VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

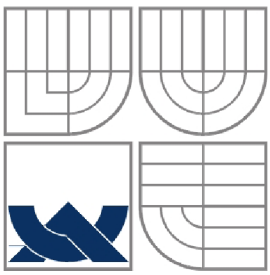
WEBOVÝ PORTÁL PRO GENEROVANÍ ZPRÁV O SÍŤOVÉM PROVOZU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

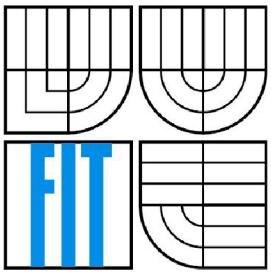
AUTOR PRÁCE
AUTHOR

MIROSLAV LÍZAL

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

WEBOVÝ PORTÁL PRO GENEROVANÍ ZPRÁV O SÍŤOVÉM PROVOZU

WEB PORTAL FOR NETWORK TRAFFIC REPORTING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MIROSLAV LÍZAL
AUTHOR

VEDOUCÍ PRÁCE — ING. JIŘÍ TOBOLA
SUPERVISOR

BRNO 2009

Abstrakt

Bakalářská práce se zabývá vývojem webového portálu pro generování zpráv o síťovém provozu. Představuje dostupné technologie pro sledování dění v síti, důkladněji se zaměřuje především na NetFlow, které je v této práci použito. Popisuje celý vývojový cyklus od analýzy, specifikace a implementace až k testování vytvořené aplikace. Výsledkem práce je systém založený na skriptovacím jazyce PHP a databázi PostgreSQL, který vytváří a zpřístupňuje zprávy o dění v síti přes WWW rozhraní. Poskytuje také možnost uložení výstupů ve formátu PDF.

Abstract

This thesis focuses on development of a web portal, which provides information about network traffic. It describes available technologies, which can be used for the network monitoring. It puts emphasis especially on NeTFlow. This paper also descirebs the whole development cycle from analysis to implementation and testing. The outcome of this work is a online system, based on PHP language and PostgreSQL database, which creates reports on the network traffic and makes them accessible via WWW. It is also able to save these reports to a PDF file.

Klíčová slova

Webový portál, NetFlow, NfDump, síťový provoz, statistika, PHP, HTML, CSS, JavaScript, PostgreSQL, RRDtool, TCPDF

Keywords

Web portal, NetFlow, NfDump, network traffic, statistics, PHP, HTML, CSS, JavaScript, PostgreSQL, RRDtool, TCPDF

Citace

Lízal Miroslav: Webový portál pro generovaní zpráv o síťovém provozu, bakalářská práce, Brno, FIT VUT v Brně, 2009

Webový portál pro generovaní zpráv o síťovém provozu

Prohlášení

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

……
Miroslav Lízal
20.5.2009

Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Jiřímu Tobolovi, za jeho odbornou pomoc a konzultace, které mi poskytoval během tvorby této práce.

Obsah

1 Úvod

V posledních desetiletích se počítačová síť bouřlivě rozvíjí. Její počátky sahají do 60. let 20. století, kdy nastaly první pokusy o propojování počítačů. Poslední dobou jsou sítě propojovány do celosvětové sítě známé pod pojmem Internet.

Sítě lze dělit dle různých kategorií. Zejména podle rozsahu sítě na LAN, MAN, WAN. LAN (Local Area Network) je síť spojující počítače v bytě, domě, případně v několika blízkých domech. MAN (Metropolitan Area Network) vzniká převážně propojováním LAN sítí, případně jednotlivých stanic na větším území, většinou obce a města. WAN (Wide Area Network) je největší typ sítě co se rozlohy týče, do této kategorie spadá mimo jiné i síť Internet.

Počítačová síť se skládá z celé škály zařízení. Nejjednodušší sítě spojující pouze dva počítače si vystačí se síťovými kartami, které jsou navzájem propojené. Pokud potřebujeme propojit více počítačů, musíme použít rozbočovač (hub) nebo přepínač (switch), přes který bude komunikace probíhat. Pro propojení sítí je již třeba směrovač (router). S růstem a propojováním více sítí bude velmi pravděpodobně potřeba využít i dalších prvků, například síťový most (bridge), opakovač (repeater), při potřebě propojit například optickou a metalickou síť měniče rozhraní (mediakonvertory). Toto je však jen výčet některých prvků potřebných pro samotnou komunikaci.

Při propojování prvků do takto rozsáhlých sítí je třeba začít myslet i na počítačovou bezpečnost. Jedním ze základních prvků je firewall – SW (např. stavový firewall iptables) nebo specializovaný HW. Dále je důležité mít přehled o tom, jaká data prochází sítí, a co se v síti děje. Pokud máme kontrolu nad sítí, můžeme odhalovat různé typy útoků – DoS (Denial of Service), DDoS (Distributed Denial of Service). Ty však nejsou jediným problémem, který může narušit bezchybný chod. Síť obsahuje mnoho zařízení, která se mohou dříve či později porouchat.

Pro monitorování sítí lze použít například SNMP (Simple Network Management Protocol). SNMP je jednoduchý rozšířený protokol pro získávání a nastavování hodnot na zařízeních, které tento protokol podporují. Nad SNMP existuje spousta nástrojů pro monitorování sítě. Za zmínku určitě stojí MRTG (The Multi Router Traffic Grapher) nebo Cacti.

Další možnost pro monitorování sítě poskytuje NetFlow technologie, která umožňuje získávat o stupeň více informací než SNMP. Technologie NetFlow, její architektura a princip bude podrobněji popsána ve druhé kapitole.

Výsledkem této práce by měl být webový portál, který je analyzován a specifikován ve třetí kapitole. Čtvrtá kapitola popisuje návrh systému, jeho implementace je obsahem páté kapitoly. Šestá kapitola je zaměřena na testování. Sedmá shrnuje dosažené výsledky a nastiňuje možnosti dalšího vývoje.

2 Monitorování síťového provozu

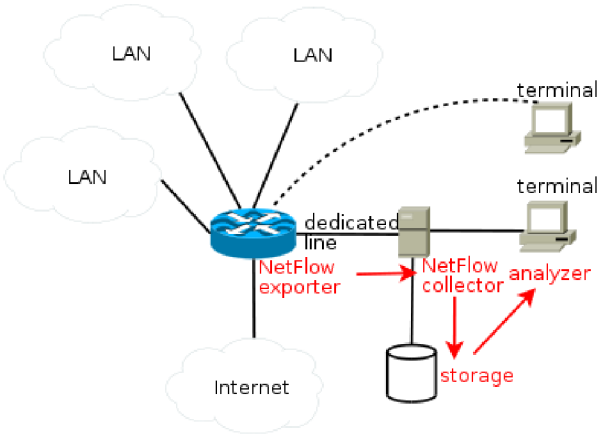
V minulosti sloužilo monitorování síťového provozu zejména k hlídání funkčnosti sítě, tj. odhalování technických problémů (přerušená síťová cesta, nefunkční síťový prvek). Dnešní požadavky jsou podstatně vyšší a vyžadují vývoj stále nových technologií.

2.1 NetFlow

NetFlow je otevřený protokol vyvinutý společností Cisco. Tato technologie nabízí možnost měření a monitorování síťového provozu. Analýzou naměřených dat je možné najít úzké místo v síti, následně tedy umožňuje efektivní plánování dalšího rozvoje sítě. Může sloužit také jako zdroj dat pro odhalování různých typů útoků nebo zjištění hlavních zdrojů provozu. ISP (Internet Service Provider) může podle NetFlow záznamů účtovat ceny služeb na základě množství přenesených dat, popřípadě zavádět FUP (Fair User Policy, zamezení znevýhodnění ostatních uživatelů jedním uživatelem, který nadměrně využívá služeb) podle získaných hodnot [1].

2.1.1 Princip NetFlow

Architektura se skládá ze dvou základních prvků – NetFlow exportér (exporter) a NetFlow kolektor (collector). Pro další práci je vhodné do architektury zahrnout úložiště (storage) NetFlow dat a analyzátor (analyzer).



