

Budování sítě s připojením do internetu v odlehlých lokalitách

Building a network with
connection to the Internet in the
areas that are not easy to reach

Jiří Jaroš

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce panu Ing. Tomáši Dulíkovi, za odborné vedení, rady a připomínky, které mi poskytoval při řešení práce.

Prohlašuji, že jsem na celé diplomové práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 08. 06. 2005

.....

Jiří Jaroš

Resumé

V dynamickém rozvoji informačních technologií se hlavním předpokladem rozvoje vysokorychlostní internet. Tato podmínka platí jak pro firemní a státní sektor, tak i pro soukromé osoby. Ačkoli má každá obec svá specifika, může má práce sloužit jako univerzální návod pro připojení menších vesnic v regionu. Správu a náklady na infrastrukturu sítě si následně řeší obec již sama.

Resume

Fast speed Internet is the most important for the development of information technology in today's fast growing world.

This applies to both private and public sector, but also to individuals.

Despite the fact that each region has its own specifications, my work can act as universal manual for the connection of smaller villages in the region. Revenue cost, such as maintenance of infrastructure network, is dealt and funded by each region itself.

ÚVOD.....	7
I. TEORETICKÁ ČÁST	7
1. OPERAČNÍ SYSTÉM	10
1.1. STAROS	10
1.2. MIKROTIK ROUTER.....	11
1.3. DEBIAN LINUX	11
2. SLUŽBY ROUTERU	13
2.1. FIREWALL	13
2.1.1. Funkce firewallu.....	14
Antivirová ochrana	14
Optimalizace připojení	14
Problémy s IP	14
Přístupová práva	14
Sdílení přístupu k internetu	15
2.1.2. Shorewall.....	15
Instalace.....	15
Konfigurace	15
2.2. DHCP SERVER	16
2.2.1. Mechanismus DHCP	17
2.2.2. Server DHCP.....	17
2.2.3. Konfigurace serveru DHCP	18
2.2.4. Deklarace.....	19
2.3. SPRÁVEDLIVÉ PŘIDĚLOVÁNÍ KAPACITY LINKY QOS – POMOCÍ ALGORITMU FAIR USER POLICY – HTB	21
2.4. SLEDOVÁNÍ SÍTĚ.....	22
2.4.1. IPTraf	23
2.4.2. Instalace.....	23
2.4.3. Sledování sítě	23
2.4.4. Monitor IP provozu	24
2.4.5. Obecné statistiky o rozhraních.....	25
2.4.6. Detailní statistika rozhraní	25
2.4.7. Statistické rozborý.....	25
2.4.8. Monitor LAN stanice	25
2.4.9. Filtry	26
3. TECHNOLOGIE PŘIPOJENÍ.....	27
3.1. CDMA.....	27
3.2. EDGE-GRPS	28
3.3. WIFI	28
4. KONFIGURACE SERVERU	29
4.1. TRANSPARENTNÍ HTTP PROXY SQUID	29
4.2. FTP SERVER.....	30
4.2.1. Instalace ProFTPD serveru	31

4.2.2.	Konfigurace ProFTPD serveru.....	32
4.2.3.	Řízení přístupu k FTP	33
4.2.4.	Jak přidat FTP do xinetd	34
4.3.	SMTP SERVER.....	34
4.3.1.	Mechanismus SMTP	35
4.3.2.	Základy SMTP	35
4.3.3.	Poštovní služba.....	37
4.3.4.	Instalace serveru Postfix	37
4.3.5.	Konfigurace serveru Postfix.....	38
4.3.6.	Provozování serveru Postfix	39
4.4.	JABBER SERVER	40
4.4.1.	Princip jabberu	40
4.4.2.	Kompilace serveru	41
4.4.3.	Konfigurace serveru	42
4.4.4.	Nastavení serveru	42
4.4.5.	Spuštění serveru	43
4.5.	VPN SERVER (POPTOP)	43
4.5.1.	Připojení k poté serveru – navazování spojení.....	43
4.5.2.	Změny na klientské stanici po navázání spojení s pptp serverem.....	44
4.5.3.	Omezení	44
4.5.4.	Ukončení spojení s pptp serverem	44
4.5.5.	Vytvoření VPN pomocí pptp v Linuxu	44
4.5.6.	Ppp.....	45
4.5.7.	Šifrování - protokol MPPE.....	45
4.5.8.	Pptp-linux.....	45
4.5.9.	Směrování	45
4.5.10.	Spuštění tunelu	45
5.	HARDWARE SÍŤE.....	47
5.1.	PŘÍSTUPOVÝ BOD	47
5.2.	POČÍTAČ.....	48
	Ukázkový konfigurační soubor pptpd.conf	57
II.	PRAKTICKÁ ČÁST	49
1	REALIZACE BEZDRÁTOVÉ SÍŤE PROVODOV	50
	SEZNAM POUŽITÉ LITERATURY	52
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	53
	SEZNAM OBRÁZKŮ	55
	SEZNAM TABULEK.....	CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.
	SEZNAM PŘÍLOH.....	56

ÚVOD

Vysokorychlostní internet zažívá v současné době nebývalý vzestup. Každý den z médií na nás útočí reklama na vysokorychlostní připojení, kvalitu poskytovaných služeb a nízkou cenu.

Pokud se malá obec rozhodne pro připojení k vysokorychlostnímu internetu, stojí před několika problémy. Zaprvé jsou to zřizovací náklady, které se musí do projektu investovat. Tím rozumíme funkční počítač s větším hardiskem, bezdrátovou kartu, všesměrovou anténu, kabeláž a bleskojistku. Vhodným doplňkem je i záložní zdroj UPC chránící síť před výpadky elektrické energie.

Zadruhé je to zvolená technologie. V současné době můžeme na trhu registrovat širokou nabídku různých způsobů připojení. Mobilní operátoři nabízejí technologii GRPS EDGE nebo CDMA, telefonní společnosti propagují ADSL a lokální provideři přichází s bezdrátovým připojením. Po výběru jedné z nabízených možností, stále ještě nemáme vyhráno. Zvolená technologie nemusí být v dané lokalitě dostatečně dimenzovaná a stabilní jako v případě GRPS EDGE nebo není v dané obci zřízená vůbec, což může být případ ADSL. Pak už nám zbývá pouze WI-FI připojení, které musí taktéž splňovat určitá kritéria. Dříve byla bezdrátová síť primárně určená pro připojení v historických budovách, protože zde odpadala složitá montáž kabelových systému. Díky této přednosti získala oblibu mezi mnohými uživateli sítě Internet, kteří nemají jinou možnost připojení. Nejdůležitější vlastností je přímá viditelnost k přístupovému bodu lokálního providera a nízké náklady na zřízení bezdrátové sítě. Když budeme provozovat svoji síť v bezlicenčním pásmu 2,4GHz, nemusíme platit žádné poplatky.

Další problém může představovat volba operačního systému. Díky dynamickému rozvoji bezdrátových sítí se na trhu objevily první operační systémy určené výhradně pro WIFI síť. Jak StarOS tak Mikrotik jsou stabilní výkonné systémy založené na jádru Linuxu. Ideální řešení však představuje prověřený operační systém Debian CZ, který je nejpoužívanější český systém v bezdrátových sítích. Lidé, sdružení v komunitních sítích, mají možnost využít spoustu služeb Debianu. Zmínil bych FTP server pro sdílení dat, SMTP pro odesílání pošty, Jabber server umožní komunikaci v síti a nesmíme zapomenout na základní bezpečnost, kterou představuje firewall.

Zbývá vyřešit administrativní záležitosti. Každá komunitní síť by měla mít právní status a ideálním řešením je založení občanského sdružení. Tato možnost skýtá hned několik

výhod. Rozhodovací postupy jsou pevně zavedeny ve stanovách sdružení, odpovědnost se dělí mezi členy, u eBanky a Poštovní spořitelny si můžeme vést svůj účet zdarma a zcela transparentně, členské příspěvky se nezdaňují a nedanitelný zisk může činit až 300 tisíc korun.

Pokud se nám podaří splnit výše popsané problémy, nebrání provozu sítě téměř nic. Zbývá si domluvit pravidelné členské příspěvky uživatelů sítě a zdatný správce sítě.

I. TEORETICKÁ ČÁST

1. OPERAČNÍ SYSTÉM

Pro realizaci bezdrátového routeru přístupového bodu jsem měl na výběr ze tří operačních systému. StarOS, Mikrotom routek a Debian Linux.

1.1. StarOS

StarOS je stabilní výkonný systém pro bezdrátové sítě založený na jádru Linuxu, snadno ovladatelný díky grafickému rozhraní. Pracuje na obyčejném PC s pevným diskem, nebo na speciálních PC kartách s malou Flash pamětí. Obsahuje všechny funkce potřebné pro výstavbu či provoz bezdrátové sítě. Základními vlastnostmi StarOS jsou ovladače karet Zcommax (čipset Prism), umožňující provoz karty jak v režimu klient, tak i v režimu AccessPoint . Dále ovladače všech karet s čipsetem Atheros, také v režimu klient i AccessPoint. StarOS podporuje celé pásmo ISM (2,3 až 2,6 GHz a 4,9 až 6,1 GHz s čipsety Atheros). Dalšími užitečnými vlastnostmi jsou možnost přihlašování klientů do sítě a ověřování jejich oprávněnosti pomocí hesla nebo MAC adresy, automatické přesměrování neregistrovaných klientů na zvolené stránky, omezování rychlosti na základě IP adresy, počítání přenesených dat, sledování a statistiky provozu v síti, přesné měření signálu, DNS, DHCP, proxy, firewall, NAT a mnoho dalších funkcí.

Aktivace systému probíhá tak, že zašlete MAC adresu jedné ze síťových karet (LAN nebo WLAN), které jsou v počítači se StarOS. Obratem vám obchodní společnost zašle SMS nebo e-mail s aktivačním klíčem.

Aktualizace firmware probíhá na základě stažení bin souborů ze stránek a následné instalace v programu.

1.2. Mikrotik router



Obr. 1. Mikrotik router

Extrémně rychlý operační systém založený původně na OS Linux. Má vše, co můžeme po moderním operační systému pro WI-FI požadovat. Obsahuje firewall, router, VPN rozhraní, NAT, telefonní bránu - nebo kombinace těchto možností. Je schopný komunikovat s cisco routery, omezovat linky, tvořit časovou základnu, DNS cache, WEB cache (PROXY) a spoustu dalších věcí. Jeho instalace je jednoduchá a jednoduchá je i jeho správa přes grafické rozhraní do windows, má integrovanou komunikaci zabezpečeným kanálem. Sám hlídá okolní linky a vyvolává akci na základě nedostupnosti, posílá E-mail, překonfigurovává se. Mezi jeho další možnosti řadíme DHCP server, Wireless Access Point, IP Telephony (H323), IPSec, Routing: BGP, OSPF, RIP, Policy Routing, Bandwith Management (omezení přenosové rychlosti pro jednotlivé uživatele), SSH a další. Poslední verze Mikrotiku obsahuje nové charakteristiky jako například rozšířenou wireless výkonnost, výkonnější QoS ovládání, P2P filtrování.

MikroTik ve verzi 2.8 Router je software nainstalovaný na IDE Flash Module a jeho zasunutím do základní desky získáváme výkonný router. Cena je 85 dolarů.

1.3. Debian Linux

Filosofie Debianu spolu s nástroji GNU, jádrem Linux a dalším svobodným softwarem vyústila do unikátní softwarové distribuce, která je známá pod názvem Debian GNU/Linux. Debian sám o sobě je naprosto volný produkt, je pouhou prací dobrovolníků a je to pravděpodobně nejvíce portovaná distribuce na světě. Podporuje dnes přes deset platforem a nainstalujeme jej při troše práce prakticky na jakýkoliv počítač, který obsahuje architekturu podporovanou linuxovým jádrem. Nenajdete v něm žádné balíčky, které nejsou zcela zdarma a pod licencí GPL.

