti virtualizácie mikrotiku.
- **log** - sekcia bude popisovať analýzu a konfiguráciu logu zariadenia
- **makeSupportFile** - sekcia bude popisovať vytvorenie súboru potrebného pre analýzu na mikrotik podpore
- **mesh** - sekcia popisuje konfiguráciu tzv. mesh technológie, technológii podobne ako v rámci časti bridge
- **MPLS** - sekcia bude popisovať možnosti konfiurácie Multi Protocol Label Switching (MPLS), jej pridanie, odstránenie ,zapnutie, vypnutie, modifikácie a ďalšie funkcie.
- **PPP** - sekcia bude popisovať konfiguráciu Point to Point Protocol (PPP) a ďalších možností Virtual Private Network (VPN) konfigurácie.
- **Queues** - sekcia budep popisovať konfiguráciu sieťových front, možnosti front, typy front a ďalšie funkcie
- **Radius** - sekcia bude popisovať nastavenie funkcie Radius - autentizačnej služby užívateľov , jeho modifikáciu, konfiguráciu a ďalšie funkcie.
- **Routing** - sekcia bude popisovať možnosti dynamického smerovania, statické smerovanie bude popísané v rámci časti IPv4, dynamické smerovacie protokoly, ich konfigurácie, a ďalšie možnosti.
- **Switch** - sekcia bude popisovať konfiguráciu prepínača, niektoré mikrotiky sú typu SwitchOS a sú štandardne prepínač. Kofiguuráciu portov, trunkov, a ďalších funkcií.
- **System** - sekcia bude popisovať časť konfigurácie systémových nástrojov, ich funkcií a konfigurácie, a ďalších funkcií.
- **Tools** - sekcia bude popisovať konfiguráciu mikrotik nástroj, a však nie všetky bolo možné odsimulovať v rámci konzolovej časti aplikácie, ich konfiguráciu, spustenie, riadenie a ďalšie funkcie.
- **Wireless** - sekcia bude obsahovať konfiguráciu bezdrátového rozhrania, moduly, módy, konfiguráciu, nastavenie, a ďalšie funkcie
- **konfiguračné súbory mimo zložiek** - sekcia popísaná v kapitole 6.1, popisuje súbory na základnú konfiuráciu mikrotiku, nastavenie základnej konfigurácie.

Ukážka súborovej štruktúry je zobrazená na obrázku 6.3:



Obr. 6.3: Štruktúra projektu konzolovej časti projektu

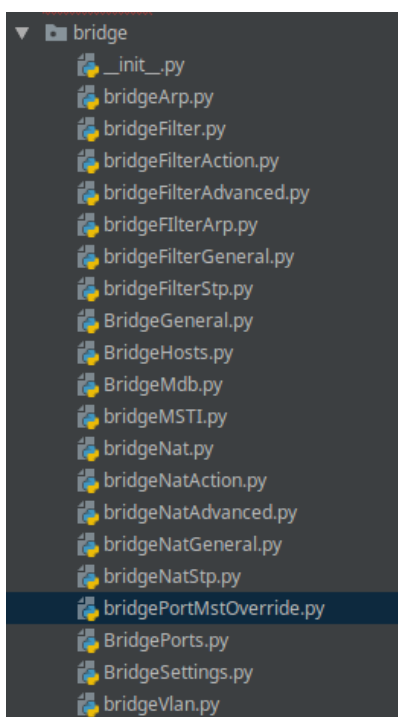
7 HLAVNÁ ČASŤ BACKENDU

Cieľom kapitoly je detailný popis backend časti aplikácie na správu mikrotikov. V jednotlivých podkapitolách bude popísaná každá zložka projektu diplomkap3.

7.1 Zložka bridge

Cieľom tejto zložky je konfigurácia bridgu na mikrotiku. Pozostáva z:

- Managementu bridgu, portov, bezpečnosti, pripojených zariadení
- Pridanie, odstránenie, zapnutie, vypnutie a komentár položiek
- Modifikácia existujúcich položiek



Obr. 7.1: Zoznam súborov zložky bridge

Niektoré pasáže sa dajú modifikovať pomocou mena položky, niektoré pomocou poradia položky. Zoznam súborov zložky nájdeme na obrázku 7.1 a v kapitole 7.1.1.

7.1.1 Popis tried zložky

Zložka obsahuje triedy:

- **bridgeArp** - trieda nastavuje funkcionality ARP v rámci bridgu
- **bridgeFilter** - trieda nastavuje funkcionality filtrovania provozu (firewall)
- **bridgeFilterAction** - trieda nastavuje akcie filtrovania provozu

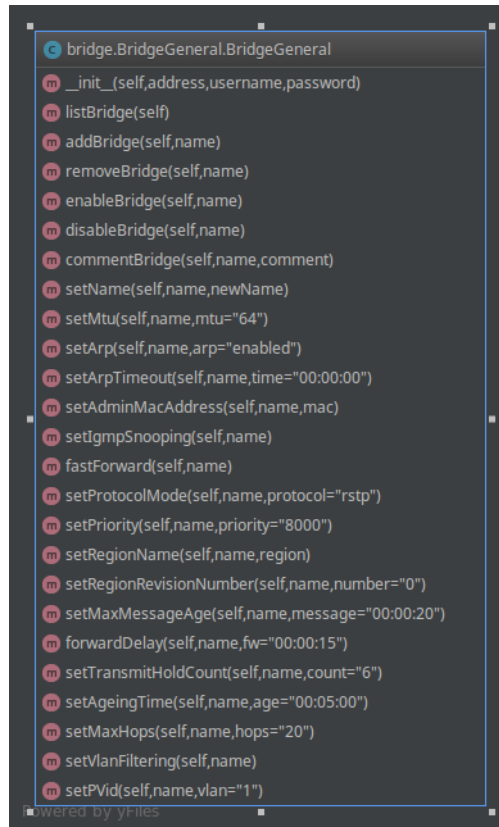
- **bridgeFilterAdvanced** - trieda nastavuje pokročilé filtrovanie
- **bridgeFilterGeneral** - trieda nastavuje genrálne nastavenie filtrovania pro-
vozu
- **bridgeHosts** - trieda ošetrue zoznam pripojených zariadení na bridge
- **bridgeMdb** - trieda ošetrue nastavenie portov pripojených zariadení
- **bridgeMSTI** - trieda nastavuje MST modul bridgu
- **bridgeNAT** - trieda ošetrue nastavenie NAT na bridgi
- **bridgeNATAction** - trieda ošetrue nastavenie akcií NAT
- **bridgeNatAdvanced** - trieda ošetrue pokročilé nastavenie NAT
- **bridgeNatGeneral** - trieda ošetrue genrálne nastavenie NAT na bridgi
- **bridgeNatStp** - trieda ošetrue nastavenie STP
- **bridgePortMstOverride** - trieda ošetrue nastavenie nanútenia MST
- **BridgePorts** - trieda ošetrue nastavenie portov bridgu
- **BridgeSettings** - trieda ošetrue globálne nastavenie bridgu
- **BridgeVlan** - trieda ošetrue globálne nastavenie VLAN

7.1.2 Vybraný analyzovaný súbor

Ako ukážku je vybratý súbor bridgeArp s popisom metód v tabuľke 7.1.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
setArpOpcode	číslo bridgu, operátor	slovník	Metóda nastaví operačný mód ARP - arp-nak, darp-error,...
setArpHardwareType	číslo bridgu, operátor	slovník	Metóda nastaví typ hardvéru (číselný kód)
setArpPacketType	číslo bridgu, typ paketu(číselné označenie)	slovník	Metóda nastaví typ paketov
setArpSrcAddr	číslo bridgu, zdrojová adresa	slovník	Metóda nastaví zdrojovú adresu bridgu
setArpSrcMacAdress	číslo bridgu, zdrojová MAC adresa, maska MAC adresy	slovník	Metóda nastaví zdrojovú MAC adresu bridgu
setArpDstMacAdress	číslo bridgu, cieľová MAC adresa	slovník	Metóda nastaví cieľovú MAC adresu bridgu.
setArpGratituous	číslo bridgu, typ arp(štandardne none)	slovník	Metóda nastaví typ ARP

Tab. 7.1: Tabuľka zoznamu metód triedy bridgeArp



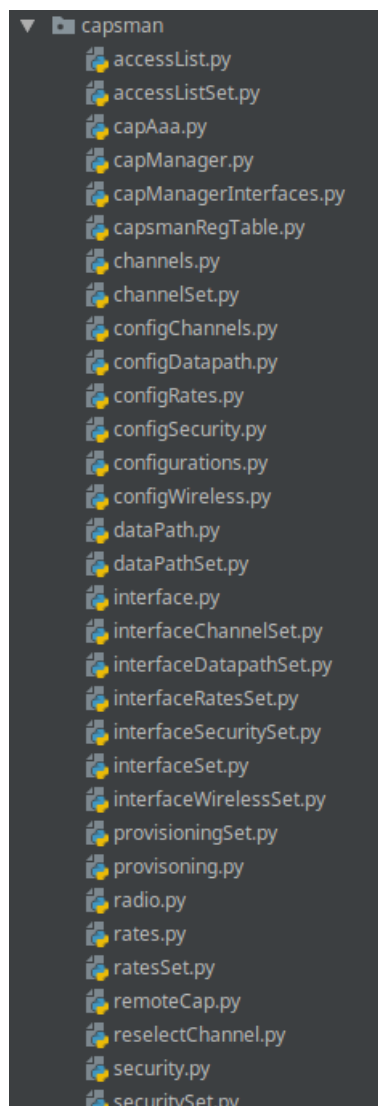
Obr. 7.2: UML diagram vybraného súboru bridgeArp

7.2 Zložka capsman

Súčasťou zložky capsman sú súbory na centrálné nastavenie WiFi pomocou mikrotik funkcionality capsman. Capsman dovoľuje nastaviť a centrálné riadiť prístupové body na centrálnom smerovači. Zložka pozostáva z:

- Správu nakonfigurovaných položiek
- Pridávanie, odstránenie, zapnutie, vypnutie a koment položiek
- Modifikácia nakonfigurovaných položiek
- Správa pripojených zariadení

Zoznam súborov nájdeme na obrázku 7.3.



