VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Monitorování síťového provozu a kontrola linuxového serveru

Bakalářská práce

2006 **Jozef Mlích**

Monitorování síťového provozu a kontrola linuxového serveru

Odevzdáno na Fakultě informačních technologií Vysokého učení technického v Brně
dne 27. dubna 2006

Prohlášení

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

. . . . . .
Jozef Mlích
27. dubna 2006

Abstrakt

Předmětem tohoto bakalářského projektu bylo seznámení se s principy komunikace v síti internet a úlohou serveru při této komunikaci. Práce zahrnuje popis technik pro sledování provozu a jednotlivé algoritmy pro řízení provozu.

Konkrétním cílem bylo vytvoření aplikace na platformě linux, která by umožnila monitorování a regulování síťového provozu pomocí programů iptables a tc.

Klíčová slova

sledovaní síťového provozu, řízení provozu, NAT, iptables, CBQ, HTB, tc, QoS, FUP

Poděkování

Tímto bych chtěl poděkovat mému vedoucímu Ing. Petrovi Weissovi za ochotu při konzultacích. Dále bych chtěl poděkovat správcům bezdrátové sítě v Dešné za poskytnutý hardware pro vývoj a testování programu. Nakonec bych chtěl poděkovat Markovi Obležarovi za testování a pomoc při hledání chyb v programu.

Abstract

The objective of this thesis was acquaint oneself with theory of network communication and role of server in this communication. This thesis includes description of practices of network monitoring and traffic shaping algorithms in detail.

More specificaly, the goal of this project was creating a linux application to monitor and manage of network traffic using programs iptables and tc.

Keywords

network traffic monitoring, traffic control, NAT, iptables, CBQ, HTB, tc, QoS, FUP

Obsah

Kapitola 1

Úvod

S rozvojem internetu, bezdrátových sítí a počítačových síti všeobecně vzrůstá potřeba tyto sítě monitorovat a spravovat. Tyto činnosti můžeme provádět buď na samotných stanicích nebo na síťových prvcích zajišťujících směřování.

1.1 Motivace

Nároky na monitorování a řízení síťového provozu jsou dané většinou snahou zaručit kvalitu služby pro jednotlivé uživatele a snahami zajistit bezpečnost a důvěrnost dat.

Mezi hlavní bezpečnostní rizika patří viry, trojské koně a hackeři. Primárním projevem takovýchto útoků je většinou poškození nebo odcizení dat. Druhotným projevem těchto útoků je specifický provoz na počítačové síti.

Potenciálně nebezpečné jsou hlavně přenosy velkých objemů dat směřované někam do internetu, velké množství otevřených spojení, dlouhodobé malé přenosy dat "když by počítač neměl nic dělat" a přenosy na atypických portech.

Statistické údaje získané sledováním systému mohou být hodnotným zdrojem dalších informací užitečných pro plánování, rozvoj a údržbu firemní počítačové sítě.

1.2 Cíle projektu

V kapitole 2 jsou shrunuty základní informace o počítačových sítích a databázích. Jsou zde popsané principy fungování firewallu a základy jeho nastavení v operačním systemu linux. Jsou zde také popsány algoritmy pro řízení provozu dostupné na linuxových serverech.

Kapitola 3 popisuje nejznámější linuxové nastroje pro monitorování stavu sítě.

Kapitola 4 shrunuje základní aspekty implementace vlastního monitorovacího nástroje. Je zde popsaný způsob sledování a správy sítě, způsob uložení dat a grafické rozhraní.

Kapitole 5 popisuje implementaci sady skriptů pro linuxový router zajištující směřování do vnější sítě na monitorování stavu sítě a automatické přizpůsobivé řízení šířky pásma pro klienty směřující packety do vnější sítě. Další požadovanou vlastností bylo zobrazovat a umožňovat nastavování základních parametrů jednolivých položek a to způsobem přijatelným pro uživatele.

V kapitole 6 jsou vyjmenované některé možnosti rozsíření programu.

Příloha A obsahuje ukázkový konfigurační soubor.

Příloha B popisuje provedená kontrolní měření šířky přenosového pásma a popis jakým způsobem byla tyto měření provedena.

Přiloha C obsahuje ukázky grafického rozhraní s popisem jednotlivých funkcí a vlastností.

Kapitola 2

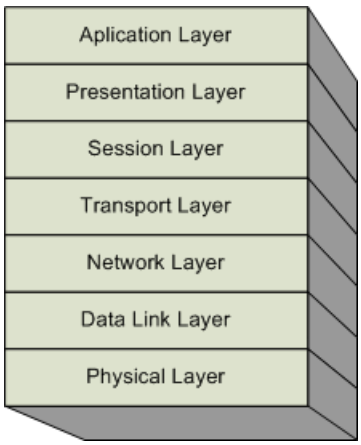
Teorie

2.1 Protokoly

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

2.1.1 Referenční model ISO/OSI

Referenční model OSI Open Systems Interconnection byl přijat jako standard mezinárodní organizací pro standartizaci ISO. Definuje 7 vrstev - viz. obrázek 2.1.



Obrázek 2.1: Referenční model ISO OSI

Fyzická vrstva

definuje napěťové úrovně signálu, konektory apod. Příkladem je protokol X.25

Linková vrstva

zabezpečuje bezchybný přenos bloků dat mezi dvěmi body počítačové sítě. Příkladem je protokol X.25 část LAPB

Síťová vrstva

zabezpečuje směřování packetů mezi více uzly sítě. Cílem této vrstvy je nalezení nejvhodnější cesty dat k cíli. Příkladem jsou protokoly CLNP a X.25 část PLP.

Transportní vrstva

zajišťuje rozdělení zprávy na jednotlivé packety a následně zpětné složení těchto packetu do původní zprávy ve správném pořadí. Příkladem jsou protokoly TP0, TP1, TP2, TP3, TP4, OSPF.

Relační vrstva

cílem této vrstvy je navazování a rušení spojení mezi koncovými účastníky. Příkladem jsou protokoly ISO 8327 a X.225

Prezentační vrstva

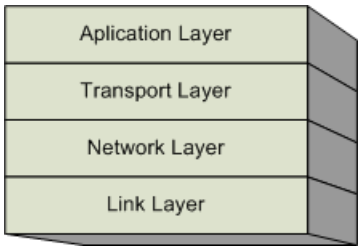
tato vrstva zajišťuje stejnou reprezentaci dat na různých platformách. V rámci této vrstvy může probíhat komprese a dekomprese dat. Jejím hlavním úkolem je řešení kompatibility. Příkladem jsou protokoly ISO 8823 a X.226

Aplikační vrstva

tato vrstva obvykle zajišťuje interakci s uživatelem. Příkladem jsou protokoly X.400, X.500, FTAM

2.1.2 Referenční model TCP/IP

V porovnání s modelem ISO/OSI je jednodušší. Jedná se v současné době o nejpoužívanější model.



Obrázek 2.2: Referenční model TCP/IP

Vrstva síťového rozhraní

plní funkce fyzické a linkové vrstvy modelu ISO/OSI . Příkladem jsou protokoly Ethernet, Wifi, Token Ring, PPP, SLIP, FDDI a ATM.

Síťová vrstva

odpovídá síťové vrstvě z modelu ISO/OSI. Jejím hlavním úkolem je tedy směřování packetů mezi více uzly sítě. Přikladem protokolů síťové vrstvy jsou protokoly IPv4, IPv6, ICMP, IGMP, ARP a RARP.

Transportní vrstva

zahrnuje transportní, relační a část prezentační vrstvy modelu ISO/OSI. Příkladem protokolů jsou protokol TCP a protokol UDP.

