Přírodovědecká fakulta Jihočeské univerzity

Bakalářská práce:

TVORBA ROUTERU NA BÁZI LINUXU

Vypracoval: Jiří Suda

Školitel: Mgr. Jiří Pech Ph.D.

České Budějovice 2011

Bibliografické údaje

Suda J., 2011: Tvorba routeru na bázi Linuxu.
[Creation of router based on Linux. B.A. Thesis, in Czech.] – 35 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Práce popisuje pojem router, existující druhy routerů a následnou tvorbu vlastního routeru na bázi osobního počítače s nainstalovanou linuxovou distribucí. Hodnotí výhody a nevýhody tohoto řešení a podrobně je popisuje. Taktéž se věnuje administraci a zabezpečení routeru.

Abstract

This thesis describes routers in general and their variants as well as the creation of my own PC issues linux-based router variant. It states pros and cons of this solution in detail. The administration and security are also mentioned.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval/a samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské, a to v nezkrácené podobě – v úpravě vzniklé vypuštěním vyznačených částí archivovaných Přírodovědeckou fakultou - elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

Datum 1.3.2011

Podpis

Poděkování

Chci poděkovat svému školiteli panu **Mgr. Jiřímu Pechovi Ph.D.** za veškerou pomoc, kterou mi poskytl.

Obsah

1 Úvod

Čistě směrovat pakety umí dnes mnoho druhů síťových zařízení. Liší se především různým druhem nasazení. Od profesionálního nasazení ve velkých infrastrukturách poskytovatelů Internetu, kde se používají výkonné routery například od společností Cisco či Foundry Networks, až po malé domácí routery sloužící většinou pouze k vzájemnému propojení počítačů a poskytnutí přístupu na internet.

Tato práce se bude zabývat zajímavým druhem routeru a to osobním počítačem s operačním systémem Linux. Tento způsob oproti ostatním vyniká především svojí vysokou konfigurovatelností a velkými možnostmi. Je tak navržen router s možnostmi, kterými běžné routery nedisponují, ať už z oblasti přidané funkcionality nebo zvýšeného zabezpečení, protože router dnes může plnit celou řadu jiných funkcí. Plnohodnotný operační systém poskytuje především výhody v ohromné variabilitě a možnostech nastavení. V dnešní době není problém pořídit levný dvoujádrový procesor s vysokým výpočetním výkonem, který poskytne další výhody.

Router bude srovnán především s domácími routery, za které poskytuje smysluplnou náhradu a dokonce je převyšuje. Na trhu totiž žádné podobné zařízení není. Tato práce tak může být návodem, jak takové zařízení vytvořit a ukázat jeho možnosti v novém a místy netradičním pohledu.

V druhé kapitole jsou definovány cíle práce, na které navazuje kapitola tři pojednávající o metodice. Kapitola čtyři zasvěcuje čtenáře do problematiky síťové komunikace a různých útoků, na které je později v práci odkazováno. V kapitole pět jsou srovnány jednotlivé druhy routerů a popisovány jejich vlastnosti. Šestá část této práce se již zabývá vlastním návrhem routeru. Jednotlivé funkce jsou vysvětleny a je zde prakticky předvedena konfigurace. Sedmá kapitola probírá testy prováděné na routeru. V předposlední části je navrhnuto možné využití již dokončeného routeru s ohledem na disponované funkce, které by v daném prostředí byly nejvíce užitečné. Závěrem je zhodnocena tato práci a srovnán navrhovaný router, v podobě osobního počítače s nainstalovaným plnohodnotným operačním systémem *Debian*, s domácím routerem.

2 Cíle práce

Cíle této práce jsou:

- Vysvětlit pojem router a jeho princip. Podat ucelený pohled na problematiku routerů – popsáním jednotlivých druhů a určení výhod a nevýhod použití osobního počítače
- Prakticky vytvořit z osobního počítače router na bázi operačního systému Linux. Určit výhody a nevýhody tohoto řešení. Podrobně toto řešení popsat
- Administrace a zabezpečení tohoto routeru
- Uvést praktické příklady možného využití navrhovaného routeru v praxi pro různé scénáře využití

3 Metodika

3.1 Výběr distribuce

Výběr distribuce je důležitá věc, kterou není dobré zanedbat. Špatně zvolená distribuce nemusí být například dostatečně stabilní, zabezpečená proti útokům anebo nemusí mít všechny potřebné funkce, které požadujeme. Distribuce nasazená na router byla vybrána po konsultaci školitelem. Je jím *Debian* ve verzi *Squeeze*, který je známý svojí stabilitou, zabezpečením, jednoduchou údržbou a celkovým zaměřením na tyto účely. Existují tři větve vývoje *Debianu*: stable, testing a unstable. Byla použita verze stable, která je vhodná pro ostré nasazení na servery, zejména kvůli svému opravdu důkladnému otestování. Podpora od vývojářů je 3 roky s tím, že od verze *Squeeze* bude prodloužena na 5 let. Další výhodou je, že okolo této distribuce je velká komunita lidí, kteří rádi pomohou.

3.2 Výběr nástrojů

Nástroje a funkce, které má navržená distribuce byly vybrány s ohledem na současné vybavení domácích routerů, především s porovnávaným routerem *Asus WL-520gU*, který byl k dispozici pro srovnání a zároveň patří mezi lepší typy v jeho kategorii. Jednoduše bylo uvažováno, jaké další možnosti tyto malé routery nemohou mít a zároveň by byly zajímavé s ohledem na praktickou využitelnost. Protože distribuce *Debian* obsahuje velké množství balíčků s programy, využívalo se právě jich, neboť již jsou perfektně otestovány na funkčnost a stabilitu. Inspirace byla čerpána z literatury a autorových praktických zkušeností.

3.3 Testovací nástroje datového toku

Pro ověření funkčnosti pokročilé kontroly datového toku, jako je přidělování minimálního datového toku, priorizace paketů, spravedlivé rozdělení pásma mezi různá spojení a dynamické přelévání volného datového toku bylo potřeba vytvořit test. Pro tento test byly primárně využívány dva protokoly schopné společně vytvořit různé podmínky simulující reálný provoz. Protokol *FTP*, známý svou agresivitou připojení a protokol *http*. Pro přenos *FTP* spojení se využíval zabudovaný *FTP* klient v programu *Total Commander*, který dokáže zobrazovat aktuální rychlost stahování, souběžně se spuštěným testem aktuálních parametrů internetového připojení dostupného z internetového portálu *rychlost.cz*. Měřena byla webová odezva (minimální, průměrná a maximální), rychlost stahování a nahrávání dat. Ukázka testu je vidět na obrázku 7.

4 Úvod do problematiky

4.1 Síťové protokoly TCP/IP

Pro komunikaci počítačů v rozsáhlé internetové síti se používají převážně protokoly *TCP/IP*. Celkově je tato problematika složitá a proto se používá k jejímu pochopení rozdělení komunikace na jednotlivé vrstvy, které jsou vidět na obrázku 1. Protokoly na jednotlivých vrstvách spolu komunikují teoreticky tak, že využívají služeb nižších vrstev. Je to podobné, jako když se telefonuje telefony, lidé mluví spolu, ale prakticky je jejich hlas přenášen po telefonní lince.

Obrázek 1

Fyzická vrstva

Fyzická vrstva je nejníže v modelu *TCP/IP* a je jí myšlena hardwarová část komunikace. Má na starosti způsob přenosu elektrických, elektromagnetický a optických signálů mezi zařízeními. Na této vrstvě operuje zařízení nazvané *hub* či *opakovač*, které přeposílá všechny signály na všechny své porty.

Linková vrstva

Druhá vrstva je Linková vrstva, která je již softwarová a propojuje počítače v jedné síti. Přenáší se po ní linkový rámec, což jsou data přenášená v blocích opatřená linkovou adresou příjemce a kontrolním součtem pro detekci chyb. Linkovou vrstvu využívá ke komunikaci zařízení *switch* taktéž *přepínač*, který přeposílá data na základě *MAC* adresy. V lokálních sítích má každý počítač svoji unikátní linkovou adresu neboli *MAC*, kterou představuje 48 bitů.

IP vrstva
Třetí v pořadí je IP vrstva neboli síťová vrstva. Paket na této vrstvě se nazývá IP datagram a je vložen do datové části linkového rámce. V případě protokolu IPv4 je tvořena IP adresou v podobě 32 bitů. Využívají ji *routery* neboli *směrovače* k přepravě paketů mezi různými sítěmi.

Vrstva TCP/UDP
Vrstva TCP/UDP je zapouzdřena v IP datagramu. Stará se o spojení mezi vzdálenými počítači. Aby mohla rozlišit komunikaci určenou pro jednotlivé aplikace, používá porty, které jednoznačně určí, které aplikaci konkrétní spojení patří. Jak TCP tak UDP mají své vlastní porty. V případě protokolu TCP se jednotlivé TCP segmenty potvrzují, takže jde o spolehlivý protokol, kdežto UDP diagramy většinou nepotřebují být kontrolovány a jsou pouze odeslány bez ověření doručení.

Aplikační vrstva
Aplikační vrstvou jsou myšleny protokoly využívající jednotlivé aplikační programy pro svoji vzájemnou komunikaci. Mezi ně patří například protokoly *HTTP, FTP, POP3, TELNET*.

4.2 Síťové a jiné útoky

Odposlouchávání
Odposlouchávání se myslí zachytávání všech dat na síti, hlavně těch, která nejsou určena pro odposlouchávající počítač. Síťové odposlouchávání může být použito například k analýze síťového provozu nebo k odhalení trojského koně, ale i k získání přístupu k cizí komunikaci. Pro zachycení dat je nutné použít nějaký program, vhodný je například *Wireshark*.