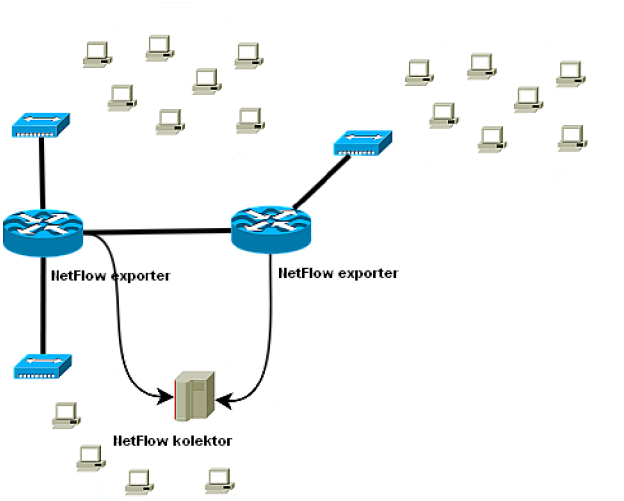
Ilustrace 1: Architektura NetFlow, převzato z [1]

NetFlow exportér (router, switch, sonda nebo jiný síťový prvek podporující technologii NetFlow) monitoruje data procházející po lince, ke které je připojen, a vytváří z nich toky (tok – flow). Tok je definován pěticí údajů – zdrojová a cílová IP adresa, zdrojový a cílový port a číslo protokolu. Tok je tedy záznam vždy jen jednoho směru komunikace, tzn. komunikace je v síti zaznamenána vždy nejméně dvěma toky. Takto vytvořené toky odesílá na NetFlow kolektor. [2]

NetFlow kolektor je připojen k exportéru vyhrazenou linkou, po které přijímá toky. Jeden NetFlow kolektor může ukládat data i z více NetFlow exportérů. Některé kolektory umožňují toky posílat i dalším kolektorům. Tato data je později možné využít pro výpočet různých statistik.

2.1.2 Standardní architektura

Architektura, která byla původně navržena společností Cisco, využívala jako exportéry přímo síťové prvky (routery, switche...). Tyto prvky, mimo své hlavní úlohy, měly za úkol generovat NetFlow záznamy, to mělo za důsledek snížení výkonu samotného síťového prvku. Jedním z řešení bylo například použít vzorkování (zkoumá se pouze každý n-tý paket). Vzorkováním se ale mohou ztrácet důležitá data pro analýzu (např. detekci útoků). Síťové prvky s podporou NetFlow technologie jsou také i podstatně dražší.



Ilustrace 2: Standardní architektura

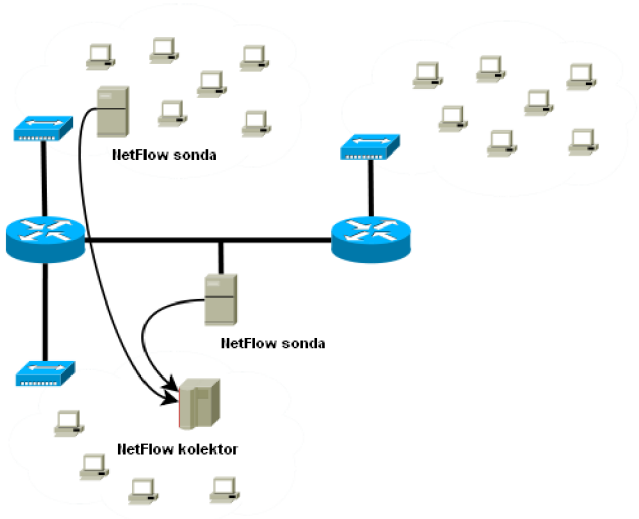
2.1.3 Moderní architektura

Výše popsané nedostatky mohou řešit NetFlow sondy, které přidávají navíc i další možnosti v jejich nasazení. NetFlow sonda je zařízení, které monitoruje pakety procházející sítí. Sondu lze, na rozdíl od

směrovačů, přepínačů a dalších prvků, připojit kamkoliv do sítě transparentním způsobem. Připojuje se pomocí TAP nebo SPAN prvků. [1]

TAP je hardwarové zařízení, které nabízí přístup k datům tekoucím počítačovou sítí. TAP prvek má vždy nejméně tři porty – dva pro připojení do sítě (mezi dva původně sousední síťové prvky) a jeden monitorovací. Tento síťový prvek je pro ostatní prvky v síti neviditelný (transparentní, nedetekovatelný). Prvky propojené přes TAP se stále domnívají, že jsou propojeny přímo. Z tohoto důvodu jsou tedy neviditelné i NetFlow sondy. Na monitorovací port jsou kopírována data tekoucí linkou mezi takto spojenými síťovými prvky. [3]

SPAN (Switched Port Analyzer) funguje na principu zrcadlení portu. Data z jednoho portu (nebo celé VLAN – Virtual LAN) se kopírují na port, na kterém probíhá monitoring. V našem případě port, na kterém je připojena NetFlow sonda. SPAN je označení, které používá společnost Cisco. Společnost 3Com tuto technologii nazývá RAP (Roving Analysis Port). [4]



Ilustrace 3: Moderní architektura

2.1.4 Verze NetFlow

NetFlow se postupem času vyvíjelo. Začalo se hojně využívat až od verze 5. Verze 2, 3 a 4 nebyly vydány. Poslední vydanou verzí je verze 9. [5]

- Verze 1: první verze NetFlow
- Verze 5: přidána podpora informací o BGP (Border Gateway Protocol) autonomních systémech a sekvenční číslování toků.

- Verze 9: přidána podpora různých technologií jako je multicast, IPSec (Internet Protocol Security) a další. Novinkou v této verzi je struktura daná šablonou, která umožňuje mnoho kombinací. Formát NetFlow verze 9 je popsán v RFC 3954 [6].

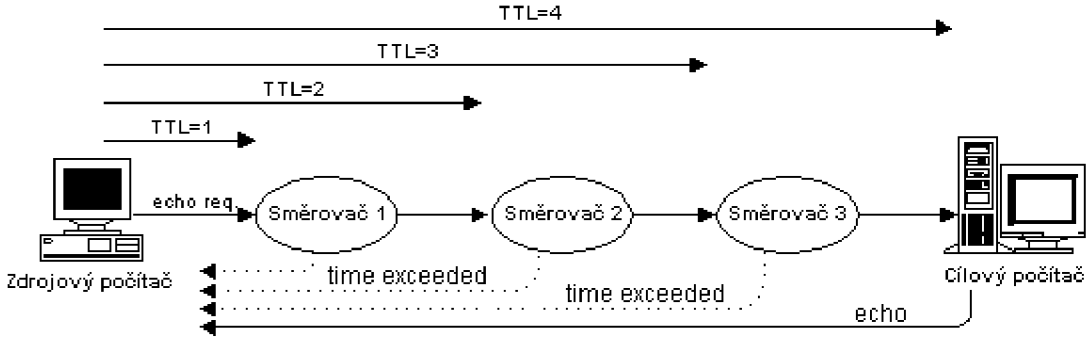
2.2 Další technologie

NetFlow není jedinou technologií umožňující monitoring sítě. Za zmínku určitě stojí velmi známé protokoly ICMP a SNMP nebo relativně nová technologie IPFIX.

2.2.1 ICMP

ICMP (Internet Control Message Protocol) je součástí IP vrstvy modelu TCP/IP. Slouží pro předávání chybových a řídících zpráv. ICMP protokol využívá například i známý příkaz ping, konkrétně příkazy Echo Request (8) a Echo Replay (0). Další zprávou je například Destination Unreachable (3) – informace o nedostupnosti cíle. [7]

Na protokolu ICMP je založen i další známý příkaz tracert / traceroute. Dokáže zobrazit celou cestu (uzly), kterou prochází data k cíli. Nejčastěji se využívá pro detekci chyby v síti (například přerušený spoj). Funguje na principu postupného zvyšování TTL (time to live).



Ilustrace 4: Princip programu traceroute, převzato z [8]

2.2.2 SNMP

SNMP (Simple Network Management Protokol) slouží pro přenos údajů o stavu sítě. Systém postavený nad touto technologií se skládá ze dvou druhů zařízení. Prvním je sledované zařízení (managed device), které uchovává údaje o své činnosti (přenesená data na daném portu, IP adresa rozhraní, stav rozhraní...). Druhým je centrální řídící stanice NMS (Network Management Station),

která získává informace o sledovaných zařízeních, analyzuje data a poskytuje je administrátorovi sítě. [7]

Podporu SNMP obsahuje velké množství zařízení od aktivních síťových prvků, přes počítačová čidla, až po tiskárny. Hodnoty lze periodicky ze zařízení číst, dále ukládat do databáze a následně je zpracovávat nebo jen vykreslit do grafu v závislosti na čase.

Každá hodnota v SNMP je identifikována pomocí OID (object identifikator) identifikátoru. OID je tvořeno několika čísly oddělenými tečkou. OID slouží jako identifikátor MIB databáze, která je uložena formou stromové struktury a popisuje jednotlivé hodnoty. MIB databáze není potřebná pro vlastní práci, pouze ji významným způsobem usnadňuje. Jedním z mnoha programů, umožňujícím práci s MIB databází, je MIB Browser. [9]

SNMP používá pro komunikaci protokol UDP, takže může docházet ke ztrátám datových paketů, i když to v rámci LAN sítě nebývá běžné.

