VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

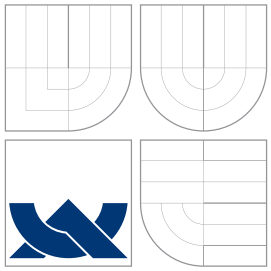
SPRÁVA HRANIČNÍHO SMĚROVAČE POSKYTOVATELE SLUŽEB INTERNETU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

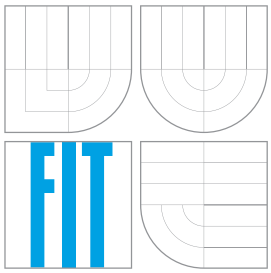
AUTOR PRÁCE
AUTHOR

JAROSLAV BARTOŇ

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SPRÁVA HRANIČNÍHO SMĚROVAČE POSKYTOVATELE SLUŽEB INTERNETU

BORDER ROUTER MANAGEMENT OF AN INTERNET SERVICE PROVIDER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JAROSLAV BARTOŇ
AUTHOR

VEDOUCÍ PRÁCE — Mgr. RYCHLÝ MAREK
SUPERVISOR

BRNO 2007

Abstrakt

Práce pojednává o možnostech správy a monitorování sítě skrz webový prohlížeč. Popisuje a implementuje metody monitorování sítě, účtování provozu, správy síťových prostředků a generování konfiguračních souborů pro síťová zařízení a služby.

Klíčová slova

směrovač, TCP/IP, účtování provozu, řízení šířky pásma, statistiky, dohledový systém, správa uživatelů

Abstract

This work deals with possibilities of monitoring and network management via web browser. It describes and implements methods of network monitoring, traffic accounting, managing of network resources and generating configuration files for network devices and services.

Keywords

router, TCP/IP, traffic accounting, bandwitch managenent, statistics, monitoring, user management

Citace

Jaroslav Bartoň: Správa hraničního směrovače poskytovatele služeb Internetu, bakalářská práce, Brno, FIT VUT v Brně, 2007

Správa hraničního směrovače poskytovatele služeb Internetu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Mgr. Marka Rychlého.

.....
Jaroslav Bartoň
12. května 2007

Poděkování

Chtěl bych poděkovat všem, kteří mi pomohli s cennými radami, nápady či připomínkami. Jmenovitě Mgr. Lubomír Hřivna a Lubor Vaca s informacemi o síti Haná Free Net a jejich představě na budoucí systém. Dále pak Mgr. Markovi Rychlému za konzultace a rady ke zpracování technické zprávy. Také bych chtěl poděkovat Mgr. Petře Navrátilové za pomoc při závěrečných úpravách.

Obsah

Kapitola 1

Úvod

V dnešní době existuje v české republice množství počítačových sítí, které mají jediný cíl – připojit uživatele k internetu. Velikost těchto sítí je velice různorodá. Od menších sítí v rodinném nebo panelovém domě, až po velké sítě poskytující přístup k internetu v několika lokalitách či v celé republice.

Také se liší způsobem fungování. Některé z těchto sítí jsou jen pro kamarády, úzký okruh zájemců a jsou nekomerční. Jiné fungují jako občanské sdružení, mají své členy a způsobem fungování připomínají firmy. Pak tu jsou také firemní sítě, které poskytují připojení k internetu.

Jednotlivé sítě se liší i v přístupu k problémům. Ty nejmenší sítě, či sítě patřící občanským sdružením problémy řeší v nejkratší možné době, snaží se vnímat stížnosti jednotlivých připojených uživatelů. Naproti tomu jsou některé firemní sítě, kde je hlavní zisk, spokojenost uživatelů je až některém z dalších míst.

Jako vždy jsou i výjimky, kde výše uvedené neplatí.

1.1 Síť Haná Free Net

Síť Haná Free Net (dále jen HFN) je občanské sdružení založené s cílem poskytovat levné a kvalitní připojení do internetu. V začátcích byla síť HFN jen v obci Příkazy – Olomoucký kraj. Postupem času se rozrostla a dnes již pokrývá 14 lokalit a pokrytí dalších se plánuje. O síť pro 260 uživatelů se stará rada čítající 7 členů a několik techniků a správců sítě. Všichni na síti pracují ve svém volném čase bez nároku na odměnu.

1.2 Cíle projektu

Při práci na bakalářské práci jsem vycházel z požadavků bezdrátové sítě HFN na budoucí systém pro správu hraničního směrovače. I přes spolupráci s HFN bylo cílem navrhnout systém tak, aby byl použitelný i pro další sítě.

Síť HFN hledala nástroj, který by usnadnil správu uživatelů v síti, umožnil ukládat kontaktní údaje jako telefon, e-mail či kontakt na službu rychlých zpráv jako je například Jabber.

Ke každému členovi sítě HFN je veden seznam přidělených privátních a veřejných IP adres. Pokud má uživatel přidělenu veřejnou IP adresu, je na hraničním routeru proveden překlad adres 1:1 mezi vnitřní a veřejnou IP adresou.

V síti HFN dále fungovalo účtování provozu. Informace o přenesených datech jsou zveřejněny na intranetu HFN. V případě, že členové občanského sdružení nerespektují stanovy a dokument o provozu na síti a nadměrně využívají společnou přípojku, je možné jim omezit šířku pásma. Omezení šířky pásma se jinak nevyužívá.

Další požadavek byl na zjednodušení správy nebo nahrazení stávajícího dohledového systému, který již nevyhovoval velikosti a dynamičnosti růstu sítě.

Kapitola 2

Teorie

Nově vytvořený informační systém je složen z několika modulů. Hlavní část aplikace je napsaná v programovacím jazyce Java a využívá web server Jetty. Je využit šablonovací systém Freemarker, který umožnil oddělit aplikační logiku od generování XHTML kódu posílaného webovému prohlížeči – je využit model Model–View–Controller (MVC).

Účtování provozu a dohledový systém využívají webový server Apache s podporou jazyka PHP a přepisu adres.

Systém používá databázi PostgreSQL a RRD databázi. PostgreSQL je hlavní databáze k ukládání dat z informačního systému, provozních dat dohledového systému a účtování provozu. Dohledový systém a účtování provozu dále používají RRD databáze, které jsou využity při generování grafů.

2.1 Java

Java je objektově orientovaný programovací jazyk vyvinutý firmou Sun Microsystems [6]. Mezi největší výhody jazyka Java patří snadná přenositelnost mezi platformami bez nutnosti program znovu překompilovat – existuje běhové prostředí pro různé platformy jako je například GNU/Linux, Solaris, MS Windows. Díky tomu, že Java byla uvolněna pod GNU/GPL [9] licencí budou další platformy přibývat.

Existuje běhové prostředí Java Runtime Environment (JRE) umožňující prgramy napsané v jazyce Java spouštět a Java Development Kit (JDK) pro vývoj a spouštění programů.

2.2 Jetty

Jetty je aplikační a webový server naprogramovaný v jazyce Java. Umožňuje poskytovat statický a dynamický obsah. Dynamický obsah může být generován servlety či stránkami Java Server Pages. Jetty pro svůj běh vyžaduje běhové prostředí JRE nebo vývojové prostředí JDK.

2.2.1 Servlet

Servlety jsou součástí Java 2 Enterprise Edition (J2EE). Jsou to programové komponenty běžící na straně serveru a napsané v jazyce Java.

Servlety načítá a spouští aplikační server. Přestože je servlet nevizuální komponenta a nemá tedy žádné uživatelské rozhraní, může generovat uživatelské rozhraní v podobě

HTML formulářů. Nejčastější využití servletů je ve spojení s komunikačním protokolem HTTP – servlety implementují princip požadavek/odpověď.

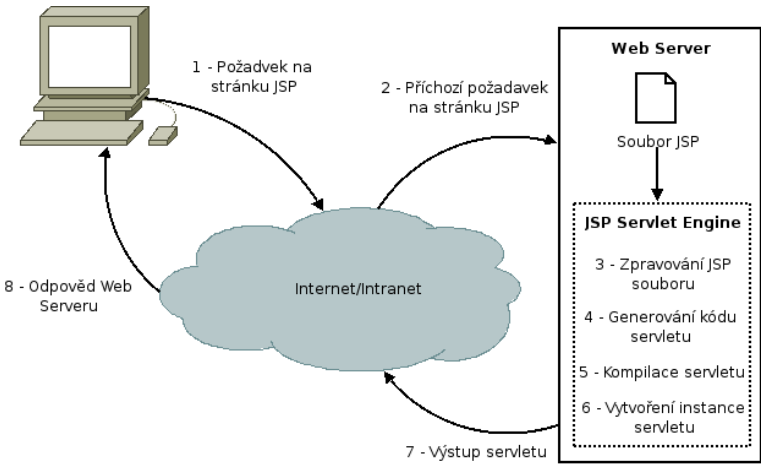
2.2.2 Java Server Pages

Java Server Pages (JSP) je technologie používaná k vývoji aplikací běžících na straně serveru. JSP jsou vlastně HTML stránky do kterých jsou vloženy speciální tagy obsahující kód v jazyce Java. Tento kód vytváří dynamické části stránek. Použití JSP je vhodné pouze pokud statický kód tvoří většinu stránky. Jinak je doporučeno použít technologii servletů.

Při prvním spuštění JSP stránky dojde k její kontrole a následnému přeložení do speciálního servletu. Servlet je pak přeložen do mezikódu (`class` soubor). Proto první zpracování JSP stránky tak trvá déle. Při dalších přístupech se používá již zkompilovaný `class` soubor.

Následně jsou u vygenerovaného servletu zavolány metody `init()` a `service()` pro inicializaci a zpracování (obrázek 2.1). Výstupem je (X)HTML kód, který můžeme poslat přes síť klientovi.

JSP stránky se mohou za běhu serveru měnit a po každé změně je třeba provést nové přeložení do servletu a následně mezikódu. To se provede při prvním přístupu ke změněné JSP stránce.



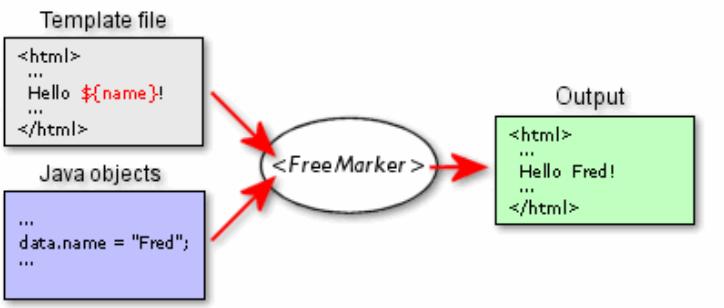
Obrázek 2.1: První zpracování JSP stránky

2.3 Freemarker

Projekt je založen na architektuře MVC (Model–View–Controller). Díky MVC bylo možné oddělit aplikační logiku od prezentační formy.