Uživatelé jej oceňují pro jeho stabilitu a bezpečnost. V současné době obsahuje více než 8000 balíčků, které spravuje velmi propracovaný a sofistikovaný balíčkovací systém. Balíčky musí splňovat velmi přísné licenční podmínky, aby se mohly stát součástí distribuce. Všechny balíčky jsou pak k dispozici na ftp a http motorech po celém světě. Tvůrci a programátoři tvrdí, že Debian má široké využití a může se používat nejen jako desktop či libovolný server, ale i jako vědecká stanice pro distribuované výpočty.

Debian je znám třemi vývojovými větvemi. Stable se vydává jednou za pár let a jak již název napovídá, jedná se o stabilní verzi distribuce s otestovaným softwarem, nikoliv však s nejnovějším. **Testing** je budoucí **Stable** verze, do které se přidávají programy z **Unstable**. Unstable je verzí, kde se mění nejnovější ostré verze programu. Přesto je i tato verze poměrně kvalitní a stabilní.

Velkou devízou Debianu je unikátní balíčkovací systém. Výhodou při nainstalování určitého balíčku je hlášení systému o závislosti mezi balíčkama, kdy uživatel sám vyhodnotí, zda je závislé balíčky nutno také nainstalovat. Pro méně zkušené uživatele nabízí Debian grafickou nádstavbu pro instalaci balíčků program Dselect.

Podle nejnovějších průzkumů využívá distribuci Debianu 30% uživatelů Linuxu.

2. SLUŽBY ROUTERU

2.1. Firewall

Na úvod si objasníme, jak funguje firewall. Firewall je soubor pravidel, jež chrání vnitřní síť LAN před útokem "zvenku" - v praxi především z Internetu. Může být pouze na jednom PC a nebo na síťovém serveru, přes který jsou všechna ostatní PC napojena. Chrání tak počítače před útokem hackerů, před viry a červy. Firewall tedy obsahuje pravidla, jež řídí komunikaci z vnitřní sítě směrem ven, komunikaci soustřeďuje do jednoho uzlu, odfiltrává nebezpečné služby, blokuje nepřátelské monitorování sítě apod.

Veškerá komunikace síťových služeb probíhá skrze porty. A právě přes tyto porty lze vést útok proti vašemu PC nebo síti. Stačí neskenovat porty a služby, které na nich běží a pokud obsahují bezpečnostní díru, lze systém infiltrovat. Postupů jak proniknout do vaší sítě je samozřejmě více.

Čísla některých portů, které se standardně využívají :

FTP (přenos souborů): 20, 21

HTTP (www stránky): 80

POP (elektronická pošta): 110

IMAP3 (elektronická pošta): 220

hra Doom: 666

ICQ: 4000

Firewall má tedy za úkol kontrolovat všechny porty a komunikaci na nich probíhající. Tu vyhodnocuje podle pravidel, jež má nastaveny a rozhoduje, zda je komunikace vyhovující či nevyhovující. Pokud narazí na nějaké pokusy o proniknutí, pokouší se je odrazit a nepustit je dovnitř počítače.

2.1.1.Funkce firewallu

Antivirová ochrana

Každý kvalitní firewall má v sobě integrovanou ochranu před příchozími i odchozími počítačovými viry, a to především v elektronické poště.

Optimalizace připojení

Optimalizací rozumíme především využití cache paměti. Ty mají za úkol ukládat často navštěvované stránky, obrázky a prezentace. Při jejich dalším otevření se nenačítají z veřejné sítě, ale pouze z lokálního serveru. Tímto způsobem vystačíme i s pomalejším připojením k internetu.

Problémy s IP

V dnešní době dochází k velkému problému s přidělováním IP adres. Každý síťový uzel připojený k internetu musí totiž mít svou IP adresu, která navíc musí být jedinečná. Toto nám ale může příjemně ošetřit firewall. Za firewallem můžou být IP adresy jakékoliv, nezávislé na providerovi a když má dané požadavky vyslat na providera, tak se pomocí IP maskarády přeloží na jednu adresu, která je providerem přidělena.

Přístupová práva

Provozovatel sítě může omezit pomocí firewallu některé síťové služby jako třeba ICQ, prohlížení lechtivých stránek nebo stahování přes KAZZu. Tyto služby lze i monitorovat a zapisovat do protokolů.

Zabezpečená komunikace

Pomocí dvojice firewallu lze provádět komunikaci skrze internet na úrovni zabezpečeného tunelu. Díky těmto tunelům lze provozovat i VPN síť.

Sdílení přístupu k internetu

Firewall může dokonce plnit funkci směrovače. Například v malé firmě je jen jedna přípojka do internetu a je potřeba, aby byla konektivita dostupná všem připojeným PC.

2.1.2. Shorewall

Shorewall není žádný server nebo démon, je to sada skriptů a konfiguračních souborů. Aktivací daných pravidel se spustí a o vše se stará jádro Linuxu. Shorewall je určen pro jádra verze 2.4 a 2.6 a nedá se použít pro starší jádra, protože neobsahují netfilter/iptables.

Instalace

Stáhneme si poslední verzi Shorewallu pro Debian a příkazem `apt-get install shorewall` nainstalujeme firewall. Změníme aktuální adresář na `/etc/shorewall`.

Konfigurace

Netfilter rozlišuje tři zóny – Internet (net), lokální síť (loc) a demilitarizovanou síť (dmz). Všechny tři zóny najdeme nadefinované v souboru `zones` v adresáři `/etc/shorewall`. Konfigurace začíná nastavení jednoduchého konfiguračního souboru `policy`, v němž je implicitně nastaveno chování z vnitřní sítě do internetu – vše povoleno, opačným směrem – vše zakázáno a všechno ostatní zamítnuto. Do souboru je důležité zanést respektive „okomentovat“ pravidlo `fw net ACCEPT`, které říká, že z firewallu je přístup na internet povolen. Výše popsaným způsobem jsme nadefinovali politiku, to však nestačí.

V dalším kroku musíme povolit porty, které budeme v budoucnu využívat např. příchozí poštu, www server nebo ssh (porty 25, 80, 22). Toho dosáhneme nadefinováním pravidel. Jako první nakonfigurujeme stanici s jednou síťovou kartou `eth0` vložení řádku do souboru `interfaces` `net eth0 detect dhcp`. Ta se zapisují do souboru `rules`. Jako první můžeme povolit výjimku pro webový server `ACCEPT net fw tcp 80`. Tento řádek nám říká, že akceptuje pakety z Internetu směřující na firewall protokolem TCP s cílovým portem 80. Analogicky můžeme pokračovat třeba `FTP ACCEPT net fw tcp 20`, `ACCEPT`

net fw tcp 21, řádek ACCEPT net fw tcp 110 nám povolí poštu pro POP3. Tímto způsobem můžeme pokračovat a povolit všechny služby, o které máme zájem.

Ještě před spuštěním Shorewallu musíme zeditovat soubor shorewall.conf , přesněji jeho implicitní volbu IP_FORWARDING na hodnotu Off, protože Shorewall je přednastavený na dvouzónový firewall s IP forwardingem. Vlastní Shorewall spustíme příkazem /etc/init.d/shorewall start. Musíme dohlédnout na to, aby byl Shorewall spuštěn po startu síťového rozhraní a aktivaci firewallu v jádře, ale před všemi síťovými démony. Shorewall má spoustu dalších nastavení, můžeme přidat podporu omezování provozu (traffic shaiping), účtování přenesených dat (accounting) či černých listin (blacklist). Vše se dá poměrně snadno udržovat.

2.2. DHCP server

Konfigurace IP adres pro malé množství serverů je poměrně jednoduchá. Máme-li však konfigurovat IP adresy pro celou školu, oddělení či firmu, můžou nastat určité problémy.

Klient i server DHCP (Dynamic Host Configuartion Protocol) v Linuxu nám můžou s tímto úkolem pomoci. Díky své konfiguraci přijímá klientský počítač svou IP adresu ze sítě. Když se spustí software klienta DHCP, vyšle klient do sítě žádost o IP adresu. Když vše funguje správně, server DHCP na síti mu odpoví a pošle mu adresu společně s dalšími informacemi pro dokončení síťové konfigurace.

Toto dynamické adresování je užitečné pro konfiguraci mobilních nebo dočasně připojených počítačů. Lidé cestující z kanceláře do kanceláře mohou připojit svůj počítač k lokální síti a obdrží správnou adresu odpovídající místu.

DHCP je norma. Každý operační systém, který umí komunikovat s ostatními servery a klienty DHCP, může proto pracovat s nástroji DHCP v Linuxu. Jedno z typických řešení, při kterém se používá server DHCP, je kancelářské prostředí, kde je mnoho klientů se systémem Windows. Systémy Windows se dají nakonfigurovat pro používání DHCP a mohou kontaktovat linuxový server, který jim přidělí IP adresu. Původ DHCP je z doby, kdy nejdůležitějším parametrem byla kapacita disku. Výrobci počítačů produkovali pracovní stanice bez pevných disků. Disky se připojovaly ze sdíleného serveru. Tyto „bezdiskové“ počítače používaly pro zavádění systému protokol s názvem BOOTP (Boot

Protocol) a získávali své IP adresy ze serveru. Protokol DHCP je původně odvozen právě z protokolu BOOTP.

2.2.1. Mechanismus DHCP

Je-li součástí konfigurace klienta získávání IP adresy ze sítě, má jeho žádost o adresu formát žádostí DHCP. Server DHCP naslouchá požadavkům klienta. Po přijetí žádosti zkontroluje server svou lokální databázi a zašle vhodnou odpověď. Odpověď vždy obsahuje adresu a navíc může obsahovat i údaje o jmenných serverech, masce a implicitní bráně.

Server DHCP si ukládá seznam adres, které může přidělit. Každá adresa se přiděluje s tzv. lhůtou, která určuje, jak dlouho může klient tuto adresu používat, než je nutné znovu kontaktovat server a požádat o její obnovení. Když lhůta vyprčí, klient již nesmí danou adresu dále používat.

Implementace linuxového serveru DHCP obsahuje několik klíčových funkcí, které jsou pro mnoho implementací serverů DHCP společné. Server se dá nakonfigurovat tak, aby přidělil jakoukoli volnou adresu ze společného fondu adres, nebo aby přidělil konkrétnímu počítači konkrétní adresu. Linuxový server DHCP může kromě plnění požadavků DHCP plnit i požadavky BOOTP.

2.2.2. Server DHCP

Server DHCP (tzv. dhcpd) je odpovědný za přidělování IP adres a ostatních informací podle žádostí klienta. Protože je protokol DHCP založen na všesměrovém vysílání, musí být server v každé podsíti, pro niž bude nutné poskytovat služby DHCP. DHCP server ISC je v podstatě implementací pro unixové počítače. Tato verze je součástí mnoha distribucí Linuxu. Nejjednodušší získání je možnost jejího stažení z adresy <http://www.isc.org/> . Instalace je obdobná jako instalace všech programů pro operační systém Linux. Naším zdrojovým adresářem je /usr/local/src :

```
[root@root /usr/local/src] # tar -xvzf dhcp-3.0.2.tar.gz
```

```
[root@root /usr/local/scr] # ./configure  
[root@root /usr/local/scr] # make  
[root@root /usr/local/scr] # make install
```

2.2.3. Konfigurace serveru DHCP

Primárním konfiguračním souborem DHP serveru ISC je implicitně `/etc/dhcpd.conf`. Konfiguračním soubor vychází se dvou myšlenek. Za prvé sada deklarací popisujících síť, hostitelské počítače a skupiny připojené k systému a možný rozsah adres přiřaditelných každé příslušné entitě. K popisu více skupin nebo klientů se dají použít vícenásobné deklarace. Je-li potřeba více řešení popisů klientů nebo hostitelských počítačů, mohou být deklarace také v sobě vnořené. Za druhé sada parametrů, které popisují chování serverů a konfiguruji příslušné odpovědi. Parametry mohou být pro skupinu deklarací lokální nebo globální.

