UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

# BAKALÁŘSKÁ PRÁCE

2016 Jaroslav Kovář

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

# Praktické úlohy WLAN

Jaroslav Kovář

Bakalářská práce

2016

**Univerzita Pardubice**
**Fakulta elektrotechniky a informatiky**
Akademický rok: **2015/2016**

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Jaroslav Kovář** |
| Osobní číslo: | **I12155** |
| Studijní program: | **B2646 Informační technologie** |
| Studijní obor: | **Informační technologie** |
| Název tématu: | **Praktické úlohy WLAN** |
| Zadávající katedra: | **Katedra informačních technologií** |

Zásady pro vypracování:

Cílem práce je vytvořit praktické výukové úlohy pro oblast datových sítí s platformou Mikrotik. Autor v teoretické části popíše hardwarové prvky, které využije pro realizaci úloh, dále popíše odlišnosti od platformy Cisco. Praktické příklady budou obsahovat zadání, výčet použitých prvků, topologii zapojení a výsledné chování sítě.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

*** CARROLL, Brandon. Bezdrátové sítě Cisco: autorizovaný výukový průvodce. Vyd. 1. Brno: Computer Press, 2011, 478 s. Samostudium. ISBN 978-80-251-2884-8.**

*** STEPHEN R. W. DISCHER. RouterOS by example: understanding MikroTik RouterOS through real life applications. College Station, Texas: MicroTik, 2011. ISBN 978-061-5547-046.**

*** GRESS, Mark L a Lee JOHNSON. Deploying and troubleshooting Cisco wireless LAN controllers. Indianapolis, IN: Cisco Press, c2010, xx, 572 p. CCIE professional development. ISBN 15-870-5814-6.**

Vedoucí bakalářské práce: **Ing. Soňa Neradová, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **31. října 2015**

Termín odevzdání bakalářské práce: **13. května 2016**

Univerzita Pardubice Česká republika

L.S.

Ing. Zdeněk Němec, Ph.D.
děkan

Mgr. Josef Horálek, Ph.D.
vedoucí katedry

V Pardubicích dne 30. dubna 2016

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 2. 9. 2016

Jaroslav Kovář

## PODĚKOVÁNÍ

Rád bych poděkoval Ing. Soně Neradové, Ph.D. za odborný dohled a pomoc při vedení bakalářské práce. Dále bych chtěl poděkovat především své rodině za projevenou podporu během celého studia.

## ANOTACE

Obsahem práce je popis zařízení Mikrotik a operačního systému Mikrotik RouterOS včetně návodů a ukázek řešení jednotlivých problémů jako jsou například: nastavení rozhraní, bridge, DHCP klient a server, DNS, směrování, nastavení bezdrátové sítě, pravidel firewallu, překladu adres a VPN. Dalším bodem je stručné srovnání s produkty jiných výrobců.

## KLÍČOVÁ SLOVA

Mikrotik, sítě, Cisco, směrování, firewall, router

## TITLE

Practical examples of WLAN

## ANNOTATION

This work contains a description of the device developed by Mikrotik and operating system Mikrotik RouterOS including instructions and examples of solutions to various problems such as: setting interface, bridge, DHCP client and server, DNS, routing, wireless network settings, firewall rules, address translation, and VPN. Another point is a brief comparison with other manufacturers' products.

## KEYWORDS

Mikrotik, networks, Cisco, routing, firewall, router

# OBSAH

# SEZNAM ILUSTRACÍ A TABULEK

# SEZNAM ZKRATEK A ZNAČEK

| | |
|---|---|
| ABR | Active Backup Router |
| AP | access point |
| ARP | Address Resolution Protocol |
| BGP | Border Gateway Protocol |
| CAP | Controlled Access Point |
| CAPsMAN | Controlled Access Point system Manager |
| CDMA | Code division multiple access |
| CLI | Command Line Interface |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name System |
| EoIP | Ethernet over IP |
| FQDN | Fully Qualified Domain Name |
| GRE | Generic Routing Encapsulation |
| HP | Hewlett-Packard |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| HW | Hardware |
| ICMP | Internet Control Message Protocol |
| IEEE | Institute of Electrical and Electronics Engineers |
| IP | internet protocol |
| ISO/OSI | International Standards Organization / Open Systen Interconnection |
| ISP | Internet service provider |
| KVM | Kernel-based Virtual Machine |
| L2TP | Layer 2 Tunneling Protocol |
| L7 | Layer 7 |
| LAN | Local Area Network |
| MAC | Media Access Control |
| NAT | Network Address Translation |
| NTP | Network Time Protocol |
| OSPF | Open Shortest Path First |
| OVPN | Open virtual private network |
| PC | personal computer |

| | |
|---|---|
| PPPoE | point-to-point over ethernet |
| PPTP | Point-to-Point Tunneling Protocol |
| RADIUS | Remote Authentication Dial In User Service |
| RIP | Routing Information Protocol |
| SPF | Shortest Path First |
| SSH | Secure Shell |
| SSID | Service Set Identifier |
| STP | Spanning Tree Protocol |
| TCP | Transmission Control Protocol |
| TDMA | Time Division Multiple Access |
| UDP | User Datagram Protocol |
| VLAN | Virtual Local Area Network |
| VLSM | Variable-Length Subnet Mask |
| VPN | virtual private network |
| WAN | Wide Area Network |
| WDS | wireless distribution system |
| WEP | Wired Equivalent Privacy |
| WISP | wireless Internet service provider |
| WISP CPE | wireless Internet service provider Customer-premises equipment |
| WPA | Wi-Fi Protected Access |
| WPA2 | Wi-Fi Protected Access II |

# 0 ÚVOD

Cílem práce je představit zařízení firmy Mikrotik a vytvořit stručný manuál pro základní konfiguraci systému RouterOS. Po nastudování této práce by měl být uživatel schopný samostatně nakonfigurovat Mikrotik RouterOS a zprovoznit i některé pokročilejší funkce jako například VPN nebo dynamické směrování. Dalším bodem je porovnání se zařízeními firmy Cisco, která jsou dnes běžně používaná ve většině velkých firem.

Firma Mikrotik je ve světě méně známou alternativou pro výstavbu síťové infrastruktury. Toto řešení je ideální zejména pro poskytovatele internetového připojení, jelikož umožňuje napojení zařízení skrze API k serveru a následnou centrální správu. Vhodné je také jako řešení hotspotů pro hotely, kavárny a penziony. Dále například jako řešení atypických konfigurací například s použitím skriptů nebo složitějších směrovacích nebo firewallových pravidel. Jelikož se jedná o světově ne tolik rozšířenou platformu, je vyhledání relevantních zdrojů velice obtížné a většina informací o konfiguraci se předává mezi uživateli na internetových fórech. Oficiální dokumentace dostupná na adrese http://wiki.mikrotik.com není tak rozsáhlá, ale je postupně doplňována a rozšiřována

Tato práce se zabývá představením zařízení Mikrotik a základní konfigurací systému RouterOS včetně příkladů pro procvičení konfigurace. Jednotlivé kapitoly jsou obohaceny o praktické ukázky konfigurace a schémat zapojení. Z těchto ukázek pak vycházejí praktické úlohy, které jsou koncipovány velice podobně, nebo se jedná přímo o rozšíření příkladu z nějaké kapitoly. Posloupnost kapitol je koncipována dle náročnosti jednotlivých nastavení a další kapitoly vycházejí z předchozích kapitol, které jsou nezbytné pro jejich pochopení. Obsahem je také stručné porovnání se zařízeními firmy Cisco.

# 1 MIKROTIK

MikroTik je lotyšská společnost, která byla založena v roce 1996 k rozvoji směrovacích a bezdrátových systémů zejména pro ISP. MikroTik nyní poskytuje hardware a software pro připojení k internetu ve většině zemí po celém světě.

Zkušenosti v oboru sítí a kompletních systémů směrování umožnil v roce 1997 vytvoření systému RouterOS, který poskytuje stabilitu, rozsáhlé možnosti nastavení a flexibilitu pro všechny druhy datových sítí. [1]

Mimo operačního systému RouterOS vyvíjí firma i systém pro rozbočovače SwitchOS.

V roce 2002 firma vytvořila vlastní hardware a značku RouterBOARD. Produkty firmy jsou dnes dostupné po celém světě a často využívané jako levnější náhrada produktů Cisco nebo HP. Sídlo společnosti se nachází Rize, hlavním městě Lotyšska. [2]

## 1.1 Hardware

### 1.1.1 RouterBoard

Nabídka zařízení značky RouterBoard je opravdu široká. Nalezneme zde jak klientská zařízení, přístupové body, domácí routery, tak i silné Cloud Core routery schopné zpracovat obrovské datové toky.

Základní desky jsou osazeny procesorem, operační pamětí a dle typu dalšími komponenty, jako jsou FastEthernet porty, GigabitEthernet porty, USB, miniPCI sloty a další.

Některé modely RouterBOARDů mohou ve svém názvu obsahovat následující písmena:

- A - Více paměti,
- H - Vyšší výkon (procesor),
- G - Gigabit ethernet,
- U - USB porty,
- R - Integrovaná bezdrátová karta,
- N - Podpora 802.11n.

Rozlišení pomocí písmen se týká pouze modelů, které mají několik verzí. [3]

### 1.1.2 PC

RouterOS je možno nainstalovat i na běžný počítač. Tuto variantu lze využít například, pokud je potřeba vyšší procesorový výkon pro zpracování pravidel (firewall). Jedna síťová karta se použije jako vstupní, druhá jako výstupní. Při sestavování je třeba používat komponenty, které

jsou kompatibilní s RouterOS, aby později nedocházelo k problémům při instalaci a následném provozu.

## 1.2 RouterOS

Systém Mikrotik RouterOS je specializovaný operační systém pro síťové směrovače. Mimo operačního systému RouterOS firma vyvíjí systém pro rozbočovače SwitchOS, a dále vyvíjí a prodává hardwarové komponenty Mikrotik RouterBOARD, které zahrnují směrovače, rozbočovače a bezdrátové komponenty. Systém Mikrotik RouterOS je výchozím systémem nainstalovaným na všech zařízeních platformy RouterBOARD a je možné ho nainstalovat i na platformu x86 a vytvořit tak směrovač ze serveru či PC. [2]

### 1.2.1 Licence

Pro běh systému jsou potřebné licence. V hardwarových zařízeních RouterBOARD jsou již licence v ceně a uživatel nemusí problematiku licencí řešit. V případě potřeby instalace systému na vlastní HW je nutno licenci zakoupit. RouterOS je možné získat v různých licencích podle požadované úrovně, které jsou odstupňované také cenou. [2]

Licence jsou vždy vázány na konkrétní instalaci, po přeformátování paměti zařízení dochází ke ztrátě licence. Licence nejsou časově omezeny, prvních 15-30 dní obsahují zdarma podporu přes email.

Prvním typem licence je Trial mode. Trial mode se aktivuje po instalaci RouterOS a funguje bez jakýchkoli omezení po dobu 24 hodin nebo do zadání licenčního klíče. Po uplynutí 24 hodin bez zadání klíče se stává zařízení nefunkčním a je nutno ho znovu nainstalovat. Po zadání licenčního klíče se na zařízení aktivuje jedna z následujících licencí:

| **Level number** | **1 (Free Demo)** | **3 (WISP CPE)** | **4 (WISP)** | **5 (WISP)** | **6 (Controller)** |
|---|---|---|---|---|---|
| **Price** | registration required | volume only | $45 | $95 | $250 |
| **Initial Config Support** | - | - | 15 days | 30 days | 30 days |
| **Wireless AP** | - | - | yes | yes | yes |
| **Wireless Client and Bridge** | - | yes | yes | yes | yes |
| **RIP, OSPF, BGP protocols** | - | yes(*) | yes | yes | yes |
| **EoIP** | 1 | unlimited | unlimited | unlimited | unlimited |
| **PPPoE tunnels** | 1 | 200 | 200 | 500 | unlimited |
| **PPTP tunnels** | 1 | 200 | 200 | 500 | unlimited |
| **L2TP tunnels** | 1 | 200 | 200 | 500 | unlimited |

| | | | | | |
|---|---|---|---|---|---|
| **OVPN tunnels** | 1 | 200 | 200 | unlimited | unlimited |
| **VLAN interfaces** | 1 | unlimited | unlimited | unlimited | unlimited |
| **HotSpot active users** | 1 | 1 | 200 | 500 | unlimited |
| **RADIUS client** | - | yes | yes | yes | yes |
| **Queues** | 1 | unlimited | unlimited | unlimited | unlimited |
| **Web proxy** | - | yes | yes | yes | yes |
| **User manager active sessions** | 1 | 10 | 20 | 50 | Unlimited |
| **Number of KVM guests** | 1 | Unlimited | Unlimited | Unlimited | Unlimited |

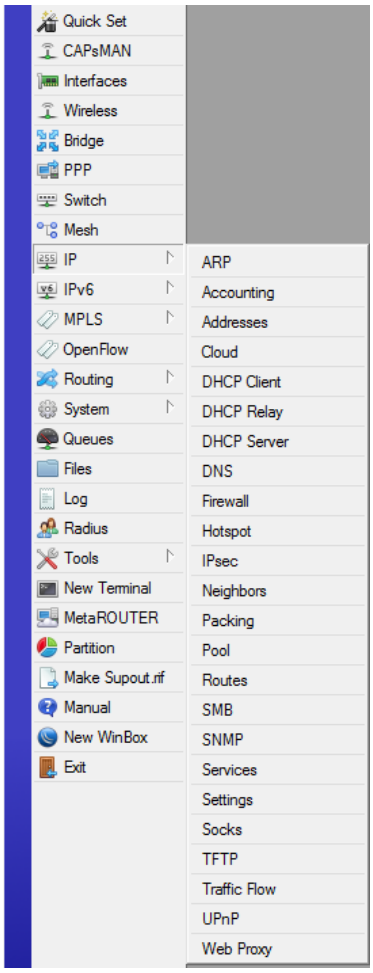
**Tabulka 1 – Porovnání dostupných licencí**

*Zdroj: Převzato z [4]*

### 1.2.2 Správa a připojení

Konfigurace zařízení je možná hned několika způsoby. Nejpohodlnější cestou je konfigurace pomocí utility **WinBox**.

Jedná se o proprietární aplikaci, umožňující konfiguraci směrovače „klikací“ metodou, tedy za použití myši. Výhodou aplikace je její velikost (desítky KB), rychlost a přehlednost. Přehlednost je nejsilnější stránkou aplikace. Oproti webovému rozhraní se zde lépe pracuje s jednotlivými položkami (samostatná okna) a tudíž umožňuje komfortnější konfiguraci a správu. Program WinBox lze ke konfiguraci zařízení použít i v případě, že zařízení nemá nakonfigurovanou IP adresu. Tato funkce se nazývá MAC server. Jak už je z názvu patrné, využívá se tedy připojení přes 2. vrstvu modelu ISO/OSI. Předpokladem pro použití tohoto připojení je nutnost zapojení do stejné sítě, ideálně pak crossover kabelem. Tuto možnost připojení lze v nastavení omezit, případně pak zakázat. [2]




**Obrázek 1 – Ovládací panel rozhraní WinBox**

*Zdroj: Vlastní*

Další klasickou možností je konfigurace přes **terminál**. K terminálu se lze připojit protokolem SSH nebo Telnet (nedoporučuje se z bezpečnostních důvodů). K terminálu neboli CLI (Command Line Interface) je též možno přistupovat skrze WinBox. Jedná se o CLI inspirovaný CLI systému IOS používaném na zařízeních Cisco. CLI nemá nic společného s linuxovým systémem, na kterém je RouterOS založen a obsahuje pouze příkazy pro RouterOS. Tím jsou v mnoha směrech omezeny možnosti směrovače (např. skriptování), ale je podpořena logika a jednoduchost konfigurace směrovače. [2]

Novinkou od verze 5 je konfigurační rozhraní **WebFig**. Jak už název napovídá, jedná se o rozhraní přístupné z webového prohlížeče, typicky na portu 80 (http) nebo 443 (https). Do verze 5 bylo webové rozhraní omezeno pouze na základní úkony, jako je nastavení IP adres a základní konfigurace DHCP. WebFig je již plnohodnotné a strukturálně shodné s utilitou WinBox.

Systém RouterOS nabízí i možnost připojení pomocí API (Application Programming Interface). Lze tak router napojit k řídicím systémům ISP nebo vytvořit vlastní aplikaci (například pomocí vypsání aktuálně připojených klientů na webové stránce).

## 1.3 Srovnání s produkty Cisco

Cenový rozdíl v hardwaru společnosti Cisco a Mikrotik je obrovský. Firma Cisco cílí na velké firmy, kde je kladen důraz na spolehlivost, bezpečnost a rychlost. Zařízení Mikrotik jsou podstatně levnější. Vzhledem k tomu, že produkty Mikrotik nemají specializované čipy a veškeré datové toky jsou řešeny procesorově, je i konfigurace složitější. Konfigurace vyžaduje větší znalost linuxových firewallů. To vyžaduje větší opatrnost při nastavování, jelikož i malá neznalost může dát příležitost potencionálnímu útočníkovi. U produktů Cisco je dokumentace velice rozsáhlá a uživatelská základna je podstatně větší. Tím pádem je i nastavení jednodušší. Firma Mikrotik cílí na menší firmy a zejména na poskytovatele internetového připojení, kteří jsou schopni z nabízeného sortimentu poskládat celou síť – od CloudCore routerů, přes vysílače a přístupové body až ke klientským zařízením a anténám.