Popis jednotlivých částí MVC:

- **Controller** – prostředník mezi **Model** a **View**, provádí zpracování požadavků
- **Model** – data se kterými pracuje **View**
- **View** – transformuje **Model** do prezentační formy



Obrázek 2.2: Generování výstupu z datového modelu a šablony

Freemarker je šablonovací systém, nástroj, který slouží ke generování textového výstupu založeného na šablonách [5]. V architektuře MVC je Freemarker na pozici View – od aplikace dostane datový model a za pomoci popisu uloženého v šabloně jej převede na výstupní soubor (viz obrázek 2.2). Popis výstupního souboru se může skládat ze statického, neměnného obsahu a značek které umožní ovlivňovat obsah souboru.

2.4 Apache a PHP

Apache je standardní webový server dostupný pro operační systémy GNU/Linux, *BSD, Solaris, MS Windows a jiné [11]. Je to nejrozšířenější webový server na internetu – používá se na více než polovině serverů.

Apache v základní instalaci bez dodatečných modulů umí poskytovat pouze statický obsah, případně obsah generovaný CGI (Common Gateway Interface) skripty. Ve spolupráci se serverem Apache se nejčastěji používá skriptovací jazyk PHP. PHP může být spouštěno jako modul serveru `mod_php`, případně za pomoci `cgi/fastcgi`. `Cgi` varianta je pomalejší, zato poskytuje vyšší bezpečnost.

Skriptovací programovací jazyk PHP byl určený především pro programování dynamických webových stránek. Lze jej začleňovat přímo do struktury (X)HTML kódu, což je výhodné při psaní menších aplikací. PHP lze použít i pro tvorbu konzolových či deskopových aplikací.

2.5 RRD databáze

RRD je zkratka pro Round Robin Database – databáze plněná údaji, které jdou v časové posloupnosti. Velikost RRD databáze zůstává v čase konstantní, na záznamy se může aplikovat agregační funkce, kdy se několik záznamů nahradí jedním (což vede k redukci uchovávaných dat) a po čase se nejstarší záznamy nahradí nově vloženými. Agregační funkce a maximální stáří záznamů je dáno při založení databáze.

Pro práci s RRD databází je využito RRDtool [4], s konektorem pro jazyk Java – RRDJtool [3]. RRDJtool byl vytvořen pro spolupráci s RRDtool ve verzi 1.1 a proto je nutno při použití s verzí 1.2 provést opravu v nativní části (Java Native Interface, JNI) zdrojových kódu RRDJtool, viz přiložená oprava (strana 40).

2.6 Relační databáze

Všechna data ze systému, informace o množství přenesených dat a stavu sítě jsou ukládány v databázi. Při uchovávání a zpracování dat jsou kladeny požadavky hlavně na perzistenci, dostupnost a bezpečnost dat. Relační databáze tyto požadavky splňují. Pro manipulaci s daty v relační databázi se obvykle používá strukturovaný dotazovací jazyk SQL (Structured Query Language).

Existuje velké množství databází. Mezi komerční databázové systémy patří například *Oracle* či *Microsoft SQL Server*. Tyto databázové systémy poskytují širokou funkcionalitu a pro firemní zákazníky důležitou podporu. Jsou určeny pro složitější, náročné aplikace.

Na druhé straně jsou databázové systémy s otevřenými zdrojovými kódy jako je *Firebird*, *MySQL* či *PostgreSQL*, které svými vlastnostmi dostačují požadavkům aplikace.

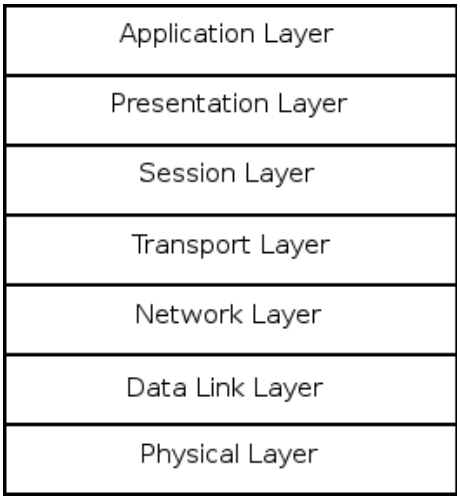
Pozor na PostgreSQL 8.1.7, která obsahuje chybu, která způsobovala nemožnost přidávat a editovat některé záznamy – chybná práce s kontrolním omezením u VARCHAR sloupce.

2.7 Síťové protokoly

Pro přenos dat po síti byl zaveden tzv. vrstvový model. Jednotlivé vrstvy poskytují určitou úroveň abstrakce a škálovatelnost. Vrstvové protokoly nespecifikují žádný konkrétní protokol, ale rozhraní mezi protokoly jednotlivých úrovní a úkoly protokolů na dané vrstvě.

2.7.1 Referenční model ISO/OSI

Referenční model ISO/OSI byl přijat jako standard mezinárodní organizací pro standardizaci ISO (čerpáno z [10]). Definuje 7 vrstev (obrázek 2.3).



Obrázek 2.3: Referenční model ISO/OSI

Fyzická vrstva

Definuje napěťové úrovně signálu, použité konektory apod. Například RS-232.

Linková vrstva

Poskytuje spojení mezi dvěmi sousedními systémy. Seřazuje přicházející rámce, stará se o nastavení parametrů přenosu linky, hlásí neopravitelné chyby. Vytváří fyzické rámce a přidává jim fyzickou adresu. Linkové protokoly Ethernet, FDDI, ATM.

Síťová vrstva

Síťová vrstva se stará o směrování v síti a síťové adresování. Poskytuje propojení mezi systémy, které spolu přímou nesousedí. Umožňuje překlenout rozdílné vlastnosti použitých technologií v přenosových sítích. Síťové protokoly IP, IPX, Appletalk.

Transportní vrstva

Zajišťuje transparentní, spolehlivý přenos dat s požadovanou kvalitou. Vyrovnává různou vlastnosti a kvalitu přenosových sítí. Provádí převod transportních adres na síťové, ale nestará se o směrování.

Relační vrstva

Smyslem vrstvy je organizovat a synchronizovat dialog mezi spolupracujícími relačními vrstvami obou systémů a řídit výměnu dat mezi nimi. Umožňuje vytvoření a ukončení relačního spojení, synchronizaci a obnovení spojení, oznamovaní výjimečných stavů.

Prezentační vrstva

Funkcí vrstvy je transformovat data do tvaru, které používají aplikace.

Formát dat (datové struktury) se může lišit na obou komunikujících systémech, navíc dochází k transformaci pro účel přenosu dat nižšími vrstvami. Mezi funkce patří například převod kódů a abeced, modifikace grafického uspořádání, přizpůsobení pořadí bajtů a podobně. Vrstva se zabývá jen strukturou dat, ale ne jejich významem.

Aplikační vrstva

Účelem vrstvy je poskytnout aplikacím přístup ke komunikačnímu systému a umožnit tak jejich spolupráci.

2.7.2 Model TCP/IP

Většina počítačových sítí (včetně Internetu) používá jako komunikační protokol TCP/IP (čerpáno z [10]). Dnes rozšířenou verzi 4 (IPv4) začíná postupně nahrazovat verze 6 (IPv6). Největší rozdíl mezi IPv4 a IPv6 je v délce adresy a tím množství koncových zařízení, která lze adresovat – u IPv4 postupně dochází k vyčerpání adresního prostoru. Referenční model TCP/IP vychází z ISO/OSI modelu a je vidět na obrázku 2.4 společně s ukázkou zapouzdření UDP datagramu.

Popis jednotlivých vrstev TCP/IP modelu:

1. **Vrstva síťového rozhraní** plní funkce fyzické vrstvy, příkladem může být Ethernet, Token Ring, WiFi.

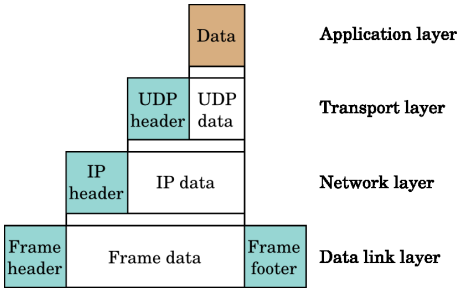
2. **Síťová vrstva** odpovídá za směrování datagramů mezi uzly sítě. Mezi protokoly síťové vrstvy patří například IPv4, IPv6, ARP, RARP, ICMP, IGMP.

3. **Transportní vrstva** poskytuje spojované (protokol TCP) a nespojované (protokol UDP) transportní služby.

4. **Aplikační vrstva** – aplikace, které využívají přenos dat po síti ke konkrétním službám pro uživatele. Například DNS, SSH, HTTP.

2.7.3 Protokol UDP

Protokol UDP je **nespojovaným** protokolem transportní vrstvy. Nezaručuje tedy doručení dat ani jejich doručení ve správném pořadí. Díky své jednoduchosti je nenáročný na zpracování (obrázek 2.4). Vhodný pro služby kde nevadí, že se některé pakety ztratí – například síťové hry.



Obrázek 2.4: Zapouzdření UDP datagramu v IP paketu

2.7.4 Protokol TCP

Protokol TCP je **spojovaným** protokolem transportní vrstvy. Zaručuje doručení veškerých přenášených dat a to ve správném pořadí. Správnost dat zabezpečuje kontrolním součtem. Využívá se všude tam, kde je třeba zaručit přenesení všech informací – HTTP, SSH a jiné.

2.7.5 DHCP

Dynamic Host Configuration Protocol (DHCP) je protokol sloužící k automatické konfiguraci sítě u připojených zařízení (čerpáno z [10]). Protokol DHCP pracuje na portu UDP/67.

DHCP umožňuje nastavit IP adresu, síťovou masku, výchozí bránu, adresu jmenných serverů, název počítače a název domény.

DHCP může IP adresy přiřazovat podle MAC adresy síťové karty, pak počítač vždy dostane stejnou IP adresu, či z vyhrazeného rozsahu. Pokud je IP adresa z vyhrazeného rozsahu, může být při vícenásobném přidělení stejná, ale není to pravidlem.

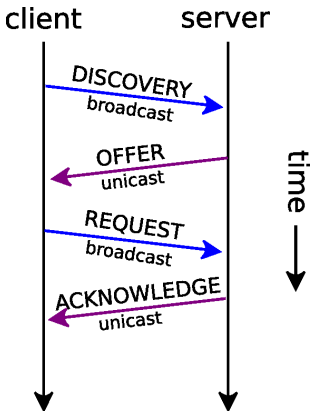
Průběh automatické konfigurace:

1. Klient odešle DHCPDISCOVERY

2. Každý DHCP server který přijal žádost odešle DHCPOFFER

3. Z nabízených IP adres si klient vybere a požádá o ni – DHCPREQUEST

4. Pokud DHCP server souhlasí s požadavkem klienta, odešle DHCPACK a na určitou dobu klientovi přidělí IP adresu
5. Před ukončením platnosti výpůjčky musí klient provést obnovení
6. Při ukončení práce na síti může klient provést uvolnění IP adresy – DHCPRELEASE



Obrázek 2.5: Typický průběh komunikace mezi DHCP klientem a serverem

2.7.6 HTTP

Na přesnos hypertextových dokumentů ve formátu HTML se používá Hyper Text Transfer Protokol. Protokol HTTP standardně využívá port TCP/80. Protokol byl HTTP byl rozšířen o podporu přenosu jakýchkoliv souborů – obrázky, zvuky a další.

