VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

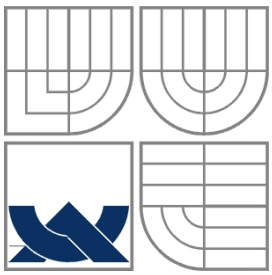
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ROBOT NA PROCHÁZENÍ P2P SÍTÍ

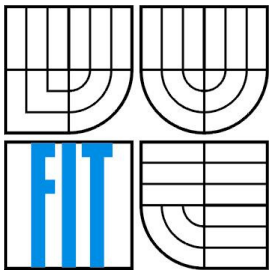
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE MICHAL HOLUB
AUTHOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ROBOT NA PROCHÁZENÍ P2P SÍTÍ

ROBOT FOR BROWSING P2P NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MICHAL HOLUB
AUTHOR

VEDOUCÍ PRÁCE — Ing. VLADIMÍR JANOUŠEK, Ph.D.
SUPERVISOR

BRNO 2009

Abstrakt

Bakalářská práce popisuje typy P2P sítí, jejich nejpoužívanější mutace a protokoly, a návrh a implementaci systému pro procházení P2P sítí. Systém na procházení P2P sítí slouží pro potřebu vyhledávání nových virových nákaz. Tento systém je, na základě zadaných klíčových slov, schopný vyhledat soubory na P2P sítích a stáhnout je na lokální disk. Jestliže jsou stažené soubory spustitelné, systém je předá na testování dalším systémům, které nejsou předmětem této bakalářské práce.

Abstract

Bachelor's thesis describes the types of P2P networks, their most common mutations and protocols, and design and implementation of the system for browsing P2P networks. The purpose of the system for browsing P2P network is to search for files with unrecognized malicious code. This system is capable to search files on P2P network based on given set of keywords, download them to local disk and to determine whether they are executable or not.

Klíčová slova

P2P, výměnné sítě, virus, eDonkey, eMule, Mldonkey, P2P protokoly, protokoly výměnných sítí, Gnutella, KaZaA, Direct Connect, BitTorrent, MZ-PE test

Keywords

P2P, peer to peer networks, virus, eDonkey, eMule, Mldonkey, P2P protocols, peer to peer protocols, Gnutella, KaZaA, Direct Connect, BitTorrent, MZ-PE test

Citace

Holub Michal: Robot na procházení P2P sítí, bakalářská práce, Brno, FIT VUT v Brně, 2009

Robot na procházení P2P sítí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením:
Ing. Vladimíra Janouška, Ph.D, UITS FIT VUT.
Další informace mi poskytl:
Pavel Krčma, AVG.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Michal Holub
11. 5. 2009

Poděkování

Děkuji Pavlu Krčmovi zastupující společnost AVG Technologies za poskytnutí odborných konzultací týkajících se návrhu a implementaci praktické části mé bakalářské práce. Dále děkuji vedoucímu bakalářské práce Ing. Vladimíru Janouškovi, Ph.D. za vedení mé bakalářské práce.