Obr. 7.3: Zoznam súborov zložky capsman

7.2.1 Popis tried zložky

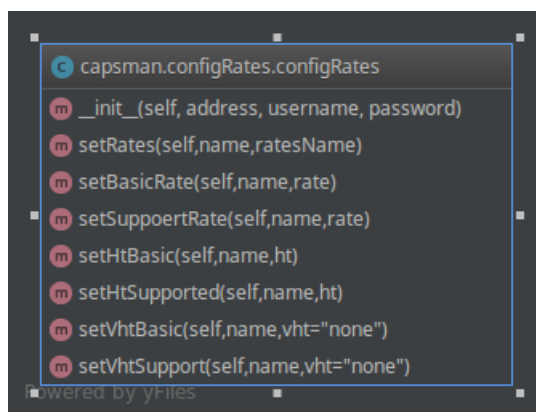
Zložka obsahuje:

- **accessList** - trieda ošetruje nastavenie prístupných listov (access listov)
- **accessListSet**- trieda ošetruje nastavenie už vytvorených access listov
- **capAaa** - trieda ošetruje nastavenie autorizačného protokolu AAA
- **capManager** - trieda ošetruje nastavenie capsman managera
- **capManagerInterfaces**- trieda ošetruje nastavenie capsman rozhraní
- **capsmanRegTable** - trieda obsahuje zoznam zaregistrovaných zariadení a ich publikácia (provisioning)
- **channel** - trieda ošetruje nastavenie kanálov v rámci WIFI pre centrálnu riadenie capsmanom

- **channelSet** - trieda ošetruje nastavenie už existujúcich profilov kanálov
- **configChannels** - trieda ošetruje nastavenie kanálov v rámci globálnej konfigurácie WiFi v rámci capsman
- **configDatapath** - trieda ošetruje nastavenie dátových ciest (datapath) v rámci globálneho konfiguračného súboru
- **configRates** - trieda ošetruje nastavenie povolených prenosových rýchlostí
- **configSecurity** - trieda ošetruje nastavenie bezpečnosti
- **configurations** - trieda ošetruje management, pridávanie, odstránenie, povolenie, zakázanie a komentovanie konfiguračných súborov
- **configWireless** - trieda ošetruje nastavenie Wireless rozhrania
- **dataPath** - trieda ošetruje management dataPath, pridávanie, odstránenie, povolenie, zakázanie a komentovanie
- **dataPathSet** - trieda ošetruje nastavenie už existujúcich dátových ciest
- **interface** - trieda ošetruje management rozhraní riadených capsmanom
- **interfaceChannelSet** - trieda ošetruje nastavenie kanálov v rámci rozhrania
- **interfaceDatpathSet** - trieda ošetruje nastavenie dátových ciest v rámci konfigurácie rozhrania capsman
- **interfaceRatesSet** - trieda ošetruje nastavenie povolených rýchlostí v rámci konfigurácie rozhrania capsmanom
- **interfaceSecuritySet** - trieda ošetruje nastavenie bezpečnosti v rámci capsman rozhrania
- **interfaceSet** - trieda ošetruje základnú konfiguráciu rozhrania
- **interfaceWirelessSet** - trieda ošetruje nastavenie WiFi profilu
- **provisioningSet** - trieda ošetruje možnosti publikácie konfigurácie - statické, dynamické
- **provisioning** - trieda ošetruje management publikácie konfigurácií
- **radio** - trieda ošetruje nastavenie publikácie pripojných prístupových bodov
- **rates** - trieda ošetruje management povolených prenosových rýchlostí
- **ratesSet** - trieda ošetruje nastavenie prenosových rýchlostí
- **remoteCap** - trieda ošetruje správu pripojených prístupových bodov - upgrade, publikáciu
- **reselectChannnels** - trieda ošetruje výber druhého kanálu prístupového bodu
- **securiity** - trieda ošetruje management bezpečnostných profilov - pridávanie, odstránenie, povolenie, zakázanie a komentovanie
- **securitySet** - trieda ošetruje nastavenie bezpečnosti už existujúcich profilov

7.2.2 Vybraný analyzovaný súbor

Pre analýzu jedného súboru zo zložky je vybratý súbor *configRates.py*. Jeho UML diagram je zobrazený na obrázku 7.4 a zoznam jeho metód je popísaný v tabuľke 7.2.



Obr. 7.4: UML diagram triedy configRates

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
setRates	názov profilu, nové meno	slovník	Metóda premenuje profil
setBasicRate	názov profilu, základné prenosové rýchlosti	slovník	Metóda nastaví prenosovú rýchlosť.
setSuppoertRate	názov profilu, podporované prenosové rýchlosti	slovník	Metóda nastaví posporované prenosové rýchlosti.
setHtBasic	názov profilu, základný prenosový kanál(y)	slovník	Metóda nastaví prenosový kanál.
setHtSupported	názov profilu, podporované prenosové kanály	slovník	Metóda nastaví podporované prenosové kanály.
setVhtBasic	názov profilu, základné virtuálne kanály	slovník	Metóda nastaví prenosový virtuálny kanál.
setVhtSupport	názov profilu, podporované prenosové virtuálne kanály	slovník	Metóda nastaví podporované prenosové kanály.

Tab. 7.2: Popis triedy configRates

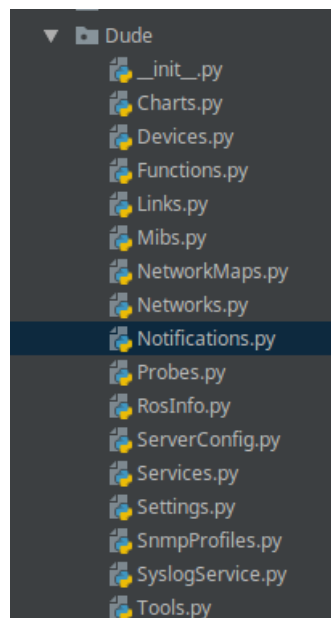
7.3 Zložka Dude

Popisovaná zložka je obsah tried, ktoré nastavujú Dude monitorovací nástroj. Plnia rovnakú funkciu ako je to popísané v kapitolách 7.1.1 a 7.2. Na obrázku 7.5 vidíme zoznam súborov.

7.3.1 Popis tried zložky

Súčasťou zložky je nastavenie nástroja Dude na centrálny monitoring mikrotikov. Obsahuje súbory:

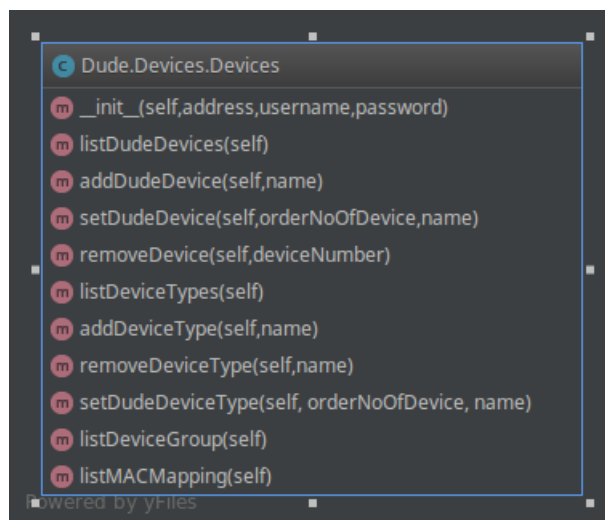
- **Devices** - trieda ošetruje nastavenie monitorovaných zariadení
- **Notifications** - trieda ošetruje nastavenie upozornení
- **Probes** - trieda ošetruje nastavenie tetsovaní spojenia
- **RosInfo** - trieda ošetruje výpis informácií ohľadom hardvéru a operačného systému routerOS
- **Services** - trieda ošetruje nastavenie služieb
- **Settings** - trieda ošetruje zapnutie a vypnutie Dude nástroju
- **ostatné knižnice** - ostatné knižnice API nepodporuje



Obr. 7.5: Zoznam súborov adresáru Dude

7.3.2 Analýza vybraného súboru

Vybraný analyzovaný súbor *Devices.py* a jeho analýza je popísaná v rámci jeho UML diagramu na obrázku 7.6 a zoznam metód je popísaný v tabuľke 7.3.



Obr. 7.6: UML diagram knižnice Devices

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listDudeDevices	žiadny	slovník	Metóda vypíše zoznam Dude zariadení.
addDudeDevice	názov zariadenia	slovník	Metóda pridá nové zariadenie
setDudeDevice	číslo zariadenia, meno zariadenia	slovník	Metóda premenuje zariadenie.
removeDevice	číslo zariadenia	slovník	Metóda zmaže zariadenie.
listDeviceTypes	žiadny	slovník	Metóda zobrazí typy zariadení.
addDeviceType	názov zariadenia	slovník	Metóda pridá nové zariadenie.
removeDeviceType	názov profilu	slovník	Metóda zmaže typ zariadenia.

Tab. 7.3: Tabuľka metód triedy Devices

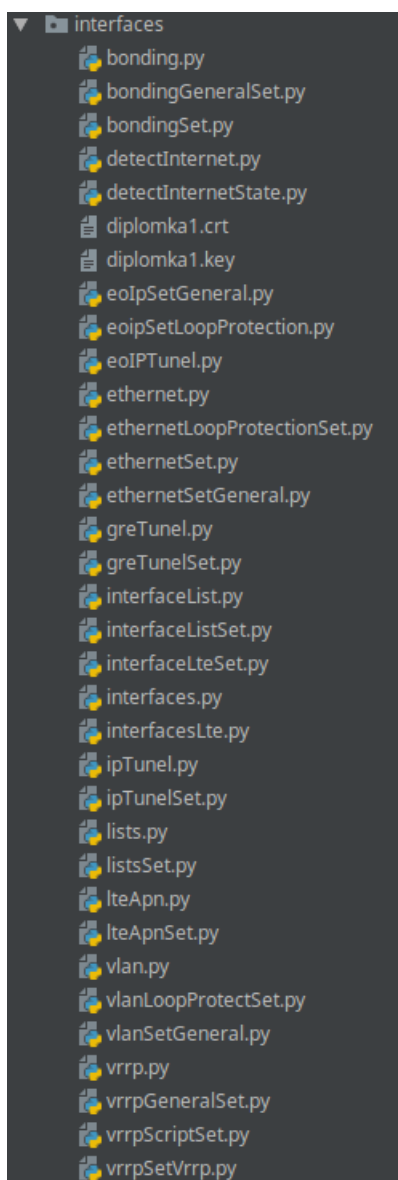
7.4 Zložka Interfaces

Popisovaná zložka je obsah tried, ktoré nastavujú rozhrania na mikrotiku. Medzi tieto rozhrania patria nastavenie VPN, ethernet, WiFi rozhraní a ďalších rozhraní. Plnia rovnakú funkciu ako je to popísané v kapitolách 7.1.1 a 7.2 a ďalších kapitolách. Na obrázku 7.7 vidíme zoznam súborov.