ARP Cache poisoning
Protokol ARP operující na druhé vrstvě síťového modelu slouží ke zjištění MAC adresy na základě znalosti IP adresy. Pošle se dotaz ARP Request s žádostí o překlad IP adresy a cílový počítač pošle odpověď ARP Reply s MAC adresou. ARP Reply by se měl normálně posílat pouze na základě ARP Requestu, ale jelikož byl vyvíjen v dobách, kdy na bezpečnost nebyl kladen důraz, tak není nijak zabezpečen. Proto můžeme snadno poslat paket ARP Reply i bez toho, že by se na překlad dotyčný počítač ptal. V ARP Reply říkáme, aby si cílový počítač zapsal útočníkovu MAC adresu místo MAC adresy počítače, na který má být namířen odposlech. Jakmile útočník odposlechne paket, tak ho přepošle na cílový počítač s již pravou MAC adresou. Tímto způsobem je možné obousměrně a nepozorovaně odposlouchávat komunikaci. Dokonce zároveň veškerou komunikaci na celé síti. Jelikož jsou data v ARP Cache uchovávána jenom po krátkou dobu, je třeba tento útok obnovovat. ARP Cache poisoning s přehledem zvládá program *Ettercap*.

DHCP Spoofing
Základem DHCP Spoofingu je nahradit DHCP server za svůj vlastní a podstrčit oběti nepravé údaje jako je výchozí brána a DNS server. Díky tomu je možno odposlouchávat, protože je tímto komunikace přesměrována skrze útočníka. Je potřeba vyřadit regulérní DHCP server

z provozu, a to například zahlcením při úplně novém DHCP dotazu a následném podstrčení oběti svojí odpovědi nebo v případě, že oběť již má přidělenou adresu vypotřebováním IP adres, které může přidělit a tím ho umlčet. Pak stačí počkat, než vyprší doba platnosti IP adresy a přidělit podvrženou.

Cold boot útok

Pokud získá útočník fyzický přístup k zašifrovanému běžícímu počítači, může použít Cold boot útok. Využívá znalosti, že šifrovací klíče jsou uloženy v operační paměti RAM, kterou pokud zchladíme na nižší teplotu, tak se data v paměti uchovají v závislosti na této teplotě i několik minut, potřebných ke zkopírování dat na jiný počítač nebo k nabootování vlastního operačního systému na napadený počítač.

Brute force útok

Tento útok jednoduše zkouší veškeré kombinace uživatelských jmen a hesel. Pokud je k dispozici zahashované heslo, tak se provádí hash všech různých kombinací a porovnává na shodu se skutečným zahashovaným heslem. Spoléhá se především na to, že uživatelé zvolí jednoduchá hesla. Účinnější obdobou je použití předem připraveného slovníku s nejčastěji používanými slovy a hesly.

5 Router

5.1 Pojem router

Router má jak už název napovídá základní úkol a tím je právě routování. Ve své podstatě se jedná o dvou nebo více portový počítač většinou se specializovaným operačním systémem, který může nabývat různých velikostí a tvarů od malých krabiček přes osobní počítače až po velké routery v datových centrech. Čím profesionálnější použití tím může disponovat různými úpravami jako je používání flash pamětí místo pevných disků. Pracuje na třetí vrstvě síťového modelu. Jedná se tedy o aktivně pracující prvek pro spojení jednotlivých sítí mezi sebou s definovanými vlastnostmi předávání.

5.2 Routování

Routování znamená předávání IP datagramů na třetí síťové vrstvě mezi jednotlivými sítěmi z jednoho svého rozhraní, ze kterého přišel IP diagram do jiného rozhraní. Pokud tedy nějaký počítač pošle IP datagram, tak je routery mezi sebou předáván do té doby, dokud není s cílovým počítačem v jedné síti. Toto předávání se uskutečňuje pomocí směrovacích tabulek, které obsahují záznamy o hierarchii sítí. Každé routování neboli hop by mělo posunout datagram blíže k cíli. Jakmile je paket v stejné síti je předán pomocí nižších vrstev cílovému. K tomu je potřeba znát MAC adresu cílového počítače. Pokud není uložena v ARP cache, tak se vyžádá pomocí protokolu ARP.

5.3 Druhy routerů

Profesionální routery

Nejznámější a nejrozšířenější společnost Cisco a její routery patří mezi absolutní špičku v síťových technologiích s velkým rozšířením ve velkých síťových infrastrukturách poskytovatelů Internetu, univerzit, velkých firem a vědeckých pracovišť. Nabízejí všechny druhy síťových zařízení. Charakteristický je pro ně použitý operační systém Cisco IOS, který se ovládá příkazovou řádkou. Nejsou vhodné pro začátečníky, protože jsou vysoce specializované a zaměřené pouze na směrování, jeho vysokou bezpečnost a síťovou efektivitu. Nelze je tudíž využít pro nějakou další činnost, která od nich ani není očekávána. Existují i další méně známé firmy jako je Foundry Networks. V domácím prostředí jsou profesionální routery prakticky nepoužitelné.

Domácí routery

Jedná se o multifunkční zařízení nasazované povětšinou do domácností a velmi malých firem. Většinou disponují dalšími přídavnými funkcemi. Ze základních funkcí jsou to:

- Čtyřportový switch kombinovaný s jedním WAN portem
- DHCP server pro přidělování informací potřebných pro připojení k Internetu
- NAT překládající vnitřní IP adresy na jednu odchozí IP adresu.
- Jednoduché ovládání prostřednictvím internetového prohlížeče
- Aktualizace firmware opravující chyby
- Kompaktní rozměry

Mezi pokročilejší funkce patří:

- Integrovaný WiFi modul pro bezdrátové připojení se standardní frekvencí 2,4Ghz nebo pokročilejší 5Ghz
- USB porty, pro připojení tiskárny nebo připojení externích disků pro sdílení dat
- FTP server pro sdílení dat
- Integrovaný softwarový firewall s možností blokování URL či přístupu MAC adres
- Možnost nahrát neoficiálně podporovaný firmware od fanoušků, přidávající většinou další funkcionalitu

Tato zařízení jsou zajímavá především svojí cenou, časovou nenáročností, spotřebou okolo 10 Wattů a většinou dostačující funkcionalitou. Na většinu těchto zařízení je možné nahrát alternativní firmware. Nejvýznamnější *OpenWRT* je minimalistický operační systém Linux používaný většinou na domácích routerech, které mají velmi malou operační paměť a slabý procesor v porovnání s osobními počítači. Díky vysoké optimalizaci a předem známému konkrétnímu nasazení, zvládnou podat dobré výkony (s dobrou funkcionalitou), za které je nutno zaplatit v podobě složité konfigurace.

Osobní počítač se specializovanou linuxovou distribucí

Jakýkoliv osobní počítač je možno přeměnit na router. Existují pro to různé předpřipravené linuxové distribuce s jednoúčelovým operačním systémem. Jako příklad uvedu distribuci *IPFire* podrobně popisovanou v [6], která patří mezi nejlepší, především pro svoji rychlou instalaci, spojenou s jednoduchostí a vysokou bezpečností. Ke konfiguraci této distribuce se používá webové prostředí. Síťová rozhraní mají pevně určený počet a využití. Jedno rozhraní směrem do Internetu, druhé rozhraní je určeno pro lokální síť, třetí WiFi a čtvrté do demilitarizované zóny. Tato distribuce zahrnuje většinu funkcí domácích routerů a přidává nějaké navíc jako je program *Snort* pro detekci vniknutí do systému.

Z hlediska spotřeby není situace tak špatná. Například u dvoujádrového procesoru *Intel Pentium Dual Core* s taktem 3Ghz se spotřeba celé sestavy pohybuje okolo 35Wattů v klidovém stavu a 50Wattů v plném zatížení. Na trhu ovšem existuje i řada úspornějších procesorů. Současný trend ve výrobě procesorů naznačuje, že situace se bude neustále zlepšovat.

6 Tvorba a konfigurace routeru

6.1 Instalace

Základem je klasická linuxová 64 bitová distribuce *Debian 6*. Instalace pomocí instalátoru je opravdu jednoduchá. Pro nás jsou v instalaci zajímavé zejména tyto dvě možnosti:

Zašifrování disku

Při rozdělení disku je možné zvolit následující možnost:

Asistované – použít celý disk a nastavit šifrované LVM

Disk se zašifruje a při zapínání bude potřeba zadat heslo. Může to být užitečné, pokud by někdo počítač ukradl a byla by na něm cenná data. Existují i jiné sofistikovanější nástroje na šifrování disků jako například *TrueCrypt* nebo *FreeOTFE*. Bohužel šifrování nenahrazuje fyzické zabezpečení proti Cold boot útoku.

Grafické uživatelské prostředí

Většina podobných zařízení používá ke svému ovládání webové rozhraní. Pro použití tohoto routeru se zaškrtne k instalaci grafické prostředí, které bude sloužit k ovládání routeru. Důvodů je několik:

- Vysoká konfigurovatelnost – nelze vytvořit webové rozhraní z důvodu, že nevíme, co přesně na routeru poběží. Můžou to být různé služby
- Uživatelská přívětivost – grafické prostředí je mnohem intuitivnější a jednodušší na ovládání, zvláště pro začátečníky v porovnání s příkazovou řádkou
- Poslouží k ovládání virtualizovaných systémů

6.2 Nastavení síťových rozhraní

Většina síťových útoků, především odposlechů je prováděna na lokální síti. Takže pokud se tento router nastaví tak, že na každém rozhraní bude mít jinou síť tak je zamezeno většině těchto útoků, jako jsou ARP Cache poisoning, DHCP Spoofing.

