VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

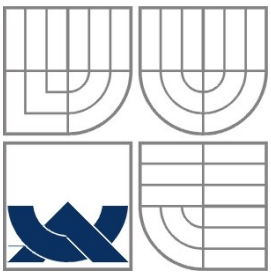
VYUŽITÍ ROUTERBOARDŮ MIKROTIK PRO VÝUKU SÍŤOVÝCH KURZŮ NA FIT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

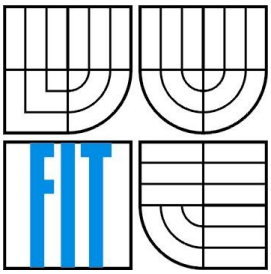
AUTOR PRÁCE
AUTHOR

PATRIK ŠEBEŇ

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

VYUŽITÍ ROUTERBOARDŮ MIKROTIK PRO VÝUKU SÍŤOVÝCH KURZŮ NA FIT

USAGE OF MIKROTIK ROUTERBOARDS FOR EDUCATION IN COMPUTER NETWORKS COURSES AT FIT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PATRIK ŠEBEŇ
AUTHOR

VEDOUCÍ PRÁCE — ING. JAROSLAV RÁB
SUPERVISOR

BRNO 2010

Abstrakt

Bakalářská práce se zabývá využitím routerboardů MikroTik při výuce síťových kurzů. Bylo v ní vytvořeno několik cvičení zaměřených na směrování, směrovací protokoly, běh a zabezpečení bezdrátových síťí. Dále se zabývá výběrem alternativního operačního systému pro směrovač RB433AH a postupem pro jeho nasazení do zařízení.

Abstract

This bachelor thesis analyzes usage of routerboard Mikrotik for education at network courses. Several exercises were created, oriented at routing, routing protocols, configuring and securing wireless networks. Further, it deals with selection of alternative operating system for RB433AH router and setting it up.

Klíčová slova

směrovač, mikrotik, openwrt, routeros

Keywords

router, mikrotik, openwrt, routeros

Citace

Patrik Šebeň: Využití routerboardů MikroTik pro výuku síťových kurzů na FIT, bakalářská práce, Brno, FIT VUT v Brně, 2010

VYUŽITÍ ROUTERBOARDŮ MIKROTIK PRO VÝUKU SÍŤOVÝCH KURZŮ NA FIT

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rába. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Patrik Šebeň
19. května 2010

Poděkování

Chcel by som poďakovať vedúcemu práce Ing. Jaroslavovi Rábovi zato, že my venoval svoj čas a odbornú pomoc pri vedení tejto práce.

Obsah

1 Úvod

Sieťové zariadenia už dávno nie sú doménou len pre odborníkov. Čoraz častejšie sa s nimi dostávajú do styku aj obyčajní užívatelia, či už ide iba o bežný prepínač, prístupový bod alebo aj smerovač. Preto sa výrobcovia takýchto zariadení snažia zjednodušiť ovládanie ako sa len dá, avšak vždy je dobré vedieť aj trocha teórie, ktorá sa za tým skrýva.

Aj keď táto práca pojednáva predovšetkým o konfigurácii a využití konkrétneho modelu smerovača routerboardu Mikrotik RB433AH v učebnom procese, nachádzajú sa v nej aj časti zaoberajúce sa teóriou z oblasti sietí, smerovania a bezpečnosti.

Práca je koncepčne členená do niekoľkých kapitol. Najskôr sa venuje objasneniu pojmov, ako smerovanie a smerovač a popisu jednotlivých typov smerovacích protokolov. Ďalšia časť je venovaná zariadeniu routerboard Mikrotik RB433AH a jeho hardwarovej špecifikácii. Neskôr je predstavený originálny operačný systém RouterOS a jeho prípadná náhrada iným operačným systémom. Nakoniec je zostavených niekoľko laboratórnych úloh pre využitie routerboardu pri vyučovaní počítačových sietí.

2 Smerovače a smerovanie

Smerovanie je proces výberu vhodnej cesty pri doprave paketu od zdroja k cieľu [1]. Smerovanie prebieha na tretej vrstve modelu OSI/ISO, kde pracujeme s logickými adresami a protokolom IP. Každé zariadenie pracujúce so sieťovou vrstvou modelu OSI/ISO sa zvyčajne podieľa na procese smerovania, avšak väčšinu smerovania v sieti internetu zabezpečujú smerovače.

Smerovač je zariadenie spájajúce dve a viac sietí za účelom prenosu dát medzi nimi. Smerovače disponujú niekoľkými sieťovými rozhraniami, ktoré nemusia byť rovnakého typu. Po prijatí paketu na niektorom rozhraní a analýze cieľovej adresy sa smerovač snaží poslať paket do jeho cieľovej destinácie za pomoci smerovacej tabuľky.

Smerovacia tabuľka je základná dátová štruktúra v smerovačoch. Obsahom každej smerovacej tabuľky sú minimálne nasledovné tri polia:

- cieľová adresa – zvyčajne v tvare 172.16.0.0/16
- metrika – cena cesty
- adresa skoku (next hop) – adresa zariadenia, ktorému bude paket poslaný na jeho ceste k cieľu

Pri smerovaní sa porovnáva cieľová adresa so smerovacou tabuľkou a vyberá sa záznam s najdlhším vyhovujúcim prefixom. V prípade, že existuje viacej ciest k cieľu, vyberá sa cesta s najmenšou metrikou. Smerovacia tabuľka vzniká za pomoci smerovacích protokolov a algoritmov. Smerovacie protokoly delíme na statické a dynamické.

Pri statickom smerovaní sa smerovacia tabuľka automaticky nemení, všetky zmeny je nutné zadávať ručne. Toto riešenie je vhodné pokiaľ sa nám štruktúra siete nemení a je zaručená jej stabilita. Výhoda spočíva v nezaťažovaní pásma smerovacími informáciami.

Dynamické smerovanie reaguje na zmeny v sieti a podľa toho aj upravuje smerovaciu tabuľku. Podľa spôsobu výmeny smerovacích informácii delíme smerovacie protokoly na:

- **Centralizované** – všetky smerovače v sieti posielajú informácie o stave okolitých sietí do smerovacieho centra. Centrum má mapu celej siete a podľa toho zostaví optimálne tabuľky pre každý smerovač a rozošle ich. Vzhľadom na príliš veľa problémov tohoto riešenia sa v praxi neuplatnilo.
- **Izolované** – každý smerovač rozhoduje samostatne. Neodosielajú sa žiadne smerovacie informácie a prichádzajúce pakety sa pošlú na všetky rozhrania okrem toho z ktorého prišli. Takéto riešenie je nehospodárne a často vznikajú cykly v sieti.
- **Distribuované** – štandardné smerovacie protokoly využívané v sieti internet. Smerovacie informácie sa postupne predávajú medzi susednými smerovačmi, až sa rozšíria do celej siete. Medzi predstaviteľov patria protokoly RIP, OSPF a BGP.
- **Hierarchické** - snažia sa riešiť problém rozľahlých sietí rozčlenením na jednotlivé oblasti. Informácie o zmenách topológie siete sa šíria v rámci oblastí a hraničné smerovače si navzájom poskytujú iba súhrnné informácie. Tento prístup využíva protokol OSPF.

Podľa spôsobu výberu najlepšej cesty delíme smerovacie algoritmy na :

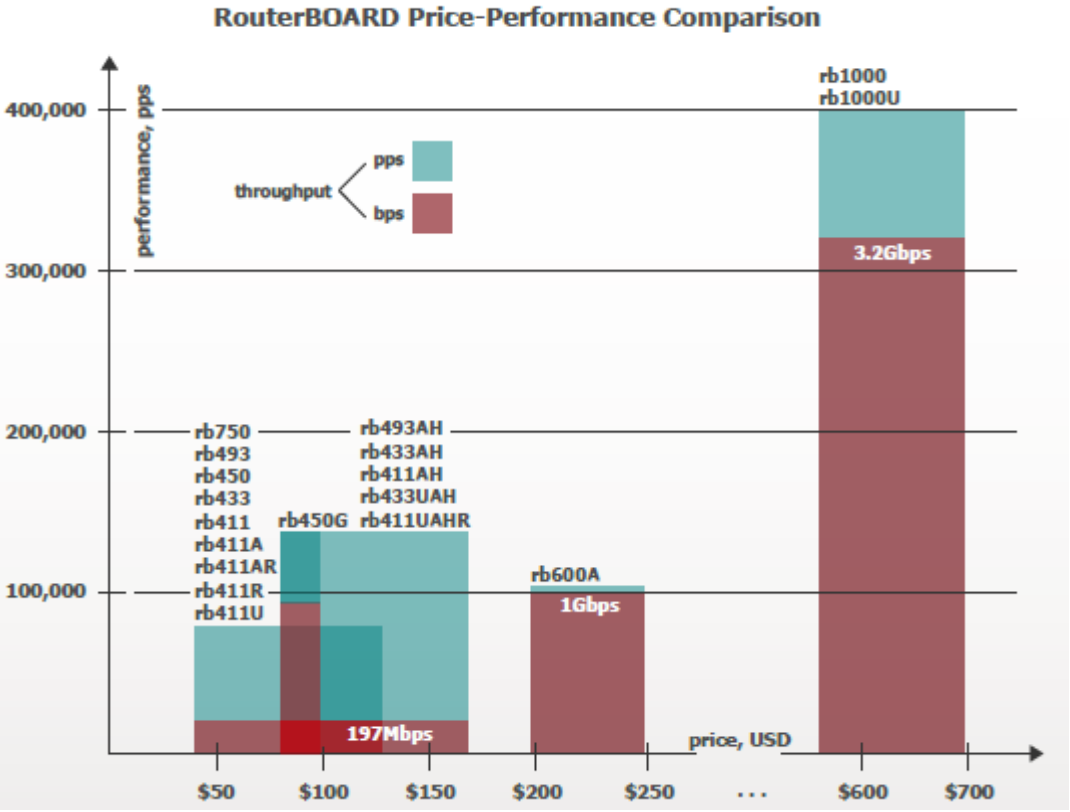
- **distance vector algoritmy –** výber ideálnej cesty spočíva v nájdení cesty s najmenším počtom preskokov napríklad protokol RIP
- **link-state algoritmy –** pri výbere cesty zvažujú aj šírku pásma. Príkladom je protokol OSPF.
- **path vector algoritmy –** fungujú na podobnom princípe ako distance vektor algoritmy, ale namiesto cieľa a vzdialenosti k nemu sa odosiela cieľ a popis cesty k nemu. Zástupcom je protokol BGP.

Podľa hierarchie internetu delíme protokoly na:

- **IGP** (interné smerovacie protokoly) – používajú sa na smerovanie vo vnútri autonómnych systémov. Požaduje sa pružnosť a rýchla reakcia na zmeny. Zástupcami sú protokoly RIP a OSPF.
- **EGP** (externé smerovacie protokoly) – slúžia na smerovanie medzi autonómnymi systémami, dôraz sa kladie hlavne na stabilitu. Zástupcom je protokol BGP.

3 Mikrotik

Firma Mikrotik založená v roku 1995 so sídlom v Lotyšsku sa zaoberá výrobou sieťových zariadení a softwaru nielen pre ISP ale aj pre bežné použitie. Routerboardy Mikrotik sú obľúbene nielen svojou prijateľnou cenou ale aj množstvom funkcií, ktoré ponúkajú. Ďalším produktom firmy je operačný systém RouterOS ktorý je okrem routerboardov možné použiť aj na PC.