7.4.1 Popis tried zložky

Súčasťou zložky je nastavenie rohraní na mikrotiku. Patria sem triedy:

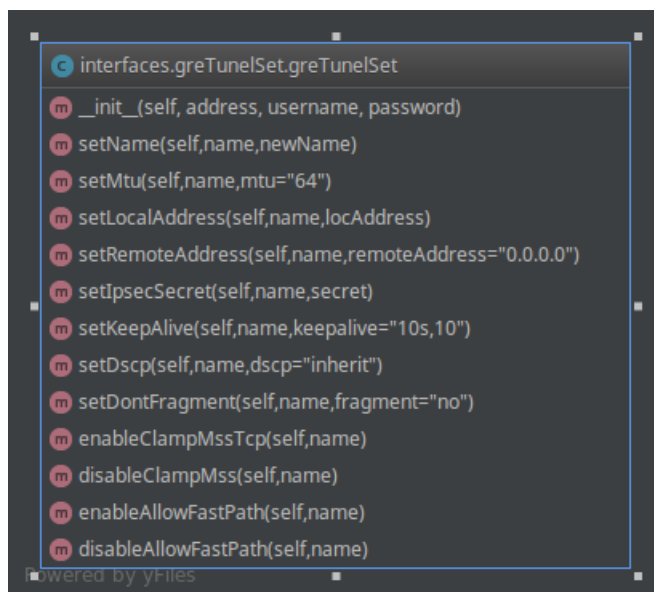
- **bonding** - trieda ošetruje nastavenie bonding rozhrania, rozhrania na nastavenie failover technológie súčasne s ďalšími dvomi triedami na nastavenie bonding - **bondingGeneralSet** a **bondingSet**
- **detectInternet** - trieda detekuje internet na vybranom rozhraní
- **eoipTunel** - trieda nastaví ethernet over IP rozhranie, spoločne s triedami **eoipSetGeneral** a **eoipSetLoopProtection**
- **ethernet** - trieda nastaví ethernet rozhrania spoločne s triedami **ethernetSet**, **ethernetSetGeneral** a **ethernetLoopProtectionSet**
- **greTunnel** - trieda nastaví rozhranie typu tunel GRE spoločne s triedou **greTunnelSet**
- **interfaceList** - trieda nastaví listrozhraní spoločne s triedou **interfaceListSet**
- **interfaces** - trieda zobrazí a nastaví všetky rozhrania na mikrotiku
- **interfaces** - trieda nastaví Long Term Evolution (LTE) rozhranie
- **ipTunnel** - trieda nastaví IP tunel spoločne s triedou **ipTunnelSet**
- **lists** - trieda nastaví listy rozhraní spoločne s triedou **listsSet**
- **lteApn** - trieda nastaví prístupové LTE body spoločne s triedou **lteApnSet**
- **vlan** - trieda nastaví VLAN spoločne s triedami **vlanLoopProtection** a **vlanSetGeneral**
- **vrrp** - trieda nastaví zálohovací mechanizmus Virtual Router Redoundency Protocol (VRRP) spoločne s triedami **vrrpGeneralSet**, **vrrpScriptSet** a **vrrpSetVrrp**



Obr. 7.7: Zoznam súborov adresáru Interfaces

7.4.2 Analýza vybraného súboru

Vybraný súbor *greTunnelSet.py* a jeho nastavenie je zobrazené v UML diagrame triedy na obrázku 7.8 a popis tried je zanalyzovaný v tabuľke 7.4.



Obr. 7.8: UML diagram greTunnelSet triedy

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
setName	meno tunelu, nové meno tunelu	slovník	Metóda prmenuj tunel rozhranie.
setMtu	názov tunelu, veľkoť MTU	slovník	Metóda nastaví MTU rozhrania.
setLocalAddress	názov tunelu, lokálna IP adresa	slovník	Metóda nastaví lokálnu adresu.
setRemoteAddress	názov tunelu, vzdialená IP adresa	slovník	Metóda nastaví vzdialenú IP adresu tunelu.
setIpsecSecret	názov tunelu, heslo	slovník	Metóda nastaví heslo na tunely.
setKeepAlive	názov tunelu, keepalive interval	slovník	Metóda nastaví keepalive interval.
setDscp	názov tunelu, hodnota DSCP	slovník	Metóda nastaví hodnotu DSCP.
setDontFragment	názov tunelu, fragmentovanie	slovník	Metóda nastaví možnosť fragmentovania (štandardne nie).
enableClampMssTcp	názov tunelu	slovník	Metóda zapne MSS pole pri fragmentovaní.
disableClampMss	názov tunelu	slovník	Metóda vypne MSS pole pri fragmentovaní.
enableAllowFastPath	názov tunelu	slovník	Metóda zapne funkciu tunelu "fast path".
disableAllowFastPath	názov tunelu	slovník	Metóda vypne funkciu tunelu "fast path".

Tab. 7.4: Tabuľka metód triedy greTunnelSet

7.5 Zložka IPv4

Popis zložky IPv4 spočíva v nastavebí rôznych IPv4 protokolov, bezpečnosti, prekladu adres, statického smerovania a ďalších možností. Triedy spočívajú globálnym

popisom totožným k popisov v kapitolách 7.1.1, 7.2 a ďalších kapitolách.

7.5.1 Popis tried zložky

Zložka obsahuje konfiguračné súbory nastavenie protokolov, adres, bezpečnosti a ďalších vecí na základe protokolu IPv4. Zložka obsahuje:

- **Accounting** - Trieda ošetruje nastavenie zabezpečenia
- **Addresses** - Trieda ošetruje nastavenie IP adres
- **Arp** - Trieda ošetruje nastavenie Address Resolution Protocol (ARP)
- **DHCPClient** - trieda ošetruje nastavenie DHCP klienta
- **DHCPRelay** - trieda ošetruje nastavenie DHCP relay agenta
- **DHCPServer** - trieda ošetruje nastavenie DHCP serveru
- **DNS** - triedy **DNScache**, **DNSGlobal** a **DNSstatic** ošetrujú nastavenie DNS protokolu global rieši management DNS serverov, cache rieši ošetrenie pridaných záznamov do DNS a static pridáva statické DNS záznamy
- **Firewall** - triedy **Firewall-GeneralSetup**, **Action**, **Addresslist**, **AdvancedSetup**, **Connections**, **ExtraSetup**, **Filter**, **L7Protocols**, **Mangle**, **NAT**, **ServicePorts**

GeneralSetup - trieda nastavuje základné vlastnosti firewallu **Action** - trieda nastavuje akcie - povolenie, zakázanie, logovanie, ...

AddressList - trieda nastavuje address listy, skupiny adres v jednej premennej

AdvancedSetup - trieda ošetruje nastavenie pokročilých vlastností firewallu -napr. povolenie address listu, kde sa bude aplikovať, skupiny rozhraní, ...

Connections - trieda ošetruje správu spojení na mikrotiku **ExtraSetup** - trieda ošetruje nastavenie napr. veľkosti hlavičky, sledovania počtu paketov za sekundu, ...

Filter - trieda ošetruje nastavenie a správu filter pravidiel

L7Protocols - trieda ošetruje nastavenie L7 protokolov - napr. torrent,...

Mangle - trieda ošetruje nastavenie Quality of Service (QoS)

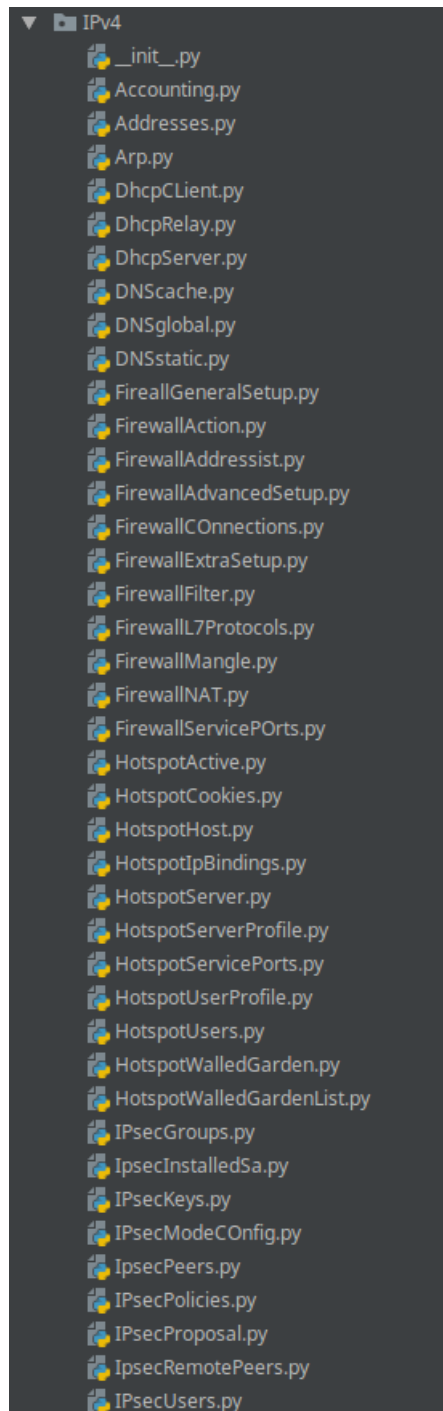
Network Address Translation (NAT) - trieda ošetruje nastavenie prekladu adres

ServicePorts - trieda ošetruje servisné porty nastavené na firewallle

- **Hotspot** - triedy **HotspotActive** a **HotspotCookies**, **HotspotHost**, **HotspotBridging**, **HotspotServer**, **HotspotServerProfile**, **HotspotServicePorts**, **HotspotUserProfile**, **HotspotUsers**, **HotspotWalledGarden**, **HotspotWalledGardenList** ošetrujú nastavenie WiFi hotspotu a to konkrétne:
HotspotActive - trieda ošetruje nastavenie aktuálne bežiaceho hotspotu
HotspotCookies - trieda ošetruje nastavenie cookies

- HotspotHost** - trieda ošetruje nastavenie správy hostov
- HotspotBridging** - trieda ošetruje nastavenie bridgu na hotspot
- HotspotServer** - trieda ošetruje nastavenie hotspot serveru
- HotspotServerProfile** - trieda ošetruje profil (konfiguračný) na nastavenie serveru hotspotu
- HotspotServicePorts** - trieda ošetruje správu servisných portov hotspotu
- HotspotUserProfile** - trieda ošetruje správu a nastavenie užívateľských profilov
- HotspotUsers** - trieda ošetruje správu pripojených užívateľov
- HotspotWalledGarden** - trieda ošetruje nastavenie bezpečnosti hotspotu
- HotspotWalledGardenList** - trieda ošetruje nastavenie "bezpečnostných listov"
- **IPsec** - triedy **IPsecGroups**, **IPsecInstalledSA**, **IPsecKeys**, **IPsecModeCinfigs**, **IPsecPeers**, **IPsecPolicies**, **IPsecProposal**, **IPsecRemotePeers**, **IPsecUsers** nastavujú IPsec tunely a pozostávajú:
 - IPsecGroups** - trieda ošetruje nastavenie IPsec skupín adries
 - IPsecInstalledSA** - trieda spravuje nainštalované adresy
 - IPsecKeys** - trieda ošetruje nastavenie kľúčov zabezpečenia
 - IPsecModeCinfigs** - trieda ošetruje nastavenie módov
 - IPsecPeers** - trieda ošetruje nastavenie fázy 1 IPsec
 - IPsecPolicies** - trieda ošetruje nastavenie fázy 2 IPsec
 - IPsecProposal** - trieda ošetruje nastavenie proposal profilov zabezpečenia tunelu
 - IPsecRemotePeers** - trieda ošetruje správu vzdialených pripojených smerovačov do tunelu
 - IPsecUsers** - trieda ošetruje správu užívateľov
 - **Neighbors** - triedy **NeighborDiscovery** a **Neighborlist** ošetrujú správu pripojených zariadení na mikrotik
 - **Packing** - trieda **Packing** ošetruje nastavenie a správu nainštalovaných balíčkov
 - **Pool** - triedy **Pool** a **PoolUsedAddresses** slúžia na konfiguráciu adresných rozsahov
 - **Route** - správa nastavení smerovania v triedach - **RouteVrf**, **RouteGeneral**, **RouteNexthops** a **RouteRules**
 - RouteVrf** - správa nastavenia Vrf smerovania
 - RouteGeneral** - správa hlavných smerovacích profilov
 - RouteNexthops** - správa "next hop" adries
 - RouteRules** - správa lokálnych smerovacích pravidiel
 - **Services** - trieda nastavuje povolené štandardné porty a služby na mikrotiku

