VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019 Matej Pancák

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

ŘÍZENÍ SYSTÉMU FRONT NA SÍŤOVÝCH PRVCÍCH
QUEUE MANAGEMENT IN NETWORK ACTIVE ELEMENTS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
Matej Pancák

VEDOUCÍ PRÁCE
SUPERVISOR
doc. Ing. Jaroslav Koton, Ph.D.

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Matej Pancák ***ID:*** 195409
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Řízení systému front na síťových prvcích

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte nástroj Queue System síťových prvků Mikrotik pro řízení datového provozu. Navrhněte efektivní způsob pro značení datových jednotek a mechanismus pro poloautomatickou či zcela automatickou aktualizaci pravidel značení. Využitím mechanismů pro vyhodnocení datového provozu navhněte skripty, které budou aktivované buď přímo na samotném síťovém prvku, či zpracovány ve vhodném externím SW, vytvořen aktuální seznam pravidel pro značení datových jednotek a tento aplikován na síťovém prvku.
Mechanismus ověřte jeho nasazením na reálném datovém provozu a zhodnoťte jeho efektivitu.

DOPORUČENÁ LITERATURA:

[1] A.S. Tanenbaum, D.J. Wetherall: Computer networks, Pearson, 2010, ISBN: 978-0132126953.

[2] Scripts, Mikrotik documentation [online]. 2017 [cit. 2018-09-16]. Dostupné z:

http://wiki.mikrotik.com/wiki/Scripts

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: doc. Ing. Jaroslav Koton, Ph.D.
Konzultant:

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

ABSTRAKT

Táto práca sa zaoberá problematikou efektivity spracovania dát na aktívnych sieťových prvkoch a následnom monitorovaní prenosových rýchlostí v testovacej a reálnej premávke. Cieľom práce je navrhnúť a implementovať algoritmus schopný automaticky optimalizovať zoznam pravidiel v nástroji Mangle. Dôsledkom tejto optimalizácie je zvýšenie efektivity spracovania dátových jednotiek a zníženie oneskorenia ako hlavného parametru QoS. Výsledkom práce je séria meraní s cieľom potvrdiť vplyv optimalizácií na parametre prenosu.

KĽÚČOVÉ SLOVÁ

Oneskorenie, QoS, Mangle, Queue, MikroTik

ABSTRACT

This thesis is mainly focused on efficiency of data handling on active network elements and monitoring of transmission speeds in real and testing networks. The purpose of this work is to create and implementate algorhytm which is able to reorganize list of Mangle rules. The purpose of this optimization is to raise data processing efficency and decrease communication delay as main QoS parameter. The result of the work is series of measurements in order to confirm the effect of optimizations on the transmission and other parameters.

KEYWORDS

Delay, QoS, Mangle, Queue, MikroTik

PANCÁK, Matej. *Řízení systému front na síťových prvcích*. Brno, 2019, 54 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Jaroslav Koton, Ph.D.

VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Řízení systému front na síťových prvcích“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

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

Brno ....

podpis autora

POĎAKOVANIE

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

Brno ....

podpis autora

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

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

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Projekt je spolufinancován Evropskou unií.

Obsah

Zoznam obrázkov

Úvod

Žijeme v dobe pokročilých multimediálnych systémov a internetu, v dobe kedy si každodenný život bez pripojenia na internet nedokážeme predstaviť. Rýchlosť pripojenia a spracovania dát, pri súčasných možnostiach streamovaného videa a IP televízie, je kľúčová.

Táto práca sa bude venovať optimalizácií sieťových prvkov MikroTik, jej cieľom bude zvýšiť efektivitu spracovania dát pomocou vhodných usporiadaní pravidiel v značkovacom nástroji Mangle implementovaného v systémoch MikroTik. Zvýšením efektivity spracovania, chcem docieliť pozitívneho ovplyvnenia jedného z kľúčových parametrov QoS a to oneskorenia.

V prvej časti sa budem zaoberať funkcionalitou a princípom novodobej dátovej komunikácie. Priblížime si základné princípy a prvky novodobých dátových sietí, predstaíme si hlavné parametre QoS. Budeme sa venovať oneskoreniu, ako jednému z hlavných parametrov QoS a opíšeme si princípy ako vzniká a predstavíme možnosti ako ho redukovať.

Systém MikroTik je v súčastnosti rozšíreným a obľúbeným prvkom v sieťach väčšieho, ale aj menšieho rozsahu, preto sa v nasledujúcej časti budem venovať práve tomu ako tento systém funguje a aké možnosti ponúka. Priblížime si aké vlastnosti tohto systému bolo možné použiť pri riešení tejto práce.

V záverečnej časti sa budem venovať konkrétnemu postupu riešenia a implementácií. Budem sa podrobnejšie venovať operačnému systému RouterOS a jeho možnostiam. V prehľadných grafoch je znázornená prenosová rýchlosť a iné dôležité parametre, ktoré sú predmetom tejto práce. Výstupom práce sú výsledky dopadu poradia pravidiel v nástroji Mangle na prenosovú rýchlosť a ďalšie sledované parametre ako napríklad zaťaženie CPU.

1 Quality of Service

V posledných rokoch sme zaznamenali obrovský pokrok v oblasti počítačov, mobilných zariadení a všeobecne možnosť pripojenia k internetu nadobudli aj zariadenia, ktoré túto možnosť v minulosti nemali. Rýchly pokrok a zvýšené požiadavky na siete viedli k nárastu záujmu o QoS (Quality of Service) a o zefektívnenie riadenia dátového toku všeobecne. Aby sa zaručilo, že multimediálnym aplikáciám bude zaručená požadovaná kvalita QoS, nestačí len sprostredkovať zdroje. Dôležité je, aby distribuované multimediálne aplikácie zabezpečovali end-to-end QoS mediálnych tokov, vzhľadom na siete a koncové terminály. Degradácia v zmluvne viazanej službe QoS je často nevyhnutná, a preto je potrebné zabezpečiť monitorovanie QoS v reálnom čase, ktoré nielenže je schopné monitorovať podporu služby QoS v sieti, ale môže tiež vykonávať činnosti v reálnom čase, aby bola udržaná prijateľná kvalita multimediálnej prezentácie pri degradácii úrovne QoS. V súčasnosti existujú rôzne druhy sietí, káblové a bezdrôtové, ktoré navzájom spolupracujú. Tieto siete majú vlastnosti QoS, ktoré sú drasticky odlišné a ktorých stupeň variability rôznych parametrov QoS, ako je šírka pásma, oneskorenie a kolísanie oneskorenia (jitter), sa značne líšia. Okrem toho existujú rôzne druhy terminálov, ako sú stolné počítače, prenosné počítače a mobilné telefóny, z ktorých každá má určitú multimediálnu podporu. Dátový prenos preto musí prispôsobiť svoju QoS podľa heterogénnych terminálov s variabilnými požiadavkami a podporou QoS [1] [2].

1.1 Hlavné parametre QoS

1.1.1 Šírka pásma a priepustnosť

Šírka pásma a priepustnosť je QoS parameter, ktorý sa vzťahuje na rýchlosť prenosu dát podporovanú sieťovým pripojením alebo rozhraním. Pre šírku pásma je najbežnejším ukazateľom počet bitov za sekundu (bps), t.j. počet prenesených dátových jednotiek za jednotku času. Multimediálne aplikácie zvyčajne vyžadujú vysokú šírku pásma v porovnaní s inými všeobecnými aplikáciami. Sieťové technológie, ktoré nepodporujú takúto vysokú šírku pásma, nemôžu prehrávať multimediálny obsah. Napríklad, technológia Bluetooth verzia 1 podporuje iba maximálnu šírku pásma 0,746 Mbps a teda zariadenia, ktoré spoliehajú na pripojenie Bluetooth, nemôžu prehrávať videá MPEG1, ktoré vyžadujú približne 1 – 2 Mbps [1]. Lepšia priepustnosť znamená lepšiu kvalitu QoS prijatú koncovým používateľom.

1.1.2 Kolísanie oneskorenia

Kolísanie oneskorenia (Jitter) sa odvoláva na časové rozdiely medzi paketmi prichádzajúcimi do cieľa. Je to spôsobené preťažením siete, časovým posunom alebo zmenou trasy. V závislosti od typu multimediálnej aplikácie môže jitter byť alebo nemusí byť významný. Napríklad aplikácie pre audio a video konferencie nie sú tolerantné k jitteru kvôli veľmi obmedzenému vyrovnávaciemu bufferu v živých prezentáciách, zatiaľ čo predregistrované multimediálne prehrávanie je zvyčajne tolerantné k jitteru, pretože moderné prehrávače ukladajú do vyrovnávacej pamäte okolo 5 s záznamu na zmiernenie vplyvu jittru [1].

1.1.3 Stratovosť

Strátovosť (Loss) označuje hlavne množstvo údajov, ktoré sa v určenom časovom intervale nedostali do cieľa. Na zníženie pravdepodobnosti straty možno použiť rôzne metódy. Napríklad poskytnutím individuálnych kanálov/garantovanej šírky pásma pre špecifické prenosy údajov, alebo retransmisiou údajov na zotavenie po strate [1].

1.1.4 Oneskorenie

Oneskorenie (Delay) je definované ako časový interval, ktorý uplynul medzi odchodom údajov od zdroja až po jeho príchod do cieľa. V prípade komunikačného systému sa oneskorenie vzťahuje na oneskorenie medzi odchodom signálu zo zdroja a jeho príchodom na miesto určenia. Môže sa pohybovať v rozmedzí od niekoľkých milisekúnd ($1 - 2\,\text{ms}$) v lokálnych sieťach (LAN), prenos medzi kontinentami môže trvať až do $100\,\text{ms}$ [3].

Z hľadiska oneskorenia je veľmi dôležitým parametrom tiež obojsmerné oneskorenie, ktoré svoj význam nadobudlo aj z hľadiska problematickej synchronizácie hodín pri meraní jednosmerného oneskorenia. Obojsmerné oneskorenie teda značí oneskorenie trasy tam a späť. V angličtine ja zaužívaný pojem *round-trip-time* (RTT). Na celkové oneskorenie má vplyv hneď niekoľko elementov siete, je zložené teda z viacerých čiastočných oneskorení popísaných v texte nižšie:

Oneskorenie spôsobené šírením signálu

Má nezanedbateľný dopad na komunikáciu v sieti, hlavne pri komunikácií na veľké vzdialenosti. Keďže pri prenášaní signálu prenášacím médiom sme vždy limitovaný rýchlosťou svetla, ktorá je $3.10^8\,\text{m/s}$ vo vakuu, $2,3.10^8\,\text{m/s}$ v medenom kábli a $2.10^8\,\text{m/s}$ v optickom kábli [3]. Vďaka neustálemu pokroku na poli optických technológií sme schopný aj na obrovské vzdialenosti dosahovať minimálnych oneskorení,

avšak ako bolo povedané, kvôli tomu, že sme stále limitovaný fyzikálnymi zákonmi, nikdy toto oneskorenie spôsobené samotným prenosom po médiu nebude možné úplne eliminovať.

Oneskorenie spôsobené dobou vysielania

Dobou vysielania rozumieme časový úsek nutný k odoslaniu paketu z uzla, je to teda doba, ktorá uplynie medzi zahájením vysielania prvého bitu a vysielaním posledného bitu dátovej jednotky. Toto oneskorenie je na prvý pohľad takmer zanedbateľné pretože dĺžka jednej dátovej jednotky je len niekoľko stovák bitov. Avšak prvok siete nepracuje len s jednou takouto jednotkou ale aj s tisícmi podobných dátových jednotiek. Z hľadiska prenosu dát služieb bežiacich v reálnom čase sa javí výhodné voliť veľkosť paketov skôr menšiu, pretože tieto pakety je možné v komunikačnom uzle rýchlo spracovať, skontrolovať a odoslať k ďalšiemu uzlu. Teda príliš veľká veľkosť paketov zvyšuje oneskorenie dobou vysielania, teda doba medzi odoslaním prvého a posledného bitu paketu je vyššia [3]. Na druhú stranu napríklad pri prenose dát pomocou FTP sa malá veľkosť prenášaných paketov nejaví výhodná z dôvodu nízkeho pomeru užitočných dát voči hlavičke. Je teda zjavné, že aj veľkosť paketu odosielaná uzlom môže mať vplyv na efektivitu prenosu a samotnú rýchlosť pripojenia, ale úzko závisí na type komunikácie.