Jedním z hlavních rozdílů je přístup k řízení datových toků. Cisco má specializovaný hardware a čipy, které řeší nativně síťovou komunikaci. Mikrotik zpracovává veškerou komunikaci procesorově, nikoli specializovanými čipy. Proto platí, že čím více pravidel, která se aplikují na danou komunikaci, tím menší bude výsledná datová propustnost.

Dalším rozdílem je přístup k software. Mikrotik vydává update RouterOS téměř týdně a aktualizace je prováděna téměř automaticky (po odsouhlasení administrátorem). Cisco vydává nové verze zřídka a update se provádí poměrně složitou cestou. Cenový rozdíl je opět obrovský.

V souhrnu lze říci, každý výrobce cílí na jinou sféru. Po Cisco platformě sáhnou data centra a velké firmy. Mikrotik zvolí menší firmy, poskytovatelé internetového připojení a nároční domácí uživatelé.

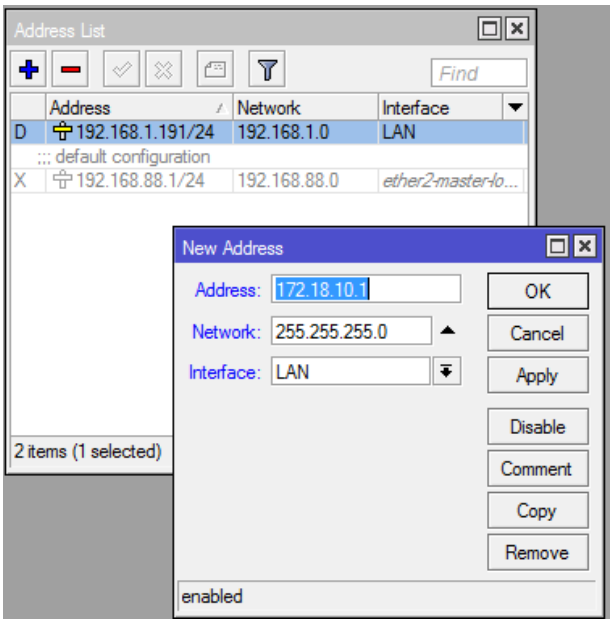
# 2 ZÁKLADNÍ NASTAVENÍ

Struktura CLI je až na pár výjimek shodná s rozložením nastavení v utilitě WinBox nebo WebFig. Pro přehlednost se méně zkušeným uživatelům této platformy doporučuje použití nástroje WinBox, který obsahuje tzv. „Safe Mode“. Tento režim zabrání nechtěnému „odříznutí se“ od routeru, ať už špatně nastaveným pravidlem na firewallu nebo vypnutím rozhraní, na které je WinBox připojen. Pokud WinBox v tomto režimu ztratí s routrerem spojení, vrátí se změny zpět. Příklady a dokumentace k systému RouterOS je dostupná na adrese: http://wiki.mikrotik.com/wiki/Manual:TOC.

## 2.1 Rozhraní a adresy

Prvním krokem při nastavování směrovače je nastavení adres na jednotlivé rozhraní. Jedno rozhraní může mít přiřazených více adres. To lze využít v případě, že vlastníte více veřejných adres a svého ISP připojeného do jednoho portu WAN.

```
/ip address
add address=192.168.240.1/24  interface=eth6
add address=192.168.250.1 netmask=255.255.255.0  interface=eth5
```




**Obrázek 2 – Konfigurace IP adres**

*Zdroj: Vlastní*

## 2.2 Bridge, STP

Skupinu portů je možno propojit bridgem a vytvořit tak virtuální switch. IP adresa se pak nastavuje pouze bridge portu, se kterým se pracuje stejně jako s klasickým rozhraním. Po vytvoření bridge je ve výchozím stavu zapnut RSTP (Rapid Spanning Tree Protocol), aby byla

eliminována možnost výskytu smyček. Pro bridge je také možné nastavit, aby komunikace procházela skrze firewall a řídila se nastavenými pravidly.

```
/interface bridge add name=bridge1
/interface bridge port add interface=7 bridge=bridge1
/interface bridge port add interface=8 bridge=bridge1
```

## 2.3 VLAN

Nejčastěji používaným protokolem pro virtuální LAN (VLAN) je IEEE 802.1Q. Jedná se o standardizovaný protokol zapouzdření, který definuje vložení čtyř bajtů jako identifikátor dané VLAN do záhlaví frame. Jedna VLAN můžu být čelenem jiné VLAN. Následující příkazy vytvoří VLAN 4,5 a 6, přičemž VLAN 6 překrývá VLAN 4 a 5. To znamená, že odchozí paket z VLAN 6 bude obsahovat VLAN-ID 6,4 a 5. Toto platí pro trunkové propoje.

```
/interface vlan
add interface=eth4 name=VLAN4 vlan-id=4
add interface=eth5 name=VLAN5 vlan-id=5
add interface=eth6 name=VLAN6 vlan-id=6
add interface=VLAN6 name=subVLAN4 vlan-id=4
add interface=VLAN6 name=subVLAN5 vlan-id=5
```

Pro vytvoření přístupového portu (access port) je nutné vytvořit bridge pro každou VLAN a následně do něj přidat VLAN a cílový port.

## 2.4 DHCP

Dynamic Host Configuration Protocol se využívá pro snadnou distribuci adres v síti. Mikrotik RouterOS v sobě implementuje jak DHCP klienta, tak server. Jak klienta, tak server je možné nakonfigurovat pro každé rozhraní odděleně. Obě služby jsou kompatibilní se standardem RFC 2131.

### 2.4.1 Klient

DHCP klient se nastavuje na určitý interface. Nabízí se zde možnost použití adresy z DHCP pro DNS a NTP. Další možností je přidání defaultní routy a její vzdálenosti. Všechny tyto možnosti jsou implicitně použity. Následujícím příkazem se vytvoří DHCP klient na rozhraní eth5 a aktivuje se.

```
/ip dhcp-client add interface=eth5 disabled=no
```

### 2.4.2 Server

Pro nadefinování DHCP serveru je nutné nejprve vytvořit rozsah adres, které budou přidělovány a přiřadit IP adresu k rozhraní, na kterém bude server spuštěn. Následně se vytvoří

síť, kde se definují adresy výchozí brány, adresa sítě, DNS server, NTP atd. V dalším kroku se server vytvoří a spustí.

```
/ip pool add name=klienti ranges=10.0.0.100-10.0.0.200
/ip address add address=10.0.0.1/24  interface=eth6
/ip dhcp-server network add address=10.0.0.0/24 dns-server=8.8.8.8
gateway=10.0.0.1
/ip dhcp-server add name=server1 interface=eth6 lease-time=12:00:00
address-pool=klienti add-arp=yes disabled=no
```

Mimo přidělování adres z rozsahu je možné pro danou MAC adresu nastavit statickou adresu, kterou dané zařízení dostane vždy.

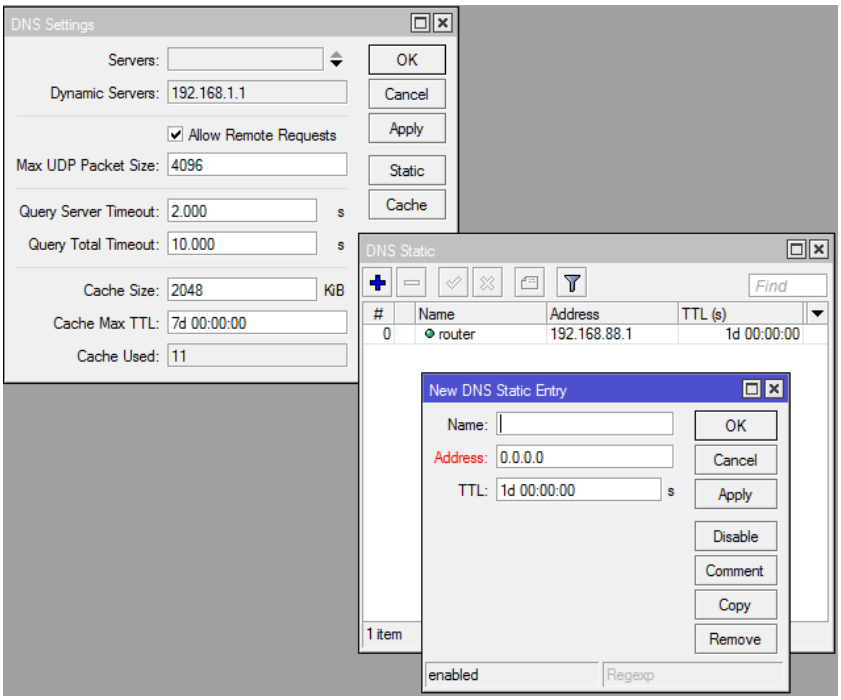
```
/ip dhcp-server lease add address=10.0.0.50 mac-address=00:11:22:33:44:55
server=server1
```

### 2.4.3 Možnosti zabezpečení

Pokud je požadavek na vytvoření sítě, kde budou adresy přiřazovány pouze dynamicky DHCP serverem, je vhodné síť ošetřit na 2 vrstvě ISO/OSI modelu. Za normálních okolností si směrovač uloží do ARP tabulky IP a MAC adresu. Poté umožňuje komunikaci. DHCP server nabízí možnost vkládání záznamů do ARP tabulky na základě přidělení adresy serverem. Poté už stačí jen na rozhraní, na kterém je spuštěn DHCP server nastavit ARP: reply-only, což znamená, že se do ARP tabulky nedostane záznam, pokud není staticky vytvořen, nebo vytvořen DHCP serverem. Zařízením se statickou IP adresou, bez validních ARP záznamů, není umožněno v rámci sítě komunikovat.

## 2.5 DNS

MikroTik RouterOS disponuje funkcí DNS serveru. Nejedná se však o plnohodnotný server, ale pouze o DNS vyrovnávací paměť s možností přidání vlastních statických záznamů. Na směrovači je možné pravidlem přesměrování portu UDP 53 „vnutit“ klientům lokální DNS server jako primární a dotazy na vzdálené servery směřovat na lokální. Tím lze zajistit například podstrčení IP adresy směrované na lokální server s chybovou hláškou omezeného přístupu do internetu. Jako FQDN lze použít i regulární výraz. Statické záznamy vyhodnocené jako regulární jsou zpracovávány prioritně.




**Obrázek 3 – Konfigurace DNS**

*Zdroj: Vlastní*

# 3 ROUTOVÁNÍ

Routování neboli směrování je základní funkcionalitou routeru neboli směrovače. Rozlišujeme dva základní typy směrování − statické a dynamické.

## 3.1 Statické

U statického je pravidlo pevně dané a nemění se na základě událostí. Typickým příkladem statického směrování je výchozí brána (default gateway). Jedná se o poslední pravidlo směrovací tabulky, které říká, že veškerá komunikace, která nebyla zpracována žádným jiným pravidlem a směrovač neví jak jí zpracovat, bude přeposlána výchozí cestou na další směrovač, který bude rozhodovat o jejím doručení.

## 3.2 Dynamické

Dynamické směrování zpravidla mění směrovací tabulku na základě změn v síti.

### 3.2.1 RIP

Protokol RIP (Routing information protocol), definovaný v RFC1058 je velmi starý. Díky jednoduchosti implementace a minimální nároky na znalosti správce sítě se stále často používá v malých sítích. Jeho metrikou je počet směrovačů na cestě k cílové síti. Jelikož eliminuje problém vzniku smyček mechanismem omezení metriky na hodnotu 15, nelze ho použít v rozsáhlé síti, kde by cestu k cíli tvořilo více než 15 směrovačů. Směrovací tabulka se rozesílá každých 30 sekund. Pokud směrovač o cestě neuslyší po dobu 180 sekund, je odstraněna.

Ve verzi 2 umožňuje protokol RIP oproti první verzi autentizaci sousedů, takže není tak jednoduché podvrhnout paket s falešnou cestou a způsobit tak nefunkčnost sítě. Další novinkou je šíření masek, takže na rozdíl od verze 1 je možné RIPv2 použít i v sítích s maskou proměnné délky (VLSM). Dále je možné směrovací tabulky šířit multicastem na místo broadcastu, čímž se omezí rušení okolních stanic, zvýší se bezpečnost a sníží se vytížení sítě. [5]

### 3.2.2 OSPF

Protokol OSPF (Open Shortest Path First) vytvořila organizace IETF přibližně v letech 1988 až 1991. Dnes je jedním z nejpoužívanějších směrovacích protokolů. Směrovač s tímto protokolem nejprve vypočte strom nejkratších cest a až pak z něj vytvoří směrovací tabulku. OSPF je typickým představitelem směrovacího protokolu typu link state. V paměti směrovače je vytvořena mapa celé sítě (topologická databáze), nad kterou je potom pomocí Dijkstrova algoritmu prováděn výpočet potřebný k nalezení nejlepší cesty do jednotlivých sítí. Protokol OSPF používá metriku označovanou jako cena (cost). Cena je číslo v rozsahu 1 až 65535, které

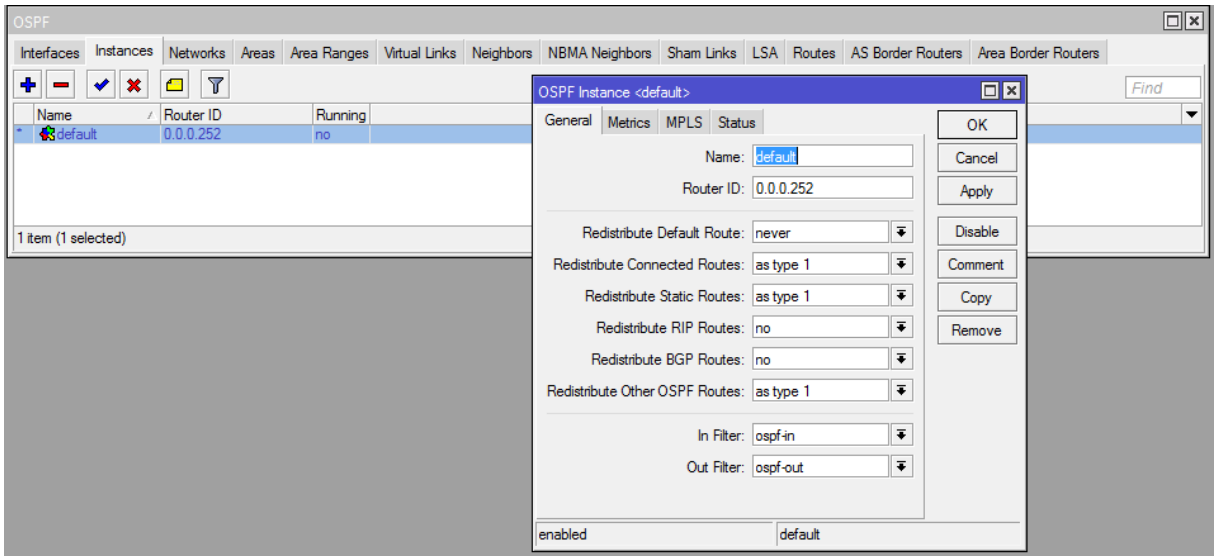
je přiřazeno ke každému rozhraní směrovače. Cena se odvozuje nepřímo úměrně od šířky pásma dané linky. Čím menší číslo, tím více bude linka preferovaná. [5]

Rozsáhlé sítě se z pravidla dělí na jednotlivé oblasti (area). Oblast je logická skupina směrovačů a linek mezi nimi. LSA se šíří pouze uvnitř dané oblasti. Výpočet SPF algoritmu se spouští pro každou oblast samostatně. Jelikož je SPF algoritmus poměrně náročný a vytěžuje směrovač, je toto nespornou výhodou. Změna v jedné z oblastí spustí výpočet SPF algoritmu pouze pro danou oblast. Oblasti jsou navzájem propojeny pomocí hraničních směrovačů (Area Border Router, ABR). Zvláštním případem je oblast 0, nazývaná jako páteřní (backbone). Tato oblast propojuje všechny ostatní oblasti. Veškerý provoz z jedné oblasti do druhé musí procházet přes oblast 0 a každá oblast musí být napojená přes hraniční směrovač (ABR) na oblast 0. Adresy v oblasti je vhodné navrhnout tak, aby byly ven propagovány sumarizované, tzn. Sítě 172.18.4.0/24, 172.18.5.0/24, 172.18.6.0/24 můžeme propagovat jako cestu do supernetu 172.18.4.0/22.

Následuje příklad jednoduchého nastavení OSPF směrování. Prvním příkazem vytvoříme instanci OSPF. Druhým nastavíme router-id. Další příkazy obsahují výčet přímo připojených sítí. Takto nastavený směrovač bude schopný komunikovat s ostatními směrovači v síti a vyměňovat si s nimi směrovací tabulky.

```
/routing ospf instance add name=default
/routing ospf instance set 0 router-id=10.10.10.1
/routing ospf network add network=10.10.1.0/30 area=backbone
/routing ospf network add network=10.10.2.0/30 area=backbone
/routing ospf network add network=10.10.20.0/24 area=backbone
/routing ospf network add network=10.10.15.0/24 area=backbone
```

Detailnější nastavení nalezneme v prostředí WinBox na kartě Routing, OSPF. Jednou z možností je i povolení redistribuce výchozí cesty.