- **Settings** - trieda nastavuje globálne IPv4 nastavenie mikrotiku
- **Samba** - triedy **Smb**, **SmbShare** a **smbUsers** ošetrujú nastavenie Samba protokolu
Smb - trieda globálne rieši nastavenie Samba profilov
SmbShare - trieda ošetruje nastavenie zdieľaných zložiek **SmbUsers** - trieda ošetruje nastavenie Samba užívateľov
- **Snmp** - trieda ošetruje nastavenie správu monitoringu zariadenia v triedach **Snmp**, **SnmpCommunity**
Snmp - trieda ošetruje globálne nastavenie SNMP protokolu
SnmpCommunity - trieda ošetruje globálne nastavenie komunnít (community stringov)
- **Socks** - trieda ošetruje nastavenie socketov v triedach **Socks**, **SocksAccess** a **SocksConnections**
- **Tftp** - trieda ošetruje nastavenie TFTP provozu
- **TrafficFlow** - trieda ošetruje nastavenie kontroly trafiky v triedach **TrafficFlow** a **TrafficFlowIpFix**
- **Upnp** - trieda ošetruje nastavuje UPNP v triedach **upnpinterface** a **upnp-settings**
- **Webproxy** - ošetrovanie nastavenia Webového proxy serveru v súboroch:
WebProxyAccess - trieda ošetruje prístup k proxy serveru
WebProxyCache - trieda ošetruje správu cache pamäti proxy serveru
WebProxyCacheContents - trieda spravuje obsah pamäti
WebProxyConnections - trieda spravuje pripojené zariadenia na proxy server
WebProxyDirect - trieda spravuje nastavenie priameho pripojenia na proxy server
WebProxyLookup - trieda ošetruje nastavenie lokálnej proxy DNS
WebProxyRefreshes - trieda ošetruje nastavenie obnovovacej frekvencie
WebProxySettings - trieda ošetruje globálne nastavenia proxy serveru



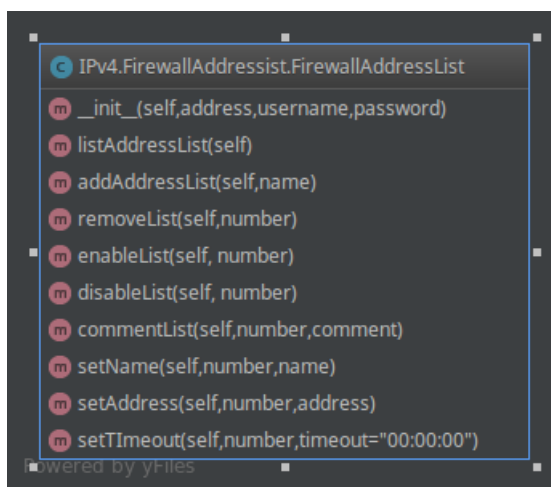
Obr. 7.9: Štruktúra časti zložky IPv4

7.5.2 Analýza vybraného súboru

Vybraný súbor *FirewallAddressist.py* popísaný UML diagramom na obrázku 7.10 a obsah jeho metód je popísaný v tabuľke 7.5.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listAddressList	žiadny	slovník	Metóda vypíše zoznam address listov.
addAddressList	meno	slovník	Metóda pridá nový address list.
removeList	meno	slovník	Metóda odstráni address list.
enableList	meno	slovník	Metóda zapne address list.
disableList	meno	slovník	Metóda vypne address list.
commentList	meno, komentár	slovník	Metóda nastaví komentár k záznamu v address liste.
setName	číslo poradia záznamu, meno	slovník	Metóda zmení address list v zázname.
setAddress	číslo poradia záznamu, adresa	slovník	Metóda zmení IP adresu položky.
setTimeout	číslo poradia záznamu, timeout hodnota	slovník	Metóda nastaví hodnotu timeoutu pre záznam v address liste.

Tab. 7.5: Obsah triedy FirewallAddressist



Obr. 7.10: UML diagram triedy FirewallAddressist

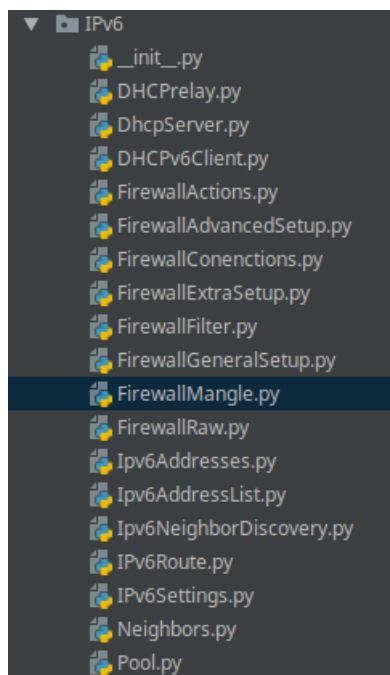
7.6 Zložka IPv6

Podstata zložky IPv6 je nastavenie IPv6 protokolu pozostávajúceho z DHCP pre IPv6, rozsahov adries, smerovania, firewallu a ďalších možností.

7.6.1 Popis tried zložky

Zložka obsahuje nastavenie funkcií prokolu IPv6 a zahrňuje:

- **DHCPv6** - nastavenie protokolu DHCPv6 v triedach:
DHCPRelay - nastavenie DHCP Relay pre verziu IPv6
DHCPv6Server - nastavenie DHCP serveru
DHCPv6Client - nastavenie DHCPv6 klienta
FirewallActions - nastavenie firewall akcií - povolenie ,zahodenie, logovanie,
...
FirewallAdvancedSetup - nastavenie pokročilých vlastností firewallu napr.
povolenie address listu, ...
FirewallConnections - Správa pripojení v o verzii IPv6
FirewallExtraSetup - Správa pokročilých nastavení napr. počet odoslaných
paketov, ...
FirewallFilter -nastavenie filter pravidiel
FirewallGeneralSetup - nastavenie hlavných vlastností pravidlaa správa
pravidiel
FirewallMangle - nastavenie QoSpre IPv6
FirewallRaw - nastavenie Raw (obdoba NAT)
IPv6 AddressList - nastavenie address listu
- **Addresses** - nastavenie a správa IPv6 adriesv triede **IPv6Addresses**
- **Neighbors** - nastavenie a správa vyhľadávania pripojených zariadení v triede **IPv6NeighborDiscovery**
- **Route** - správa smerovania v triede **IPv6Route**
- **Settings** - správa nastavenia IPv6 na globálnej úrovni v triede **IPv6Settings**
- **Neighbors** - správa pripojených zariadení v triede **Neighbors**
- **Pool** - správa rozsahov adries v triede **Pool**



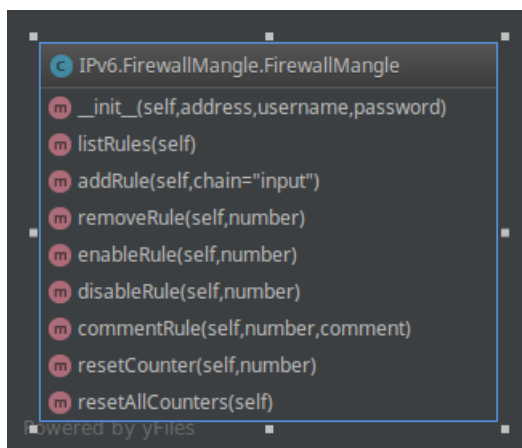
Obr. 7.11: Zoznam súborov zložky IPv6

7.6.2 Anylýza vybraného súboru

Vybraný súbor *FirewallManngle.py* je zobrazené UML diagramom triedy na obrázku 7.13 a obsah metód je zobrazený v tabuľke 7.6.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listRules	žiadny	slovník	Metóda vypíše zoznam pravidiel.
addRule	chain	slovník	Metóda pridá nové pravidlo.
removeRule	poradové číslo pravidla	slovník	Metóda odstráni pravidlo.
enableRule	poradové číslo pravidla	slovník	Metóda zapne pravidlo.
disableRule	poradové číslo pravidla	slovník	Metóda vypne pravidlo.
commentList	poradové číslo pravidla, komentár	slovník	Metóda nastaví komentár k pravidlu.
resetCounter	poradové číslo pravidla	slovník	Metóda zmaže počítadlo paketov a bytov pre konkrétne pravidlo.
resetAllCounters	žiadny	slovník	Metóda zmaže počítadlo všetkých pravidiel paketov a bytov.

Tab. 7.6: Tabuľka triedy FirewallMangle



Obr. 7.12: UML diagram triedy FirewallMangle

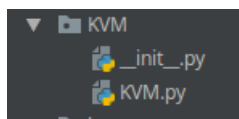
7.7 Zložky KVM, log a makeSupportFile

Obsahom zložiek je nastavenie virtuálneho mikrotiku, nastavenie logovania a vytvorenie súboru, ktorý je možné odsláť na mikrotik podporu na analýzu. Celkový obsah je popísaný nižšie.

7.7.1 Popis triedy zložky KVM

Nastavenie virtuálnych mikrotikov alebo "mikrotiku v mikrotiku" je možné pomocou tzv. KVM. Zložka obsahuje triedy:

- **KVM** - trieda na nastavenie virtualizácie na mikrotiku pomocou triedy **KVM**

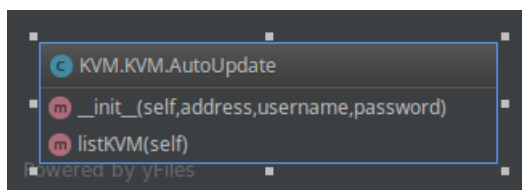


Obr. 7.13: Zoznam súborov zložky KVM

Popis súboru *KVM.py* je popísaný na obrázku 7.14 a jeho obsah metód je popísaný v tabuľke 7.7.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listKVM	žiadny	slovník	Metóda zobrazí všetky virtuálne mikrotiky.

Tab. 7.7: Tabuľka metód triedy KVM

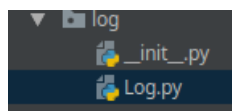


Obr. 7.14: UML diagram triedy KVM

7.7.2 Popis triedy zložky log

Zložka log popisuje výpis systémového logu. Obsahuje triedy:

- **log** - trieda log na výpis systémového logu

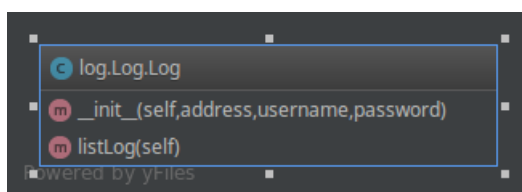


Obr. 7.15: Zoznam súborov zložky log

Zoznam použitých metód je popísaný UML diagramom triedy na obrázku 7.16 a popísaný v tabuľke 7.8.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listLog	žiadny	slovník	Metóda zobrazí systémový log.