2.2.3 IPFIX

IPFIX (IP Flow Information Export) je exportní formát, jehož cílem je sjednotit právě exportní formáty. Tento exportní formát je založen na NetFlow verze 9. Tok je opět definován jako libovolný počet paketů stejné zdrojové a cílové IP adresy, zdrojového a cílového portu a čísla protokolu.

Požadavky na tento formát jsou popsány v RFC 3917. IPFIX upřednostňuje SCTP (Stream Control Transmission Protocol), ale umožňuje použít i TCP a UDP protokol pro odesílání exportovaných informací. Toto RFC mimo jiné obsahuje požadavek na pasivní monitorování QoS (Quality of Service). [10]

3 Analýza a specifikace

Před započetím jakékoliv práce je třeba nejprve provést analýzu požadavků. Provedení důkladné analýzy je nezbytnou součástí úspěšného vyřešení problému. V této fázi jsou zjištěny především požadavky na daný systém, jsou stanovena vstupní a požadovaná výstupní data. Po provedení analýzy následuje vytvoření specifikace, podle které bude aplikace vypracována.

3.1 Aplikace

Již ze zadání práce vyplývá první základní požadavek, a to, že se bude jednat o webovou aplikaci. Webovou aplikací rozumíme aplikaci, která je poskytována skrze (internetovou nebo intranetovou) síť. Tento typ aplikací je poslední dobou velice oblíben.

Za jeho oblibou stojí zejména odpadající potřeba instalace specializovaného programu. Požadován je pouze internetový prohlížeč, který je v dnešní době ve většině případů součástí základní instalace operačního systému (MS Windows – Internet Explorer; Linux – Firefox (Iceweasel), Konqueror; Mac OS – Safari).

Webová aplikace (z pohledu klienta) není závislá na operačním systému. Internetový prohlížeč může plnit funkci tzv. tenkého klienta (veškerá logika aplikace je prováděna na serveru a klient zobrazuje pouze výsledek).

Důležitou vlastností webové aplikace je bezesporu její snadné nasazení (pouze jedna instalace) a následná údržba. Odpadá tím instalace u všech uživatelů, kteří potřebují danou aplikaci používat. Při použití standardní (desktopové) aplikace bychom museli při vydání nové verze aktualizovat opět všechny instalace.

Dalším požadavkem na aplikaci by měla být možnost její lokalizace do různých jazyků. Ve výchozím stavu bude aplikace lokalizována do českého a anglického jazyka. S tímto požadavkem úzce souvisí i použité kódování, ve kterém bude aplikace naprogramována. Pro bezproblémové zobrazení v různých jazycích je nejlepší použít kódování UTF-8.

Aplikace by se měla správně zobrazovat alespoň v prohlížečích Mozilla Firefox (v aktuální verzi 3.x) a Internet Explorer verze 7. Vzhled aplikace by měl zůstat konzistentní při různých rozlišeních obrazovky. Minimální podporované rozlišení, pro které se webová stránka zobrazí správně, bude 1024x768 obrazových bodů (při nižším se mohou zobrazit posuvníky). Aplikace musí správně pracovat jak s použitím protokolu HTTP (HyperText Transfer Protocol), tak s použitím protokolu HTTPS (HyperText Transfer Protocol Secure).

3.2 Statistiky

Aplikace musí umožňovat tvorbu široké škály statistik a příslušných grafů. Definici statistik by bylo vhodné přidávat bez zásahu do programového kódu. Systém tedy nebude omezen na předpřipravené definice statistik, ale bude umožňovat přidání nových definic a stávající definice umožní upravovat.

Jednotlivé statistiky se budou skládat vždy z grafu a tabulky. Takovouto statistiku budeme nazývat kapitolou. Kapitoly bude možné sdružovat do větších celků nazývaných reporty. Po vytvoření reportu bude tedy možné do něho zařadit jednu nebo více kapitol.

Aplikace by měla poskytovat dva typy statistik. Takzvané „Top“ a „Traffic“. Reporty budou dostupné přímo online a navíc i offline. Offline reportem se myslí report zaslaný emailem.

3.2.1 Top statistika

V „Top“ statistikách nás budou zajímat zejména údaje, které nějakou hodnotou „vyčnívají z řady“. Nejčastěji to budou například stanice s největším přenosem dat, nejvíce spojeními nebo třeba nejčastěji používané protokoly, za danou časovou jednotku.

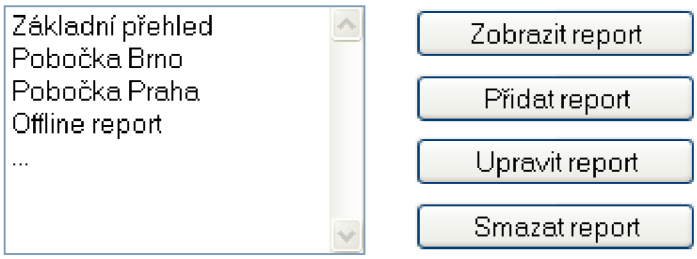
Top statistika se bude skládat z tabulky o definovaném počtu „top“ hodnot. Hodnoty z této tabulky budou zobrazeny v grafu, který by měl být přehledný. Z tohoto důvodu bude použit graf výsečový, který nejlépe zobrazí rozložení zastoupení jednotlivých hodnot.

3.2.2 Traffic statistika

Vhodné by také bylo zobrazovat statistiky hodnot v závislosti na čase. Pro tento typ statistik se nejlépe hodí například graf vytížení sítě v průběhu času. V jednom grafu by mělo být možné zobrazit více než jednu hodnotu (například pro statistiku download vs. upload). Traffic statistika bude tedy znázorněna grafem spojnicovým, kde na ose y bude zobrazena sledovaná hodnota a na ose x časový interval.

3.2.3 Uživatelské rozhraní

Z pohledu uživatelského rozhraní bude při přechodu do příslušné sekce portálu zobrazen formulář s výběrem dostupných reportů, po výběru se daný report zobrazí. Dále by z tohoto místa mělo být možné vytvořit report nový, nebo upravit a smazat existující. Na obrázku níže je vidět návrh, jak by tato navigace mohla vypadat.



Ilustrace 5: Navigace mezi reporty

3.2.4 Export

Uživatelské reporty bude možné zasílat na email. Exportovaný report by měl být co nejvíce podobný online reportu. V případě již vzniklého požadavku na export by měla být možnost provést export reportu i z webového rozhraní.

Pro export bude použit formát PDF (Portable Document Format), který zajistí, že se dokument zobrazí vždy tak, jak bude vyexportován. Jedná se o standardizovaný formát, který je možné zobrazit v různých operačních systémech.

3.3 Uživatelé

Existuje spousta nástrojů, které umožňují monitorovat stav sítě. Výsledek této práce by měl být zaměřený, na rozdíl od většiny ostatních aplikací, spíše na méně zkušené uživatele, kterým nabídne srozumitelný přehled o dění v síti.

Aplikace bude víceuživatelská. Každý uživatel si bude moci definovat vlastní reporty, ukládat své uživatelské nastavení aplikace. O uživatelích je třeba uchovávat základní informace – jméno a email. Kvůli ochraně naměřených dat bude třeba zajistit dostatečnou autentizaci.

Veškerá nastavení bude uživatel provádět v sekci „Nastavení“. Zejména se bude jednat o nastavení jména, emailu, jazyka, offline reportů a intervalu jejich zasílání.

Pro uložení uživatelského nastavení (informace o uživateli), vytvořených reportech a příslušnost kapitoly k danému reportu bude využita databáze. Důvody pro použití plnohodnotné databáze jsou popsány ve čtvrté kapitole.