Důležité pro server DHCP je skutečnost, že každý server má jedinečnou síť s jedinečnými adresami, je nezbytné, aby měl každý server své vlastní konfigurační soubory. Stejně jako většina konfiguračních souborů v Unixu je i konfigurační soubor `dhcpd.conf` v textovém formátu ASCII a můžeme jej upravovat v libovolném textovém editoru. Obecná struktura konfiguračního souboru vypadá takto:

Globální parametry;

Deklarace 1

[parametry deklarace 1]

[vnořená deklarace]

Deklarace 2

[parametry deklarace 2]

[vnořená deklarace]

Blok deklarací sdružuje určitou skupinu klientů. Na každý blok deklarací se dají použít jiné parametry.

2.2.4. Deklarace

Z různých důvodů budeme chtít sdružovat různé klienty do skupin například podle organizačního členění, návrhu sítě nebo administrativních požadavků. Při vytváření skupin klientů nám pomůžou deklarace.

Group

Opakované rozepisování parametrů a deklarací pro každý hostitelský počítač může zkomplikovat správu konfiguračního souboru. Deklarace Group nám umožňuje použít určitou skupinu parametrů a deklarací na seznam klientů, sdílených sítí nebo podsítí.

Group label

[parametry]

[pod-deklarace]

Kde label je uživatelem definované jméno pro identifikaci skupiny. Blok parametry obsahuje seznam parametrů, které se dají použít v případě, že je třeba dále popsat další klienty, kteří mohou být členy aktuální deklarace.

Host

Deklarace host se používá tehdy, chceme-li kromě parametrů, určených pro skupinu aplikovat na konkrétní hostitelský počítač i určitou skupinu pravidel a deklarací. Je to běžné při zavádění systému s pevnou adresou nebo pro klienty BOOTP

Host label

[parametry]

[pod-deklarace]

Label je uživatelem definované jméno pro skupinu hostitelských počítačů. Položky parametry a pod-deklarace jsou stejné jako v deklaraci group.

Shared-Network

Deklarace shared-network sdružuje skupinu adres členů stejné fyzické sítě. Umožňuje to seskupování parametrů a deklarací pro administrativní účely.

shared-network label

[parametry]

[pod-deklarace]

Label je uživatelem definované jméno pro sdílenou síť. Položky parametry a deklarace jsou stejné jako v předchozích deklaracích.

Subnet

Deklarace se používá, chceme-li aplikovat skupinu parametrů nebo deklarací na skupinu adres, které odpovídají popisu této deklarace.

subnet subnet-number **netmask** netmask

[parametry]

[pod-deklarace]

Subnet-number je síť, kterou chcete deklarovat jako zdroj IP adres pro přidělování jednotlivým hostitelským počítačům. netmask je maska sítě pro podsít'.

Range

U dynamického zavádění systému specifikuje deklarace range rozsah platných adres pro přidělování klientům.

Range [dynamic-bootp] starting-adress [ending-adress];

Klíčové slovo dynamic-bootp se používá pro upozornění serveru na to, že následující rozsah adres je určen pro protokol BOOTP. Položky starting-adress a volitelná ending-adress jsou skutečnými adresami počátečního a koncového bloku IP adres. Předpokládá se, že bloky na sebe navazují a nacházejí se ve stejné podsíti adres.

Parametry

Používání parametrů změní chování serveru pro určitou skupinu klientů.

Always+reply-rfc1048

Používá se zejména pro klienty BOOTP. Existují klienti BOOTP, kteří vyžadují odezvu ze serveru, aby byly plně kompatibilní s BOOTP RFC. Díky tomuto parametru je tento požadavek splněn.

Authoritative

Parametr authoritative se používá pro označení určité sítě za autoritativní. Server implicitně předpokládá, že síť je autoritativní. Jestliže určitý segment sítě není

autoritativní, server zašle klientovi zpět kód DHCPNAP. Klient poté pravděpodobně zkusí poslat svůj požadavek znovu.

Default-lease-time seconds;

Hodnota seconds je doba alokovaná pro přidělenou IP adresu, pokud klient nepožádal o její prodloužení.

Filename filename;

V některých aplikacích musí znát klient DHCP jméno souboru, který se používá pro zavádění systému.

Fixed-address

Tento parametr se vyskytuje pouze v deklaraci host. Specifikuje skupinu adres, které je možné přidělit klientovi.

Get-lease-hostname [true | false];

Je-li zadaná hodnota true, server převede všechny adresy v rozsahu deklarací a použije pro volbu hostname.

Hardware [ethernet | token-ring] hardware-adress;

Aby mohl server identifikovat určitý hostitelský počítač, musí se použít parametr hardware. Položka hardware-adress, někdy taky nazvaná MAC adresa, je fyzická adresa rozhraní. Typicky ji tvoří skupina hexadecimálních oktětů oddělených dvojtečkami.

Max-lease-time seconds;

Klient má možnost žádostí o prodloužení trvání lhůty. Žádost je povolena pouze za předpokladu, že trvání lhůty nepřekročí počet sekund daných touto volbou.

V současné době má DHCP server kolem 60 voleb.

2.3. Spravedlivé přidělování kapacity linky QoS – pomocí algoritmu Fair User Policy – HTB

V současné době stále rostou požadavky na přenosovou kapacitu sítí. Zvýšení rychlosti přenosu bývá většinou velice nákladné a v některých případech i ne zcela možné. Jednou z možností je zefektivnit již existující technologii. A to například tak, že rozdělíme pásmo podle potřebných přenosových rychlostí. Jedna z nejlepších možností jak to provést je

CBQ (Class-Based Queueing) - rozdělení přenosového pásma na základě tříd. Každá třída (uživatel) bude mít garantováno resp. vyhrazeno tak velké pásmo podle toho, jaká je jeho požadovaná důležitost.

Vlastní konfigurace probíhá utilitou `tc`, která je součástí balíku `iproute`. Dále potřebujeme inicializační skript, který volá utilitku `tc`. V Debianu je to soubor `shaper`, který nalezneme v adresáři `/etc/shaper`. Samotná konfigurace se provádí v souborech `cbq-XXXX.název`, kde `XXXX` jsou dva bajty určující identifikaci třídy a název je libovolný popis. Mezi základní volby konfiguračního souboru patří :

DEVICE=ifname(název rozhraní),**bandwidth** (fyzická propustnost rozhraní),**weight** (1/10 fyz. propustnosti rozhraní)

RATE=speed (požadovaná rychlost)

WEIGHT=speed (speed - RATE / 10)

PRIO=1-8 nastavení priorit)

PARENT=class_id (půjčování kapacity od nadřazené třídy)

LEAF=none|tb|sfq (nastavení queueing discipline

BOUNDED=yes|no (yes zakáže půjčování kapacity od nadřazené třídy)

ISOLATED=yes|no (nastavíme-li yes, třída nebude propůjčovat svou kapacitu)

QUANTUM=bytes (specifikuje velikost paketu při použití sfq)

PERTURB=seconds (perioda transformace funkce)

RULE=srcip/prefix:srcport,dstip/prefix:dstport (určuje pakety, kterých se nastavení třídy týká)

TIME=dow, dow, dow../from-till;rate/weight (můžeme nastavit přenosovou kapacitu pro jednotlivé dny a hodiny)

Vše máme nastaveno, a proto můžeme spustit skript příkazem `/etc/init-d/shaper start`.

Pokud něco nefunguje, stačí napsat příkaz `tc -s class show dev eth1` - vypíše kompletní výpis všech tříd, přenesených, limitovaných i zahozených paketů.

2.4. Sledování sítě

Každá kvalitně postavená síť musí mít postavenou zpětnou komunikační vazbu. Touto vazbou rozumíme monitorování a sledování provozu sítě v reálném čase. Díky tomuto

můžeme sledovat, co se v síti děje, předcházet problémům a nedorozuměním, pozorovat statistiky, případně trestat neposlušné uživatele.

2.4.1. IPTraf

Program, který se dá k takovéto činnosti využít, se nazývá IPTraf. IPTraf je konzolový program se schopností vytvářet statistické přehledy, které umí i logovat, a který podporuje všechny důležité protokoly. Pro zřejmost uvádím na prvním místě malý přehled protokolů podporovaných tímto programem.

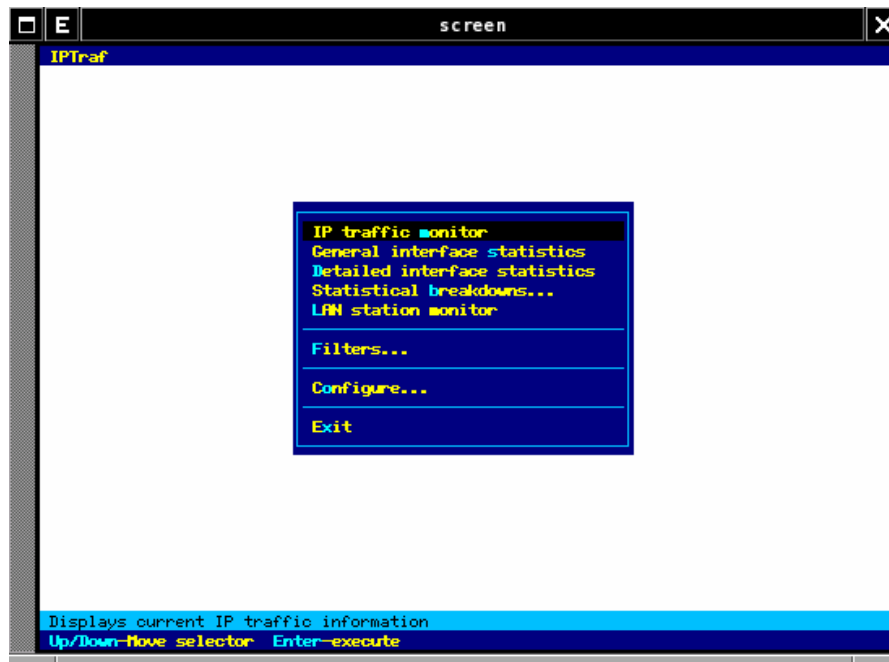
- *IP* - Internet Protocol
- *TCP* - Transmission Control Protocol
- *UDP* - User Datagram Protocol
- *ICMP* - Internet Control Message Protocol
- *IGMP* - Internet Group Management Protocol
- *IGP/IGRP* - Interior Gateway Protocols/ Interior Gateway Routing Protocol

2.4.2. Instalace

Instalace probíhá tradičním způsobem. Po stažení instalačních balíčků zdrojové kódy rozbalíme a provedeme překlad `configure`, `make`, `make install`. Pro Debian můžeme použít příkaz v příkazovém řádku `apt-get install iptraf`.

2.4.3. Sledování sítě

Obsluha programu IPTraf probíhá v textovém řádku a veškeré funkce jsou logicky rozděleny do nabídek, které se zobrazí po spuštění. Vše můžeme obsluhovat pomocí klávesnice, případně klávesových zkratk.