Obsah

Seznam tabulek

Seznam obrázků

1 Úvod

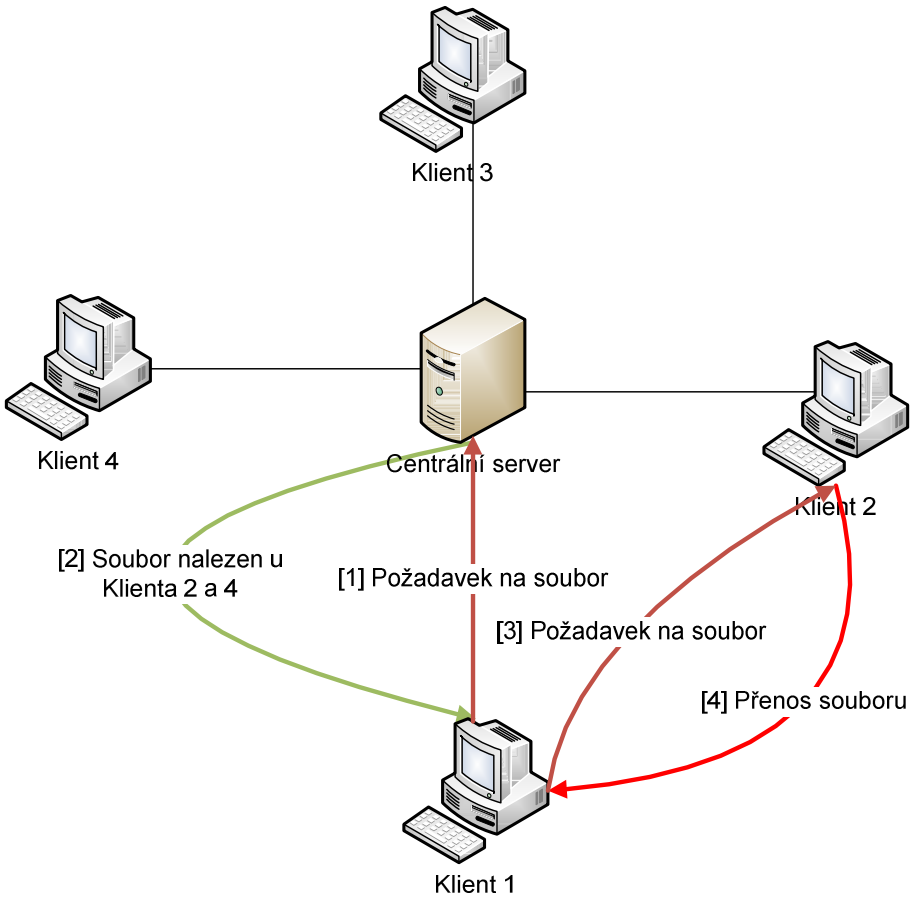
Tato bakalářská práce je zpracováním tématu „Robot na procházení P2P sítí“. Toto téma bylo zadáno ve spolupráci se společností AVG Technologies CZ. Cílem je navrhnout a implementovat systém, který se připojí na specifikovanou P2P síť a na základě zadaných klíčových slov stahuje z této sítě soubory, které jsou podrobeny testům a vyhodnoceny jako spustitelné či nespustitelné. Spustitelné soubory jsou ponechány na testování dalšími systémy.

2 Typy P2P schémat

V této kapitole jsou rozebrány teoretická schémata, jejich výhody a nevýhody. Konkrétní implementace P2P protokolů jsou popsány v kapitole 3.

2.1 Centralizované P2P schéma

Díky existenci pouze jednoho serveru umožňuje velice zjednodušené vyhledávání. Centrální server obsahuje informace o každém připojeném klientovi a souborech, které sdílí. Při každém dotazu klienta na server je vytvořen seznam výsledků odpovídající dotazu, který je odeslán zpět klientovi. Na obrázku 2-1 je zobrazen postup vyhledávání a stahování dat na centralizované P2P síti.



Obrázek 2-1: Centralizované P2P schéma

2.1.1 Výhody

Z principu struktury tohoto schématu je zřejmé, že největší výhoda spočívá v rychlém vyhledávání informací a často jednodušší implementaci protokolu.

2.1.2 Nevýhody

Centralizovaná architektura sítě je více zranitelná díky existenci pouze jednoho serveru. Vzhledem k tomu, že každý klient při přihlášení musí poslat serveru seznam sdílených souborů, se nemůže stát, že server nenalezne hledaný soubor, pokud je zanesen do databáze na serveru. Nicméně klient může během doby, kdy je přihlášen na server, změnit seznam souborů, které sdílí. Tato změna se nemusí ihned replikovat na server. Zda se změna replikuje ihned, či zda klient posílá seznam sdílených souborů periodicky, záleží na typu protokolu.