Aplikační vrstva

odpovídá aplikační vrstvě a části prezentační vrstvy z modelu ISO/OSI. Příklady používaných protokolů jsou DNS, FTP, TSL/SSL, IRC, NNTP, IMAP, SMTP, POP3, SIP, SSH, TELNET, BitTorent.

2.1.3 Protokol UDP

Je protokol transportní vrstvy, který je nespojovaný. Nezaručuje tedy doručení dat ve správném pořadí. Nezaručuje ani správnost dat. Pro svou jednoduchost se je méně výpočetně náročný. Je vhodný především pro služby, u kterých nevadí, že se sem tam ztratí nějaký packet. Příkladem takovýchto služeb je např. přenos videa v reálném čase.

2.1.4 Protokol TCP

Je protokol transportní vrstvy (viz. obrázek 2.2), který je spojovaný. Zaručuje doručení dat ve správném pořadí a správnost dat zabezpečuje kontrolním součtem. Je komplikovanější než protokol UDP. Používá se pro implementaci většiny služeb. Příkladem jsou HTTP, IMAP a SSH.

2.1.5 DHCP

Dynamic Host Configuration Protocol je protokol aplikační vrstvy (viz. obrázek 2.2), který pracuje na UDP portu 67. Přidělení nastavení probíhá ve čtyřech fázích:

1. klient pošle žádost o přidělení nastavení do sítě DHCPDISCOVER
2. server pošle odpověď obsahující nastavení DHCPOFFER
3. klient potvrdí příjmutí přidělených údajů DHCPREQUEST
4. server potvrdí přidělení nastavení DHCPACK nebo zamítne přidělení nastavení DHCPDECLINE
5. klient může žádat větší množství údajů, než mu bylo poskytuto při DHCPACK pomocí DHCPINFORM
6. klient může nahlásit serveru uvolnění ip adresy pomocí DHCPRELEASE

Všechny tyto data jsou posílaný všesměrově (pomocí broadcastů). Na serveru je uloženo nastavení, rozsahy ip adres, které je možné přídělit, seznam už přidělených adres a doba platnosti těchto výpůjček. Klient si ukládá současně s nastavením i dobu platnosti a v definovaných intervalech před vypršením platnosti několikrát žádá o prodloužení platnosti ip adresy a o nové nastavení.

Nastavení, které je klientovi přiděleno obvykle obsahuje ip adresu, masku sítě, výchozí bránu pro směřování a seznam DNS serverů. Seznam možných nastavení je možné nalézt v příslušných RFC nebo případně v dokumentaci k danému serveru. *Někteří vyrobci mají definováno vlastní nastavení* [9].

2.1.6 ARP

Address Resolution Protocol je protokol síťové vrstvy (viz. obrázek 2.2) slouží k překladu MAC adresy na IP adresu a nazpět. Podmínkou pro dostupnost MAC adresy je propojení zařízení na úrovni síťové vrstvy. MAC adresa neboli fyzická adresa je obvykle specifická pro daného výrobce a dané zařízení.

2.2 Služby routeru

V této podkapitole jsou popsané požadavky na router a jejich implementace v operačních systémech typu Linux. Hlavním požadavkem je směřování packetů z jedné sítě do druhé sítě.

2.2.1 Firewall

Firewall je aplikace, která je umístěna někde mezi dvěmi síťovými uzly. Primární funkcí firewallu je rozhodout, které packety mohou přejít z jednoho uzlu do druhého. Většina firewallu v současnosti pracuje na síťové a transportní úrovni, protože zpracování dat na aplikační úrovni je relativně výpočetně a programově náročné.

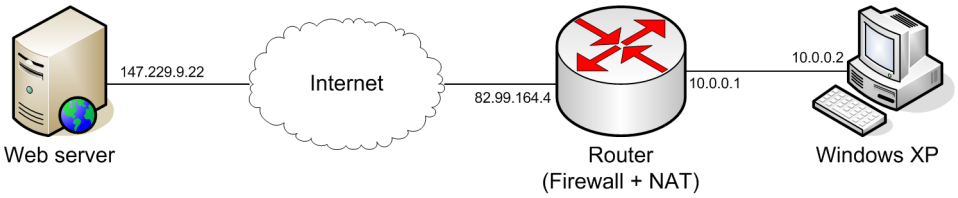
Rozlišujeme firewally packetové (bezstavové) a stavové. Příkladem bezstavového firewallu je ipchains, které bylo nahrazeno v novějších linuxech programem iptables. Stavový firewall oproti bezstavovému je schopen rozlišit jestli packety patří k nějakému spojení.

Při rozpoznávání jestli dané packety patří ke spojení může dojít k problémům a to v případě, že protokol pracuje na více portech jako je tomu například u protokolu FTP nebo PPTP (VPN).

V případě, že je počítač použit jako router, tak je obvykle součástí firewallu i NAT.

2.2.2 NAT

Network address translation je mechanizmus, který částečně řeší problém nedostatku ip adres. Celá privátní síť se z pohledu internetu jeví jako jedna ip adresa (viz. obrázek 2.3). Při pokusu prohlédnout si www stránku na počítači, který je v privátní síti je postup následující:



Obrázek 2.3: Schéma připojení privátní sítě přes NAT k internetu

1. Počítač v privátní síti pošle HTTP request s požadavkem na zobrazení stránky na svou výchozí bránu, což je v tomto případě síťový prvek označený jako router.

2. Router příjme packet z privátní sítě a uloží si zdrojovou ip adresu a číslo portu. V hlavičce TCP requestu přepíše ip adresu počítače z privátní sítě na svou vlastní adresu a pošle request WWW serveru

3. WWW server příjme request, zpracuje jej a odešle odpověď na Router - www server na základě TCP hlaviček vůbec netuší, že existuje nějaká privátní síť.

4. Router příjme odpoveď, podíva se do tabulky NAT a zjistí, že je určena pro počítač v privátní síti, upraví TCP hlavičku odpovědi tak, že nahradí svou vlastní veřejnou ip adresu za adresu počítače v privátní sítí. Nakonec pošle odpověd do privátní sítě.

Vytváření a rušení záznamů v tabulce spojení NATu je v případě TCP spojení dané vytvářením a rušením relace (SYN a FIN packety). V případě ztráty spojení je rušení záznamů zabezpečeno časovým razítkem.

2.2.3 Proxy

Proxy servery fungují prinicipiálně velmi podobně jako NAT. Rozdíl je v tom, že klient ve většině případů musí nastavit používání proxy serveru ručně. Proxy server si navíc vytváří tzv. cache. Do této cache si ukládá odpovědi serveru na jednotlivé requesty a v případě, že se opakují stejné požadavky, tak jsou použita data, která jsou uložena v této cache. Tato uspora linky do internetu s sebou přináší určité problémy jako je platnost dat a důvěrnost dat.

2.2.4 iptables

iptables je služba, která poskytuje v linuxových systémech firewall a NAT. Zpracování packetu je dané sadou pravidel, které definují co se stane s packetem. Ve výchozím nastavení obsahují iptables 3 tabulky: filter (výchozí), nat a mangle.

Tabulka filter je základní tabulkou pravidel, obsahuje řetězy (chainy) INPUT pro zpracování příchozích packetů, OUTPUT pro zpracovaní odchozích packetů a FORWARD pro zpracování packetů, které jsou průchozí. Tabulka nat se používá v případě, že se vytváří nové spojení obsahuje chainy PREROUTING, OUTPUT a POSTROUTING. Tabulka mangle je určena pro specifické úpravy packetů. Pravidla se na packet aplikují postupně shora dolů. Pořadí zpracování chainů je na obrázku 2.4.