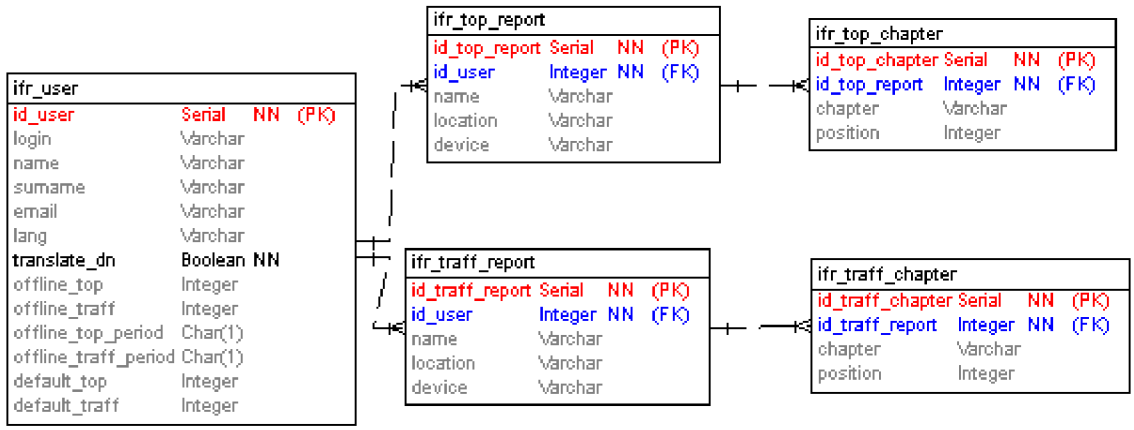
4 Návrh systému

Před samotnou implementací bylo třeba vhodně navrhnout systém (vytvořit model) tak, aby nebylo nutné v průběhu samotné implementace měnit mnoho věcí.

Dále tedy bylo nutné vhodně namodelovat systém a požadavky na něj popsané ve fázi analýzy a specifikace. Pro namodelování datového úložiště informací o uživatelích je využit ER (Entity-relationship) diagram popisující vztahy mezi entitami v systému. Samotná funkčnost systému bude popsána pomocí diagramu případů užití (Use Case diagram).

4.1 ER diagram

ER diagram obsahuje typy entit a vztahů mezi nimi. Entita v ER diagramu představuje objekt reálného světa. V našem případě je to například uživatel.



Ilustrace 6: Schéma databáze

Popis jednotlivých entit:

ifr_user – Obsahuje informace o uživateli a jeho nastavení (zejména nastavení vybrané lokalizace aplikace, vybrané výchozí reporty, offline reporty a periody jejich zasílání).

ifr_top_report – Uchovává uživatelem definované top reporty. U každého reportu je možné definovat jeho jméno, fyzické umístění, ze kterého je tvořen, a zařízení, ze kterého pocházejí data. Každý takto vytvořený report je vázán na uživatele, který ho vytvořil.

ifr_top-chapter – Obsahuje kapitoly navázané na daný report. Důležitým atributem této entity je pozice. Pozice určuje pozici kapitoly v reportu.

ifr_traff_report – Stejná jako top_report, jen obsahuje data o top kapitolách.

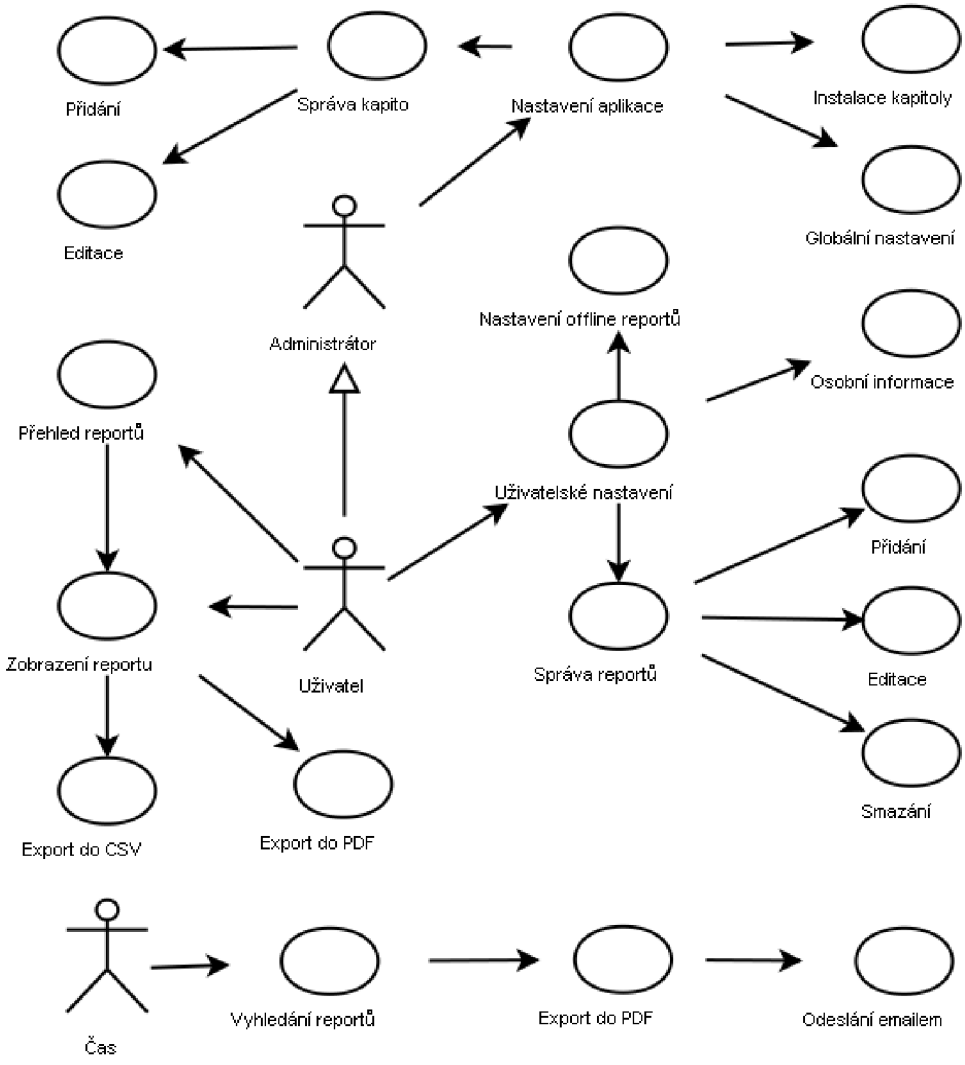
ifr_traff_chapter – Stejná jako top_chapter, jen obsahuje data o top kapitolách.

4.2 Diagram případů užití

Diagram případů užití (Use Case diagram) je jedním z UML (Unified Modeling Language) nástrojů. Definuje role v aplikaci a akce, které mohou být prováděny.

V naší aplikaci jsou role tři. První rolí je uživatel, specializací vznikne role druhá – administrátor systému. Administrátor má oproti uživateli možnost nastavovat aplikaci, tzn. upravovat globální nastavení, instalovat nové kapitoly a upravovat je.

Neméně důležitou rolí je čas. Čas slouží jako spouštěč části aplikace, která se stará o vyhledání reportů, které je třeba vyexportovat a odeslat uživateli emailem (offline reporty).



Ilustrace 7: Diagram případů užití

5 Implementace

5.1 Použité technologie

Základním stavebním kamenem webové aplikace je webový server, který má na starosti zpracování HTTP požadavků. Webových serverů je mnoho, například lighttpd, IIS (Internet Information Services) pro operační systém Microsoft Windows nebo velice oblíbený a nejrozšířenější Apache, který je multiplatformní. Apache je snadno konfigurovatelný, bezproblémový, snadno rozšiřitelný webový server. Z těchto důvodů byl právě Apache verze 2 použit jako pevná základna celého systému.

Jako programovací (skriptovací) jazyk bylo zvoleno rozšířené PHP. Pro uchovávání dat spravovaných aplikací je samozřejmostí plnohodnotná databáze. Jedním z důvodů je snadnější práce, ve srovnání s využíváním textových nebo XML souborů, které by pro tento účel dostačovaly. Dále byla databáze použita zejména s výhledem na možnosti dalšího rozšiřování aplikace. Tato případná rozšíření budou popsána v poslední kapitole.

Pro práci s NetFlow daty byl zvolen nástroj NfDump (navedení na tento nástroj bylo i v doporučené literatuře).

5.1.1 PHP

Vznik jazyka PHP (Personal Home Page) se datuje do roku 1994, kdy si dánský programátor Rasmus Lerdorf vytvořil tento nástroj pro svoji vlastní potřebu. První verze byla vydána v roce 1995. V roce 1998 vyšla verze 3, která obsahovala přepsaný parser programátory Zeevem Suraskim a Andim Gutmansem. Ve verzi 3 byl také změněn název jazyka na „PHP: HyperText Procesor“ a jednalo se o první verzi, která běžela také pod Windows. V roce 2000 byla vydána verze 4, která byla vyvíjena do roku 2008. Jednou z hlavních novinek ve verzi 5 je vylepšená podpora objektově orientovaného programování. Tato verze se vyvíjí od roku 2004 a nyní je jedinou stabilní verzí, která je stále aktivně podporována. V přípravě je již verze 6, která by měla plně podporovat kódování Unicode [11] [12].