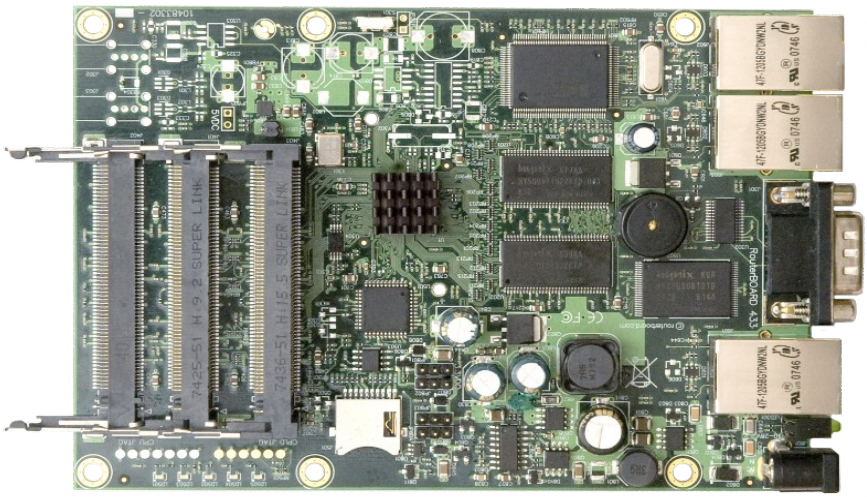
Obrázok 1: porovnanie ceny a modelu routerboardov Mikrotik

3.1 Routerboard Mikrotik RB433AH

Model RB433AH patrí medzi lepšie vybavené routerboardy. Bez problémov zvláda funkcie bezdrôtového AP, smerovača, firewallu alebo DHCP servra.

Hardwarova špecifikácia modelu RB433AH (Obr. 2):

- CPU Atheros AR7161 680MHz
- 128MB DDR SDRAM onboard memory
- 64MB onboard NAND memory
- 1x slot pre microSD kartu
- 3x 10/100 Mbit/s ethernet port s Auto-MDI/X
- 3x MiniPCI slot
- RS232C seriový port
- napájacie napätie 10..28V, POE
- RouterOS v3 Level5 licencia



Obrázok 2: RB433AH

Jednotlivé modely sa od seba líšia zvyčajne počtom eternetovách portov, MiniPCI slotov, veľkosťou pamätí a rýchlosťou CPU.

Procesor Atheros AR7161 je postavený na architektúre MIPS [12]. Architektúra MIPS dostala svoj názov podľa použitia takzvanej non-interlocked pipeline. Výhodou tejto architektúry je, že procesor môže spracovávať súčasne viac inštrukcii za predpokladu, že inštrukcie nevyužívajú rovnaké prostriedky procesora. Architektúru MIPS radíme medzi RISC procesory.

3.2 Príslušenstvo

Aby som otestoval čo najviac z možností, ktoré nám routerboard RB433AH ponúka, mal som k dispozícii jednu bezdrôtovú kartu R52.

Hardwarova špecifikácia bezdrôtovej karty R52 (Obr. 3):

- čipset Atheros AR5414
- 802.11a/b/g
- bezpečnosť
  - 64/128 WEP
  - TKIP a AES-CCM šifrovanie
  - WPA a WPA2
- certifikáty
  - FCC a EC
- frekvencie
  - 802.11b/g - 2.192 – 2.507 (5 MHz krok); 2.224 – 2.539 (5MHz krok)
  - 802.11a - 4.920 – 6.100 (5 MHz krok)

Obrázok 3: bezdrôtová karta R52

3.3 RouterOS

RouterOS je operačný systém firmy Mikrotik distribuovaný so smerovačmi postavený na linuxovom jadre verzie 2.6. Umožnuje aj beh na PC pričom je zaručená podpora viac jadrových procesorov, viac procesorových systémov, nových základných dosiek, ethernetových rozhraní do rýchlosti 10Gbit/s, 802.11a/b/g/n bezdrôtových kariet a 3G modemov.

Konfigurácia systému je možná pomocou lokálneho prístupu (klávesnica, monitor), sériového portu a terminálovej aplikácie, cez sieť pomocou telnetu alebo ssh a pomocou webového rozhrania. Existuje aj aplikácia Winbox, ktorá poskytuje užívateľsky príjemnejšie grafické rozhranie.

Čo sa týka funkčnosti nám RouterOS umožňuje:

- smerovanie statické a dynamické
  - IPv4 RIPv1 a v2, OSPF v2, BGP v4
  - IPv6 RIPng, OSPFv3, BGP
- firewall
- MPLS
- VNP
- Wireless
  - 802.11a/b/g/n klient alebo AP
  - WEP,WPA,WPA2 šifrovanie
  - ACL
  - RTS/CTS
- hotspot
- QoS
- Web Proxy
- DHCP server
- RADIUS klient a server
- SNMP

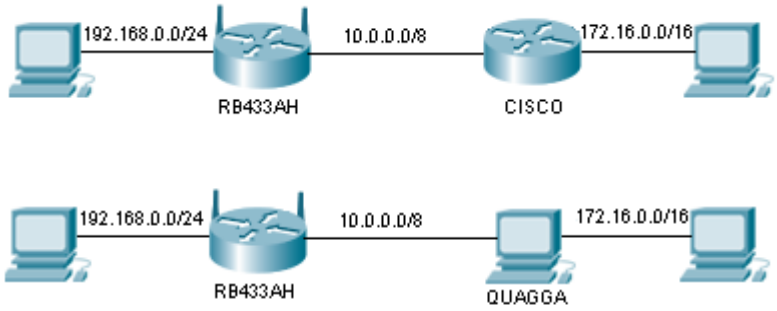
Pre použitie operačného systému RouterOS je nutné zaobstarať si licenčný kľúč. Je možné si vybrať zo šiestich stupňov, ktoré sa líšia možnosťou použitia niektorých funkcií, počtom pripojení a cenou.

Level number	0 (FREE)	1 (DEMO)	3 (WISP CPE)	4 (WISP)	5 (WISP)	6 (Controller)
Upgradable To	-	no upgrades	ROS v4.x	ROS v4.x	ROS v5.x	ROS v5.x
Initial Config Support	-	-	-	15 days	30 days	30 days
Wireless AP	24h limit	-	-	yes	yes	yes
Wireless Client, Bridge	24h limit	-	yes	yes	yes	yes
RIP, OSPF, BGP protocols	24h limit	-	yes	yes	yes	yes
EoIP tunnels	24h limit	1	1	unlimited	unlimited	unlimited
PPPoE tunnels	24h limit	1	1	200	500	unlimited
PPTP tunnels	24h limit	1	1	200	unlimited	unlimited
L2TP tunnels	24h limit	1	1	200	unlimited	unlimited
OVPN tunnels	24h limit	1	1	200	unlimited	unlimited
VLAN interfaces	24h limit	1	1	unlimited	unlimited	unlimited
P2P firewall rules	24h limit	1	1	unlimited	unlimited	unlimited
NAT rules	24h limit	1	unlimited	unlimited	unlimited	unlimited
HotSpot active users	24h limit	1	1	200	500	unlimited
RADIUS client	24h limit	-	yes	yes	yes	yes
Queues	24h limit	1	unlimited	unlimited	unlimited	unlimited
Web proxy	24h limit	-	yes	yes	yes	yes
User manager active sessions	24h limit	1	10	20	50	Unlimited

Obrázok 4: prehľad licencii systému RouterOS

4 Interoperabilita

Súčasťou zadania bolo otestovať interoperabilitu implementácie smerovacích protokolov RIP a OSPF routerboardu mikrotik RB433AH s inými implementáciami. Pre účely testovania, som zvolil smerovač od firmy cisco a počítač s linuxovou distribúciou ubuntu 9.10 so smerovacím nástrojom quagga. Jednotlivé sieťové komponenty som zapojil nasledovne (Obr. 5).



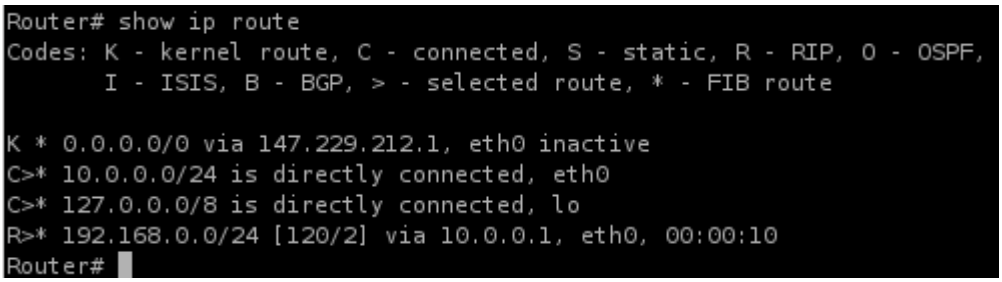
Obrázok 5: zapojenie testovacej zostavy

Prvým testovaným smerovacím protokolom bol protokol RIP. Konfigurácia jednotlivých zariadení:

- RB433AH
  - `/ip address`
  - `add address=10.0.0.1 netmask=255.0.0.0 interface=ether1`
  - `add address=192.168.0.1 netmask=255.255.255.0 interface=ether1`
  - `/routing rip`
  - `network add network=10.0.0.0/8`
  - `network add network=192.168.0.0/24`
  - `set redistribute-connected=yes`
- CISCO
  - `interface fa0/0`
  - `ip address 10.0.0.2 255.0.0.0`
  - `no shutdown`
  - `interface fa0/1`
  - `ip address 172.16.0.0 255.255.0.0`
  - `no shutdown`
  - `router rip`
  - `network 10.0.0.0`
  - `network 172.16.0.0`
- QUAGGA
  - `ifconfig eth0 10.0.0.2 netmask 255.0.0.0`
  - `ifconfig eth1 172.16.0.1 netmask 255.255.0.0`
  - `router rip`
  - `network 10.0.0.0/8`
  - `network 172.16.0.0/16`

Pre použitie nástroja quagga je nutná počiatočná konfigurácia, ktorú je možné nájsť na oficiálnych webových stránkach vývojárov [11].

Ako je možné vidieť (Obr. 6) jednotlivé cesty sa nám rozdistribuovali medzi smerovačmi, a konektivitu overíme príkazom ping.

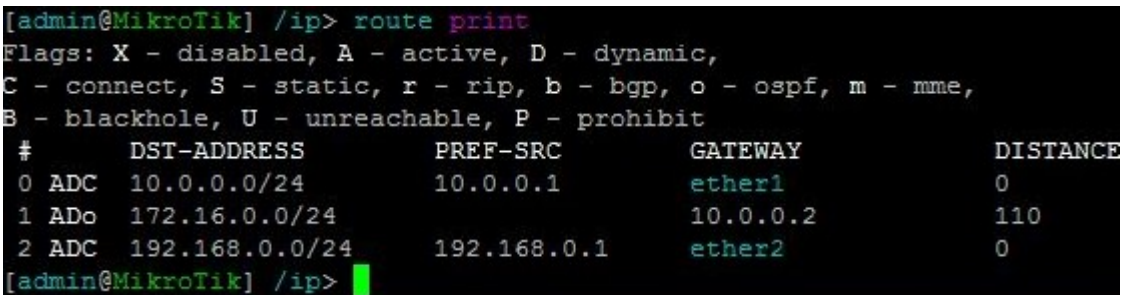


Obrázok 6: smerovacia tabuľka nástroja quagga

Druhým testovaným smerovacím protokolom bol protokol OSPF. Jednotlivé zariadenia boli nakonfigurované nasledovne:

- RB433AH
  - `/ip address`
  - `add address=10.0.0.1 netmask=255.0.0.0 interface=ether1`
  - `add address=192.168.0.1 netmask=255.255.255.0 interface=ether1`
  - `/routing ospf`
  - `network add network=10.0.0.0/8 area=backbone`
  - `network add network=192.168.0.0/24 area=backbone`
  - `instance set redistribute-connected=as-type-2`
- CISCO
  - `interface fa0/0`
  - `ip address 10.0.0.2 255.0.0.0`
  - `no shutdown`
  - `interface fa0/1`
  - `ip address 172.16.0.0 255.255.0.0`
  - `no shutdown`
  - `router ospf 1`
  - `network 10.0.0.0 0.255.255.255 area 0`
  - `network 172.16.0.0 0.0.255.255 area 0`
- QUAGGA
  - `ifconfig eth0 10.0.0.2 netmask 255.0.0.0`
  - `ifconfig eth1 172.16.0.1 netmask 255.255.0.0`
  - `router ospf`
  - `network 10.0.0.0/8 area 0`
  - `network 172.16.0.0/16 area 0`

Po kontrole smerovacích tabuliek (Obr. 7) môžeme povedať, že aj protokol OSPF pracuje správne.