**Obrázek 4 – Konfigurace OSPF**

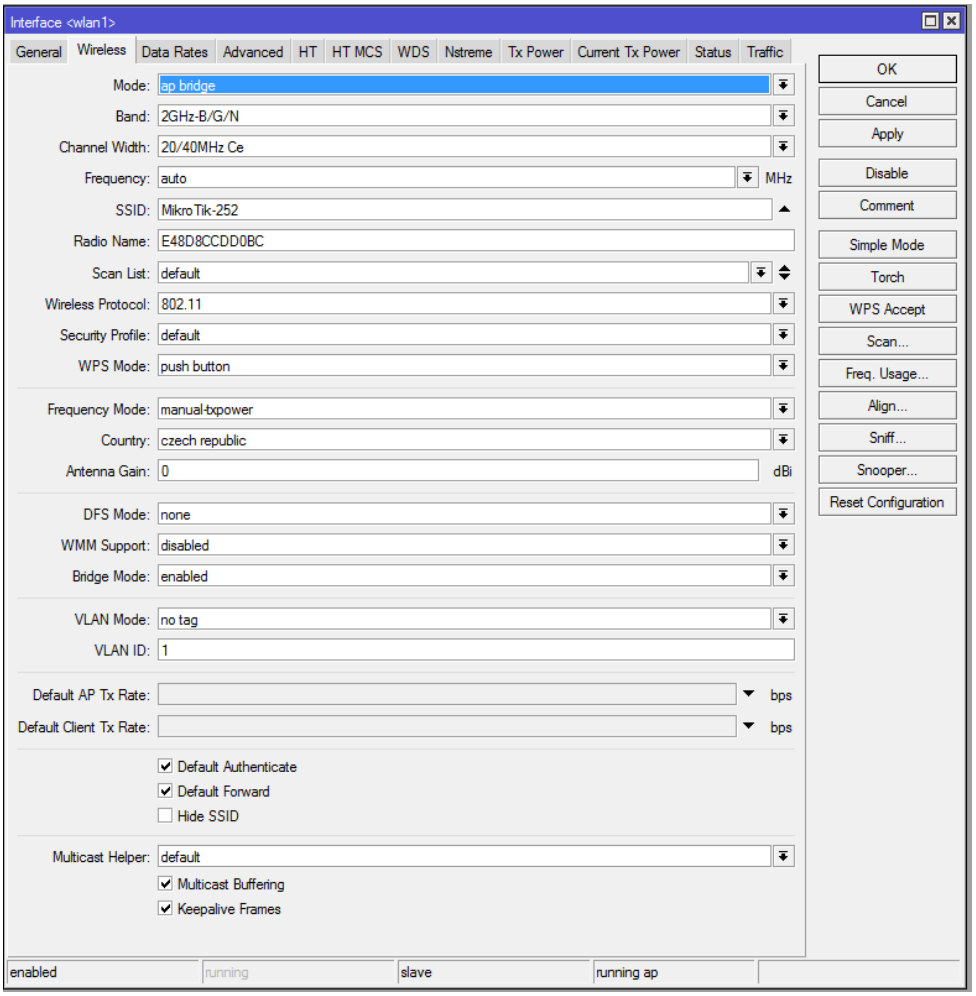
*Zdroj: Vlastní*

### 3.2.3 BGP

Protokol BGP (Border Gateway Protocol) je vhodný především pro velmi rozsáhlé sítě, které jsou tvořeny více autonomními oblastmi. BGP nepracuje s mapou sítě (topologickou databází) jako to dělá například OSPF, ale s grafem propojení autonomních systémů. V tomto grafu jsou pak vyhledávány cesty mezi sítěmi v různých autonomních systémech. Cesta je definována jako posloupnost čísel autonomních systémů (AS), přes které se lze k cílové síti dostat. Na rozdíl od vnitřních směrovacích protokolů nemá BGP jednoznačnou metriku, podle níž by za všech okolností automaticky volil nejkratší cesty do jednotlivých cílových sítí. Při směrování mezi autonomními systémy totiž směrujeme provoz přes cizí autonomní systémy. Autonomní systémy pod cizí správou mohou mít různá pravidla a politiky podle kterých je řízen provoz ve smyslu ze které sítě do které přes kterou linku. Tato pravidla se definují jako směrovací politika (routing policy). Podle těchto pravidel se poté určuje cesta z konkrétní sítě do jiné konkrétní sítě. [5]

# 4 WIRELESS

Velká část routerboardů je osazena bezdrátovou kartou, nebo obsahuje rozšiřující slot miniPCI do kterého je možné kartu vložit. RouterOS podporuje standard IEEE 802.11, což poskytuje kompletní podporu pro 802.11a, 802.11b, 802.11g, 802.11n a 802.11ac. Samozřejmostí je podpora šifrování WEP, WPA a WPA2. Bezdrátová karta může být nakonfigurována v několika režimech. Natavení bezdrátové karty obsahuje velké množství proměnných, tudíž je doporučeno ho provádět v grafickém prostředí (WinBox, WebFig), kde je vše přehledně nastavitelné krok po kroku.




**Obrázek 5 – Nastavení bezdrátového rozhraní**

*Zdroj: Vlastní*

## 4.1 AP bridge

AP Bridge je jeden z úplně základních režimů. V této konfiguraci funguje směrovač jako klasický přístupový bod (Access Point, AP). Bezdrátové rozhraní se pomocí mostu propojí s ostatními porty a zařízení může fungovat jako obyčejný domácí router.

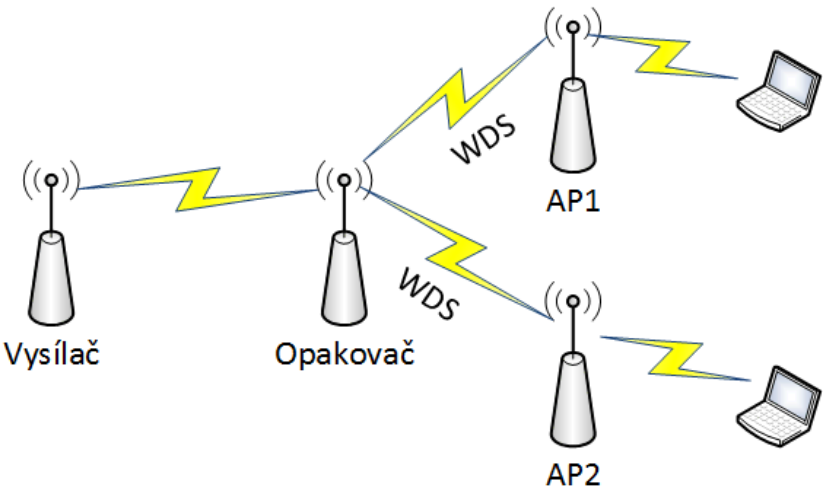
V tomto režimu RouterOS podporuje tvorbu virtuálních přístupových bodů. Tato funkce umožní z jedné fyzické karty vysílat hned několik sítí s různým SSID a různým způsobem zabezpečení. Virtuální síť je vysílána na stejné frekvenci jako hlavní síť, pod kterou je vytvořena.

## 4.2 Station

Bezdrátové zařízení v jakémkoli režimu Station bude vyhledávat přístupový bod a bude se k němu připojovat. Jednotlivé režimy se liší zejména v tom, jakým způsobem jsou předávány adresy z 2. síťové vrstvy mezi bezdrátovými prvky. To přímo ovlivňuje chování sítě, pokud chceme používat režim přemostění (bridge). Zařízení se připojuje k vysílači na stejné frekvenci, na které vysílá daná síť. Další podmínkou pro připojení je nastavení stejného protokolu.

## 4.3 WDS

Wireless Distribution System (WDS) je systém, který umožňuje bezdrátové propojení přístupových bodů v síti normy IEEE 802.11. Umožňuje rozšířit bezdrátovou síť pomocí více přístupových bodů bez nutnosti jejich propojení kabelovou páteřní sítí. Funkce WDS Bridge vytvoří mezi připojenými zařízeními transparentní most. Takto se dají například propojit přístupové body. Aby bylo možné jednotlivé body spojit, musí pracovat na stejné frekvenci.




**Obrázek 6 – Schéma propojení přístupových bodů pomocí WDS**

*Zdroj: Vlastní*

## 4.4 Nstreme, nv2

Nv2 je proprietární protokol společnosti Mikrotik. Umožňuje bezdrátovou komunikaci dvěma zařízením fungujícím na systému RouterOS. Bezdrátové karty směrovačů se nastaví stejně jako v klasickém režimu point-to-point, pouze se přepne protokol z 802.11 na Nv2. Nv2 je založen

na TDMA (Time Division Multiple Access) namísto CSMA (Carrier Sense Multiple Access) technologie přístupu k médiím používané v běžných zařízeních 802.11. Nv2 implementuje dynamický výběr frekvence, to umožňuje spolehlivou komunikaci v zarušených oblastech. Pokud se nastavuje již funkční spoj, je nezbytné, aby se nový protokol nejprve nastavil na klientském zařízení. Pokud se bude postupovat opačně, klient se po změně protokolu na AP již nepřipojí a nebude možné ho konfigurovat. [6]

Nstreme používá podobný princip jako nv2, je taktéž založen na TDMA. Jedná se o Mikrotik proprietární protokol, od kterého se pomalu upouští s nástupem novějších protokolů.

# 5 FIREWALL

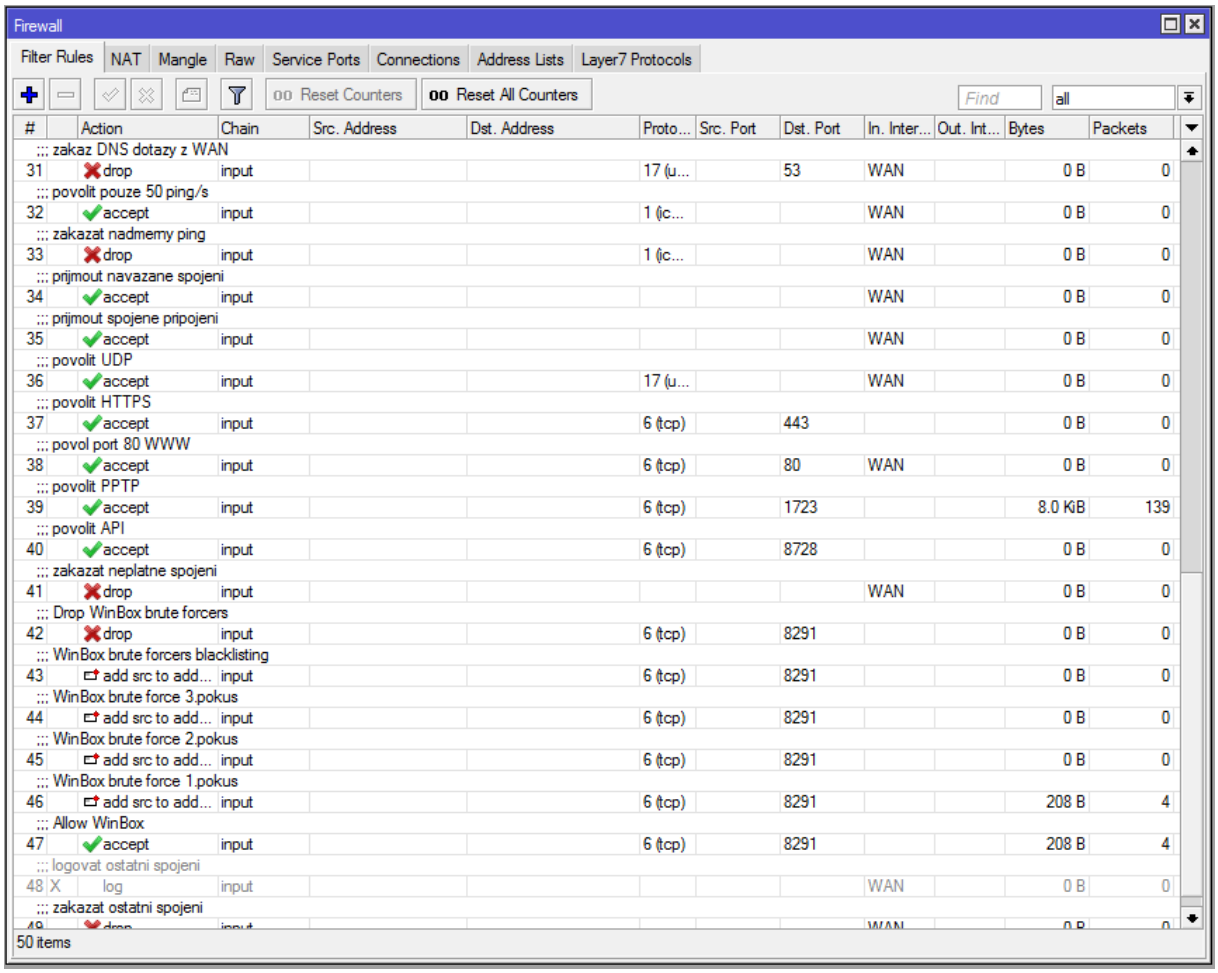
Brána firewall provádí filtrování paketů a tím zajišťuje bezpečnostní funkce, které se používají k řízení toku dat do, z a přes router. Spolu s Network Address Translation složí firewall jako nástroj pro zamezení neoprávněného přístupu k přímo připojeným sítím, hostům, na samotný router a také filtruje odchozí provoz. Správně nakonfigurovaný firewall hraje klíčovou roli v efektivním zabezpečení celé sítě. Mikrotik RouterOS má velice dobře implementovanou bránu firewall se spousto pokročilých funkcí a možnostmi filtrování. Mezi hlavní přednosti patří možnosti:

- Stavová kontrola paketů
- Filtrování na 7. (aplikační) vrstvě, například dle regulérního výrazu (regexpu)
- Peer-to-peer filtrování
- Klasifikace provozu dle:
    - Zdrojové MAC adresy.
    - IP adresy (konkrétní, rozsahu nebo seznamu).
    - IP protokolu.
    - Portu nebo rozsahu portů.
    - Obsahu paketu.
    - Velikosti paketu.

Toto je jen výčet některých z mnoha funkcí a kritérií, které lze při nastavování použít.

## 5.1 Pravidla

Celý systém filtrování je založen na souboru pravidel, která jsou procházena odshora směrem dolů. Každé pravidlo se skládá z podmínkové části a výkonné části. Ve chvíli kdy je podmínka pravidla splněna, provede se výkonná část pravidla.

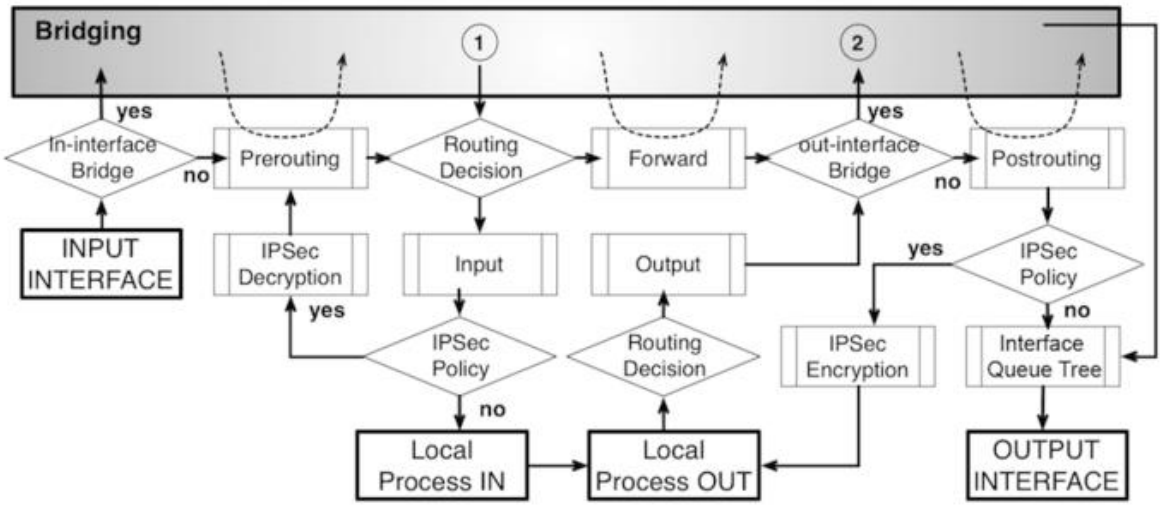



**Obrázek 7 – Nastavení pravidel firewallu**

*Zdroj: Vlastní*

Základem každého pravidla je „Chain“, v překladu řetězec. Řetězec určuje, kdy má být paket zpracován. K dispozici jsou tři předem definované řetězce, které nelze odstranit:

- Input – slouží ke zpracování paketů vstupujících do směrovače přes jedno z rozhraní s cílovou IP adresou, která je jednou z adres směrovače. Pakety procházející přes směrovač (forward) nejsou zpracovány v rozporu s tímto pravidlem.
- Output – slouží ke zpracování paketů odcházejících jedním z rozhraní směrovače. Pakety procházející přes směrovač (forward) nejsou zpracovány v rozporu s tímto pravidlem.
- Forward – slouží ke zpracování paketů procházejících směrovačem v libovolném směru.




**Obrázek 8 – Schéma průchodu paketu směrovačem**

*Zdroj: Převzato z [7]*

Další řetězce je možno vytvořit dle specifických požadavků. Například je možné si vytvořit pravidlo pro „Domácí síť“. Veškerá komunikace sítě 192.168.1.0/24 bude zpracována řetězcem „DomaciSit“. V dalších pravidlech pro tuto síť pak není nutné nastavovat zdrojové nebo cílové adresy, pokud použijeme nově vytvořený řetězec „DomaciSit“.

```
/ip firewall filter add action=jump chain=forward jump-target=DomaciSit
src-address=192.168.1.0/24
/ip firewall filter add action=jump chain=forward jump-target=DomaciSit
dst-address=192.168.1.0/24
```

Ve výkonné části pravidla je možné použít níže popsané nastavení:

- *accept* – přijme paket a další pravidla se již neaplikují,
- *add-dst-to-address-list* – přidá cílovou adresu do listu definovaného parametrem `address-list`,
- *add-src-to-address-list* – přidá zdrojovou adresu do listu definovaného parametrem `address-list`,
- *drop* – paket je zahozen bez další odezvy,
- *jump* – paket přeskočí na uživatelem definovaný řetězec dle parametru `jump-target`, kde je dále zpracován,
- *log* – vloží do systémového logu zprávu s následujícím obsahem: vstupní rozhraní, výstupní rozhraní, zdrojová MAC adresa, protokol, zdrojová IP adresa a port, cílová IP adresa a port, velikost paketu. Poté, co je pravidlo zpracováno pokračuje se na další v pořadí (stejně jakou u akce `passthrough`),

- *passthrough* – ignoruje toto pravidlo a posouvá se na další v pořadí (používá se na statistiky a podobně),
- *reject* – paket je zahozen a je odeslána ICMP zpráva o odmítnutí,
- *return* – vrací paket zpět do řetězce, kterým byl zpracován,
- *tarpit* – paket je zachycen a je dále udržováno TCP spojení (odesílá se odpověď o navázání spojení SYN/ACK na příchozí TCP SYN paket).