PHP umožňuje přístup ke spoustě dalších technologií, jako jsou databáze (např. dBase, Firebird, MS SQL, MySQL, PostgreSQL), autentizační služby (Kerberos, Radius), práce se souborovým systémem, mailové služby (IMAP, POP3, SMTP), práce s XML a další množství služeb (cURL, FTP...) [12]

Pro PHP také existuje mnoho rozšíření a frameworků. Za zmínku stojí alespoň PEAR (PHP Extension and Aplication Repository) nebo Zend Framework.

5.1.2 HTML

HTML (HyperText Markup Language) je značkovací jazyk. Od verze 2, která byla vydána v roce 1994, patří do rodiny jazyků SGML (Standard Generalized Markup Language). HTML je jedním z jazyků pro vytváření webových stránek. Jazyk HTML byl navržen v roce 1990 v CERNu (Evropská organizace pro jaderný výzkum - Conseil Européen pour la Recherche Nucléaire) spolu s protokolem HTTP (Hypertext Transfer Protocol) pro jeho přenos.

Od verze 3.2 se standardizací (doporučeními) jazyka zabývá organizace World Wide Web Consortium (W3C), která vznikla v roce 1994. Před založením organizace vznikaly různě upravené verze jazyka HTML, které nebyly kompatibilní s verzemi ostatních firem. Tato organizace si dala za cíl sjednotit specifikace různých firem a dohodnout se s nimi na základech jazyka. V této verzi byly přidány tabulky, podpora formátování vzhledu a zarovnání textu. [13]

Na konci roku 1997 vyšla verze 4, do které byla, i mimo jiné, přidána podpora rámů. Tato verze se snaží dosáhnout původního účelu jazyka – oddělit vzhled od významu (sémantiky). Vzhled by měl být definován styly. Verze 4 měla být původně poslední verzí jazyka, po které byl plánován přechod na XHTML (následník HTML využívající univerzální jazyk XML). Vývoj XHTML se některým společnostem nezamlouval, a tak vznikla pracovní skupina The Web Hypertext Application Technology Working Group (WHATWG). V roce 2007 bylo odhlasováno, že vznikne i verze 5, jejímž základem bude specifikace Web Aplications 1.0 a Web Forms 2.0 pracovní skupiny WHATWG. [14]

5.1.3 CSS

CSS (Cascading Style Sheets – tabulky kaskádových stylů) je jazyk navržený organizací W3C. Jeho hlavním smyslem je oddělení struktury a vzhledu dokumentu.

Přináší větší možnosti formátování než samotné HTML. Není třeba nastavovat vzhled jednotlivých HTML dokumentů, ale umožňuje pomocí jednoho stylu definovat vzhled celé webové prezentace. Pouhým přepsáním jednoho souboru lze upravit vzhled celé webové prezentace bez nutnosti editovat samotné HTML dokumenty. Bez stylů by to znamenalo projít všechny zdrojové kódy, vyhledat měněnou značku a upravit její vzhled.

S využitím stylů se zmenšuje i velikost výsledné stránky a tedy i přenášená data. Lze snadno definovat i více stylů a uživatel si může vybrat takový, který mu nejvíce vyhovuje, popřípadě definovat styl pro různá zařízení (např. PDA, mobilní telefon, PC, tiskárna).

Hlavní nevýhodou kaskádových stylů je různý stupeň podpory a implementace v jednotlivých prohlížečích. Napsat tedy kód, který bude ve všech prohlížečích interpretován stejně, nebo alespoň podobně, může být v některých případech nesnadný úkol. Zejména bývá problém s prohlížečem

Internet Explorer firmy Microsoft. Tato situace se výrazně zlepšila ve verzi 7 a verze 8 bude údajně dodržovat veškeré standardy.

5.1.4 JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk vyvinutý společností Netscape. Využívá se zejména jako interpretovaný programovací jazyk pro webové stránky na straně klienta.

Jeho syntaxe vychází z jazyků C a Java. S jazykem Java nemá kromě podobného názvu a podobné syntaxe nic společného. JavaScript je, na rozdíl od Javy, která je staticky typovaná, typovaný dynamicky (stejně jako většina skriptovacích jazyků). Jazyk je „Case sensitive“ (citlivý na velikost písmen). [15]

Využívá se zejména pro předzpracování odesílaných dat z formulářů. Před odesláním formuláře mohou být zadaná data zkontrolována a uživatel upozorněn na chyby bez nutnosti odeslání dat na server. V žádném případě by se ale nemělo spoléhat pouze na kontrolu dat na straně uživatele. Další ze spousty možností JavaScriptu je například schopnost měnit styly již načtené stránky nebo periodicky vyvolávat operace (například běžící hodiny).

JavaScript je také nedílnou součástí dnes hojně využívané technologie AJAX (Asynchronous JavaScript and XML), která umožňuje interakci se serverem, bez nutnosti načítat znovu celou stránku.

5.1.5 NfDump

Pro pochopení dalšího textu by nejprve bylo vhodné upozornit na to, že NfDump je sada nástrojů. Názvem NfDump se také označuje jeden nástroj z tohoto balíku. Je tedy třeba rozlišovat, zda se bavíme o celém balíku nástrojů, nebo přímo o nástroji NfDump [16].

Všechny nástroje z tohoto balíku podporují NetFlow verzi 5, 7 a 9. Součástí balíku jsou tyto programy:

- nfcapd – Čte NetFlow data ze sítě a ukládá je do souborů.
- nfdump – Umožňuje přístup k uloženým datům nástrojem nfcapd.
- nfprofile – Filtruje data uložená programem nfcapd dle zadaných filtrů a ukládá je do souborů pro pozdější využití.
- nfreplay – Čte data z uložených souborů programem nfcapd a posílá je po síti dalším počítačům.
- nfclean.pl – Vzorový skript pro mazání starých dat. Vhodný například pro automatické periodické spouštění třeba pomocí systémového nástroje CRON.
- ft2nfdump – Konvertuje flow-tools data do formátu NfDumpu.

Program NfDump obsahuje řadu přepínačů a parametrů, které jsou podobné těm v nástroji tcpdump. Právě pomocí tohoto nástroje jsou tvořeny „Top“ statistiky. Umožňuje definovat počet „top“ řádků, které nás zajímají, filtry a výstupní formát.

Výrazy pro filtr lze spojovat pomocí logických operátorů *and*, *or* a *not*. Je možné filtrovat dle verze IP protokolu, čísla protokolu (TCP, UDP, ICMP...), zdrojové a cílové IP adresy, adresy sítě, portu, rozhraní (síťový interface), směru toku (IN, OUT), příznaků (flags – např. ACK, SYN, FIN,..), TOS (Type of Service), délky trvání a dalších.

Výstup lze definovat dle potřeb pomocí parametru přepínače *-o*, tím lze zajistit výstup vhodný pro další zpracování (parser).

Výsledný příkaz pro nfdump pomocí kterého chceme zjistit pět nejpoužívanějších portů pro komunikaci stanice 192.168.3.108 řazených dle přeneseného množství dat:

```
nfdump -R /cesta/k/adresari/s/daty/ -n 5 -s port/bytes 'IP 192.168.3.108'
```

Pro podrobný popis se doporučuje projít manuálové stránky programu (man nfdump).

5.1.6 PostgreSQL

PostgreSQL je objektově-relační databázový systém. Je šířen pod BSD licencí. Vyniká především stabilitou a rychlostí. Podporuje normy SQL92 a SQL99. Cizí klíče, pohledy a transakce jsou ve dnešních databázích považovány za standard. Mile potěší stored procedury, triggery, možnost definovat vlastní typy.

Mimo základních typů podporuje i typy geometrické (bod, plocha, cesta, kružnice,...), typ pro ukládání síťových adres IPv4 a IPv6 (cidr, inet) a MAC adres (macaddr) a funkce pro práci s nimi.

Za zmínku ve spojení s PostgreSQL stojí PostGIS. PostGIS přidává podporu pro ukládání geodat. Přidává další geometrické typy, prostorové funkce pro práci s vektorovými objekty, pro výpočty vzdáleností a mnoho dalších. Tuto databázi podporuje například Grass (open source GIS nástroj) [17].

Pro PostgreSQL vyšla řada seriálů na serveru www.linuxsoft.cz. Jedná se o jeden z nejlepších zdrojů v češtině. V prvních kapitolách je průřez databázovými technologiemi, popsána instalace, základní práce s databází. Dále se věnuje i pokročilejším tématům, jako jsou uložené procedury, triggery, správa uživatelů.

5.1.7 RRDtool

RRDtool je sada nástrojů pro práci s databází založených na uchovávání, zobrazování a práce s hodnotami v závislosti na čase. Data se ukládají v časových intervalech – krocích. Vhodné pro ukládání stavu sítě (aktuální vytížení linky), zaznamenávání teplot, vytížení procesoru, stavu paměti.