Pro výpis názvů připojených rozhraní (eth0-9) a ověření správného nastavení přiřazení IP se používá příkaz:

```
ip address show
```

Do konfiguračního souboru *interfaces*, který je k nalezení v */etc/network/interfaces* se takto nastaví síťová rozhraní, která nesměřují do internetu:

```
auto eth0
iface eth0 inet static
address 192.168.1.200
netmask 255.255.255.0
```

Stejně se nastavují všechny ostatní síťové karty. Každé rozhraní je ovšem nastaveno do jiné sítě. Pro přehlednost se pokračuje inkrementálně. Takže další adresa bude 192.168.2.200.

6.3 DHCP server

Pro automatické přidělování síťových adres počítačům je zapotřebí nakonfigurovat DHCP server. Nastavení se provádí v */etc/dhcp/dhcpd.conf*. Tento soubor nastavuje rozsah přidělovaných adres analogicky pro každou síť zvlášť:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
     range 192.168.1.1 192.168.1.199;
     option routers 192.168.1.200;
     option broadcast-address 192.168.1.255;
     option domain-name-servers 192.168.1.200;

}
```

V konfiguračním souboru se nastaví na 3 dny parametr *max-lease-time*, který znamená maximální propůjčenou dobu IP adresy. Údaj je udáván v sekundách. Toto zaručuje ochranu proti DHCP Spoofingu.

V souboru */etc/default/isc-dhcp-server* se definuje proměnná *Interfaces* pro všechna rozhraní. Parametr -4 je důležitý, protože jinak se DHCP server nespustí s protokolem IPv4, ale implicitně s IPv6.

```
INTERFACES=" -4 eth0 eth1"
```

Jakmile je hotovo je potřeba server restartovat, aby si načetl nové nastavení. Použije se příkaz:

```
/etc/init.d/isc-dhcp-server restart
```

Veškeré záznamy o propůjčení IP adres od DHCP serveru se ukládají na disk do souboru */var/lib/dhcp/dhcpd.leases.*

6.4 Povolení routování

K tomu, aby router mohl vykonávat svoji nejzákladnější funkci, kterou je předávání paketů z jednoho rozhraní do jiného rozhraní neboli routovaní je třeba odkomentovat v konfiguračním souboru */etc/sysctl.conf* tento řádek:

```
net.ipv4.ip_forward=1
```

6.5 DNS server

Úkolem DNS serveru je překládat jména na IP adresy, které se lépe pamatují. Výhoda vlastního DNS serveru je to, že si udržuje DNS cache a při opakovaném dotazu vezme údaj z cache. Samozřejmě je tady možnost pojmenovat servery vlastní, která ovšem nebude asi moc využívána.

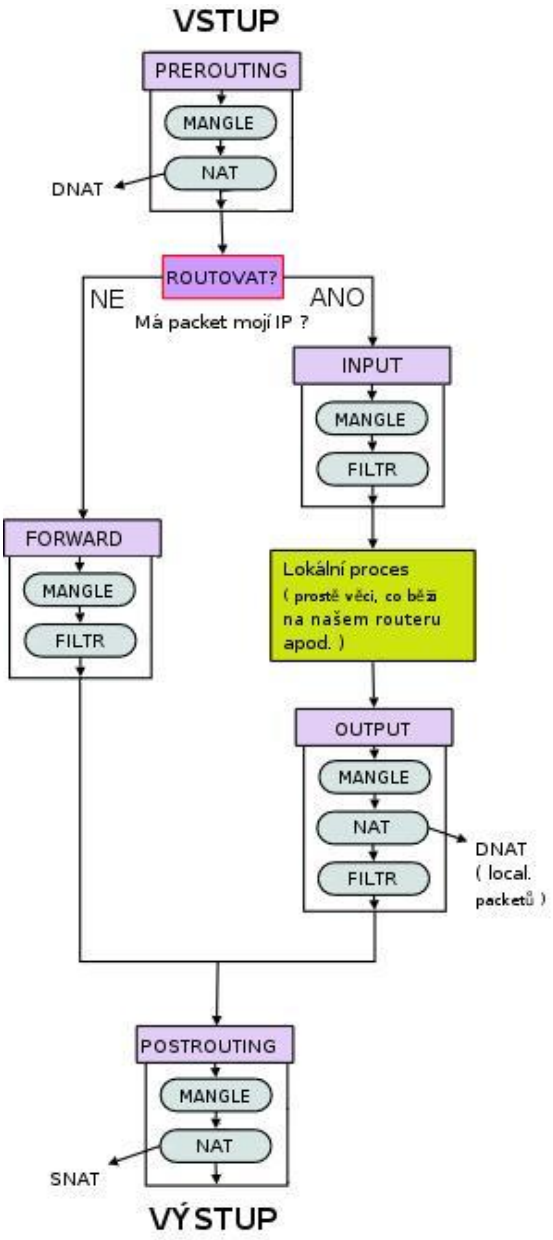
Nainstaluje se program *bind9*, což je nejpoužívanější DNS server pro Linux:

```
apt-get install bind9
```

Konfigurační soubor je umístěn v */etc/bind/named.conf.options*. Do něj se nastaví parametr *forwarders* na IP adresu serveru našeho poskytovatele internetového připojení. Počítače se o něm dozvědí, pokud požádají o IP adresu DHCP serveru, s kterou se jim nastaví i tento DNS server.

6.6 Netfilter

Netfilter je sada nástrojů pro filtrování paketů. Nejznámější z nich je nástroj *iptables*, který se používá pro filtrování povolených a předem definovaných paketů na třetí vrstvě síťového modelu, označování paketů a překlad adres NAT. Na převzatém [9] a upraveném obrázku 2 je vidět zpracování paketu při průchodu *iptables.*



Obrázek 2

Při vstupu paketu do *netfilteru*, je nutné nejprve projít řetězcem PREROUTING. Pokud je paket určen tomuto routeru, vstupuje do řetězce INPUT a následně je předán konkrétnímu procesu, případně odeslán paket nový podle řetězce OUTPUT. Pokud je paket určen někomu jinému předá se do řetězce FORWARD. Všechny odchozí pakety ještě projdou řetězcem POSTROUTING.

Řetězce mohou obsahovat tyto tabulky:

- Mangle – slouží k označkování paketů například pro program *tc*
- NAT – překládá adresy na cílové či zdrojové z lokálních nebo odchozích adres
- Filtr – obsahuje jednotlivá pravidla pro filtrování komunikace

Pro vlastní realizaci je potřeba vytvořit script *firewall.sh*, který bude využit k přidání pravidel pro filtrování, značkování a překlad adres NAT. Tento soubor se vytvoří například v */sbin/firewall.sh* a na první řádek se napíše, že jde o script:

```
#! /bin/bash
```

Tento soubor se musí označit jako spustitelný pomocí příkazu:

```
chmod 700 /sbin/firewall.sh
```

Aby se script firewall.sh spustil automaticky po zapnutí systému je třeba do souboru */etc/network/interfaces* vložit k rozhraní směřujícímu do internetu tuto řádku:

```
pre-up /sbin/firewall.sh
```

Stavový firewall

Podobá se paketovému firewallu, ale navíc je obohacen o stavové filtrování portů. Jedná se o základní obranu sítě. Komunikace směrem z Internetu se zakáže, až na vysloveně povolené služby. Základní politika se nastaví pro FORWARD a INPUT na zahazování paketů. Řetězec OUTPUT bude povolen.

```
iptables –P INPUT DROP
iptables –P OUTPUT ACCEPT
iptables –P FORWARD DROP
```

Povolí se místní smyčka, která je důležitá pro některé systémové funkce:

```
iptables –A INPUT –i lo –j ACCEPT
```

Pakety, které mají již navázané spojení, budou propuštěny:

```
iptables –A INPUT –m state --state ESTABLISHED, RELATED
     -j ACCEPT
iptables –A FORWARD –m state --state ESTABLISHED, RELATED
     –j ACCEPT
```

Pro správné fungování DNS serveru je třeba povolit komunikaci na portu 53 pro lokální sítě:

```
iptables –A INPUT –p udp –i eth0 --dport 53 –j ACCEPT
iptables –A INPUT –p udp –i eth1 --dport 53 –j ACCEPT
```

Veškerý přístup se bude logovat:

```
iptables –A INPUT –j LOG
```

Komunikace z vnitřní sítě bude povolena:

```
iptables –A FORWARD –s 192.168.1.0/24 –i eth0 –j ACCEPT
iptables –A FORWARD –s 192.168.2.0/24 –i eth1 –j ACCEPT
```

Je třeba ještě povolit všechny služby běžící na virtualizovaném systému jako například webový server:

```
iptables –A FORWARD –p tcp –o vboxnet0 --dport 80 –j
    ACCEPT
```

Dále je ještě třeba nastavit přesměrování z veřejné IP adresy na virtualizovaný server s webovým serverem:

```
iptables –t nat –A PREROUTING –p tcp –d 192.168.0.53
    --dport 80 –j DNAT 192.168.3.100
```

Překlad adres NAT

Z důvodu akutního nedostatku volných IPv4 adres byla vyvinuta technologie pro překlad adres NAT. Funguje tak, že IPv4 adresa přidělená od poskytovatele připojení je nasazena na stroj, který připojuje počítače k Internetu. Tento stroj překládá IPv4 adresy z vnitřní sítě na IP adresu pro vnější síť, tedy Internet. Z pohledu Internetové sítě se jeví veškerá komunikace z vnitřní sítě do vnější sítě jako vedená z jednoho počítače.

Do scriptu firewall.sh se vloží tento řádek s parametrem *–o*, určujícím výstupní rozhraní, který nastavuje překlad adres NAT pomocí programu *iptables*:

```
/sbin/iptables –t nat –A PREROUTING –o eth2 –j MASQUERADE
```

6.7 Přidělování datového toku & QoS

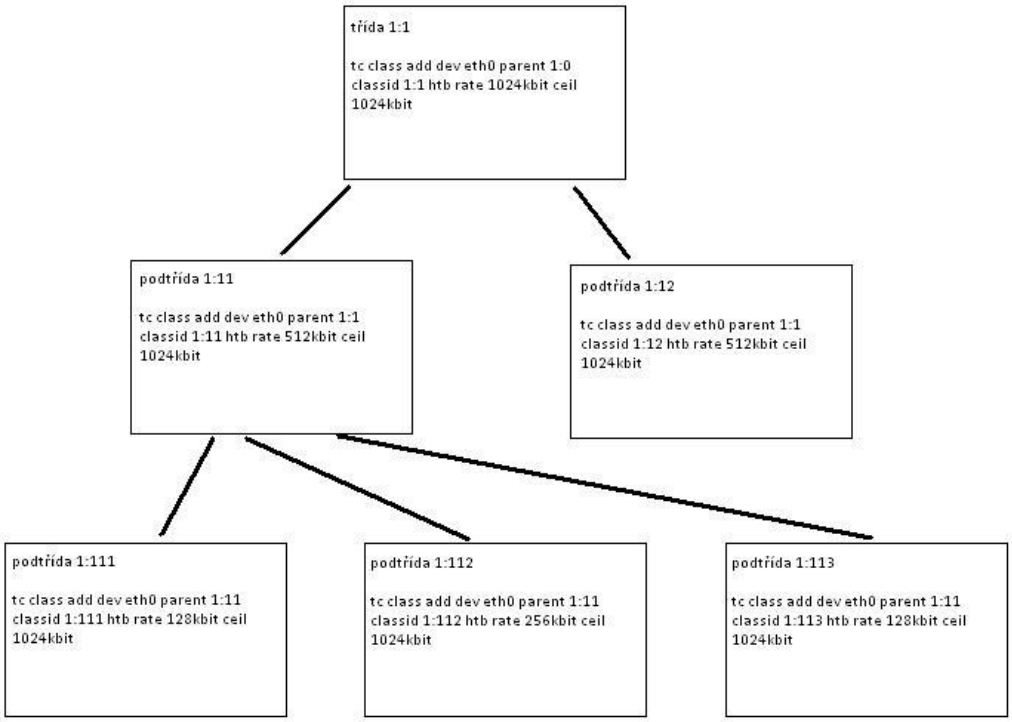
Za velký problém u sdíleného připojení k Internetu se považuje přerozdělování datového toku. Velice často se stává, že jeden uživatel společného připojení začne stahovat velké množství dat a tím razantně omezí ostatní uživatele. Někdy je dokonce zpomalení až za hranicí rozumné použitelnosti jejich připojení jak je psáno v [2].

V Linuxu se na tento problém používá program *tc* (traffic control) z balíku *iproute2*. Ten využívá principu návrhu TCP protokolu, který odesílá pakety rychlostí závislou na rychlosti zpětně obdržených potvrzovacích paketů. Takže pokud program *tc* uměle pozdrží tyto pakety před odesláním, odesílatel automaticky zpomalí rychlost odesílání. Je důležité si uvědomit, že omezovat připojení je možné pouze na rozhraní v odchozím směru.

Jedna možnost je napevno přidělit rychlosti na každé rozhraní bez možnosti využití nevyužívaného datového toku nebo lépe přidělit minimální zaručené rychlosti a propůjčovat nevyužitý tok ostatním rozhraním. Toto umožňuje řadič paketů HTB. Tento řadič dokáže jakkoliv řadit pakety v závislosti na jejich označení. Pro rozlišení paketů se používá označkování pomocí programu i*ptables* v tabulce *mangle*. Označkování je platné jenom v rámci jednoho stroje. Neznačkuje se přímo do paketů.

Na každé rozhraní je přiřazeno minimálně jedno řadící pravidlo *qdisc* (queuing discipline), které se může dělit na třídy *class* umožňující další podrobnější nastavení. Celé HTB je koncipováno jako stromová datová struktura. Takže každé pravidlo může mít pouze jednoho rodiče a neomezené množství potomků. Na kořen stromu je připojeno rozhraní. A na listy stromu jsou aplikovány jednotlivé filtry.

Minimální zaručený datový tok, který je přidělen všem potomkům jednoho rodiče, nemůže přesáhnout celkový minimální datový tok (rate) rodiče (obrázek 3), jinak řadič HTB nebude spolehlivě fungovat. Toto platí i pro celkový maximální tok (ceil).



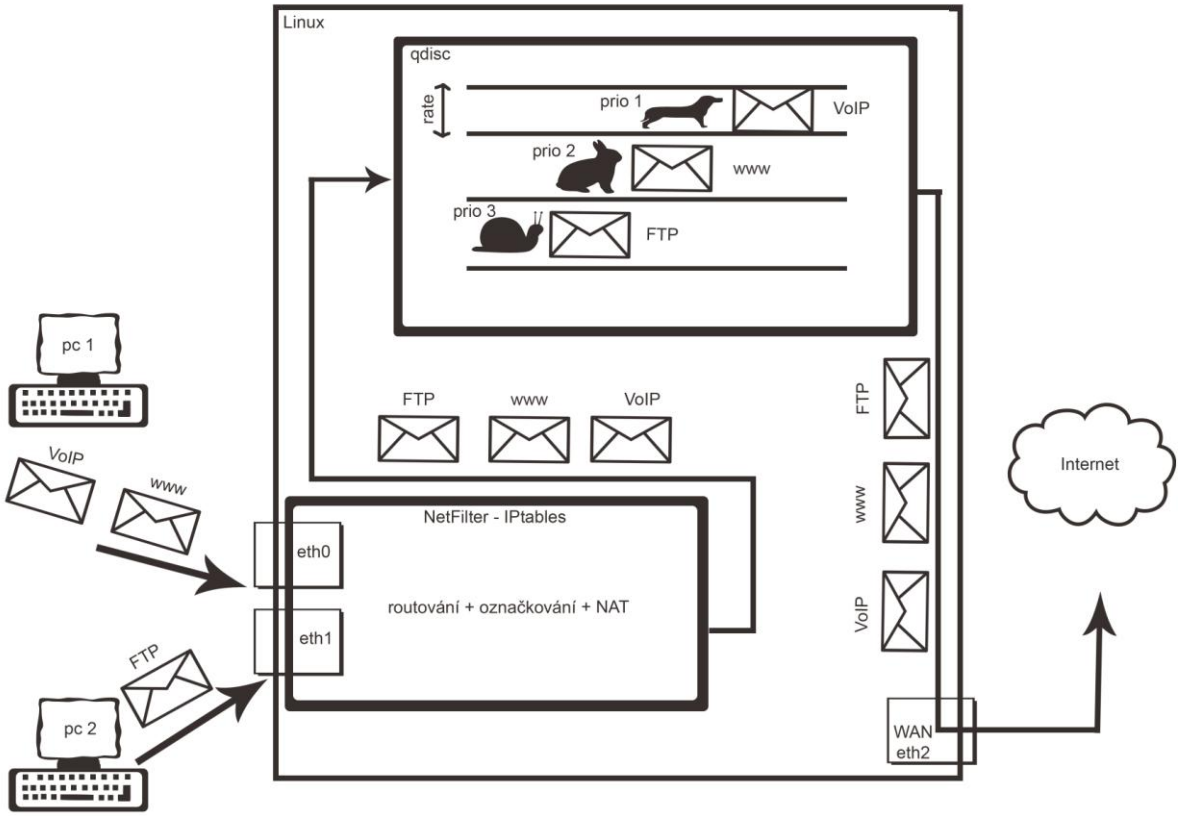
Obrázek 3

Přidělení datového toku je dobré zkombinovat s QoS (Quality of Service), kterým je možné upřednostňovat jednotlivé druhy spojení, náchylné na kvalitu odezvy vůči ostatním spojení. Pro každý stupeň se vytvoří samostatná třída. Bude použito tří stupňů QoS:

- prio 1 – nejvíce upřednostňované spojení typu Internetové telefonie a her
- prio 2 – všechny explicitně nenastavené spojení budou zařazeny do této skupiny
- prio 3 – nejméně upřednostňované spojení typu stahování souborů a www

Teoretický princip je názorně vidět na obrázku 4, kde byly vyslány z dvou počítačů tři pakety (FTP, HTTP, VoIP). Při vstupu na rozhraní prochází pakety *netfilterem*, kde je rozhodnuto, na které rozhraní budou přeposlány dále, označkovány podle druhu paketu a přeloženy IP adresy vnitřní sítě na IP adresu vnější sítě. Jakmile je systém (kernel) chce odeslat, předá je qdiscu, který pakety přeorganizuje na základě značek a předem daných pravidel a qdisc je odevzdá zpět systému pro odeslání na odchozí port. Nutno dodat, že

jednotlivé počítače či sítě by mohly mít každý svojí vlastní frontu nezávislou na ostatních frontách.



Obrázek 4

V případě příchozího datového toku, je situace složitější, protože *qdisc* se dá přiřadit pouze na jedno rozhraní a to pouze v odchozím směru. Toto dokáže vyřešit virtuální síťové rozhraní, které se předřadí před lokální porty. Podpora síťového virtuálního rozhraní je přímo v jádře s modulem *ifb*. Pro jeho zavedení se používá příkaz:

```
modprobe ifb numifbs=1
ip link set ifb0 up
```

Alternativu nabízí *IMQ*, které není pro jeho metody řešení obsaženo v jádře Linuxu. Na virtuálním rozhraní *ifb* je možné provádět již vše stejným principem jako v případě odchozího datového toku. Pro přesměrování se použijí pro každé rozhraní příkazy:

```
tc qdisc del dev eth0 root
tc qdisc add dev eth0 root handle 1:0 prio
tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32
     match u32 0 0 flowid 1:1 action mirred egress
     redirect dev ifb0
```

Nyní se vytvoří stromová struktura pro rozdělování odesílaných datových toků pomocí řadící fronty HTB jak je naznačeno na obrázku 3:

```
tc class add dev eth2 parent 1:1 classid 1:11 htb rate
     450kbit ceil 1024kbit
tc class add dev eth2 parent 1:1 classid 1:12 htb rate
     450kbit ceil 1024kbit

tc class add dev eth2 parent 1:11 classid 1:111 htb rate
     128kbit ceil 1024kbit prio 1
tc class add dev eth2 parent 1:11 classid 1:112 htb rate
     128kbit ceil 1024kbit prio 2
tc class add dev eth2 parent 1:11 classid 1:113 htb rate
     128kbit ceil 1024kbit prio 3

tc class add dev eth2 parent 1:12 classid 1:121 htb rate
     128kbit ceil 1024kbit prio 1
tc class add dev eth2 parent 1:12 classid 1:122 htb rate
     128kbit ceil 1024kbit prio 2
tc class add dev eth2 parent 1:12 classid 1:123 htb rate
     128kbit ceil 1024kbit prio 3
```

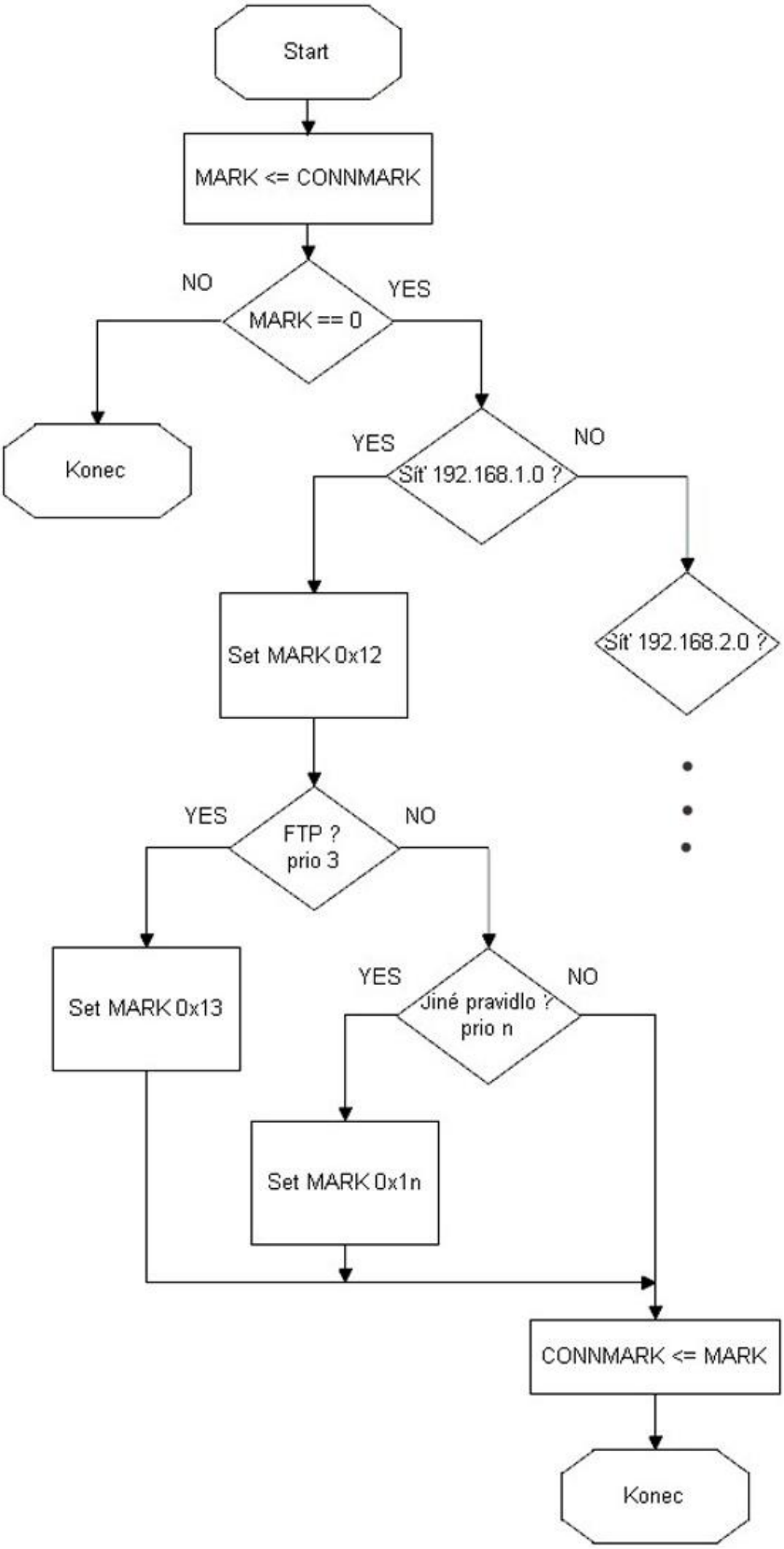
Na každou třídu je ještě zavěšena řadící fronta SFQ, která zaručí spravedlivé přerozdělení pro každé spojení v jednotlivých třídách:

```
tc qdisc add dev eth2 parent 1:111 handle 111:0 sfq
     perturb 10
tc qdisc add dev eth2 parent 1:112 handle 112:0 sfq
     perturb 10
tc qdisc add dev eth2 parent 1:113 handle 113:0 sfq
     perturb 10
tc qdisc add dev eth2 parent 1:121 handle 121:0 sfq
     perturb 10
tc qdisc add dev eth2 parent 1:122 handle 122:0 sfq
     perturb 10
tc qdisc add dev eth2 parent 1:123 handle 123:0 sfq
     perturb 10
```

Filtery se nasadí do tříd, aby bylo jasné, které označkované pakety do nich patří:

```
tc filter add dev eth2 parent 1:0 protocol ip prio 1
     handle 0x11 fw flowid 1:111
tc filter add dev eth2 parent 1:0 protocol ip prio 2
     handle 0x12 fw flowid 1:112
tc filter add dev eth2 parent 1:0 protocol ip prio 3
     handle 0x13 fw flowid 1:113
tc filter add dev eth2 parent 1:0 protocol ip prio 1
     handle 0x21 fw flowid 1:121
tc filter add dev eth2 parent 1:0 protocol ip prio 2
     handle 0x22 fw flowid 1:122
tc filter add dev eth2 parent 1:0 protocol ip prio 3
     handle 0x23 fw flowid 1:123
```

Analogicky se vše provede i pro příchozí spojení. Teď již stačí pakety pouze označkovat. Teoretický princip je naznačen na obrázku 5.



Obrázek 5

Pro snadné značkování celého spojení se zavedou moduly *ip_conntrack* a *ip_conntrack_ftp* pomocí těchto příkazů:

```
modprobe ip_conntrack
modprobe ip_conntrack_ftp
```

Pro zjednodušení budou uvedeny jenom dvě služby z jedné sítě. Protokol FTP, komunikující na portech 20 a 21, který bude zařazen do prio 3 a na spojení velmi citlivá počítačová hra Counter-Strike, komunikující na portu z rozsahu 27015-27019, která bude prio 1. Vše ostatní neoznačené bude prio 2. V proměnné CONNMARK je uložena značka spojení. Pokud je CONNMARK = 0 jedná se o nové spojení a je třeba ho označit. V opačném případě se neprovede nic.