Oneskorenie v uzle

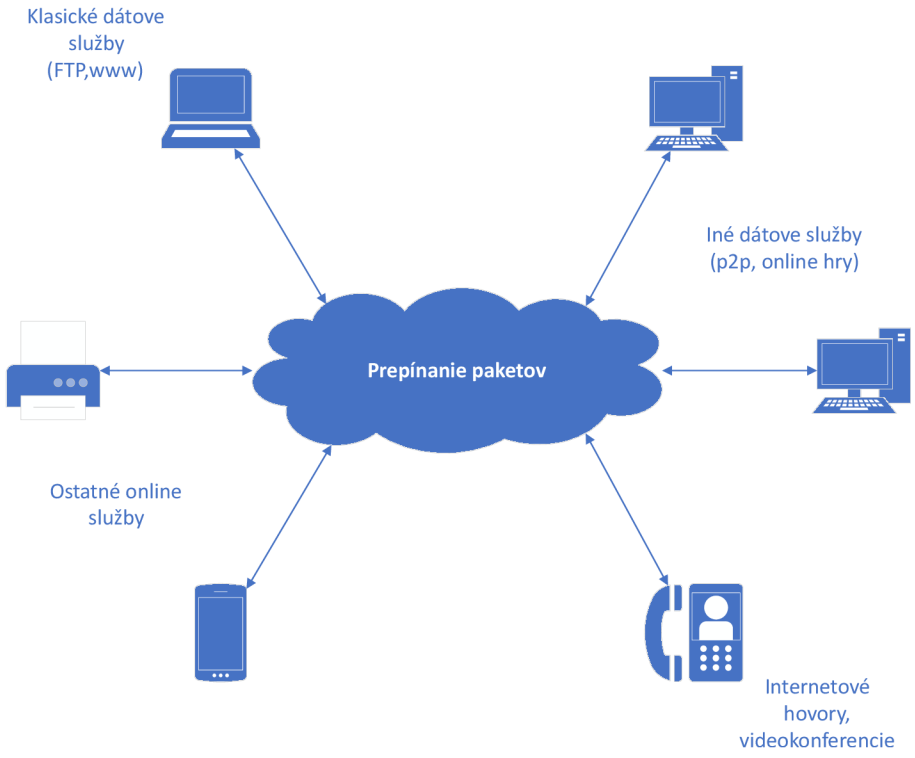
V každom aktívnom sieťovom prvku je nutné prichádzajúce pakety spracovať. Spracovanie a následné úkony, ktoré je potrebné s prenášanými dátami vykonať, zaberajú určitý čas. Tento čas taktiež prispieva k celkovému oneskoreniu pri prenose.

1.2 Mechanizmy zaisťujúce QoS v dátových sieťach

Aby bolo možné v moderných sieťach založených na prepínaní dátových jednotiek (paketov) zaistiť istý typ garantovaného prenosu, bolo nevyhnutné nahradiť zastaralé sieťové technológie, ktoré bývali často založené na zdanlivo spravodlivom mechanizme „best effort“. V rámci tohoto zdanlivo spravodlivého mechanizmu sa aktívny prvok snaží vyhovieť všetkým typom komunikácie s rovnakou prioritou, avšak bez akejkoľvek garancie. Prvok nedokáže garantovať žiadne časové medze pre spracovanie dátovej jednotky, dokonca ani to, či bude daná jednotka vôbec spracovaná. Tento nedostatok sieťových prostriedkov síce ovplyvňuje všetky dátové toky, ale má odlišný dopad na rôzne typy sieťových služieb. Napríklad pri prenose veľkého súboru môže mať narušenie integrity a strata niektorých paketov fatálne následky, ale prípadne zvýšené oneskorenie nemá žiaden vplyv na výslednú integritu dát. Naopak,

pri prenose digitálneho hlasu alebo videa je vyžadované malé oneskorenie, ale služba je schopná tolerovať určitú mieru strát.

V telekomunikačných sieťach zaisťujúcich rôzne typy služieb (Obr. 1.1) s rôznymi požiadavkami na prenos, je takýto mechanizmus spracovania dát nevyhovujúci. Aby bol prenos viacerých rozdielnych služieb cez jednu sieť efektívny, musia sa implementovať mechanizmy, ktoré dokážu rozlíšiť dátové jednotky jednotlivých dátových služieb a následne im zaistiť istý spôsob zachádzania. V súčastnosti je jedným z najpoužívanejších riešení otázky QoS mechanizmus Diferencovaných služieb (*DiffServ*), ktorý delí sieťovú premávku do niekoľkých tried a následne zaisťuje rôzne zaobchádzanie s danými triedami. Predtým bol často využívaným prístupom k zaisteniu QoS mechanizmus integrovaných služieb (*IntServ*) [4]. Bližší popis týchto mechanizmov je uvedený v sekcií 1.2.7.



Obr. 1.1: Moderná komunikačná sieť.

1.2.1 Klasifikácia sieťového prenosu

Aby bolo možné implementovať QoS do siete musia byť splnené isté požiadavky. Jednou z nevyhnutných vecí, bez ktorých by QoS nemohlo fungovať je aj klasifikácia sieťového prenosu (packet classification), ktorá je zároveň aj prvým krokom pri triedení dátových jednotiek [2].

Klasifikáciou sieťového prenosu môžeme nazvať proces radenia paketov do skupín na základe vopred dohodnutých pravidiel. Mechanizmy sa z pravidla riadia podľa informácií uložených v hlavičke dátovej jednotky. Medzi dva najčastejšie typy klasifikácie patria [4]:

- Zlúčené vyhodnotenie (Behaviour Aggregate – BA). Tento typ klasifikácie vyberá dátovej jednotky podľa jediného identifikátoru a to podľa značky umiestnenej v hlavičke IP paketu, v poli DSCP.
- Viacpoložková klasifikácia (Multi-Field Classification – MF) vyberá pakety na základe jednej, alebo viacerých položiek v hlavičke protokolu IP, poprípade TCP/UDP, ako napríklad: zdrojová adresa, cílová adresa, port resp. ich kombinácie.

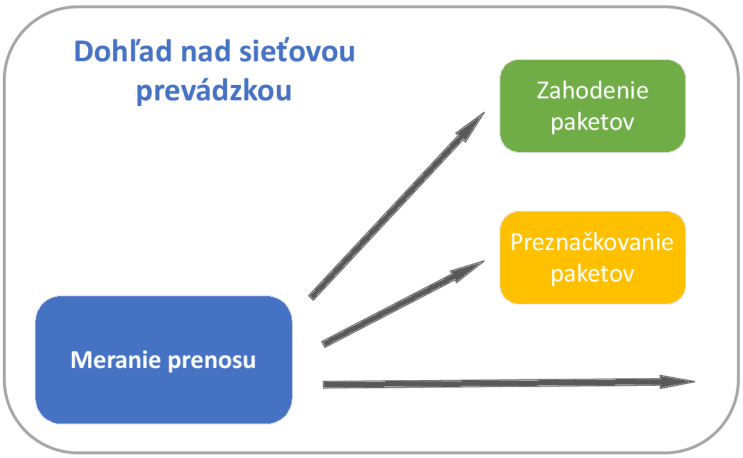
1.2.2 Značkovanie paketov

Značkovanie paketov slúži k označeniu príslušnosti paketu do danej triedy a je druhým krokom procesu triedenia dátových jednotiek. Príkladom značky môže byť napríklad IP adresa zroja, IP adresa cieľového portu, typ prenosového protokolu, DSCP, poprípade ich kombinácia. Samotná značka je vytvorená vložením istých hodnôt záhlavia IP datagramu.

Ak je paket vstupujúci do smerovača už označený iným prvkom v sieti, môže smerovač zmeniť značku. K preznačovaniu paketov dochádza najčastejšie kvôli tomu, že paket vybočuje z predom dohodnutých parametrov prenosu. Ďalším dôvodom preznačovania paketov môže byť skutočnosť, že paket prechádza z jednej siete do druhej, kde sa používa odlišný spôsob resp. pravidlá značenia [4].

1.2.3 Dohľad nad sieťovou prevádzkou

Ďalším dôležitým elementom pre zaistenie QoS je dohľad nad sieťovou prevádzkou. Úlohou tohto mechanizmu je zaistiť aby sa dátový tok vstupujúci do siete pohyboval v medziach dohodnutých medzi zákazníkom a poskytovateľom pripojenia. Dohľad nad prevádzkou sa uskutočňuje na základe výsledkov merania dátových tokov, kedy po prekročení istých vopred dohodnutých hraníc sa zvolí ďalší spôsob spracovania viz Obr. 1.2. Zvolený spôsob spracovania môže napríklad opätovne preznačkovať paket, zahodiť paketu alebo zachovať pôvodnú značku. Ak dohľad nad sieťovou prevádzkou zistí, že dochádza k prekročeniu dohodnutých parametrov prenosu, tak môže dôjsť k preznačeniu, vrámci ktorého je paketom pridelená značka prideľujúca nižšiu prioritu. Následne v prípade potreby môžu smerovače spracovanie paketov odložiť resp. dané pakety prednostne zahodiť [4].



Obr. 1.2: Dohľad nad sieťovou prevádzkou.

1.2.4 Meranie prenosu

Meranie prenosov je dôležitou súčasťou systému dohľadu nad sieťovou prevádzkou, kde sa kontroluje prevádzka prichádzajúca na vstupné porty. Medzi najčastejšie overované parametre patria: garantovaná priemerná prenosová rýchlosť (Commited Information Rate – CIR) a maximálna okamžitá prenosová rýchlosť (Peak Information Rate – PIR) [4].

Maximálna okamžitá prenosová rýchlosť PIR určuje maximálny povolený počet bitov odoslaných v jednom okamihu. Ide o typický parameter prenosu, vopred dohodnutý medzi poskytovateľom pripojenia a zákazníkom v dohode o úrovni služieb SLA (Service level Agreement).

Garantovaná priemerná prenosová rýchlosť CIR špecifikuje dlhodobú priemernú rýchlosť dát, ktorých prenos je zaručený užívateľom vrámci dohody SLA. Základnou jednotkou tohto parametru je množstvo bytov prenesených za jednotku času teda Bps. Na základe toho, že pakety sú často prenášané v zhlukoch prerušovaných kratšími či dlhšími pauzami, je dlhodobá priemerná rýchlosť CIR menšia ako PIR [4]. Meranie prenosu uskutočňuje hneď niekoľko mechanizmov. Medzi najpoužívanejší z nich patrí mechanizmus *Token - Bucket* (TB). Výsledky merania sú následne zohľadnené pri procese značkovanie alebo rozhodovaní o zahodení paketu.

Mechanizmu TB si je možné predstaviť ako „nádobu“, ktorá obsahuje určitý počet tokenov. Každý z týchto tokenov je v podstate povolením k odoslaniu istého množstva dát, najčastejšie jeden token predstavuje 1 byte dát. Na počiatku merania je nádoba plná tokenov, po príchode paketu sa overí, či nádoba obsahuje dostatočné množstvo paketov aspoň odpovedajúce veľkosti daného paketu, ak obsahuje, paket je poslaný k ďalšiemu spracovaniu (značkovanie, zaradenie do fronty). Zároveň je z nádoby odobraný odpovedajúci počet tokenov vzhľadom k veľkosti spracovaného

paketu. Pokiaľ nádoba neobsahuje dostatočný počet tokenov, môže byť paket spracovaný alternatívnou cestou teda môže byť zahodený alebo uložený do vyrovnávacej pamäte, kde bude pozdržaný kým sa nádoba nenaplní dostatočným počtom tokenov. Tokeny sú do nádoby doplňovaní plynule s konštantnou rýchlosťou, pokiaľ nádoba nie je plná [4].

1.2.5 Riadené odosielanie paketov

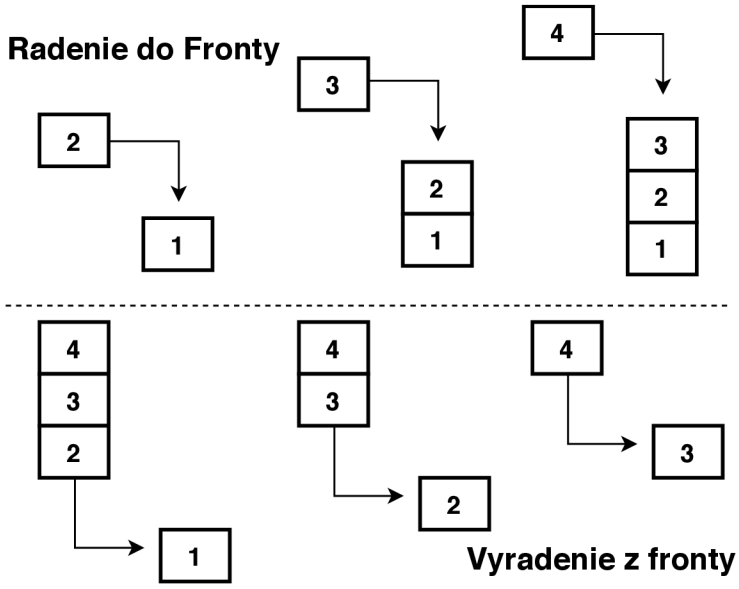
Dôležitou súčasťou mechanizmu k zaisteniu odlišného zaobchádzania s rôznymi druhmi dátových tokov v smerovačoch je radenie dátových jednotiek do oddelených front a diferencovaný spôsob odosielania paketov z týchto front. Okrem samotného odosielania paketov podľa príslušného prioritného mechanizmu, je ďalším dôležitým krokom riadenie odosielania a dohľad nad dostupnými sieťovými prostriedkami, predovšetkým nad šírkou pásma výstupného portu.