Na webových stránkách projektu je dostupný přehledný manuál a velké množství tutoriálů, které usnadňují začátky s tímto nástrojem.

Tento nástroj byl pro jeho povahu využit ke generování „Traffic“ kapitol, protože generuje přehledné grafy, na které jsou uživatelé zvyklí i z různých dalších programů (MRTG, Cacti,...)

5.2 Použité knihovny

5.2.1 pChart

Pro generování „Top“ kapitol, přesněji grafů do těchto kapitol, byla zvolena tato knihovna. Umožňuje generovat velké množství typů grafů - spojnicové, sloupcové, výsečové a 3D výsečové. Pro zobrazování „top“ veličin je vhodný právě výsečový graf, ve kterém je nejlépe vidět zastoupení jednotlivých položek.

Na webových stránkách projektu je dostupná nápověda s příklady. Jedním z důvodů proč byla vybrána právě tato knihovna, je možnost si ji vyzkoušet přímo na webu i s možností zobrazení výsledného zdrojového kódu. Knihovna je stále aktivně vyvíjena.

5.2.2 TCPDF

Při výběru knihovny hrála hlavní roli jednoduchost. Pro generování PDF se hojně využívá knihovna FPDF. Tato knihovna má však podstatný nedostatek, neumožňuje exportovat texty v kódování UTF-8. Možností by bylo texty před exportem převést do jiného kódování (např. ISO-8859-2), se kterým knihovna FPDF nemá potíže. Nastal by však problém v případě potřeby přidání některého z exotičtějších jazyků, jehož abeceda obsahuje speciální znaky. Proto bylo důležité najít takové způsob, které by umožňoval exportovat dokumenty bez těchto, nepříliš systémových, pomocných kliček.

Řešením je, právě v této práci použitá, knihovna TCPDF. Vychází z výše zmíněné knihovny FPDF a podporuje mimo jiné i export textů v kódování UTF-8. Knihovna je distribuovaná pod licencí GNU LGPL (GNU Lesser General Public License).

Instalace probíhá, jako u většiny PHP knihoven, pouhým nakopírováním a nastavením hodnot v konfiguračním souboru a nastavením příslušných práv adresářů.

Tato knihovna je zajímavá také tím, že mimo běžné sazby umožňuje vysázet i HTML stránku. Při sazbě HTML stránky je třeba dávat pozor na to, co všechno je knihovna schopna vysázet, a stále kontrolovat, zda je dosaženo požadovaného výsledku. Při práci s ní bylo například zjištěno, že není schopna správně vysázet tabulku obsahující obrázek (nebyl vysázen). Jistě však ulehčí práci v základní sazbě tabulek, které jinak není lehké vygenerovat tak, aby vypadaly, jak je požadováno.

Za velký klad knihovny lze určitě považovat velmi kvalitní dokumentaci a spoustu vzorových příkladů. Knihovna je stále aktivně vyvíjena (zatím poslední verze byla vydána v květnu 2009).

5.3 Struktura aplikace

Aplikace byla od začátku navrhnuta tak, aby ji bylo možné kdykoliv snadno rozšiřovat pomocí takzvaných zásuvných modulů (plug-in). V základu obsahuje následující moduly: Přehled, Top reporty, Traffic reporty, Nastavení a Nápověda.

Systém je navržen tak, že jeden modul je jedna programová třída. Tvorba modulu je velice snadná. Aby byl modul použitelný, třída musí obsahovat atributy *$name*, *$icon*, *$action*, konstruktor a funkci *main()*. Atribut *$name* definuje název modulu, *$icon* ikonu, která bude zobrazena vedle názvu modulu a *$action*.

```
<?php
class Nazev_modulu
{
  var $name = "Nazev_modulu";
  var $icon = "Ikona_modulu.gif";
  var $action = NULL;

  function Nazev_modulu() {
    $this->action = "./?plugin=".substr(basename(__FILE__), 0, -4);
  }
  function main() {
    global $l,$user,$plugins;
    // VLASTNI KOD
  }
}
$nazev_modulu = new Nazev_modulu();
?>
```

Ilustrace 8: Šablona modulu

5.3.1 Přehled

Slouží jako výchozí obrazovka aplikace po úspěšném přihlášení. Obsahuje přehled nadefinovaných Top a Traffic reportů a možnost definovat nový report. Obsahuje také dva grafy pro základní přehled. O které grafy se jedná, je definováno globálně pro celou aplikaci v konfiguračním souboru položkami *STATUS_TOP_CHAPTER* a *STATUS_TOP_INTERVAL* pro Top report a *STATUS_TRAFF_CHAPTER* a *STATUS_TRAFF_INTERVAL* pro Traffic report.

Původní návrh tento modul neobsahoval, potřeba jeho vzniku nastala až v průběhu vývoje, zejména z důvodu uživatelské přívětivosti. Důvody budou podrobněji popsány níže.

5.3.2 Top reporty

Při specifikaci bylo definováno, že výchozí stránkou reportu bude rozcestník vedoucí na jednotlivé reporty. Již při programování bylo zjištěno, že se v žádném případě nejedná o efektivní řešení a při přepínání se mezi reporty je třeba velkého množství kliknutí. Z tohoto důvodu vznikl výše popsaný rozcestník (modul *Přehled*). Jako první report se tedy v modulu zobrazí výchozí report, který si vybere uživatel v nastavení aplikace. Pro ještě snadnější přepínání bylo vedle názvu reportu přidáno „rozbalovátko“ (selectbox), které při výběru reportu uživatele automaticky přesměruje na daný report.

Další problém vznikl při prvních pokusech o získávání dat z aplikace NfDump. Bylo zjištěno, že se data načítají příliš dlouho (zejména pro delší časové intervaly). Report může obsahovat i více kapitol, potřebný čas pro zobrazení reportu se tedy násobil počtem kapitol v reportu. Data z NfDumpu jsou potřebná na dvou místech – v samotném reportu (tabulka) a ve skriptu pro generování výsečového grafu. Řešením by bylo si již jednou načtená data ukládat jako mezivýsledek i pro graf (předávat například pomocí session). Načtení dat by však bylo i tak časově velice náročné. Bylo tedy třeba najít jinou možnost.

Nejjednodušším řešením bylo si data předpřipravit a poté jen pracovat s těmito hodnotami. Bylo třeba určit časové intervaly, pro které budou data přichystána. Výsledkem jsou intervaly: Dnes, Tento týden, Tento měsíc, Včera, Minulý týden, Minulý měsíc. Později byl přidán ještě jeden interval – Uživatelský. Tato hodnota je globální pro celou aplikaci a její editaci má povolenou pouze administrátor.

Pro předpočítávání statistik byla použita sada perlových a bashových skriptů. Předpočítané hodnoty se nacházejí v adresáři *./reporter-data/top-reports/*. Tento adresář obsahuje další adresáře, v nichž jsou již umístěny adresáře se samotnými kapitolami (co adresář, to jedna kapitola). Každý adresář reprezentuje časovou jednotku, za kterou jsou statistiky vyhodnoceny, např. den, týden, měsíc. Kapitola obsahuje předpočítaná data ve všech jazykových mutacích (hlavičky sloupců) v souborech *table_jazyk.csv*.

Každá kapitola navíc obsahuje konfigurační soubor *chapter.cfg* obsahující texty s popisem kapitoly a soubor *timestamp.txt* s časovým rozmezím, ve kterém jsou data aktuálně předpočítána.

Pro možnost snadného získání hodnot pro případný přenos do jiné aplikace byla přidána funkcionalita pro export do CSV (Comma-separated values). Exportovat lze celý report, ale i jednotlivé kapitoly. Titulky sloupců jsou exportovány v jazyce, který má uživatel aktuálně v aplikaci nastaven.

5.3.3 Traffic reporty

Traffic reporty, jak bylo popsáno výše, obsahují data v závislosti na čase. Pro zobrazení tohoto typu statistik se jevilo jako vhodné použít sadu nástrojů RRDtool. RRD databáze je snadné jednoduše definovat v grafické nástavbě NfDumpu NfSen.

Na rozdíl od Top reportů zde nebyl problém s délkou trvání výpočtů. Je tedy možné uživateli nechat volnou ruku při výběru časového období, za které budou zobrazeny statistiky. Vybírací lišta časového období má předdefinované intervaly, umožňuje snadné posouvání o zadaný interval, ale i ruční zadání začátku a konce intervalu.