Tab. 7.8: Tabuľka metód triedy log



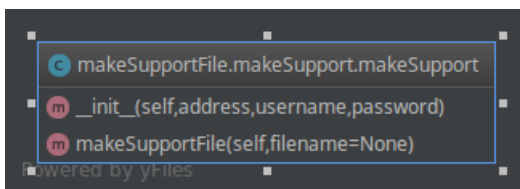
Obr. 7.16: UML diagram triedy log

7.7.3 Popis triedy makeSupportFile

Zložka log obsahuje súbor makeSupportFile spoločne striedou makeSupportFile, vytvorí a odošle súbor na podporu. Trieda makeSupport má na starosti vytvorenie súboru pre podporu na analýzu. Obsah súboru je popísaný v tabuľke 7.9 a UML diagramom triedy na obrázku 7.17.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
makeSupportFile	názov súboru	slovník	Metóda vytvorí súbor na podporu.

Tab. 7.9: Tabuľka metód triedy makeSupport



Obr. 7.17: UML diagram triedy makeSupport

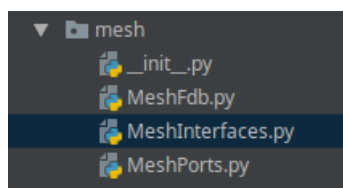
7.8 Zložka Mesh

Účelom zložky mesh je nastavenie tzv. mesh siete, mesh portov, správa pripojených zariadení, atď. Zložka pozostáva obdobne ako je to v kapitolách 7.1 a ďalších kapitolách.

7.8.1 Popis tried zložky

Zložka pozostáva z tried:

- **MeshFdb** - trieda ošetruje nastavenie a správu FDB prvkov
- **MeshInterfaces** - trieda ošetruje nastavenie a správu mesh rozhraní
- **MeshPorts** - trieda ošetruje nastavenie a správu mesh portov



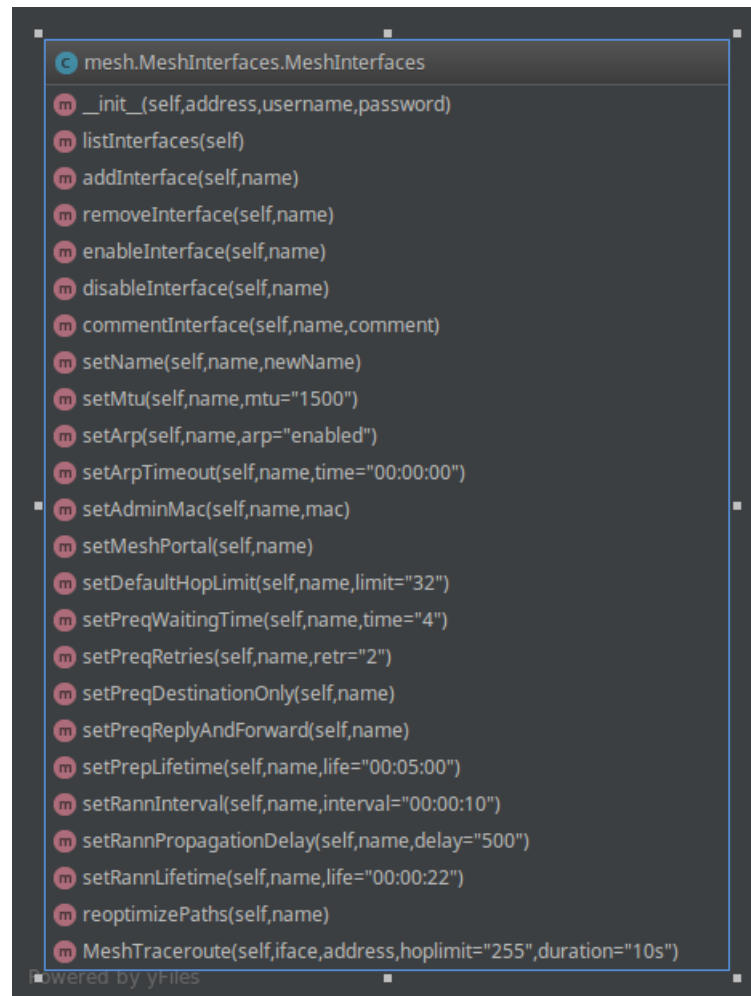
Obr. 7.18: Zoznam súborov zložky mesh

7.8.2 Analýza vybraného súboru

Pre vybraný analyzovaný súbor *MeshInterfaces.py*. Analýza tried je zobrazená na UML diagrame na obrázku 7.19 a popis metód je popísaný v tabuľke 7.10.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listInterfaces	žiadny	slovník	Metóda zobrazí všetky rozhrania.
addInterface	meno rozhrania	slovník	Metóda pridá rozhranie.
removeInterface	meno rozhrania	slovník	Metóda zmaže rozhranie.
enableInterface	meno rozhrania	slovník	Metóda zapne rozhranie.
disableInterface	meno rozhrania	slovník	Metóda vypne rozhranie.
commentInterface	meno rozhrania, komentár	slovník	Metóda okomentuje rozhranie.
setName	meno rozhrania, nové meno rozhrania	slovník	Metóda premenuje rozhranie.
setMtu	meno rozhrania, MTU	slovník	Metóda nastaví veľkosť MTU.
setArp	meno rozhrania, Arp mód (enabled štandardne)	slovník	Metóda nastaví mód ARP protokolu.
setArpTimeout	meno rozhrania, nastavenie hodnoty timeoutu	slovník	Metóda nastaví timeout ARP.
setAdminMac	meno rozhrania, admin MAC adresa	slovník	Metóda nastaví admin adresu typu MAC.
setMeshPortal	meno rozhrania	slovník	Metóda nastaví mesh portál.
setDefaultHopLimit	meno rozhrania, limit (štandardne 2)	slovník	Metóda nastaví maximálny počet "prekokov".
setPreqWaitingTime	meno rozhrania, časová hodnota	slovník	Metóda nastaví čas čakania záznamu v portáli.
setPreqRetries	meno rozhrania, počet opakovaní spojenia	slovník	Metóda nastaví počet možných opakovaní spojenia.
setPreqDestinationOnly	meno rozhrania	slovník	Metóda nastaví spôsob spracovania dát v cieľi.
setPreqReplyAndForward	meno rozhrania	slovník	Metóda nastaví spôsob spracovania odozvy prijatej správy.
setPrepLifetime	meno rozhrania, doba života	slovník	Metóda nastaví hodnotu doby života zariadenia na portály.
setRannInterval	meno rozhrania, interval	slovník	Metóda nastaví interval doby trvania metódy rann.
setRannPropagationDelay	meno rozhrania, spozdenie	slovník	Metóda nastaví delay systému rann.
setRannLifetime	meno rozhrania, doba života	slovník	Metóda nastaví dobu života rann.
reoptimizePaths	meno rozhrania	slovník	Metóda optimalizuje cestu k cieľu na rozhraní.
MeshTraceroute	rozhranie, adresa, limit (štandardne 255), trvanie (štandardne 10 s)	slovník	Metóda obsahuje traceroute na mesh adresu.

Tab. 7.10: Tabuľka metód v triede MeshInterfaces



Obr. 7.19: UML diagram triedy MeshInterfaces

7.9 Zložka MPLS

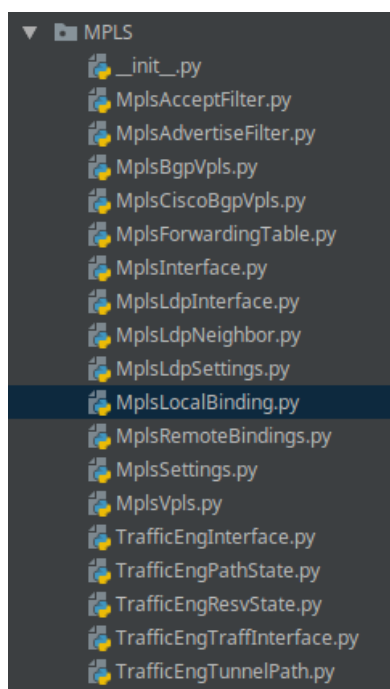
Multi Protocol Label Switing (MPLS) a jeho zložka globálne popísaná rovnakým spôsobom ako zložka v kapitole 7.1 a ďalších kapitolách. Triedy obsiahnuté v zložke slúžia na nastavenie MPLS prepnania v počítačových sieťach.

7.9.1 Popis tried zložky

Zložka pozostáva z tried:

- **MplsAcceptFilter** - trieda ošetruje nastavenie filktrovanie trafiky
- **MplsAdvertiseFilter** - trieda ošetruje publikovanie filtrov
- **MplsBgpVpls** - trieda ošetruje nastavenie VPLS v rámci BGP protokolu
- **MplsCiscoBgpVpls** - trieda ošetruje nastavenie VPLS v rámci technológií firmy CISCO

- **MplsForwardingTable** - trieda ošetruje správu forwarding tabuľky
- **MplsLdpInterface** - trieda ošetruje správu a nastavenie rozhraní MPLS
- **MplsLdpNeighbor** - trieda ošetruje správu a nastavenie susedov
- **MplsLocalBinding** - trieda ošetruje nastavenie a správu lokálnych pripojení
- **MplsRemoteBimndings** - trieda ošetruje nastavenie a správu vzdialených pripojení
- **MplsSettings** - trieda ošetruje globálne nastavenie MPLS
- **MplsVpls** - trieda ošetruje správu a nastavenie VPLS
- **TrafficEngInterface** - trieda ošetruje správu a nastavenie rozhrania prijímania trafiky
- **TrafficEngPathState** - trieda ošetruje nastavenie a správu ciest tzv. "path state"
- **TrafficResvState** - trieda ošetruje nastavenie a správu trafiky na strane prijímacej strany
- **TrafficEngTraffInterface** - trieda ošetruje nastavenie a správu rozhraní riadenia trafiky
- **TrafficEngTunnelPath** - trieda ošetruje nastavenie a správu tunelov v rámci kontroly trafiky



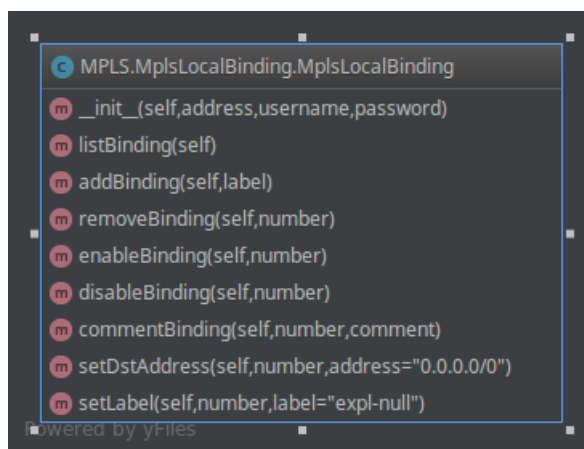
Obr. 7.20: Zoznam súborov zložky MPLS

7.9.2 Anylýza vybraného súboru

Vybraný analyzovaný súbor *MplsLocalBinding.py* je popísaný jeho UML diagramom na obrázku 7.21 a v tabuľke 7.11.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listBindings	žiadny	slovník	Metóda zobrazí všetky preklady.
addBinding	žiadny	slovník	Metóda pridá preklad.
removeBinding	poradové číslo prekladu	slovník	Metóda zmaže záznam prekladu.
enableBinding	poradové číslo prekladu	slovník	Metóda zapne záznam prekladu.
disableBinding	poradové číslo prekladu	slovník	Metóda vypne záznam prekladu.
commentBinding	poradové číslo prekladu, komentár	slovník	Metóda okomentuje záznam prekladu.
setDstAddress	poradové číslo prekladu cieľová IP adresa	slovník	Metóda nastaví cieľovú adresu tunelu.
setLabel	poradové číslo prekladu, označenie tunelu	slovník	Metóda nastaví označenie tunelu.