2.2 Decentralizované P2P schéma

Základní myšlenka je odstranit nejzranitelnější část sítě (centrální server) a nahradit ho strukturou koncových bodů (uzlů), které spolu komunikují na stejné úrovni. Princip šíření informací v tomto typu sítě je podobný práci HUBu (rozbočovače) na sítích typu ethernet. Každá informace je přeposlána všem sousedům, kromě souseda, od kterého informace pochází. Uzel, který se připojuje do sítě, musí znát aspoň jeden uzel nacházející se v síti. Po připojení k libovolnému uzlu je tato událost komunikována všem ostatním uzlům a zároveň nově připojený uzel obdrží seznam všech uzlů na síti. Princip šíření ostatních typů informací po této síti, např. dotaz na soubor, je obdobný. Klient pošle dotaz na všechny sousedící uzly. Ty dotaz přepošlou svým sousedům a pokud některý z nich má požadovaný soubor, spojí se přímo s uzlem, který tento soubor požadoval.

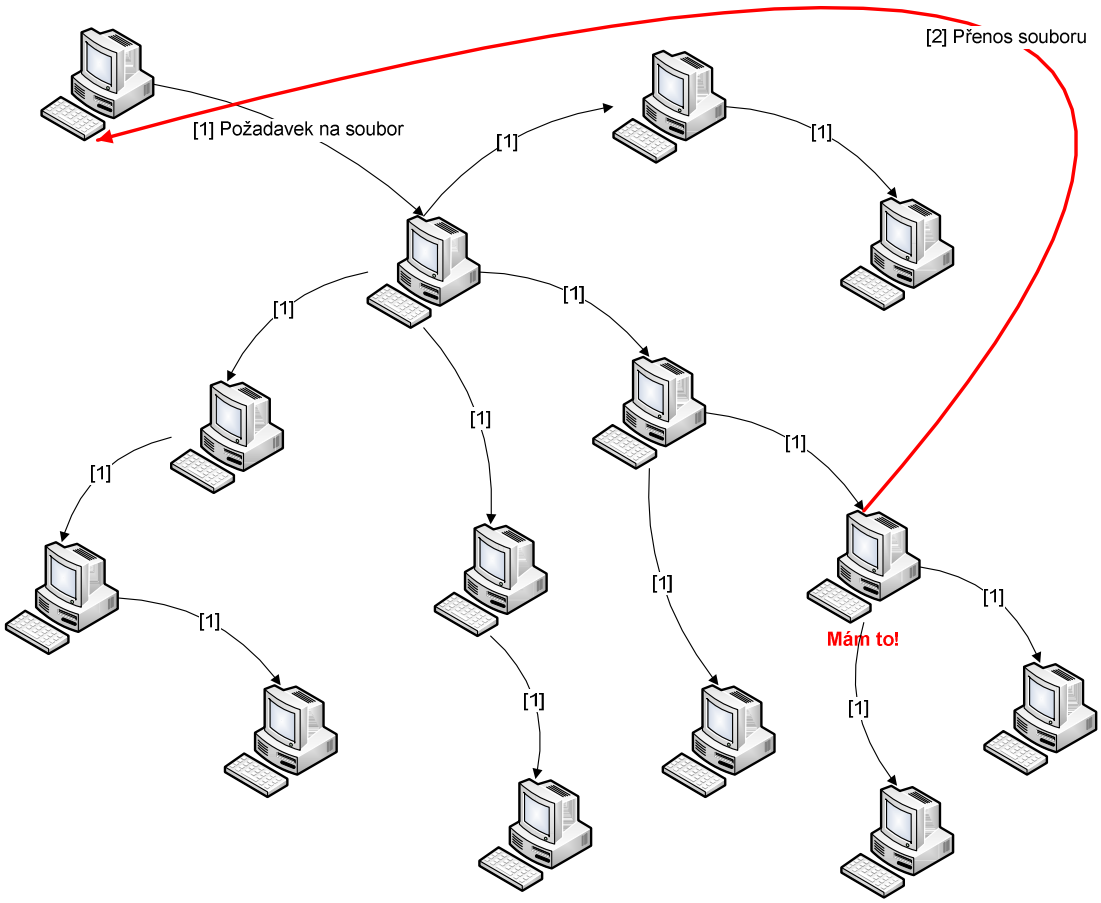
Teoreticky je počet uzlů v takové sítí neomezený, v praxi má však každá komunikovaná informace nastavené TTL (Time to live - počet uzlů, skrz které může projít) jako prevence proti zahlcení sítě.