HTTPS přidává protokolu HTTP podporu zabezpečené komunikace. Data nejsou přenášena v prostém textu, ale jsou šifrována za pomoci SSL nebo TLS. Protokol HTTPS poskytuje zvýšenou bezpečnost před odposloucháváním či podvržením dat.

2.8 URL

URL (Uniform Resource Locator) je řetězec znaků s definovanou strukturou a slouží k jednoznačné specifikaci umístění zdrojů informací (dokument, služba) na Internetu.

URL definuje doménové jméno, port, umístění zdroje na serveru a komunikační protokol kterým je možné ke zdroji přistoupit.

Příklad URL:

`http://blog.djaara.net/wordpress/`

Kde `http` označuje komunikační protokol, `blog.djaara.net` je doménové jméno serveru, `/wordpress/` definuje umístění zdroje na serveru. V tomto případě není specifikován port, protože protokol `http` standardně využívá port TCP/80. Pokud by byl použit nestandardní port, je možné ho v URL specifikovat pomocí dvojtečky za doménovým jménem a následně číslem portu.

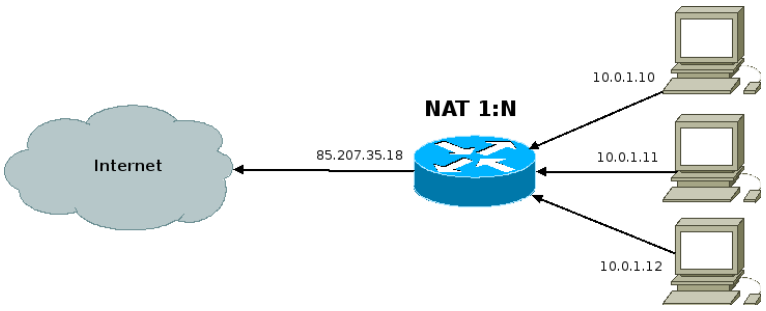
2.9 Překlad adres

Protože u protokolu IPv4 který je v současnosti nejpoužívanějším Internetovým protokolem postupně dochází k vyčerpání adresního prostoru, bylo nutné vymyslet způsob, jak tento trend zpomalit – přechod na novější IPv6 je velice nákladný.

Překlad adres je znám pod názvem NAT (Network Address Translation), je funkce síťového routeru, který při průchodu paketu mění IP adresy v hlavičce (obrázek 2.6).

2.9.1 NAT 1:N

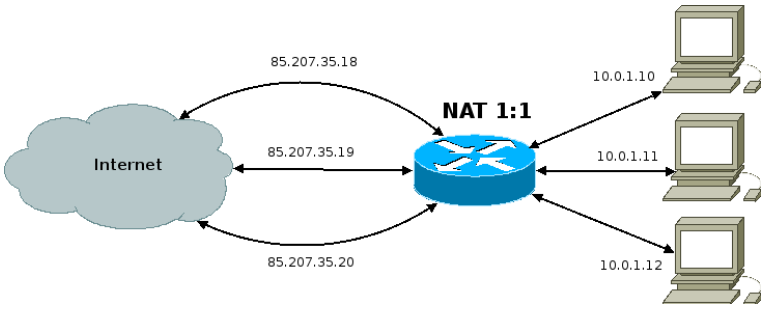
Mezi nejpoužívanější konfigurace NAT patří **NAT 1:N**, kdy jednu veřejnou IP adresu využívá několik zařízení s neveřejnou adresou. Pro vnější síť pak mají všechny tyto zařízení stejnou IP adresu (obrázek 2.6). Tento způsob překladu adres neumožňuje plnohodnotné využívání sítě – není možné se spojit s počítačem ve vnitřní síti. Komunikaci vždy musí zahájit počítač z vnitřní sítě.



Obrázek 2.6: Příklad překladu adres 1:N

2.9.2 NAT 1:1

Dalším možným nastavením je **NAT 1:1**, kdy je právě jedna interní adresa přeložena na adresu veřejnou a naopak (obrázek 2.7). To umožňuje počítači s adresou z neveřejného rozsahu plnohodnotný přístup na internet, u paketů přicházejících na veřejnou IP adresu se cílová adresa přeloží na neveřejnou IP adresu a pošle se dále do vnitřní sítě.



Obrázek 2.7: Příklad překladu adres 1:1

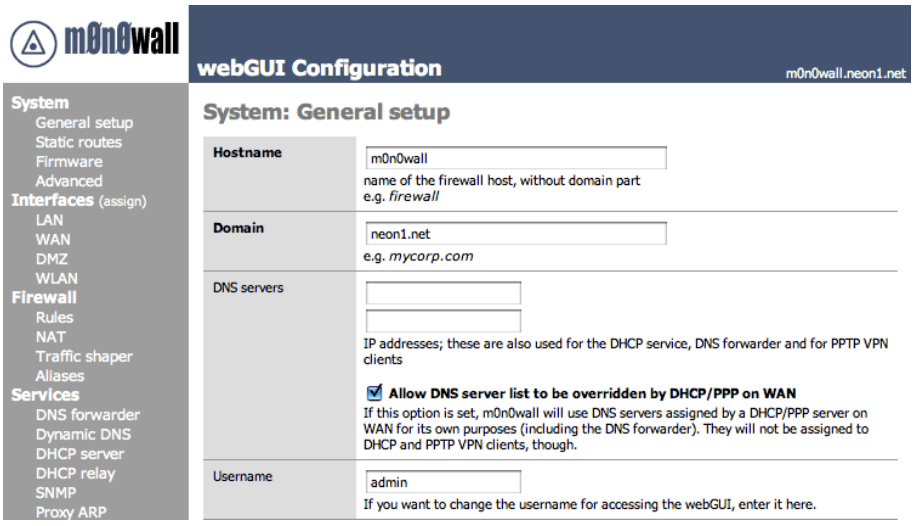
Kapitola 3

Nástroje pro správu a sledování stavu směrovače

Jedním z cílů práce bylo prozkoumat open-source nástroje pro správu a sledování stavu směrovače. Od kompletních řešení po nástroje určené pro konkrétní účely.

3.1 m0n0wall

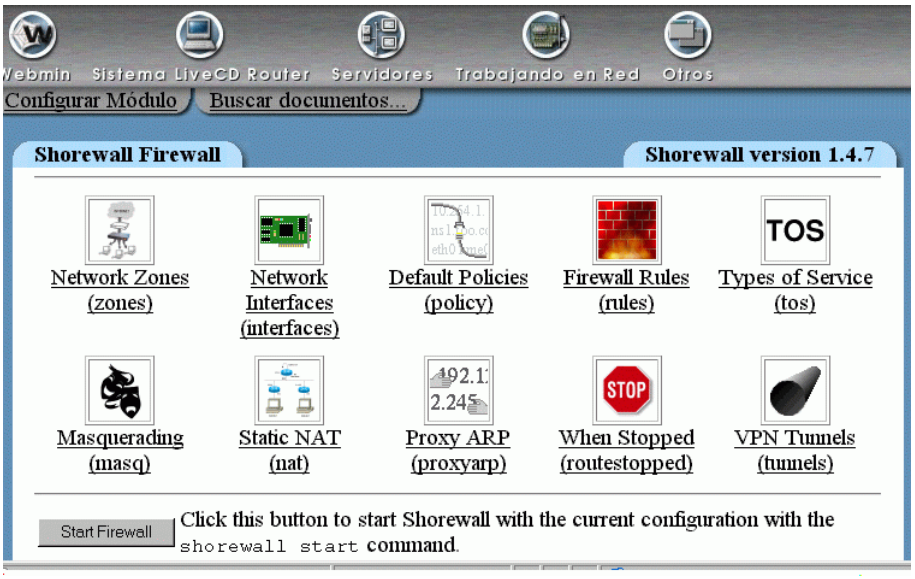
M0n0wall je speciální distribuce založená na FreeBSD určená k provozu z CD-ROM či flash paměti. Je dostupná na adrese http://m0n0.ch/wall/. Může sloužit jako router, bridge, firewall, VPN či DHCP server. Existuje verze pro architekturu *x86* (PC), také pro verze *routerboardy* či *wrapy* (vestavěná zařízení).



Obrázek 3.1: Ukázka systému m0n0wall

3.2 Linux LiveCD Router

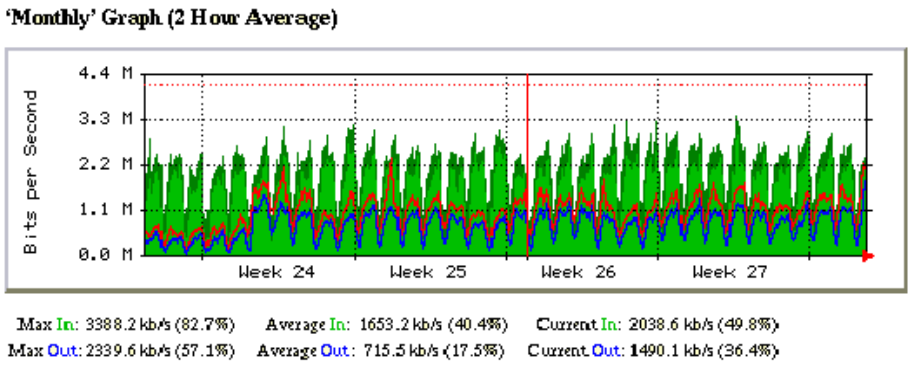
Linux LiveCD Router umožňuje sdílet a filtrovat internetové připojení. Obsahuje Shorewall pro nastavení firewallu a překladu adres. Umožňuje řídit šířku pásma, obsahuje DHCP server, funguje jako mezipaměť pro službu DNS, podporuje SNMP a MRTG. Lze získat z adresy http://www.wifi.com.ar/english/cdrouter/.