Tab. 7.11: Tabuľka zoznamu metód triedy MplsLocalBindings



Obr. 7.21: UML diagram triedy MplsLocalBindings

7.10 Zložka PPP

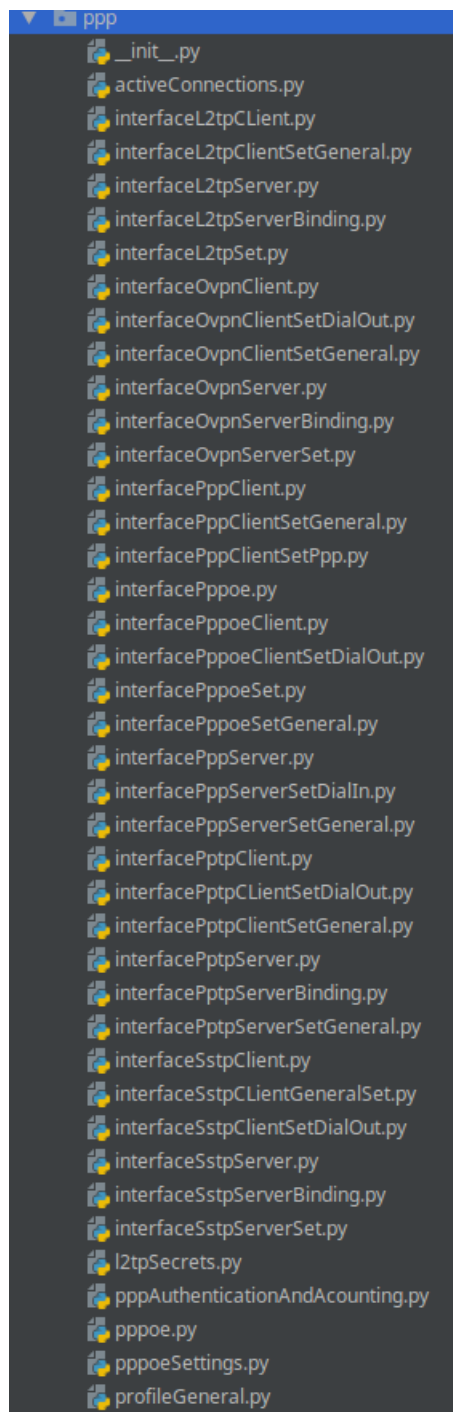
Úlohou zložky Point to point Protocol (PPP) je riadenie a konfigurácia VPN spojenia rôznych typov napr. Open VPN (OVPN), SSTP, L2TP, ... Štruktúra zložky je rovnaká ako v predchádzajúcich zložkách zahrňujúc kapitolu 7.1.1 a ďalšie kapitoly.

7.10.1 Popis tried zložky

Zložka obsahuje triedy:

- **activeConenctions** - trieda ošetruje správu pripojených zariadení
- **interfaceL2TP** - triedy
 - interfaceL2TPClient** - nastavenie L2TP klienta
 - interfaceL2TPSetGeneral** - nastavenie globálnych nastavení klienta
 - interfaceL2TPServer** - nastavenie L2TP servera
 - interfaceL2TPServerBinding** - nastavenie spravovateľného L2TP server rozhrania
 - interfaceL2TPSet** - globálne nastavenie rozhrania L2TP
- **interfaceOvpn** - obsahuje triedy:
 - interfaceOvpnClient** - nastavenie OVPN klienta
 - interfaceOvpnClientSetDialOut** - nastavenie vytáčania OVPN klienta
 - interfaceOvpnClientSetGenral** - globálne nastavenie OVPN klienta
 - interfaceOvpnServer** - nastavenie a správa OVPN serverov
 - interfaceOvpnServerBiding** - nastavenie spravovateľného rozhrania OVPN servera
 - interfaceOvpnServerSet** - nastavenie OVPN servera
- **interfacePpp** - obsahuje triedy:
 - interfacePppClient** - nastavenie PPP klienta
 - interfacePppClientSetGenral** - globálne nastavenie PPP klienta
 - interfacePppClientSetPpp** - nastavenie PPP konfigurácie klienta
 - interfacePppServerDialIn** - nastavenie vytáčania servera
 - interfacePppServerSetGeneral** - nastavenie globálnej konfigurácie ppp server profilu
 - pppAuthenticationAndAccounting** - nastavenie zabezpečenia PPP
- **interfacePppoe** - obsahuje triedy:
 - interfacePppoe** - správa rozhraní PPPoE
 - interfacePppoeClient** - správa PPPoE klientov
 - interfacePppoeClientSetDialOut** - správa nastavenia vytákania klienta
 - interfacePppoeSet** - nastavenie PPPoE rozhrania
 - interfacePppoeSetGeneral** - nastavenie PPPoE globálneho nastavenia
 - pppoe** - správa pppoe rozhraní
 - pppoeSettings** - globálne nastavenie PPPoE
- **interfacePptp** - obsahuje triedy:
 - interfacePptpServer** - nastavenie Pptp servera
 - interfacePptpServerBinding** - nastavenie spojení PPTP servera
 - interfacePptpServerSetGenral** - hlavné nastavenie PPTP server profilu

- interfacePptpClientDialOut** - nastavenie vytáčania PPTP klienta
- interfacePptpClientSetGeneral** - základné nastaveni PPTP klienta
- **interfaceSstp** - obsahuje triedy:
 - interfaceSstpClient** - nastavenie SSTP klienta
 - interfaceSstpClientGeneralSet** - základné nastavenie SSTP klienta
 - interfaceSstpClietSetDialOut** - nastavenie vytáčania SSTP klienta
 - interfaceSstpServer** - nastavenie SSTP serveru
 - interfaceSstpServerBinding** - nastavenie správy spojení servera SSTP
 - interfaceSstpServerSet** - nastavenie SSTP servera
- **l2tpSecrets** - nastavenie hesiel L2TP profilov
- **profile** - obsahuje triedy:
 - profileGeneral** - hlavné nastavenie užívateľov
 - profileLimits** - natsavenie obmedzenia užívateľa
 - profileProtocols** - nastavenie protokolov užívateľa
 - profileQueue** - nastavenie fronty užívateľa
 - profile** - správa užívateľov
 - profileScripts** - nastavenie skriptov pri prihlásení a odhlásení užívateľa
- **secrets** - trieda obsahuje:
 - secrets** - správa hesiel
 - secrtesSettings** - nastavenie hesiel



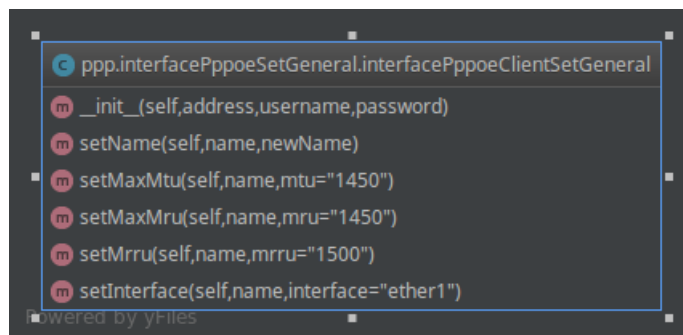
Obr. 7.22: Zoznam tried v zložke PPP

7.10.2 Analyzovaný súbor

Analyzovaný súbor *interfacePppoeSetGeneral.py* je popísaný na UML diagrame 7.23 a v tabuľke metód 7.12.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
setName	meno rozhrania, nové meno rozhrania	slovník	Metóda premenuje rozhranie.
setMaxMtu	meno rozhrania, nastavenie MTU	slovník	Metóda nastaví veľkosť MTU.
setMaxMru	meno rozhrania, veľkosťMRU	slovník	Metóda nastaví veľkosť MRU.
setMrru	meno rozhrania, veľkosťMRRU	slovník	Metóda nastaví veľkosť MRRU.
setInterface	meno rozhrania, rozhranie	slovník	Metóda nastaví rozhranie na profil.

Tab. 7.12: Tabuľka popisu metód triedy interfacePppoeGeneralSet



Obr. 7.23: UML diagram triedy interfacePppoeGeneralSet

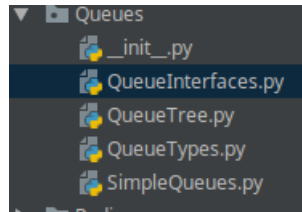
7.11 Zložka Queues

Zložka popisuje nastavenie sietovej fronty rôznych typov. Popis celkovej zložky je rovnaký ako v kapitole 7.1 a ďalších kapitolách.

7.11.1 Zoznam tried zložky

Zložka obsahuje triedy:

- **QueueInterfaces** - nastavenie rozhraní
- **QueueTree** - nastavenie stromu fronty
- **QueueTypes** - nastavenie existujúcich a nových typov fronty
- **SimpleQueues** - nastavenie jednoduchých front