Ilustrace 9: Lišta pro výběr intervalu

Každý report má vlastní konfigurační soubor. Ten obsahuje mimo jiné popis kapitoly, titulek,... Důležitou hodnotou je zdrojový adresář s nadefinovanými RRD soubory. Při požadavku na vykreslení kapitoly je tento adresář rekurzivně prohledán, jsou vybrány všechny soubory s příponou rrd. Takto nalezené rrd databáze jsou následně projity, vybrány všechny datové zdroje (data source) a vygenerován příkaz pro rrdtool graph, který vrátí obrázek dle požadovaných parametrů (výška, šířka, titulek, popis os, typ).

```
/usr/bin/rrdtool graph - \
--imgformat=PNG \
--start=1239603845 \ --end=1242023045 \
--title="Celkovy provoz" \
--rigid \
--base=1000 \ --height=120 \ --width=600 \
--alt-autoscale-max \
--lower-limit=0 \
COMMENT:"13.04.2009 08\:04 - 11.05.2009 08\:05\c" \
--vertical-label="Toky/s" \
--font TITLE:12: \ --font AXIS:8: \ --font LEGEND:10: \ --font UNIT:8: \
DEF:a="/data/nfsen/profiles-stat/live/p3001.rrd":flows:AVERAGE
DEF:b="/data/nfsen/profiles-stat/live/p3000.rrd":flows:AVERAGE
CDEF:cdefa=a,1,* CDEF:cdefb=b,1,*
LINE:cdefa#000000:"p3001"
LINE:cdefb#FF0000:"p3000"
```

Ilustrace 10: Příkaz pro vygenerování grafu ze dvou databází s jedním data source

5.3.4 Nastavení

Tato část aplikace obsluhuje uživatelské a aplikační nastavení. Ne všechny části jsou dostupné všem uživatelům. Informace o uživateli a jeho reportech jsou ukládány do databáze. Jedinou výjimkou je heslo. Při změně se heslo změní v .htpasswd souboru pomocí programu htpasswd. Cesta k souboru .htpasswd je definována v konfiguračním souboru (více o autentizaci bude zmíněno níže).

Důležitou částí je sekce pro vytváření, editace a mazání vytvořených reportů. Report se skládá z více kapitol. Je umožněno definovat jejich pořadí v reportu. To se děje klikáním na šipky u kapitol (nahoru a dolů), případně je možné definovat pozici již při přidání kapitoly. Přesun spočívá v nastavení atributu pozice, tzn. pokud přesuneme kapitolu směrem nahoru, musíme přesunout i kapitolu nad ní o jednu pozici dolů. Je však třeba ohlídat i případy přidávání a mazání kapitol tak, aby nevznikaly stejné pozice v případě přidávání kapitol a sekvence pozic nebyla přerušena při mazání.

Jedna z dalších funkcionalit *Instalovat kapitolu* umožňuje přidání kapitoly nahráním speciálního souboru typu tar.gz. Tento archiv musí obsahovat skript install.sh a může obsahovat další potřebné soubory pro nainstalování kapitoly. Po nahrání archivu na server je provedena kontrola, zda se jedná opravdu o archiv (typ souboru *application/x-gzip*), ten je zkopírován do adresáře *./upload/*, rozbalen do dočasného adresáře */tmp/*, a poté je spuštěn instalační skript *install.sh*.

5.3.5 Nápověda

Jedná se o téměř statickou stránku, která pouze zobrazuje text. Zobrazuje také informaci o tom, která verze je právě nainstalována. Tato informace je důležitá při hlášení problému v aplikaci, aby bylo možné zjistit, v které verzi hledat problém, popřípadě nasadit novější verzi, v případě, že problém byl již dříve vyřešen.

5.4 Přihlašování

Přihlašování (autentizace) je řešena pomocí Basic Authentication obsaženém v protokolu HTTP. Toto řešení bylo zvoleno zejména pro možnost centrální autentizace v případě, že by se nástroj využíval ve spojení s dalšími nástroji v případě reálného nasazení.

Správa uživatelů tedy není součástí této práce, ale je možné ji kdykoliv, v případě potřeby, dodat jako zásuvný modul, který se nahraje do aplikace a povolí se jeho použití.

Nastavení ověřování je řešeno pomocí souboru *.htaccess* v hlavním adresáři programu. Pro odesílací skript *send_offline_report.php* není vyžadována žádná autentizace, ale je nastavena restrikce

allow from, která umožňuje přístup pouze z IP adresy 127.0.0.1 (localhost). Z důvodu snadného exportu není vyžadována autentizace ani pro skript *export_pdf.php*.

5.5 Lokalizace

Ze specifikace, která byla definována v druhé kapitole, vyplynula otázka, jak řešit lokalizaci aplikace a navrhnout co nejsnadnější funkční řešení tohoto problému. Již od začátku implementace bylo třeba dodržovat základní předpoklad, že žádný text nebude moci být vložen do aplikace „napevno".

Byly zvažovány dvě možnosti. První možností bylo použít nějaký standardní nástroj pro překlad. Takovýmto nástrojem je Gettext, který je dostupný i v PHP. Nebyl použit zejména z důvodu složitějšího překladu textů. Pro efektivní překlad je třeba instalovat další aplikace. Pro Windows například poEdit. Hodí se tedy spíše pro rozsáhlejší projekty.

Byla tedy využita druhá možnost přístupu k lokalizaci, která spočívá ve vytvoření vlastního jednoduchého systému. Základním požadavkem byla jednoduchost, spolehlivost a případné upozornění na požadavek o neexistujícím překladu.

Vznikla jednoduchá třída *Language*. Jazyky, které bude nabízet pro překlad, jsou definovány v atributech třídy:

- $aLang - seznam dostupných jazyků (cz, en,..)
- $sLang – pole konfiguračních hodnot jednotlivých jazyků

```
var $aLang = Array("en","cz"); // dostupné jazyky
var $sLang = Array(
  "en" => Array("ico" => "images/en.png", "name" => "English"),
  "cz" => Array("ico" => "images/cz.png", "name" => "Česky")
); // nastavení jednotlivých jazyků
```

Ilustrace 11: Definice dostupných jazyků

Přeložené texty se získávají pomocí metody *gt(string text_pro_preklad)*.

Jazykové soubory jsou uchovávány v adresáři *./lang/*. Pokud není dostupný překlad pro požadovaný jazyk, procházejí se postupně jazykové soubory v pořadí, v jakém jsou uvedeny v poli $aLang. V případě, že není překlad nalezen ani v nich, je vypsán nepřeložený řetězec.

Veškeré chyby se zaznamenávají a je tedy snadné odhalit nepřeložené řetězce. Podrobnější popis funkčnosti je obsažen v programové dokumentaci.

5.6 Offline reporty

Systém umožňuje definovat tzv. Offline reporty. Offline reportem se rozumí exportovaný report do PDF zaslaný uživateli emailem. V pravidelných intervalech se spouští skript *send_offline_report.php*, který z databáze vybere všechny reporty, které je třeba odeslat. Postupně jsou reporty zpracovány skripty *./backend/offline_top.sh* a *./backend/offline_traff.sh*.

Vytvoření PDF exportu probíhá úplně stejně jako ruční export z webového rozhraní. Skript se přihlásí jako daný uživatel, stáhne PDF soubor a uloží jej na disk k dalšímu zpracování (odeslání).

Tyto skripty mohou emaily přímo odesílat nebo generovat textový soubor, který bude zpracován jiným programem. Připraveny jsou obě varianty, stačí jen odkomentovat / zakomentovat preferovanou možnost.

5.7 Databáze

Pro práci s databází byla napsána třída, která má za úkol vytvářet mezivrstvu mezi databází a programovou částí pracující s databází.

Tento přístup je vhodný zejména z důvodu následného snadného změnění typu databáze. Nahráním jiné třídy, která zachovává rozhraní použité v aktuální verzi třídy, je tedy možné snadno přecházet například mezi PostgreSQL, MySQL a dalšími. Bezproblémový přechod však závisí na tom, že SQL příkazy budou ve všech databázích funkční. Bylo dbáno na to, aby tento požadavek byl dodržován a dotazy testovány právě na PostgreSQL a MySQL. V případě dalšího rozšiřování by bylo vhodné myslet na různou syntaxi v jednotlivých databázích a dodržet stále snadnou přenositelnost. Přehled rozdílů v různých databázích je například na [18] .

Další výhodou této mezivrstvy je centralizované místo, přes které se provádějí veškeré dotazy. V případě potřeby lze například logovat veškeré dotazy vykonávané nad databází jednoduchou úpravou této třídy a není nutné procházet a upravovat všechny zdrojové kódy. Tento přístup se osvědčil i při ladění a následném testování aplikace, kdy lze jednoduše zjistit, jaké dotazy se spouští a kde nastává problém. Tímto lze alespoň částečně získat funkcionalitu dostupnou v pokročilejších databázích a databázových nástrojích (např. Profiler z MS SQL,..).