Obr. 2. Program IPTraf

2.4.4. Monitor IP provozu

Po jeho zvolení můžeme sledovat libovolné síťové rozhraní, které chceme sledovat. Lze si vybrat lokální (*lo*, loopback) nebo síťové rozhraní (např. *eth0*, *eth1* apod.). Program detekuje a umožňuje sledovat následující rozhraní:

- lokální *loopback* rozhraní
- všechna jádrem podporovaná ethernetová rozhraní
- všechna *FDDI* rozhraní podporovaná jádrem
- *SLIP*
- asynchronní *PPP*
- synchronní *PPP* přes *ISDN* (Integrated Services Digital Network)
- *ISDN* s "*Raw IP*" enkapsulací (zapouzdřením)
- *ISDN* s *Cisco HDLC* enkapsulací
- *IP* pro paralelní linky

2.4.5. Obecné statistiky o rozhraních

Dalším režimem programu IPTraf jsou obecné statistiky o rozhraních. V dialogu můžeme sledovat počet bloků dat přenesených přes naše síťová rozhraní, názvy síťových rozhraní, celkový počet prošlých paketů, počet IP datagramů, „Non-IP“ bloků dat, vadné IP datagramy a aktivitu neboli tempo toku na každém rozhraní. Je-li zapnuto logování, je kopie statistik zapisována do souboru /var/log/iptraf/rozhraní stats general log.

2.4.6. Detailní statistika rozhraní

Tato služba poskytuje přehlednou statistiku o vybraném rozhraní, která je zobrazena v uspořádané tabulce. Ve výpise vidíme úplné počty přenesených paketů a bytů, počty příchozích paketů a bytů, odchozích paketů a bytů pro všechny protokoly, čili pro protokoly IP, TCP, UDP a ICMP v samostatných řádcích a pro ostatní protokoly a Non-IP v dalších řádcích.

2.4.7. Statistické rozbor

V položce Statistické rozbor se nám zobrazí podnabídka, ve které se můžeme rozhodnout, zda chceme rozbor dle velikosti paketů nebo dle TCP/UDP portu. Po výběru „dle velikosti paketů“ se nám objeví tabulka, kde snadno zjistíme, kolik paketů o různých velikostech projde přes rozhraní. Při druhé volbě pozorujeme kolik paketů a bytů přišlo na určitý port paketem TCP či datagramem UDP.

2.4.8. Monitor LAN stanice

Monitor LAN stanice je poslední volbou, kterou máme k dispozici v programu IPTraf. Můžeme si přet zvolit, zda chceme sledovat všechna rozhraní nebo pouze jediné. Informace nalezneme ve zdvojených řádcích, kde se specifikuje typ stanice (Ethernet, PLIP, Token

ring). Druhý řádek nás informuje o příchozích a odchozích počtech paketů, celkový počet IP datagramů a bytů.

2.4.9. Filtry

Velmi užitečnou součástí programu jsou filtry. Poskytují nám mnoho možností navíc, ať už využíváme předdefinované filtry nebo námi nově vytvořené filtry. Filtry se aplikují na protokoly TCP, UDP, ARP, RARP a v rámci Non-IP či ostatní IP datagramy. V definovaných filtrech lze nastavit hodnoty hostname či IP adres obou konců spojení, masky, porty s možnostmi filtrů nastavit do výpisu „include“ nebo „exclude“.

Program disponuje dalšími možnostmi, které ovlivňujeme v nastavení programu. Hojně využívané bývá logování nebo příkazový řádek. IPTraf je program velmi jednoduchý, snadno se ovládá, je rychlý a nenáročný na zdroje.

3. TECHNOLOGIE PŘIPOJENÍ

3.1. CDMA

Jedná se o bezdrátové vysokorychlostní připojení na internet. CDMA, neboli Code Division Multiple Access je mnohonásobný přístup do sítě kódovým dělením. Jde o připojení podobnému GPRS, ale několikanásobně rychlejší a využívající jiné funkční frekvence. Pracuje v pásmu 450 MHz.

Technologie CDMA využívá starou analogovou NMT síť, kterou Eurotel provozuje již od počátku devadesátých let. Datová síť Eurotel je založena na standardu CDMA 2000 1xEV-DO, který patří mezi standardy určené pro síť třetí generace. Teoreticky umožňuje CDMA rychlost až 2.48 Mb/s , kvůli dodržení kvality poskytované služby však Eurotel omezil max. rychlost na 800kb/s a pro případné zájemce je nutné uvést, že v současné době se rychlost stahování pohybuje kolem 200 - 350 kb/s a odesílání cca 80 - 156 kb/s.

Nevýhodou CDMA je jeho prozatím nižší pokrytí. Eurotel uvádí 80 % pokrytí, nikoliv však území, ale populace. Jinak řečeno 8 z 10 Čechů by mělo být v dosahu CDMA. Bohužel praxe ukazuje, že skutečné pokrytí neodpovídá přesně mapě pokrytí, kterou prezentuje Eurotel.

CDMA je nejvýhodnější, když budete chtít využívat výhod mobilního internetu, například prostřednictvím připojení CDMA modemu k přenosnému počítači. V současné době je sice výběr modemů velice omezený, dá se ale předpokládat, že v nejbližší budoucnosti se na trhu objeví nejen CDMA modem v podobě PCMCIA karty, ale i jako součást mobilního telefonu.

Pro naše potřeby je technologie CDMA nevhodným řešením, protože obec Provodov není pokryta signálem CDMA. Z osobních zkušeností tuto technologii velmi nedoporučuji. Měl jsem možnost využít akci Eurotel na vyzkoušení této technologie na 14 dnů zdarma. Kvalita a rychlost je proměnlivá, od výborných hodnot blízkých udávaného maxima klesá až někde k úrovni dial-up spojení. Druhým problémem byla instalace, kdy dva ze tří počítačů mi odmítli komunikovat s modemem. Navíc dle informací techniků Eurotelu si pro zesíťování musíme zakoupit router za další 4.tisíce korun.

3.2. EDGE-GRPS

Zkratkové slovo EDGE skrývá označení Enhanced Data Rates for Global Evolution, technologie používá pro svůj provoz stejné spektrum používané GSM telefony. Díky začlenění pěti nových kódovacích schémat a jinému způsobu modulace (8PSK) je možné v síti podporující EDGE dosáhnout maximální rychlost downloadu i uploadu 236,8 kb/s při použití osmi timeslotů, čtyři pro upload a čtyři pro směr k uživateli. V praxi je ale zatím nasazováno EDGE (EGPRS) Class 10, které nabízí 4+2 timesloty a rychlost uploadu je tak poloviční ve srovnání s downloadem.

Ve srovnání s CDMA, které nabízí Eurotel, má EDGE výrazně horší dobu odezvy (ping), typicky u CDMA se tato hodnota pohybuje kolem 150 ms, u EDGE je čtyřikrát vyšší, což znemožňuje aplikaci technologie voice over IP (internetová volání) nebo hraní online akčních her. Výhodou ale je rozšířená podpora EDGE v nových telefonech, kdy nejsou potřeba žádné specializované ovladače, které způsobují u CDMA problém uživatelům Linuxu.

3.3. WIFI

WiFi je bezdrátová, síť určená primárně k náhradě kabelového ethernetu v bezlicenčním pásmu, které je dostupné prakticky v celém civilizovaném světě. Samotný název WiFi vytvořilo WECA – (Wireless Ethernet Compatibility Alliance) a v principu jde o bezdrátovou technologii v bezlicenčním nekoordinovaném pásmu 2,4 GHz (ISM - Industry, Science, Medical), založenou na protokolu 802.11b. WiFi je pouze komerční název, který je fakticky pouze podmnožinou 802.11b. Hlavní výhodou této technologie je její nízká cena, způsobená mimo jiné tím, že certifikovaná zařízení jsou k dispozici ve velkých sériích.

V úvodní fázi vývoje se uplatnila především v historických budovách a posléze nalezla široké využití pro připojení k Internetu.

4. KONFIGURACE SERVERU

4.1. Transparentní HTTP proxy Squid

Proxy-server samotný většinou slouží jako relativně bezpečné spojení dvou sítí, které mezi nimi umožňuje jen přesně definovanou komunikaci. Můžeme například povolit HTTP/FTP přístup jen vybrané skupině lidí nebo počítačů k vybraným zdrojům. Počítače nemusí být schopné samy komunikovat s druhou sítí. Přínosem proxy-serveru je především možnost oddělení sítí a umožnění jen vybrané komunikace.

Nespornou výhodou Squidu je ušetření na provozu, který vzniká při otevírání stejných dokumentů jednotlivých uživatelů v síti. Naopak problémem může být aktuálnost požadovaných informací, proto musíme zvolit kompromis při konfiguraci mezi aktuálností a provozem na síti. Typickým případem pro využití Squidu je připojení naší sítě s Internetem.

Základní rozdíl mezi proxy serverem a transparentním je v tom, že proxy se nastaví v prohlížeči u každého uživatele, kdežto transparentní se v prohlížeči nenastavuje. Nastavuje se ve firewallu tak, aby veškerý provoz směřující z portu 80 byl přesměrován na port 3128.

Konfigurace se nastavuje v souboru squid.conf. Je vhodné nastavit http_port (číslo portu, kam se připojují klienti) a cache_mem (velikost „keše“ v paměti). Složitější je nastavení ACL (Access Control List). ACL třídy se definují pomocí řádek začínajících acl a požadavky do jednotlivých tříd můžeme řadit podle opravdu mnoha různých parametrů: IP, port, čas, cílová IP, cílová URL, vlastní autentizace a další. ACL operátory pak, na základě tříd, ve kterých požadavek je, určují, co s ním bude dál. Veškeré logy jdou do adresáře /var/log/squid. cache, kde můžeme analyzovat případné chyby. Vlastní transparentní server používáme tak, že veškerý provoz je přesměrován na Squid.

4.2. FTP Server

FTP je jednou z nejvíce používaných služeb v bezdrátových sítích. Slouží k přenosu mezi lokálním a vzdáleným počítačem. FTP používá přenosový protokol File Transfer Protocol. Pro přístup k FTP serveru se používá nejčastěji FTP klient, který poskytuje výhody grafického prostředí a nevyžaduje hlubší znalosti FTP protokolu.

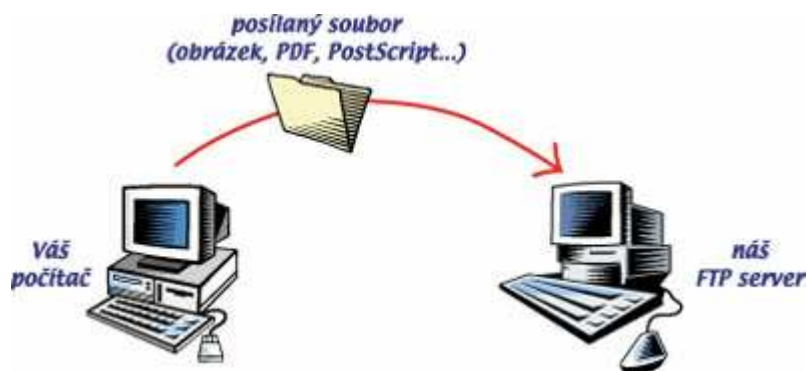
Když se chce FTP klient připojit k serveru, náhodně zvolí dostatečně vysoké číslo zdrojového portu (vyšší než 1024) a jako cílový port zadá č.21. Port 21 je definován standardem FTP. Když klient zašle požadavek na přenos souboru, vybere server opět náhodně číslo portu, zašle je klientovi a čeká, až jej klient na tomto portu kontaktuje. Klient použije další z náhodně vybraných portů k vytvoření nového spojení. Původní spojení zůstává otevřeno, aby si klient se serverem mohli vyměňovat doplňující informace např. příkaz k přerušení přenosu.