Riadené odosielanie paketov plánuje každý port samostatne. Na základe informácií v smerovacej tabuľke sú prichádzajúce pakety najprv prenesené na požadovaný výstupný port. Každý výstupný port realizuje klasifikáciu paketov a zaradí ich do príslušných front. Následne potom blok riadenia určí, z ktorej fronty bude odoslaný paket na výstup [4]. V nasledujúcom text si bližšie priblížime najbežnejšie metódy riadeného odosielania paketov.

First-In-First-Out (FIFO)

Fronta typu FIFO je jednou z prvých metód na organizáciu a manipuláciu dátového toku vo vyrovnávacej pamäti, kde je najskôr spracovaný najstarší (prvý) záznam, teda správy opúšťajú frontu v poradí v akom do nej prišli. Pre svoju jednoduchosť sa teda nejedná úplne o frontu, ktorá by svojím mechanizmom pomáhala realizovať QoS. Často využívanou modifikáciou FIFO je jeho opozitum a to LIFO (Last In, First Out), kde je najprv spracovaný najmladší vstup resp. „top of stack“ [5].

Komunikačné sieťové prvky ako napríklad prepínače a smerovače používajú FIFO na uchovávanie paketov na ceste k ich ďalšiemu cieľu. FIFO je predvoleným algoritmom riadenia paketov do front takmer na každom rozhraní. Princíp FIFO je znázornený na obrázku 1.3.



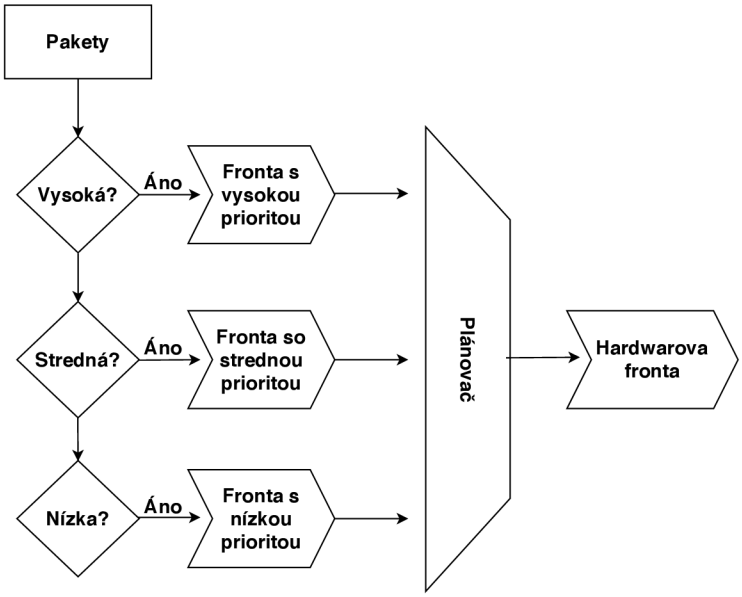
Obr. 1.3: FIFO.

Priority queueing (PQ)

Ďalším zo systémov plánovaného odosielania paketov je prioritný systém front PQ. Mechanizmus PQ obsahuje štyri fronty: fronty s vysokou, strednou, normálnou resp. východzou a nízkou prioritou [6].

Ak nebude paket priradený do jednej z týchto front bude automaticky preradený do východzej čakacej fronty. Pokiaľ má fronta s vysokou prioritou pakety, plánovač PQ presmeruje pakety iba z fronty s vysokou prioritou. Ak je fronta s vysokou prioritou prázdna, spracuje sa jeden paket z fronty strednej priority. Ak sú obidve fronty s vysokou a strednou prioritou prázdne, spracuje sa jeden paket z fronty s normálnou prioritou a ak sú fronty s prioritami pre vysokú, strednú a normálnu prioritu prázdne, spracuje sa jeden paket z fronty s nízkou prioritou. Po spracovaní resp. vyradení jedného paketu (z ľubovoľného radu) sa plánovač vždy spustí opäť tým, že skontroluje, či fronta s vysokou prioritou má nejaké pakety čakajúce, skôr ako skontroluje poradie nižších priorít v poradí.

Avšak pri používaní PQ je nutné si uvedomiť, že pokiaľ je väčšina prichádzajúcich paketov priradená do fronty s vysokou prioritou, žiadnej inej fronte nebude venovaná pozornosť [6]. S týmto javom často nazývaným aj „hladovka front s nižšou prioritou“, je nutné pri využívaní PQ rátať. Obrázok 1.4 zobrazuje princíp mechanizmu PQ.



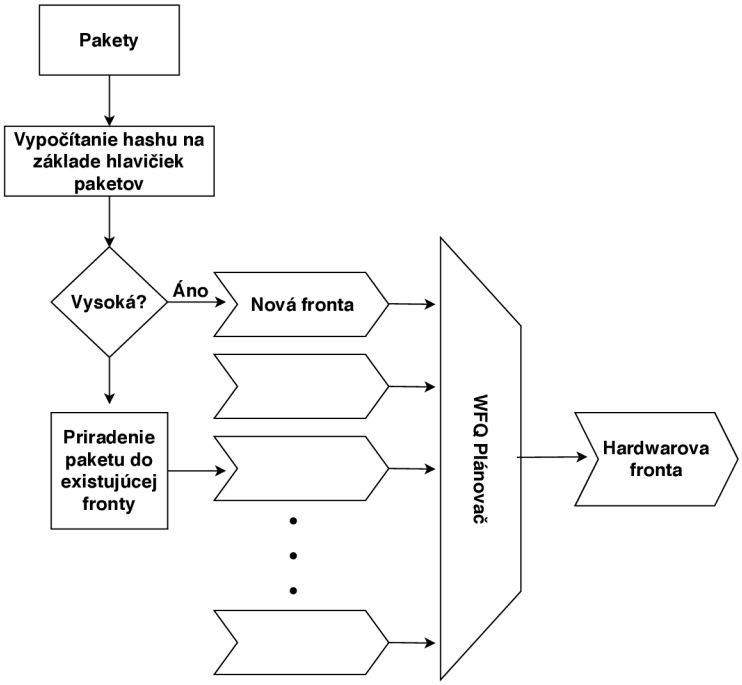
Obr. 1.4: Priority queueing.

Weighted fair queueing (WFQ)

Mechanizmus WFQ je navrhnutý tak, aby minimalizoval nutnosť konfigurácie a automaticky sa prispôsoboval meniacim sa podmienkam sieťovej prevádzky. Je dôležité poznamenať, že počet existujúcich front v systéme WFQ je založený na počte aktívnych tokov. Dátove toky sú identifikované hashom vygenerovaným z údajov uvedených v záhlaví IP paketu [6].

WFQ dynamicky vytvára a odstraňuje fronty. Počet front, ktoré systém WFQ môže vytvoriť pre aktívne toky, je obmedzený. Maximálny počet front, nazývaných tiež dynamické fronty WFQ, je štandardne 256 s možnosťou rozšírenia až na 4096 [6]. Keď počet aktívnych tokov prekročí maximálny možný, nové toky sa priradia existujúcim frontám. Obrázok 1.5 znázorňuje WFQ mechanizmus vytvárania front. Aj pri tomto mechanizme sa však potýkame nedostatkami ako napríklad:

- Premávka nemôže byť radená do front, ktoré boli definované užívateľom.
- WFQ nedokáže garantovať žiadne špecifické požiadavky na šírku pásma pre dané dátove toky.



Obr. 1.5: Weighted fair queueing.

Class-based weighted fair queueing (CB WFQ)

Mechanizmus CBWFQ predovšetkým rieši nedostatky mechanizmu WFQ, teda umožňuje vytvárať používateľom definované triedy, z ktorých každá je priradená k vlastnej fronte. Každá fronta dostáva používateľom definovanú (minimálnu) záruku na šírku pásma, ale ak je k dispozícií väčšia šírka pásma, môže ju využiť. Na rozdiel od PQ, žiadna fronta v CBWFQ nie je „hladujúca“, teda obslúžené sú vždy všetky fronty. CBWFQ môže vytvoriť až 64 front, jednu pre každú triedu definovanú užívateľom [6]. Každá fronta využíva FIFO s definovanou zárukou na šírku pásma a maximálnym limitom paketov.

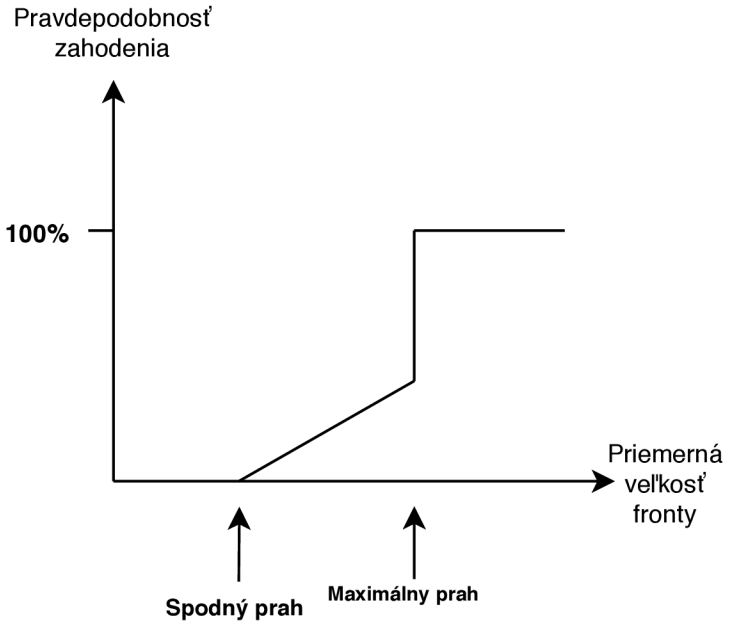
1.2.6 Mechanizmy zabraňujúce preťaženiu

Random early detection (RED)

RED je mechanizmus zabraňujúci preťaženiu fronty. RED zahadzuje náhodne vybrané pakety pred tým, ako sa fronta zaplní. Rýchlosť zahadzovania sa zvyšuje s veľkosťou fronty, teda s rastúcou veľkosťou fronty rastie aj pravdepodobnosť zahodenia prichádzajúcich paketov.

Ako je možné vidieť na obrázku 1.6, RED sa vo všeobecnosti riadi troma základnými konfiguračnými parametrami: spodný prah (minimum treshold), vrchný prah (maximum treshold) a MPD (Mark probability denominator). Ak je veľkosť

fronty menšia ako minimálny prah, nedochádza k žiadnemu zahodeniu paketov. S postupným nárastom dĺžky fronty za minimálnu prahovú hranicu, rastie aj množstvo zahodených paketov. Ak dĺžka fronty presiahne maximálnu prahovú hranicu, všetky prichádzajúce pakety budú zahodené [7] .

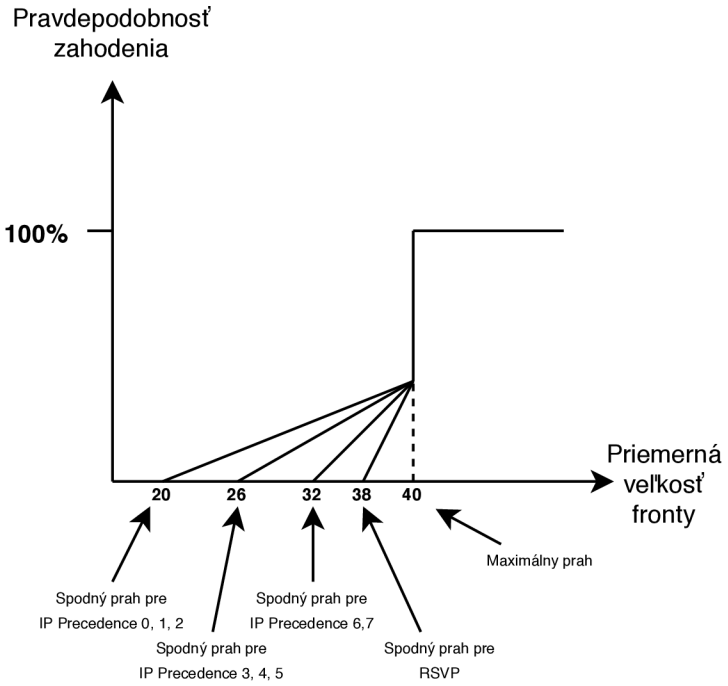


Obr. 1.6: RED.

Weighted Random Early Detection (WRED)

WRED má narozdiel od RED schopnosť rozlišovať medzi prenosom s vysokou a nízkou prioritou. Pre každú prioritu prenosu je možné nakonfigurovať vlastný profil s vlastnými hodnotami parametrov (minimálny prah, maximálny prah a MPD). Priorita prenosu je založená na hodnotách IP Precedence, alebo DSCP.

Obrázok 1.7 znázorňuje príklad, v ktorom je minimálna prahová hodnota pre prevádzku s prioritnými hodnotami IP 0, 1 a 2 nastavená na 20, pre prevádzku s prioritnými hodnotami IP 3, 4 a 5 je nastavená na 26 a minimálny prah pre prevádzku s prioritnými hodnotami IP 6 a 7 je nastavený na 32 [7]. WRED nie je vhodný na použitie napríklad pri hlasových službách, pretože hlasové služby sú extrémne citlivé na zahadzovanie paketov a sú postavené na protokole UDP.