2.2.1 Výhody

Tento typ sítě je velice odolný proti výpadkům a pokusům o zastavení činnosti sítě.

2.2.2 Nevýhody

Hledání na decentralizovaných sítích je poměrně pomalejší. I když se hledaný soubor na síti nachází, není garantováno, že TTL dotazu nevyprší dříve, než dorazí k uzlu, který tento soubor má.



Obrázek 2-2: Decentralizované P2P schéma

3 Nejpoužívanější P2P sítě a protokoly, které používají

Existují dva základní modely P2P schémat: centralizované a decentralizované. V praxi se čistě centralizované nebo decentralizované sítě používají velice zřídka nebo vůbec. Většinou se můžeme setkat s centralizovanými sítěmi s použitím clusteru (shluku) serverů nebo decentralizovanými sítěmi s dvěma třídami uzlů: uzly zajišťující směrování dotazů (servery) a uzly, které dotazy pokládají a vyřizují (klienti).

3.1 Gnutella

V prvních revizích byla Gnutella čistě decentralizovanou sítí jejíž princip je popsán v kapitole 2.2.

Novější verze Gnutelly jsou tvořeny z uzlů typu leaf (klient) a uzlů ultranode, které slouží jako směrovače dotazů. Každý klient je připojen k malému počtu směrovačů (typicky 3). Každý směrovač je připojen k dalším směrovačům (typicky 32). Když chce klient vyhledat soubor, pošle dotaz na všechny směrovače, ke kterým je připojen pomocí protokolu QRT (query routing table). Směrovač si přidá dotaz do své tabulky dotazů, kterou si pravidelně vyměňuje se svými sousedy. Když některý ze směrovačů ví, který klient má požadované data, pošle tuto informaci klientovi, který o ní žádal. Pokud může klient, který má požadovaná data, přijímat příchozí spojení, je vytvořeno spojení mezi klienty a soubor přenesen. Pokud nemůže, jeden ze směrovačů požádá klienta s daty, aby se pokusil spojit s klientem, který požadoval data. Pokud ani jeden z klientů nemůže přijímat příchozí spojení, soubor nemůže být přenesen.