Obrázek 3.2: Ukázka systému Linux LiveCD Router

3.3 MRTG

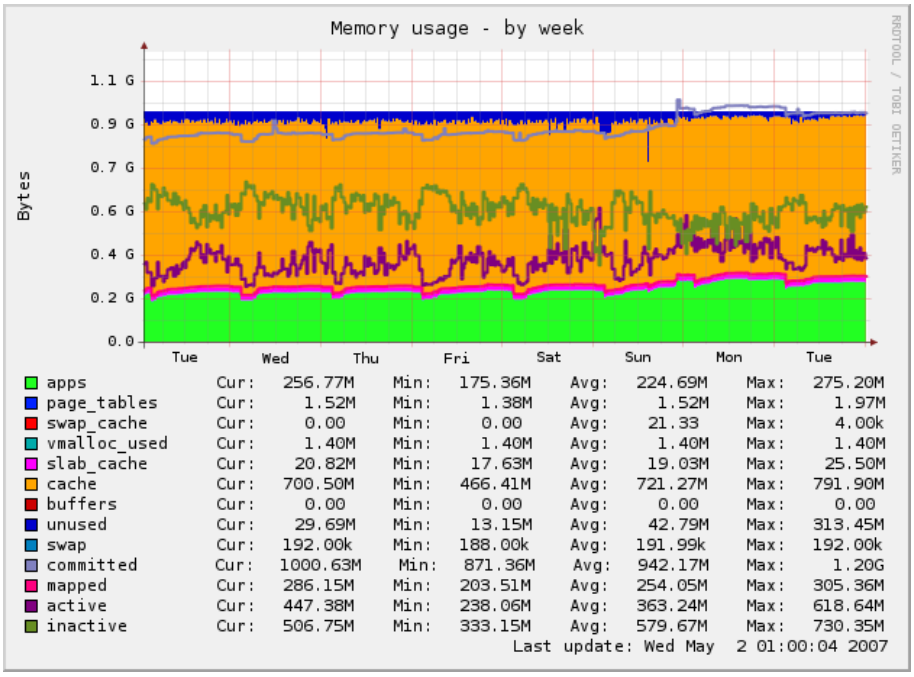
Multi Router Traffic Grapher (http://oss.oetiker.ch/mrtg/) je multiplatformní nástroj, který je schopen vytvářet velmi komplexní grafy. Primárně je určený k sledování datových toků na jednotlivých síťových zařízeních na serveru. Dále umožňuje sledovat různé ukazatele na serveru. Za pomoci SNMP může sbírat data z inteligentních směrovačů a přepínačů.



Obrázek 3.3: Ukázka systému MRTG

3.4 Munin

Systém Munin slouží k monitorování stavu serverů. Stejně jako MRTG umí vytvářet velmi komplexní grafy s informacemi o stavu jednotlivých strojů. Data sbírá za pomocí agentů, které je třeba nainstalovat na každý monitorovaný stroj. Kromě základních ukazatelů jako je využití paměti či procesoru lze monitorovat počet SQL dotazů pro MySQL databázi, množství přečtených a zapsaných dat na disk a další. Domovská stránka projektu Munin je http://munin.projects.linpro.no/.



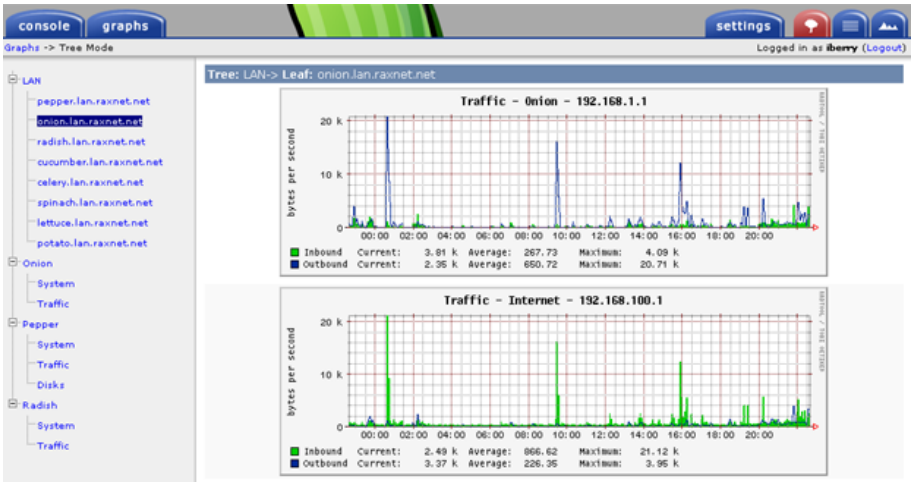
Obrázek 3.4: Ukázka systému Munin - využití paměti

3.5 Cacti

Cacti je komplexní dohledový systém určený k monitorování sítí. K ukládání dat používá MySQL databázi a RRDtool. Sleduje průtok dat na jednotlivých rozhraních, umožňuje správu zařízení, umožňuje podrobné nastavení vykreslování grafů. Cacti naleznete na adrese http://cacti.net/.

3.6 Nagios

Nagios je monitorovací program, který sleduje stav sítě, služeb a jednotlivých hostů. Obsahuje sadu pluginů. Každý plugin je v podstatě jednoduchý program nebo skript, který umí prověřit dostupnost určitého stroje či služby. Rozlišuje mezi službami a stroji a pokud je stroj nedostupný pomocí příkazu ping, nezkouší už testovat dostupnost služeb. Také umí definovat závislosti mezi službami. Více informací naleznete na http://nagios.org/.



Obrázek 3.5: Ukázka systému Cacti

3.7 Nmap

Nmap je multiplatformní nástroj pro aktivní i pasivní skenování sítě. Umí zjistit jaké porty jsou na daném počítači otevřené které jsou filtrované. Také se může pokusit zjistit verzi aplikace která naslouchá na otevřeném portu. U standardních portů píše i název služby pro kterou se tento port používá. Lze nastavit několik strategií testování cílového počítače.

Kapitola 4

Implementace

4.1 Dohledový systém

Při zkoumání dostupných nástrojů pro monitorování sítě a dostupnosti služeb jsem objevil několik velice zajímavých nástrojů. Jednoduché nástroje byly velice efektivní, ale nebyly schopny pracovat v různorodých sítích a většinou byly určeny pro konkrétní síťové technologie (převážně Cisco). Komplexnější nástroje pak měly velice zdlouhavou a nepřehlednou konfiguraci, používaly pro sběr dat protokol SNMP a nebyly moc efektivní. Při práci na dohledovém systému jsem čerpal z [1].

4.1.1 Cíl práce

Na základě těchto zjištění (kapitola 4.1) bylo rozhodnuto vytvořit vlastní specializovaný počítačový program umožňující dohled nad rozsáhlými sítěmi.

Program bude kompletně řízen přes webové rozhraní. Přístup do webového rozhraní bude chráněn jménem a heslem. Při vzniku problému (nedostupnost uzlu, služby) bude informovat technika pomocí sítě Jabber, případně SMS či e-mailem.

Mezi sledované parametry patří:

- doba odezvy (roud trip time, RTT)
- dostupnost routerů a routerů
- dostupnost klíčových služeb
- ztrátovost linky

4.1.2 Filozofie dohledového systému

Snaha přehledně zobrazit veškeré relevantní informace ztroskotala na jejich množství. Z tohoto důvodu se hledaly cesty, jak zobrazit informace co nejpřehledněji.

Díky přehlednému zobrazení dat pomocí grafů se podařilo vytvořit uživatelsky přívětivý systém s jednoduchým ovládáním.

4.1.3 Metody sběru dat

Pro ověřování funkčnosti a kvality linek jsou použity ICMP pakety `echo request` a `echo reply`. Server, který provádí monitorování sítě generuje paket `echo request` a posílá ho testovanému cíli. Cílové zařízeni po přijetí `echo request` odpoví `echo reply`. Doba mezi odesláním požadavku a přijetím odpovědi se nazývá Round Trip Time (RTT). Pro toto testování je využit program `fping`.

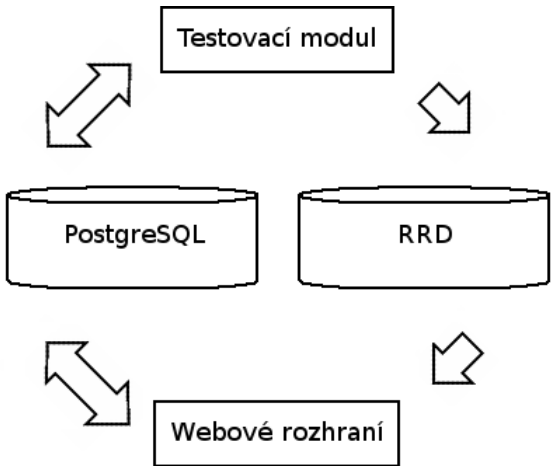
Ověření dostupnosti služby je prováděno programem `nmap`. Program `nmap` se pokouší připojit na zadaný port a vypíše informace o tom zda je služba dostupná.

Dohledový systém dále může sledovat vytížení systému, obsazení paměti, využití disku, odstup signál/šum u WiFi karty a také jak dlouho již daný počítač běží – uptime. Aby bylo možné sledovat i tyto informace, je třeba na sledovaný systém nahrát agenta (strana 39) – jednoduchý skript, který naslouchá na TCP portu 1234. Spouštění je zajištěno pomocí „superserveru“ `inetd` či `xinetd`.

4.1.4 Ukládání dat

Informace o délce jednotlivých výpadků, času posledního testování a poznámce k jednotlivým uzlům či službám jsou uloženy v SQL databázi. V té jsou ukládány i informace o tom, koho v případě problémů informovat.

Informace o RTT, dostupnosti služeb a informace stavu systému jsou ukládány do RRD databáze za pomoci RRDJtool konektoru k RRDtool (strana 7).

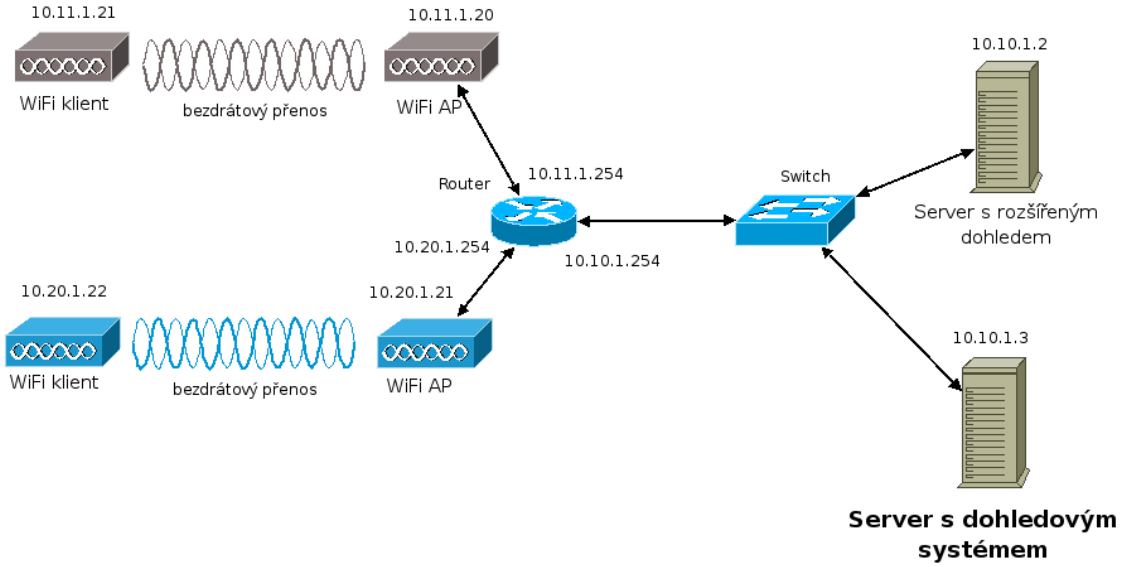


Obrázek 4.1: Blokové schéma dohledového systému

4.1.5 Metodika testování dostupnosti routerů

Veškeré testování probíhá ze serveru s nainstalovaným dohledovým systémem.