Jednotlivé pravidla umožnují propouštění, blokování, zahazování, značkování a úpravy hlaviček packetu na základě typu, hlaviček a stavu spojení. Úpravy hlaviček jsou většinou spojené se záměnou veřejné ip adresy za privátní a naopak (viz. kapitola 2.2.2).

Značkování packetů umožnuje řízení provozu a tvarování toků dat (angl. *traffic shaping*). Tato problematika je popsána v následující kapitole (2.3).

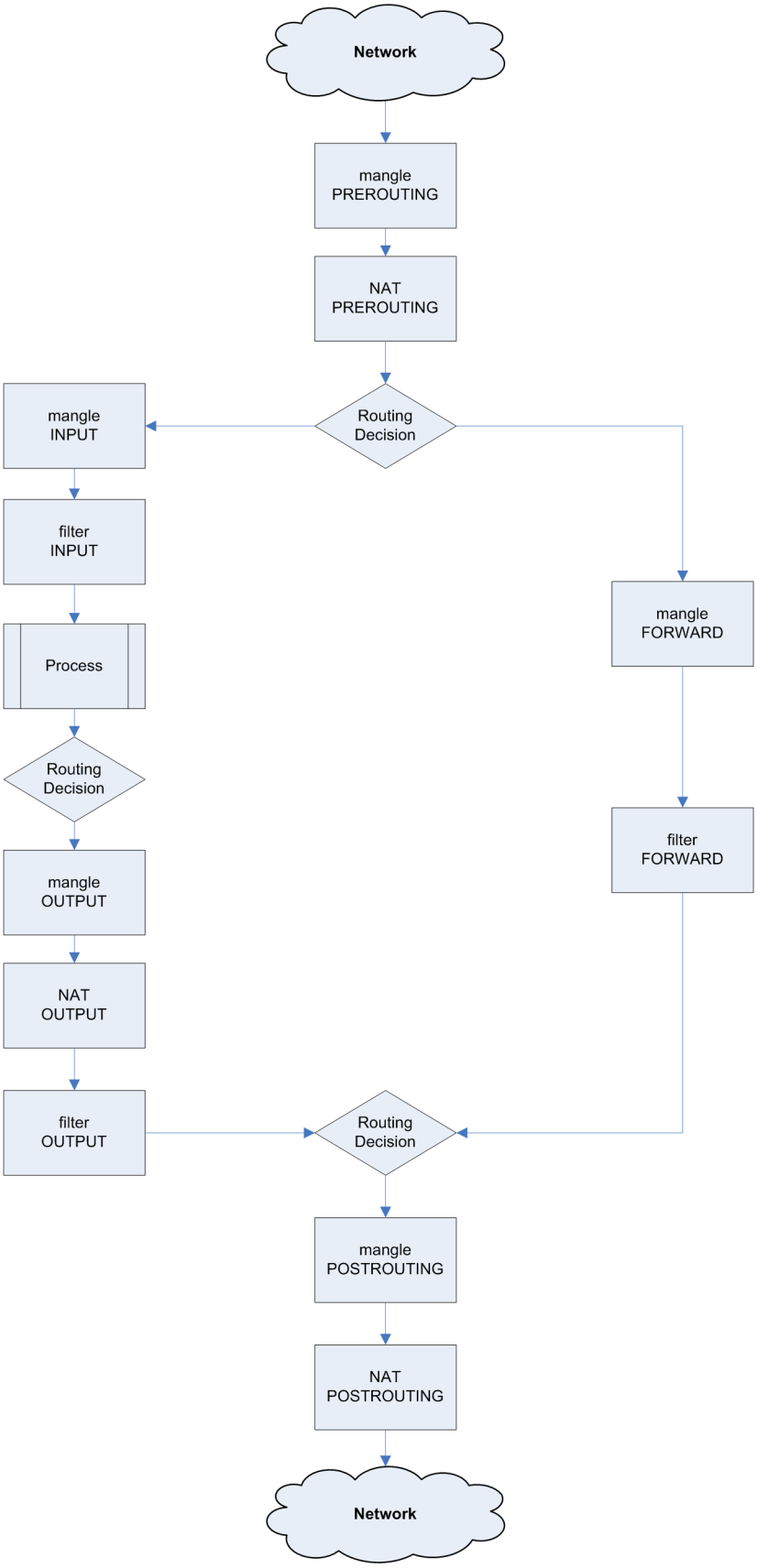
2.3 Kvalita služby

QoS neboli Quality of Service je výraz shrunující požadavky na službu. Tyto parametry by měli být definovány už na Síťové vrstvě (viz obrázek 2.2). Bohužel protokol IP definuje pouze parametr TOS[1], který mnohdy neposkytuje dostatečný prostor pro nastavování rychlosti doručení packetu. Pro obcházení těchto vlastností se používá některých vlastností protokolu TCP jako je spolehlivost doručení. Tyto vlastnosti jsou implementovány v operačních systémech typu linux pomocí programu '`tc`', k používání je nutné zapnout ještě podporu v jadre (`linux kernel configuration - networking - networking options - QoS and/or fair queing`)

2.3.1 Šířka pásma

Šířka pásma je definovaná jako množství přenesených data za jednotku času. Je kritická především pro multimediální aplikace.

[1] Type Of Service



Obrázek 2.4: Průchod packetu přes iptables

2.3.2 Doba odezvy

Doba odezvy angl. latence je definovaná jako doba mezi přijetím a odesláním packetu. Jedná se o zpoždění na výstupním zařízení a zpoždění na přenosové cestě. Tento ukazatel je kritický především pro interaktivní aplikace. Při přenosu online televizního vysílaní nevadí zpoždění několika sekund, ale při přenosu VoIP služeb (tedy telefonování přes internet) je několikasekundové zpoždění citelné.

2.3.3 Změna doby odezvy

Změna doby odezvy (angl. *jitter*) je kritická v případě, kdy je kladen důraz na malou dobu odezvy a packety jsou doručeny ve špatném pořadí.

2.3.4 tc

`tc` je nástroj pro správu front především na výstupním zařízení v linuxových systémech. Je součástí balíčku iproute2. Tímto nástrojem je možné pomocí skupiny front (qdisc) a tříd (class) vytvořit strom, který definuje pořadí a rychlost odesílaní dat. Pomocí filtru se následně nastaví, do které fronty se mají dané packety zařadit.

Prostřednictvím skupin front (qdisc) umožnuje nastavit způsob řazení dat na výstupním zařízení. Typy front jsou dvojího druhu: buď s nebo bez tříd (classfull nebo classless). Třídy umožnují definovat dílčí parametry. Každá třída může mít buď právě jeden qdisc nebo několik podtříd.

Výběr fronty je daný filtrem. Běžně se používá filtr `u32`, který vybírá frontu podle hodnoty v hlavičce packetu. V případě, že je součástí routeru NAT[2], tak dochází ke změnám těchto hlaviček proto není možné tuto techniku použít. Dalším používaným filtrem je `fw`, který vybírá frontu podle značky `MARK` přidělené na firewallu. Filtr `tc_index` je určený pro DiffServ a testuje hodnotu položky TOS[3] záhlaví packetu. Filtr `route` vybírá frontu podle cílové sítě.

CBQ

Class Based Queuing umožnuje vytváření a manipulaci se stromem tříd. Pro každou třídu lze nastavit omezení rychlosti nebo upřednostnění používaní nekteré ze tříd. Omezování rychlosti je založeno na výpočtu nevyužité šířky přenosové linky a průměrné velikosti packetu.

HTB

Hierarchical Token Bucket nabízí podobné vlastnosti jako CBQ. Je založeno na algoritmu TBF. HTB má oproti CBQ jednodušší syntaxi. Umožnuje definovat jaká je garantovaná rychlost a jaká je maximální sdílená rychlost.