Tato architektura vznikla na počátku 70. let 20 století a vyšla z tehdy platného předpokladu, že uživatele Internetu jsou odborníci, kteří budou síť využívat bez postranních úmyslů. To se zpočátku dařilo zajistit díky faktu, že infrastrukturu pro Internet poskytovala americká Národní vědecká nadace (NSF – National Science Foundation) a komerční firmy měly přístup k síti pouze tehdy, účastnily-li se vědeckých projektů společně s některým z výzkumných ústavů. Většina účastníků tak pocházela z akademické sféry.

Mezi lety 1990 1991 přestala NSF poskytovat finance na provoz internetové infrastruktury a její úlohu musela převzít komerční sféra. Zpočátku to nevadilo, ale pak byla uvedena služba World Wide Web a počet uživatelů prudce stoupl. To samozřejmě přineslo problémy s bezpečností. Řada společností dnes používá firewally, které mají jejich vnitřní síť chránit proti „zlému Internetu“. Firewally ovšem považují přichozí spojení mezi porty s vysokými čísly za velmi nebezpečná, a tak často sledují požadavky přicházející přes FTP a spojení na portech s vysokými čísly povolují pouze tehdy, je-li třeba přenášet data ze vzdáleného počítače do lokální sítě.

Je nutné si uvědomit, že v době, kdy byl standard FTP vytvořen, byla velmi důležitá otázka přenosové kapacity. Když chtěl někdo přenést soubor z místa A do místa B a přitom seděl v místě C, musel nejdřív přenést soubor z místa A do C a pak do B, což vedlo k zbytečnému plýtvání přenosovou kapacitou. Tvůrci protokolu FTP ale přišli s řešením:

nalezli způsob, který dovolí uživateli v místě C iniciovat přenos souboru z A do B. Tento způsob se označuje jako pasivní přenos.



Obr. 3. Princip technologie FTP

4.2.1. Instalace ProFTPD serveru

Nejnovější verzi aplikace (démon proftpd) je možné získat na adrese <http://www.proftpd.org/>. Další informace o ProFTPD lze získat na adrese <http://www.proftpd.org/docs/>. V současné době je aktuální verze 1.2.10.

Balíček stáhneme na disk a můžeme jej rozbalit na `/usr/local/src` příkazem :

```
[root@root /usr/local/src] # tar -xvzf proftpd-1.2.10.tar.bz2
```

Po důkladném prostudování souborů README a INSTALL nacházející se v kořenovém adresáři softwarového balíčku. Následuje konfigurace, která se provádí příkazem:

```
[root@root /usr/local/src/proftpd-1.2.10] # ./configure
```

Skript *configure* implicitně použije jak předponu cestu `/usr/local` a nainstaluje většinu binárních souborů ProFTPD do tohoto adresáře. Nyní k překladu programu použijeme nástroj *make*:

```
[root@root /usr/local/src/proftpd-1.2.10] # ./make
```

Po bezproblémovém přeložení zadáme příkaz *make install*, čímž se instalace dokončí. Na náš systém se nainstalují čtyři binární soubory. `Ftpcount` a `ftpwho` se instalují do `/usr/local/bin`. `Proftpd` a `ftpsht` se instalují do adresáře `/usr/local/sbin`. Nástroj `ftpcount`

počítá, kolik uživatelů FTP se přihlásilo do našeho systému. Nástroj `ftppwho` pak zobrazuje uživatelské názvy lidí přihlášených do systému. Konečně binární soubor `proftpd` představuje samotný démon FTP, který vykonává samotnou práci serveru.

4.2.2. Konfigurace ProFTPD serveru

Konfigurační soubor pro ProFTPD je implicitně uložen v adresáři `/usr/local/etc/` a soubor se jmenuje `proftpd.conf`. Pokud jsme instalovali ProFTPD s předponou nastavenou na `/`, souborem bude `/etc/proftpd.conf`. Uvádím příklad konfiguračního souboru dodaného s ProFTPD. Podívejme se na některá nastavení serveru. Direktivu **ServerName** lze upravit dle přání. Je to jméno, které se bude objevovat při přihlášení. **ServerType** v našem příkladu by se měl nastavit na `inetd`, chceme-li provozovat ProFTPD samostatně, lze ponechat direktivu na `standalone`. Typ serveru `standalone` běží pořád a poslouchá na portu. **DefaultServer** on # slouží pro virtuální servery, pro jeden server je nepotřebný. **ServerAdmin** j_jaros@ft.utb.cz # je email na administrátora serveru. **Port** je direktiva, která určuje, na kterém portu běží FTP. **AccessGrantMsg** "User %u logged in." # je zpráva, kterou napíše server po správném zalogování, v našem případě je to uživatel ten a ten je zalogovaný. **MaxInstances** 30 je maximum spuštěných procesů, slouží jako obrana proti DoS útoku. **MaxClients** 10 "Promiň, max %m uživatelů – zkus později" je hláška, která nám udává maximální počet současně přihlášených uživatelů. **Umask** 022 # maska, která se aplikuje na nové vytvořené soubory a adresáře. Jistě si všimnete globálních direktiv **User** a **Group**. Ty říkají serveru, jako který uživatel a skupina má běžet. Kvůli bezpečnosti je dobré obě nastavit na `nobody`. Samozřejmě musí v našem systému existovat uživatel `nobody` a skupina `nobody`. Nebudou-li uživatelé existovat, server ProFTPD nebude fungovat. Nyní se dostáváme k direktivě `<Anonymus ~ftp>`. Zde můžeme nastavit různé parametry pro anonymní přístup na váš FTP server. Direktivy **User** a **Group** se implicitně liší od globálního uživatele a skupiny. Proces serveru tak může nechat vystupovat anonymní uživatele jako uživatele `ftp` a skupinou `ftp`. Protože jsme anonymní přístup nastavili tímto způsobem, domovský adresář uživatele `ftp` se bude používat pro anonymní přístup. Zkontrolujeme lokální soubor hesel, abychom viděli, kde domovský adresář `ftp` leží. Většinou to bývá adresář `/var/ftp`. Nakonec zmíním i direktivu `<Limit WRITE>`, jež říká FTP serveru, aby nenechal žádného anonymního uživatele na server cokoli zapisovat.

4.2.3. Řízení přístupu k FTP

K démonu FTP lze přidělit dva druhy přístupu. První je tzv. anonymní přístup, zatímco druhý umožňuje, aby se uživatel přes FTP přihlásil do svého domovského adresáře.

- **Povolení přístupu pro běžné i anonymní uživatele**

Podle implicitního konfiguračního souboru dodávaného s ProFTPD se mohou do systému přihlašovat anonymní i běžní uživatelé. ProFTPD je implicitně nastaven tak, aby se k ověřování totožnosti uživatelů používaly soubory /etc/passwd a /etc/shadow.

- **Povolení pouze anonymního přístupu**

Chceme-li, aby k našemu FTP serveru měli přístup pouze uživatelé ftp nebo anonymous, musíme použít direktivu :

```
<Limit LOGIN>
```

```
DenyAll
```

```
</Limit LOGIN>
```

Uvedená direktiva říká FTP serveru, aby zakázal přístup běžným uživatelům. V rámci direktivy <Anonymous> musíte potlačit globální direktivu <Limit LOGIN>:

```
<Anonymous ~ftp>
```

```
<Limit LOGIN>
```

```
AllowAll
```

```
</Limit >
```

- **Povolení přístupu pouze pro běžné uživatele**

Tato možnost je přesným opakem výše uvedeného příkladu. V bloku direktivy <Anonymous> je nutno nastavit DenyAll.

Po provedení všech změn je nutné server odzkoušet a zkontrolovat, že se mohou přihlásit jen ti uživatelé, pro které jsme chtěli přístup povolit.

4.2.4. Jak přidat FTP do xinetd

Chceme-li ProFTPD spouštět pomocí xinetd, měli bychom přidat soubor do adresáře /etc/xinetd.d/ jenž popisuje, která služba se pro FTP spouští. Uvedu příklad konfiguračního souboru xinetd, konkrétně /etc/xinetd.d/ftp:

Service ftp

```
{  
  
    flags = REUSE  
  
    socket_type = stream  
  
    instance = 50  
  
    wait = no  
  
    user = root  
  
    server = /usr/local/sbin/proftpd  
  
    bind = 192.168.1.1  
  
}
```

Po změnách konfigurace a nastavení *xinetd.conf* musíte *xinetd* restartovat, aby reagoval na žádosti FTP. To uděláme jednoduchým znovuspuštěním *xinetd* prostřednictvím skriptů *rc*:

```
[root@root /] # /etc/init.d/xinetd restart
```

4.3. SMTP server

Protokol SMTP (Simple Mail Transfer Protocol) je v podstatě normou pro přenos pošty po Internetu. Tuto normu podporují všichni, kdo chtějí na svém serveru přijímat a odesílat poštu přes Internet. SMTP je také protokol aplikační vrstvy, který dovoluje zasílání a přijímání zpráv (využívá služeb protokolu TCP/IP). Je možné jej používat i v soukromých poštovních službách i v mnohých interních sítích, protože je nezávislý na platformách a je dostupný na všech užívaných operačních systémech.

4.3.1. Mechanismus SMTP

Protokol SMTP definuje způsob, jakým se posílají zprávy s jednoho hostitelského počítače na jiný. Nedefinuje způsob ukládání zpráv nebo jak by měl poštovní server posílat poštu, aby byla pro příjemce čitelná.

Právě v jednoduchosti SMTP je jeho síla. Tato jednoduchost je dána dynamickou povahou sítí z počátku 80.let. Protokol byl vyvinut v roce 1982. SMTP byl první normou pro elektronickou poštu, která byla nezávislá na způsobu přenosu.

Protokol SMTP je také nezávislý na operačních systémech. Každý systém si tak může ukládat poštu svým způsobem bez ohledu na to, jak ji ukládá odesílatel. Je to podobné fungování telefonní sítě: každý poskytovatel telekomunikačních služeb má vlastní nezávislý evidenční systém; mezi jednotlivými poskytovateli však musí existovat dohoda o nějakém standardním způsobu spojení jejich sítí, aby byl přenos telefonátů z jedné sítě do druhé průhledný.

4.3.2. Základy SMTP

V tomto příkladu chci demonstrovat, jak protokol SMTP zasílá zprávy z jednoho hostitelského počítače na jiný. Můj počítač je odesílajícím hostitelským počítačem a kterýkoliv počítač, ke kterému se připojuji, je příjemcem.

Protokol SMTP vyžaduje od hostitelského počítače pouze schopnost zasílat prostý text v ASCII na jiný hostitelský počítač. Obvykle se to realizuje kontaktováním portu SMTP (port 25) na poštovním serveru. Můžeme to provést pomocí programu Telnet:

```
[root@root /root] # telnet mailserver 25
```

Hostitelský počítač *mailserver* je poštovní server příjemce. Číslo 25 informuje službu Telnet o skutečnosti, že chcete komunikovat s portem 25 serveru a nikoliv s normálním portem 25 (Port 23 se používá pro vzdálená přihlášení a port 25 pro server SMTP). Poštovní server odpoví pozdravem:

```
220 mail ESMTP Sendmail 8.9.3/8.9.3; 26. Apr 2005 22:10:08 -0700 (PDT)
```

Existuje celá řada příkazů SMTP, ale čtyři nejdůležitější jsou:

HELO

MAIL FROM:

RCPT TO:

DATA

Příkaz HELO se používá, když se klient představuje serveru. Parametr tohoto příkazu je jméno hostitelského počítače, který iniciuje spojení. Většina webových serverů této informaci příliš nedůvěřuje a raději si ji ještě kontrolují.