Standardně se využívá restriktivního firewallu, tedy logiky „co není povoleno, je zakázáno“. To znamená, že se vytvoří soubor základních pravidel pro komunikaci do vnější sítě a jako poslední se vytvoří pravidlo ve smyslu „všechno ostatní zahoď“. Tím pádem, je komunikace, která neodpovídá ani jednomu povolujícímu pravidlu, automaticky zahozena. Následující příkazy demonstrují jednoduché základní nastavení pravidel, sloužící k zabezpečení směrovače a vnitřní sítě.

```
/ip firewall filter
add action=drop chain=input dst-port=53 in-interface=WAN protocol=udp
add chain=input in-interface=WAN limit=2,2:packet packet-size=0-512
protocol=icmp
add action=drop chain=input in-interface=WAN protocol=icmp
add chain=input connection-state=established in-interface=WAN
add chain=input connection-state=related in-interface=WAN
add action=drop chain=input connection-state=invalid in-interface=WAN
add chain=input dst-port=8291 protocol=tcp
add action=drop chain=input in-interface=WAN
```

Prvním pravidlem se eliminuje možnost přístupu na interní DNS server z WAN. Druhé pravidlo povoluje komunikaci ICMP z WAN portu, ale pouze 50 paketu za sekundu, s maximální velikostí 512B. ICMP komunikace, která neprojde druhým pravidlem je na třetím pravidle zakázána. Čtvrté pravidlo propustí již navázaná spojení, páté pak povolí spojení příbuzná již navázaným spojením. Další pravidlo zahazuje neplatná spojení. Předposledním pravidlem se povolí připojení na port TCP 8291, což je port standardně používaný pro konfiguraci pomocí utility WinBox. Všechna ostatní komunikace, která neodpovídá žádnému z výše uvedených pravidel je v poslední fázi zahozena.

## 5.2 Značkování

Pro potřeby lepšího rozčlenění komunikace a následného filtrování je možné pakety nebo celá spojení značkovat. Značkování se neprojeví v okolních sítích, funguje jen v rámci směrovače. Téměř všechny funkce směrovače jsou schopné se značkováním pracovat. Značkování se

nejčastěji používá pro implementaci QoS (Quality of Service), kde se na základě značkování určuje priorita daného spojení. Značkovat je též možné i pro účely routování.

```
/ip firewall mangle
add action=mark-routing chain=prerouting new-routing-mark=doma_nat
passthrough=no src-address-list=domaci_sit_NAT
/ip route
add distance=1 gateway=vpn_domu routing-mark=doma_nat
```

Prvním příkazem vytvoříme značkování směrování. Komunikace adres z adresního listu „domaci_sit_NAT“ bude označkováno směrovací značkou „doma_nat“. Dalším příkazem vytvoříme výchozí cestu pro směrování paketů se značkou „doma_nat“. Adresy z výše zmíněného adresního listu budou při průchodu směrovány na směrovač připojený za rozhraním „vpn_domu“.

Pro účely QoS je vhodné značkovat komunikaci dle protokolů, kterým se bude nastavovat priorita, případně omezovat rychlost. Následující příkazy rozdělují příchozí komunikaci na portu 80 (standardní webové stránky a stahování) na 3 druhy. Spojení do velikosti 0,5MB je klasifikováno jako obyčejný přenos značkou „www_in“. Spojení od 0,5MB do 50MB je označeno značkou „www_in_download“. Komunikace, která přenese více než 50MB dat je klasifikována jako stahování velkého souboru a označena značkou „www_in_download_big“. S takto označenou komunikací je možné dále pracovat například omezením rychlosti pro stahování velkých souborů a dalšími způsoby.

```
/ip firewall mangle
add action=mark-packet chain=prerouting connection-bytes=50000000-0 in-
interface=WAN new-packet-mark=www_in_download_big passthrough=no
protocol=tcp src-port=80
add action=mark-packet chain=prerouting connection-bytes=500000-0 in-
interface=WAN new-packet-mark=www_in_download passthrough=no protocol=tcp
src-port=80
add action=mark-packet chain=prerouting connection-bytes=0-500000 in-
interface=WAN new-packet-mark=www_in passthrough=no protocol=tcp src-
port=80
```

## 5.3 Adresní listy

Účelem adresních listů je převážně zjednodušení konfigurace. Není třeba nastavovat pravidlo pro několik IP adres odděleně, ale lze je přidat do adresního listu a vytvořit poté jedno pravidlo pro celý adresní list. Záznamy v adresních listech dělíme na dva typy:

- Statické – záznamy trvalé.
- Dynamické – záznamy s předem určenou dobou, po kterou budou platné. Po vypršení této doby dojde k automatickému odstranění záznamu z adresního listu.

V následujícím příkladu se vytvoří jeden dynamický záznam s platností 24 hodin a jeden statický s permanentní platností (do manuálního odstranění).

```
/ip firewall address-list
add list=zakaz  address=10.10.10.1 timeout=1d
add list=zakaz  address=10.10.10.2
```

Ideálním využitím pro dynamické adresní listy je například čítač přihlášení s následným zablokováním po několika neúspěšných pokusech.

```
/ip firewall filter

add action=drop chain=input dst-port=8291 protocol=tcp src-address-
list=wb_blacklist

add action=add-src-to-address-list address-list=wb_blacklist address-list-
timeout=1w chain=input comment="blacklist" connection-state=new dst-
port=8291 protocol=tcp src-address-list=pokus_3

add action=add-src-to-address-list address-list=pokus_3 address-list-
timeout=1m chain=input comment="3.pokus" connection-state=new dst-port=8291
protocol=tcp src-address-list=pokus_2

add action=add-src-to-address-list address-list=pokus_2 address-list-
timeout=1m chain=input comment="2.pokus" connection-state=new dst-port=8291
protocol=tcp src-address-list=pokus_1

add action=add-src-to-address-list address-list=pokus_1 address-list-
timeout=1m chain=input comment="1.pokus" connection-state=new dst-port=8291
protocol=tcp

add chain=input dst-port=8291 protocol=tcp
```

Soubor těchto pravidel funguje následovně. Při prvním pokusu o přihlášení (navázáním nového spojení) je zdrojová IP adresa přidána do adresního listu „pokus_1“ s časovačem nastaveným na jednu minutu. Pokud bude zaznamenán další pokus o přihlášení z IP adresy, která se již nachází v adresním listu „pokus_1“, zapíše se do adresního listu „pokus_2“, následně „pokus_3“. V případě výskytu IP adresy v listu „pokus_3“ a zaznamenaném další přihlášení dochází k přesunu na „blacklist“ po dobu jednoho týdne. IP adresy ze seznamu „blacklist“ nemají povolený přístup na port TCP 8291 a jejich komunikace je zahazována.

## 5.4 L7 filtrování

Jedním z kritérií při filtrování může být konkrétní aplikace či specifický protokol, který neužívá pravidelně stejný port, podle kterého by se dal identifikovat. Mikrotik RouterOS umožňuje filtrování na základě obsahu paketu. L7 filtr hledá otisk v prvních 10 paketech spojení nebo prvních 2KB. V případě, že není nalezena shoda, další kontrola spojení se již neprovádí. Prvním krokem je vytvoření otisku, který se bude v paketech hledat.

```
/ip firewall layer7-protocol
add name=vnc regexp="^rfb 00[1-9]\\.00[0-9]\\x0a\$"
```

Tímto příkazem se přidá otisk, který detekuje komunikaci VNC. Následujícím příkazem zakážeme komunikaci, která se identifikuje jako VNC na základě regulérního výrazu (regexp).

```
/ip firewall filter
add action=drop chain=forward layer7-protocol=vnc
```

Tato metoda filtrování je podstatně náročnější na zpracování, proto se doporučuje ji využívat jen, pokud je to nezbytně nutné a nedá se filtrování provádět jinak (například na základě čísla portu atd.). Ne všechny aplikace se dají tímto způsobem spolehlivě detekovat.

# 6 NAT

Network Address Translation je internetový standard RFC 1918, který umožňuje přeložit více privátních adres za jednu nebo více veřejných adres. Router provádějící překlad adres je z pravidla používán jako výchozí brána pro lokální síť. Existují dva typy překladu adres, zdrojový a cílový. Zařízení za směrovačem provádějícím překlad adres nemají skutečnou end-to-end konektivitu. Některé protokoly nemusí po překladu adres fungovat správně. Příkladem může být UDP komunikace P2P nebo IPsec spojení. [8]

Hlavní výhodou NAT je úspora IP adres. Vzhledem k narůstajícímu počtu uživatelů internetu je toto podstatná vlastnost, kterou dnes využívají poskytovatelé internetu, mobilní operátoři a hlavně firmy či domácí sítě. Další nezanedbatelnou výhodou je zvýšení bezpečnosti vnitřní sítě skrytím vnitřních adres a jejich oddělením NAT prvkem. Jedná se o nejjednodušší způsob ochrany proti neoprávněnému vniknutí z vnější sítě. Zařazením NAT prvku mezi vnitřní a vnější síť se automaticky vytvoří jednoduchý firewall. Žádný počítač z vnější sítě nemůže kontaktovat počítač na vnitřní síti, dokud spojení nenaváže tento počítač sám. [9]

## 6.1 Zdrojový (Source NAT)

Tento typ překladu se aplikuje na pakety procházející přes router z vnitřní sítě do sítě vnější. Typicky se používá na většině routerů v domácnostech, kde odděluje lokální síť od sítě ISP.

Základním a nejjednodušším nastavení je **maškaráda** (Masquerade). Následující příkaz způsobí, že veškerý provoz odcházející portem WAN bude „schován“ za IP adresu portu WAN.

```
/ip firewall nat add chain=srcnat action=masquerade out-interface=WAN
```

Další možností je překlad konkrétních podsítí na konkrétní IP adresy. Následující příkazy vytvoří dvě pravidla, pro každou podsíť jedno. Data pocházející ze sítě 192.168.1.0/24 budou přeložena na IP 1.1.1.2. Data ze sítě 192.168.2.0/24 budou přeložena na adresu 1.1.1.2. Pokud nebudou nastavena další pravidla, IP adresy paketů z jiných sítí nebudou překládány, dále se budou řídit směrovacími pravidly, tzn., nebudou „maskována“ za IP adresu odchozího portu.

```
/ip firewall nat
add chain=srcnat src-address=192.168.1.0/24 action=src-nat to-
addresses=1.1.1.1 out-interface=Public
add chain=srcnat src-address=192.168.2.0/24 action=src-nat to-
addresses=1.1.1.2 out-interface=Public
```

## 6.2 Cílový (Destination NAT)

Cílový překlad adres je ve své podstatě opakem zdrojového překladu adres. Adresy z vnější sítě se překládají do vnitřní sítě. Je možné převádět veškerý datový provoz, tedy provádět překlad adres, nebo překládat komunikaci jen z vybraného portu nebo rozsahu portů na vybraný port nebo rozsah. Tato funkce se nazývá Port forwarding. Praktické využití najde například, pokud se v lokální síti nachází web server, ke kterému je potřeba povolit přístup i z vnější sítě (WAN).

```
/ip firewall nat add chain=dstnat dst-address=1.1.1.1 dst-port=80
action=dst-nat protocol=tcp to-address=192.168.1.1 to-port=80
```

Výše uvedený příkaz zajistí, že komunikace příchozí na adresu 1.1.1.1 vnějšího rozhraní na portu 80 bude přeposlána na port 80 na adresu 192.168.1.1 ve vnitřní síti.

Následující pravidlo povolí serveru v lokální síti inicializovat spojení do vnější sítě pod adresou 1.1.1.1. Dalším pravidlem povolíme inicializaci spojení z vnější sítě (příchozí spojení na adresu 1.1.1.1) do vnitřní sítě na adresu 192.168.1.1. Ve své podstatě se jedná o přemapování vnější adresy na vnitřní. [10]

```
/ip firewall nat add chain=dstnat dst-address=1.1.1.1 action=dst-nat to-
addresses=192.168.1.1

/ip firewall nat add chain=srcnat src-address=192.168.1.1 action=src-nat
to-addresses=1.1.1.1
```

## 6.3 1:1 mapování

Přemapování celé sítě nebo jen rozsahu adres je méně využívané, ale je též možné. Příkladem praktického využití je nasazení do firmy, kde výrobní robot a jeho každé čidlo má přiřazenou vlastní IP z dané podsítě. Pro sjednocení a přístup ke konfiguraci z jedné stě by bylo zapotřebí složité směrování. Touto funkcí jsme schopni přemapovat IP adresy z rozsahu definovaného pro roboty na rozsah, který bude přijatelný pro propojení s produkční sítí. Příklad tohoto nastavení znázorňují následující příkazy, na kterých je vidět, že rozsah 1.1.1.1-1.1.1.20 a 2.2.2.1-2.2.2.20 se přamapuje do jedné sítě na rozsah 192.168.1-40.

```
/ip firewall nat add chain=dstnat dst-address=1.1.1.1-1.1.1.20
action=netmap to-addresses=192.168.1.1-192.168.1.20
/ip firewall nat add chain=srcnat src-address=192.168.1.1-192.168.1.20
action=netmap to-addresses=1.1.1.1-1.1.1.20
/ip firewall nat add chain=dstnat dst-address=2.2.2.1-2.2.2.20
action=netmap to-addresses=192.168.1.21-192.168.1.40
/ip firewall nat add chain=srcnat src-address=192.168.1.21-192.168.1.40
action=netmap to-addresses=2.2.2.1-2.2.2.20
```

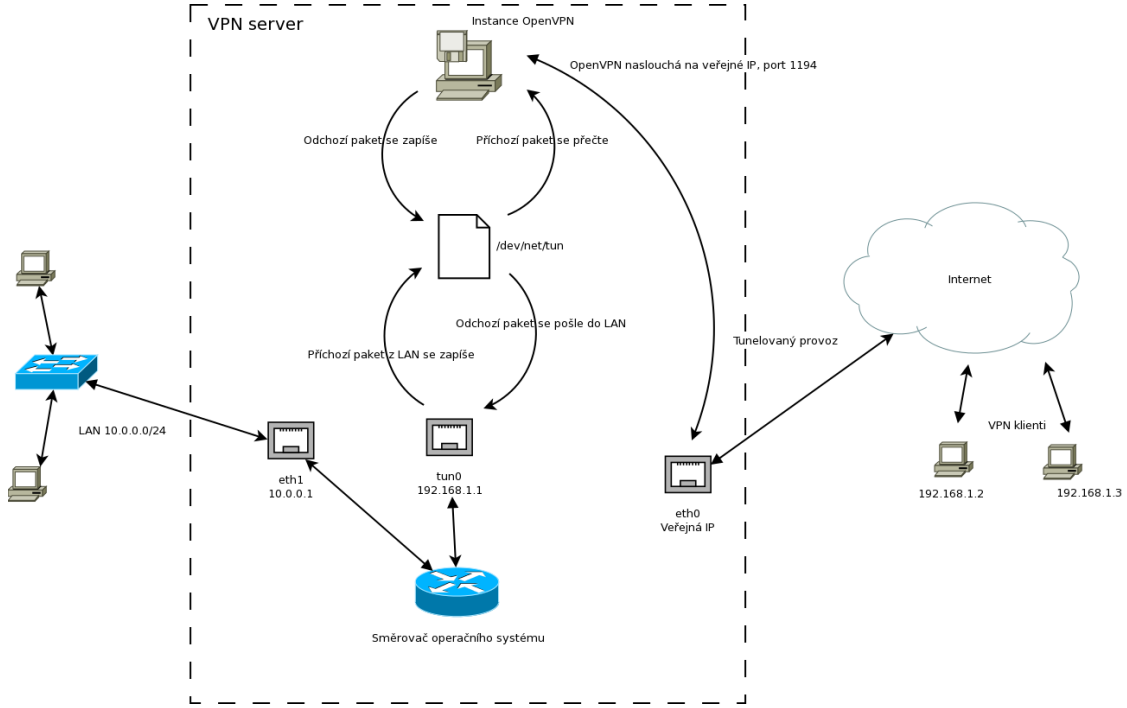
# 7 VPN A TUNELY

Termín VPN (Virtuální privátní síť) se používá pro počítačové sítě typicky postavené nad veřejnou síťovou infrastrukturou. Termínem virtuální rozumíme fakt, že síť fyzicky neexistuje, resp. neexistují její fyzické hardwarové komponenty (směrovače, přepínače, atd.). Taková síť typicky využívá již existující infrastrukturu a provoz této sítě je celý vložen do paketů transportní vrstvy.

VPN můžeme rozdělit na dva typy podle využití. Prvním typem může být Point-to-Point VPN, která zpravidla spojuje dva body a umožňuje jim komunikovat skrze sít třetí. Za druhý typ můžeme považovat RAS VPN (Remote Access Virtual Private Network). Tento typ konfigurace umožňuje více klientů se připojit k jednomu bodu, typicky pak k podnikovým infrastrukturám.