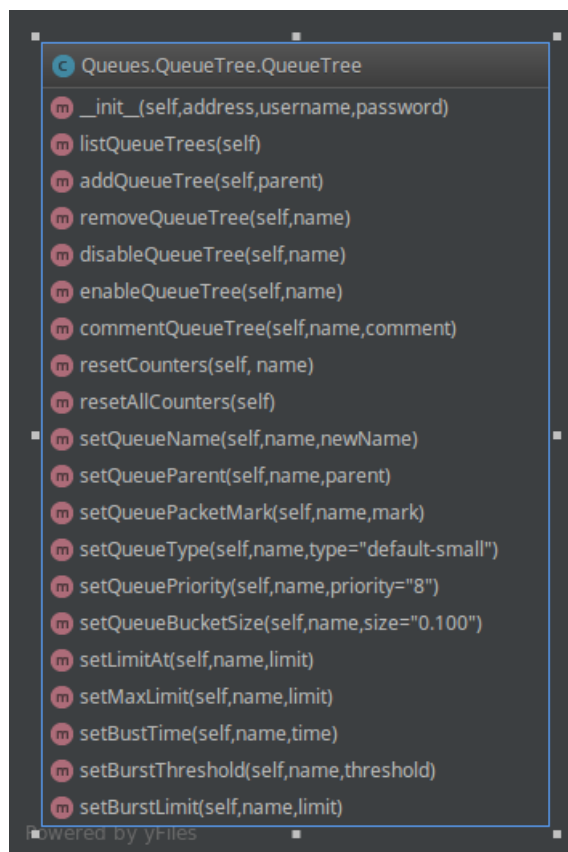
Obr. 7.24: Zoznam súborov zložky Queues

7.11.2 Analáza vybraného súboru

Vybraný analyzovaný súbor *QueueTree.py* je popísaný v tabuľke 7.13 a na UML diagrame 7.25.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listQueueTrees	žiadny	slovník	Metóda vypíše zoznam stromov.
addQueueTree	rodič	slovník	Metóda pridá nový strom.
removeQueueTree	menostromu	slovník	Metóda odstráni strom.
disableQueueTree	meno stromu	slovník	Metóda vypne strom.
enableQueueTree	meno stromu	slovník	Metóda zapne strom.
commentQueueTree	meno stromu, komentár	slovník	Metóda okomentuje strom.
resetCounters	meno stromu	slovník	Metóda zmaže štatistiky stromu.
resetAllCounters	žiadny	slovník	Metóda zmaže všetky štatistiky.
setQueueName	meno, nové meno	slovník	Metóda premenuje strom.
setQueueParent	meno stromu, rodič	slovník	Metóda nastaví rodiča stromu.
setQueuePacketMark	meno stromu, označenie paketu	slovník	Metóda nastaví označenie paketov prichádzajúcich a odchádzajúcich zo stromu.
setQueueType	meno stromu, typ fronty	slovník	Metóda nastaví typ fronty.
setQueuePriority	meno stromu, priorita	slovník	Metóda nastaví prioritu fronty.
setQueueBuckezSize	meno stromu, veľkosť úložiska fronty	slovník	Metóda nastaví veľkosť úložiska na fronty.
setLimitAt	meno stromu, limit	slovník	Metóda nastaví monimálny limit na frontu.
setMaxLimit	meno stromu, maximálny limit	slovník	Metóda nastaví maximálny limit fronty.
setBurstTime	meno sstromu, hodnota zhluku	slovník	Metóda nastaví čas zhluku fronty.
setBurstThreshold	meno stromu, prahová hranice	slovník	Metóda nastaví prahovú hranicu zhluku front.
setBurstLimit	meno stromu, limit zhluku	slovník	Metóda nastaví limit zhluku.

Tab. 7.13: Tabuľka zoznamu metód triedy QueueTree



Obr. 7.25: UML diagram triedy QueueTree

7.12 Zložka Radius

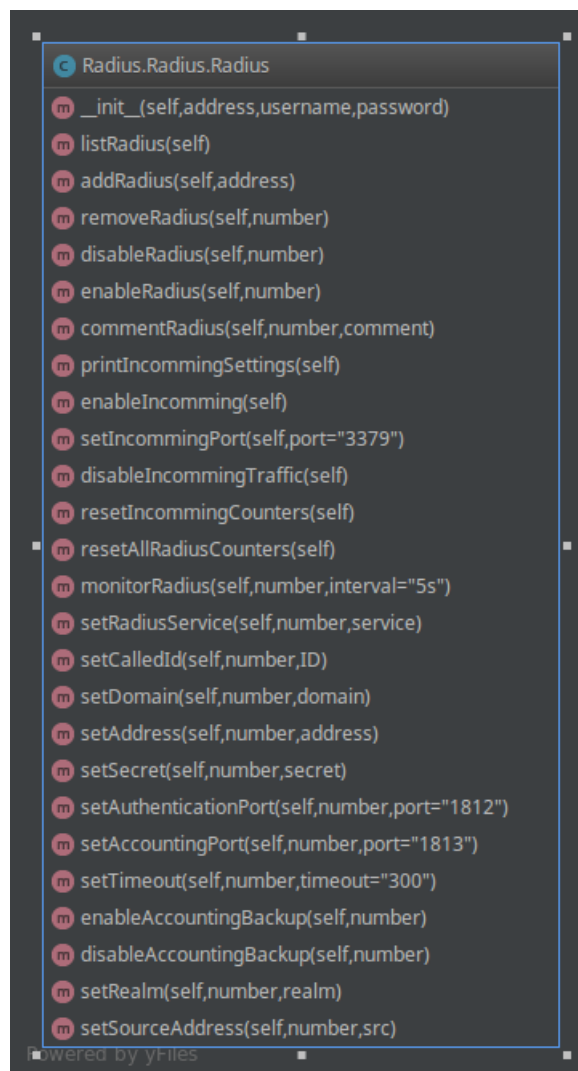
Zložka Radius pozostáva z nastavenia Radiusu na mikrotiku. Radius pozostáva so súborov tried:

- **Radius** - nastavenie a správa radiusu

V tabuľke 7.14 a na UML diagrame 7.26 vidíme popis triedy Radius.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listRadius	žiadny	slovník	Metóda vypíše zoznam RADIUS serverov.
addRadius	adresa	slovník	Metóda pridá nový RADIUS server.
removeRadius	poradové číslo serveru	slovník	Metóda odstráni RADIUS server.
disableRadius	poradové číslo serveru	slovník	Metóda vypne RADIUS server.
enableRadius	poradové číslo serveru	slovník	Metóda zapne RADIUS server.
commentRadius	poradové číslo serveru, komentár	slovník	Metóda okomentuje RADIUS server.
printIncommingSettings	žiadny	slovník	Metóda vypíše štatistiky RADIUS serveru.
enableIncomming	žiadny	slovník	Metóda povolí prichádzajúcu trafiku.
setIncommingPort	port	slovník	Metóda nastaví port na prichádzajúcu trafiku.
disableIncommingTraffic	žiadny	slovník	Metóda vypne prichádzajúcu trafiku.
setQueuePacketMark	meno stromu, označenie paketu	slovník	Metóda nastaví označenie paketov prichádzajúcich a odchádzajúcich zo stromu.
resetIncommingCounters	žiadny	slovník	Metóda resetuje štatistiky spojenia.
resetAllRadiusCounters	žiadny	slovník	Metóda resetuje všetky štatistiky spojenia.
monitorRadius	poradové číslo serveru, interval	slovník	Metóda nastaví monitoring RADIUS serveru.
setRadiusService	poradové číslo serveru, meno služby	slovník	Metóda nastaví službu RADIUS.
setCalledId	poradové číslo serveru, užívateľ	slovník	Metóda nastaví vytáčaného užívateľa.
setDomain	poradové číslo serveru, doména	slovník	Metóda nastaví doménu.
setAddress	poradové číslo serveru, adresa serveru	slovník	Metóda nastaví adresu serveru.
setSecret	poradové číslo serveru, heslo	slovník	Metóda nastaví heslo na server.
setAuthenticationPort	poradové číslo serveru, port	slovník	Metóda nastaví autentikačný port.
setAccountingPort	poradové číslo serveru, port	slovník	Metóda nastaví port protokolu AAA.
setTimeout	poradové číslo serveru, timeout	slovník	Metóda nastaví timeout serveru.
enableAccountingBackup	poradové číslo serveru	slovník	Metóda zapne zálohovanie.
disableAccountingBackup	poradové číslo serveru	slovník	Metóda vypne zálohovanie.
setRealm	poradové číslo serveru, doména	slovník	Metóda nastaví sadu domén.
setSourceAddress	poradové číslo serveru, zdrojová adresa	slovník	Metóda nastaví zdrojovú adresu serveru.

Tab. 7.14: Tabuľka metód triedy Radius



Obr. 7.26: UML diagram triedy Radius

7.13 Zložka routing

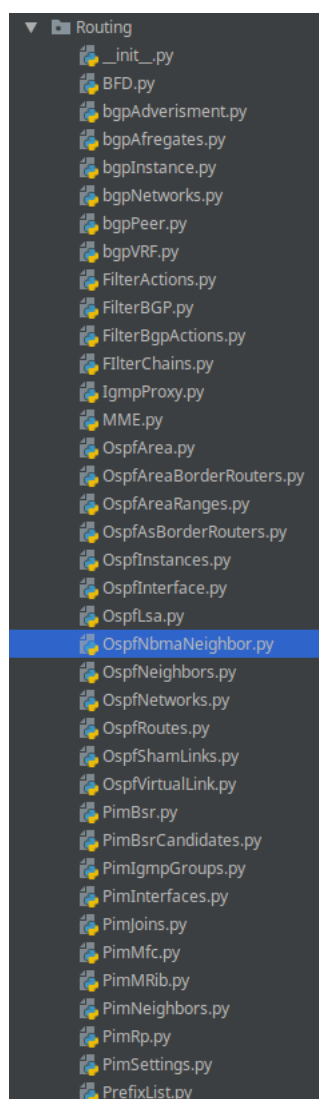
Zložka popisuje možnosti dynamického smerovania na mikrotiku. Rovnako ako kapitoly predtým, jej štruktúra je postavená na základe rovnakom ako je popísaný v kapitolách 7.1.1 a ostatných kapitolách.

7.13.1 Zoznam tried zložiek

Zložka pozostáva z tried:

- **BFD** - trieda ošetruje nastavenie BFD
- **BGP** - triedy ošetrujú nastavenie BGP protokolu
- **Filter** - triedy ošetrujú nastavenie BGP bezpečnosti

- **IgmProxy** - triedy ošetrujú nastavenie IGMP proxy
- **MME** - trieda ošetruje nastavenie MME
- **OSPF** - triedy ošetrujú nastavenie OSPF
- **PIM** - trieda ošetruje nastavenie PIM
- **RIP** - trieda ošetruje nastavenie protokolu RIP
- **RoutingFilter** - triedy ošetrujú nastavenie flultrácie komunikácie protokolu BGP
- **RP** - trieda ošetruje nastavenie Randevou point
- **VPNRoutes** - trieda ošetruje správu ciest VPN tunelu



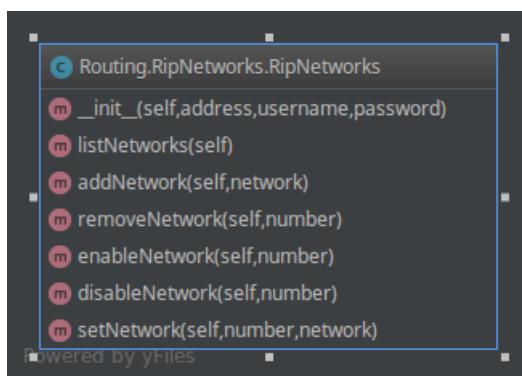
Obr. 7.27: Zoznam súborov zložky Routing

7.13.2 Analýza vybraného súboru

Analýzou vybraného súboru *RipNetworks.py* v UML diagrame 7.28 a v tabuľke 7.15.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listNetwork	žiadny	slovník	Metóda vypíše zoznam sietí.
addNetwork	sieť	slovník	Metóda pridá novú sieť.
removeNetwork	poradové číslo siete	slovník	Metóda odstráni sieť.
disableNetwork	poradové číslo siete	slovník	Metóda vypne sieť.
enableNetwork	poradové číslo siete	slovník	Metóda zapne sieť.
setNetwork	poradové číslo siete, sieť	slovník	Metóda nastaví existujúcu sieť.