Obr. 1.7: WRED.

1.2.7 Sieťová podpora a protokoly pre QoS

IntServ (Integrated Services)

V modeli IntServ je vždy pred uskutočnením prenosu potrebný signalizačný protokol, ktorý informuje smerovače, že tok paketov vyžaduje špeciálne spracovanie QoS [8].

Model architektúry IntServ bol motivovaný potrebami aplikácií v reálnom čase, ako sú vzdialené video, multimediálne konferencie, vizualizácia a virtuálna realita. Poskytuje spôsob ako poskytovať end-to-end QoS, ktorý aplikácie v reálnom čase vyžadujú, explicitným spravovaním sieťových zdrojov. IntServ využíva protokol RSVP (Resource Reservation Protokol), ktorý explicitne signalizuje potreby QoS prevádzky aplikácie pozdĺž všetkými uzlami v end-to-end ceste cez sieť. Ak všetky uzly pozdĺž cesty môžu vyhradiť potrebnú šírku pásma, aplikácia môže začať vysielať.

DiffServ (Differentiated Services)

DiffServ je sieťová architektúra, ktorá špecifikuje jednoduchý a pružný mechanizmus na klasifikovanie a riadenie sieťového prenosu, zabezpečuje taktiež podporu QoS v moderných IP sieťach. Tento mechanizmu je napríklad schopný zabezpečiť nízke oneskorenie pre služby ako videokonferencie alebo streamované média, zatiaľ čo menej náročným službám z hľadiska oneskorenia, zabezpečuje jednoduchý mechanizmus „best effort“ [4].

Pre účely značenia paketov sa najčastejšie využíva osembitové pole v hlavičke IP paketu, označované ako IP ToS (Internet Protocol Type of Service). Smerovače menia hodnoty v tomto poli podľa potrieb daného prenosu. Pole sa skladá z troch bitov určujúcich prioritu na základe služby, troch bitov určujúcich požiadavky na prenos a dvoch nevyužitých bitov viz Obr. 1.8.

Pri modely DiffServ je štruktúra poľa ToS pozmenená a nahradená poľom DS field (Diffrentiated Services field), ktoré ma taktiež 8-bitov. Pri tomto štandarde je značkovaný každý paket samostatne, kedy do poľa DS je vkladaná 6-bitová hodnota DSCP (Differentiated Services Code Point). Štruktúra poľa DS je zobrazená na obrázku 1.9. Skupina smerovačov, ktorá implementuje spoločné, administrátorom definované DiffServ pravidlá, sa nazýva *DiffServ doména* [9].

Obr. 1.8: Štruktúra poľa ToS.

Obr. 1.9: Štruktúra poľa DS.

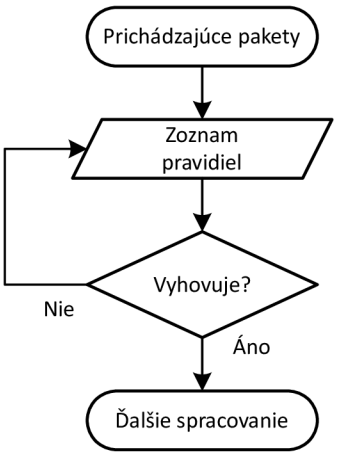
Model DiffServ rozdeľuje prenos na malý počet tried a alokuje zdroje na základe triedy. Pretože služba DiffServ má iba niekoľko tried prevádzky, môže byť priamo v pakte označená „trieda“ prenosu [10].

2 Využitie nástrojov systému MikroTik

2.1 Využitie nástroja Mangle

Mangle je druh „značkovača“, ktorý označuje pakety za účelom budúceho spracovania. Mnoho ďalších nástrojov v službe RouterOS využíva tieto značky, napr.: fronty, NAT, smerovanie. Služby na základe tejto značky spracujú paket podľa užívateľových požiadaviek. Značky, ktoré paketom priradí nástroj Mangle, sú využívané len vrámci daného smerovača, nie sú prenášané sieťou [11].

Nástroj Mangle značí každý prichádzajúci paket, čo môže byť pomerne náročné na hardware daného MikroTik zariadenia, obzvlášť ak sa jedná o pravidlo, ktoré musí spĺňať mnoho parametrov z IP hlavičky alebo zoznamu adries. Prichádzajúci paket je zadržiavaný vo vyrovnávacej pamäti a porovnávaný s pravidlami zo zoznamu vytvoreného administrátorom, systém porovnáva pakety postupne s každým pravidlom až kým nenarazí na pravidlo, ktoré spĺňa jeho požiadavky. Základný princíp fungovania nástroja Mangle môžeme vidieť na obrázku Obr. 2.1. Ako už bolo povedané, zložité značkovanie si vyžaduje zvýšené požiadavky na hardware, ktoré často nie sú dostupné. Cieľom mojej práce je práve optimalizácia zoznamu pravidiel tak, aby bol hardware zariadenia zaťažovaný čo najmenej a tým bola zvýšená aj prenosová rýchlosť a efektivita spracovania paketov. Tohoto výsledku sa pokúsim dosiahnuť navrhnutím vhodného riešenia, ako meniť poradie pravidiel tak, aby práve to najpoužívanejšie pravidlo (to cez ktoré „pretečie“ najväčší objem dát) bolo na prvom mieste v zozname pravidiel Mangle. Podľa očakávaní by sa väčšia časť premávky mala dostať cez značkovací systém hneď pri prvom pravidle, čo by malo zabrániť viacnásobnému kontrolovaniu v ďalších, nevyhovujúcich pravidlách.



Obr. 2.1: Základný princíp fungovania nástroja Mangle.

2.2 Využitie nástroja Queue

Zariadenia MikroTik využívajú systém front k limitizácií a prioritizácií prenosu [12]:

- obmedzením rýchlosti prenosu na základe IP adries, masky, protokolu, portu atď.
- obmedzením peer-to-peer prenosov
- zvýšením priority jednému paketovému toku nad ostatnými
- limitovaním prenosu v závislosti na čase

Implementácia front v systémoch MikroTik je založená na báze HTB (Hierarchical Token Bucket). HTB dovoľuje vytvárať hierarchické štruktúry front, a definuje vzťahy medzi jednotlivými frontami [12]. MikroTik ponúka dve možnosti ako konfigurovať fronty v RouterOS – queue simple a queue tree.

2.2.1 Queue simple

Navrhnuté na zjednodušenie konfigurácie jednoduchých, každodenných frontových úloh. Simple queues je možné využiť taktiež na vybudovanie pokročilých QoS aplikácií. Majú mnoho užitočných vlastností:

- Radenie peer-to-peer prenosov do front
- Aplikovať pravidlá fronty počas zvolených časových intervalov
- Rôzne druhy priorít
- Využívanie značkovača Mangle

Simple queues striktne dodržujú poradie – každý paket musí prechádzať cez každú frontu, kým nedosiahne frontu, ktorej podmienky vyhovujú parametrom paketu alebo bude prechádzať až do konca zoznamu. Teda v prípade zoznamu o dĺžke 1000 front, paket ktorý by spĺňal podmienky až poslednej fronty, bude musieť prejsť cez 999 front predtým, ako dosiahne cieľ.

2.2.2 Queue Tree

Slúži najmä pre implementáciu pokročilých úloh (ako sú globálne priority politiky, obmedzenia skupín používateľov). Vyžaduje označené toky paketov z nástroja Mangle. Pomocou Queue Tree vytvárame iba jednu smerovú frontu v jednom z HTB [12]. Je to tiež jediný spôsob ako vytvoriť frontu pre samostatné rozhranie.

2.3 Využitie skriptovania

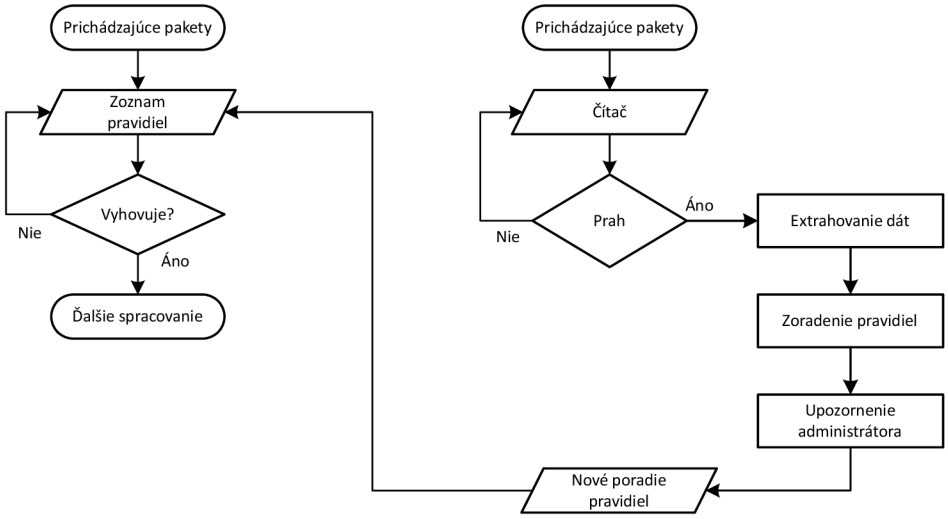
Skripty sú jednou z vlastností operačného systému RouterOS s ktorým sa môžeme stretnúť na zariadeniach od spoločnosti MikroTik. Pomocou skriptov sme schopný dynamicky a ľubovoľne meniť konfiguráciu smerovača. Príkazy sa zadávajú pomocou

terminálu, alebo je ich možné nahrať aj ako súbor s príponou „.rsc“. Skriptovanie na zariadeniach s RouterOS so sebou prináša aj niekoľko obmedzení. Napríklad dáta s ktorými môžeme pracovať musia musia byť uložené na tom danom smerovačy, kde beží skript. Ak chcem skriptu predať ľubovoľné parametre je nutné využiť možnosti globálnej premennej. V RouterOS majú všetky príkazy rovnakú syntax. Všetkým častiam sa predávajú parametre rovnakým spôsobom [13]. Syntax vyzerá následovne:

```
[prefix] [path] command [uparam] [param=[value] .. [param=[value]]
```

3 Implementácia riešení

V tejto kapitole sa budem podrobne venovať návrhu a riešeniu tejto práce. V jednotlivých sekciách si postupne priblížime návrh riešenia, konfiguráciu testovacej siete využitej pri riešení danej problematiky, konfiguráciu samotného aktívneho sieťového prvku, riešenie skriptov. Riešenie bude vychádzať z vývojového diagramu zobrazeného na obrázku 3.1.



Obr. 3.1: Vývojový diagram návrhu riešnia.

V ľavej časti vývojového diagramu je zobrazený základný princíp fungovania nástroja Mangle. Princíp fungovania tohoto nástroja je opísaný v sekcií 2.1.

Na pravej strane vývojového diagramu môžeme vidieť doplnkový algoritmus pre zaistenie plne automatického reorganizovania zoznamu pravidiel Mangle, extrahovania potrebných dát a nakoniec notifikovania administrátora o uskutočnených zmenách. Všetky tieto úkony sú obsiahnuté v jednom skripte a v nasledujúcom texte budú predstavené jednotlivé časti skriptu plniace konkrétnu funkciu.

Po dosiahnutí istého prahu (počtu prenesených bitov v danom pravidle) dôjde k extrahovaniu užitočných dát z nástroja Mangle. Tieto dáta budú uložené na aktívnom prvku v textovom súbore, zoradené pod sebou v takom poradí ako boli v zozname Mangle. Tento súbor bude následne využitý pri notifikovaní administrátora prostredníctvom emailu.

Ďalším krokom v mojom riešení je automatická reorganizácia pravidiel v nástroji Mangle. Vyhotovený skript zoradí zoznam pravidiel na základe ich využitia *zostupne*, teda od najviac využívaného pravidla po najmenej využívané pravidlo. Využiteľnosť

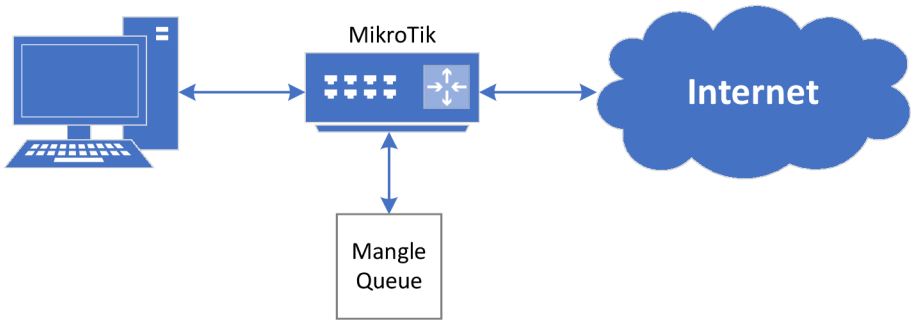
pravidla je možné merať na základe množstva označených paketov daným pravidlom, resp. množstvom bytov prenesených danými paketmi. Samotný skript s bližším popisom môžeme nájsť v kapitole 3.3.