Příkaz MAIL FROM: vyžaduje parametr e-mailové adresy odesílatele. Sděluje tak poštovnímu serveru původ e-mailu, např.

MAIL FROM: dugarry07@tiscali.cz , znamená, že zpráva je od odesílatele :
dugarry07@tiscali.cz.

Příkaz RCTP TO: také vyžaduje parametr – adresu elektronické pošty. Tato e-mailová adresa je adresou příjemce zprávy.

Díky těmto příkazům server ví, kdo je odesílatel a kdo příjemce a teď se ještě musí dozvědět, jakou zprávu má odeslat. K tomu slouží příkaz DATA. Po jeho zadání očekává server celou jeho zprávu s příslušným záhlavím, po němž následuje prázdný řádek, dále tečka a další prázdný řádek. V následujícím příkladu tedy odešle dugarry07@tiscali.cz příjemci j_jaros@ft.utb.cz následující zprávu:

DATA

354 Enter mail, end with “.” On a line by itself

From: Dugarry<dugarry07@tiscali.cz>

To: Jaros <j_jaros@ft.utb.cz>

Subjekt: Seznam doporučené literatury.

Date: Sat, 1 Apr 2005 08:00:10 -0700 (PDT)

Posílám Ti seznam doporučené literatury, fešáku!

250 NAA28719 Message accepted for delivery

A to je téměř vše, Pro ukončení stačí již napsat pouze příkaz QUIT.

4.3.3. Poštovní služba

Poštovní služba má tři základní komponenty. Tzv. mail user agent (MUA) neboli pošťák je to, co vidí, tj. programy jako Eudora, Outlook a Pine. Pošťák je zodpovědný pouze za přečtení zprávy a vytvoření nové. Mail transfer agent (MTA) neboli poštovník řídí proces přenosu zprávy z jednoho serveru na jiný; takovým poštovníkem je Sendmail. A konečně mail delivery agent (MDA) neboli poštovní doručovatel se stará o doručení zprávy ze serveru do poštovní schránky uživatele. Mnoho poštovních systémů tyto komponenty integruje. Server Microsoft Exchange nebo Lotus Domino např. slučuje funkce poštovníka a poštovního doručovatele do jednoho systému. Naproti tomu Postfix pracuje pouze jako poštovník a přenechává úkol doručení lokální pošty jinému programu. Mnohé starší konfigurace stále používají pro doručování pošty implicitní program `/bin/mail`.

4.3.4. Instalace serveru Postfix

V této kapitole si probereme instalaci Postfixu ze zdrojového kódu na náš linuxový systém. V níže uvedených krocích si ukážeme, jak Postfix přeložit a nainstalovat na náš systém.

S adresy <http://www.postfix.org/> si stáhneme poslední verzi postfix-2.1 a rozbalíme zdrojový kód z archivu tar zabaleného gzipem.

```
[root@root /usr/local/src] # tar -xvzf postfix-2.1.tar.gz
```

Změníme aktuální adresář na postfix-2.1

```
[root@root /usr/local/src] # cd postfix-2.1
```

Zadáme příkaz make.

```
[root@root /usr/local/src/postfix-2.1] # make
```

Vytvořím uživatele postfix a skupinu postdrop. Ve skupině postdrop nesmí být žádný uživatel.

```
[root@root /usr/local/src/postfix-2.1] # useradd postfix
```

```
[root@root /usr/local/src/postfix-2.1] # groupadd postdrop
```

```
[root@root /usr/local/src/postfix-2.1] # make install
```

Ve výši uvedených krocích jsme přeložili a nainstalovali poštovní systém Postfix. Skript `make install` nám dá vědět, které změny je případně nutno provést, např. zapomeneme-li přidat uživatele postfix. Po instalaci Postfix můžeme změnit adresáře na `/etc/postfix` a server Postfix nakonfigurovat.

4.3.5. Konfigurace serveru Postfix

Vlastní konfiguraci poštovního serveru Postfix provedeme prostřednictvím souboru `/etc/postfix/main.cf`. Dalším konfiguračním souborem je `master.cf`. Jedná se o soubor pro konfiguraci procesů serveru Postfix. Může být užitečný pro nastavování Postfixu na klientech, aby např. neakceptovali e-maily a přeposílal je na centrální poštovní uzel.

Soubor `main.cf` je příliš velký na to, abychom rozebrali všechny jeho volby, probereme se pouze ty hlavní a nezbytné pro spuštění a provozování našeho poštovního serveru.

Myhostname se používá jako parametr k nastavení názvu, pod kterým bude Postfix přijímat poštu. Můžeme jej nazvat **myhostname** = mail.jj.com

Mydomain určuje hlavní doménu, kterou budeme obsluhovat např. mydomain = jj.com.

Myorigin – veškerá elektronická pošta odeslaná z našeho emailového serveru bude reprezentovat svůj původ pomocí tohoto parametru. Můžeme jej nastavit buď na `$myhostname` nebo `$mydomain` např. myorigin = \$mydomain. Pomocí znaku `$` můžeme používat proměnné ostatních parametrů.

Mydestination – tento parametr uvádí, které bude poštovní server Postfix považovat za konečný cíl doručení. Tato hodnota se zpravidla nastavuje na název hostitelského počítače, název domény, přičemž může obsahovat další názvy např. mydestination = \$myhostname, localhost.\$mydomain, \$mydomain, mail.\$mydomain, ftp.\$mydomain

Mail_spool_directory – poštovní server Postfix můžeme provozovat ve dvou režimech doručování – přímo do poštovní schránky uživatele, nebo na centrální adresář pro ukládání elektronické pošty. Elektronická pošta se zpravidla ukládá do `/var/spool/mail`. Obě varianty mají svá pro i proti. Proměnná v konfiguračním souboru vypadá takto: mail_spool_directory = /var/spool/mail

Mynetworks – proměnná mynetworks je z hlediska konfigurace velmi důležitá. Umožňuje nastavit, které servery mohou přenášet poštu přes náš server Postfix. Obvykle budeme

chtít, aby to bylo pouze z lokálních klientských stanic a ničeho jiného. V opačném případě se vystavujeme riziku zneužití ze strany různých spammerů. Proměnou bychom mohli nastavit takto: `mynetworks = 192.168.1.0/24, 127.0.0.8`

Nastavíme-li tento parametr, potlačíme tím možnost nastavit argument `mynetworks_style`. Parametr `mynetworks_style` umožňuje specifikovat klíčová slova jako `vlase`, `subnet` či `host`. Těmito nastaveními se říká serveru, aby důvěřoval uvedeným sítím, do kterých server patří.

4.3.6. Provozování serveru Postfix

Po instalaci a nastavení systému Postfix jej musíme zprovoznit.. Spuštění poštovního serveru Postfix je velmi snadné. K příkazu Postfix uvedeme parametr `start`:

```
[root@root /usr/bin/ ] # postfix start
```

Uděláme-li jakékoliv změny v konfiguračních souborech, musíme dát Postfixu signál, aby změny znovu zavedl příkazem `reload`:

```
[root@root /usr/bin/ ] # postfix reload
```

Může se stát, že dojde k zaplnění fronty zpráv elektronické pošty. Příčin může být několik - poruchy na síti nebo problémy s jiným poštovními servery. Chceme-li zkontrolovat frontu na našem poštovním serveru, zadáme příkaz:

```
[root@root /usr/bin/ ] # mailq
```

Tímto příkazem se zobrazí veškeré zprávy ve frontě serveru Postfix. Jedná se o nejjednodušší způsob ověření, zda funguje poštovní server správně. Dalším důležitým příkazem je `newaliases`. Soubor `/etc/aliases` obsahuje seznam aliasů elektronické pošty. Používá se k vytvoření seznamů a aliasů elektronické pošty pro celé pracoviště. Kdykoliv provedeme změny v souboru `/etc/aliases`, musíme o nich zpravit Postfix, a to příkazem `newaliases`. Tímto příkazem se aktualizuje databáze systému Postfix a budeme informováni o tom, kolik jmen bylo doplněno.

Na závěr bychom měli provést ještě celkovou kontrolu systému. Prvním krokem je použít lokální poštovní program např. Pine nebo mutt a odeslat mail sobě samému. Pokud vše funguje, pošleme mail na vzdálený počítač přičemž kontrolu provedeme užitím příkazu

mailq, abychom viděli, kdy byla zpráva odeslána. Posledním krokem je odeslání zprávy mimo naši síť tj. na Internet. Můžeme-li přijímat elektronickou poštu od zbytku světa, jsme hotovi.

4.4. Jabber server

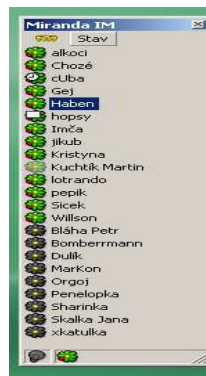
4.4.1. Princip jabberu

Co je to Jabber? Je to progresivní komunikační systém umožňující rychlé zasílání krátkých zpráv po Internetu (IM – instant messaging). Na rozdíl od konkurenčních programů nabízí Jabber několik výhod. Jedná se o otevřený systém na bázi XML, což znamená, že je kompatibilní s většinou konkurenčních sítí. V praxi to znamená, že nepotřebujete jednoho klienta pro ICQ, dalšího pro MSN messenger nebo Yahoo!

Další jeho výhodou je jeho rozšířenost. V současné době existují klientské programy pro téměř všechny dostupné operační systémy - Windows, Linux, MacOS, Windows CE, Javu a dokonce i pro mobilní telefony. Je to díky tomu, že protokol Jabberu, server i většina klientů jsou publikovány pod licencí GNU GPL (open-source).

S Jabberem se “neupisujeme” žádné firmě jako třeba v případě ICQ, jež náleží AOL. Můžeme si vybrat klientský program i server, na který se přihlašujeme. Seznam našich kontaktů je uložený na serveru a my tedy můžeme komunikovat odkudkoliv na světě a díky množství klientů i z libovolného operačního systému. Díky použití jabberapletu nebo webjabu se můžete přihlásit dokonce i přes webový prohlížeč. Navíc Jabber vás neotravuje reklamou.

Jabber splňuje všechny podmínky bezpečnosti. Naše konto chrání heslem, se serverem se můžeme spojit přes kódovaný SSL protokol a můžeme využít šifrování a digitální podpis přes PGP.



Obr. 4. Komunikace přes ICQ

4.4.2. Kompilace serveru

Dříve než zahájíme vlastní kompilaci Jabber serveru, musíme zajistit, aby byly nainstalovány knihovny libpth, libdl a libresolv. Vesměš jde o relativně základní knihovny a neměl by být problém je doinstalovat do systému. Pokud budeme chtít používat SSL, musíme doinstalovat i knihovny s příslušnými hlavičkovými soubory.

Prvním krokem vlastní kompilace je stažení zdrojových souborů z adresy <http://www.jabber.org>. Vytvoříme adresář `/usr/local/jabber/` a zkopírujeme do něj zdrojové soubory. Následuje rozbalení balíčku:

```
[root@root /usr/local/src] # gzip -d jabber-1.4.1.tar.gz
```

```
[root@root /usr/local/src] # tar -xf jabber-1.4.1.tar
```

Tímto příkazem se vytvoří adresář `jabber-1.4.1` a po něm následují dva obligátní skripty **./configure** a **make**.

Po kompilaci se vytvoří v adresáři `jabberd` očekávaný démon - `jabberd`. V dalších adresářích jsou pak knihovny (přípona `.so`), které budeme ještě potřebovat. Pro každého agenta (službu či transportní vrstvu) najdete v archivu soubor, ve kterém je uvedeno, jak jej zkompileovat a nastavit.