## 7.1 OpenVPN

Jedním z nejrozšířenějších protokolů je OpenVPN. Podporu tohoto protokolu nalezneme u většiny linuxových zařízení s podporou balíčkovacích systémů. Toto virtuální rozhraní se chová velmi jednoduše. Jakýkoliv paket, který vstupuje do tohoto rozhraní, se rozbalí a jeho obsah se zapíše do systémového souboru (např. */dev/net/tun)*. Takto funguje oboustranná komunikace s virtuálním rozhraním. [11]




**Obrázek 9 – Princip komunikace serveru s rozhraním**

*Zdroj: Převzato z [11] (Roman Pavlík, 2013)*

Obrázek výše ukazuje již pokročilejší využití OpenVPN na serveru. OpenVPN naslouchá na veřejné IP adrese fyzického rozhraní. Standardně se používá port 1194. VPN server spojuje klienty a umožňuje jim se připojit do sítě 10.0.0.0/8. Pro přístup do dítě 10.0.0.0/8 je nezbytné, aby měl každý klient nastavenou cestu do sítě přes bránu 192.168.1.1.

OpenVPN může pracovat ve dvou režimech – TUN nebo TAP. Režimy se liší pouze způsobem práce s ethernetovými rámci. Režim TUN umožňuje přístup k L2 vrstvě, TAP až k Vrstvě L3. [12]

Prvním krokem je základní nastavení, IP adresy na rozhraních, NAT do internetu, výchozí brána, atd. Dalším krokem je nastavení rozsahu, který bude využíván klienty.

```
/ip pool add name=openvpn ranges=10.10.10.10-10.10.10.100
```

Další příkazy vytvoří profil, který definuje například IP adresu virtuálního rozhraní.

```
/ppp profile add change-tcp-mss=default local-address=10.10.10.1
name="your_profile" only-one=default remote-address=openvpn use-
compression=default use-encryption=required use-vj-compression=default
```

Aby bylo možné se přihlásit a ověřit uživatelským jménem a heslem, je nutné vytvořit účet. Účty je stejně tak možné ověřovat vůči RADIUS serveru.

```
/ppp secret add disabled=no limit-bytes-in=0 limit-bytes-out=0
name="username" password="password" routes="" service=any
```

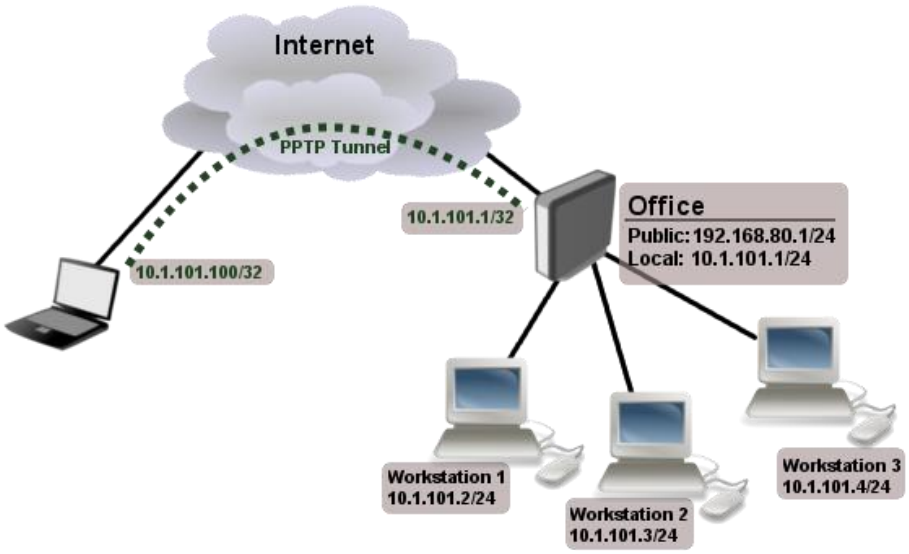
Předposledním, avšak nejdůležitějším krokem je vytvoření samotného virtuálního rozhraní, které bude používáno OpenVPN serverem. V posledním kroku pouze povolíme na firewallu příchozí komunikaci na portu 1194 TCP, aby bylo možné se k serveru připojit.

```
/interface ovpn-server server set auth=sha1,md5 certificate=router_cert
cipher=blowfish128,aes128,aes192,aes256 default-profile=your_profile
enabled=yes keepalive-timeout=disabled max-mtu=1500 mode=ip netmask=29
port=1194 require-client-certificate=no

/ip firewall filter add action=accept chain=input comment="OpenVPN"
disabled=no dst-port=1194 protocol=tcp
```

## 7.2 PPTP, L2TP

**PPTP** (Point-to-point Tunneling Protocol) je protokol, který umožňuje vytvářet **VPN** (Virtual Private Network) napříč sítěmi třetích stran (typickým příkladem je síť internet). Umožňuje se šifrovaným tunelem spojit za využití sítě třetí strany, typicky pak sítě internet, z různé lokality bez nutnosti dedikované linky. Je pravdou, že od používání protokolu PPTP se již upouští.

Aktuálně se spíše používá novější verze L2TP, nebo komunitní implementace VPN – OpenVPN. Nicméně vzhledem k jednoduchosti nasazení stále nalézá své uplatnění. Protokol PPTP využívá pro komunikaci protokol GRE (47) a TCP portu číslo 1723. Následující obrázek zobrazuje příklad konfigurace PPTP VPN v režimu vzdáleného připojení klientů. Dalším scénářem by mohlo být propojení dvou směrovačů tunelem (Site-to-Site).



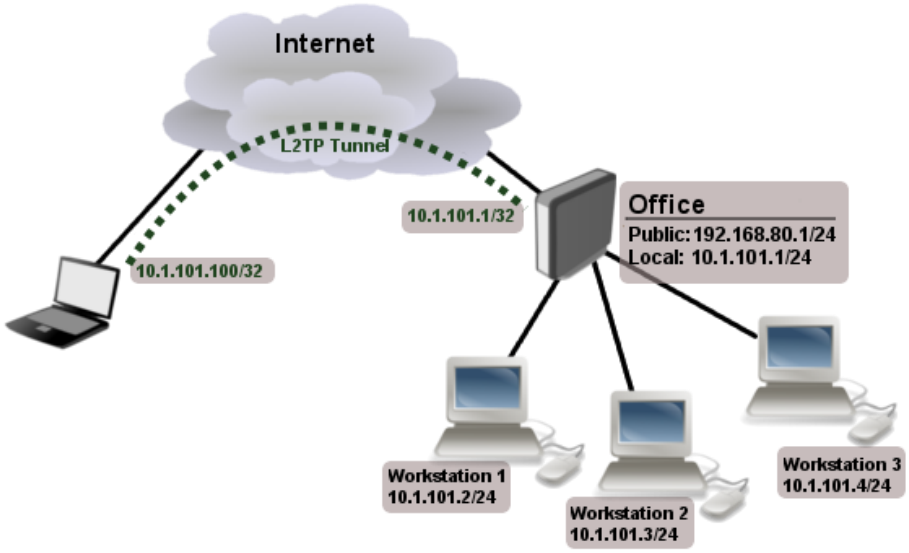

**Obrázek 10 – Schéma zapojení sítě s použitím tunelu PPTP**

*Zdroj: Převzato z [13]*

Dosažení funkční konfigurace zobrazené na obrázku výše je velice triviální. Prvním příkazem vytvoříme uživatelský účet v lokální databázi směrovače. Druhý příkaz povolí službu PPTP serveru. Případně je možné v záložce PPP v prostředí WinBox konfiguraci dále upravit (definovat profil s rozsahem adres pro více klientů, nastavit nebo upravit šifrování, atd.). Aby mohl připojený klient komunikovat s ostatními zařízeními v síti, je nutné nastavit na lokálním rozhraní sítě (ethernet nebo bridge, záleží na konkrétním nastavení) parametr arp na proxy-arp. Pokud se za směrovačem nachází i jiná síť, je nutností nastavit správné směrování na klientské straně. [13]

```
/ppp secret add name=Laptop service=pptp password=123 local-
address=10.1.101.1 remote-address=10.1.101.100

/interface pptp-server server set enabled=yes

/interface bridge set LAN arp=proxy-arp
```

**L2TP** (Layer 2 Tuneling Protocol) je standardní tunelovací protokol, který zapouzdřuje PPP komunikaci. Pokud volíme mezi protokoly PPTP a L2TP, je L2TP vhodnější volbou, jelikož podporuje šifrování přes IPsec. Spojení je navazováno na portu 1701 UDP. Jak už je z názvu patrné, tunelem prochází komunikace již na L2 vrstvě, což umožňuje nativní přímý přístup ke vzdáleným zařízením ve stejné síti. Jak je vidět níže, konfigurace je téměř stejná, liší se jen název protokolu. [14]



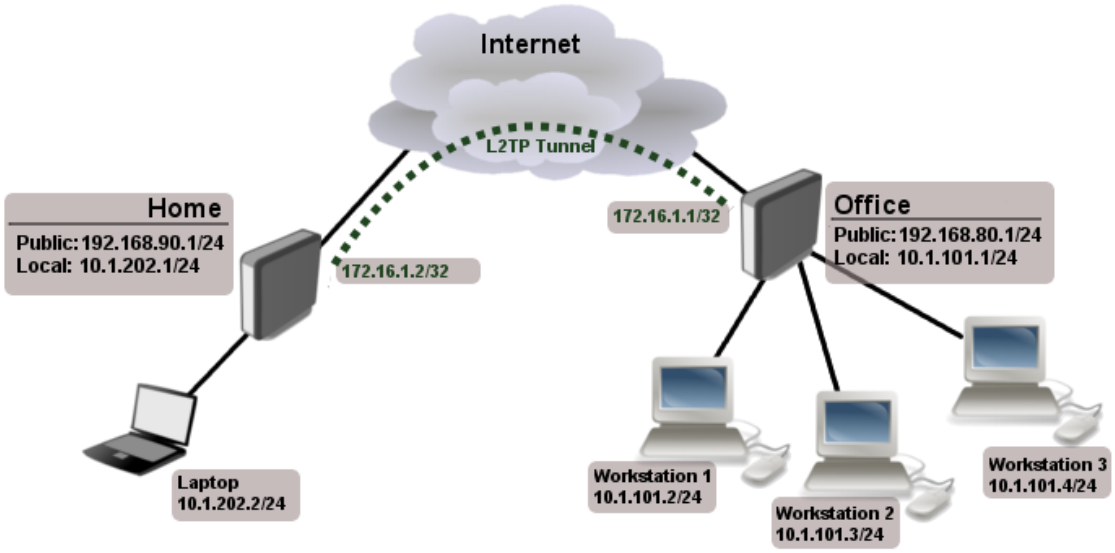

**Obrázek 11 – Schéma zapojení sítě s použitím tunelu L2TP**

*Zdroj: Převzato z [14]*

```
/ppp secret add name=Laptop service=l2tp password=123 local-
address=10.1.101.1 remote-address=10.1.101.100

/interface l2tp-server server set enabled=yes

/interface bridge set LAN arp=proxy-arp
```

## 7.3 IPsec

Jedná se o bezpečnostní mechanizmus založený na kryptografické bázi. O IPsec by se dalo hovořit jako o rozšíření týkající se třetí síťové vrstvy referenčního modelu ISO/OSI. IPsec může být použit s jakýmkoli protokolem postaveným nad IP výše. Bezpečnost zaručuje podpora řady šifrovacích algoritmů hashovacích funkcí jako jsou SHA-1, SHA-2 a MD5. Jedná se o ideální doplnění a zabezpečení tunelovacích protokolů jako je například L2TP. [15]




**Obrázek 12 – Schéma zapojení sítě point-to-point za použití L2TP a IPsec**

*Zdroj: Převzato z [14]*

**Office:**

```
/ppp secret add name=Home service=l2tp password=heslo local-
address=172.16.1.1 remote-address=172.16.1.2 routes="10.1.202.0/24
172.16.1.2 1"
/interface l2tp-server server set enabled=yes use-ipsec=yes ipsec-
secret=MojeTajneHeslo default-profile=default
/ip firewall filter
add chain=input protocol=udp port=1701,500,4500
add chain=input protocol=ipsec-esp
```

**Home:**

```
/interface l2tp-client add user=Home password=heslo use-ipsec=yes ipsec-
secret=MojeTajneHeslo connect-to=192.168.80.1 disabled=no
/ip route add dst-address=10.1.101.0/24 gateway=l2tp-out1
```

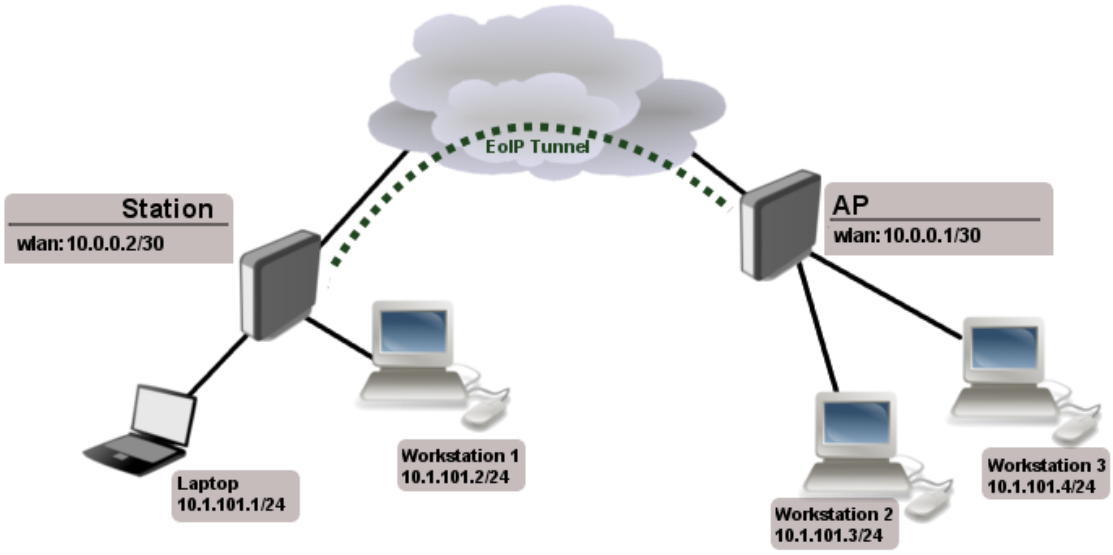
Při nastavování Office směrovače nesmíme zapomenout nastavit pravidla na firewall, jinak bude příchozí spojení zahozeno. Stejně tak nesmíme opomenout přidání cesty do domácí sítě, což je možno udělat již při vytváření uživatelského účtu. Cesta se pak do směrovací tabulky přidá jen ve chvíli, kdy je spojení s klientem navázáno.

Na straně Home směrovače je nastavení jednodušší. Postačí vytvoření L2TP-client rozhraní, zadání správných přihlašovacích údajů a cílové IP adresy. Pro komunikaci s firemní sítí je nutno přidat cestu, která směruje provoz na IP adresy 10.1.101.0/24 na rozhraní L2TP tunelu. [16]

## 7.4 EoIP

Ethernet over IP (EoIP) je tunelovací protokol, který vytvoří virtuální rozhraní, které se chová stejně jako klasické fyzické rozhraní. Využívá pro komunikaci protokol GRE (47). EoIP tunel

může fungovat skrze jiný tunel, jako například L2TP nebo PPTP. Ethernet over IP funguje ve výsledku stejně, jako když se spojí pomocí funkce bridge dvě fyzická rozhraní. Umožňuje tak transparentně propojit dva vzdálené směrovače skrze třetí síť nebo pomocí jiného tunelu. Následující příklad zobrazuje možnost propojení dvou koncových bodů pomocí bezdrátové sítě a EoIP tunelu. Jedná se pouze o příklad, toto schéma zapojení by šlo řešit i jinými způsoby viz kapitola 4.3 WDS.




**Obrázek 13 – Schéma zapojení sítě za použití EoIP tunelu**

*Zdroj: Převzato z [17]*

**AP:**
```
/interface eoip add name="vzdaleny" tunnel-id=18 remote-address=10.0.0.2
/interface bridge port add bridge=LAN interface=vzdaleny
```

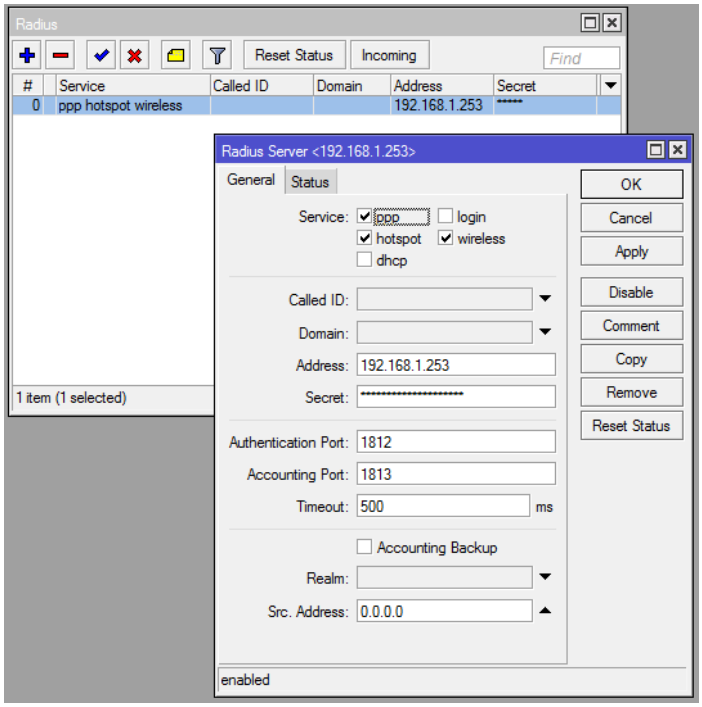
**Station:**
```
/interface eoip add name="hlavni" tunnel-id=18 remote-address=10.0.0.1
/interface bridge port add bridge=LAN interface=hlavni
```

Předpokladem pro funkčnost EoIP tunelu je správné nastavení tunel-id, které se musí na obou koncích tunelu shodovat. Další podmínkou nezbytnou pro funkčnost je, že zařízení na sebe musí tzv. „vidět“. To znamená, že je možno z jednoho na druhé otestovat spojení například pomocí příkazu Ping. Příkazy výše nastavují tunel samotný a virtuální rozhraní přidávají do již existujícího bridge LAN, aby byla komunikace mezi zařízeními plně transparentní.