U každého uzlu se sbírají základní parametry (doba odezvy a ztrátovost linky). Doba RTT (více u Metody sběru dat – strana 18) se udává standardně v milisekundách případně v mikrosekundách. Poměr mezi počtem přijatých a počtem odeslaných paketů se označuje ztrátovost (Packet Loss, PL). Ztrátovost se udává v %. Limity ztrátovosti pro WiFi linku a ne-WiFi linku lze nastavit v souboru `Bodos.properties`.



Obrázek 4.2: Práce dohledového systému

U rozšířených uzlů jsou navíc přes síť pomocí protokolu TCP na portu 1234 zjišťované následující informace: doba běhu (uptime), vytížení procesoru, využití paměti a odkládacího prostoru (swap), odstup signál/šum. Tyto rozšířené informace jsou zobrazeny v samostatném grafu.

Doba běhu zařízení je doba od posledního naběhnutí operačního systému v daném zařízení. V grafu je znázorněna barvou pozadí. Aktuální doba běhu je zobrazena v legendě grafu.

Vytížení procesoru je definováno jako poměr derivace rozdílu doby běhu a nevyužité doby procesoru (idle time) ku derivaci doby běhu a vyjadřuje se v %. Využití paměti a odkládacího prostoru je definováno jako poměr využitého prostoru ku celkově dostupnému prostoru. Odstup signál/šum je počítán z úrovně signálu a šumu.

Pokud ztrátovost linky překročí hranici nastavenou pro daný typ spoje, je tento spoj označen za nedostupný (obrázek 4.2). Při nedostupnosti delší než 10 minut systém odešle servisním technikům informaci o nedostupnosti uzlu (uzel 10.11.1.20). Následují li za nedostupným uzlem další uzly (uzel 10.11.1.21), je technik informován pouze o výpadku kořenového uzlu. Při obnovení dostupnosti technikovi dorazí oznámení o všech nově dostupných uzlech.

4.2 Účtování provozu

Účtování provozu a požadavky na něj jsou většinou specifické pro každou síť. V Haná Free Net se informace o přenesených datech pro jednotlivé IP adresy berou z routeru s nainstalovaným systémem StarOS.

4.2.1 Cíl práce

Protože StarOS není volně dostupný systém bylo třeba zajistit aby bylo možné sbírat informace o přenesených datech i z jiných zdrojů. Takovým zdrojem může být jakýkoliv počítač, který dodává data na určeném portu v požadovaném formátu.

Výhodou StarOS-u je, že kdykoliv se na síti objeví nová, dosud neznámá IP adresa, automaticky o ní začne poskytovat informace.

4.2.2 Formát dat poskytovaný StarOS-em

StarOS poskytuje informace o přenesených datech na portu TCP/800. Po stažení informací jsou počítadla vynulována. Pokud by bylo třeba znovu získat tyto informace, můžeme o ně StarOS požádat na portu TCP/801.

Výstupem je textový soubor, kdy na každém řádku je informace o množství bytů a paketů přenesených mezi zdrojovou a cílovou adresou. Jako oddělovač jednotlivých polí slouží znak ␣ (mezera).

1	2	3	4	
10.11.1.1	10.10.1.3	272556	246	* *

Tabulka 4.1: Ukázka jednoho řádku z výstupu StarOS routeru

Popis jednotlivých položek:

1. zrojová IP adresa
2. cílová IP adresa
3. přenesených Bytů
4. přenesených paketů

Jako zdroj informací o přenesených datech lze použít jakýkoliv systém, který lze nakonfigurovat tak aby poskytoval data v požadovaném formátu. To lze udělat například pomoci skriptu spouštěného „superserverem“ `xinetd` a programu `iptables` na systému GNU/Linux.

4.2.3 Metody sběru dat

Systém počítání přenesených dat se připojí na port TCP/800 a získá tak informace o přenesených datech. Poté projde všechny řádky a spočítá kolik která IP adresa přijala a odeslala bytů a paketů.

V nastavení je možno určit zda se mají brát v úvahu i data přenesená mezi počítači ve vnitřní síti. Sběr dat se provádí každých 5 minut na základě časovače.

4.2.4 Ukládání dat

Informace o přenesených datech se ukládají do PostgreSQL databáze a jednou za hodinu se provede hodinový součet pro každou IP adresu a opět se uloží do PostgreSQL databáze. Pětiminutové záznamy jsou smazány, nejsou již dále potřeba.

Systém počítání přenesených dat může data ukládat i do RRD databáze, která je pro každou IP adresu samostatná. Tyto RRD databáze jsou použity ke generování grafů.

Blokové schéma je velice podobné blokovému schématu sběru dat u dohledového systému (obrázek 4.1) s tím rozdílem, že místo `testovacího modulu` by byl `sběr dat`.

4.3 Informační systém

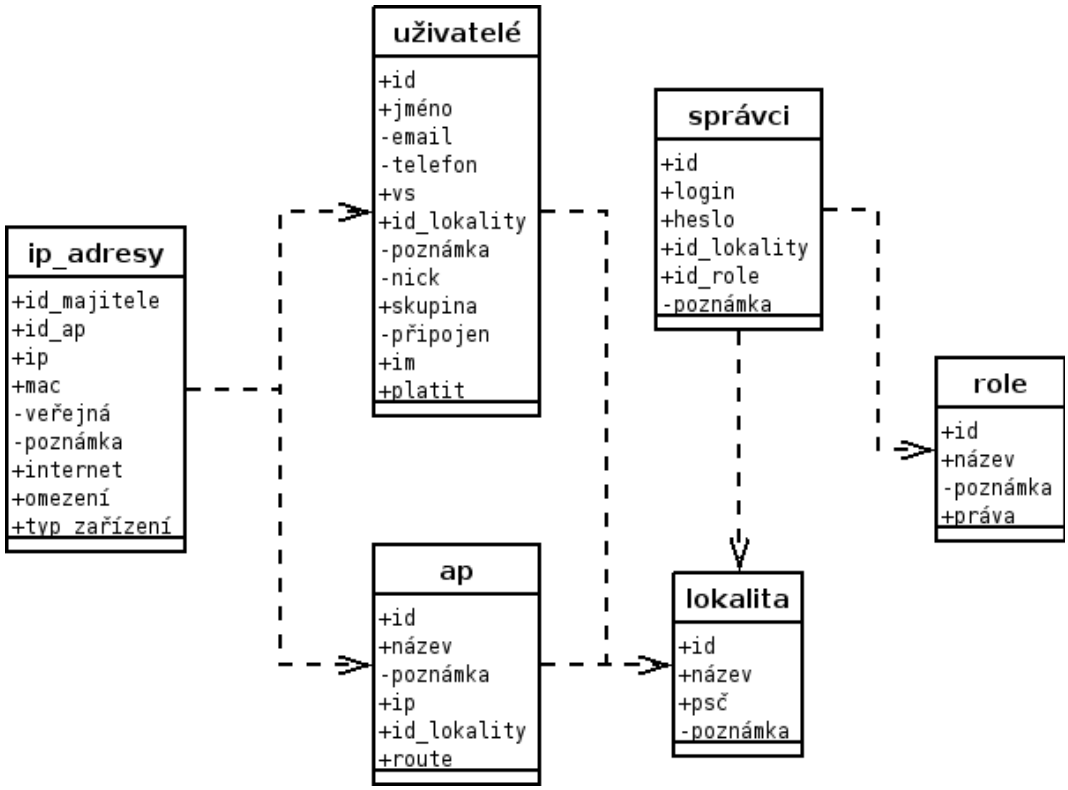
Informační systém slouží ke správě lokalit, přístupových bodů, uživatelů a přidělených privátních a veřejných IP adres.

4.3.1 Cíl práce

Bylo třeba navrhnout a implementovat komplexní informační systém, aby se maximálně zjednodušila evidence. Tento systém musí umožnit správu uživatelů a nastavování sítě a některých služeb.

4.3.2 Spravované údaje

Informační systém uchovává informace z několika modulů. Povinné atributy začínají znakem +, nepovinné znakem − (obrázek 4.3).



Obrázek 4.3: Scheěma uchovávaných informací

Ke všem uchovávaným údajům, které může změnit některý z administrátorů je vedeno kdy a kým byl údaj přidán a také kdy a kým byl editován. U editace je uložena pouze poslední změna. Historie editací není uchovávana.

4.3.3 Kontrola zadávaných údajů

U všech údajů, kde je požadavek na jejich správnost se provádí jejich kontrola. Tato kontrola probíhá na 2 úrovních. První úroveň je v aplikaci, kdy kontroluji vstupy porovnávím

s regulárním výrazem popisujícím možné hodnoty v daném poli.

Pomocí regulárního výrazu je možné zkontrolovat MAC adresy, IP adresy, telefonní čísla, e-mailové adresy, poštovní směrovací čísla, čísla a řetězce.

Druhá úroveň kontroly je na úrovni databázového stroje s kontrolními a integritními omezeními. Kontrolní omezení opět kontrolují správnost řetězců pomocí regulárních výrazů – pro případ, že člověk pracuje přímo s databází. Integritní omezení pak zajišťují, že položky na kterých jsou závislé další záznamy nepůjdou smazat.

4.3.4 Ochrana hesel

Veškerá komunikace s informačním systémem i dohledovým systémem probíhá za použití protokolu HTTPS. Díky tomu je obtížné odchytit komunikaci mezi serverem a klientem. Na straně serveru jsou hesla uložena v databázi v šifrované podobě (MD5). Kdyby se podařilo odcizit obsah databáze, stále bude problém získat hesla v otevřené podobě. Při přihlašování uživatele je zadané heslo zašifrováno a porovnáno se šifrovanou podobou uloženou v databázi.

4.3.5 Kontrola oprávnění

Ne všichni uživatelé informačního systému mají přístup ke všem informacím v systému uloženým. Aby se uživatel dostal jen k údajům, kam má povolen přístup, systém při každém požadavku zkontroluje, zda uživatelovo oprávnění dostačuje k tomu aby mohl provést požadovanou akci. Uživatel bez dostatečného oprávnění je přesměrován na osobní stránku.