```
iptables -t mangle -A FORWARD -j CONNMARK --restore-mark
iptables -t mangle -A FORWARD -m mark --mark 0 -j conn
```

V nově definovaném řetězci se zjistí, o jakou síť se jedná:

```
iptables -t mangle -N conn
iptables -t mangle -A conn -s 192.168.1.0/24 -o eth2 -g
     conn_sn1
iptables -t mangle -A conn -s 192.168.2.0/24 -o eth2 -g
     conn_sn2
iptables -t mangle -A conn -j RETURN
```

Pokud by to byla například síť 192.168.1.0/24 provede se podle pravidel označení paketů:

```
iptables -t mangle -N conn_sn1
iptables -t mangle -A conn_sn1 -j MARK --set-mark 0x12
iptables -t mangle -A conn_sn1 -p tcp -m multiport
     --dports 20:21 -j MARK --set-mark 0x13
iptables -t mangle -A conn_sn1 -p tcp -m multiport
     --dports 27015:27019 -j MARK --set-mark 0x11
iptables -t mangle -A conn_sn1 -j CONNMARK --save-mark
iptables -t mangle -A conn_sn1 -j RETURN
```

Veškerou konfiguraci je možné uložit do výše vytvářeného souboru *firewall.sh*, která se spouští po startu nebo pro lepší přehlednost vytvořit spouštěcí skript nový.

6.8 Sdílení souborů – Samba

Samba je open-source balík, pomocí kterého je možné sdílet prostředky mezi stanicemi s operačním systémem Windows prostřednictvím protokolu SMB. Tento router zároveň může sloužit jako NAS server pro zálohu či autentizované sdílení souborů. Nejlepší je zvolit právě tento protokol, kvůli vzájemné kompatibilitě a celkovému komplexnímu řešení.

Instalace Samby se provede příkazem:

```
apt-get install samba
```

Doinstalování grafického uživatelského prostředí, ze kterého se bude Samba server ovládat:

apt-get install gadmin-samba

Konfigurační soubor Samby je uložen v */etc/samba/smb.conf*. Konfigurační rozhraní se spouští:

Aplikace->Systémové nástroje-> Gadmin Samba.

Na první záložce *Server settings* se nastaví parametry:

- Server host name – jméno Samba serveru (SambaServer)
- Workgroup – název pracovní skupiny (Home)
- Security level – způsob autentizace (User – každý uživatel se přihlašuje jménem a heslem)
- Allowed hosts and network – povolení sítí ze kterých je možno se přihlásit (zadají se všechny lokální sítě ve tvaru 192.168.1.0/24 192.168.2.0/24)

V záložce *Users* se vytvářejí uživatelé a přiřazují se jim jména a hesla, pomocí nichž se budou přihlašovat z jiných počítačů. Při nastavování hesel je dbáno na zásady bezpečného hesla. To znamená osmimístné heslo s alfanumerickými a speciálními znaky jako ochranu proti uhádnutí či Brute force útoku. Na záložce *Shares* je nastavení oprávnění složek, které budou sdíleny:

- Shared name – název sdílené složky v Sambě
- Shared directory – cesta k sdílené složce. Pokud již vybíráme existující složku s daty, je důležité, aby měla složka nastavená přístupová práva pro uživatele nebo skupinu
- Access allowed – uživatelé kteří mají k složce přístup
- Write access – povolení zápisu
- Browsable: yes – sdílená složka bude zneviditelněná, ale pořád bude přístupná
- Guest ok: yes – povolí přístup bez hesla

Pro výpis právě připojených klientů k serveru Samba na routeru lze použít příkaz:

smbstatus

Takto nasdílené složky se mohou snadno připojit ve Windows jako místní disk. Je možné připojit a sdílet i tiskárnu.

Z hlediska bezpečnosti je Sambu vhodné umístit na virtuální systém v programu *VirtualBox*. Protože pokud by útočník dokázal využít chyby v Sambě a získal přístup k celému stroji, celá síť by byla ohrožena. V případě umístění ve virtuálním operačním systému je toto riziko minimalizováno.

6.9 Automatické aktualizace systému

Aktualizace je potřeba provádět pravidelně, ať už z hlediska bezpečnosti nebo pouze jako opravu nějakého problému. Pokud by se neaktualizovalo, mohl by útočník pomocí nějaké nově nalezené neopravené chyby získat například přístup do systému. U domácích routerů je toto nutné provádět ručně. Automatizovaný způsob pomocí služby je vhodnější z uživatelského komfortu. V grafickém prostředí se nastavuje v:

Systém -> Správa -> Správce aktualizací

Po zvolení záložky Aktualizace se zaškrtne kontrolovat aktualizace denně. Dále je nutno vybrat možnost - Instalovat bezpečnostní aktualizace bez potvrzení. Nyní každý den Správce aktualizací zkontroluje a případně zaktualizuje systém.

6.10 Zákaz nepotřebných služeb

Je důležité mít na paměti, že čím více služeb je spuštěno, tím více existuje bezpečnostních hrozeb, jak se píše v [2]. Proto je dobré všechny nevyužívané služby zakázat. V grafickém prostředí najdeme spouštěné služby:

Systém -> Správa -> Služby

Zakážou se tedy všechny nepotřebné služby jako:

- nfs-common
- portmap

6.11 Administrace

K administraci lze lokálně použít grafické uživatelské prostředí. Pro vzdálené připojení se použije program *Team Viewer* 6, který je pro nekomerční použití zdarma. Jedná se o vzdálenou sdílenou plochu s dalšími funkcemi. Zcela bezplatnou alternativu nabízí *NX Server*, ke kterému je pro přístup z internetu potřeba veřejná IP adresa. Komunikace *Team Viewer*u je zašifrovaná a komprimovaná. Je dostupný pro operační systémy Windows, Linux, Mac OS a dokonce i mobilní systémy Apple iOS a Android.

Instalace programu se provede příkazem:

```
apt-get install teamviewer6
```

Pomocí tohoto programu je také možné přehrávat data. Na počítač, který se připojuje k routeru, se pouze stáhne a spustí stejný program pro připojení *Team Viewer* a vyplní přihlašovací číslo a heslo, které je třeba předem nastavit na routeru.

6.12 Program ARPwatch

Výbornou detekci ARP Cache poisoningu představuje program *ARPwatch*, který si jednoduše trvale zapamatovává všechny záznamy z ARP Cache tabulky. Jakmile objeví nějakou známou MAC adresu u jiné IP adresy, okamžitě o tom informuje prostřednictvím emailu.

Program nainstalujeme příkazem:

```
apt-get install arpwatch
```

Nastavení, se kterým se program *ARPwatch* spouští se nastavuje v */etc/default/arpwatch.* Tento soubor se zedituje a nastaví tento argument do tvaru:

```
ARGS=" -f /var/lib/arpwatch/arpdatabaze -i eth2 -N -m
     email@neco.cz"
```

Kde parametr -i udává interface, -N označuje promiskuitní mód, -m znamená emailovou adresu, kam se odešle upozornění. Před prvním spuštěním je potřeba vytvořit prázdný soubor arpdatabaze. Jinak se program nespustí. Na výpis programu je možno se podívat tímto příkazem:

```
cat /var/log/syslog | grep arpwatch
```

Pro ověření zda je ARPwatch aktivní se nechá použít příkaz:

```
ps -fe | grep arpwatch
```

Zápis do databáze je prováděn na disk a to jednou za 15 minut, nebo při ukončení programu. ARPwatch se spustí tímto příkazem:

```
/etc/init.d/arpwatch start
```

6.13 Virtualizovaný operační systém

Pro provoz dalších služeb jako jsou webový, poštovní, datový nebo jiný server máme tři možnosti:

- Pořídit další počítač – jak je doporučováno v [2]
- Provozovat vše na jednom počítači – podle [2] se nedoporučuje, ale často používá
- Spustit další virtualizovaný systém

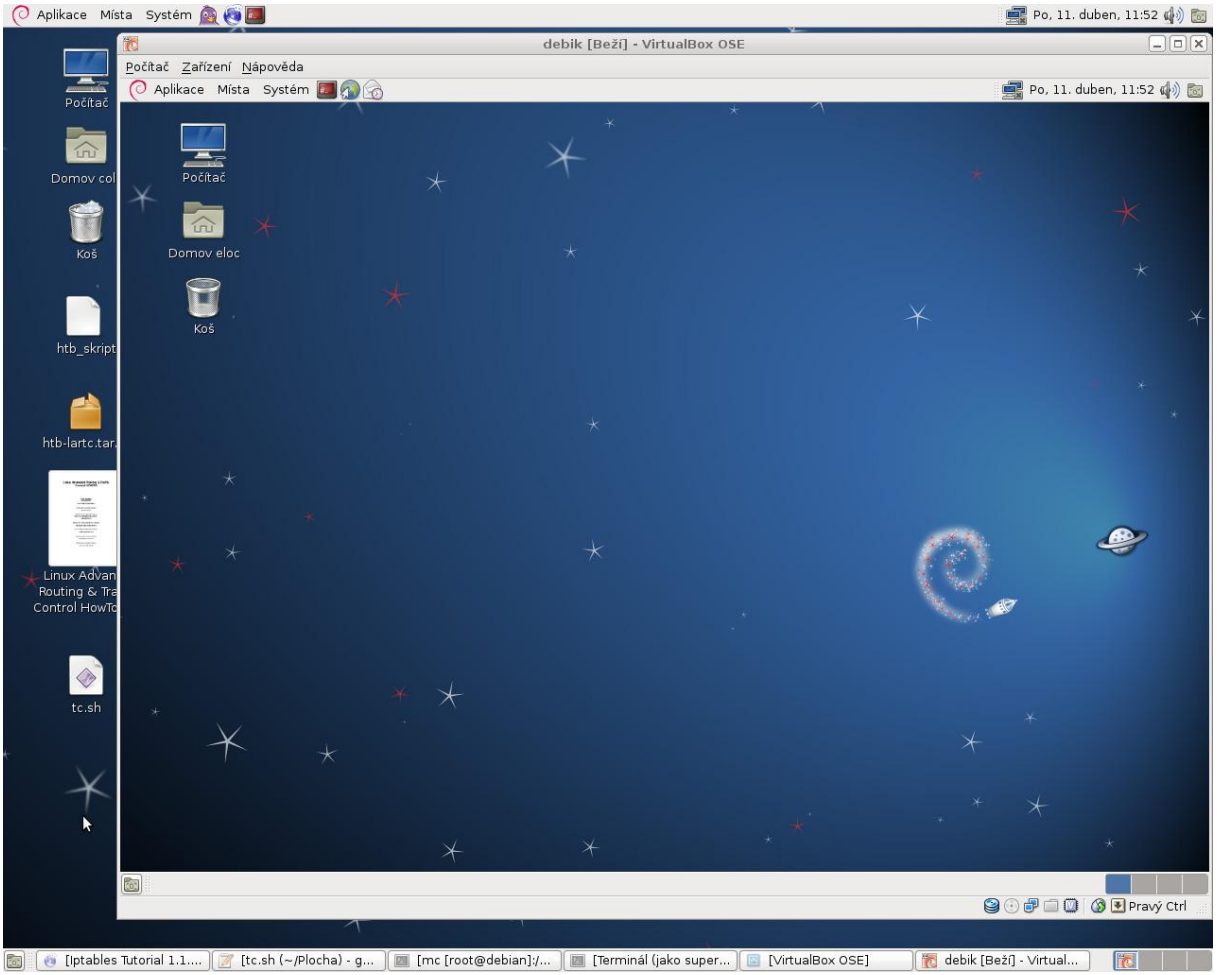
Bezpečnostní a cenový kompromis mezi prvními dvěma možnostmi je možnost třetí. Spuštění úplného virtualizovaného systému má nejednu výhodu:

- Dynamické přelévání výkonu – vše může běžet na jednom počítači a výkon jednoho silnějšího počítače se může dynamicky přelévat podle aktuální potřeby mezi jednotlivými systémy

- Velice snadná zálohovatelnost a přenositelnost – celý systém je pouze jeden soubor. Obnova či záloha trvá pouze o něco málo delší dobu, než je doba nutná k překopírování tohoto souboru
- Zvýšená bezpečnost – pokud by se povedlo útočníkovi napadnout a získat přístup k virtualizovanému systému, tak je velice obtížné se dostat do hostitelského operačního systému

Za dobrý nástroj pro úplnou virtualizaci je považován *VirtualBox*. Existují i další vizualizační nástroje jako *VMware*, ale ty zdaleka nejsou tak uživatelsky přívětivé. *VirtualBox* se nainstaluje příkazem:

```
apt-get install virtualbox-ose
```



Obrázek 6

Pokud se bude přidávat virtualizovaný operační systém, tak v nastavení virtuálního pevného disku je možnost zvolit - Dynamicky se zvětšující obraz. To zajistí, že pokud se nastaví při instalaci třeba 2TB pevný disk, tak v hostitelském operačním systému bude soubor s virtualizovaným systémem zabírat pouze tolik místa, kolik je tam skutečně nahráno dat. To umožní například jednodušší přechod při upgradu pevného disku.

Pro nastavení sítě se ve VirtualBoxu vybere:

Soubor -> Předvolby -> Síť

A zde se dvakrát klikne na vboxnet0, což je virtuální rozhraní nastavovaného routeru, ke kterému bude připojen virtuální operační systém. V této kartě se manuálně nastaví IP adresa a maska sítě. V záložce DHCP server se pak provede jeho deaktivace odškrtnutím - Povolit server.

Nyní po nainstalování operačního systému se klikne pravým tlačítkem myši na jeho ikonu a vybere:

Nastavení -> Síť -> Karta 1 -> Připojena k: Síť pouze s hostem

Aby mohly virtualizovaný systém obsluhovat všechna jádra používaného procesoru, tak se vybere na ikoně instalovaného systému:

Nastavení -> Systém -> Procesor

A v záložce *Procesor* se nastaví počet jader v našem počítači. Praktická funkčnost je vidět na obrázku 6.

7 Testování

7.1 Testovací síť

Testovací síť, do které byl router nasazen a zkoušen, se skládala ze dvou počítačů napojených přímo na navrhovaný router na rozhraní eth0 a eth1. Každý počítač byl v jiné síti 192.168.1.0/24 a 192.168.2.0/24. Pro nasimulování připojení k Internetu, byla tato síť z rozhraní eth2 192.168.0.0/24 lokálně připojena k domácímu routeru Asus WL-520gU, který byl připojen do internetu. Do domácího routeru byl napojen další třetí počítač, simulující počítač na internetu pro nahrávání souborů. Na všech počítačích byly zprovozněny služby potřebné pro testování, jako jsou *FTP* servery a *Samba* servery.

7.2 Testování rychlostí

Hodnoty nastavené v programu *tc*, pro pokročilou kontrolu datového toku jsou vždy přibližně o 10% větší než naměřené v testu na portálu rychlost.cz. Vysvětlením je zřejmě to, že se v programu *tc* započítají veškerá data, i s hlavičkami jednotlivých protokolů zatímco v druhém případě jenom holá data. Zároveň je obsažena nějaká chyba měření, která je zanedbatelná s ohledem na celkové zaměření testů, přesto k její minimalizaci bylo každé měření provedeno pětkrát a aritmeticky zprůměrováno.

K testování sloužila softwarově omezená linka 10240kbit pro stahování a 1024kbit pro nahrávání směrem do Internetu nastavená v programu *tc.*

Dynamické přelévání volného datového toku

Spuštěn je pouze samotný test. První počítač má celou linku pro sebe na provedení testů. Minimálně má zaručeno 256kbit pro nahrávání a 2560kbit pro stahování, využívá ale celé pásmo

Down [kbit]	Up [kbit]	Min ping [ms]	Avg ping [ms]	Max ping [ms]
8697	910	34	45	82
9403	919	34	46	77
9531	917	22	46	65
8012	917	22	34	52
8507	918	18	39	59
8830	**916**	**26**	**42**	**67**

Z testu je vidět, že nevyužitý datový tok je opravdu přesně půjčován.

Bez omezení

Oba počítače plně využívaly každý své přidělené minimální pásmo. Nebyly nastaveny žádné priorizace. Druhý počítač nahrával pomocí *FTP* jeden soubor na třetí počítač, první měřil rychlost jeho spojení.

Down [kbit]	Up [kbit]	Min ping [ms]	Avg ping [ms]	Max ping [ms]
2356	450	95	133	239
2251	460	92	116	202
1782	434	86	123	219
1843	438	105	124	204
3040	466	89	124	240
2254	**450**	**93**	**124**	**221**

Z této tabulky je vidět téměř čtyřnásobné zvýšení webové odezvy. Čtyřnásobný pokles je také u rychlosti stahování, byť druhý počítač stahoval pouze kontrolní pakety TCP spojení. Jakékoliv aktivity závislé na stavu připojení jsou nyní téměř nepoužitelné.

Minimální datový tok

Pro první měřící počítač se nastavil minimální zaručený datový tok nahrávání 256kbit a maximální 1024kbit. Druhý počítač měl nastavení 768/1024kbit. Oba počítače plně využívaly každý své přidělené minimální pásmo. Nebyly nastaveny žádné priorizace. Druhý počítač nahrával pomocí FTP soubor na třetí počítač, první měřil rychlost jeho spojení. Bylo zjišťováno, zda bylo rozdělené spravedlivě a správně podle nastavení.

Down [kbit]	Up [kbit]	Min ping [ms]	Avg ping [ms]	Max ping [ms]
8503	234	20	41	70
8727	236	21	31	45
8470	236	33	43	60
8259	234	22	38	58
9131	236	32	37	58
8618	**235**	**26**	**38**	**58**

Tento test ukázal, že rozdělení funguje výborně.

Priorizace

Oba počítače mají nastaveno minimálně 256kbit a maximálně 1024kbit s tím, že první měřící počítač má prioritu 1 a druhý prioritu 2.

Down [kbit]	Up [kbit]	Min ping [ms]	Avg ping [ms]	Max ping [ms]
8400	692	33	41	73
8073	694	32	39	59
9356	686	28	38	59
9677	685	32	39	57
9179	694	34	38	57
8937	**690**	**32**	**39**	**61**

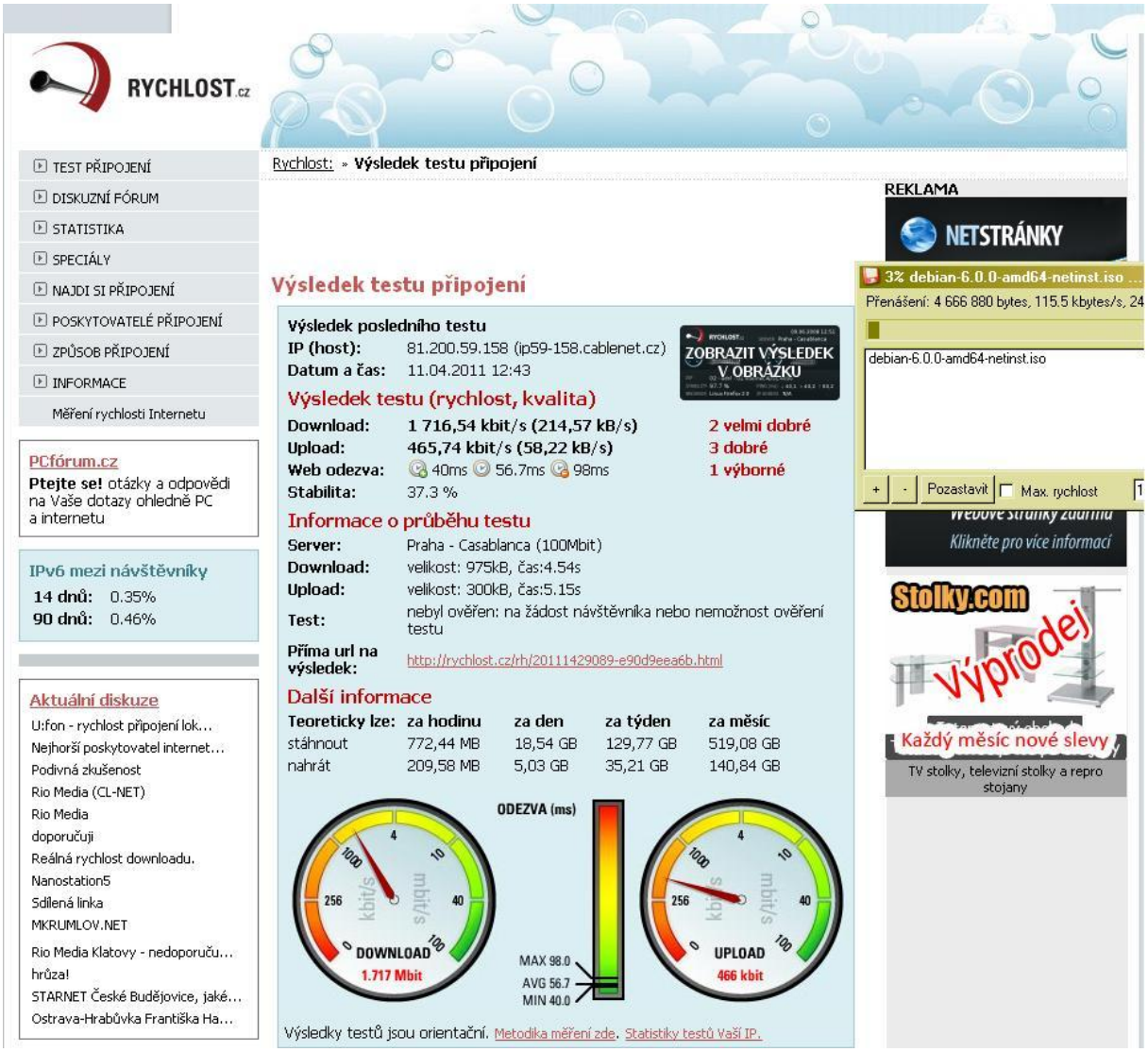
Druhému počítači bylo zaručeno jeho minimální pásmo, ale veškeré volné nepřidělené prostředky byly ponechány priorizovanému počítači.

Spravedlivé rozdělení

Test je stejný jako test výše provedený test „Bez omezení“. Je v něm ale pomocí SFQ ošetřeno spravedlivé rozdělení ve stejné prioritní třídě pro každé spojení.

Down [kbit]	Up [kbit]	Min ping [ms]	Avg ping [ms]	Max ping [ms]
4411	466	38	56	89
4891	463	44	60	93
4562	463	36	60	108
4469	470	41	58	108
4279	465	45	56	67
4522	**465**	**41**	**58**	**93**

Tímto byly nasimulovány dvě služby stejné priority, z níž jedna je citlivá na připojení a druhá má naopak tendence zabrat si celé spojení pro sebe. Výsledné odezvy jsou asi dvakrát horší než při priorizování webového spojení. Ale i přesto se hodí řadič paketů na jednotlivé úrovně priorizace zařadit. Pomůže to v priorizovaných úrovních dodržet určitou spravedlnost rozdělení pro spojení v jedné kategorii.