## 7.5 RADIUS

Jedná se o zkratku z angličtiny - Remote Authentication Dial-In User Service, v překladu: Služba pro vzdálené ověření uživatele. Funkcí tohoto systému je vzdálené ověření uživatele vůči centrální databázi, například databázi uživatelů z doménového serveru Active Directory

Služba v systému RouterOS umožňuje ověřování uživatelů například pro protokoly: PPP, PPPoE, PPTP, L2TP a službu HotSpot. RADISUS server jo kontaktován pouze tehdy, pokud není stejnojmenný uživatel nalezen v lokální databázi směrovače. Nastavení je velice jednoduché. Na obrázku níže je vidět základní nastavení. Na kartě Status je možné kontrolovat funkčnost služby a komunikaci s RADIUS serverem.




**Obrázek 14 – Konfigurace RADIUS připojení**

*Zdroj: Vlastní*

# 8 NÁSTROJE

## 8.1 Ping

Příkaz ping slouží k otestování dostupnosti a odezvy daného zařízení. Využívá protokol ICMP. Hlavním vstupním parametrem je cílová IP adresa. Možno je též nastavit zdrojovou IP adresu, čímž se dá simulovat ping z jiného rozhraní a testovat funkčnost směrování. Dále je možné nastavit například velikost paketu, čímž můžeme do jisté míry testovat propustnost sítě a odezvu.

```
/ping 192.168.64.1 src-address=10.0.0.2
```

## 8.2 TraceRoute

Nástroj TraceRoute funguje obdobně jako například příkaz tracert ve Windows. Vstupním parametrem je IP adresa cílového zařízení. Výstupem je seznam průchozích směrovačů a průběžná statistika odezvy na každém zařízení. Tento nástroj je shledán velice užitečným zejména v případě nastavování směrování a testování průchodnosti sítě.

```
/tool traceroute seznam.cz
```

## 8.3 IP scan

IP scan je velice užitečný nástroj v případě, že hledáte zařízení na síti a neznáte jeho IP adresu. Tento problém by se dal řešit softwarem pro PC. Mikrotik RouterOS však obsahuje nástroj, který to zvládne. Vstupními parametry tohoto nástroje jsou: rozsah IP adres a rozhraní, které se má pro skenování použít. Po vyplnění a spuštění vyhledávání stačí jen vyčkat.

```
/tool ip-scan address-range=192.168.1.1-192.168.1.254 interface=LAN
```

```
[admin@Firewall] > /tool ip-scan address-range=192.168.1.1-192.168.1.254 interface=LAN
Flags: D - dhcp
  ADDRESS         MAC-ADDRESS        TIME DNS                SNMP                NETBIOS
  192.168.1.1     D8:58:D7:00:11:B5   0ms
  192.168.1.254                       0ms
  192.168.1.40    0C:89:10:5A:CF:50   1ms
  192.168.1.18    00:13:3B:53:06:73   0ms
  192.168.1.26    F0:4F:7C:32:83:73
  192.168.1.249                       2ms
  192.168.1.250   C8:CB:B8:62:7A:2A   4ms
  192.168.1.252                       6ms
  192.168.1.253   00:11:32:24:34:16   6ms
-- [Q quit|D dump|C-z pause]
```

**Obrázek 15 - Výstup z příkazu ip-scan**

*Zdroj: Vlastní*

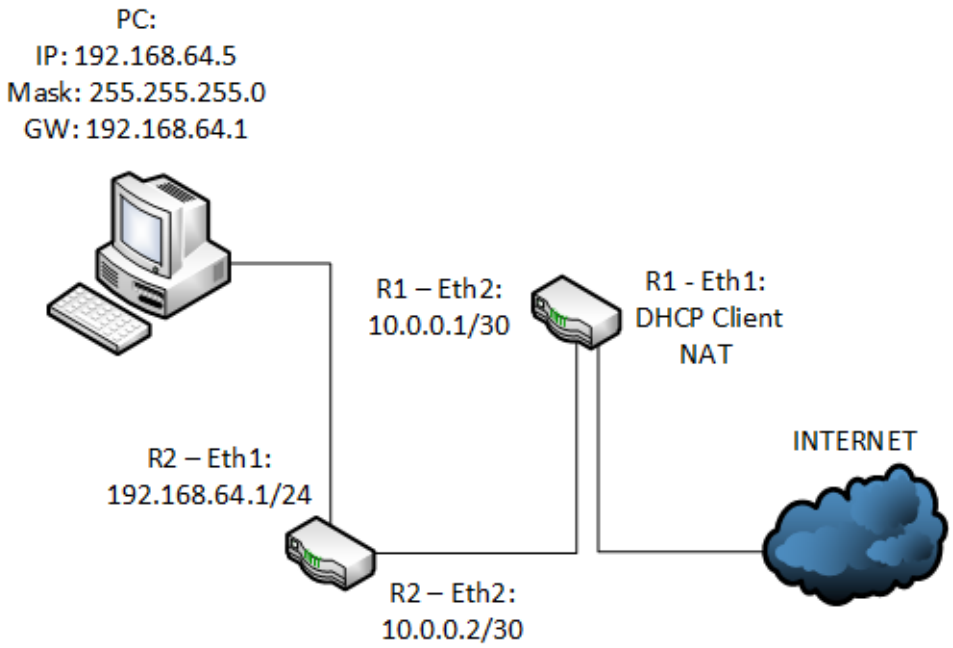
# 9 PRAKTICKÉ ÚLOHY

Všechny úlohy jsou realizovány na RouterBoardech hAP ac lite (RB952Ui-5ac2nD). Použit byl RouterOS ve verzi 6.35.1. Před začátkem nastavování každé úlohy byly zařízení obnoveny do výchozího nastavení bez úvodní konfigurace, to znamená, že zařízení jsou čistá, úplně bez jakékoli konfigurace. Pro první konfiguraci je nutné zařízení připojit pomocí utility WinBox na MAC adresu (program sám zařízení na síti vyhledá), jelikož není nakonfigurována žádná IP adresa (0.0.0.0). Výchozí uživatelské jméno je admin a heslo nevyplněno.

```
/system reset-configuration keep-users=no no-defaults=yes
```

## 9.1 Základní nastavení adres a směrování

Úkolem je zapojit a zprovoznit schéma zobrazené níže. Pokud bude konfigurace provedena správně, bude se možné z PC dostat do sítě internet. Pro přehlednost je vhodné nejprve nastavit jméno zařízení. Dalším krokem je nastavení IP adres na jednotlivých rozhraních. DHCP server bude přidělovat IP adresu, masku, NTP server a DNS server. Toto rozhraní bude výchozí branou do internetu.




**Obrázek 16 – Schéma č. 1**

*Zdroj: Vlastní*

**R1:**
```
/system identity set name=R1
/ip dhcp-client add interface=ether1 use-peer-dns=yes use-peer-ntp=yes add-
default-route=yes disabled=no

/ip address add interface=ether2 disabled=no address=10.0.0.1/30
```

**R2:**
```
/system identity set name=R2
/ip address add interface=ether1 disabled=no address=192.168.64.1/24
/ip address add interface=ether2 disabled=no address=10.0.0.2/30
```

V tomto stavu by mělo být možné se z R1 dostat do internetu a oba porty eth2 by spolu měly komunikovat (odpovídat na ping). Je žádoucí aby R1 fungoval jako DNS server, proto je nutné funkci povolit. Dalším krokem je přidání cesty do sítě 192.168.64.0/24 na R1. Na R2 je třeba nastavit server pro překlad adres a přidat výchozí bránu směrovanou na rozhraní eth2 směrovače R1. V tuto chvíli by mělo být možné se dostat z PC na IP 10.0.0.1 za použití pingu. Též by měla fungovat komunikace DNS.

**R1:**
```
/ip dns set allow-remote-requests=yes
/ip route add dst-address=192.168.64.0/24 gateway=10.0.0.2
```

**R2:**
```
/ip dns set servers=10.0.0.1
/ip dns set allow-remote-requests=yes
/ip route add dst-address=0.0.0.0/0 gateway=10.0.0.1
```

Do internetu se však stále není z PC možné dostat, jelikož směrovače za R1 směrem do internetu neznají síť 192.168.64.1/24. Znají pouze přímo připojenou síť na směrovači R1 rozhraní eth1. Proto je nutné veškerou odchozí komunikaci ze sítě za toto rozhraní maskovat. Překlad adres (NAT) zajistí následující příkaz. Odchozí komunikace pak bude mít zdrojovou IP adresu stejnou jako je IP adresa rozhraní eth1 na R1.

**R1:**
```
/ip firewall nat add chain=srcnat out-interface=ether1 action=masquerade
```

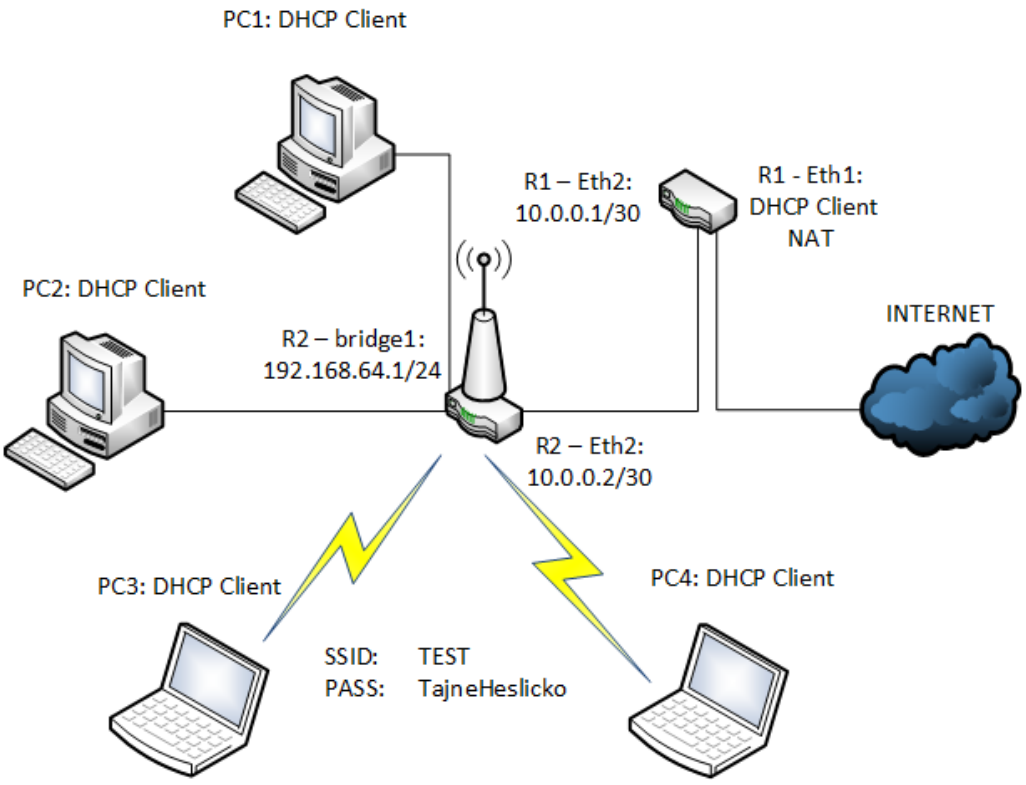
## 9.2 Rozšířená konfigurace

Tento příklad bude vycházet z kapitoly 9.1 a bude stávající konfiguraci rozšiřovat. Požadavkem je, aby porty eth3, eth4, eth5 a bezdrátová síť byly propojeny bridgem. Adresy budou klientským stanicím přiřazovány DHCP serverem z rozsahu 192.168.64.100-250. Pokud nebude IP adresa přiřazena DHCP serverem, nebude umožněno zařízení v rámci sítě komunikovat. SSID bezdrátové sítě bude nastaveno na TEST a heslo s použitím šifrování WPA2 bude nastaveno na „TajneHeslicko“. Veškeré změny konfigurace se budou odehrávat pouze na R2.

Prvním krokem bude vytvoření bridge. V druhém kroku nastavíme IP adresu na bridge a přidáme do bridge jednotlivé porty. Aby bylo možné změnit rozhraní již definované IP adresy, musíme znát její číslo. Jak je vidět na obrázku níže, IP adresa 192.168.64.1/24 je definována pod číslem 0. Změníme rozhraní na bridge1.

```
[admin@R2] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   192.168.64.1/24    192.168.64.0    ether1
 1   10.0.0.2/30        10.0.0.0        ether2
[admin@R2] >
```

**Obrázek 17 – Výpis IP adres**




**Obrázek 18 – Schéma č. 2**

*Zdroj: Vlastní*

```
/interface bridge add name=bridge
/int bridge port add bridge=bridge1 interface=ether3
/int bridge port add bridge=bridge1 interface=ether4
/int bridge port add bridge=bridge1 interface=ether5
/ip address set interface=bridge1 numbers=0
```

V tuto chvíli je jedno, do kterého z nastavených portů PC se statickou IP adresou připojíme, funkčnost bude stejná.

```
/ip pool add name=pool1 ranges=192.168.64.100-192.168.64.250
/ip dhcp-server network add address=192.168.64.0/24 dns-server=192.168.64.1
gateway=192.168.64.1

/ip dhcp-server add name=DHCP address-pool=pool1 interface=bridge1 lease-
time=1d add-arp=yes disabled=no
```

Příkazy výše definují rozsah IP adres, ze kterého bude DHCP server přiřazovat a síť, která má být obsluhována DHCP serverem. Posledním příkazem se vytvoří samotná instance DHCP serveru na rozhraní bridge1. Zbývá nastavení bezdrátové sítě a zabezpečení. Opět dochází k editaci již vytvořeného záznamu, tudíž je nutné, nejprve zjisti jeho číslo.

```
[admin@R2] > interface wireless print
Flags: X - disabled, R - running
 0 X  name="wlan1" mtu=1500 l2mtu=1600 mac-address=E4:8D:8C:CD:D0:BC arp=enabled
      interface-type=Atheros AR9300 mode=station ssid="MikroTik" frequency=2412
      band=2ghz-b/g channel-width=20mhz scan-list=default wireless-protocol=any
      vlan-mode=no-tag vlan-id=1 wds-mode=disabled wds-default-bridge=none
      wds-ignore-ssid=no bridge-mode=enabled default-authentication=yes
      default-forwarding=yes default-ap-tx-limit=0 default-client-tx-limit=0
      hide-ssid=no security-profile=default compression=no

 1 X  name="wlan2" mtu=1500 l2mtu=1600 mac-address=E4:8D:8C:CD:D0:BB arp=enabled
      interface-type=Atheros AR9888 mode=station ssid="MikroTik" frequency=5180
      band=5ghz-a channel-width=20mhz scan-list=default wireless-protocol=any
      vlan-mode=no-tag vlan-id=1 wds-mode=disabled wds-default-bridge=none
      wds-ignore-ssid=no bridge-mode=enabled default-authentication=yes
      default-forwarding=yes default-ap-tx-limit=0 default-client-tx-limit=0
      hide-ssid=no security-profile=default compression=no
[admin@R2] >
```

**Obrázek 19 – Výpis bezdrátových rozhraní**

*Zdroj: Vlastní*

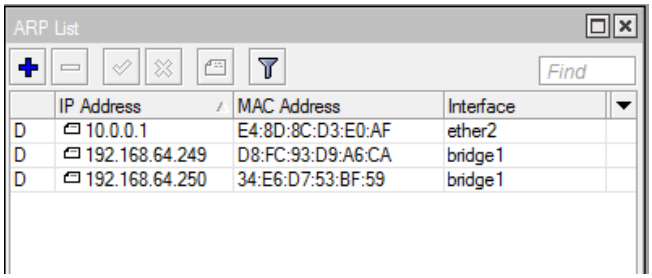
Pracovat budeme s bezdrátovou kartou na frekvenci 2,4Ghz, tudíž použijeme ID 0. Nejprve vytvoříme profil zabezpečení, který se použije při nastavení samotného rozhraní. Poté nastavíme rozhraní na požadované parametry. K bezdrátové síti se dá již připojit, ale jelikož nemá nastavenou IP adresu, není možno komunikovat v síti. Přidáme proto rozhraní do bridge1. Po přidání do bridge1 se začíná bezdrátová síť chovat stejně, jako pevná. Připojený klient dostává IP adresu z DHCP serveru a může komunikovat do sítě internet.

```
/interface wireless security-profiles add name=TEST authentication-
types=wpa2-psk,wpa-psk unicast-ciphers=aes-ccm group-ciphers=aes-ccm wpa-
pre-shared-key=TajneHeslicko wpa2-pre-shared-key=TajneHeslicko
mode=dynamic-keys

/interface wireless set numbers=0 mode=ap-bridge band=2ghz-b/g/n channel-
width=20/40mhz-Ce frequency=auto ssid=TEST wireless-protocol=802.11
security-profile=TEST disabled=no

/interface bridge port add bridge=bridge1 interface=wlan1
```

Zbývá nastavit zabezpečení DHCP serveru. Je nutné, aby bylo na instanci DHPC nastaveno add-arp=yes, jinak by nemohl komunikovat z daného rozhraní nebo bridge nikdo.

```
/interface bridge set numbers=0 arp=reply-only
```

Po potvrzení tohoto příkazu směrovač přestane automaticky přidávat záznamy do ARP tabulky a učit se tak adresy, které nebyly přiřazeny DHCP serverem.