Obrázok 7: smerovacia tabuľka routerboardu

5 Nasadenie alernatívneho OS

Napriek tomu, že RouterOS poskytuje bohatú škálu možností a konfiguruje sa jednoducho pomocou konzole a ešte jednoduchšie cez grafické rozhranie aplikácie Winbox, nemusí každému vyhovovať. Medzi nevýhody operačného systému RouterOS patrí hlavne nutnosť zakúpenia licencie pre jeho použitie a chýbajúca možnosť úpravy systému podľa vlastných potrieb.

5.1 Výber alternatívneho OS

Vzhľadom nato, že RouterOS je postavený na linuxovom jadre bude vhodné hľadať taktiež linuxovú alternatívu. Pravdepodobne najjednoduchšou možnosťou bude nájsť operačný systém, ktorý podporuje RB433AH priamo.

Ďalšou možnosťou je úprava existujúceho linuxového systému pre chod na požadovanom zariadení. Táto voľba môže byť časovo príliš náročná a napriek vynaloženému času sa nemusíme dopracovať k úspešnému koncu vzhľadom na príliš veľké rozdiely medzi zariadeniami (chýbajúca podpora konkrétneho procesora, rozdielne pamäťové úložiská). V prípade, že sa nám aj podarí upraviť systém do požadovanej podoby, môžu sa vyskytnúť problémy s jeho importom do zariadenia.

Riešením nášho problému môže byť Linux form scratch, čo je zostavenie linuxového systému od základov zo zdrojových kódov [13]. Toto riešenie je zvyčajne jediná možnosť v prípade veľmi hardwarovo špecifických platforiem, avšak v prípade takéhoto systému nemôžeme očakávať komunitnú podporu.

Keďže sa mi podarilo nájsť systém OpenWrt, ktorý je schopný behu na RB433AH a podporuje jeho hardware, nebolo nutné sa zaoberať zvyšnými riešeniami.

5.2 OpenWrt

OpenWrt je linuxová distribúcia určená pre vstavané zariadenia ako smerovače [10]. Pôvodne podporovala iba zariadenia Linksys serie WRT54G, dnes sa však podpora rozrástla aj na iných výrobcov ako Netgear, Asus, D-Link a ďalších. Hlavnou prednosťou systému je prínos nových funkcií, ktoré aktuálny hardware ponúka, ale v originálnom systéme chýbajú, prípadne nie sú úplne funkčné. Ovládanie je riešené štandardne pomocou príkazového riadku, dá sa však použiť aj webové rozhranie.

Systém OpenWrt je tiež ako RouterOS postavený na linuxovom jadre verzie 2.6. V prípade behu na zariadeniach využívajúcich flash pamäť používa dva súborové systémy SquashFS a JFFS2. Zaujímavosťou je, že každá verzia je pomenovaná podľa alkoholického nápoja a po prihlásení je možné vzhliadnuť aj jeho recept. Aktuálnou stabilnou verziou je 10.03 Backfire ktorá vyšla 6. apríla 2010, preto bola pre väčšinu testov použitá verzia 08.09 Kamikaze.

5.3 Import do zariadenia

Existuje niekoľko postupov ako dostať systém OpenWrt do zariadenia routerboard RB433AH. Prvým variantom je netboot, kde bootujeme operačný systém zo siete bez narušenia originálneho systému. Tento variant je vhodný pre prvotné testovanie funkčnosti systému, pretože nie pri každom reštarte zariadenia musíme mať k dispozícii TFTP server s obrazom linuxu. Druhou variantov je prepísanie

NAND pamäte, čiže nahradenie originálneho systému RouterOS. Vzhľadom nato, že licenčný kľúč sa uchováva na inom mieste, je možné bez problémov vrátiť originálny systém naspäť. RouterOS od verzie 3.24 a v4.0beta3 ponúka možnosť behu vlastných obrazov v metaroutry. Metarouter je virtuálny router, ktorý nám umožňuje beh iných systémov paralelne so systémom RouterOS.

5.3.1 Netboot

Prvým krokom je nastavenie bootloadera zariadenia RB433AH, kde musíme nastaviť bootovanie zo siete [7]. Routerboard pripojíme k PC pomocou sériového rozhrania a zahájime komunikáciu príkazom

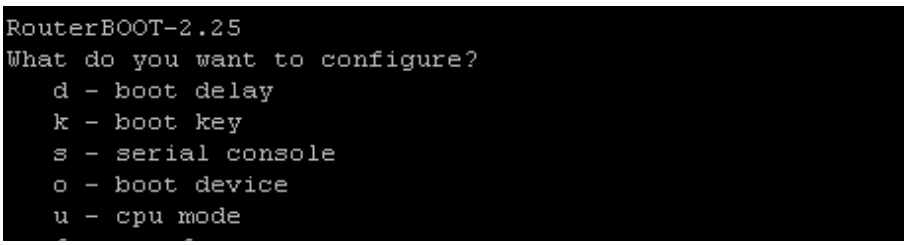
```
cu -1 ttyS0 -s 115200
```

kde parameter -1 označuje číslo sériového portu a -s udáva rýchlosť v bitoch za sekundu. V prípade iného prístupu údaje, ktoré potrebujeme sú:

- 115200bps
- 8 data bits
- 1 stop bit
- žiadna parita

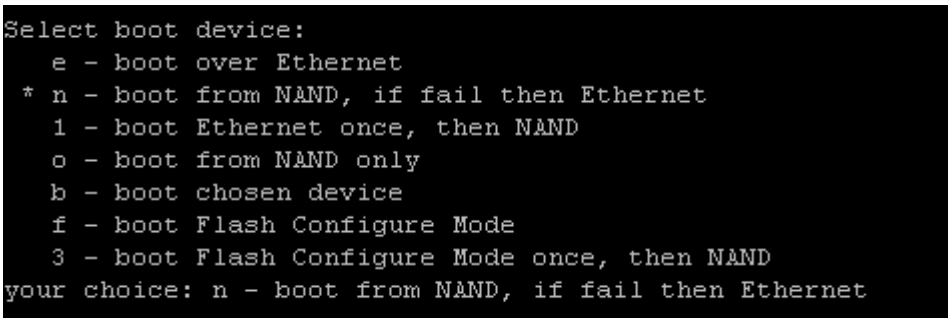
Obrázok 8: bootovacia obrazovka

Po úspešnom pripojení vidíme router bootovat (Obr. 8). Pre prístup do nastavení bootloadera je nutné do dvoch sekúnd stlačiť ľubovoľnú klávesu. V menu (Obr.9) zvolíme možnosť boot device stlačením klávesy o.



Obrázok 9: menu bootloadera

Nakoniec zvolíme možnosť boot over Ethernet (Obr. 10) zadaním klávesy e a ukončíme konfiuráciu.

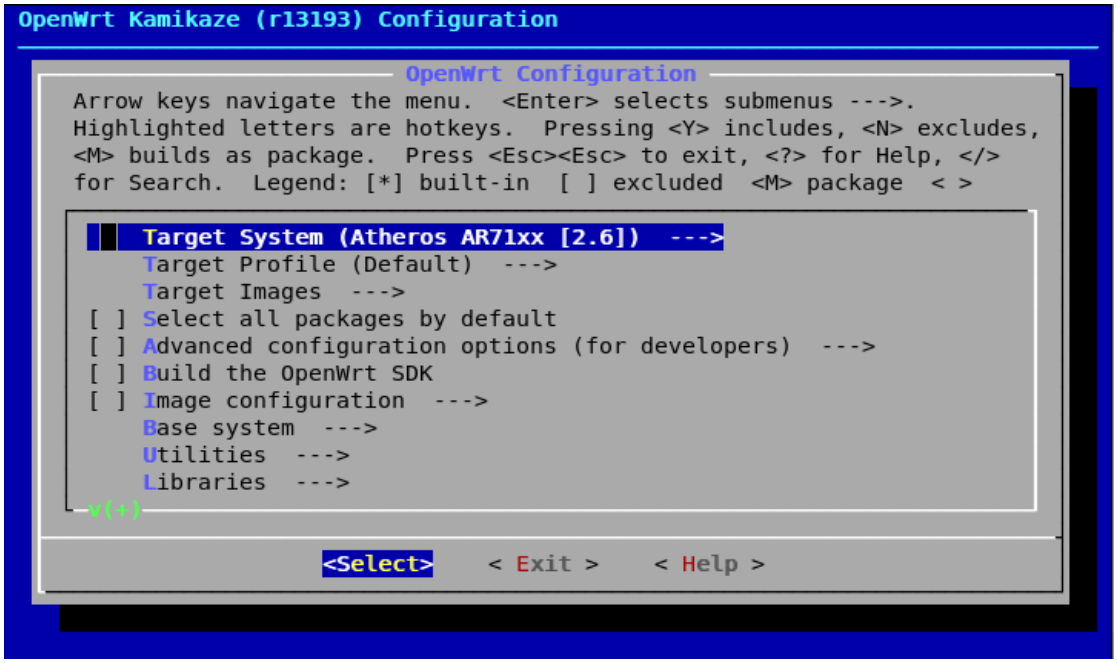


Obrázok 10: výber bootovacieho zariadenia

Teraz môžeme pristúpiť k príprave samotného systému. Budeme potrebovať stiahnuť zdrojové kódy pre kompiláciu systému:

```
mkdir ~/openwrt
cd ~/openwrt
sudo apt-get install svn
svn co https://svn.openwrt.org/openwrt/trunk/
```

Jednotlivé príkazy sa môžu líšiť podľa použitej linuxovej distribúcie. Keď máme k dispozícii všetko potrebné, je možné zahájiť kompiláciu systému pre netboot. Systém OpenWrt používa pre konfiguráciu jadra nástroj Buildroot (Obr. 11). Buildroot je sada makefileov a patchov, pomocou ktorých jednoducho vytvoríme obraz linuxového jadra [5]. Je vhodný predovšetkým pre prácu s vstavanými zariadeniami, pretože zvyčajne nepoužívajú klasické x86 procesory, ale hlavne MIPS a ARM procesory.

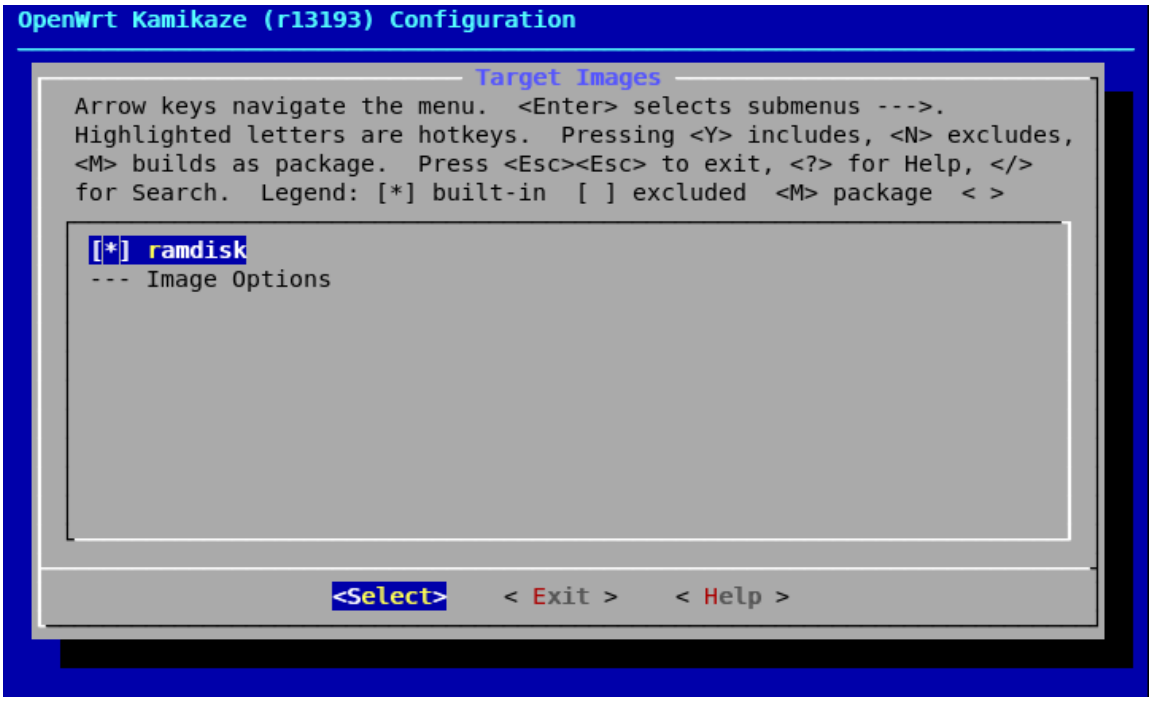


Obrázok 11: prostredie buildrootu

Konfiguráciu systému zahájime príkazom:

```
make menuconfig
```

V položke Target System musíme zvoliť vhodný typ procesora v našom prípade to je Atheros AR71xx[2.6] a v Target Images (Obr. 12) zvolíme možnosť ramdisk.