4.3.6 Generování konfiguračních souborů

Při zadávání, editaci a mazání údajů majících souvislost se sítí se vždy přegenerují konfigurační soubory, které mohla tato změna zasáhnout. Přehled těchto souborů najdete v příloze B (strana 37).

Pokud je konfigurační soubor určený pro některé zařízení v síti, jméno souboru pak obsahuje IP adresu daného zařízení. To kam se mají vygenerované soubory ukládat se nastavuje v konfiguračním souboru `configgenerator.properties`.

Generování konfiguračních souborů má na starosti systém Freemarker.

Kapitola 5

Další vývoj

5.1 Distribuovaný dohledový systém

V současné době není dohledový systém distribuovaný a tak je zatížen několika problémy. Při testování větší sítě je linka od serveru, který provádí monitorování výrazně vytížena a zátěž stoupá lineárně s počtem monitorovaných uzlů.

Pokud dojde k výpadku spoje, nemusí být možná komunikace mezi serverem s dohledovým systémem a sítí za vypadeným spojem. Přesto zbytek sítě může nadále fungovat a to včetně internetu, pokud je na jeho straně internetová přípojka. Dohledový systém přesto bude upozorňovat na problémy a nebude schopen sledovat kvalitu sítě.

Jednou z množností jak snížit zátěž sítě při monitorování většího počtu uzlů a také mít přehled o síti i při lokálním výpadku je distribuovaný dohledový systém. Na každém uzlu by běžel jednoduchý program, který by monitoroval několik dalších uzlů a výsledky by posílal hlavnímu uzlu dohledového systému. V případě, že nebude možné poslat data dohledovému systému, tak je bude skladovat lokálně až do chvíle, kdy bude možné hlavní uzel informovat o naměřených hodnotách.

5.2 Účtování provozu

Účtování provozu by šlo rozšířit o zobrazení statistik pro jednoho uživatele – namísto statistik pro jednotlivé IP adresy by bylo možné sledovat využití sítě všemi počítači uživatele.

Také by šlo upravit systém účtování provozu pro sběr dat z více zdrojů. Například pokud je síť připojena k více poskykotavelům internetu a připojení je přivedeno v rozdílných částech sítě.

5.3 Rozdrobení uživatelských práv

V současné době jsou v informačním systému 3 úrovně oprávnění. Nastavení oprávnění je docela hrubé a nelze tak přesně určit k čemu bude mít uživatel přístup. Proto by bylo vhodné uživatelská oprávnění rozdělit na menší části a umožnit jejich samostatné přidělování z informačního systému.

5.4 Distribuce konfiguračních souborů

Vygenerované konfigurační soubory jsou pojmenované podle služby, které slouží a obsahují i IP adresu zařízení pro které byly vytvořeny. To by šlo využít pro jejich automatickou distribuci a restart služeb (pokud to vyžadují).

5.5 Doplnění správy IP adres o správu DNS serveru

Správa IP adres by mohla být doplněna o správu záznamů u DNS serveru. Při použití DNS serveru jako je například *MyDNS*, kdy se data berou přímo z databázového serveru, by pak mohlo být každé zařízení dostupné i pod doménovým jménem a snížily by se tak nároky na techniky a jejich paměť na IP adresy.

5.6 Sjednocení uživatelského rozhraní

Každá z těchto částí má samostatné webové rozhraní a přístup k nim je na jiné URL. A to přesto, že výkonná část se spouští společně.

5.7 Rozhraní pro účetní

Do informačního systému by se přidala role účetní a systém by se upravil tak, aby odpovídal potřebám účetní. Umožňoval evidenci o platbách, import dat z banky a další.

Kapitola 6

Závěr

Vytvořil jsem kompletní informační systém pro správu prostředků sítě. Upravil jsem stávající systém účtování provozu – zvláště pak část sběru dat, kterou se podařilo více než 10x zrychlit a integrovat do informačního systému. Původní systém vycházel ze StarOS IP accounting [2].

Využil jsem již dříve vytvořený dohledový systém a zaintegroval sběr dat do informačního systému. Tento systém překonal původní systém jednoduchou správou a množstvím sledovaných hodnot.

Data jsou ukládána do PostgreSQL a RRD databáze. Pro práci s RRD databází jsem použil RRDJtool. U RRDJtool bylo třeba opravit JNI rozhraní pro spolupráci s RRDtool verze 1.2 – viz oprava (strana 40).

Možnosti dalšího vývoje jsou popsány v kapitole 5.

Literatura

[1] Dittrich, P.: Síťový dohledový systém. Pardubice. 2004. 23 s. Střední průmyslová škola elektrotechnická. Maturitní zkouška.

[2] Gacík, M.: IP Accounting 1.3. [online], [cit. 2007-04-25].
URL ⟨http://www.star-os.sk/skf/doc/mod_act.php⟩

[3] Markovic, S.: RRDJtool. [online], [cit. 2007-04-25].
URL ⟨http://www.dnseurope.net/opensource/rrdjtool/⟩

[4] Oetiker, T.: RRDTool - Logging and Graphing. [online], [cit. 2007-04-18].
URL ⟨http://oss.oetiker.ch/rrdtool/⟩

[5] WWW stránky: Freemarker. [online], [cit. 2007-04-25].
URL ⟨http://freemarker.sourceforge.net/⟩

[6] WWW stránky: Java Technology. [online], [cit. 2007-04-25].
URL ⟨http://java.sun.com/⟩

[7] WWW stránky: Jetty Webserver. [online], [cit. 2007-04-25].
URL ⟨http://jetty.mortbay.org/⟩

[8] WWW stránky: PostgreSQL: The world's most advanced open source databas. [online], [cit. 2007-04-25].
URL ⟨http://www.postgresql.org/⟩

[9] WWW stránky: Sun Opens Java. [online], [cit. 2007-04-25].
URL ⟨http://www.sun.com/2006-1113/feature/story.jsp⟩

[10] WWW stránky: Wikipedia, the free encyklopedia. [online], [cit. 2007-05-07].
URL ⟨http://en.wikipedia.org/⟩

[11] WWW stránky: The Apache HTTP Server Project. [online], [cit. 2007-04-25].
URL ⟨http://httpd.apache.org/⟩

[12] WWW stránky: DHCP server. [online], [cit. 2007-04-18].
URL ⟨http://www.isc.org/index.pl?/sw/dhcp/⟩

Seznam příloh

Příloha A

Uživatelský manuál

A.1 Popis adresářové struktury

```
|-- bin
|   |-- WEB-INF                adresář s webovou aplikací
|   |   |-- freemaker          šablony systému Freemarker
|   |   `-- web.xml            konfigurace webové aplikace
|   |-- ie7                    skripty pro kompatibilitu s MSIE
|   `-- styles-dark.css        vzhled webové aplikace
|-- build.xml                  build skript pro Apache Ant
|-- hfnis.sql                  schéma databáze
|-- htdocs
|   |-- bodos                  dohledový systém
|   `-- data                   informace o přenesených datech
|-- jetty                      aplikační/web server jetty
|-- libs                       použité knihovny
|-- native                     JNI část RRDJtool
`-- src                        zdrojové kódy v jazyce Java
```

A.2 Požadavky

- vývojové prostředí jazyka Java – JDK6 nebo vyšší
- PostgreSQL databázi
- webový server s podporou jazyka PHP verze 4.3 a vyšší a podporou SSL
- RRDtool verze nejméně 1.1
- RRDJtool (pro spolupráci s RRDtool verze 1.2 přiložena oprava)
- Apache Ant

Program byl vyvíjen za použití JDK6, PostgreSQL ve verzi 8.1.8, jako webový server byl použit Apache 2.2.3 s PHP ve verzi 5.2.0. Dále RRDtool ve verzi 1.2.19.

A.3 Instalace

Informační systém

Instalaci zahájíte vytvořením databázového uživatele a databáze, kterou bude systém používat (viz dokumentace [8]). Do databáze naimportujte obsah souboru `hfnis.sql`.

Dále zkompilujte zdrojové kódy HFNis. To se provede příkazem `ant`. Přeložené zdrojové kódy budou uloženy v adresáři `bin/`.

Pokud chcete aplikaci nainstalovat na jiné místo než je současné umístění, zkopírujte adresáře `bin/`, `jetty/` a `libs/` na zvolené místo.

V adresáři `bin/WEB-INF/` upravíte soubor `web.xml` ve kterém nastavíte správnou cestu k adresáři se šablonami pro Freemarker.

Dále v adresáři `bin/WEB-INF/classes/` upravíte nastavení jednotlivých modulů informačního systému.

Nyní můžete přejít do adresáře `jetty/ssl/`, kde se vytvoří `keystore` s certifikátem pro šifrovanou komunikaci (viz dokumentace [7]).

V adresáři `jetty/etc/` upravíte soubor `hfn.xml`, kde nastavíme heslo k `keystore` a umístění webové aplikace.

Když je vše nastaveno, můžete v adresáři `jetty/` spustit skript `run.sh`. Pokud je vše nastaveno správně, naběhne informační systém, připojí se jabber bot a nastartuje se dohledový systém a systém počítání přenesených dat. Na obrazovku se budou vypisovat provozní informace podle nastavení *log4j*.

Aby bylo možné program spustit na pozadí a odhlásit se, lze použít příkazu `nohup`.

Webové rozhraní informačního systému je dostupné na portu 8080 – ⟨`https://hfnis.example.net:8080/`⟩. Jméno `hfnis.example.net` nahraďte jménem serveru, na kterém jste systém nainstalovali. Pro přihlášení do systému použijte uživatelské jméno *admin* s heslem *admin*.

Webové rozhraní dohledového systému

Webový server musí podporovat SSL spojení (protokol HTTPS) dostupné na standardním portu 443.

Soubory z `htdocs/bodos/` nakopírujte do adresáře s webovým serverem (nejčastěji `/var/www/bodos/`). Následně cílovém adresáři upravte soubor `src/config.php` ve kterém nastavte připojení k databázovému serveru, cesty k RRD databázím a také cestu k nainstalovanému programu `rrdtool`.

Dohledový systém pak bude dostupný na adrese ⟨`https://hfnis.example.net/bodos/`⟩ (závisí na použitém webovém serveru).

Webové rozhraní účtování provozu

Soubory z `htdocs/data/` nakopírujte do adresáře s webovým serverem (nejčastěji `/var/www/data/`). Následně v cílovém adresáři upravte soubor `config.php`.