Obrázek 7

7.3 Ostatní testy

Kromě výše uvedených rychlostních testů byla otestována také celková funkčnost a dostupnost celé sítě jak je popsáno v kapitole pět.

8 Další možnosti využití

Pro již zmíněnou vysokou konfigurovatelnost, je možné používat jakékoliv služby. Ovšem nemusí vždy být využito všech navrhovaných. Níže jsou uvedené scénáře možného nasazení routeru, které autor vysloveně doporučuje s konkrétními funkcemi.

8.1 Rodinný dům a byt

Navrhovaný router by mohl najít využití například v rodinném domě nebo bytě s několika uživateli Internetu jako lepší náhrada za domácí WiFi router. Centrální sdílení souborů se v domácnosti mezi více počítači hodí. Jedná se o důvěryhodné prostředí, proto by byly restrikce poměrně malé. Všechny počítače by mohly být na jedné podsíti třeba z důvodu hraní počítačových her, které je dobré nastavit jako priorizované s určitým zaručeným minimálním datovým tokem. S tímto nastavením již nebude problém zároveň stahovat soubory a používat na spojení citlivé služby jako hraní her či internetová telefonie.

8.2 Bytový dům

Další nasazení by bylo možné v bytovém domě s dvěma až šesti byty. Sousedé by se domluvili na společném pořízení internetového připojení a nasazení osobního počítače s linuxovou distribucí. Všichni by měli garantovanou svojí poměrnou část datového toku, ale navíc s dynamicky se přelévajícím volným datovým tokem. Tímto by mohli společně dosáhnout nižších nákladů a zároveň mít rychlejší služby. Na každý síťový port routeru by byl připojen switch, do kterého by se připojily všechny počítače z celého bytu. Tím by mohly být na společné síti, ovšem odděleny od ostatních bytů. Tento router by jim mohl zároveň sloužit k zálohování či autorizovanému ukládání dat a tím zastávat funkci NAS serveru. Zároveň by mohli spustit na tomto stroji například osobní webové stránky a šetřit tak za na Internetu jinak zpoplatněnou službu. Takový router, by zůstával neustále zapnutý, což se přímo nabízí k spuštění většího datového stahování, které by mohlo nerušeně probíhat i přes noc.

8.3 Malá firma

Jak se píše v [2] hodně malých firem, zřejmě v rámci úspor provozuje firewall s poštovními a jinými servery na jednom počítači. Bohužel toto řešení není vhodné z hlediska bezpečnosti. Zde by se například velmi uplatnil virtualizovaný operační systém, který by situaci vylepšil a zároveň zachoval cenovou příjemnost řešení s možností poskytnutí dalších nadstandardní funkcí.

9 Závěr

V této práci se povedlo navrhnout zajímavé zařízení kombinující velkou škálu funkcí. V porovnání s ostatními porovnávanými je osobní počítač s linuxouvou distribucí zařaditelný funkčně nad upravené domácí WiFi routery běžícími na ořezaném Linuxu. Toto řešení je zároveň dostatečně odůvodnitelné přes jisté nevýhody v podobě:

- větších rozměrů
- spotřeby
- hlučnosti
- pořizovacích nákladů
- složitějšího nastavení

Tyto nevýhody dostatečně vyvažují:

+ větší výpočetní výkon
+ výrazně rychlejší a větší datové úložiště
+ podpora protokolu IPv6
+ vlastní služby (www, ftp, smtp a další servery)
+ vysoká univerzálnost
+ sofistikovanější kontrola a správa síťového provozu
+ podpora autorizovaného přístupu k datům
+ šifrované uložení a přenos dat o vysokých rychlostech
+ možnost automatické aktualizace
+ oddělení portů

Tento multifunkční router může být použit v širokém spektru různých scénářů použití, ale možnost jeho nasazení končí u středních firem, protože ty nejsou omezeny finančními prostředky a zvolí specializované profesionální řešení. Operační systém Linux je zajímavý a umožňuje velkou spoustu věcí, problémem je ovšem občas neexistující či neúplná dokumentace.

10 Bibliografie

[1] KABELOVÁ, Alena; DOSTÁLEK, Libor. Velký průvodce protokoly TCP/IP a systémem DNS. 5 . Brno: Computer Press, 2008. 542 s. ISBN 978-80-2512236-5, EAN:9788025122365

[2] KRČMÁŘ, Petr. *Linux : tipy a triky pro bezpečnost*. Vyd. 1. Praha : Grada Publishing, 2004. 208 s. ISBN 80-247-0812-4.

[3] Lupa [online]. 2006 [cit. 2010-04-09]. Odposloucháváme data na přepínaném Ethernetu. Dostupné z WWW:< http://www.lupa.cz/clanky/odposlouchavame-data-na-prepinanem-ethernetu-1/>. ISSN 1213-0702

[4] Lupa [online]. 2006 [cit. 2010-04-09]. Bráníme se odposlechu ARP Cache Poisoning a připojení počítače k síti. Dostupné z WWW: < http://www.lupa.cz/clanky/arp-cache-poisoning-a-pripojeni-pocitace-k-siti/>. ISSN 1213-0702

[5] RAY, Deborah S.; RAY, Eric J. Unix – podrobný průvodce. Praha: Grada, 2009. 416 s. ISBN 9788024721255.

[6] ŠTRAUCH, Adam. *Root* [online]. 2010 [cit. 2011-04-10]. IPFire: domácí router za 15 minut. Dostupné z WWW: <http://www.root.cz/clanky/ipfire-domaci-router-za-15-minut/>. ISSN 1213-0702

[7] DEVERA, Martin. *Luxik.cdi* [online]. 2002 [cit. 2011-04-10]. HTB Linux - user guide. Dostupné z WWW: <http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm/>. ISSN 1213-0702

[8] Linux Foundation. *The Linux Foundation* [online]. 2009 [cit. 2011-04-10]. Ifb. Dostupné z WWW: <http://www.linuxfoundation.org/collaborate/workgroups/networking/ifb/>. ISSN 1213-0702

[9] DAVAINE, Max. *Abclinuxu* [online]. 2006 [cit. 2011-04-10]. Traffic shaping - 2 (IMQ a úvod do shapingu). Dostupné z WWW: < http://www.abclinuxu.cz/clanky/site/traffic-shaping-2-imq-a-uvod-do-shapingu/>. ISSN 1213-0702

[10] DOČEKAL, Michal. *Linux Expres* [online]. 2010 [cit. 2011-04-13]. Správa linuxového serveru: Linuxový firewall, základy iptables. Dostupné z WWW: <http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-linuxovy-firewall-zaklady-iptables>. ISSN 1213-0702