Nasledujúcim a zároveň posledným krokom je notifikovanie administrátora o zmenách vykonaných so zoznamom pravidiel. Notifikovanie bude uskutočnené pomocou emailu, kde prílohou bude súbor s extrahovanými údajmi z prvého kroku môjho optimalizačného algoritmu viz 3.4

Všetky skripty, aj s kontrolou prekročenia maximálneho povoleného prahu prenesených bytov, sú spúšťané každých 12h pomocou nástroja *Scheduler*, ktorého jednoduchú konfiguráciu môžeme nájsť v sekcií 3.1.5. Po vykonaní reorganizácie pravidiel a notifikovaní administrátora budú počítadla v jednotlivých pravidlách resetované na nulové hodnoty. Samozrejme administrátor je schopný nastaviť nástroj Scheduler tak, aby spúšťal skripty aj v iných intervaloch, závisí to však na potrebách danej siete. Ak je potrebné zoznam pravidiel meniť častejšie z dôvodu veľkej rozmanitosti premávky, administrátor jednoducho nastaví kratší časový interval spúšťania.

3.1 Príprava testovacieho prostredia

Aby bolo možné celú myšlienku realizovať, bolo potrebné vytvoriť jednoduchú testovaciu sieť viz 3.2. Účelom vytvorenej testovacej siete je implementácia navrhovaných riešení na nástrojoch systému MikroTik a otestovanie v reálnej komunikácií. V tejto sekcií si opíšeme základné kroky, ktoré viedli ku konfigurácií tejto siete a spomínaných nástrojov.



Obr. 3.2: Topológia testovacej siete

3.1.1 Konfigurácia testovacej siete

Nato aby sme vedeli otestovať naše postupy na reálnom dátovom prenose musíme splniť predpoklad správne nakonfigurovanej siete. Bez splnenia tohto predpokladu

nebude naša stanica schopná komunikovať s prostredím internetu a teda ani prijímať žiadne dátové toky. Testovaciu sieť budeme vytvárať pomocou zariadenia *MikroTik hAP Lite*. So smerovačom je možné komunikovať prostredníctvom webového rozhrania, alebo pomocou aplikácie s názvom *WinBox* priamo od spoločnosti MikroTik. V mojom prípade si ukážeme konfiguráciu prostredníctvom aplikácie. Testovací smerovač pripojíme do siete internet pomocou ethernetového káblu s koncovkou RJ-45, zapojeného do portu s nadpisom „WAN“. Router pripojíme k PC taktiež pomocou ethernetového káblu zapojeného do portu s nadpisom napríklad „ether2“. V aplikácií vyberieme zo zariadení to naše a zadáme prihlasovacie údaje zadané výrobcom. K správnej funkcionalite siete je potrebné nakonfigurovať DNS, DHCP, NAT a firewall. Sieť WiFi vytvoríme ako bridge wireles rozhrania a ostatných portov. Po správnom nakonfigurovaní testovacej siete môžu stanice (PC) komunikovať s prostredím internetu.

3.1.2 Konfigurácia nástroja Mangle

Cieľom našej práce je ukázať vplyv usporiadania pravidiel Mangle na vyťaženie hardwaru routra a rýchlosť prenosu. Preto si niekoľko takýchto pravidiel nakonfigurujeme. Z vlastností Router OS vyplýva, že nástroj mangle môžeme konfigurovať napríklad pomocou grafického rozhrania alebo pomocou navrhnutých skriptov. Pre naše potreby bude potrebné nakonfigurovať také množstvo pravidiel, na ktorom bude ľahké demonštrovať prípadne spomalenie, poprípade zrýchlenie prenosu.

Vytvorenie pravidla môže vyzerať napríklad takto:

```
/ip firewall mangle
add action=mark-connection chain=forward comment=client-download-con
in-interface=ether1 new-connection-mark=clients passthrough=yes
```

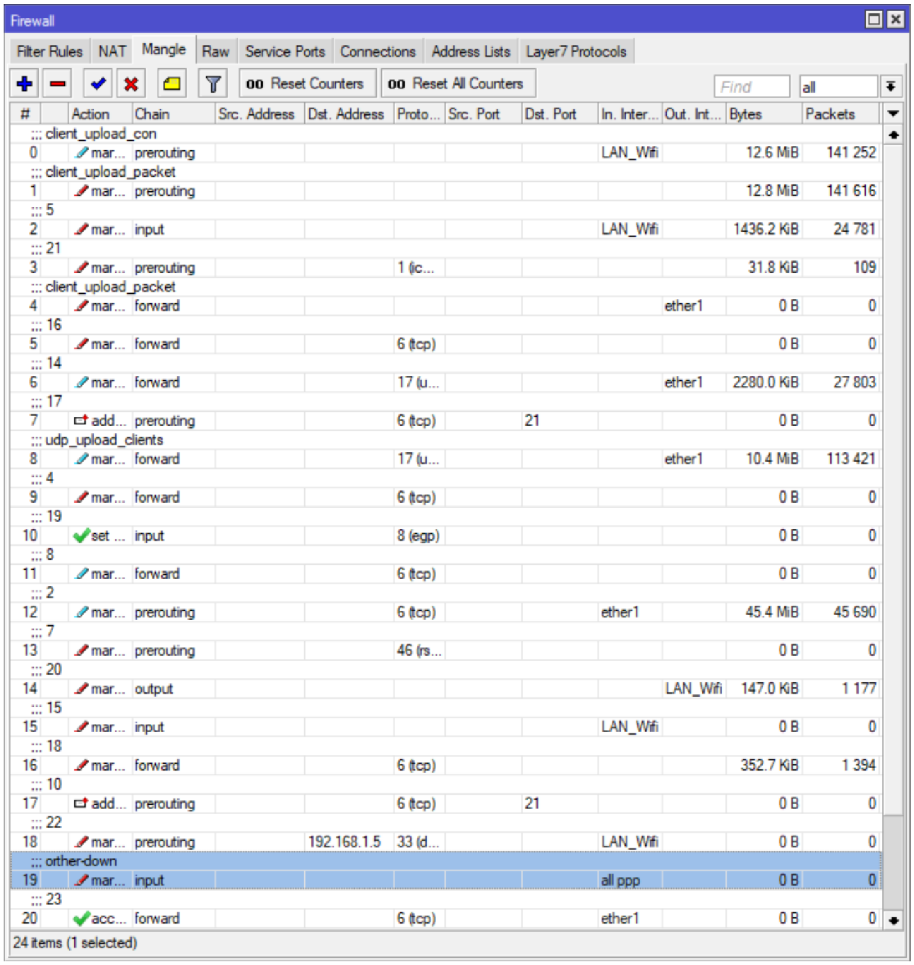
Z daného príkladu si môžeme všimnúť, že toto pravidlo je určené pre všetky pakety, ktoré vstupujú do našej siete pretože sú viazané na port `ether1`, ktorý je používaný ako WAN port. Pakety budú označené značkou `clients`. V tejto ukážke bol pridaný aj komentár `client-download-con`, ktorý nie je povinnou súčasťou pri vytváraní pravidiel, ale zvyšuje prehľadnosť a zjednodušuje orientáciu v následnom zozname pravidiel. Obdobným postup boli vytvorené ďalšie potrebné pravidlá.

Je však nutné poznamenať, že pre účely mojej práce (teda potreby simulovať väčšiu sieť kde sa môže vyskytovať aj niekoľko desiatok pravidiel) boli vytvorené aj pravidlá, ktoré pri type prevádzky v testovacej sieti nie sú využívané resp. testovaným smerovačom na ktorom je nakonfigurovaný značkovač Mangle, neprechádza potrebný typ dátových jednotiek, ktorý by splnil špecifikácie označenia daných pravidiel. Pre skutočné potreby mojej testovacej siete plne postačuje zopár pravidiel.

Ostatné pravidlá, kde si môžete všimnúť nulovú využiteľnosť, tvoria tzv. umelú záťaž práve z dôvodov, ktoré boli uvedené vyššie. Nižšie môžeme vidieť príklad zdrojového kódu niektorého z pravidiel využitého v testovacej sieti.

```
/ip firewall mangle
add action=mark-packet chain=prerouting comment=client-up-packet
    connection-mark=CLient_upload new-packet-mark=client-up-packet
    passthrough=yes
```

Na uvedenom príklade vidíme, že ako značku pre spojenie sme tentokrát použili `CLient_upload` a každému paketu bude pridaná značka `client-up-packet`. Pre vytvorenie celého zoznamu pravidiel pomocou skriptu, je nutné aby tieto jednotlivé pravidlá boli uložené v jednom súbore a v poradí akom ich chceme mať uložené v samotnom zozname pravidiel Mangle. Takto vyzerajúci skript vytvorí zoznam pravidiel, ktorý si môžeme pozrieť v záložke nástroja Mangle v grafickom rozhraní RouterOS viz obr. 3.3.



Obr. 3.3: Záložka Mangle

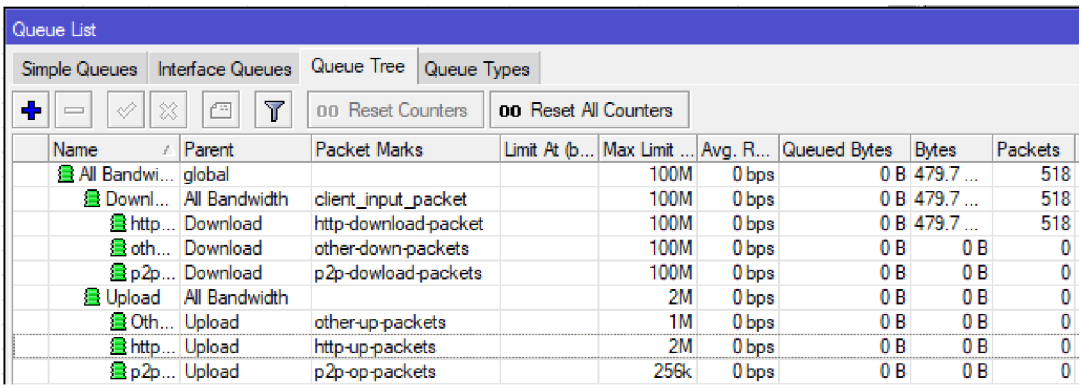
3.1.3 Konfigurácia nástroja Queues

Nástroj Queues nám umožňuje pohodlne vytvoriť jednoduché alebo aj zložité stromové štruktúry front, kde následne môžeme aplikovať značenie paketov vytvorených v nástroji Mangle. K vytváraniu stromových štruktúr je opäť možné použiť grafické rozhranie, alebo konzolu do ktorej zadaním série príkazov vytvoríme žiadané fronty. Tento nástroj umožňuje hierarchické usporiadanie front, kde existujú nadriadené fronty (parent) a nižšie postavené resp. podriadené fronty (children).

Pre každú frontu je možné nastaviť hneď niekoľko možných parametrov ako napriklad: názov fronty, maximálnu rýchlosť prenosu dátových jednotiek, prioritu spracovania dátových jednotiek nachádzajúcich sa v danej fronte. Pre konfiguráciu nástroja Queue je tiež nutné uviesť, ktoré označené pakety prislúchajú danej fronte. Výslednú úspešnú konfiguráciu nástroja Queue na testovacom zariadení môžete vidieť na obrázku 3.4. V texte nižšie môžete tiež vidieť príklad zdrojového kódu pre vytvorenie kompletnej štruktúry front:

```
/queue tree
add name=AllBandwidth parent=global

add max-limit=100M name=Download packet-mark=client_input_packet
parent=AllBandwidth priority=1

add max-limit=100M name=http-download packet-mark=http-download-packet
parent=Download priority=1 queue=pcq-download-default

add max-limit=100M name=other-download packet-mark=other-down-packets
parent=Download priority=1 queue=pcq-download-default

add max-limit=100M name=p2p-download packet-mark=p2p-dowload-packets
parent=Download priority=2 queue=pcq-download-default

add max-limit=10M name=Upload parent=AllBandwidth

add max-limit=10M name=http-upload packet-mark=http-up-packets
parent=Upload priority=1 queue=pcq-upload-default

add max-limit=256k name=p2p-upload packet-mark=p2p-op-packets
parent=Upload queue=pcq-upload-default
```

Z príkladu zdrojového kódu si môžeme všimnúť že v našej štruktúre sa nachádza fronta `AllBandwidth`, ktorá je rodičovskou frontou pre jej dve dcérske fronty `Download` a `Upload`. Tieto dve dcérske fronty obsahujú ďalšie dcérske fronty, čím sme vytvorili žiadanú hierarchickú stromovú štruktúru. Parametrom `packet-mark` priradíme k fronte pakety s danou značkou. Ďalšími dôležitými parametrami sú napríklad `priority` a `max-limit`, pomocou ktorých určíme akú prioritu majú mať dátové jednotky spadajúce do konkrétnej fronty resp. maximálnu rýchlosť odosielania daných jednotiek.