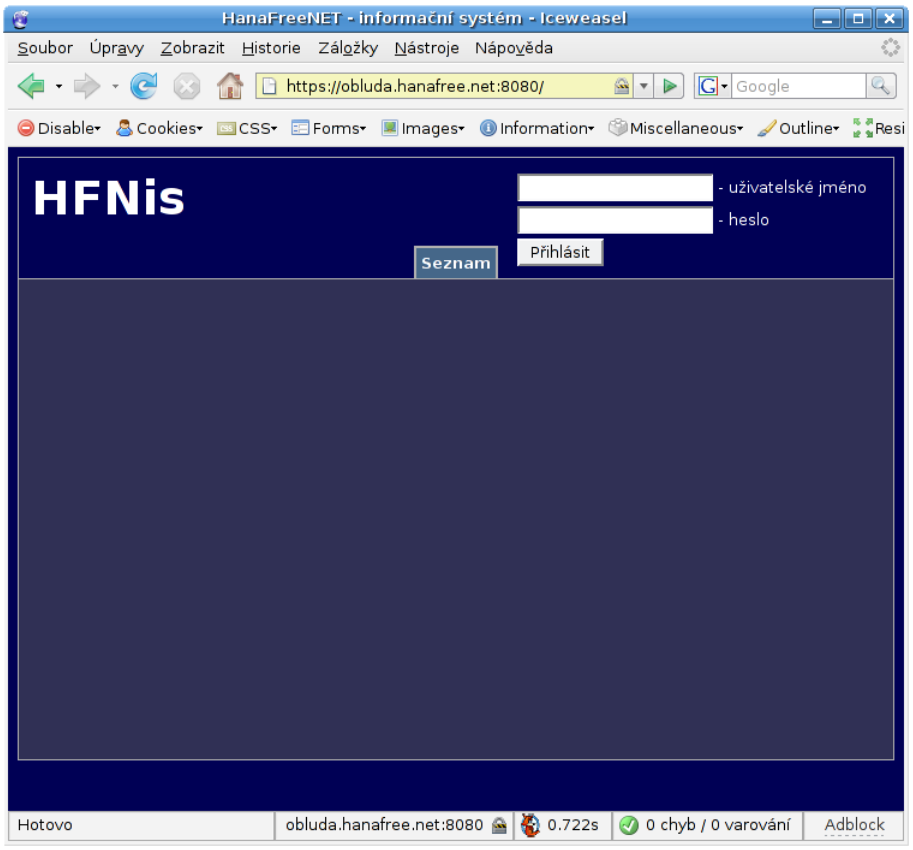
Je třeba nastavit `base_path` podle URL na které bude webové rozhraní dostupné. Pokud budete chtít používat „pěkné“ URL, je třeba mít na webovém serveru povolen přepis adres (`mod_rewrite` u serveru Apache [11]). Dále nastavte připojení k databázovému serveru, cesty k RRD databázím a také cestu k nainstalovanému programu `rrdtool`. Můžete zde i změnit hodnoty přenesených dat pro jednotlivé stupně podbarvení.

Informace o přenesených datech budou dostupné na adrese ⟨`http://hfnis.example.net/data/`⟩ (závisí na použitém webovém serveru).

A.4 Uživatelské rozhraní

A.4.1 Informační systém

Vstup do uživatelského rozhraní informačního systému je chráněn jménem a heslem. Po instalaci je v systému veden uživatel *admin* s heslem *admin*. Uživatelské jméno lze kdykoliv změnit v databázi po instalaci systému nebo ještě před jeho instalací úpravou instalačního skriptu databázového schématu.



Obrázek A.1: Přihlašovací stránka do informačního systému

Po přihlášení (obrázek A.1) se uživatel dostane na stránku, kde vidí přehled posledních 10 přihlášení a také zde může změnit své heslo. Uživatel s právy připojovacícho týmu nebo administrátora zde také vidí přehled uživatelů, kteří čekají na připojení k síti (obrázek A.2).

Obrázek A.2: Informace o uživateli, který čeká na připojení do sítě

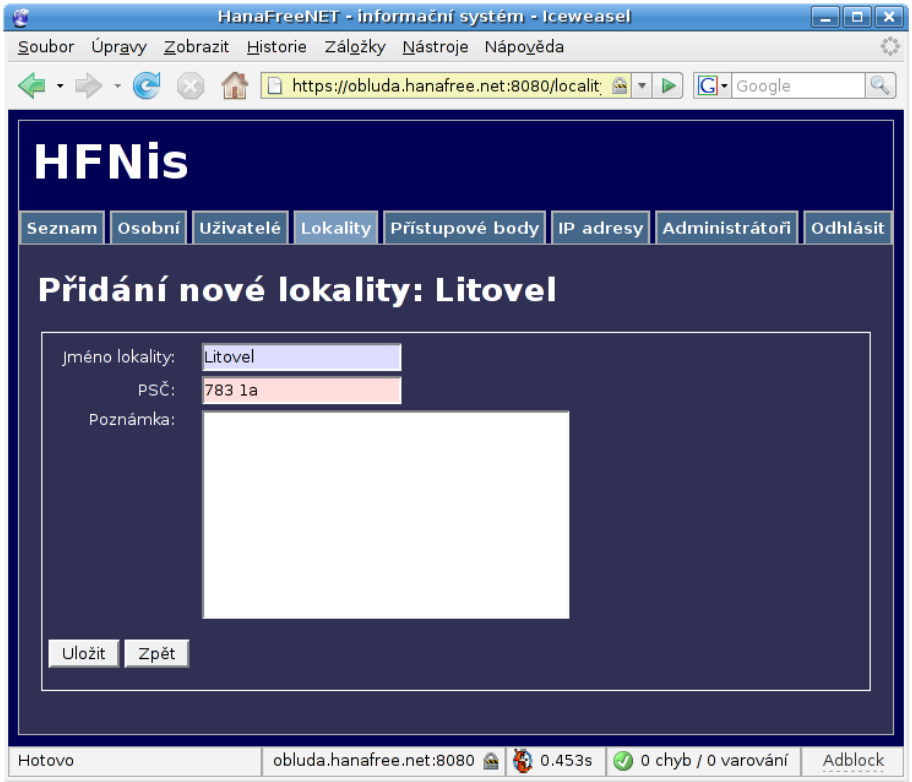
V horním nabídkovém pruhu se lze přepnout na další karty (může se lišít podle úrovně oprávnění uživatele):

- **Seznam** – seznam všech uživatelů sítě, veřejně přístupný
- **Uživatelé** – správa uživatelů sítě
- **Lokality** – správa pokrytých lokalit
- **Přístupové body** – správa přístupových bodů
- **IP adresy** – správa přidělených IP adres
- **Administrátoři** – správa uživatelů informačního systému

Na každé kartě jsou v tabulce zobrazeny hodnoty odpovídající jménu karty. Je možno přidávat, mazat či editovat veškeré informace (v závislosti na oprávnění uživatele). Přidání se provede stisknutím tlačítka *Přidej*. Zobrazí se formulář, do kterého se vyplní informace a stisknutím tlačítka *Uložit* se provede kontrola a uložení. U povinných položek je vstupní pole podbarveno světle modře.

Pokud byla některá položka chybně zadána a neprošla kontrolou na správnost, neprovede se uložení dat a uživatel uvidí opět formulář s předvyplněnými údaji a červeným podbarvením zvýrazněným vstupním polem ve kterém došlo k chybě (obrázek A.3).

Při mazání informací z informačního systému je nutné jejich mazání potvrdit. Po potvrzení operace je informace označena za neaktivní a nadále ponechána v databázi. Je to kvůli případnému dohledávání informací. Také nelze smazat objekt na kterém závisí další objekty jako je například *lokalita* nebo *přístupový bod*.



Obrázek A.3: Ukázka zvýraznění povinného údaje a chybného vstupu

A.4.2 Dohledový systém

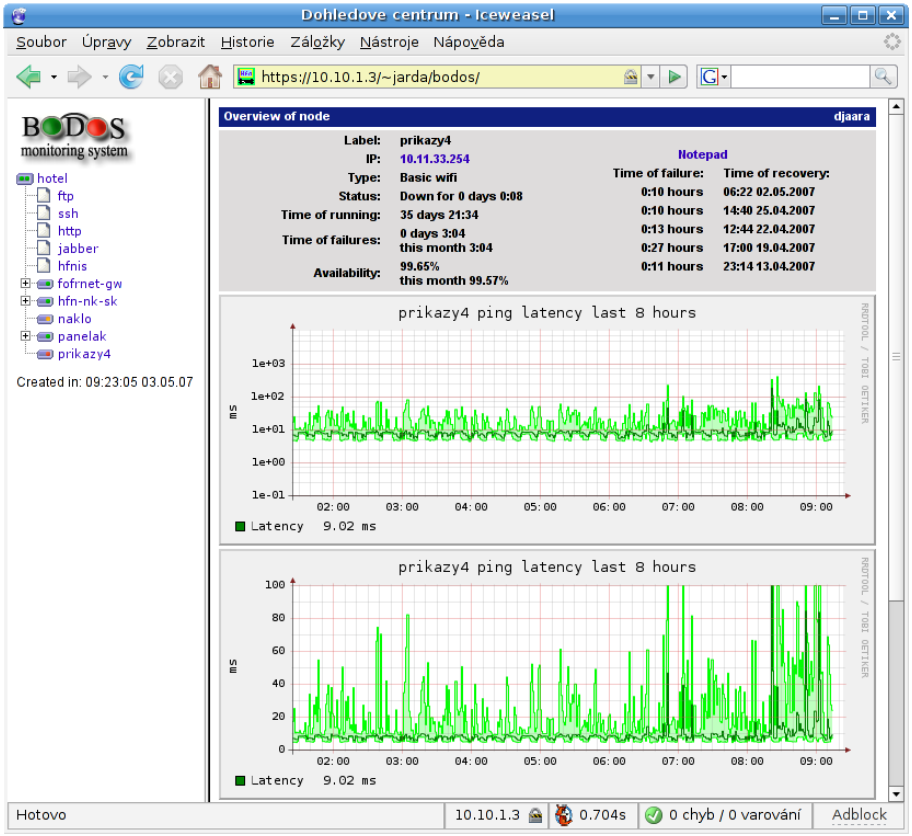
Dohledový systém vyžaduje zadání uživatelského jména a hesla při přístupu na jeho stránky. Uživatelské jméno a heslo je stejné jako jméno a heslo do informačního systému. Přihlášení je ale omezeno na uživatele úrovně „administrátor“ a „připojovací tým“. Uživateli s právy „dědinář“ není přístup povolen.

Uživatelské rozhraní je rozděleno do 2 částí (obrázek A.4). V levé části je interaktivní mapa topologie sítě, která je koncipována jako rozbalitelný strom. Tento rám se každé 2 minuty obnoví. Ve stromu jsou automaticky rozbalené části, které obsahují nedostupné uzly. Pokud se vedle ikony uzlu nachází bílý čtvereček se znakem +, je větev nacházející se za tímto uzlem skryta. Naopak pokud bílý čtvereček obsahuje znak −, jsou bezprostředně následující uzly zobrazeny. Při kliknutí na tento čtvereček se zobrazí nebo skryje příslušná část stromu a změní se znak z + na − a obráceně.