FIFO

First in first out je výchozí typ fronty používané v Linuxu. FIFO fronta nemůže obsahovat třídy.

TBF

Token Bucket Filter je fronta určená pro snižování rychlosti přenosové linky. Lze jí použít i pro velké objemy dat. TBF fronta nemůže obsahovat třídy.

[2] Network Address Translation
[3] Type Of Service

RED

Random Early Detection je fronta určená pro snižování rychlosti. Algoritmus se snaží inteligentně zabránit zaplnění fronty. Cílem je zmenšit velikost front což je vhodné pro interaktivní služby. RED fronta nemůže obsahovat třídy.

SFQ

Stochastic Fairness Queuing nereguluje rychlost odesílaní a šířku pásma, ale plánuje dobu odeslání packetu. Cílem je zaručit rovnoměrné poskytnutí pásma pro různé služby. SFQ fronta nemůže obsahovat třídy.

2.4 Databáze

Prakticky všechny data o stavu sítě lze získat pasivním sledováním provozu. Při monitorování provozu rozdělíme informace na dva druhy: statistické údaje množství přenesených dat za jednotku času, tedy pouze rámcový přehled o komunikaci na síti a kvalitativní údaje, které jsou rozsáhlejší a poskytují detailnější přehled o jednotlivých spojeních. Vzhledem k tomu, že kvalitativní data jsou rozsáhlá, tak z dlouhodobého hlediska není únosné je uchovávat.

2.4.1 UML

Unified Modeling Language je jazyk určený na modelování a specifikaci objektů používaný v softwarovém inženýrství. Jazyk UML byl oficiálně definovaný konzociem OMG [10]. Existuje přímá korespondence mezi návrhem datových struktur v jazyce UML a implementací těchto datových struktur pomocí databáze a jazyku SQL[4].

2.4.2 Relace

Buďte A, B množiny a nechť $\rho \subseteq A \times B$. Pak ρ se nazývá *binární relace mezi množinami* A, B. Je–li speciálně $A = B$, nazývá se ρ *binární relací na množině* A.

ρ je *binární relace mezi množinami* $A, B \iff \rho \subseteq A \times B$

2.4.3 Relační databáze

Při uchovávání a spravování dat je kladeno na data několik požadavků jako je perzistence dat, sdílený, bezpečný, rychlý, jednoduchý, abstraktní a integrovaný způsob přístup k datům. Relační databázové systémy nám tyto vlastnosti většinou poskytují. Tato úroveň rozhodování se nazývá Online Transaction Processing.

Relační databázový systém je založený na relačním modelu a relační algebře. Data jsou uspořádána do relací, nad kterými jsou definovány přípustné operace. Na ovládání relačních databází se používa strukturovaný dotazovací jazyk SQL.

2.4.4 Porovnání databází

Komeční databázové systémy jako je Oracle a Microsoft SQL poskytují velmi širokou funkcionalitu. Jsou určené především pro velké korporace a složitější aplikace.

[4] Structured Query Language

PostgreSQL poskytuje velmi dobrou funkcionalitu včetně uložených procedur, triggerů, vnořených SQL dotazů a kontroly integrity dat na základě cizích klíčů. Tento systém je vhodný především pro středně velké aplikace.

Firebird je šířený pod IDPL licencí. Jeho vlastnosti jsou srovnatelné s PostgreSQL. Proti PostgreSQL má navíc *embeded* verzi, takže je možné tento server přibalit k aplikaci.

SQLite je velmi jednoduchý databázový systém, který neumožnuje ani uložené procedury ani vnořené SQL dotazy. Podpora JOIN příkazů není úplná. Vyhodou je, že je tento systém je šířený pod public domain licencí, je velmi malý a je možné bez problémů přiložit tento systém k aplikaci.

MySQL je databázový server šířený pod dvojí licencí. Pro open source použití pod GPL licencí, pro komerční využití pod OEM Commercial Licence. MySQL ver. 3.1 se vyznačovala velmi rychlým a jednoduchým přístupem k datům. Podpora vnořených dotazů a kontroly integritních omezení byla doplněna ve verzi 4.1. Podpora uložených procedur a triggerů byla doplněna ve verzi 5.0. MySQL je velmi rozšířené a ma velmi dobrou podporu ve skriptovacím jazyku PHP.

2.4.5 Datové skladiště

Datové skladiště jsou určena pro zpracovávání velkých objemů dat. Předzpracovná data se ukládají tak, aby se při jejich zobrazování ušetřil čas. Úplonost dat nemusí být v porovnání s rychlostí zpracování až tak důležitá. Tato úroveň rozhodování se názývá OLAP Online Analytical Processing.

Získávání dat z datových skladišť se nazývá *data mining*.

Produkty Microsoft Analysis Services, IBM DB2, Palo, Mondrian implementují technologie datových skladišť a OLAP.

Kapitola 3

Nástroje na monitorování sítě

Před tím než začneme implementovat vlastní nástroje pro analýzu provozu na síti, tak je vhodné seznamit se s ostatními veřejně dostupnými nástroji. Většinu informací uvedených v této kapitole jsem čerpal z [2].

3.1 MRTG - Multi Router Traffic Grapher

MRTG je multiplatformní nástroj, který je schopen vytvářet velmi komplexní grafy. Umožnuje sledovat různé ukazatele na serveru. Primárně je určený k sledování datových toků na jednotlivých síťových zařízeních na serveru. Za pomocí SNMP může vytahovat data z inteligentních routerů a switchů. Tento nástroj je distribuovaný pod licencí GNU GPL.

3.2 IPTraf

iptraf je konzolová interaktivní utilita na sledování aktuálního provozu na routeru. Poskytuje detailní informace o jednotlivých spojeních i statitstické údaje všech spojeních. Neposkytuje už ale informace o obsahu těchto packetů. Tento nástroj je distribuovaný pod licencí GNU GPL.

3.3 NetSNMP

Netsnmp je sada konzolových nástrojů, které umí vytvářet a zpracovávat SNMP packety. Tyto nástroje jsou neinteraktivní a jsou určený dalšímu dávkovému zpracování. Tyto nástroje jsou distribuované pod BSD-like licencí.

3.4 TCPDump

tcpdump je konzolový nástroj, který umožnuje umožnuje zobrazaní a uložení packetů přenášených přes server, včetně obsahu těchto packetů. Tento nástroj je distribuovaný pod BSD licencí.

3.5 Ethereal

Ethereal je grafická aplikace, která umožnuje zobrazení, uložení a analýzu packetů přenášených přes server. Pod pojmem analýza dat se myslí jednak statistické vyhodnocení dat, ale i zobrazení celých TCP streamů. Tento program je distribuován pod GNU GPL licencí.

3.6 Webalizer

Webalizer je konzolový program pro zpracovávání logů z webových a ftp serverů. Z těchto logů vytváří statistické údaje v podobě www stránek (html a png). Tento program je distribuovaný pod licencí GNU GPL.

3.7 Awstats

awstats je konzolový program na zpracování logů z webových, ftp a poštovních serverů. Z těchto logů vytváří statistické údaje v podobě www stránek (html a png). Tento program je distribuovaný pod licencí GNU GPL.