Name	Parent	Packet Marks	Limit At (b...	Max Limit ...	Avg. R...	Queued Bytes	Bytes	Packets
All Bandwi...	global			100M	0 bps	0 B	479.7 ...	518
Downl...	All Bandwidth	client_input_packet		100M	0 bps	0 B	479.7 ...	518
http...	Download	http-download-packet		100M	0 bps	0 B	479.7 ...	518
oth...	Download	other-down-packets		100M	0 bps	0 B	0 B	0
p2p...	Download	p2p-dowload-packets		100M	0 bps	0 B	0 B	0
Upload	All Bandwidth			2M	0 bps	0 B	0 B	0
Oth...	Upload	other-up-packets		1M	0 bps	0 B	0 B	0
http...	Upload	http-up-packets		2M	0 bps	0 B	0 B	0
p2p...	Upload	p2p-op-packets		256k	0 bps	0 B	0 B	0

Obr. 3.4: Záložka Queue Tree

3.1.4 Konfigurácia nástroja Email

Email je nástroj, ktorý umožňuje odosielanie emailov zo smerovača od spoločnosti MikroTik. Tento nástroj sa najčastejšie využíva k odosielaniu konfiguračných záloh a exportov administrátorovi siete.

Tento nástroj môžeme pohodlne konfigurovať pohodlne pomocou grafického rozhrania v záložke „tools“ a následne „Email“. Ďalšou možnosťou konfigurácie je využitie terminálu a zadanie konfiguračných príkazov. Príklad na konfiguráciu nástroja Email pomocou terminálu:

```
/tool e-mail set address=74.125.141.108
from=mikrotik.router123@gmail.com
password=<heslo>  port=587 start-tls=yes
user=mikrotik.router123@gmail.com
```

3.1.5 Konfigurácia nástroja Scheduler

Pomocou nástroja scheduler môžeme spúšťať rôzne skripty v presne daný moment, v špecifikovanom časovom intervale alebo kombináciou týchto možností. Scheduler je možne nakonfigurovať pomocou jednoduchého grafického rozhrania alebo pomocou zadania konfiguračných príkazov do terminálu. Príklad zdrojového kódu využitého v mojej práci môžete vidieť v nasledujúcom texte:

```
/system scheduler
add interval=12h name=Notification on-event=HOTOVYCELKOVYSKRIPT
policy=ftp,reboot,read,write,policy,test,password,sniff,sensitive,
romon start-date=apr/14/2019 start-time=15:10:24
```

V zdrojovom kóde si môžeme všimnúť hneď niekoľko parametrov. Parameter `interval` určuje časový interval, po ktorom sa bude spúšťať skript, ktorého názov je uvedený v parametri `on-event`.

3.2 Extrahovanie potrebných dát

Po konfigurácií testovacej siete sa dostávame priamo k úkonom vedúcim k optimalizovaniu zoznamu pravidiel, ktoré sú vyobrazené už v spomínanom vývojom diagrame môjho riešenia 3.1.

V tejto sekcií je popísané ako boli získané pre nás relevantné dáta, nevyhnutné k našej práci a dôležité pre ďalšie spracovanie v podobe uloženia do súboru, ktorým následne notifikujeme administrátora. V nástroji Mangle je implementované „počítadlo“ (counter), ktoré nám pre tieto potreby úplne postačí. Pri každom z pravidiel je vedená štatistika o množstve prenesených dát a paketov. Pomocou skriptu nižšie sme schopný tieto dáta získať a zhromaždiť.

```
:global fileContent "";
:global numRules ;
:global numRules [/ip firewall mangle print count-only];

/file remove [find name~"pravidlaMangle"];
/file print file="pravidlaMangle";

/delay delay-time=1s;
/file set [find name~"pravidlaMangle"] contents="";
/delay delay-time=1s;
```

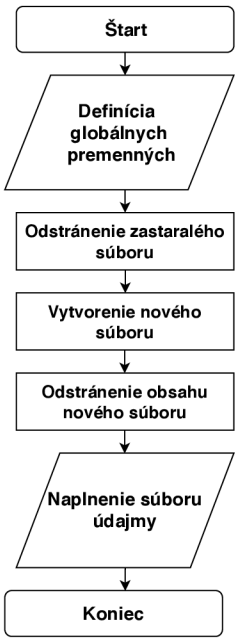
```
for i from=1 to=\$numRules do={
:set fileContent [/ip firewall mangle get (\$i-1) bytes];
/file set "pravidlaMangle.txt" contents=([get "pravidlaMangle.txt„
  contents].\$fileContent."   "."\n");
/delay delay-time=1s;
}
```

Logický postup úkonov vykonaných týmto skriptom môžete vidieť vo vývojovom diagrame na obrázku 3.5.

Na začiatku skriptu sú definované premenné `fileContent` (určuje obsah vytvorených testových súborov) a `numRules` (pomocná premenná na určenie počtu pravidiel).

Ako si môžete všimnúť za príkazom pre vytvorenie nového súboru , je ešte príkaz `delay dealy-time=1s`, ktorý zabezpečí pozastavenie kompilovania zdrojového kódu po dobu jednej sekundy. Tento príkaz v skripte uvádzam zo skúseností stým, že výpočtový výkon zariadenia niekedy nebol dostačujúci a zariadenie nestačilo spraviť všetky potrebné úkony vedúce k vytvoreniu nového súboru, zatiaľ čo skript pokračoval ďalej a dáta nemali byť kde zapísané keďže súbor ešte nebol vytvorený.

V cykle „for“ prebieha získavanie dát z nástroja mangle a naplnenie pripraveného textového súboru. Jednotlivé údaje sú ukladané pod seba. Tento úkon je posledným krokom skriptu pre extrakciu dát viz 3.5.



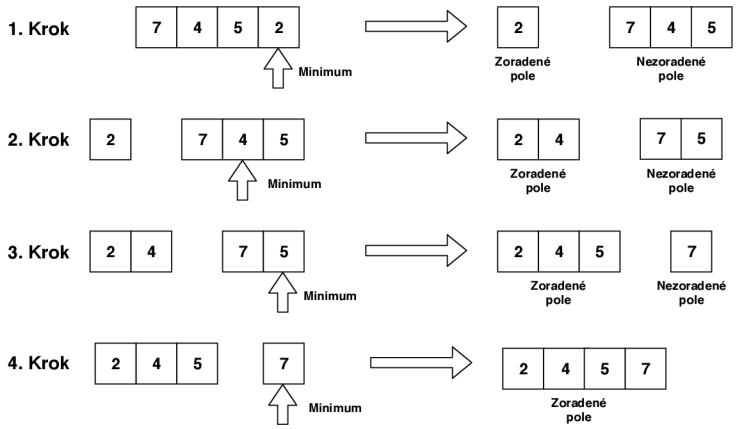
Obr. 3.5: Vývojový diagram skriptu pre extrahovanie dát.

3.3 Usporiadanie pravidiel v nástroji Mangle

V tejto sekcií sa dostávame k druhému bodu môjho riešenia a to k autonómnemu preusporiadaniu pravidiel v nástroji Mangle. Preusporiadanie pravidiel je realizované pomocou skriptu, ktorého zdrojový kód môžete vidieť v následujúcom texte:

```
:global numRules;
:global numRules [/ip firewall mangle print count-only];
:global min;
:set min $numRules;

for i from=0 to=$numRules do={
        /ip firewall mangle print stats;
        :set min $numRules;
        global j 0;
        :set j ($i);
        for x from=$j to=$numRules do={
                if ([/ip firewall mangle get ($min) bytes]<
                [/ip firewall mangle get ($x) bytes])
                        do={
                        :set min $x;
                        }
                ;delay delay-time=1s;
                }
        /ip firewall mangle move ($min) ($i);
}
```

K usporiadaniu pravidiel od najviac využívaného po najmenej využívane bol použitý mechanizmus *Selection-sort*, ktorý nevyniká efektivitou a rýchlosťou medzi triediacimi algoritmami, ale z dôvodu obmedzení spôsobenými skriptovaním na zariadeniach MikroTik, sa perfektne hodí pre účely v mojej práci. Základný princíp tohto mechanizmu môžete vidieť na obrázku 3.6. Algoritmus nájde v množine hodnôt tú najvyššiu a presunie ju na správnu pozíciu. V ďalších krokoch algoritmus prehľadáva len pole hodnôt, ktoré je neusporiadané.



Obr. 3.6: Princíp fungovania Selection-sort mechanizmu.

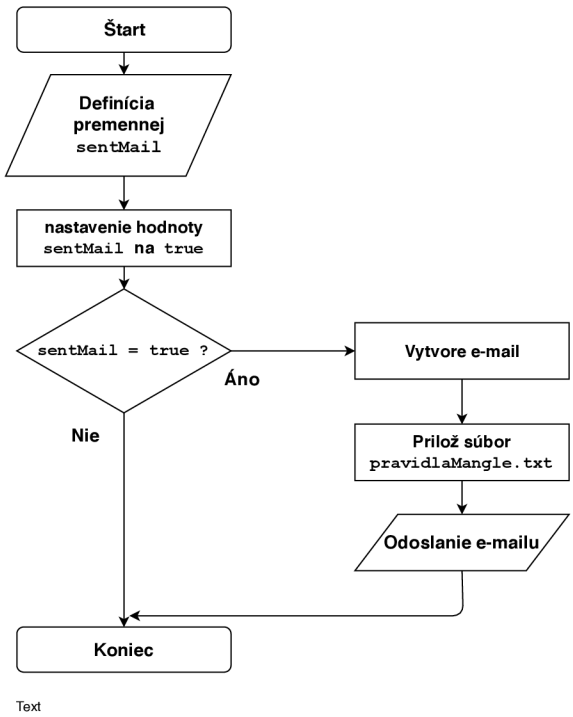
3.4 Notifikovanie administrátora

Posledným krokom môjho optimalizačného algoritmu je notifikovanie administrátora. Notifikovanie je realizované pomocou emailu odoslaného na mailovú adresu administrátora. Súčasťou notifikačného emailu je priložený textový súbor obsahujúci informácie o vyťaženosti pravidiel, ktorý bol vytvorený pomocou skriptu z prvého kroku optimalizačného algoritmu opísaného v sekcií 3.2.

Na to aby bolo možné odosielať emaily zo zariadenia je nutná správna konfigurácia nástroja *Email*, ktorá je uvedená v 3.1.4.

```
if ($sentMail=true) do={
        [/tool e-mail send to="<mailova adresa>"
        subject="<subject>"
        file="pravidlaMangle.txt" body="<body>"];
        }
set sentMail false;
}
```

V texte vyššie je uvedený skript pre odosielanie notifikačného emailu. V parametry `body` môže byť doplnený voliteľný text emailu, takisto parameter `subject` môže byť doplnený o voliteľný text. Na obrázku 3.7 je vyobrazený vývojový diagram skriptu pre odosielanie notifikačných emailov.



Obr. 3.7: Vývojový diagram skriptu pre odosielanie emailov.

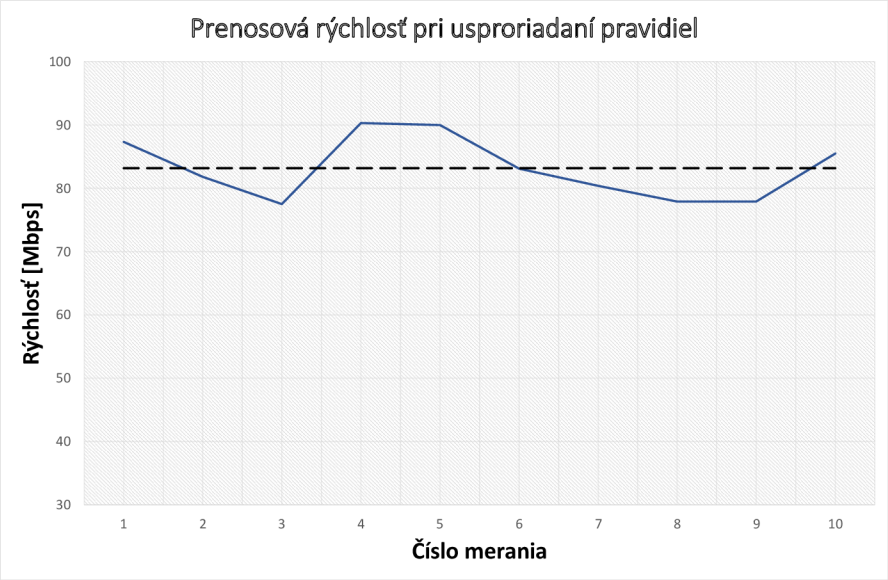
3.5 Dosiahnuté výsledky

V tejto sekcií sú prezentované dosiahnuté výsledky. Sledovanými parametrami boli prenosová rýchlosť a vyťaženosť CPU smerovača.