4.4.3. Konfigurace serveru

V adresáři /directory/jabber/jabber-1.4.1/ se nachází nejdůležitější soubor nastavení jabber.xml. Soubor je dostatečně okomentovaný a stačí v něm změnit pouze pár drobností. Jabber server umožňuje jen několik málo služeb. Pokud požadujeme přímo službu serveru, je její jméno shodné s doménovým jménem stroje, na kterém server běží. Pokud ovšem chceme něco navíc např. přístup do sítě ICQ nebo třeba využít konferenci, musíme se odkázat na jméno služby, kterou požadujeme. Jméno služby je pak složeno z jejího názvu (icq, conference, ...) a jména serveru. Například server se jménem jabbernet.cz poskytuje službu ICQ se jménem icq, celé jméno služby je pak icq.jabbernet.cz.

4.4.4. Nastavení serveru

Celá konfigurace se provádí v souboru jabber.xml a to včetně všech požadovaných agentů. Pokud použijeme při startu démona přepínač -h, pak nemusíme v konfiguračním souboru jabber.xml dělat žádné změny, ale to není správný přístup, a proto nastavíme v sekci <service id="sessions"> úplné doménové jméno našeho serveru. V našem případě to bude název jabbernet. Doporučuji přepsat uvítací hlášení začínající tagem <welcome> a také můžeme změnit část týkající se administrátora, která začíná tagem <admin>. Na uvážení každého administrátora je, zda přepsat část týkající se informací o serveru <vCard> a informace vypisující se při zakládání nového konta - <register notify="yes">.

Velmi důležitou sekci je část konfiguračního souboru jabber.xml začínající tagem <browse>. V této sekci jsou umístěny informace o jednotlivých službách neboli agentech. Informace zde uložené se vypisují uživateli v jeho klientovi.

Pro ověření správnosti nastavení vašeho konfiguračního souboru jabber.xml existuje webová stránka <http://www.xml.com/pub/a/tools/ruwf/check.html> , kde si můžete ověřit, zda-li je náš soubor korektně nastaven. Nesprávně nastavený konfigurační soubor démon nespustí.

4.4.5. Spuštění serveru

Ačkoliv máme v konfiguračním souboru jabber.xml doménové jméno serveru, přesto je dobré spouštět server s parametrem `-h`. dalším užitečným parametrem je `-D`, který spouští server v debug režimu. Náš server bychom mohli spouštět následujícím příkazem:

```
cd /directory/jabber/jabber-1.4.1/  
./jabberd/jabberd -h jaberrnet.cz
```

Provozní informace se ukládají implicitně do adresáře `/spool/` podle jednotlivých agentů. Standardně jsou k dispozici i dva logy s příponou `.log`, které obsahují informace o běhu serveru.

4.5. VPN server (Poptop)

VPN (Virtual Private Network) umožňuje vytvořit přímé spojení mezi dvěma body, reprezentované v tomto případě ptp klientem a ptp serverem, které je zabezpečené pomocí šifrování veškeré komunikace šifrovacími algoritmy. Dá se říci, že se na otevřené síti vytvoří mezi serverem a klientem tunel. VPN se dá využívat k různým účelům jako je propojení dvou vzdálených sítí např. dvě pobočky jedné společnosti v různých městech nebo připojení jednoho počítače do vzdálené sítě. V prvním zmíněném případě po vytvoření VPN se dvě propojené sítě chovají jako jeden celek. Ve druhém případě se po vytvoření VPN počítač zaměstnance stane součástí firemní sítě tj. vypadá to, jako by jeho počítač byl umístěn uvnitř počítačové sítě společnosti.

4.5.1. Připojení k poté serveru – navazování spojení

Při připojování k ptp serveru probíhá nejdříve úvodní komunikace mezi klientem (strojem, který se chce připojit) a serverem. Poté server ověřuje, zda se klient může připojit tím, že porovná zadané přihlašovací jméno a heslo se svým seznamem registrovaných uživatelů a jejich hesel. Jestliže přihlašovací jméno nebo heslo nesouhlasí, server spojení zruší. Jestliže vše souhlasí, tak server zašle klientovi některé potřebné údaje jako jsou rychlost navázaného spojení, IP adresa klienta a serveru a adresy DNS serverů v univerzitní síti. Poté je spojení aktivováno.

4.5.2. Změny na klientské stanici po navázání spojení s pptp serverem

Klient přijme údaje, které mu pošle server a nastaví si novou IP adresu, kterou mu přidělil server. S touto adresou bude probíhat po dobu spojení veškerá následující komunikace na internetu. Dále si přenastaví pravidla pro směrování veškerých dat, která bude klient odesílat. Po spojení budou veškerá data nejdříve směrována na pptp server a ten je pak následně pošle dál. Obráceně, veškerá data, která budou určena pro klientský stroj budou nejdříve posílána pptp serveru, který je následně pošle klientovi.

4.5.3. Omezení

Aby se omezilo šíření virů a červů, je na VPN serveru spuštěné filtrování síťových paketů. V současné době to znamená, že na počítači připojeného do VPN sítě nebudeme moci sdílet soubory a tiskárny pomocí "Sdílení souborů a tiskáren v sítích Windows".

4.5.4. Ukončení spojení s pptp serverem

Jestliže se rozhodnete ukončit spojení s pptp serverem, zašle se tento požadavek na pptp server. Ten spojení ukončí. Na klientském stroji se veškeré změny, které byly provedeny po navázání spojení uvedou do původního stavu.

4.5.5. Vytvoření VPN pomocí pptp v Linuxu

Funkce PPTP jako VPN spočívá v zapouzdření point-to-point protokolu (PPP) do IP a jeho protunelování skrz IP síť. Veškerou komunikaci, autentizaci a šifrování je obstaráváno téměř výhradně pomocí PPP. Z toho vyplývá, že je třeba mít, kromě vlastního pptp klienta, nainstalovaného ppp démona (pppd) pro komunikaci pomocí point-to-point protokolu. Problém se tedy rozpadl na dvě části. První je instalace ppp démona a druhá je instalace a konfigurace pptp klienta.

4.5.6. Ppp

Ppp démon (pppd) je standardně dodáván s každou linuxovou distribucí, takže by ho mělo stačit nainstalovat z příslušného balíčku.

4.5.7. Šifrování - protokol MPPE

Aby mohl být přenos přes VPN šifrován, musí být MPPE podporováno nejen ppp démonem, ale zároveň musí být podpora MPPE zakompilována přímo do jádra operačního systému. Bude tedy třeba zjistit, zda je šifrování podporováno v jádru i démonem. To zjistíme zadáním příkazu `# modprobe ppp-compress-18`, který nahraje MPPE modul.

4.5.8. Pptp-linux

Program pptp-linux umožňuje připojení k virtuální privátní síti založené na protokolu PPTP. Balík bývá součástí Linuxových distribucí, nebo je možné ho získat ze stránek projektu PPTP Client. Ideálním řešením je užití programu pptp-php-gtk, což je grafické rozhraní k balíku pptp-client. Umožňuje jednoduše nastavovat a měnit parametry pptp tunelu, zapínat a vypínat pptp tunel.

4.5.9. Směrování

Jestliže je navázáno spojení s pptp serverem, je třeba zajistit, aby byl síťový přenos směrován do tunelu a tím byla zpřístupněna síť právě přes VPN. Je několik způsobů, jak nastavit směrování. Například je možné to tunelu směrovat pouze data týkající se naší sítě a ostatní pravidla nechat v původním stavu. Nebo je možné směrovat veškerá data do tunelu. Druhá možnost je jednodušší a používanější.

4.5.10. Spuštění tunelu

Tunel je možné spustit několika způsoby:

- použitím pptp-php-gtk

- příkazem `pptp-command`
- přímo spuštěním `ppp` démona (příkaz `pon` jméno_tune lu)

5. HARDWARE SÍTĚ

5.1. Přístupový bod

WIFI karta ZCOM XI-626 + pigtail RSMA-N/m (kabel H155 - 20cm)

Frekvence	2,4 GHz
Standard sítě	IEEE 802.11b
Rychlost sítě	11, 5.5, 2, 1Mbps
Bezpečnost	128 WEP, 64 WEP
Použití	Infrastructure, Ad-Hoc, AP
Chipset	Prism 2,5
Operační systém	Windows, Linux

Zkratová bleskojistka pro 2,4 GHz

Provozní kmitočtové pásmo: 2,3 - 2,6 GHz

Činitel útlumu: lepší než -0,6 dB.

Činitel zpětného odrazu: lepší než - 20 dB

Nízkoútlumový kabel Belden H1000

Vhodný pro vedení signálu 2,4 GHz

Délka kabelu přesahuje 10 metrů

Přesný pájecí konektor pro kabel H1000

Přesný konektor, u kterého není třeba upravovat střed vodiče

WAVENAT - PAN17 - 17 dB/2.4GHz sektorová anténa

Zisk: 17 db

Vyzařovací úhel (H/V): 110/15 stupňů

Polarizace: horizontální

SWR: < 1,2

Impedance: 50 Ohm

Hmotnost: 0,6 kg

Rozměry: 31 x 10 x 2,5 cm

Zakončení: N female



Obr. 5. Sektorová anténa

Záložní zdroj APC UPC 500

Max. výstupní výkon (W):300

Provoz při plném zatížení (min.):5,7

Přepět'ová ochrana:Ano

Váha - včetně baterie (kg):6

Rozměry - šířka (mm):90 mm

Rozměry - výška (mm):170 mm

Rozměry - hloubka (mm):280 mm

5.2. Počítač

Procesor	Intel Celeron 850MHz
----------	----------------------

Paměť	256MB
-------	-------

Harddisk	120GB
----------	-------

Monitor	15"
---------	-----

II. PRAKTICKÁ ČÁST

1 REALIZACE BEZDRÁTOVÉ SÍTĚ PROVODOV

Po zadání bakalářské práce jsem nabídl její realizaci starostovi obce Provodov, panu Gallovi, který moji iniciativu přivítal a slíbil mi pomoc. Za první úkol jsem si stanovil výběr technologie připojení. Volil jsem ze tří možností – CDMA, GRPS EDGE a bezdrátové připojení. První dvě technologie jsem zamítl pro jejich nestabilitu a nespolehlivost. Volba padla na bezdrátové připojení, které se stále rozvíjí a poskytuje stabilní připojení k Internetu.

Další problém byla dostupnost dané technologie v lokalitě obce Provodov. Zkoumali jsme v mapě a posléze i v terénu trasu, kudy bychom mohli vést signál. Kvůli přírodním překážkám jsme postupně zavrhlí příjem signálu z kopce Komonec nad Provodovem, Baby nad Zlínem a z Doubrav. Signál se nakonec bude odebírat z přístupového bodu nad obcí Březůvky, kde má stožár T-Mobile. Odtud je přímá viditelnost na vodárnu nad obcí Provodov, kde bude tzv. accesspoint.

Na odběr konektivity jsem oslovil tři společnosti. ČRa (České radiokomunikace a.s.), Avonet s.r.o. a Kontaktuj s.r.o. Výše zmíněné společnosti nám poslali obchodní nabídku. Společnost Kontaktuj s.r.o. je na trhu nová a nemá ještě vybudovanou potřebnou infrastrukturu. ČRa má naopak nejlepší infrastrukturu a vlastní většinu stožárů v okolí. Nabízí i pronájmy stožárů pokud bychom chtěli vést signál ve vlastní režii. Ceny ČRa jsou zajímavé až od 1Mbit. Naše volba tedy padla na společnost Avonet, s.r.o., která má s bezdrátovou technologií velké zkušenosti. Signál bude vést z přístupového bodu na Březůvkách a vybuduje si i svůj vlastní přístupový bod.