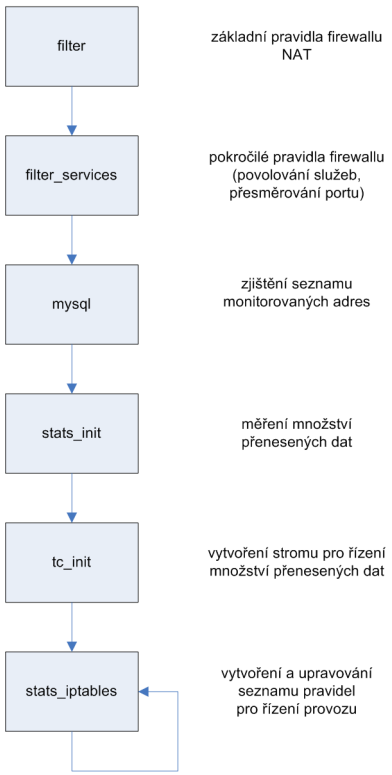
3.8 NMAP

NMAP je multiplatformní nástroj na aktivní i pasivní skenování sítě. Umožnuje vytvořit seznam otevřených portů na specifikovaném počítači a běžně používaných služeb umožnuje identifikovat jaká služba popřípadě verze na tomto portu běží. Na základě těchto služeb je schopen identifikovat operační systém, případně jeho verzi. Dále umožnuje rozesílat paralelně ICMP packety (ping), tedy zjištovat jestli jsou stanice online v relativně krátkém čase. Tento program je distribuovaný pod licencí GNU GPL.

Kapitola 4

Implementace

Implementace lze rozdělit do tří částí: inicializaci systému po startu, měření a zaznamenávání dat a zobrazování dat. Pořadí jednotlivých činností potřebných pro měření dat přenesených přes server a řízení šířky pásma je naznačené na obrázku 4.1. Zaznamenávání dat a řízení provozu je prováděno buď pomocí plánováče úloh `cron` nebo v nekonečné smyčce. Zobrazování dat se provádí v separátním procesu.

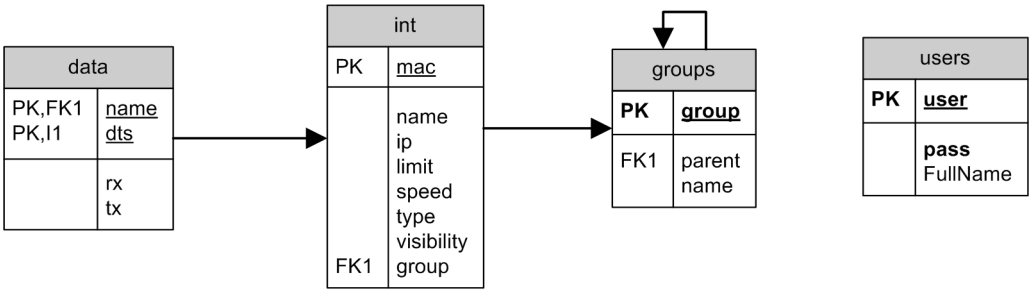


Obrázek 4.1: měření a řízení trafficu pomocí iptables

Nastavení jednolivých parametrů potřebných pro sběr dat a řízení provozu se provádí buď v konfiguračním souboru '`/etc/stats.conf`' nebo v databázi. Parametry grafického rozhraní se nastavují v samostatném konfiguračním souboru. Příklad konfiguračního souboru je v příloze A.

4.1 Datové struktury

Použité datové struktury jsou znazorněné v Entity Relationship Diagramu 4.2. Tabulky *data* a *int* vyzniklé na základě příslušných entit jsou plněny dávkově jednotlivými skripty. Správce může v tabulce *int* ručně nastavovat některé vlastnosti. Tabulka *groups* je určena pro logické rozčlenění jednotlivých uživatelů a zařízení. Tabulka *users* je určená na uchovávání seznamu správců je jejich přístupových hesel.



Obrázek 4.2: ER-Diagram

4.2 Sběr dat

4.2.1 netstat

Modul netstat implementuje získávání a zobrazení detailních informací o jednotlivých tcp spojeních. Data o jednotlivých spojeních jsou k dispozici v '`/proc/net/ip_conntrack`'. Data se zpracují pomocí regulárního výrazu a zobrazí se naformátovaný výsledek. Nad tabulku s jednolivými spojeními lze vytvořit jednoduchý filtr podle jednotlivých stanic a tcp portů. K jednotlivým TCP portům jsou přiřazeny jména služeb podle souboru '`/etc/services`'. Tímto způsobem bohužel nelze identifikovat služby, které používají dynamické čísla portů jako je například DC++ nebo Skype. Není také vidět množství přenesených dat vztahující k těmto spojením.

4.2.2 stats_dev

Modul shromažďuje informace o množství přenesených dat na jednotlivých zařízeních. Zdrojem dat je soubor '`/proc/net/dev`' jenž se zpracovává shellovým skriptem. Tento shellový skript spouštíme pravidelně v intervalu 5 minut pomocí cronu.

4.2.3 stats_iptables

Jak název napovídá, tak si modul zobrazí nějakou tabulku iptables a přečte z ní údaje o množství přenesených dat. Z tohoto důvodu jsou při startu routeru vytvořeny tabulky `STATS-IN` a `STATS-OUT` a do nich jsou zařazeny jednotlivé ip adresy, které se mají monitorovat.

K ip adresám je na přiřazena MAC adresa zařízení a naměřené údaje se zapíší do databáze. Přiřazování fyzických adres k ip adresám probíhá na základě informací z DHCP serveru, tedy souboru '`/var/state/dhcpd.leases`' a ve druhé fázy na základě ARP tabulky, kterou operační systém linux zpřístupňuje pomocí souboru '`/proc/net/arp`'. Tento soubor obsahuje i magické konstanty, jejichž význam lze najít v souboru '`/usr/include/net/if_arp.h`'.

Seznam sledovaných ip adres je na základě DHCP a ARP tabulek rozšířený o nové stanice.

Nakonec se na základě dat v databázi vytvoří pravidla pro značkování packetů a řízení přidělování šířky pásma.

4.2.4 stats_apache

Tento modul řeší různé problémy spojené s analýzou provozu webového serveru. Data jsou získávané z logů webového serveru. Prvním problémem je možnost nastavení velkého množství formátů logu. Nejjednodušším řešením by bylo v nastavení webového serveru zvolit nějaký jednoduchý formát logu, který obsahuje pouze základní informace. Problémem tohoto řešení je ztráta velkého množství informací.

V případě, že nechceme přijít o tyto informace, tak můžeme buď používat jako zdroj dat přímo tento log a generovat z něj přímo reporty obsahující statistické údaje o používaných prohlížečích, nejčastější stahovaných souborech, apod. Další možnosti by bylo kopírovat data do nějaké databáze a reporty generovat až z databáze. Dalším problémem je způsob ukládání předgenerovaných dat.

Na webserverech je obvyklé používání tzv. virtuálních domén. Nejjednodušším způsobem jak odlišit data jednotlivých virtuálních domén je nastavením vytváření samostaných logů pro každou doménu a jejich samostatné zpracování.

4.2.5 stats_qmail

Tento modul zpracovává logy poštovního serveru qmail. Obdobně jako v případě webového serveru je vhodné rozlišovat data z několika virtuálních domén.

4.3 Řízení provozu

V případě poloautomatického adaptivního řízení je nastavování parametrů přenosových linek na serveru úzce spjaté se sběrem dat. Úprava parametrů přenosových linek většinou probíhá těsně po vyhodnocení posledního stavu.

Pro nastavování parametrů přenosových linek se používá na linuxových systémech program 'tc' v kombinaci s příslušnými moduly jádra.