3.5.1 Meranie s využitím FTP serveru

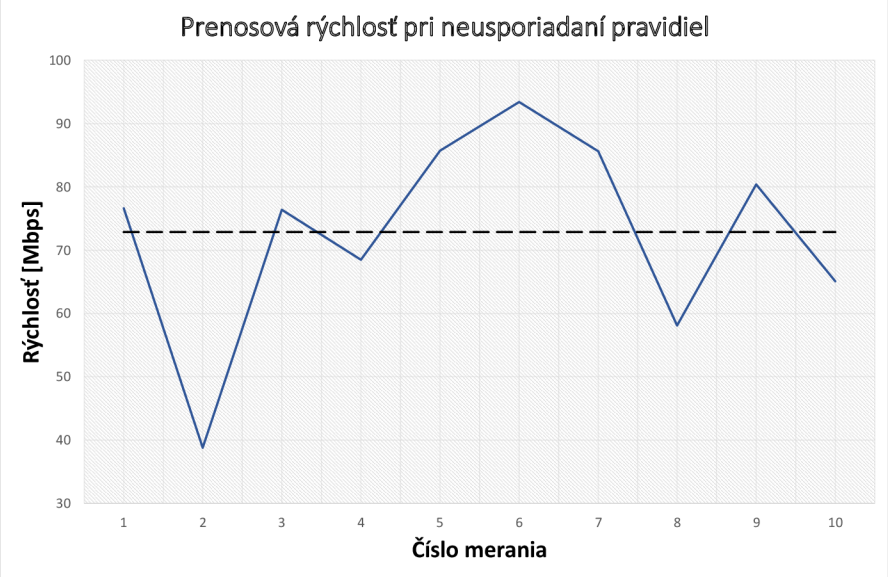
Výsledkom zmeny poradia Mangle pravidiel od najviac využívaných po najmenej využívané by malo byť zníženie hardwarového zaťaženia smerovača a zvýšenie prenosovej rýchlosti. Testy prenosovej rýchlosti v tejto sekcií boli uskutočnené pomocou servera *testmy.net*. Rýchlosť je testovaná tak, že sú sťahované testovacie dáta o veľkosti 100 MB a na základe času za aký sa celý objem dát preniesol, je vyhodnotená priemerná prenosová rýchlosť.

Merania boli uskutočnené desaťkrát. Ako prvé bolo meranie pri ideálnom usporiadaní pravidiel, teda pravidlá sú zoradené v poradí od najviac využívaných po tie najmenej využívané. Najviac využívaným pravidlom sa myslí to, cez ktoré „pretečie“ najväčší objem dát. Výsledok tohto merania je znázornený na obrázku 3.8 kde sú znázornené dosiahnuté prenosové rýchlosti a taktiež aj priemerná rýchlosť (prerušovaná čierna čiara). Zmena pravidiel bola uskutočnená pomocou skriptu popísaného v sekcií 3.3.



Obr. 3.8: Prenosová rýchlosť pri usporiadaní pravidiel

Na obrázku 3.9 je znázornená prenosová rýchlosť pri pravidlách usporiadaných tak, že najmenej používané pravidlá sú v zozname na prvých miestach a tie najpoužívanejšie sú v zozname Mangle posledné.



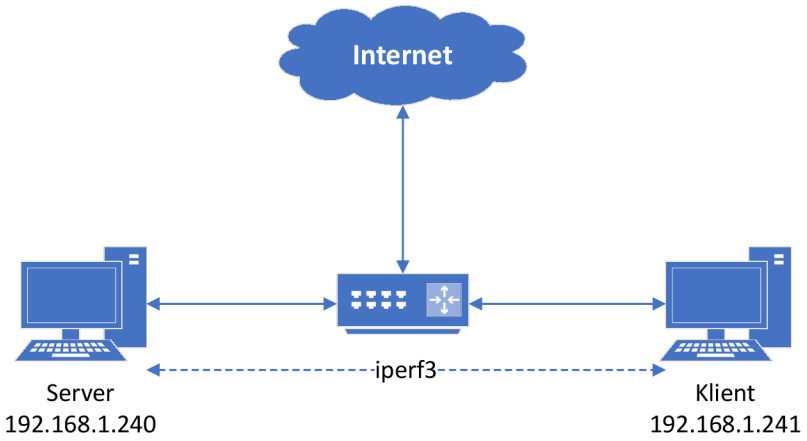
Obr. 3.9: Prenosová rýchlosť pri neusporiadaní pravidiel

Pri usporiadaní pravidiel bola dosiahnutá priemerná rýchlosť 83, 4 Mbps a pri neusporiadaní pravidiel bola dosiahnutá priemerná rýchlosť 72, 86 Mbps, čo značí nárast prenosovej rýchlosti o 13%. Štatisticky teda nedochádza k ohromnému nárastu prenosových rýchlosti na optimalizovaných prvkoch, ale ak sa daná optimalizácia uplatní na viacerých prvkoch v sieti za sebou, nárast rýchlostí už nemusí byť zanedbateľný. Toto meranie by teda malo verne simulovať situáciu, kedy užívateľ sťahuje potrebné dáta napríklad z FTP serverov umiestených v inej sieti. Teda dátové jednotky „cestujú“ cez viacero preskokov a sietí, tak ako to býva pri sťahovaní dát aj v reálnej prevádzke.

3.5.2 Testovanie s využitím nástroja iperf3

Nástroj *iperf3* je jednoducho konfigurovateľným a silným nástrojom pri testovaní možností sietí LAN. Umožňuje testovanie maximálnej dosiahnuteľnej šírky pásma. Umožňuje taktiež nastavovanie rôznych parametrov ako časovanie a použité protokoly (TCP, UDP, SCTP). Pomocou týchto testov bolo mojím cieľom poukázať na zlepšenie prenosových rýchlosti priamo na optimalizovanom zariadení, a vyvarovať sa tak možným vplyvom okolitých resp. vonkajších sietí, ktoré sa mohli vyskytnúť pri meraní v sekcií 3.5.1.

Pre účely testovania maximálnych prenosových možností mojej siete sú zapotreby dve stanice na ktorých budú nakonfigurované server a klient. Topológiu testovacej siete vytvorenej za účelom meraní nástrojom iperf3 môžete vidieť na obrázku 3.10.



Obr. 3.10: Topológia testovacej siete pri meraniach nástrojom iperf3

Po stiahnutí a extrahovaní súborov z oficiálnych stránok nástroja *iperf3* si otvoríme v danom adresárovom okne program PowerShell. Pre konfiguráciu serveru zadáme do programu PowerShell príkaz:

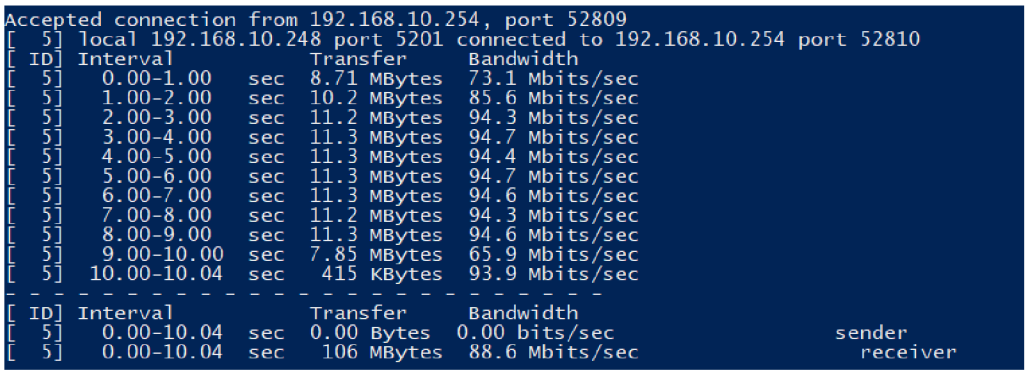
```
./iperf3 -s
```

Klienta na inej stanici nakonfigurujeme obdobný spôsobom, zadaním príkazu:

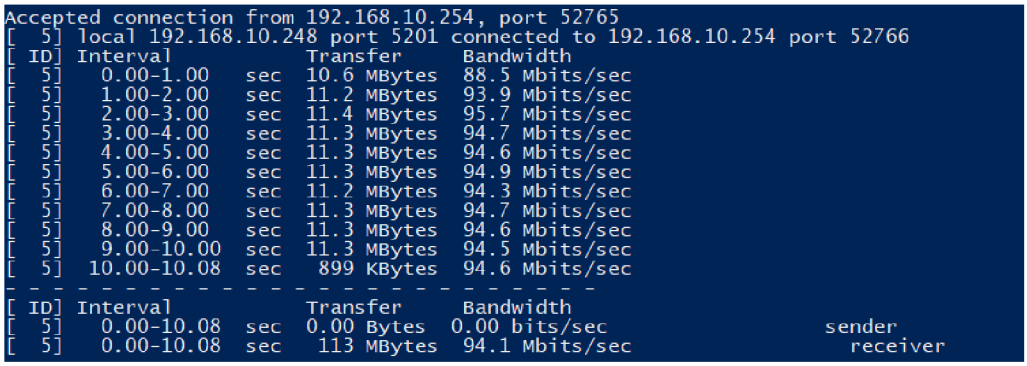
```
./iperf3 -c 192.168.1.240
```

kde daná IP adresa je adresou stanice na ktorej je spustený server. Za predpokladu, že je sieť nakonfigurovaná správne, tak prebehne test maximálnej dostupnej šírky pásma.

Na obrázku 3.11 môžete vidieť priebeh merania maximálnej šírky pásma pri neusporiadaných pravidlách, kde priemerná prenosová rýchlosť bola 88, 6 Mbps, po usporiadaní zoznamu pravidiel bola maximálna prenosová rýchlosť 94, 1 Mbps viz Obr. 3.12, čo predstavuje zlepšenie o 6%.



Obr. 3.11: Prenosové rýchlosti z nástroja iperf3 pri neusporiadaní pravidiel



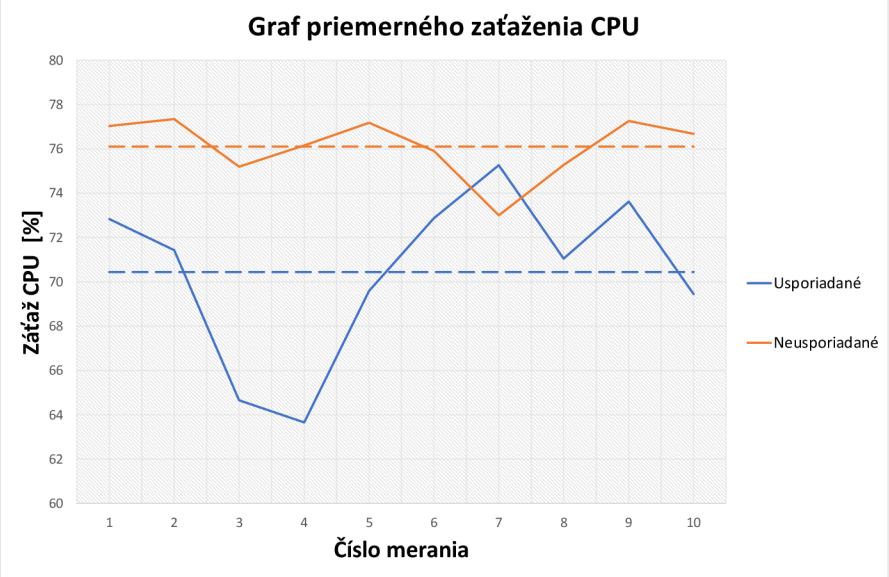
Obr. 3.12: Prenosové rýchlosti z nástroja iperf3 pri usporiadaní pravidiel

3.5.3 Testovanie vyťaženosti CPU

K testovaniu vplyvu usporiadania pravidiel Mangle na vyťaženosť procesorovej jednotky smerovača bol využitý jednoduchý skript, ktorý zabezpečoval výpis zaťaženosti procesora do konzoly, z kadiaľ boli následne údaje spracované do grafovej podoby viz Obr. 3.13. Zdrojový kód skriptu pre získavanie hodnôt o zaťažení cpu:

```
for i from=0 to=100 do={
        :put [/system resource get cpu-load];
        delay delay-time=0.2;
}
```

Ako vyplýva zo zdrojového kódu, skript zabezpečuje výpis procesorového zaťaženia v intervaloch $0,2\,\mathrm{s}$ po dobu 20 sekúnd (sto opakovaní cyklu).



Obr. 3.13: Grafy priemerného zaťaženia CPU.

Pre oba prípady (usporiadané/neusporiadané) bolo uskutočnených desať meraní, ktoré prebiehali za totožných podmienok teda počas sťahovania rovnakého množstva dát, z totožného FTP servera. Zo súboru hodnôt vytvoreného počas každého merania bola vytvorená priemerná hodnota zaťaženia CPU, ktorá bola následne vynesená do grafov. Z grafov je evidentné, že v prípade meraní s optimalizovaným zoznamom pravidiel je CPU zariadenia zaťažované výrazne menej ako pri meraniach s neoptimalizovaným zoznamom.

4 Záver

Cieľom tejto bakalárskej práce bolo pomocou vhodných úprav pravidiel v nástroji Mangle a Queue zvýšiť efektivitu spracovania paketov. V prvej kapitole boli opísané teoretické základy nutné k uvedeniu do danej problematiky. Opísané hlavné parametre QoS, mechanizmy front, ktoré umožňujú QoS implementovať.

V nasledujúcej kapitole je opísané využitie a základné vlastnosti nástrojov systému MikroTik ako napríklad nástroj Mangle, Queue. V tejto kapitole je taktiež opísané využitie skriptov v RouterOS a príklad využitia a tvorby takýchto skriptov.