**Obrázek 20 – Výpis ARP tabulky**

*Zdroj: Vlastní*

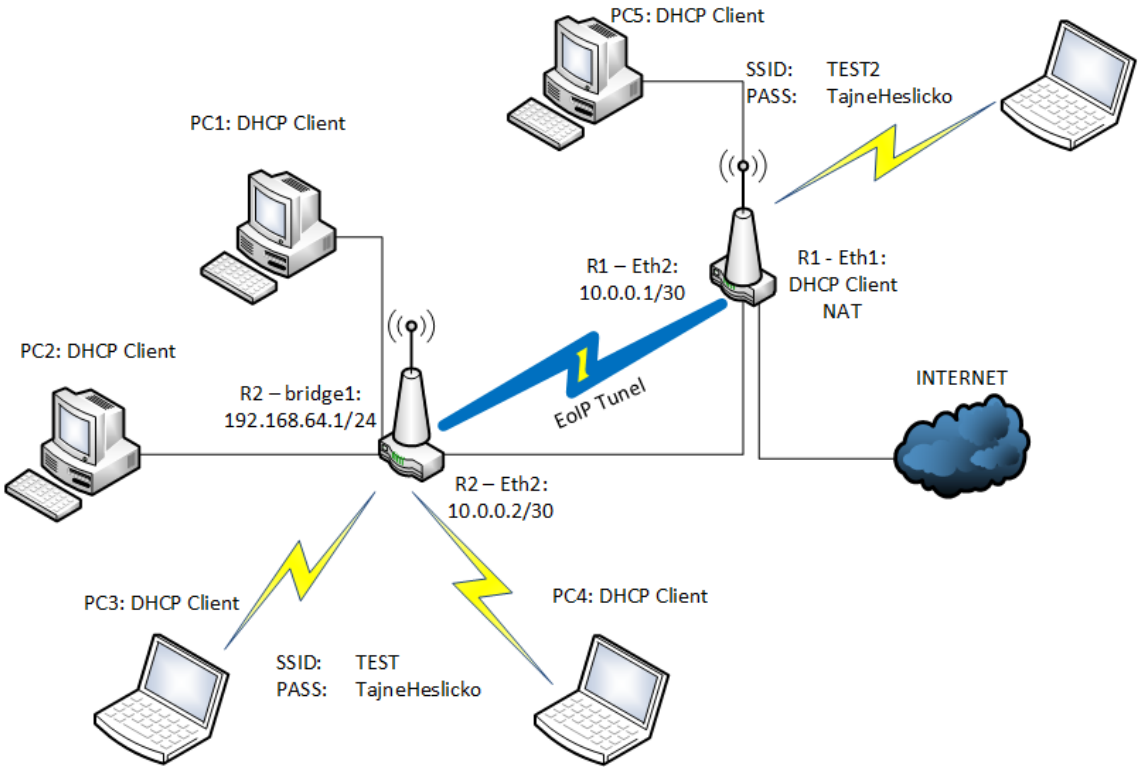
## 9.3 Rozšíření o EoIP tunel

Posledním požadavkem na rozšíření předchozích konfigurací je vytvoření tunelu mezi R1 a R2, který vytvoří transparentní spoj pro nově vytvořenou síť na R1. Nová bezdrátová síť na R1 se bude jmenovat TEST2 a heslo bude stejné jako v případě bezdrátové sítě na R2. Klientům připojeným na směrovači R1 bude umožněno komunikovat pouze v rámci lokální sítě, provoz do internetu a ostatních sítí bude omezen.

Prvním krokem je vytvoření bridge na R1, přidání rozhraní do bridge a zprovoznění bezdrátové sítě. V tomto případě je postup velice podobný jako v minulém případě.

**R2:**

```
/interface bridge add name=bridge1
/int bridge port add bridge=bridge1 interface=ether3
/int bridge port add bridge=bridge1 interface=ether4
/int bridge port add bridge=bridge1 interface=ether5

/interface wireless security-profiles add name=TEST2 authentication-
types=wpa2-psk,wpa-psk unicast-ciphers=aes-ccm group-ciphers=aes-ccm wpa-
pre-shared-key=TajneHeslicko wpa2-pre-shared-key=TajneHeslicko
mode=dynamic-keys

/interface wireless set numbers=0 mode=ap-bridge band=2ghz-b/g/n channel-
width=20/40mhz-Ce frequency=auto ssid=TEST2 wireless-protocol=802.11
security-profile=TEST2 disabled=no

/interface bridge port add bridge=bridge1 interface=wlan1
```




**Obrázek 21 – Schéma č. 3**

*Zdroj: Vlastní*

Nyní přejdeme k vytvoření samotného tunelu. Tunel se vytváří mezi dvěma koncovými body, což je v tomto případě R1 a R2. Tunel se musí po vytvoření přidat do bridge na obou zařízeních.

**R1 :**

```
/interface eoip add name=tunel1 remote-address=10.0.0.2 tunnel-id=64 local-
address=10.0.0.1

/int bridge port add bridge=bridge1 interface=tunel1
```

**R2 :**

```
/interface eoip add name=tunel1 remote-address=10.0.0.1 tunnel-id=64 local-
address=10.0.0.2

/int bridge port add bridge=bridge1 interface=tunel1
```

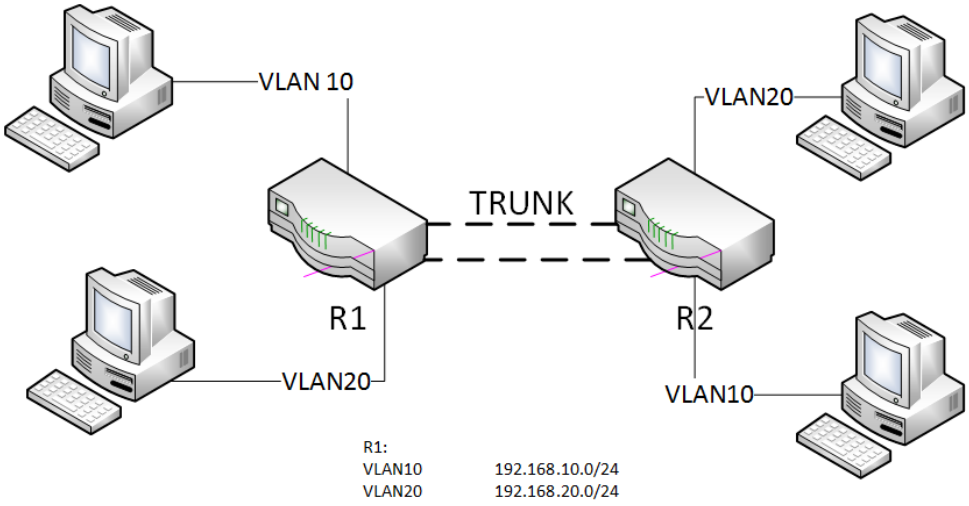
Nyní je síť na obou přístupových bodech propojena a funkční. Posledním požadavkem je omezení provozu klientů připojených k R1 do jiné sítě než 192.168.64.0/24. Toho dosáhneme použitím následujících pravidel. Prvním povolíme, aby firewall zasahoval i do lokálního provozu přes bridge. Druhé pravidlo říká, že veškerý provoz odcházející tunelem bude označen značkou ZAKAZ. Poté je nutné v pravidlech firewallu nadefinovat sítě, které budou mít provoz přes tunel povolený (v tomto případě jen 192.168.64.0/24). Dále platí, že pokud provoz nebyl zachycen prvním pravidlem, pokračuje na další, které říká, že veškeré pakety označené jako ZAKAZ budou zahozeny.

**R1:**

```
/interface bridge settings set use-ip-firewall=yes
/interface bridge filter add action=mark-packet chain=forward new-packet-
mark=ZAKAZ out-interface=tunel1

/ip firewall filter add chain=forward dst-address=192.168.64.0/24
/ip firewall filter add action=drop chain=forward packet-mark=ZAKAZ
```

## 9.4 VLAN

Tento příklad popíše základní použití a nastavení VLAN. R1 je zvolen za core router, to znamená, že bude přepínat komunikaci mezi jednotlivými VLAN. Trunk je vytvořen pomocí bridge.




**Obrázek 22 – Schéma č. 4**

*Zdroj: Vlastní*

Prvním krokem je vytvoření trunku mezi směrovači na portech eth4 a eth5 symetricky. Port eth3 bude použit pro VLAN 10 a eth4 pro VLAN 20. Na směrovači R1 jsou nastaveny IP adresy jednotlivých VLAN, které jsou použity jako výchozí brány sítí. Konfigurace R2 je v tomto případě zrcadlově stejná. Jediný rozdíl je ve jménu směrovače a v nastavení IP adres.

**R1:**

```
/system identity set name=R1
/interface bridge add name=bridge10
/interface bridge add name=bridge20
/interface bridge add name=trunk
/interface bridge port add bridge=trunk interface=ether5
/interface bridge port add bridge=trunk interface=ether4
/interface vlan add interface=trunk name=VLAN10 vlan-id=10
/interface vlan add interface=trunk name=VLAN20 vlan-id=20
/interface bridge port add bridge=bridge10 interface=VLAN10
/interface bridge port add bridge=bridge20 interface=VLAN20
/interface bridge port add bridge=bridge10 interface=ether3
/interface bridge port add bridge=bridge20 interface=ether2
```

```
/ip address add address=192.168.10.1/24 interface=VLAN10
network=192.168.10.0
/ip address add address=192.168.20.1/24 interface=VLAN20
network=192.168.20.0
```

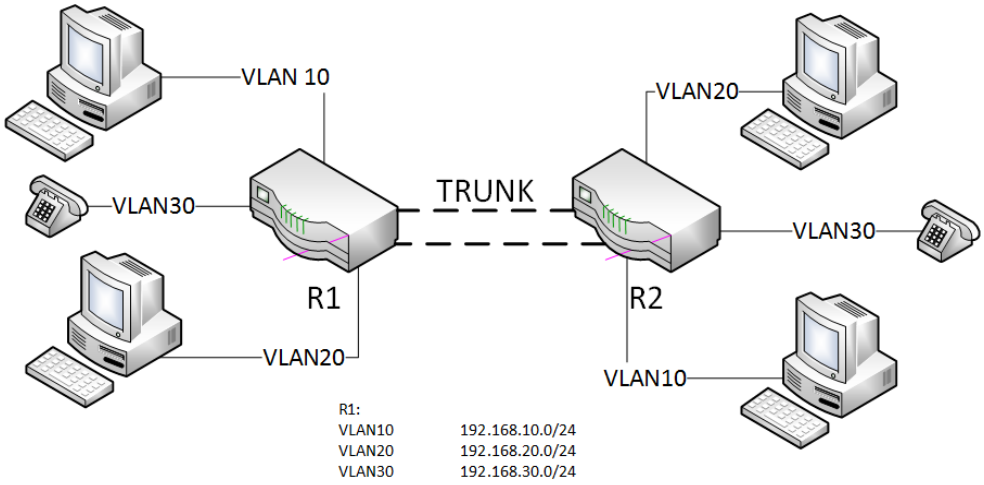
**R2:**

```
/system identity set name=R2
/interface bridge add name=bridge10
/interface bridge add name=bridge20
/interface bridge add name=trunk
/interface bridge port add bridge=trunk interface=ether5
/interface bridge port add bridge=trunk interface=ether4
/interface vlan add interface=trunk name=VLAN10 vlan-id=10
/interface vlan add interface=trunk name=VLAN20 vlan-id=20
/interface bridge port add bridge=bridge10 interface=VLAN10
/interface bridge port add bridge=bridge20 interface=VLAN20
/interface bridge port add bridge=bridge10 interface=ether3
/interface bridge port add bridge=bridge20 interface=ether2
```

Při správném zapojení a nastavením dle ukázky bude možné komunikovat v rámci VLAN a i mezi jednotlivými VLAN.

## 9.5 VLAN s firewallem

Tato úloha vychází z předchozího příkladu (9.4 VLAN) a rozšiřuje jej. Ve schématu přibyla VLAN30 určená pro IP telefony. Z VLAN30 nebude možné komunikovat s ostatními VLAN. To znamená, že se z adres192.168.30.0/24 nebude možné spojit se zařízením na síti 192.168.10.0/24 nebi 192.168.20.0/24. Dalším požadavkem je přidělování adres DHPC servere pro jednotlivé VLAN.




**Obrázek 23 – Schéma č. 5**

*Zdroj: Vlastní*

Vytvoříme VLAN30, bridge30, přidáme VLAN30 na tunk a port eth1. V této konfiguraci jsou VLAN funkční a komunikují i mezi sebou. Nastavíme příslušné rozsahy adres pro DHCP server, vytvoříme instance a sítě.

**R1:**
```
/interface bridge add name=bridge30
/interface vlan add interface=trunk name=VLAN30 vlan-id=30
/interface bridge port add bridge=bridge30 interface=VLAN30
/interface bridge port add bridge=bridge30 interface=ether1
/ip address add address=192.168.30.1/24 interface=VLAN30
network=192.168.10.0

/ip pool add name=vlan10 ranges=192.168.10.100-192.168.10.200
/ip pool add name=vlan20 ranges=192.168.20.100-192.168.20.200
/ip pool add name=vlan30 ranges=192.168.30.100-192.168.30.200

/ip dhcp-server add add-arp=yes address-pool=vlan10 disabled=no
interface=bridge10 lease-time=1d name=vlan10
/ip dhcp-server add add-arp=yes address-pool=vlan20 disabled=no
interface=bridge20 lease-time=1d name=vlan20
/ip dhcp-server add add-arp=yes address-pool=vlan30 disabled=no
interface=bridge30 lease-time=1d name=vlan30

/ip dhcp-server network add address=192.168.10.0/24 dns-server=192.168.10.1
gateway=192.168.10.1
/ip dhcp-server network add address=192.168.20.0/24 dns-server=192.168.20.1
gateway=192.168.20.1
/ip dhcp-server network add address=192.168.30.0/24 dns-server=192.168.30.1
gateway=192.168.30.1
```

**R2:**
```
/interface bridge add name=bridge30
/interface vlan add interface=trunk name=VLAN30 vlan-id=30
/interface bridge port add bridge=bridge30 interface=VLAN30
/interface bridge port add bridge=bridge30 interface=ether1
```

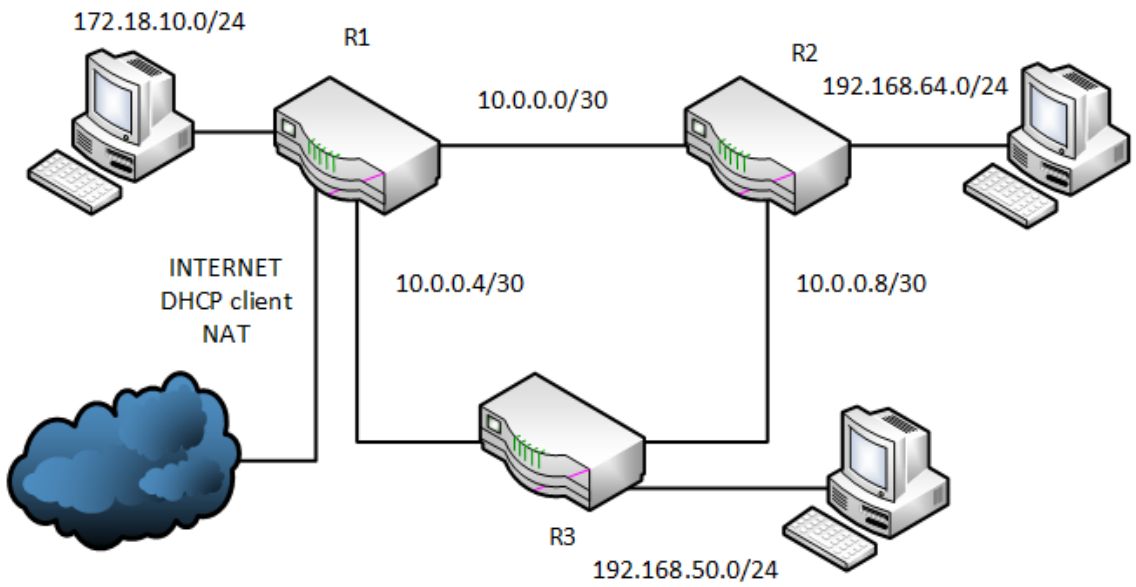
Posledním krokem je odfiltrování sítě 192.168.30.0/24 od všech ostatních. Toho docílíme jednoduchými pravidly na firewallu.

**R1:**
```
/ip firewall filter add action=drop chain=forward dst-
address=192.168.30.0/24
/ip firewall filter add action=drop chain=forward src-
address=192.168.30.0/24
```

## 9.6 OSPF

Cílem této úlohy je realizovat schéma zapojení třech směrovačů a jejich nastavení, zejména pak směrování pomocí protokolu OSPF. Po nastavení IP adres můžeme rovnou přejít k nastavení samotného OSPF. Instance OSPF default a backbone area jsou vytvořeny automaticky. Zbývá na každém směrovači doplnit výčet přímo připojených sítí. Na R1 je nutné nastavit NAT na

WAN portu. Jelikož cesta do internetu vede přes směrovač R1, je žádoucí, aby cestu do internetu šířil dále do sítě.