6 Testování

Testování probíhalo vždy po dokončení určité části modulu tak, aby se na vytvořenou část dalo stoprocentně spolehnout při dalším vývoji. Dokončený modulu byl vždy testován jako celek. Tím se zabránilo mnohým chybám, které by vznikly rozkopírováním některých opakujících se částí kódu, kterých je ale minimum (problém byl vždy co nejlépe dekomponován tak, aby vzniklo jen potřebné minimum v kódu se opakujících částí). Vznikla tedy spousta funkcí, které řeší pouze dílčí problémy.

Při testování byly opraveny všechny nalezené chyby. V nepravidelných intervalech byly také vytvářeny „verze“ aplikace. První verze obsahovala pouze základní funkčnost. S každou další verzí byly odhalovány chyby a místa aplikace, která měla nějaký nedostatek. Případně byla aplikace rozšířena o funkcionalitu, která nebyla původně navržena, ale dělala systém stále použitelnějším v případě jeho budoucího reálného nasazení.

Ke konci vývoje se jednalo již spíše o chyby ve vzhledu a doladění chování aplikace v prohlížeči Internet Explorer. V průběhu vývoje byl používán zejména prohlížeč Firefox hlavně z důvodu existence výborného nástroje Firebug, který velkou měrou usnadňuje vývoj a hledání sémantických chyb (např. špatně vygenerovaný HTML kód).

Firma Microsoft uvolnila stabilní verzi Internet Exploreru verze 8. Bylo tedy vhodné otestovat funkčnost systému i pod touto novou verzí prohlížeče. Kromě drobných nedostatků, které byly snadno odstraněny, nebylo třeba nijak výrazně zasahovat do výsledného kódu.

7 Závěr

V této závěrečné kapitole jsou zhodnoceny dosažené výsledky, aktuální funkcionalita projektu a možnosti dalšího vývoje aplikace.

7.1 Dosažené výsledky

Vytvořená aplikace pokrývá všechny požadavky vytyčené ve specifikaci a tím je připravena k případnému reálnému nasazení. Celý vývoj od první analýzy požadavků do odladění systému trval devět měsíců. Systém běžel více jak polovinu této doby v síti a byl průběžně dolaďován. Postupně do něj byla také přidávána další potřebná funkcionalita pro efektivní práci uživatele.

Tato aplikace je jedna z mála, která umožňuje orientaci v monitorovaných hodnotách i méně zkušeným uživatelům a prezentuje výsledky měření velmi srozumitelnou formou. Exportované reporty jsou k nahlédnutí v přílohách 1 a 2.

Systém tedy splňuje nejen všechny body zadání práce, ale obsahuje i rozšíření nad jeho rámec.

7.2 Další vývoj

I když aplikace splňuje všechny požadavky dané specifikací, neznamená to, že je naprosto dokonalá, a že neexistuje žádná možnost jejího rozšíření.

7.2.1 Top reporty

Aktuální situace s předpočítáváním top reportů není zcela ideální. Tento nedostatek by mohl být vyřešen zapojením databáze do výpočtu požadovaných dat. Také právě proto již byl použit plnohodnotný SQL server i pro ukládání uživatelských hodnot, a nikoli jen XML soubory, které by zatím dostačovaly.

Tato myšlenka vznikla již ve fázi analýzy, ale potřeba takového řešení se ukázala jako značná až při prvních testech a práci s NfDumpem. Z důvodu potřeby vysokého výkonu a spolehlivosti byla zavržena databáze MySQL a použito výkonnější řešení ve formě PostgreSQL, který poskytuje i lepší funkcionalitu [19].

Tato změna by přinesla možnost uživatelsky definovaných intervalů a práce s daty, jako je nyní v Traffic reportech. Velmi pravděpodobně by byl použit stejný formulář pro výběr intervalu jako v Traffic reportech.

Pro ukládání dat do databáze z NfDumpu by musel vzniknout specializovaný program. Z důvodu efektivity by se zřejmě jednalo o klasický program napsaný nejspíše v C/C++.

7.2.2 Správa aplikace

Systém je rozšiřitelný pomocí modulů, v případě požadavků ho lze dále jednoduše rozšiřovat. S tím souvisí i instalace modulů. Mohla by probíhat pouhým nahráním speciálně vytvořeného instalačního balíčku, stejně jako je nyní umožněno instalovat nové kapitoly.

Instalaci všech prvků by bylo vhodné centralizovat – ze stejného místa by se instalovaly kapitoly, moduly i další případná rozšíření. Tato část aplikace by se mohla stát místem, které by mohlo oslabit bezpečnost systému. Z tohoto důvodu budou muset vzniknou další potřebná rozšíření, která zavádějí další bezpečnostní prvky. Možností je více, jmenujme například nutnost provedení analýzy obsahu instalačního balíčku a zavedení „podepisování“ instalačního balíčku.

7.2.3 Další moduly

V nynějším stavu umožňuje aplikace především monitorování objemu přenášených dat. Pro komplexnější monitorování by bylo vhodné doplnit aplikaci modulem, který by umožňoval detekci různých typů útoků v síti a jejich následnou evidenci.

Literatura

[1] Wikipedia: *NetFlow*. [online]. [cit. 2009-05-02].
URL <http://cs.wikipedia.org/wiki/Netflow>

[2] P. Čeleda, K. Bartoš, V. Krmíček, P. Minařík. Projekt CAMNEP - systém detekce průniku ve vysokorychlostních počítačových sítích. Zpravodaj ÚVT MU. ISSN 1212-0901, 2009, roč. XIX, č. 3,s. 3-8.

[3] Wikipedia: Network TAP. [online]. [cit. 2009-05-02].
URL <http://en.wikipedia.org/wiki/Network_tap>

[4] Cisco: *Catalyst Switch Port Analyzer (SPAN) Configuration Example.* [online].
[cit. 2009-05-02].
URL <http://www.cisco.com/application/pdf/paws/10570/41.pdf>

[5] Caligare: *Netflow :: Versions.* [online]. [cit. 2009-05-03].
URL <http://netflow.caligare.com/netflow_format.htm>

[6] RFC 3954: *Cisco Systems NetFlow Services Export Version 9*. [online]. [cit. 2009-05-03].
URL <http://www.ietf.org/rfc/rfc3954.txt>

[7] Matoušek Petr: *Není datům v síti těsno?*, In: CONNECT!, roč. 2005, č. 11, Brno, CZ, s. 7-8, ISSN 1211-3085.

[8] Dostálek Libor: *Velký průvodce protokoly TCP/IP: bezpečnost.* Computer Press, 2003, ISBN 80-7226-849-X.
Dostupný z WWW <http://www.cpress.cz/knihy/tcp-ip-bezp/CD-0x/5-12.gif>

[9] SNMP – Simple Network Management protocol. [online]. [cit. 2009-05-06].
URL <http://www.samuraj-cz.com/clanek/snmp-simple-network-management-protocol/>

[10] RFC 3917: *Requirements for IP Flow Information Export (IPFIX).* [online]. [cit. 2009-05-07].
URL <http://tools.ietf.org/html/rfc3917>

[11] Isaacs Scott: *Dynamické HTML*. Computer Press, 1998. ISBN 80-7226-083-9.

[12] PHP: History of PHP. [online]. [cit. 2009-04-29].
URL <http://cz2.php.net/manual/en/history.php.php>

[13] Dave Raggett, Jenny Lam, Ian Alexander and Michael Kmiec: *Raggett on HTML 4*. Addison Wesley Longman 1998. ISBN 0-201-17805-2.
Dostupný z WWW <http://www.w3.org/People/Raggett/book4/ch02.html>

[14] Wikipedia: *HTML*. [online]. [cit. 2009-04-27].
URL <http://en.wikipedia.org/wiki/HTML>

[15] Písek Slavoj: *JavaScript efektivní nástroj pro oživení WWW stránek.* Grada Publishing, Praha, 2001, ISBN 80-247-0014-X

[16] NFDUMP. [online]. [cit. 2009-04-20].
URL <http://nfdump.sourceforge.net/>

[17] Olšovský Marek: *PostgreSQL.* [online]. [cit. 2009-04-23].
URL <http://www.linuxsoft.cz/article_list.php?offset=0&id_kategory=222>

[18] Comparison of different SQL implementations. [online]. [cit. 2009-04-23].
URL <http://troels.arvin.dk/db/rdbms/>

[19] Comparsion of Oracle, MySQL and PostgreSQL DBMS. [online]. [cit. 2009-04-25]
URL <http://www-css.fnal.gov/dsg/external/freeware/mysql-vs-pgsql.html>

Seznam příloh