Obrázok 12: výber ramdisk formátu

Následne ukončíme, uložíme konfiguráciu, ukončíme buildroot a môžeme zahájiť kompiláciu:

```
make
```

Po úspešnej kompilácii nájdeme v položke /bin výsledný obraz openwrt-ar71xx-vmlinux-initramfs.elf, ktorý ďalej použijeme. K importu systému do zariadenia budeme na hostiteľskom systéme potrebovať dhcp a tftp server.

```
sudo apt-get install dnsmasq
sudo apt-get install tftpd-hpa
```

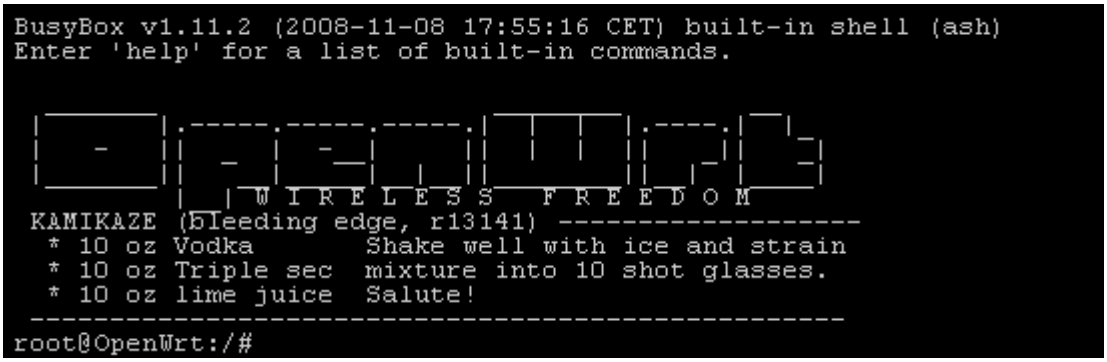
Teraz potrebujeme upraviť konfiguráciu dhcp servera, aby vyhovoval našim potrebám. To vykonáme nasledovnými zmenami v súbore /etc/dnsmasq.conf :

```
#nastavíme požadovaný rozsah ip adries
     dhcp-range=192.168.6.100,192.168.6.119
#pridelíme ip adresu routerboardu(namiesto XX doplníme konkrétnu
MAC adresu)
     dhcp-host=00:0c:42:XX:XX:XX,192.168.6.101
#nastavíme umiestnenie obrazu pre netboot
     dhcp-boot=/tftpboot/openwrt-ar71xx-vmlinux-
     initramfs.elf,boothost,192.168.6.1
```

Z vyššie uvedenej konfigurácie vyplýva, že musíme súbor openwrt-ar71xx-vmlinux-initramfs.elf umiestniť do priečinka /tftpboot. Ak máme všetko nakonfigurované, môžeme spustiť služby a nabootovať RB433AH.

```
sudo /etc/init.d/tftpd-hpa start
sudo /etc/init.d/dnsmasq start
```

Pokiaľ nenastali žiadne problémy, mali by sme mať systém OpenWrt (Obr. 13) nabootovaný v zariadení.



Obrázok 13: úpešne zavedený systém OpenWrt

5.3.2 Permanentné uloženie systému

Ak sa nám podarilo zostaviť systém a úspešne ho importovať do zariadenia pomocou bootovania zo siete, môžeme uvažovať o jeho permanentnom uchovaní v zariadení. K prepisu NAND pamäte a originálneho systému budeme potrebovať beh systému OpenWrt. To dosiahneme pomocou návodu z predchádzajúcej kapitoly.

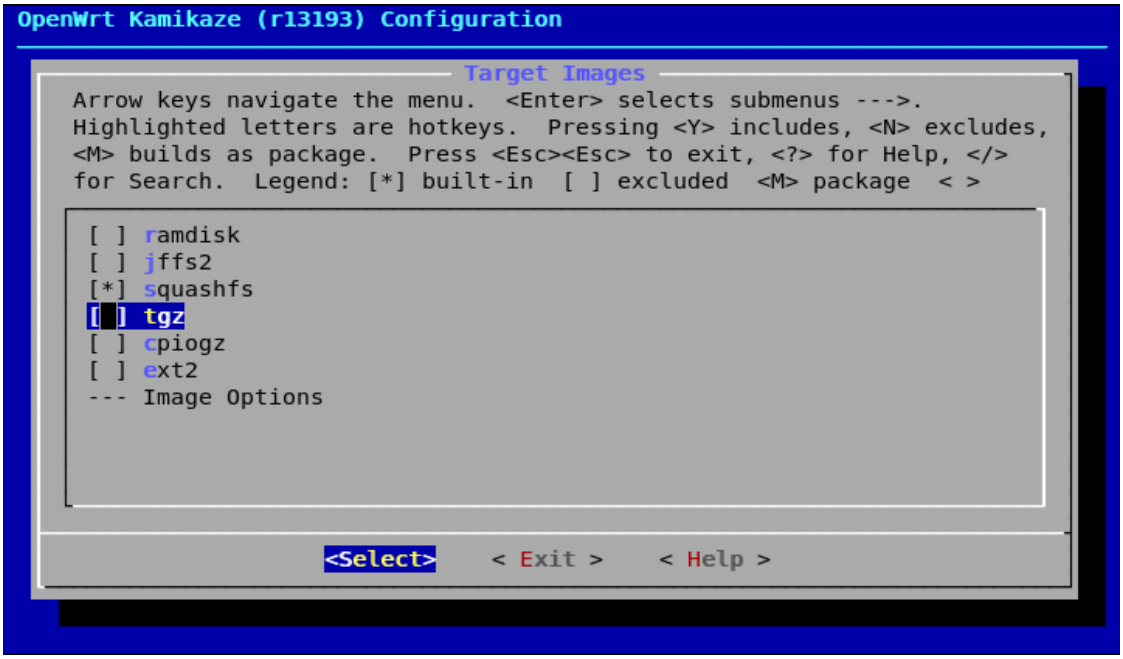
Podobne ako pri netboote musíme aj teraz zostaviť pomocou Buildrootu nový systém, pretože obraz jadra použitého pre netboot nie je vhodný k zápisu do NAND pamäte. Konfiguráciu zahájime príkazom:

```
make menuconfig
```

Zvolíme procesor Atheros AR71xx[2.6] v položke Target System a v ponuke Target Images vyberieme namiesto ramdisku položku tzg (Obr. 14). Zahájime kompiláciu príkazom:

```
make
```

Konfiguráciou buildrootu môžeme ďalej nastaviť počiatočnú sieťovú konfiguráciu, alebo pridať do systému balíčky ktoré budeme pri práci vyžadovať. Nové balíčky je možné v prípade prístupu na internet stiahnuť a doinštalovať pomocou balíčkového systému opkg prípadne ipkq (staršie verzie OpenWrt).



Obrázok 14: výber tgz formátu

Tentokrát nájdeme v podadresáry /bin dva súbory :

- openwrt-ar71xx-vmlinux.elf – linuxové jadro
- openwrt-ar71xx-rootfs.tgz – root filesystem

Ďalším krokom je nahradenie originálneho systému, systémom práve skompilovaným. Potrebujeme zabezpečiť, aby hostiteľský počítač aj routerboard boli na rovnakej podsieti, v našom prípade 192.168.0.0/24. V prípade, že sme nastavili ip adresu RB433AH už pri konfigurácii obrazu, nie sú nasledujúce príkazy nutné:

```
ip l s br-lan down
brctl delbr br-lan
ip a a 192.168.0.2/24 dev eth0
```

Pokiaľ nám sieť funguje (overíme príkazom ping) môžeme pripojiť disky:

```
mkdir /mnt/kernel
mkdir /mnt/root
mount /dev/mtdblock1 /mnt/kernel
mount /dev/mtdblock2 /mnt/root
```

Ďalej musíme vymazať dáta na mtdblock1 a mtdblock2 (v prípade vynechania tohto kroku sa môže systém správať nepredvídateľne):

```
umount /mnt/kernel
umount /mnt/root
mtd erase /dev/mtdblock1
mtd erase /dev/mtdblock2
mount /dev/mtdblock1 /mnt/kernel
mount /dev/mtdblock2 /mnt/root
```

Pretože systém OpenWrt nepracuje z príponou .tgz musíme ho prebaliť, najlepšie na .tar:

```
cd bin/
mkdir root
cp openwrt-ar71xx-rootfs.tgz root/
cd root
tar xf openwrt-ar71xx-rootfs.tgz
rm openwrt-ar71xx-rootfs.tgz
tar cf ../root.tar *
```

Pomocou nástroja scp prenesieme požadované súbory do routerboardu:

```
scp 192.168.0.1:/cesta/k/bin/root.tar /mnt/root/
scp 192.168.0.1:/cesta/k/bin/openwrt-ar71xx-vmlinux.elf /mnt/kernel/kernel
```

Nakoniec rozbalíme root :

```
cd /mnt/root
tar xf root.tar
rm root.tar
```

Systém OpenWrt máme v tejto fáze uložený v NAND pamäti a ostáva nám zmeniť nastavenie bootloadera zariadenia do pôvodného stavu. Reštartujeme zariadenie a v menu bootloadera zvolíme možnosť o – boot device a následne vyberieme položku o – boot from NAND only.

5.3.3 Metarouter

V prípade, že vyžadujeme od systému RouterOS nadštandardné vlastnosti, ktoré neposkytuje, ale nechceme alebo nemôžme ho nahradiť inou alternatívou, je metarouter to správne riešenie. Metarouter je virtuálny router, na ktorom od verzie RouterOS 3.24 a 4.0beta3 môžeme použiť aj vlastnoručne vytvorené obrazy systémov [9]. Aktuálne môžeme vytvoriť až 8 metaroutrov, tento počet je v pláne v budúcnosti navýšiť. RouterOS umožňuje pripojiť k virtuálnemu rozhraniu rozhranie fyzické a tým nahradiť originálny systém, avšak za cenu istého zníženia výkonu. Systém OpenWrt je schopný behu na metaroutry a umožňuje nám tak pridať funkcie webového servera alebo databáze.