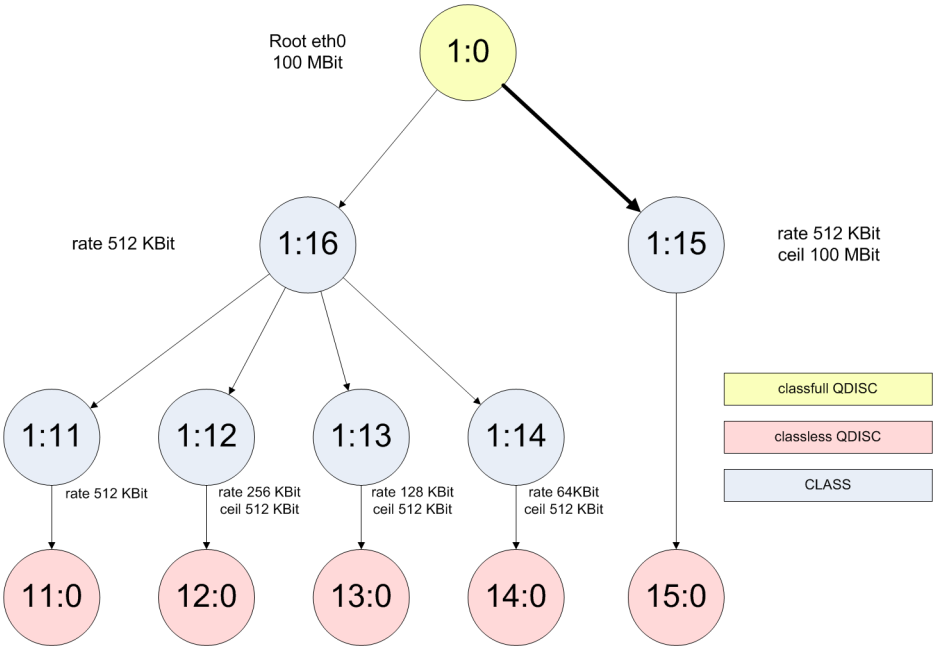
Pro každé zařízení se vytvoří strom přidělování pásma viz. obr. 4.3. Jako výchozí způsob řazení do front se nastaví algoritmus HTB. Provoz se rozdělí na data přenášená po místní síti (třída 1:16) a data přenášená do internetu (třída 1:15). Provoz směřovaný do internetu se rozdělí na 4 rychlostní skupiny. Pro každou rychlostní skupinu se vytvoří fronta, která zajistí rovnoměrné přidělení pásma pro jednotlivé služby v dané rychlostní skupině. Tustá čára na obrázku znázorňuje výchozí (neznačkovanou) skupinu.

Značkování dat pro zařazení do jednotlivých rychlostních skupin se provádí pomocí iptables v tabulce `mangle` v řetězu `FORWARD`. Měření množství přenesených dat se provádí v tabulce `fitler` v řetězu `FORWARD`.

Změny pravidel pro značkování jsou prováděny nezávisle na změnách pravidel pro měření množství přenesených dat, proto nedochází při změně začkování ke ztrátě informace o množství přenesených dat.

4.4 Grafické rozhraní

Pro interakci s uživatelem je možné zvolit libovolný internetový prohlížeč jako tenkého klienta (viz. příloha C). Všechny data se zobrazují jako dynamicky generované webové stránky a obrázky. Uživatel tedy může pracovat s grafickým rozhraním vzdáleně na libovoné platformě.



Obrázek 4.3: Strom pravidel pro řízení provozu na přenosové lince

Grafické rozhraní je rozčleněno na část pro správce a část pro uživatele. Přičemž uživatelská část systému je přístupná bez přihlášení a zobrazí pouze statistiky patřící k ip adrese, ze které uživatel přistupuje. Část systému určená pro správce umožnuje zobrazit si data o libovolném sledovaném ukazateli a měnit parametry jednotlivých položek.

Je možné nechat si zobrazit data zpětně v definovaném časovém období.

Aby se eliminovaly opakující se SQL dotazy na zjištění statistických dat, tak se zjištěné mezivýsledky ukládájí do sessions. Při zobrazování dat za jednotlivá časová období se pro vykreslení grafu i tabulky používá minimální počet spojení s databází.

4.5 Zabezpečení hesel

Přihlašovací údaje se posílají pouze jednou, při další komunikaci se posíla jenom hash identifikující dané spojeni. V případě, že webový prohlížeč umožnuje skriptování typu javascript, tak se místo hesla posíla pouze md5 hash.

4.6 Vykreslování grafů

V případě, že se mají zobrazit data pouze o jedné stanici, tak se graf vykreslí jako histogram pomocí html tabulky a obrázků, které mají výšku nastavenou tak, aby odpovídala množství přenesených dat v daném časovém úseku. Místo y-ové osy se údaj o množství přenesených dat zobrazuje na událost vyvolanou ukázáním kurzorem myši na požadovaný údaj.

V případě, že se mají zobrazit data o více stanicích, tak se graf vyreslí jako spojicový x,y-graf pomocí knihovny gd2.

Knihovna gd2 umožnuje vykreslení pouze základních tvarů jako jsou čáry, obdelníky, kruhy, vykreslení textu, ale neobsahuje žádné komplexní mechanismi pro vyreslení grafů. Bylo proto nutné implementovat třídu poskytující potřebnou míru abstrakce.

Kapitola 5

Možnosti dalšího vývoje

5.1 Zabezpečení

Jedna poučka říká, že žádné zabezpečení není dostatečné. Zabezpečení rozhraní pro správu lze posílit, tak že se spustí na serveru podporující https. Bylo by možné vytvořit speciální konzoli pro správu a tlustého klienta, kde by se autorizační a autentizační data přenášeli např. pomocí technologie SAML. V tomto směru by bylo možné přidat podporu ověřování hesla vůči kerberosu nebo ldap.

V části sběru statistik je možné zlepšit zabezpečení, tak že se oddělí části přistupující k jádru a musí běžet pod uživatelem root od částí které nevyžadují tohoto uživatele. Dále by v konfiguračních souborech nemělo být heslo pro připojení k databázi uložene jako prostý text.

5.2 Přidávání dalších modulů

Je možné přidávat další moduly monitorující stavy routerů, zatížění procesoru, obsazenost paměti, obsazenost disků a různých služeb jako např. sendmail, postgreSQL, rc5 proxy, samby, různé druhy FTP serverů.

5.3 Analýza dat na aplikační úrovni

Tato analýza by měla zahrnovat rozeznávání služeb, které si dynamický volí port na kterém běží. Příkladem takovýchto služeb je třeba Skype, BitTorent nebo Direct Connect. Dalším přínosem je kontrola přenášených dat ve zmíněných protokolech i v ostatních běžných protokolech jako je http, sntp nebo imap.

5.4 Sledování provozu 'Živě'

Stávající podoba programu umožnuje pouze dlouhodobé sledování vybraných parametrů. Toto vylepšení by mělo zahrnovat možnost sledování jednotlivých TCP spojení a množství přenesených dat z jednotlivých ip adres v krátkých intervalech (každé 3 sekundy) poskytovat informace o změnách.

5.5 Měnit pravidla v závilosti na denní době

Tato změna má zlepšit flexibilitu sítě. Běžný uživatel pracuje přes den, stahování větších objemů dat může probíhat v noci. Tímto způsobem tedy můžeme notoricky stahující uživatele odsunout do nočních hodin.

5.6 Proxy

Sledování dat procházející přes proxy přináší určité problémy. Pochopitelně je potřeba nastavit logování a zpracovávat data z logu. Je potřeba pro každý jednotlivý request určit zdrojovou ip adresu. Vzhledem k tomu, že zobrazovaný interval je 1 hodina a je obyvklé, že vzniká řádově tisíce záznamů za hodinu, tak je nutné tyto záznamy agregovat. Dalším problémem je, že data jsou požadována různými počítači, ale ke stažení těchto dat na server dochází jenom jednou. Není tedy jasné kterému klientovi a kolik přenesených dat započítat.