Práca sa ďalej venuje implementácií riešení použitých v semestrálnej práci. Je tu predstavená základná konfigurácia testovacej siete. Základné konfigurácie nástrojov Mangle, Queue, Mail a Scheduler. Časť tejto kapitoly sa venuje skriptom pomocou, ktorých dochádza k plne automatickému reorganizovaniu pravidiel a notifikovaniu administrátora.

V poslednej kapitole sú zobrazené výsledky meraní prenosových rýchlosti a zaťaženia CPU. Konečný a kompletný skript pre reorganizáciu pravidiel v nástroji Mangle je uvedený v prílohe A.2. Súčasťou príloh sú aj zdrojové kódy skriptov pre konfiguráciu nástrojov Mangle, Queue, Email a Scheduler.

Literatúra

[1] EL SADDIK, Abdulmotaleb. *Quality of Service in Multimedia Networks.*[online]. Boston, MA, 2006, 8-24 [cit. 2018-11-23]. DOI: <10.1007/0-387-30038-4_199.>. Dostupné z: <https://link.springer.com/content/pdf/10.1007%2F0-387-30038-4_199.pdf>.

[2] HUSTON, Geoff. *Next steps for the IP QoS architecture.*[online]. 2000, 10-24 [cit. 2018-11-23]. DOI: 10.17487/RFC2990. Dostupné z: <https://www.rfc-editor.org/rfc/rfc2990.txt>.

[3] JEŘÁBEK, Jan. *Komunikační technológie.* Brno: Vysoké učení technické v Brně, 2013, 174 stran [cit. 2018-11-23]. ISBN 978-80-214-4713-4.

[4] MOLNÁR, Karol. *Hardware počítačových sítí.* Brno: Vysoké učení technické v Brně, 2012 [cit. 2019-05-13]. ISBN 978-80-214-4449-2.

[5] ZHAN, F Benjamin. *Three fastest shortest path algorithms on real road networks: Data structures and procedures.* Journal of geographic information and decision analysis [online]. 1997, 69 – 82 [cit. 2018-12-11]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.7898&rep=rep1&type=pdf>.

[6] BALCHUNAS, Aaron. *QoS and Queueing* [online]. [cit. 2018-12-09]. Dostupné z: <https://www.routeralley.com/guides/qos_queuing.pdf>

[7] CISCO. *Release 12.1 Quality of Service Solutions Configuration Guide – Congestion Avoidance Overview* [online]. [cit. 2019-05-08]. Dostupné z: <https://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfconav.html>

[8] *Integrated Services.* Cisco [online]. [cit. 2018-12-09]. Dostupné z: <https://www.cisco.com/c/en/us/products/ios-nx-os-software/integrated-services/index.html>.

[9] GROSSMAN, Dan. *New terminology and clarifications for diffserv.* [online].In: 2002,s. 1-8 [cit. 2019-05-07]. Dostupné z: <https://tools.ietf.org/html/rfc3260>.

[10] *Differentiated Services.* Cisco [online]. [cit. 2018-12-09]. Dostupné z: <https://www.cisco.com/c/en/us/products/ios-nx-os-software/differentiated-services/index.html>.

[11] *MikroTik Documentation: Manual/Mangle* [online]. [cit. 2018-12-08]. Dostupné z: <`https://wiki.mikrotik.com/wiki/Manual:IP/Firewall/Mangle`>.

[12] *MikroTik Documentation: Manual/Queue* [online]. [cit. 2018-12-08]. Dostupné z: <`https://wiki.mikrotik.com/wiki/Manual:Queue`>.

[13] *MikroTik Documentation: Manual/Scripting* [online]. [cit. 2018-12-08]. Dostupné z: <`https://wiki.mikrotik.com/wiki/Manual:Scripting`>.

Zoznam symbolov, veličín a skratiek

BA	Behaviour Aggregate
CBWFQ	Algoritmus – Class-Based Weighted Fair Queueing
CIR	Commited Information Rate
PIR	Peak Information Rate
CPU	Control Processor Unit
CQ	Algoritmus – Custom Queueing
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DNS	Domain Name Server
DSCP	Differentiated Services Code Point
FIFO	Algoritmus – First In First Out
CPU	File Transport Protocol
HTB	Hierarchical Token Bucket
IntServ	Integrated Services
LAN	Local Area Network
LIFO	Last In First Out
MF	Multi-Field Classification
MPD	Mark Probability Denominator
NAT	Network Address Translator
PQ	Priority Queueing
RED	Random Early Detection
RSVP	Resource Reservation Protocol
RTT	Round Trip Time
SLA	Service Level Agreement
TB	Token Bucket
TCP	Transmission Control Protocol
ToS	Type of Service
UDP	User Datagram Protocol
QoS	Quality of Service
WFQ	Weighted Fair Queueing
WRED	Weighted Random Early Detection

Zoznam príloh

A Zdrojové kódy využívaných skriptov

A.1 Konfiguračné skripty

A.1.1 Konfigurácia nástroja Mangle

```
/ip firewall mangle
add action=mark-connection chain=prerouting comment=2 in-
   interface=ether1
    in-interface-list=all new-connection-mark=clients
       passthrough=yes
    protocol=tcp
add action=mark-connection chain=prerouting comment=6 in-
   interface=LAN_Wifi
    new-connection-mark=CLient_upload passthrough=yes
add action=mark-packet chain=prerouting comment=13 co
   nnection-mark=
    CLient_upload new-packet-mark=client-up-packet
       passthrough=yes
add action=mark-connection chain=forward comment=1 new-co
   nnection-mark=
    udp-upload-clients out-interface=ether1 passthrough=n
       o
add action=mark-packet chain=input comment=5 in-interface
   =LAN_Wifi
    new-packet-mark=http-up-packets passthrough=yes
add action=mark-packet chain=prerouting comment=21 new-
   packet-mark=
    client_input_packet passthrough=yes protocol=icmp
add action=mark-packet chain=forward comment=11 connectio
   n-mark=clients
    new-packet-mark=client-up-packet out-interface=ether1
        passthrough=no
add action=mark-packet chain=forward comment=16 new-
   packet-mark=
    http-download-packet packet-mark=client_input_packet
       passthrough=no port=
    80,443 protocol=tcp
```

```
add action=mark-connection chain=forward comment=14 new-c
   onnection-mark=
    udp-upload-clients out-interface=ether1 passthrough=
       yes protocol=udp
add action=add-dst-to-address-list address-list=ftpinit
   address-list-timeout=\
    none-dynamic chain=prerouting comment=17 dst-address-
       list=!ftpok \
    dst-port=21 protocol=tcp
add action=mark-packet chain=forward comment=4 new-packet
   -mark=\
    other-down-packets packet-mark=other-down-packets
       passthrough=yes \
    protocol=tcp
add action=set-priority chain=input comment=19 new-prio
   rity=8 passthrough=yes \
    protocol=egp
add action=mark-connection chain=forward comment=8 co
   nnection-mark=no-mark \
    in-interface-list=all new-connection-mark=tcp-downlo
       ad-all passthrough=\
    yes protocol=tcp
add action=mark-packet chain=prerouting comment=7 new-
   packet-mark=\
    other-down-packets packet-mark=other-down-packets
       passthrough=no \
    protocol=rsvp
add action=mark-packet chain=input comment=15 in-
   interface=LAN_Wifi \
    new-packet-mark=other-down-packets packet-mark=
       client_input_packet \
    passthrough=yes
add action=mark-packet chain=forward comment=18 new-
   packet-mark=\
    http-up-packets packet-mark=client-up-packet passthro
       ugh=no port=80,443 \
    protocol=tcp
add action=add-dst-to-address-list address-list=atls
   address-list-timeout=\
```

```
    none-dynamic chain=prerouting comment=10 dst-port=21
       layer7-protocol=tls7 \
    protocol=tcp
add action=mark-packet chain=prerouting comment=22 dst-
   address=192.168.1.5 \
    in-interface=LAN_Wifi new-packet-mark=client-up-
       packet passthrough=no \
    protocol=dccp
add action=mark-packet chain=input comment=orther-down in
   -interface=all-ppp \
    new-packet-mark=other-down-packets passthrough=yes
add action=accept chain=forward comment=23 in-interface=
   ether1 port=20 \
    protocol=tcp
add action=mark-connection chain=forward comment=24 new-c
   onnection-mark=\
    CLient_upload out-interface=ether1 passthrough=yes pr
       otocol=tcp src-port=\
    20
add action=mark-connection chain=forward comment=12 in-
   interface=LAN_Wifi \
    log=yes new-connection-mark=tcp-download-all passthro
       ugh=no protocol=udp
add action=mark-packet chain=output comment=20 new-packet
   -mark=\
    client-up-packet out-interface=LAN_Wifi passthrough=
       yes
add action=mark-packet chain=forward comment=3 new-packet
   -mark=\
    other-down-packets packet-mark=client_input_packet
       passthrough=no
```

A.1.2 Konfigurácia nástroja Queue

```
/queue tree
add name=AllBandwidth parent=global
add max-limit=1G name=Download packet-mark=
   client_input_packet parent=\
    AllBandwidth priority=1
add max-limit=1G name=http-download packet-mark=http-do
   wnload-packet parent=\
    Download priority=1 queue=pcq-download-default
add max-limit=1G name=other-download packet-mark=other-do
   wn-packets parent=\
    Download priority=1 queue=pcq-download-default
add max-limit=10M name=Upload parent=AllBandwidth
add max-limit=10M name=http-upload packet-mark=http-up-
   packets parent=Upload \
    priority=1 queue=pcq-upload-default
add max-limit=1G name=p2p-download packet-mark=p2p-dowlo
   ad-packets parent=\
    Download priority=2 queue=pcq-download-default
add max-limit=256k name=p2p-upload packet-mark=p2p-op-
   packets parent=Upload \
    queue=pcq-upload-default
```

A.1.3 Konfigurácia nástroja Mail

```
/tool e-mail
set address=74.125.141.108 from=mikrotik.router123@gmail.
   com password=\
    Mikrotikrouter1 port=587 start-tls=yes user=mikrotik.
       router123@gmail.com
```

A.1.4 Konfigurácia nástroja Scheduler

```
/system scheduler
add interval=12h name=Notification on-event=
   HOTOVYCELKOVYSKRIPT policy=\
    ftp,reboot,read,write,policy,test,password,sniff,
       sensitive,romon \
    start-date=apr/14/2019 start-time=15:10:24
add interval=11h name=fileCreate on-event=hotovyscript po
   licy=\
    ftp,reboot,read,write,policy,test,password,sniff,
       sensitive,romon \
    start-date=apr/14/2019 start-time=15:25:18
```

A.2 Skript autonómneho reorganizovania pravidiel

```
global sentMail;
set sentMail true;
:global fileContent "";
# pocet pravidiel
:global numRules ;
:global numRules [/ip firewall mangle print count-only];
global min;
set min $numRules;

#kontrola tresholdu
for i from=0 to=$numRules do={
  if ([/ip firewall mangle get $i bytes] > 200000000
  && $sentMail=true) do={

#extrahovanie dát a vytvorenie súboru
  /file remove [find name~"pravidlaMangle"];
  /file print file="pravidlaMangle";
  /delay delay-time=1s;
  /file set [find name~"pravidlaMangle"] contents="";
  /delay delay-time=1s;

#Naplnenie súboru údajmi
```

```
  for i from=1 to=$numRules do={
    :set fileContent [/ip firewall mangle get ($i-1)
       bytes];
    /file set "pravidlaMangle.txt" contents=([get "
       pravidlaMangle.txt"        contents].$fileContent.
       "␣␣␣"."\n")
  /delay delay-time=1s;
  }

#Zoradenie pravidiel

  for i from=0 to=$numRules do={
    /ip firewall mangle print stats;
    :set min $numRules;
    global j 0;
    :set j ($i);
    for x from=$j to=$numRules do={
      if ([/ip firewall mangle get ($min) bytes]<
      [/ip firewall mangle get ($x) bytes]) do={
        :set min $x;}
      };
  /ip firewall mangle move ($min) ($i);}

#Notifikačný email

/delay delay-time=1s;
  /tool e-mail send to="xpanca00@vutbr.cz" subject="
     PREKROCENA␣UROVEN␣␣␣PRENOSU"file="pravidlaMangle.txt
     " body="Dobry␣den␣prajem␣pan␣administrator,\n\nbol␣d
     osiahnuty␣treshold␣200MB␣pri␣pravidlach␣MANGLE.␣Zo
     znam␣pravidiel␣bol␣teda␣optimalizovaný␣a␣zoradený␣do
     ␣správneho␣poradia.␣V␣prilohe␣je␣prilozeny␣textovy␣
     subor␣obsahujuci␣vypis␣mnozstva␣prenesenych␣Bytov␣
     cez␣jednotlive␣␣␣pravidla␣Mangle.\n\nPekny␣zvysok␣
     vecera␣prajem,\n\nVas␣Mikrotik␣hap␣Lite" };
  set sentMail false}
set sentMail false;
}
```

B Obsah priloženého CD

- Elektronická verzia bakalárskej práce vo formáte PDF
- Konfiguračný súbor zariadenia MikroTik