Systém OpenWrt vhodný pre metarouter môžeme stiahnuť z oficiálnych stránok firmy Mikrotik. V prípade, že vyžadujeme nadštandardnú funkčnosť, je lepším riešením vlastnoručne zostaviť systém aj s pridanými balíčkami. Ak sme sa rozhodli pre druhú variantu budeme opäť potrebovať všetky zdrojové súbory:

```
mkdir openwrt
cd openwrt
svn co svn://svn.openwrt.org/openwrt/trunk
cd trunk
```

Ďalej budeme potrebovať aktuálny patch zo stránok firmy Mikrotik:

```
wget http://www.mikrotik.com/download/openwrt-metarouter-1.1.patch
```

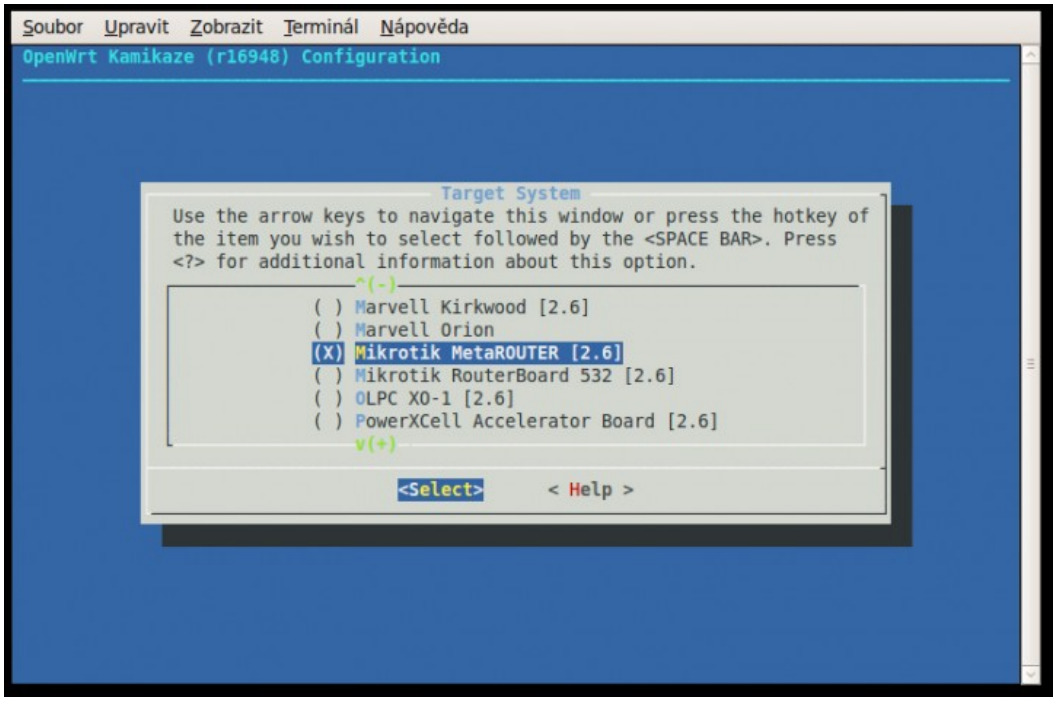
Pretože sa systém OpenWrt rýchlo vyvíja, je možné, že nebude aplikovateľný na najnovšiu revíziu. V tom prípade treba skúsiť staršiu revíziu. Patch aplikujeme príkazom:

```
patch -p0 < openwrt-metarouter-1.1.patch
```

Všetko potrebné pre konfiguráciu Buildrootu a kompiláciu máme k dispozícii. Postupujeme ako v predchádzajúcich prípadoch, avšak s menšími odlišnosťami:

```
make menuconfig
```

V ponuke nastavíme Target System ako Mikrotik METAROUTER[2.6] a Target Image ako tzg (Obr. 15).



Obrázok 15: konfigúrácia systému pre metarouter

Zahájime kompiláciu:

```
make
```

Výsledný obraz potrebujeme preniesť do routerboardu, buď pomocou nástroja scp alebo aplikácie Winbox (Obr. 16):

```
scp openwrt-metarouter-rootfs.tgz admin@10.0.0.1
```

Vytvoríme virtuálny router :

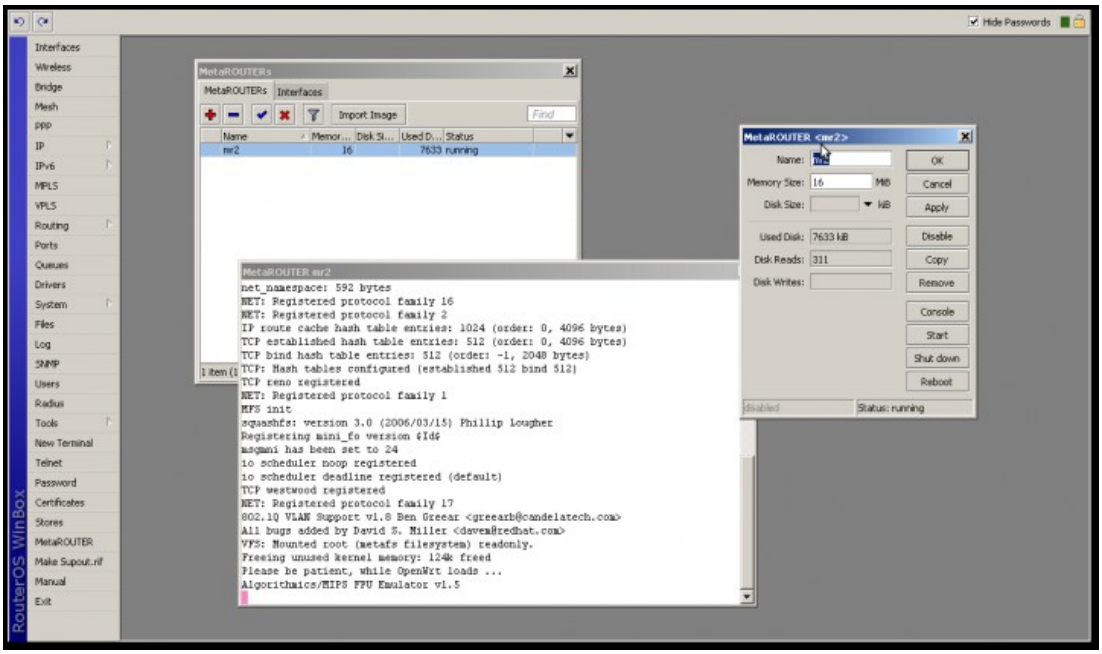
```
[admin@MikroTik] > metarouter
[admin@MikroTik] /metarouter> import-image
      file-name=openwrt-metarouter-rootfs.tgz
```

Vytvorený virtuálny router má k dispozícii 16MiB pamäte a neobmedzený diskový priestor (v rámci možností routerboardu). Tieto hodnoty sme samozrejme schopní upraviť podľa našich potrieb. Takto vytvorený metarouter zatiaľ nemá prístup k sieti, preto musíme prepojiť virtuálne rozhranie s niektorým z fyzických. Pravdepodobne najjednoduchšou voľbou je nakonfigurovať most (bridge):

```
[admin@MikroTik] /metarouter> /interface bridge
[admin@MikroTik] /interface bridge> add
[admin@MikroTik] /interface bridge> port
[admin@MikroTik] /interface bridge port> add bridge=bridge1
     interface=ether1
```

Vytvorili sme most na prvom ethernetovom rozhraní (ether1). Na tento most je nutné pripojiť aj virtuálne rozhranie metaroutra:

```
[admin@MikroTik] /interface bridge port> / metarouter
[admin@MikroTik] /metarouter interface>
[admin@MikroTik]  /metarouter  interface>  add  virtual-machine=mr2
     dynamic-bridge=bridge1 type=dynamic
```



Obrázok 16: prostredie aplikácie Winbox

Ostáva nám nastaviť ip adresu v systéme OpenWrt. Do systému sa prepneme príkazom:

```
[admin@MikroTik] /metarouter> console 0
ifconfig br-lan ip adresa netmask maska siete
```

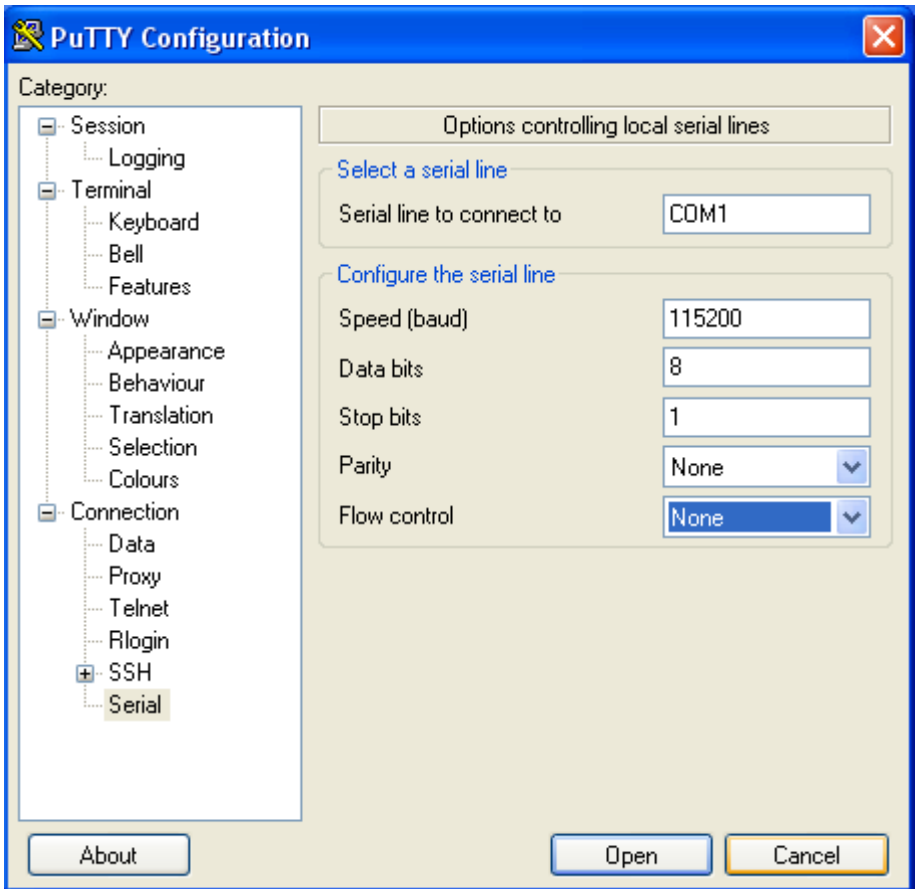
Číslo 0 udáva poradie metaroutra v zozname, ale pre prehľadnosť môže byť nahradené jeho názvom. Do originálneho systému sa vrátime pomocou klávesovej kombinácie Ctrl + A a Q.

6 Laboratórne cvičenia

Hlavnou náplňou tejto práce bolo vytvoriť niekoľko laboratórnych cvičení s využitím routerboardu RB433AH, ktoré bude možné využiť pri výuke sieťových kurzov. Pretože čas vyhradený pre cvičenia nie je neobmedzený a nemôže byť celý venovaný smerovaniu a konfigurácii jedného zariadenia, snažil som sa vytvoriť úlohy, ktoré poskytnú študentom dostatok informácii o preberanej tématike a súčasne budu získané znalosti schopní využiť v bežnom živote, ale aj profesionálnej praxi. Úlohy sú zamerané na základy smerovania (protokoly RIP a OSPF), konfiguráciu bezdrôtových sietí, bezpečnosť a preklad adries.

6.1 Konfigurácia

Pre konfiguráciu zariadenia máme k dispozícii niekoľko možností [2]. Prvou možnosťou je grafické rozhranie programu Winbox, ktoré je bohužiaľ iba pre systémy Windows, preto sa sním nebudem ďalej zaoberať. Druhým variantom je sériový port RS232C, ktorý pripájame ku COM portu počítača (Obr. 17).



Obrázok 17: pripojenie cez sériový port nástrojom putty

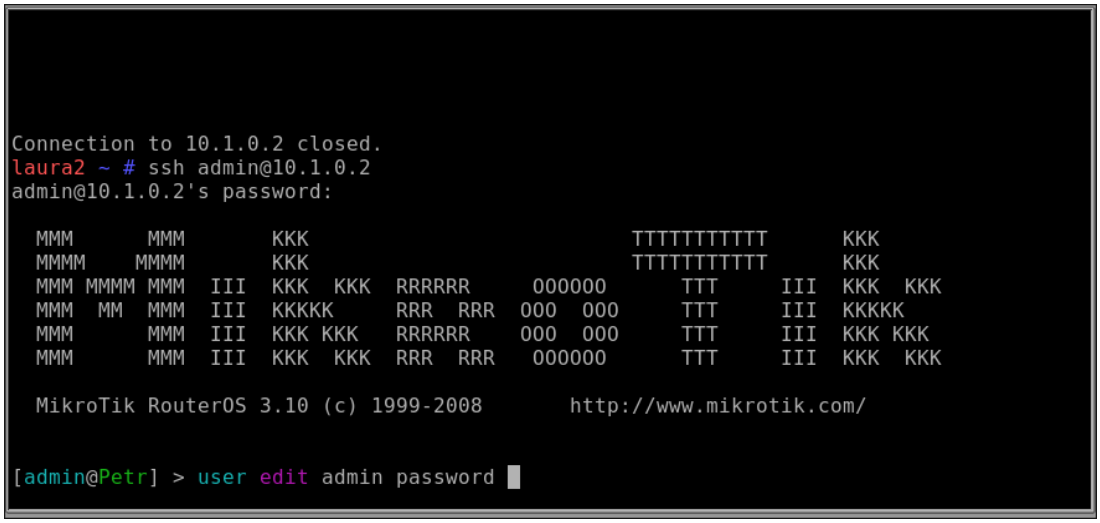
Aby sme sa nemuseli pripájať sériovým rozhraním potrebujeme nastaviť na niektorom z ethernetových portov ip adresu:

```
/ip address add address=10.0.0.1/24 interface=ether1
```

Po pridelení ip adresy je možné pripojenie pomocou telnetu alebo SSH (iba ak máme nastavené heslo).