Po zajištění výše zmíněných náležitostí jsem ve spolupráci se starostou obce Provodov panem Gallou oslovili občany vesnice. V první fázi reagovalo na naší výzvu přibližně 20 rodin, a těm jsem nabídl rychlost 256/128 kbps. V posledních dnech se zvýšil počet zájemců ještě o dalších deset, a tak jsem přistoupil na rychlost 512/256 kbps. Na schůzi občanů jsme, společně s garantem mé práce ing. Dulíkem, informovali občany o výhodách a nevýhodách bezdrátového připojení a sítě. Dále jsme projednali ceny, vstupní poplatky, založení Občanského sdružení a náklady na nákup antény a WIFI karty. V neposlední řadě jsme vybrali správce sítě a správce hardware.

ZÁVĚR

Bezdrátové připojení k Internetu je v současné době velmi slibně rozvíjející se technologie. Pokud obec nemá jinou možnost kvalitního připojení, je WIFI technologie dobrou volbou. Pro mnohé vesnice je také jedinou možnou volbou připojení.

V únoru jsme se společně s garantem mé bakalářské práce rozhodovali o obci, které nabídneme realizaci bezdrátového připojení a naše volba padla na Provodov. Následně před námi vyvstalo několik zásadních problémů. Řešil jsem návrh možné infrastruktury bezdrátové komunitní sítě, typ technologie připojení, poskytovatele konektivity, volbu vhodného operačního systému a služby poskytující tento systém. Dnes můžu říct, že všechny úkoly jsem splnil a spuštění bezdrátového internetu je otázkou podepsání běžných formalit a smluv. Bylo rozhodnuto, že poskytovatelem konektivity bude společnost Avonet s.r.o., operačním systémem bude Debian CZ a typem připojení bude WIFI technologie. Vše bude spravovat občanské sdružení UNART, respektive jeho pobočka v založená v Provodově.

Velkou výhodou bezdrátového připojení jsou levné pořizovací ceny jak pro klientské stanice, tak pro vybudování Accesspointu. Každý klient si musí na své náklady pořídit WIFI kartu a anténu v celkové hodnotě kolem 2.500,-Kč. U AP jsou náklady vyšší a to kolem 20.000,-Kč. Jsou v nich zahrnuty zřizovací náklady na vybudování AP, nákup počítače a antény. V obci Provodov přispěje obecní úřad 10.000,-Kč.

I proto se občané obce rozhodli založit Občanské sdružení a zrealizovat si vlastní síť. V současné době má o připojení zájem více než 25 domácností. Obec Provodov v budoucnosti počítá s využitím sítě jako komunikačního kanálu pro své občany. Má v plánu usnadnit občanům platby za odpad, místní poplatky a účty za kabelovou televizi. Výhledově chce zprovoznit webové rozhraní a informovat o kulturním, sportovním a společenském dění v obci. V neposlední řadě se zde objeví také zprávy ze zasedání obce, z jednání zastupitelstva a plány rozvoje obce. Individuální využití sítě je velmi pestré. Občané se na schůzi pořádané ve spolupráci se starostou obce zajímali o homebanking, nakupování na Internetu či levné telefonování. Mladší generace se ptala na možnost chatování, stahování a hraní her.

Budoucnost vybudované sítě je dále již pouze v režii obce a jejich představitelů a záleží jen na nich, kterým směrem se bude ubírat.

SEZNAM POUŽITÉ LITERATURY

- [1] Steven Graham, Steve Shah : Administrace systému Linux, Grada, 2003
ISBN 80 -247 – 0641 -5
- [2] Linux Advanced Routing & Traffic Kontrol – <http://lartc.org/>
- [3] Radius server - <http://www.gnu.org/software/radius/radius.html>
- [4] Poptop – open source VPN server - <http://www.poptop.org/>
- [5] Shorewall – firewall for IP Tables - <http://shorewall.net>
- [6] Squid proxy HTTP cache - <http://www.squid-cache.org/>
- [7] QoS - <http://www.pasnet.cz/sluzby/qos.html>
- [8] Squid proxy HTTP cache - <http://www.root.cz/>
- [9] StarOS - <http://star-os.cz/>
- [10] Mikrotik router - <http://www.mikrotik.com/>
- [11] Debian CZ - <http://www.debian.cz/>
- [12] CDMA - <http://www.eurotel.cz/>
- [13] CDMA - <http://www.maringold.cz/>
- [14] GRPS – EDGE - <http://www.eurotel.cz/>
- [15] Statistiky provozu na síti IPTraf - <http://iptraf.seul.org/>
- [16] Statistiky provozu na síti IPTraf - <http://www.abclinuxu.cz/clanky/show/87378>
- [17] Jabber – komunikační server - <http://www.jabber.org/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

Zkr.	Vysvětlení.
ADSL	Asymetric Digital Subscriber Line - ADSL je v současné době nejčastěji využívaný typ DSL. Vyznačuje se asymetrickým připojením.
GRPS-EDGE	GPRS - je technika pro paketově orientovaný přenos dat EDGE - je systém poskytující vysokorychlostní přenos dat
ICQ	<i>I seek you (hledám tě)</i> je komunikační systém
BOOTP	BOOTP (<i>Bootstrap Protocol</i>) - služba přidělování konfigurace TCP/IP klientům
HTB	Hierarchichal Token Bucket – algoritmus Fair User Policy
ACL	Access Control Protokol
HTTP	Hypertext Transfer Protokol - nejpoužívanější protokol na internetu, který slouží k zobrazování internetových stránek
QoS	Quality of Service - schopnost garantovat požadovanou kvalitu služeb
WIFI	Wireless Fidelity
FTP	File Transfer Protokol – protokol, umožňující přenos souborů
SMTP	Simple Mail Transfer Protocol - jednoduchý protokol pro odesílání a přenos pošty
VPN	Virtual Private Network – privátní virtuální síť umožňující bezpečné propojení vzdálených sítí zabezpečeným tunelem
DNS	Domain Name Server - překládá IP adresu
DHCP	DHCP (D ynamic H ost C onfiguration P rotokol) - služba přidělování konfigurace TCP/IP klientům
CDMA	C ode D ivision M ultiple A ccess - neboli vícenásobný přístup s dělením kódu
NAT	NAT je stará technologie, která byla zavedena vzhledem k nedostatku IP adres na Internetu

POP Post Office Protocol - slouží k načtení elektronické pošty do e-mailového klienta

SEZNAM OBRÁZKŮ

Obr. 2. Mikrotik router	11
Obr. 2. Program IPTraf	24
Obr. 3. Princip technologie FTP	31
Obr. 4. Komunikace přes ICQ	41
Obr. 5. Sektorová anténa	48

SEZNAM PŘÍLOH

P I První.

PŘÍLOHA P I: PRVNÍ.

Ukázkový konfigurační soubor pppd.conf

```
#####
#####
# $Id: pppd.conf,v 1.8 2004/04/28 11:36:07 quozl Exp $
#
# Sample Poptop configuration file /etc/pppd.conf
#
# Changes are effective when pppd is restarted.
#####
#####

# TAG: ppp
#       Path to the pppd program, default '/usr/sbin/pppd' on Linux
#
#ppp /usr/sbin/pppd

# TAG: option
#       Specifies the location of the PPP options file.
#       By default PPP looks in '/etc/ppp/options'
#
option /etc/ppp/options.pppd

# TAG: debug
#       Turns on (more) debugging to syslog
#
#debug

# TAG: stimeout
#       Specifies timeout (in seconds) on starting ctrl connection
#
# stimeout 10

# TAG: noipparam
#       Suppress the passing of the client's IP address to PPP, which is
#       done by default otherwise.
#
#noipparam

# TAG: logwtmp
#       Use wtmp(5) to record client connections and disconnections.
#
logwtmp

# TAG: bcrelay <if>
#       Turns on broadcast relay to clients from interface <if>
#
#bcrelay eth1

# TAG: localip
# TAG: remoteip
#       Specifies the local and remote IP address ranges.
#
#       Any addresses work as long as the local machine takes care of the
#       routing. But if you want to use MS-Windows networking, you
#       should
#       use IP addresses out of the LAN address space and use the
#       proxyarp
#       option in the pppd options file, or run bcrelay.
#
```

```
# You can specify single IP addresses separated by commas or you
can
# specify ranges, or both. For example:
#
# 192.168.0.234,192.168.0.245-249,192.168.0.254
#
# IMPORTANT RESTRICTIONS:
#
# 1. No spaces are permitted between commas or within addresses.
#
# 2. If you give more IP addresses than MAX_CONNECTIONS, it will
# start at the beginning of the list and go until it gets
# MAX_CONNECTIONS IPs. Others will be ignored.
#
# 3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238,
# you must type 234-238 if you mean this.
#
# 4. If you give a single localIP, that's ok - all local IPs will
# be set to the given one. You MUST still give at least one
remote
# IP for each simultaneous client.
#
# (Recommended)
#localip 192.168.0.1
#remoteip 192.168.0.234-238,192.168.0.245
# or
#localip 192.168.0.234-238,192.168.0.245
#remoteip 192.168.1.234-238,192.168.1.245
```

```
# This is a basic ProFTPD configuration file (rename it to
# 'proftpd.conf' for actual use. It establishes a single server
# and a single anonymous login. It assumes that you have a user/group
# "nobody" and "ftp" for normal operation and anon.
```

```
ServerName                "Jaros FTP"
ServerType                standalone
DefaultServer            on
ServerAdmin               j_jaros@ft.utb.cz # email na
administrátora serveru
```

```
# Port 21 is the standard FTP port.
Port                      21
```

```
# Umask 022 is a good standard umask to prevent new dirs and files
# from being group and world writable.
Umask                    022
```

```
# To prevent DoS attacks, set the maximum number of child processes
# to 30. If you need to allow more than 30 concurrent connections
# at once, simply increase this value. Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
```

```

# (such as xinetd).
MaxInstances                      30

# Set the user and group under which the server will run.
User                             nobody
Group                            nogroup

AccessGrantMsg "User %u logged in." # zpráva kterou napíše server po
správném zalogování

# To cause every FTP user to be "jailed" (chrooted) into their home
# directory, uncomment this line.
#DefaultRoot ~

# Normally, we want files to be overwriteable.
<Directory />
    AllowOverwrite                on
</Directory>

# A basic anonymous configuration, no upload directories.  If you do not
# want anonymous users, simply delete this entire <Anonymous> section.
<Anonymous ~ftp>
    User                          ftp
    Group                         ftp

    # We want clients to be able to login with "anonymous" as well as "ftp"
    UserAlias                     anonymous ftp

    # Limit the maximum number of anonymous logins
    MaxClients                    10

    # We want 'welcome.msg' displayed at login, and '.message' displayed
    # in each newly chdired directory.
    DisplayLogin                  welcome.msg
    DisplayFirstChdir             .message

    # Limit WRITE everywhere in the anonymous chroot
    <Limit WRITE>
        DenyAll
    </Limit>
</Anonymous>

```

Konfigurační soubor jabberu

```
<service type="icq" jid="icq.jabbernet.cz" name="ICQ">
  <ns>jabber:iq:gateway</ns>
  <ns>jabber:iq:register</ns>
  <ns>jabber:iq:search</ns>
</service>
```

Výše uvedená informace je určena pro klienta, my však musíme uvést ještě informace pro server, která by mohla vypadat následovně:

```
<service id="icq.jabbernet.cz">
  <icqtrans xmlns="jabber:config:icqtrans">
    <instructions>Please enter your ICQ number ...</instructions>
    <search>Search for ICQ users</search>
    ...
  </icqtrans>
  <load>
    <icqtrans>./icq-transport-0.9/icqtrans.so</icqtrans>
  </load>
</service>
```