5.7 Hodnocení důvěryhodnosti

Systém by mohl implementovat hodnocení důvěrihodnosti stanic v intranetu i počítačů na internetu v závislosti na datech získaných měřením. Podobně jako je to v projektu SECURE (viz. [5]).

Kapitola 6

Závěr

Vytvořil jsem sadu BASH skriptů a PHP skriptů pro sledování, analýzu a řízení provozu dat procházejících serverem. Na sledování provozu jsem použil iptables podle článků [6] a [1]. Pro řízení provozu jsem použil program tc. Teoretický podklad mi poskytla diplomová práce [4] a články [8], [3]. Praktické znalosti jsem čerpal ze seriálu [7].

Data jsem ukládal do databáze MySQL a zobrazoval pomocí PHP a knihovny gd2. Databáze MySQL byla zvolena kvůli dobré podpoře v PHP, dobré dostupnosti a licenci.

Kontrola správnosti nastavení byla provedena pomocí několika měření. Výsledky jednotlivých měření jsou v příloze B.

Možnosti dalšího vývoje jsou popsané v kapitole 5.

Literatura

[1] Oskar Andreasson. Iptables tutorial 1.2.0. http://iptables-tutorial.frozentux.net/iptables-tutorial.html.

[2] James M. Kretchmar. *Administrace a diagnostika sítí*. Computer Press, Brno, 2004.

[3] Otakar Lávička. Klasifikace provozu sítě. http://www.fi.muni.cz/~kas/p090/referaty/2005-podzim/st/qos-xlavick1.html, 2005.

[4] Jakub Mácha. Kontrola síťového provozu. Master's thesis, Masarykova Univerzita, fakulta informatiky, 2000.

[5] Marián Miško. Firewall s reputačním systémem. Master's thesis, VUT BRNO, Fakulta Informačních Techonologií, 2005.

[6] Miroslav Petříček. Stavíme firewall. http://www.root.cz/serialy/stavime-firewall/, 2001.

[7] Radek Podgorný. Seriál o htb. http://www.root.cz/clanky/htb-jemny-uvod/, 2003.

[8] Vladimír Smotlacha. Technická zpráva cesnetu - qos v linuxu. http://www.cesnet.cz/doc/techzpravy/2001/20/, 2001.

[9] www stránky. Wikipedia - dhcp. http://en.wikipedia.org/wiki/DHCP.

[10] www stránky. Wikipedia - uml. http://en.wikipedia.org/wiki/Unified_Modeling_Language.

Dodatek A

Konfigurační soubor

```
#!/bin/sh
#
# konfiguracni soubor
#

mysql="/server/mysql5/bin/mysql"
tc="/sbin/tc"

sysdata="/proc/net/dev"
dhcpleases="/var/state/dhcp/dhcpd.leases"
arp="/proc/net/arp"
base="/var/tmp/stats"
webdir="/server/apache2/htdocs/stats"

DBsocket="localhost:/tmp/mysql.sock"
DBuser="root"
DBpass=""
DBname="stats"

# dev_inet="eth1"
# dev_local="eth0 eth2"

# dev_inet = (interface InterfaceSpeed Down1 Down2 Down3 Down4)

# dev_local[0..n] = (interface InterfaceSpeed Down1 Down2 Down3 Down4)

dev_inet="eth1 100mbit 512kbit 256kbit 128kbit 64kbit"
dev_local[0]="eth0 10mbit 2mbit 1mbit 512kbit 128kbit"
#dev_local[1]="eth2 11mbit 2mbit 1mbit 500kbit 100kbit"

dev_local_trust=true
```

Dodatek B

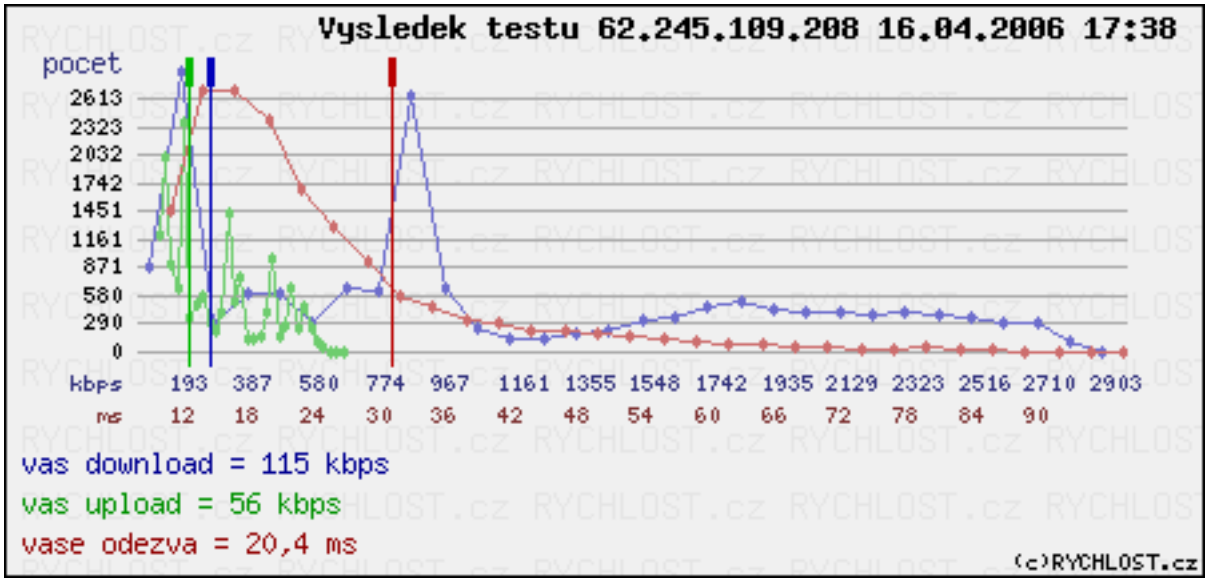
Měření rychlosti

B.1 Pomocí internetového serveru

Graf byl pořízený na serveru http://www.rychlost.cz/

Podmínky měření (4. rychlostní stupeň, bežný provoz):

1. předpokládaná rychlost downloadu: 128 KBit/s
2. předpokládaná rychlost uploadu: 64 KBit/s