Ak sa jedná o prvý štart routerboardu je login admin a heslo nie je nastavené, preto pre zvýšenie bezpečnosti je vhodné ho nastaviť:

```
user edit admin password
```

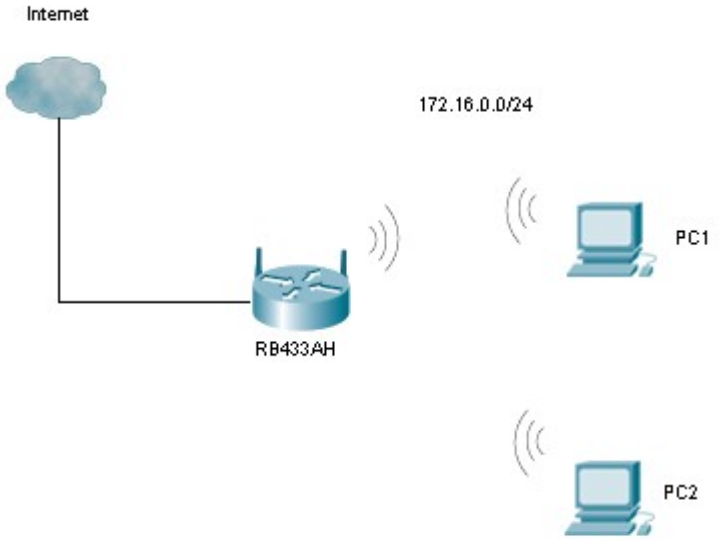


Obrázok 18: pripojenie pomocou ssh

Existuje aj možnosť konfigurácie pomocou webového rozhrania.

6.2 Úloha č.1

Úlohou študenta bude nakonfigurovať routerboard, aby pracoval ako AP s prideľovaním ip adries a zabezpečiť pripojenie k internetu. Bezdrôtový prenos musí byť šifrovaný pomocou vhodného bezpečnostného protokolu.



Obrázok 19: požadové zapojenie úlohy č.1

Bezdrôtový prístupový bod (anglicky access point -AP) je zariadenie, ku ktorému sa klienti siete pripájajú. Klienti nekomunikujú priamo, ale prostredníctvom AP, vďaka čomu nemusia byť vo vzájomnom rádiovom spojení. Tento typ siete sa nazýva infra-štruktúrna sieť. Opakom sú siete typu ad-hoc, kde dvaja a viac klientov musí byť vo vzájomnom dosahu a komunikácia neprebieha prostredníctvom prostredníka. Medzi nevýhody bezdrôtových sietí patria vyššie bezpečnostné riziká, klesajúca kvalita spojenia narastajúcou vzdialenosťou a stratovosť dát pri veľkom počte komunikujúcich zariadení v jednom pásme.

Prvým krokom bude konfigurácia AP a pridelenie ip adresy. Adresu siete vyberieme z rozsahu privátnych adries:

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

```
/ip address add address=172.16.0.1 netmask=255.255.255.0
          interface=wlan1
```

Pokiaľ pracujeme vo viacerých skupinách, bude vhodné zvoliť rozdielne adresy sietí. Položka interface určuje, ktorej bezdrôtovej karte adresu prideľujeme. Pokiaľ máme kariet viac, sú číslované od 1 vyššie.

Bezdrôtovú kartu treba následne nastaviť do režimu AP. Tu buď upravíme už predtým vytvorený profil alebo založíme nový:

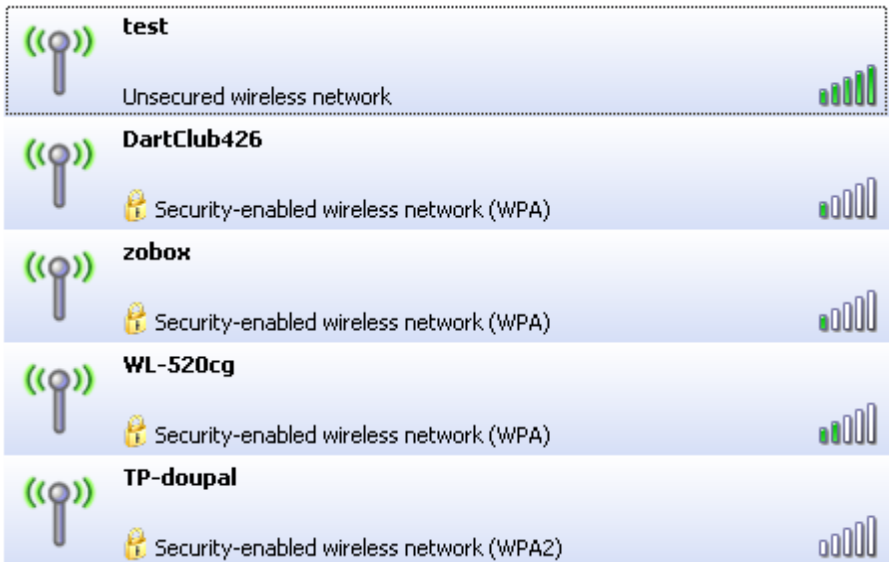
```
/interface wireless
```

```
set mode=ap-bridge ssid=test band=2.4ghz-b/g frequency=2412
          numbers=0
enable 0
```

Položka mode určuje režim práce bezdrôtovej karty. Máme na výber režim prístupového bodu, mostu alebo klienta. Názov siete, ktorý klienti pri pripájaní môžu vidieť, zadávame do položky ssid. Frekvenciu nastavíme tak, aby sa nám jednotlivé bezdrôtové siete rušili čo najmenej. Ideálny je výber kanálov 1, 6 a 11:

- kanál 1 – 2412
- kanál 6 – 2437
- kanál 11 – 2462

V tejto fáze by klientské stanice mali vidieť sieť s názvom test (Obr. 20).



Obrázok 20: zoznam dostupných sietí

K takto vytvorenej sieti sa môže pripojiť ktokoľvek a prenos dát nie je šifrovaný, preto je vhodné použiť niektorý z bezpečnostných protokolov WEP, WPA alebo WPA2.

Pretože dáta v bezdrôtových sieťach sú prenášané voľným priestorom, sú náchylnejšie k odposluchu. To bol jeden z dôvodov pre vývoj rôznych druhov zabezpečenia. WEP je jedným z prvých bezpečnostných protokolov používaných v bezdrôtovom prenose dát [4]. Bol predstavený v roku 1997 a je súčasťou štandardu IEEE 802.11 od roku 1999. Napriek tomu, že dnes na jeho prelomenie stačí niekoľko minút, je stále dosť rozšírený. Práve kvôli nedostatkom WEPu sa vyvinulo WPA, ako dočasné riešenie, kým nebude dokončená špecifikácia IEEE 802.11i [8]. Pomocou práce s kľúčmi sa snaží eliminovať slabé miesta jeho predchodcu. Po dokončení štandardu IEEE 802.11i vzniklo WPA2, ktoré bohužiaľ nie je podporované staršími zariadeniami. Hlavným rozdielom medzi WPA a WPA2 je pridanie nového algoritmu CCMP založeného na AES, ktorý je zatiaľ považovaný za úplne bezpečný.

Bezpečnostný profil nastavíme nasledovne:

```
/interface wireless security-profiles
add name=wpa2-test mode=dynamic-keys authentication-types=wpa2-psk
          wpa2-pre-shared-key=kluc12345 unicast-ciphers=tkip
          group-cipher=tkip
/interface wireless set security-profile=wpa2-test
```

Pre prehľadnosť si objasníme význam jednotlivých položiek:

- name – názov profilu
- mode – výber typu šifrovania, dynamic-keys = WPA
- authentication-types – výber typu autentifikácie
- unicast-ciphers – typ šifrovania pre unicast
- group-cipher – typ šifrovania pre broadcast a multicast
- wpa2-pre-shared-key – heslo pre pripojenie

Bezpečnosť komunikácie by mala byť dostatočná a môžeme nakonfigurovať dhcp server, aby klienti nemuseli nastavovať ip adresy ručne:

```
/ip dhcp-server
setup
```

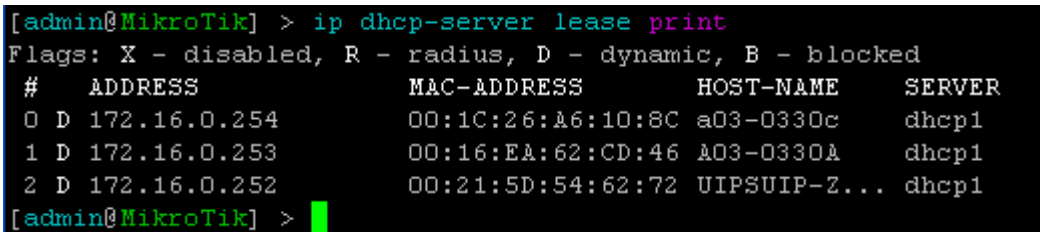
Príkaz setup nás postupne prevádza (Obr. 21) jednoduchým konfiguračným manažérom, ktorý sa nás postupne pýta na rozhranie, na ktorom dhcp server bude pracovať, adresu siete, adresu brány, rozsah ip adries, ktoré môže prideliť, adresu DNS servera a dobu na ktorú pridelí adresu.

```
[admin@MikroTik] > ip dhcp-server
[admin@MikroTik] /ip dhcp-server> setup
Select interface to run DHCP server on

dhcp server interface: wlan1
Select network for DHCP addresses

dhcp address space: 172.16.0.0/24
Select gateway for given network

gateway for dhcp network: 172.16.0.1
Select pool of ip addresses given out by DHCP server

addresses to give out: 172.16.0.2-172.16.0.254
Select DNS servers

dns servers: 172.16.0.1
Select lease time

lease time: 3d
[admin@MikroTik] /ip dhcp-server>
```

Obrázok 21: sprievodca konfiguráciou DHCP servra

Sledovať pridelené adresy môžeme príkazom:

```
/ip dhcp-server lease print
```



Obrázok 22: prehľad pridelených adries

Posledným krokom je povoliť klientom bezdrôtovej siete pripojenie k internetu. Budeme potrebovať, aby nám poskytovateľ internetu pridelil adresu, preto nastavíme dhcp klienta na jednom z eternetových rozhraní:

```
/ip dhcp-client add interface=ether2 disabled=no
```

Zoznam rozhraní a ich ip adries vyvoláme nasledovne:

```
/ip address print
```

Poskytovateľ internetu nám vo väčšine prípadov pridelí iba jednu verejnú ip adresu, preto musíme nakonfigurovať jednoduchý NAT:

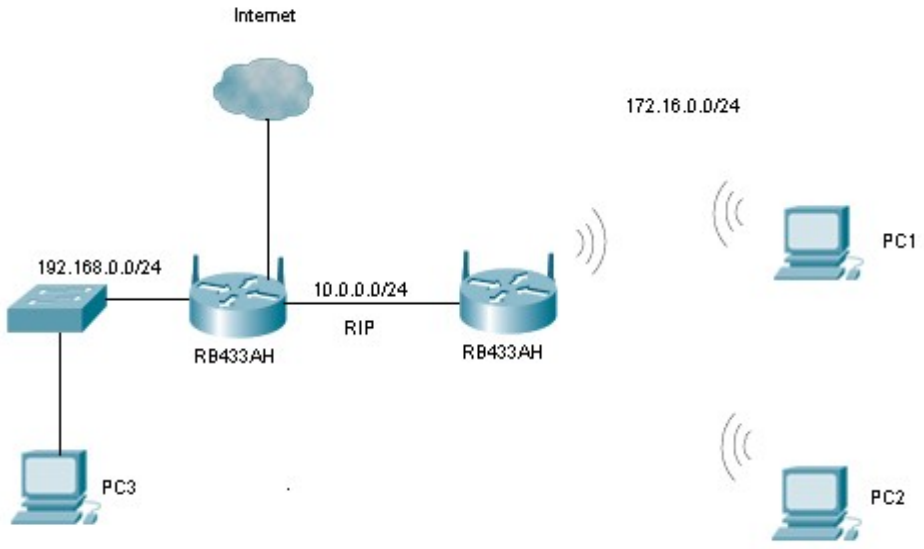
```
/ip firewall nat add chain=srcnat action=masquerade
        out-interface=ether2
```

NAT je technika prekladu ip adries zvyčajne privátnych na verejnú [14]. Princíp fungovania je nasledovný. Paketu prichádzajúcemu z lokálnej siete a smerujúcemu do internetu, sa v smerovači jeho zdrojová adresa preloží na verejnú adresu smerovača. Záznam o tejto zmene (adresa cieľa a port) sa uchová v tabuľke a pri odpovedi z internetu sa podľa toho určí komu na lokálnej sieti je odpoveď smerovaná. NAT zvyšuje bezpečnosť počítačov za ním pripojených, pretože útočník nemôže komunikovať priamo s jeho cieľom. Bohužiaľ má takýto preklad adries aj svoje nevýhody v podobe problémov s tunelovacími protokolmi ako IPsec, pretože pri preklade sa upravuje hlavička paketu a to ma za následok chyby pri kontrolných súčtoch tunelovacích protokolov.