Tab. 7.15: Tabuľka popisu metód triedy RipNetworks



Obr. 7.28: UML diagram súboru RipNetworks

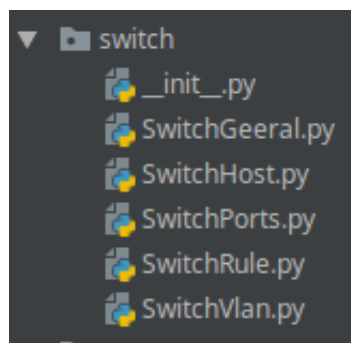
7.14 Zložka switch

Cieľom zložky switch je nastavenie virtuálneho switchu na mikrotiku. Jeho infraštruktúra je identická s kapitolami 7.1.1 a ostatnými kapitolami.

7.14.1 Zoznam tried zložky

Obsahom zložky sú triedy:

- **SwitchGeneral** - hlavné nastavenie prepínača
- **SwitchHost** - nastavenie hostov
- **SwitchPorts** - nastavenie portov prepínača
- **SwitchRule** - nastavenie pravidiel filtrovania trafiky na prepínač
- **SwitchVlan** - nastavenie virtuálnych LAN sietí



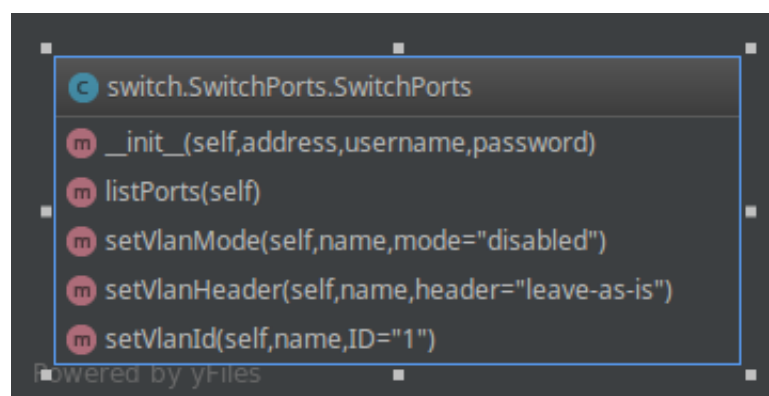
Obr. 7.29: Zoznamsúborov zložky switch

7.14.2 Analýza vybraného súboru

Analýza vybraného súboru *switchPorts.py* je odzrkadlená v UML diagrame 7.30 a v tabuľke metód 7.16.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
listPorts	žiadny	slovník	Metóda vypíše zoznam portov switchu.
setVlanMode	meno switchu, vlan mód	slovník	Metóda nastaví VLAN mód portu.
setVlanHeader	meno switchu, hlavička VLAN	slovník	Metóda nastaví hlavičku VLAN.
setVlanId	meno switchu, VLAN ID	slovník	Metóda nastaví číslo VLAN.

Tab. 7.16: Tabuľka zoznamu metód triedy switchPort



Obr. 7.30: UML digram triedy switchPorts

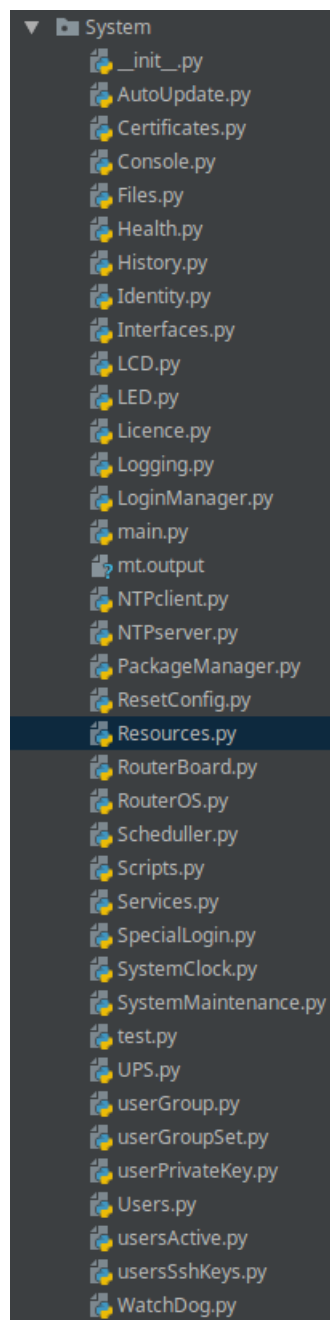
7.15 Zložka System

Účelom zložky system je nastavenie systémových nástrojov prvku mikrotik. Jej infraštruktúra je totžná k infraštruktúre kapitoly 7.1.1 a ďalších kapitol.

7.15.1 Zoznam súborov zložky

Zložka pozostáva zo súborov:

- **AutoUpdate** - správa automatických aktualizácií
- **Certificates** - správa systémových certifikátov
- **Console** - nastavenie konzolového portu
- **Files** - správa súborvej infraštruktúry
- **Health** - správa kontroly stavu hardvéru mikrotiku
- **History** - správa histórie zmien na mikrotiku
- **Identity** - správa nastavenia hostname
- **Interfaces** - správa nastavenia konzolových rozhraní
- **LCD** - správa nastavenia kontroly LCD displeja
- **Licence** - správa licencie na mikrotiku
- **Logging** - správa logovania
- **NTPClient** - správa klienta protokolu NTP
- **NTPServer** - správa serveru protokolu NTP
- **PackageManager** - správa aktualizácií mikrotiku a systémových balíkov
- **ResetConfig** - správa resetovania konfigurácie mikrotiku
- **RouterBoard** - správa získavania informácií o Routerboarde
- **RouterOS** - správa informácií o operačnom systéme
- **Scheduller** - správa plánovaných úloh
- **Scripts** - správa systémových skriptov
- **Services** - správa systémových služieb
- **SpecialLogin** - správa špeciálneho admin prihlasovacieho účtu
- **SystemClock** - správa systémového času
- **SystemMaintenance** - správa reštartu a vypnutia mikrotiku
- **UPS** - správa konektoru na mikrotiku (AC adaptéru)
- **user** - triedy nastavujú užívateľov, skupiny,...
- **WatchDog** - správa kontroly prvkov na mikrotiku



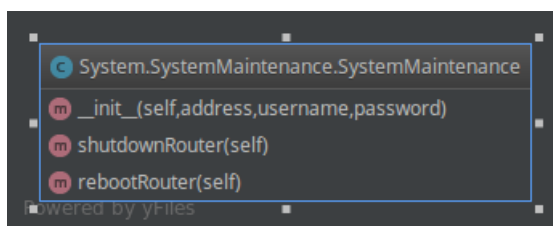
Obr. 7.31: Zoznam súborov triedy system

7.15.2 Analýza vybraného súboru

Vybraný analyzovaný súbor *SystemMaintenance.py* je popísaný v UML diagrame 7.32 a v tabuľke 7.17.

Názov metódy	Vstup	Výstup	Vysvetlenie metódy
shutdownRouter	žiadny	slovník	Metóda vypne router
rebootRouter	žiadny	slovník	Metóda reštartuje router.

Tab. 7.17: Tabuľka metód triedy SystemMaintenance



Obr. 7.32: UML diagram triedy SystemMaintenance

LITERATÚRA

- [1] *Manual:API* [online]. 2014, [cit. 24. 03. 2018]. Dostupné z URL: <<https://wiki.mikrotik.com/wiki/Manual:API>>
- [2] *How to Install Configure Easy-RSA* [online]. 2013, [cit. 24. 03. 2018]. Dostupné z URL: <<https://docs.bigchaindb.com/projects/server/en/latest/production-deployment-template/easy-rsa.html>>
- [3] *MAC Level Access (Telnet and Winbox)* [online]. 2007, [cit. 26. 03. 2018]. Dostupné z URL: <<https://mikrotik.com/testdocs/ros/2.9/tools/mactelnet.php>>
- [4] Mitchell Anicas *OpenSSL Essentials: Working with SSL Certificates, Private Keys and CSRs* [online]. 2012, [cit. 24. 03. 2018]. Dostupné z URL: <<https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-certificates-private-keys-and-csrs>>
- [5] *os — Miscellaneous operating system interfaces* [online]. 2012, [cit. 04. 04. 2018]. Dostupné z URL: <<https://docs.python.org/2/library/os.html>>
- [6] *CISCO: Open Shortest Path First (OSPF)* [online]. 2009, [cit. 09. 11. 2014]. Dostupné z URL: <<http://www.cisco.com/c/en/us/products/ios-nx-os-software/open-shortest-path-first-ospf/index.html>>.
- [7] *Pexpect version 4.4* [online]. 2018, [cit. 02. 04. 2018]. Dostupné z URL: <<https://pexpect.readthedocs.io/en/stable/>>.
- [8] *Installation* [online]. 2018, [cit. 02. 04. 2018]. Dostupné z URL: <<https://pexpect.readthedocs.io/en/stable/install.html>>.
- [9] *pxssh (version 2.3)* [online]. 2018, [cit. 02. 04. 2018]. Dostupné z URL: <<http://pexpect.sourceforge.net/pxssh.html>>.
- [10] *Python 2.7.2 Release* [online]. 2018, [cit. 02. 04. 2018]. Dostupné z URL: <<https://www.python.org/download/releases/2.7.2/>>.
- [11] Tatu Ylonen *SSH PROTOCOL*) [online]. 2017, [cit. 26. 03. 2018]. Dostupné z URL: <<https://www.ssh.com/ssh/protocol/>>
- [12] *What is telnet?*) [online]. 2018, [cit. 26. 03. 2018]. Dostupné z URL: <<https://kb.iu.edu/d/aayd>>
- [13] *telnetlib — Telnet client*) [online]. 2018, [cit. 04. 04. 2018]. Dostupné z URL: <<https://docs.python.org/3.1/library/telnetlib.html>>

- [14] *Manual: Webfig* [online]. 2018, [cit. 26.03.2018]. Dostupné z URL: <<https://wiki.mikrotik.com/wiki/Manual:Webfig>>
- [15] *What's New In Python 3.0* [online]. 2018, [cit. 02.04.2018]. Dostupné z URL: <<https://docs.python.org/3.0/whatsnew/3.0.html>>.
- [16] *tikapy* [online]. 2018, [cit. 02.04.2018]. Dostupné z URL: <<https://github.com/vshn/tikapy/blob/master/README.rst>>.
- [17] *Manual: Winbox* [online]. 2018, [cit. 26.03.2018]. Dostupné z URL: <<https://wiki.mikrotik.com/wiki/Manual:Winbox>>
- [18] *Install the Certification Authority* [online]. 2017, [cit. 24.03.2018]. Dostupné z URL: <<https://docs.microsoft.com/en-us/windows-server/networking/core-network-guide/cncg/server-certs/install-the-certification-authority>>

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

AP Prístupový bod

API Application programable interface

API-SSL Application programable interface Secure Socket Layer

FTP File Transfer Protocol

GUI Graphical User Interface

IDE Integrated Developement Envinroment

IP Internet Protocol

IPSEC Internet Protocol Security

MAC macintosh

MAC Media Access Control

MPLS Multi Protocol Label Swiching

OS Operačný systém

PPP Point to Point Protocol

SSL Secure Socket Layer

TLS Transport Layer Security

UML Unified Modeling Language

VPN Virtual Private Network

ZOZNAM PRÍLOH