Obrázek B.1: měření na serveru rychlost.cz

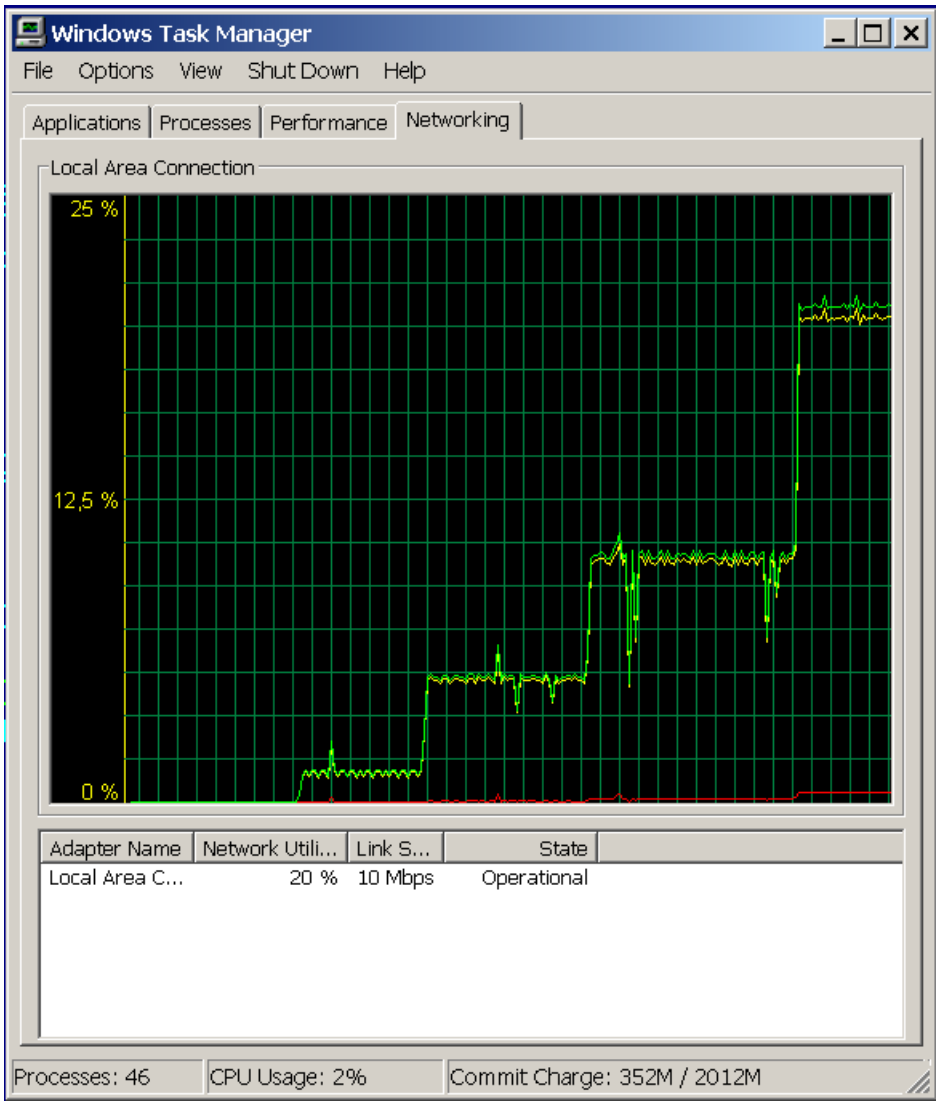
Předpokládaná rychlost se liší od naměřené cca o 10%. Odychlka je způsobená pravděpodobně dalším zatížením linky jinou aplikací. Dalším vyrazným faktorem, který mohl ovlivnit měření je poměr velikosti jednoho rámce vzlhedem nastavené šířce pásma.

Použitím algoritmu RED by pravděpodobně možné dosáhnout hladšího a přesnějšího průběhu. Algoritmy SFQ a TBF neberou ohled na způsob výpočtu plánování odesílání packetu.

B.2 Pomocí správce úloh

Graf byl pořízený na *Windows XP SP2 Eng* pomocí programu správce úloh '`taskmgr`'
Podmínky měření:

1. přenášená data: video ve formátu AVI, DivX
2. přenosový protokol: HTTP
3. měřící nástroje: Apache 1.3.33 + wget 1.9.1
4. předpokládaná rychlost downloadu 128 KBit/s, 512 KBit/s, 1 MBit/s a 2 MBit/s



Obrázek B.2: zatížení linky podle správce úloh

Graf zachycuje zatížení linky při testování jednotlivých rychlostních kategorií. Červená čára znázorňuje množství odesílaných dat, žlutá čára znázorňuje množství příjmutých dat a zelená čára znázorňuje celkové zatížení přenosové linky.

Z grafu je vidět, že při změně rychlostní kategorie nedochází v době změny značkování ani k úplnému zastavení přenosu ani k nekontrolovanému přenosu.

Drobnější výkyvy rychlosti "kostrbatost" průběhu je dána vlastnostmi protokolů Ethernet, IP a TCP. Malé výkyvy nad definovaný limit jsou dané změnou rychlostní kategorie přes webové rozhraní na stanici (současně je vidět i nárůst odeslaných dat). Komunikace se samotným serverem nepatří k sledované a omezované komunikaci.

Výraznější výkyvy pozorované při rychlosti 1 MBit jsou způsobené nějakým vnějším vlivem jako je zatížení webového serveru další aplikací apod.

Dodatek C

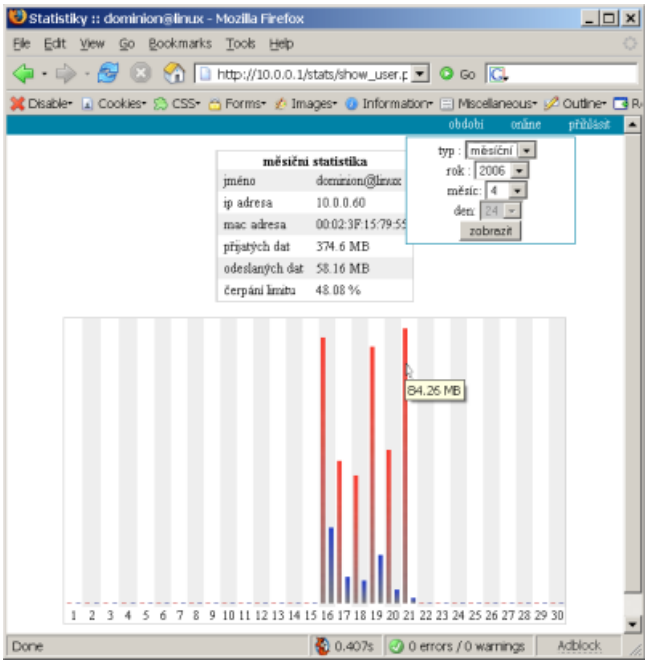
Grafické rozhraní

Internetový prohlížeč Opera Mini je naprogramovaný v javě. Pro používaní je potřebná minimálně podpora MIDP 1.0.

Obrázek C.1: screenshoty Opera Mini - Siemens C65

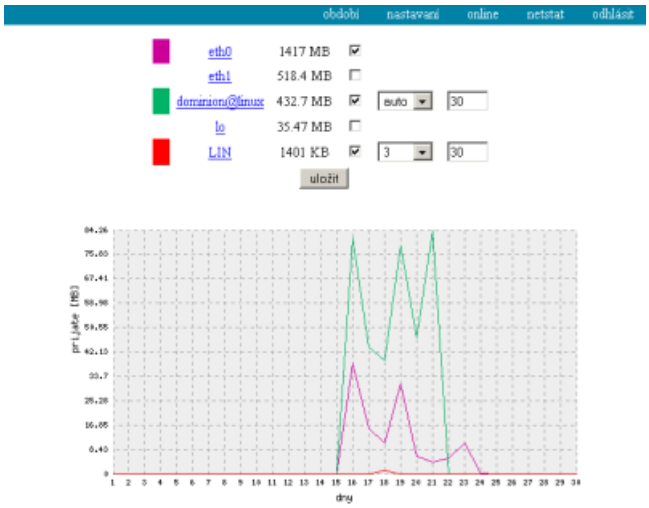
Do uživatelské části (viz. obrázek C.2) má přístup každý uživatel bez přihlášení - zobrazují se mu pouze údaje o stanici, které k programu přistupuje. Pomocí tlačítka *období* si muže listovat historií statistik. Údaje se zobrazují ve formě tabulky a histogramu. Histogram poskytuje detailní přehled o množství přenesených dat a tabulka zobrazuje sumu množství přenesených dat za dané období.

Do části pro spravce systému lze přistupovat pouze po přihlášení. Správce systému má stejné možnosti jako uživatel, navíc se mu zobrazují souhrnné údaje o všech stanicích. Pomocí tlačítka



Obrázek C.2: Rozhraní pro uživatele

nastavení (viz. obrázek C.3) může nastavovat rychlosti a limity jednotlivým uživatelům. Pomocí checkboxu nastavuje viditelnost položek v základním zobrazení. Data jsou seřazeny sestupně podle množství přenesených dat, prvních pět položek se vykresluje do grafu.



Obrázek C.3: Rozhraní pro správce