Po úspešnom absolvovaní úlohy, by mal študent získať dostatočné znalosti z oblasti bezdrôtových sietí a ich zabezpečení. Ďalej si osvojí základy prekladu adries jeho využitia a taktiež bude schopný nakonfigurovať dhcp server. V praxi sú znalosti využitelné pri vytváraní domácej bezdrôtovej siete s sprístupom na internet. Cvičenie časovo vychádza na 45 minút až hodinu.

6.3 Úloha č.2

Úlohou študenta bude zoznámiť sa so základmi smerovania a dynamický smerovacím protokolom RIP.



Obrázok 23: požadované zapojenie úlohy č.2

K úlohe budeme potrebovať minimálne dva routerboardy a niekoľko počítačových staníc. Prvým krokom bude vytvorenie jednotlivých sietí

```
/ip address add address=ip adresa netmask=maska podsiete
         interface=fyzické rozhranie
```

Komunikáciu jednotlivých zariadení overíme príkazom ping. Ak nám zariadenia medzi sebou komunikujú, môžeme začať konfiguráciu dynamického smerovania.

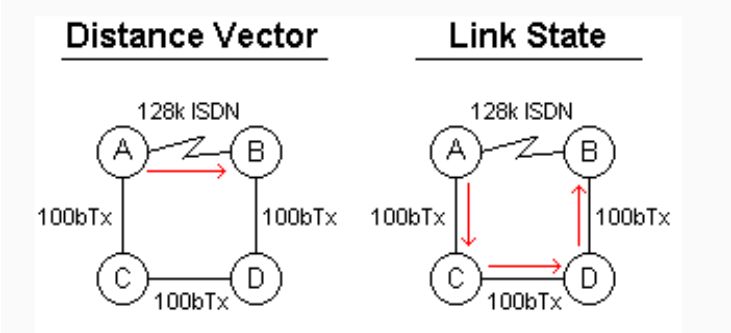
Protokol RIP je najjednoduchším dnes používaným smerovacím protokolom [1,3]. Funguje tak, že každých 30 sekúnd vysiela vše-smerovo všetky cesty ktoré pozná. To však môže zbytočne v prípade veľkých sietí spotrebúvať značnú šírku pásma. RIP nepoužíva spúšťané aktualizácie, čo znamená, že v prípade zmeny siete musíme čakať na 30 sekundový interval než sa začnú zmeny distribuovať. Preto siete s protokolom RIP konvergujú veľmi pomaly, a to je v dnešnej dobe neprípustné. Ďalší problém je, že masky sietí nie sú oznamované, pracuje sa len s triedami.

Nedostatky protokolu sa snaží riešiť jeho aktualizácia RIPv2 pridaním podpory pre protokoly CIDR a VLSM a ďalšími úpravami:

- aktualizácie sú odosielané pomocou viac-smerovej adresy 224.0.0.9 namiesto vše-smerového vysielania
- je možnosť pridať suseda, ktorému budú aktualizácie posielané jednosmerným vysielaním, čo prispieva k znižovaniu záťaže
- pridaná podpora overovania medzi smerovačmi

Protokol RIP patri medzi distance-vektor protokoly. To znamená, že pri určovaní najlepšej cesty k cieľu berie do úvahy iba počet preskokov a neuvažuje nad šírkou pásma. Preto by dátam po ním vybranej ceste mohlo dosiahnutie cieľa trvať zbytočne dlho. Maximálny počet preskokov v protokole

RIP je 15, preto cieľ vzdialený ďalej, je považovaný za nedosiahnuteľný. Vo verzii 2 bol tento limit zvýšení na 255.



Obrázok 24: porovnanie výberu najlepšej cesty DV a LS

Zadáme routerboardu siete, ktoré chceme propagovať:

```
/routing rip
/network add netwok=10.0.0.0/24
/network add netwok=192.168.0.0/24
```

Ďalej povolíme distribúciu pripojených sietí a statických ciest (ak ich chceme propagovať) :

```
/routing rip set redistribute-connected=yes
/routing rip set redistribute-static=yes
```

V tomto bode by sa nám už mali cesty propagovať medzi smerovačmi (ak ubehol aktualizačný interval) a presvedčiť sa o tom môžeme výpisom smerovacej tabuľky:

```
/ip route print
```

Ďalej zabezpečíme, aby sa nám cesty nešírili na rozhraniach, kde to nie je nutné:

```
/routing rip interface
add interface=ether3 passive=yes
```

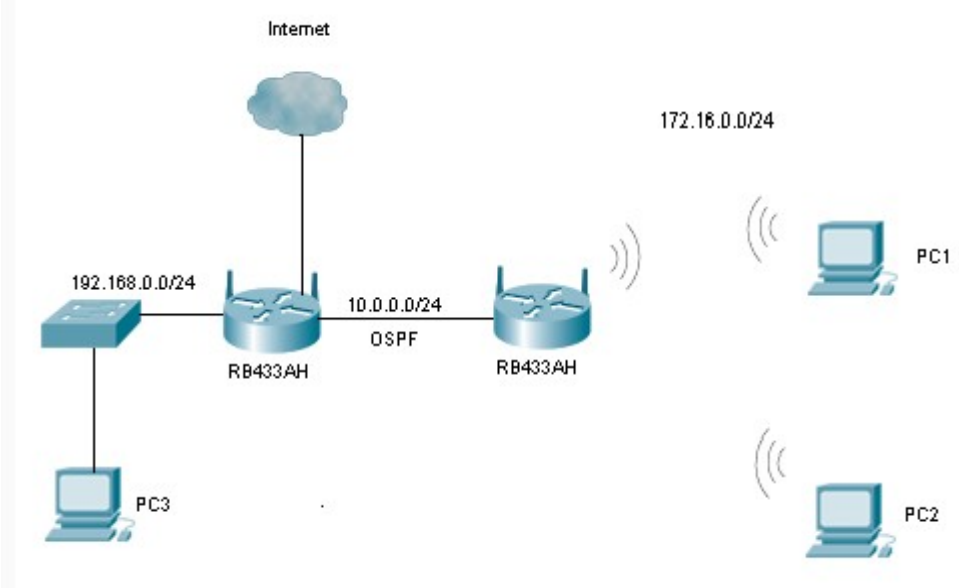
Nakoniec zabezpečíme pripojenie k internetu:

```
/ip dhcp-client add interface=ether3 disabled=no
/ip firewall nat add chain=srcnat action=masquerade
         out-interface=ether3
```

Po úspešnom absolvovaní úlohy, by mal študent získať základné znalosti z oblasti smerovania a smerovacích protokolov. Osvojí si vlastnosti protokolu RIP a jeho využitie. V praxi sa protokol RIP používa dnes už zriedka, ale môžeme ho využiť pri smerovaní v menších sieťach. Cvičenie časovo vychádza na 30 až 45 minút.

6.4 Úloha č.3

Úlohou študenta bude zoznámiť sa s pokročilejším smerovacím protokolom OSPF a zostaviť predloženú konfiguráciu siete.



Obrázok 25: požadované zapojenie úlohy č.3

K úlohe budeme potrebovať minimálne dva routerboardy a niekoľko počítačových staníc. Prvým krokom bude vytvorenie jednotlivých sietí

```
/ip address add address=ip adresa netmask=maska podsiete
         interface=fyzické rozhranie
```

Komunikáciu jednotlivých zariadení overíme príkazom ping. Ak nám zariadenia medzi sebou komunikujú, môžeme začať konfiguráciu dynamického smerovania.

Protokol OSPF patrí medzi link-state protokoly[1,6]. Pri výbere najvhodnejšej cesty berie do úvahy šírku pásma. Metrika jednotlivých spojení sa počíta ako 100 000 000/šírka pásma spojenia v b/s, čiže linke o rýchlosti 10MB/s bude pridelená metrika 10. Pre výpočet najkratších ciest sa používa Dijkstrov algoritmus. Protokol OSPF rozdeľuje siete na oblasti (areas). Oblasť 0 (backbone) je hlavná oblasť, s ktorou musia byť všetky ostatné prepojené. Smerovače sú tiež rozdelené podľa ich funkcie v sieti na:

- Interný smerovač – nachádza sa iba v jednej oblasti v jednom autonómnom systéme
- Oblastný smerovač – nachádza sa vo viacerých oblastiach v jednom autonómnom systéme
- Hraničný smerovač – spája niekoľko autonómnych systémov OSPF, alebo a.s. OSPF s a.s. využívajúcim iný smerovací protokolmi
- „Backbone“ smerovač – nachádza sa v oblasti 0
- Poverený smerovač - hlavný smerovač v oblasti, ktorý posiela aktualizácie ostatným smerovačom o cestách
- Záložný poverený smerovač – preberá úlohy povereného smerovača v prípade jeho zlyhania

Hlavnými prednosťami protokolu OSPF je, že nás núti členiť sieť do oblastí, čím sa stáva prehľadnejšou jeho rýchlosť konvergencie.

Vytvoríme novú oblasť, pokiaľ nechceme použiť oblasť 0:

```
/routing ospf area
add name=skola area-id=10.0.0.0
```

Zadáme routerboardu siete, ktoré chceme propagovať:

```
/routing ospf
/network add netwok=10.0.0.0/24 area=backbone
/network add netwok=192.168.0.0/24 area=backbone
```

Ďalej povolíme distribúciu pripojených sietí a statických ciest (ak ich chceme propagovať) :

```
/routing ospf instance set redistribute-connected=as-type-1
/routing ospf instance set redistribute-static=as-type-1
```

V tomto bode by sa nám už mali cesty propagovať medzi smerovačmi a presvedčiť sa o tom môžeme výpisom smerovacej tabuľky:

```
/ip route print
```

Ďalej zabezpečíme, aby sa nám cesty nešírili na rozhraniach, kde to nieje nutné:

```
/routing ospf interface
add interface=ether3 passive=yes
```

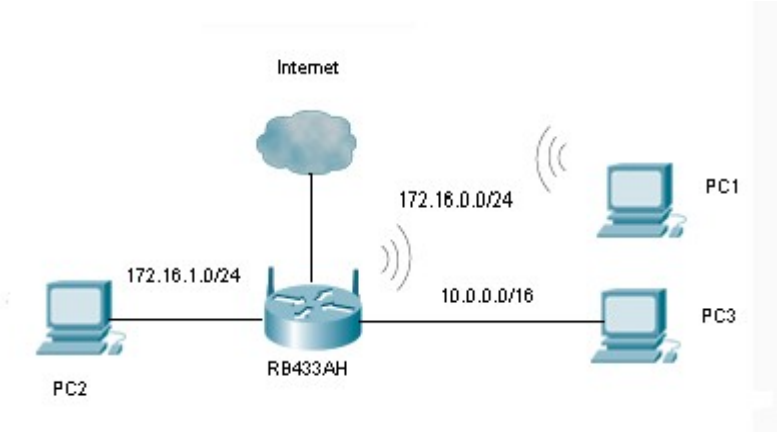
Nakoniec zabezpečíme pripojenie k internetu:

```
/ip dhcp-client add interface=ether3 disabled=no
/ip firewall nat add chain=srcnat action=masquerade
     out-interface=ether3
```

Po úspešnom absolvovaní úlohy, by mal študent získať základné znalosti z oblasti smerovania a smerovacích protokolov. Osvojí si vlastnosti protokolu OSPF a jeho využitie. V praxi sa protokol OSPF používa hlavne v sieťach, kde potrebujeme rýchlu konvergenciu a používame aj iné smerovače než od výrobcu cisco. Cvičenie časovo vychádza na 30 až 45 minút.