Pokud se vedle ikony uzlu nenachází bílý čtvereček, nenavazuje na uzel žádná další větev. Uzly u nichž bylo testování pozastaveno jsou označeny oranžovou tečkou a uzly, které jsou nedostupné jsou označeny tečkou červenou. Uzly, které jsou dostupné a probíhá u nich testování jsou označeny zeleně.

Při kliknutí na název uzlu dojde k zobrazení informací o něm v pravé části obrazovky. Je zde titulkový pruh s menu a jméno aktuálně přihlášeného uživatele. Pod titulkovým pruhem jsou informace o uzlu. Dále následuje několik grafů s naměřenými hodnotami.

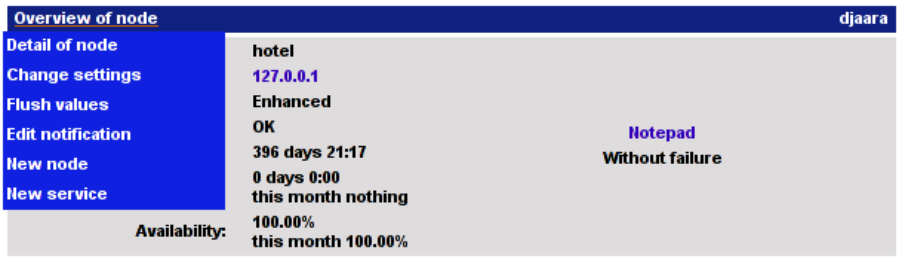
Menu se zobrazí po kliknutí na levou části titulkového pruhu s nápisem „Overview of node“ (obrázek A.5). Pomocí tohoto menu lze provádět následující úpravy a nastavení:



Obrázek A.4: Dohledový systém – informace o uzlu

- **Detail of node** – zobrazí detailní informace o uzlu
- **Change settings** – umožní měnit nastavení uzlu
- **Flush values** – smaže veškerá naměřená data
- **Move node** – přesun uzlu na jiné místo sítě
- **Delete node** – smaže uzel, pokud na něj nenavazuje žádný další uzel
- **New node** – přidání nového uzlu
- **New service** – přidá sledování služby pro aktuální uzel
- **Edit service** – umožní změnit nastavení služby
- **Delete service** – smaže službu
- **Edit notification** – seznam techniků, kteří mají být informováni o problémech na síti

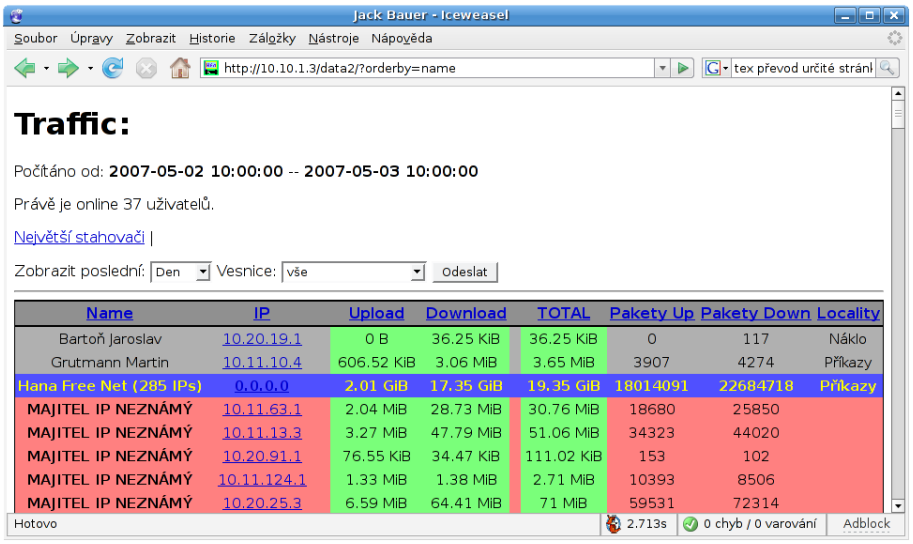
Obsah menu se mění v závislosti na tom zda je zobrazena informace o uzlu či službě. Nabídka **Edit notification** je přístupná jen u kořenového uzlu.



Obrázek A.5: Dohledový systém – menu

A.4.3 Účtování provozu

Na úvodní stránce přehledu přenesených dat je seznam všech uživatelů, kteří byli v posledních 24 hodinách aktivní (obrázek A.6). Tento seznam lze setřídit podle jednotlivých sloupců kliknutím na jejich záhlaví.



Obrázek A.6: Výchozí stránka přehledu přenesených dat

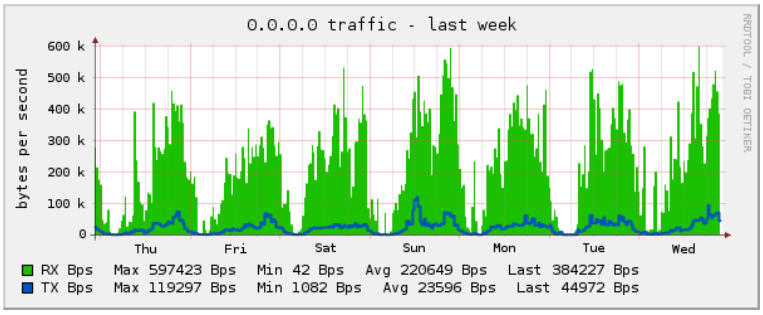
Při prvním přístupu jsou zobrazeny přenesená data za posledních 24 hodin. Je možné si zvolit zobrazení za poslední měsíc a poslední rok. Vždy je vidět pro jaké období jsou platné zobrazené informace.

Množství zobrazených informací lze snížit výběrem konkrétní lokality. Pak budou ve výpisu jen uživatelé z dané lokality.

Přehled přenesených dat má několik vzhledů, které můžete změnit v menu ve spodní části stránky.

Informace o tom, podle kterého sloupce je přehled seřazen, vybraný vzhled, vybraná lokality a délka časového úseku jsou uloženy na vašem počítači v souboru „cookie“. Při další návštěvě stránek je pak použito stejné nastavení.

Pokud v přehledu kliknete na IP adresu, dostanete se ke grafům a detailnímu přehledu přenosu dat konkrétní IP adresy.



Obrázek A.7: Graf přenosu za poslední týden

Jsou zde zobrazeny informace za posledních 24 hodin, poslední měsíc a poslední rok a také grafy průtoku (obrázek A.7).

Příloha B

Generované konfigurační soubory

B.1 DHCP server

Informační systém generuje konfigurační soubory ve formátu kompatibilním s ISC DHCP serverem [12]. Jméno domény a IP adresa jemnného serveru (Domain Name System, DNS) se bere z konfiguračního souboru.

```
option domain-name "hanafree.net";
option domain-name-servers 10.10.1.3;

ddns-update-style none;

default-lease-time 7200;
max-lease-time 720000;

authoritative;

log-facility local7;

subnet 10.20.0.0 netmastk 255.255.0.0 {
        option routers 10.20.1.254;
option broadcast-address 10.20.255.255;

        host h1 {
                hardware ethernet 00:0a:e4:4b:0e:2a;
                fixed-address 10.20.19.1;
        }

}
```

B.2 Autentizace uživatelů

Autentizace uživatelů je prováděna na základě MAC-listů na straně přípojného bodu. Informační systém potom z informací uložených v databázi generuje soubor se seznamem MAC adres povolených na daném AP.

MAC-list pro ap 10.20.1.1 – `maclist-10.20.1.1`**:**

```
00:0b:6b:37:8e:9e
00:b9:2e:ab:57:7a
```

B.3 Omezení IP adres s přístupem na internet

Ne všechny IP adresy v síti musí mít povolen přístup k internetu. Všechny IP adresy, které mají povolenu komunikaci do internetu jsou uloženy v souboru `ip2-internet`.

Ukázka souboru se seznamem adres:

```
10.20.19.1
10.20.19.2
10.10.1.1
10.10.1.2
10.11.14.1
```

B.4 Veřejné IP adresy

Jelikož se v celé síti Haná Free Net používají IP adresy z neveřejného rozsahu, je nutné na haničním směrovači zajistit namapování neveřejné adresy na adresu veřejnou. Soubor s popisem se jmenuje `ip-mapping`.

Ukázka souboru s mapováním IP adres:

```
10.20.19.1 77.48.27.227
10.10.1.2 77.48.27.228
```

B.5 Omezení šířky pásma

V síti Haná Free Net se omezení šířky pásma uplatňuje jen na uživatele, kteří opakovaně poruší pravidla o provozu na síti. Výstupem tohoto module je seznam IP adres, které mají mít na daném přístupovém bodu omezenu šířku pásma.

Tento seznam by šel využít i jako základ pokročilejšího generování pravidel pro řízení šířky pásma.

Omezení šířky pásma na ap 10.11.1.30 – `shaper-10.11.1.30`**:**

```
10.11.10.4
10.11.10.14
```

Příloha C

Agent dohledového systému

Následuje kód agenta dohledového systému:

```
#!/bin/bash
export LANG=C
< /proc/uptime awk '{print \$1 }'
< /proc/uptime awk '{print \$2 }'
free | grep Mem: | awk '{ print \$3/\$2}'
free | grep Swap: | awk '{ print \$3/\$2}'
df | grep hda1 | awk '{ print $3/$2}'
```

Konfigurační soubor `xinetd` superserveru:

```
service bodos
{
        disable         = no
        socket_type     = stream
        protocol        = tcp
        user            = root
        wait            = no
        server          = /usr/sbin/bodos
        only_from       = 127.0.0.1
}
```

Aby tento skript spouštěný přes `xinetd` superserver fungovat, je nutné do souboru `/etc/services` přidat následující řádek:

```
bodos           1234/tcp
```

Příloha D

Oprava pro RRDJtool

Nativní část RRDJtool nešla zkompilovat kvůli nekompatibilitě API mezi verzemi librrd (1.1 – 1.2). Dále je přiložena oprava, po jejíž aplikaci již lze nativní část RRDJtool zkompilovat.

```
--- Rrd-o.c     2003-02-19 12:10:40.000000000 +0100
+++ Rrd.c       2007-03-14 00:53:24.000000000 +0100
@@ -142,7 +142,8 @@ JNIEXPORT jint JNICALL Java_rrd_Rrd_crea
        char **tokens = getTokens(env, ar, n);
        char **calcpr;
        int xsize, ysize, i;
-       int status = rrd_graph(n, tokens, &calcpr, &xsize, &ysize);
+       double ymin, ymax;
+       int status = rrd_graph(n, tokens, &calcpr, &xsize, &ysize, NULL,
+               &ymin, &ymax);
        preserveError();
        if(status != -1) {
                preserveGraphOutput(calcpr, xsize, ysize);
```