**Obrázek 24 – Schéma č. 6**

*Zdroj: Vlastní*

**R1:**
```
/ip dhcp-client add interface=ether5 use-peer-dns=yes use-peer-ntp=yes add-
default-route=yes disabled=no
/ip addres add address=172.18.10.1/24 interface=ether4 network=172.18.10.0
/ip address add address=10.0.0.1/30 interface=ether1 network=10.0.0.0
/ip address add address=10.0.0.5/30 interface=ether2 network=10.0.0.4
/routing ospf network add area=backbone network=10.0.0.0/30
/routing ospf network add area=backbone network=172.18.10.0/24
/routing ospf network add area=backbone network=10.0.0.4/30
/routing ospf instance set numbers=0 distribute-default=always-as-type-1
redistribute-bgp=as-type-1 redistribute-connected=as-type-1 redistribute-
other-ospf=as-type-1 redistribute-rip=as-type-1 redistribute-static=as-
type-1
/ip firewall nat add action=masquerade chain=srcnat out-interface=ether5
```

**R2:**
```
/ip addres add address=192.168.64.0/24 interface=ether4
network=192.168.64.0
/ip address add address=10.0.0.2/30 interface=ether1 network=10.0.0.0
/ip address add address=10.0.0.9/30 interface=ether2 network=10.0.0.8
/routing ospf network add area=backbone network=10.0.0.8/30
/routing ospf network add area=backbone network=192.168.64.0/24
/routing ospf network add area=backbone network=10.0.0.4/30
/routing ospf instance set numbers=0 redistribute-bgp=as-type-1
redistribute-connected=as-type-1 redistribute-other-ospf=as-type-1
redistribute-rip=as-type-1 redistribute-static=as-type-1
```

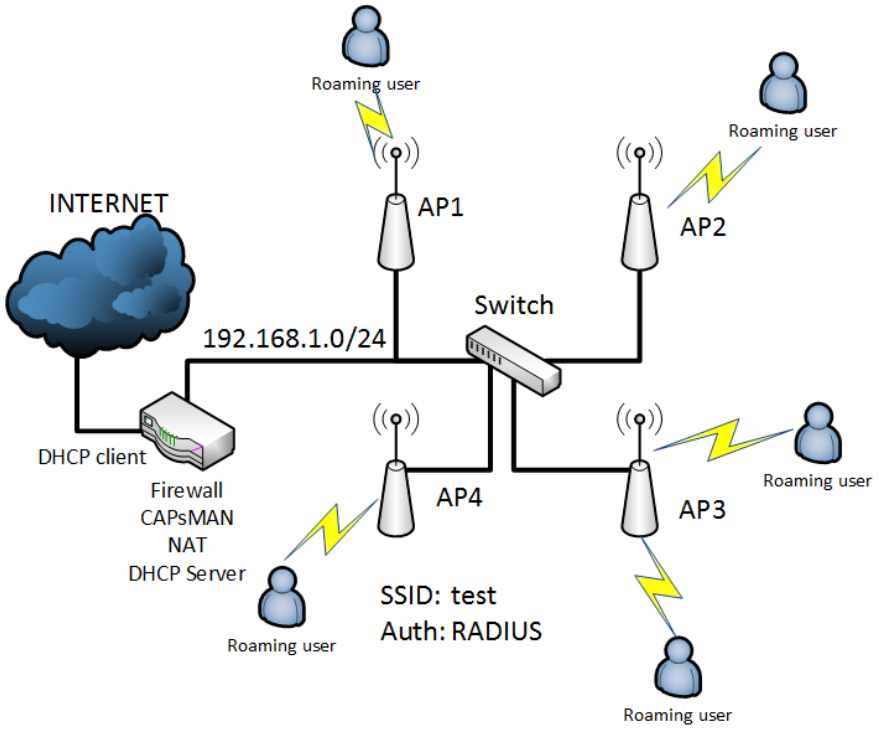
**R3:**
```
/ip address add address=10.0.0.6/30 interface=ether1 network=10.0.0.4
/ip address add address=10.0.0.10/30 interface=ether2 network=10.0.0.8
/routing ospf network add area=backbone network=10.0.0.8/30
```

```
/routing ospf network add area=backbone network=10.0.0.4/30
/routing ospf instance set numbers=0 redistribute-bgp=as-type-1
redistribute-connected=as-type-1 redistribute-other-ospf=as-type-1
redistribute-rip=as-type-1 redistribute-static=as-type-1
```

Výsledkem by měla být komplexní síť, která si automaticky předává informace o nově připojených sítích. Jelikož nejsou specifikována žádná přístupová pravidla, mělo by být možné se připojit z jedné sítě do druhé. Nutností je samozřejmě správné nastavení výchozích bran na klientských zařízeních.

## 9.7 CAPsMAN a centrální řízení přístupových bodů

CAPsMAN funguje na principu klient – server. Na hlavním směrovači je spuštěn a jednotlivé přístupové body jsou zkonfigurovány tak, aby se připojovali a přebírali konfiguraci z hlavního směrovače. Přístupové body pak tvoří síť umožňující roaming jednotlivých uživatelů. Uživatelé jsou automaticky přepínání mezi jednotlivými přístupovými body.




**Obrázek 25 – Schéma č. 7**

*Zdroj: Vlastní*

Nejpodstatnější je konfigurace hlavního směrovače. Dle předchozích úloh nastavíme na eth1 DHCP klienta, vytvoříme pravidlo pro překlad adres do internetu (rozhraní eth1). Dalším krokem bude vytvoření bridge pro LAN, nastavení IP na bridge, přidání portů do bridge a nastavení DHCP serveru. Po této konfiguraci by měl být eth1 připojen do internetu a ostatní porty připojeny do bridge. Otestujte funkčnost konfigurace – přístup z LAN do internetu,

funkčnost překladu adres a DHCP serveru. Pokud je toto základní nastavení funkční, přistoupíme ke konfiguraci CAPsMAN.

Prvním krokem je definování bezpečnostních politik pro bezdrátové sítě. V tomto případě vytváříme dvě. První s ověřením přes RADIUS, druhou pak pro testovací účely. Dalším příkazem vytvoříme šablonu s parametry bezdrátové sítě. Je žádoucí, aby se nově nalezené přístupové body automaticky zkonfigurovaly bez nutnosti zásahu, o což se stará následující příkaz. Zbývá pouze nastavení mostu do lokální sítě a aktivace CAPsMAN.

**Firewall:**

```
/caps-man security add authentication-types=wpa2-eap eap-
methods=passthrough eap-radius-accounting=no encryption=aes-ccm group-
encryption=aes-ccm name=test tls-mode=dont-verify-certificate

/caps-man security add authentication-types=wpa-psk,wpa2-psk
encryption=aes-ccm group-encryption=aes-ccm name=heslo
passphrase=TajneHeslicko

/caps-man configuration add country="czech republic" datapath=test hide-
ssid=no mode=ap name=test security=heslo ssid=test

/caps-man provisioning add action=create-dynamic-enabled master-
configuration=test

/caps-man datapath add bridge=LAN client-to-client-forwarding=yes name=test
/caps-man manager set ca-certificate=auto certificate=auto enabled=yes
```

Nastavení jednotlivých přístupových bodů je v tomto případě stejné. Postačí vytvořit bridge pro bezdrátové rozhraní a eth1, eth1 nastavit jako DHCP klienta a povolit CAP na rozhraní wlan1. Jako rozhraní pro spojení s CAPsMAN se nastaví eth1, který je dle schématu připojen k hlavnímu směrovači.

**AP1,AP2, …, APn:**

```
/interface bridge add name=LAN
/interface bridge port add bridge=LAN interface=ether1
/interface bridge port add bridge=LAN interface=wlan1
/ip dhcp-client add default-route-distance=0 dhcp-options=hostname,clientid
disabled=no interface=ether1
/interface wireless cap set bridge=LAN certificate=request discovery-
interfaces=ether1 enabled=yes interfaces=wlan1
```

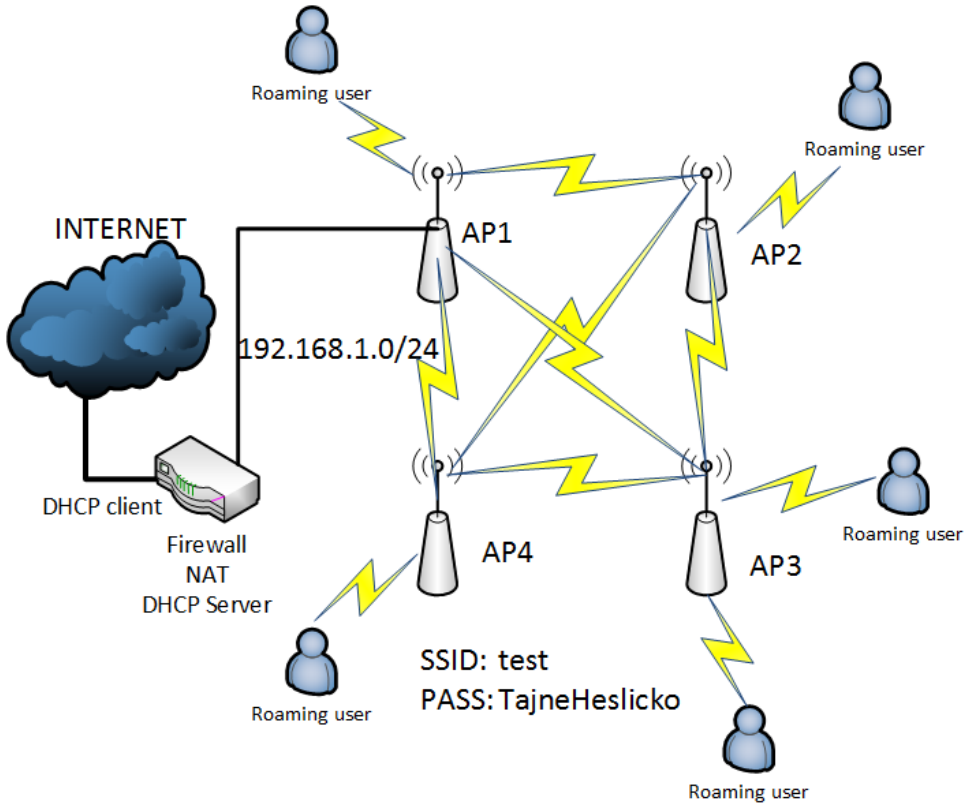
Pokud vše funguje jak má, změňte profil zabezpečení a otestujte ověření pomocí RADIUS serveru.

**Firewall:**

```
/radius add address=192.168.1.2 secret=Heslo service=wireless timeout=500ms
/caps-man configuration set numbers=0 security=test
```

## 9.8 WDS Mesh

Cílem této úlohy je vytvoření systému bezdrátových přístupových bodů, které budou bezdrátově také propojeny. Jednotlivé přístupové body jsou nastaveny v režimu AP bridge a umožňují klientům se připojit. Mezi sebou jsou propojeny dynamickým WDS. Je hned několik variant realizace. V tomto případě je schéma realizováno pomocí hvězdy (mesh). Firewall je nastaven, jako brána do sítě internet viz úloha 9.7 CAPsMAN a centrální řízení přístupových bodů.




**Obrázek 26 – Schéma č. 8**

*Zdroj: Vlastní*

V tomto případě je postup na všech přístupových bodech stejný. V prvním kroku se vytvoří mesh. V kroku druhém definujeme profil zabezpečení bezdrátové sítě a dále pak nastavujeme bezdrátové rozhraní. Oproti minulým příkladům je zde rozdíl v nastavení WDS. Je důležité do již vytvořené meshe přidat rozhraní bezdrátové sítě, aby se později mohli klienti připojit a komunikovat. Rozhraní mesh nastavíme jako DHCP klienta.

**AP1, AP2, …, APn:**

```
/interface mesh add name=mesh
/interface wireless security-profiles add authentication-types=wpa-
psk,wpa2-psk mode=dynamic-keys name=test wpa-pre-shared-key=TajneHeslicko
wpa2-pre-shared-key=TajneHeslicko
```

```
/interface wireless set [ find default-name=wlan1 ] band=2ghz-b/g/n
disabled=no mode=ap-bridge security-profile=test ssid=test wds-default-
bridge=mesh wds-mode=dynamic-mesh wireless-protocol=802.11

/interface mesh port add interface=ether1 mesh=wlan1

/ip dhcp-client add default-route-distance=0 dhcp-options=hostname,clientid
disabled=no interface=mesh
```

V posledním kroku je třeba povolit na AP1 komunikaci s DHCP serverem a výchozí bránou této sítě. Zbývá pouze otestovat funkčnost roamingu mezi jednotlivými přístupovými body.

**AP1:**

```
/interface mesh port add interface=ether1 mesh=mesh
```

# 10 ZÁVĚR

Tato bakalářská práce se zaměřuje na možnosti využití zařízení s operačním systémem firmy Mikrotik RouterOS. Nejprve bylo popsáno hardwarové vybavení, na kterém byla celá práce realizována. Po představení Hardware byl představen též software nezbytný pro realizaci. Firma Mikrotik je velice perspektivní a její zařízení mají vysokou uplatnitelnost a obrovskou výhodu v poměru cena a výkon. Nastavení je složitější než u běžných směrovačů, ale nabízí s nimi nesrovnatelné možnosti. Po nějaké době práce s těmito zařízeními se stává ovládání velice přehledným a intuitivním. Další nespornou výhodou je komfort při nastavování systému v prostředí WinBoxu a možnost otevření terminálu přímo v utilitě WinBox.

V této práci byla představena celá řada možností a většina z nich byla popsána na názorných příkladech. Výhody a nevýhody tohoto systému jsou porovnány s ostatními operačními systémy, dnes běžně provozovanými. Z tohoto srovnání plyne, že nasazení RouterOS je opodstatněné v případě menších podnikových sítí, sítí s velkými nároky a specifickými požadavky na provoz a u náročných domácích uživatelů. Domácím uživatelům nastavení ulehčí přehledný průvodce, který je schopen během pár kroků nastavit směrovač do provozuschopného stavu.

Praktická část obsahuje již konkrétní příklady, na kterých jsou demonstrovány možnosti a funkčnost operačního systému RouterOS. Největší pozornost je věnována nejběžněji používaným a tudíž nejdůležitějším typům nastavení jako je například přiřazení adres na jednotlivá rozhraní, DHCP, vytvoření mostu mezi jednotlivými porty, propojení sítí pomocí VPN, VLAN a tunelů. Dále stojí za zmínku nastavení bezdrátového rozhraní, možnosti směrování, ať už statické nebo dynamické a překlady adres.

Při realizaci této práce byl velkou komplikací nedostatek relevantních zdrojů. Jelikož se nejedná o tolik uživatelsky rozšířenou platformu a neexistuje dostatečně rozsáhlý manuál, bylo nutné hodně improvizovat a nastavení reálně odzkoušet.

# 11 POUŽITÁ LITERATURA


[1] About us. *Routers & wireless: MikroTik* [online]. Latvia: MikroTik, 2014 [cit. 2016-09-01]. Dostupné z: www.mikrotik.com/aboutus

[2] STRANÍK, JAN. *Systém směrování na více bran pomocí směrovače Mikrotik*. BRNO, 2012. Diplomová. Vedoucí práce Ing. Mojmír Jelínek.

[3] RouterBOARD.sk. *RouterBOARD.sk: Vaše platforma pro routery a bezdrátové sítě* [online]. RouterBOARD.sk, 2016 [cit. 2016-09-01]. Dostupné z: RouterBOARD.sk

[4] Manual:License: MikroTik Wiki. *Manuál RouterOS* [online]. [cit. 2016-09-01]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:License

[5] GRYGÁREK, Petr. *Směrované a přepínané sítě* [online]. Ostrava, 2014 [cit. 2016-09-01]. Dostupné z: http://wh.cs.vsb.cz/sps/images/4/46/Mult.pdf

[6] Manual:Nv2: MikroTik Wiki. *Manuál RouterOS* [online]. [cit. 2016-09-01]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:Nv2

[7] Manual:Packet Flow: MikroTik Wiki. *Manuál RouterOS* [online]. [cit. 2016-09-01]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:Packet_Flow

[8] Manual:IP/Firewall/NAT: MikroTik Wiki. *Manuál RouterOS* [online]. [cit. 2016-09-01]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:IP/Firewall/NAT

[9] GRYGÁREK, Petr. *Network Address Translation (NAT)* [online]. Ostrava, 2014 [cit. 2016-09-01]. Dostupné z: http://www.cs.vsb.cz/grygarek/TPS/NAT/NAT.html

[10] Manual:IP/Firewall/NAT: MikroTik Wiki. *Manuál RouterOS* [online]. [cit. 2016-09-01]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:IP/Firewall/NAT#Destination_NAT

[11] VPN s OpenVPN. *Roman Pavlík* [online]. Pavlík [cit. 2016-09-01]. Dostupné z: https://roman-pavlik.cz/vpn-s-openvpn-uvod

[12] OpenVPN: MikroTik Wiki. *Manuál RouterOS* [online]. [cit. 2016-09-01]. Dostupné z: http://wiki.mikrotik.com/wiki/OpenVPN

[13] Manual:Interface/PPTP: MikroTik Wiki. *Manuál RouterOS* [online]. [cit. 2016-09-01]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:Interface/PPTP

[14] Manual:Interface/L2TP: MikroTik Wiki. *Manuál RouterOS* [online]. [cit. 2016-09-01]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:Interface/L2TP