6.5 Úloha č.4

Úlohou študenta bude zoznámiť sa s fungovaním firewallu a nastavením jeho pravidiel.



Obrázok 26: požadované zapojenie úlohy č.4

Firewall slúži ako kontrolný bod, ktorý definuje pravidlá pre komunikáciu medzi sieťami.

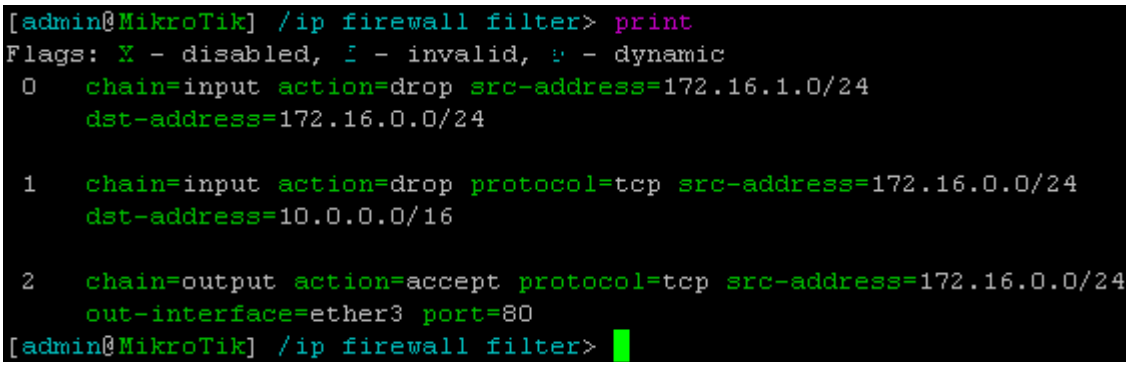
Pri vytváraní pravidla môžeme vychádzať z nasledujúcich parametrov:

- chain – výber postupnosti pri prechádzaní pravidlami bežne zvolíme imput, output, forward
- action – akcia vykonaná pri zhode s pravidlom
- dst-address – cieľová adresa
- dst-address-list – vopred vytvorený zoznam cieľových adries
- dst-address-type – typ cieľovej adresy (unicast, local, broadcast, multicast)
- dst-port – cieľový port
- src-address – zdrojová adresa
- src-address-list – vopred vytvorený zoznam zdrojových adries
- src-address-type – typ zdrojovej adresy (unicast, local, broadcast, multicast)
- src-port – zdrojový port
- in-interface – rozhranie, ktorým packet prišiel
- out-interface – rozhranie, ktorým packet odchádza
- protocol – použitý protokol

Vytvoríme niekoľko pravidiel pre demonštráciu:

```
/ip firewall filter
add chain=input dst-address=172.16.0.0/24 src-address=172.16.1.0/24
     action=drop
add  chain=input  dst-address=10.0.0.0/16  src-address=172.16.0.0/24
     action=drop protocol=tcp
add chain=output src-address=172.16.1.0/24 action=drop
     interface=ether3
```

- prvé pravidlo zakazuje komunikáciu od PC2 k PC1
- druhé pravidlo zakazuje komunikáciu tcp protokolom od PC1 k PC3
- tretie pravidlo zakazuje prístup k internetu pre PC2 na porte 80



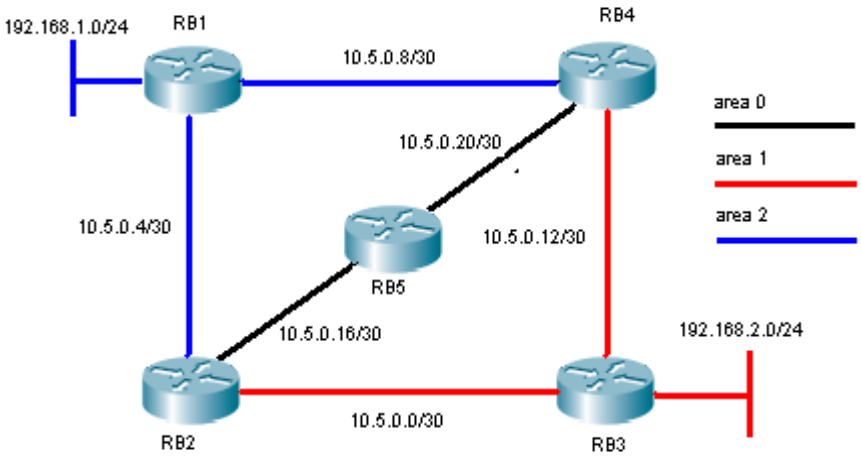
Obrázok 27: zoznam pravidiel firewallu

Pravidlá môžeme povoliť, zakázať, prípadne odobrať príkazmi enable, disable a remove.

Po úspešnom absolvovaní úlohy, by mal študent získať základné znalosti z oblasti bezpečnosti počítačovvých sietí. Osvojí konfiguráciu firewallu a jeho využitie. V praxi využije znalosti či už pri konfiguracii firewallu na domácom PC, alebo firemnom smerovači. Cvičenie časovo vychádza na 30 až 45 minút.

6.6 Úloha č.5

Úlohou študenta je zostaviť väčšiu sieť a použitím protokolu OSPF ju rozdeliť na niekoľko oblastí. Taktiež bude zoznámený s možnosťou zabezpečenia prenosu smerovacích informácii.



Obrázok 28: požadované zapojenie úlohy č.5

Konfigurácia jednotlivých smerovačov sa líši iba v rozlišných ip adresách a oblastiach, do ktorých patria, preto uvediem iba konfiguráciu smerovača RB2.

Pridelíme ip addresy rozhraniam:

```
/ip address add address=10.5.0.6/30 interface=ether1
/ip address add address=10.5.0.1/30 interface=ether3
/ip address add address=10.5.0.17/30 interface=ether2
```

Vytvoríme oblasti okrem oblasti 0, ktorá je už vytvorená:

```
/routing ospf area add area-id=0.0.0.1 name=area_1
/routing ospf area add area-id=0.0.0.2 name=area_2
```

Pridámé siete, ktoré chceme distribuovať:

```
/routing ospf network add network=10.5.0.4/30 area=area_2
/routing ospf network add network=10.5.0.0/30 area=area_1
/routing ospf network add network=10.5.0.16/30 area=backbone
```

Zabezpečíme prenášané smerovacie informácie:

```
/routing ospf  interface add interface=ether1 authentication=md5
     authentication-key=heslo
```

```
/routing ospf  interface add interface=ether2 authentication=md5
     authentication-key=heslo
/routing ospf  interface add interface=ether3 authentication=md5
     authentication-key=heslo
```

Ak sa nám rozdistribuovali smetovacie informácie, môžeme testovať rýchlosť konvergencie protokolu OSPF prerušovaním jednotlivých spojov. Testovať konektivitu a sledovať zmenu trasy paketov je možné pomocou príkazov:

```
/ping ip adresa
/tools traceroute ip addresa
```

Po úspešnom absolvovaní úlohy, by mal študent získať pokočilé znalosti z oblasti dynamického smerovania počítačových sietí. Osvojí metody autentizácie smerovacích informácii.V praxi sa protokol OSPF používa pri smerovaní v autonómnych systémoch. Cvičenie časovo vychádza 60 až 90 minút.

7 Záver

Routerboard mikrotik RB433AH nám poskytuje širokú škálu využitia, či už v bežnej domácnosti, väčšej firme, alebo pri výuke smerovania, smerovacích protokolov, bezpečnosti a bezdrôtových sietí.

V úvode práce som sa venoval hardwarovým parametrom zariadenia a jeho príslušenstvu. Ďalej bol predstavený originálny operačný systém RouterOS, jeho klady a zápory, vlastnosti a využitie. Pre splnenie požiadavky nahradenia originálneho systému som zvolil linuxovú distribúciu OpenWrt vhodnú predovšetkým pre vstavané zariadenia. Prezentoval som návody pre zostavenie a konfiguráciu systému, ako aj metódy importu výsledného systému do zariadenia. Nakoniec som vytvoril niekoľko úloh využiteľných pri praktickej výuke sieťových kurzov.

Práca poskytuje možnosť naviazať v podobe zostavenia ďalších cvičení využívajúcich pokročilé funkcie routerboardu. Ďalšou možnosťou je úprava systému OpenWrt tak, aby bol routerboard použiteľný napríklad ako databázový alebo webový server.

Domnievam sa, že som úspešne vypracoval zadanie práce a dúfam, že nadobudnuté znalosti z oblasti počítačových sietí a nasadenia alternatívnych operačných systémov na vstavané zariadenia, využijem aj v budúcnosti.

Literatúra

[1] DONAHUE, Gary A. *Kompletní průvodce síťového experta*. Brno : Computer press, 2009. 528 s. ISBN 978-80-251-2247-1.

[2] ŠTRAUCH, Adam. Mikrotik: seznámení s Wi-Fi krabičkou. *Root.cz* [online]. 7. 11. 2008, [cit. 2010-05-01]. Dostupný z WWW: <http://www.root.cz/clanky/mikrotik-seznameni-s-wi-fi-krabickou/>

[3] *RIP* [online]. 30.4.2009 [cit. 2010-05-10]. MikroTik wiki. Dostupné z WWW: <http://wiki.mikrotik.com/wiki/Manual:Routing/RIP>

[4] Wired Equivalent Privacy In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010, [cit. 2010-05-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Wired_Equivalent_Privacy>

[5] *BuildRoot* [online]. 1999 [cit. 2010-05-14]. Dostupné z WWW: <http://www.buildroot.org/>

[6] *OSPF* [online]. 30.4.2009 [cit. 2010-05-10]. MikroTik wiki. Dostupné z WWW: <http://wiki.mikrotik.com/wiki/Manual:Routing/OSPF>

[7] ŠTRAUCH, Adam. Plnohodnotný Linux na RouterBoardu s OpenWRT. *Root.cz* [online]. 14. 7. 2009, [cit. 2010-04-15]. Dostupný z WWW: <http://www.root.cz/clanky/plnohodnotny-linux-na-routerboardu-s-openwrt/>

[8] Wi-Fi Protected Access In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010, [cit. 2010-05-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access>

[9] *Metarouter* [online]. 14.4.2009 [cit. 2010-05-10]. MikroTik wiki. Dostupné z WWW: <http://wiki.mikrotik.com/wiki/Manual:Metarouter>

[10] *OpenWrt* [online]. 2010 [cit. 2010-05-01]. Dostupné z WWW: <http://openwrt.org/>

[11] *Quagga documentation* [online]. 2006 [cit. 2010-05-11]. Dostupné z WWW: <http://www.quagga.net/docs/quagga.pdf>

[12] MIPS architecture In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010, [cit. 2010-05-10]. Dostupné z WWW: <http://en.wikipedia.org/wiki/MIPS_architecture>

[13] *Linux from scratch* [online]. 2010 [cit. 2010-05-01]. Dostupné z WWW: <http://www.linuxfromscratch.org/>

[14] Network address translation In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010, [cit. 2010-05-17]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Network_address_translation>.

Príloha A

Na priložemon dvd nájdeme v priečinkoch pomenovaných podľa jednotlivých úloh uložené konfigurácie routerboardov.