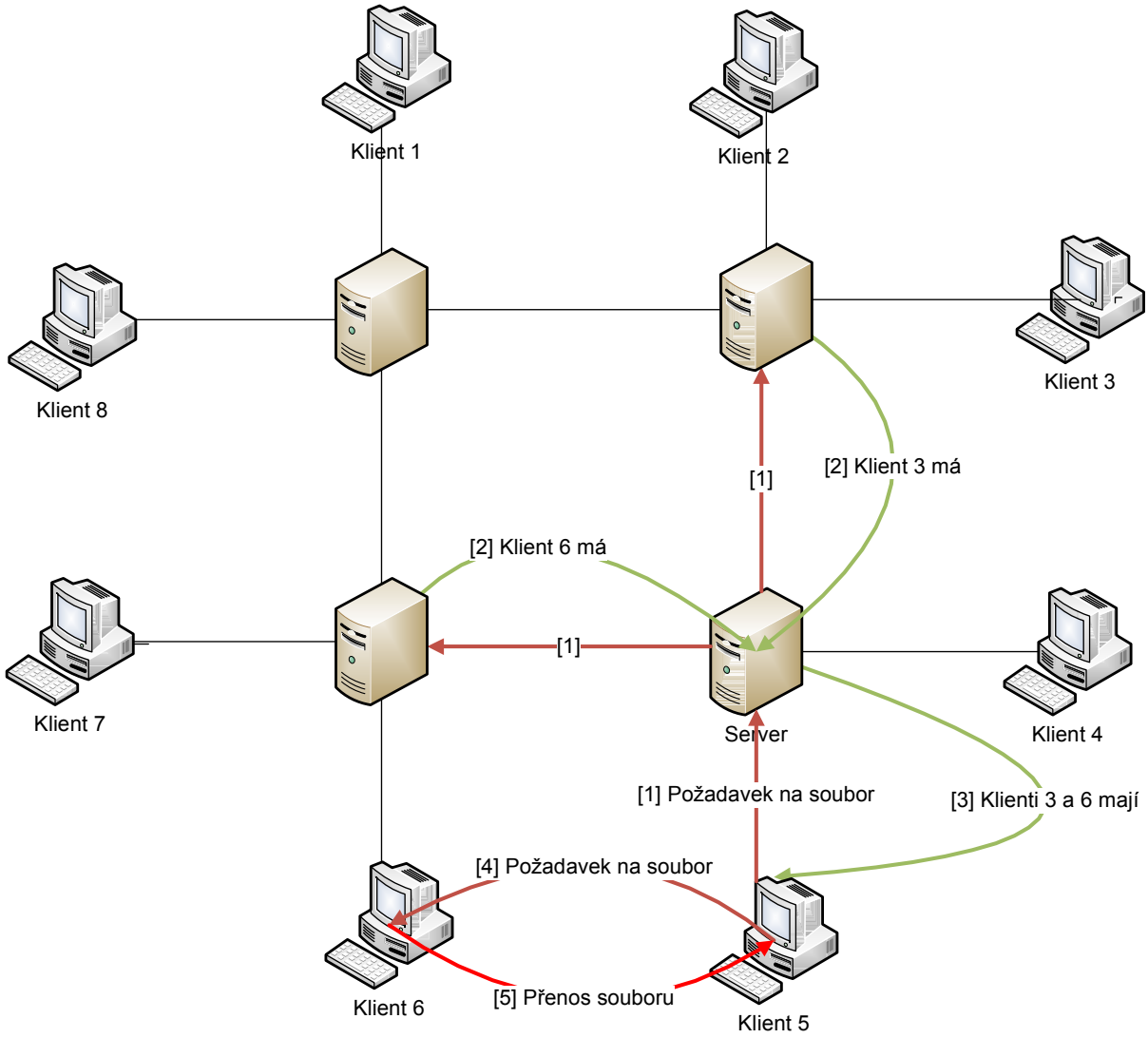
Ačkoli byla Gnutella původně proprietárním protokolem, díky reverznímu inženýrství vzniklo velké množství open source klientů díky čemuž byla stále populárnější.

3.2 KaZaA

Síť KaZaA používá protokol FastTrack. Skládá se z klientů ON (ordinary node) a serverů SN (super node). Klient se připojí do sítě přihlášením na jeden ze serverů a pošle mu metadata (index souborů, které sdílí). Klient komunikuje pouze s tímto serverem, nicméně obdrží i seznam dalších serverů pro případ výpadku serveru, ke kterému je momentálně připojen. Pokud klient hledá soubor, jeho dotaz se nejprve snaží uspokojit server, ke kterému je připojen, vyhledáváním v indexu souborů jeho klientů. Pokud daný soubor server mezi svými klienty nenalezne, přepošle dotaz dalším serverům, s kterými je momentálně v kontaktu (typicky kolem 10 serverů). Pokud některý ze serverů nalezne soubor ve svém indexu, pošle tuto informaci původnímu serveru jako seznam klientů sdílejících soubor. Původní server po obdržení všech seznamů od svých sousedních serverů tyto seznamy spojí a přepošle je klientovi, který hledal soubor. Následně je na uživateli, aby si vybral od kterého klienta bude soubor stahovat.

Každý server periodicky mění množinu serverů, ke kterým je připojen z důvodu větší pestrosti nabízených dat. Díky tomu se celá síť stále dynamicky mění. Klient se může stát serverem pokud může přijímat příchozí spojení a má dostatek prostředků k rychlému vyhodnocování dotazů (CPU, RAM, propustnost sítě).

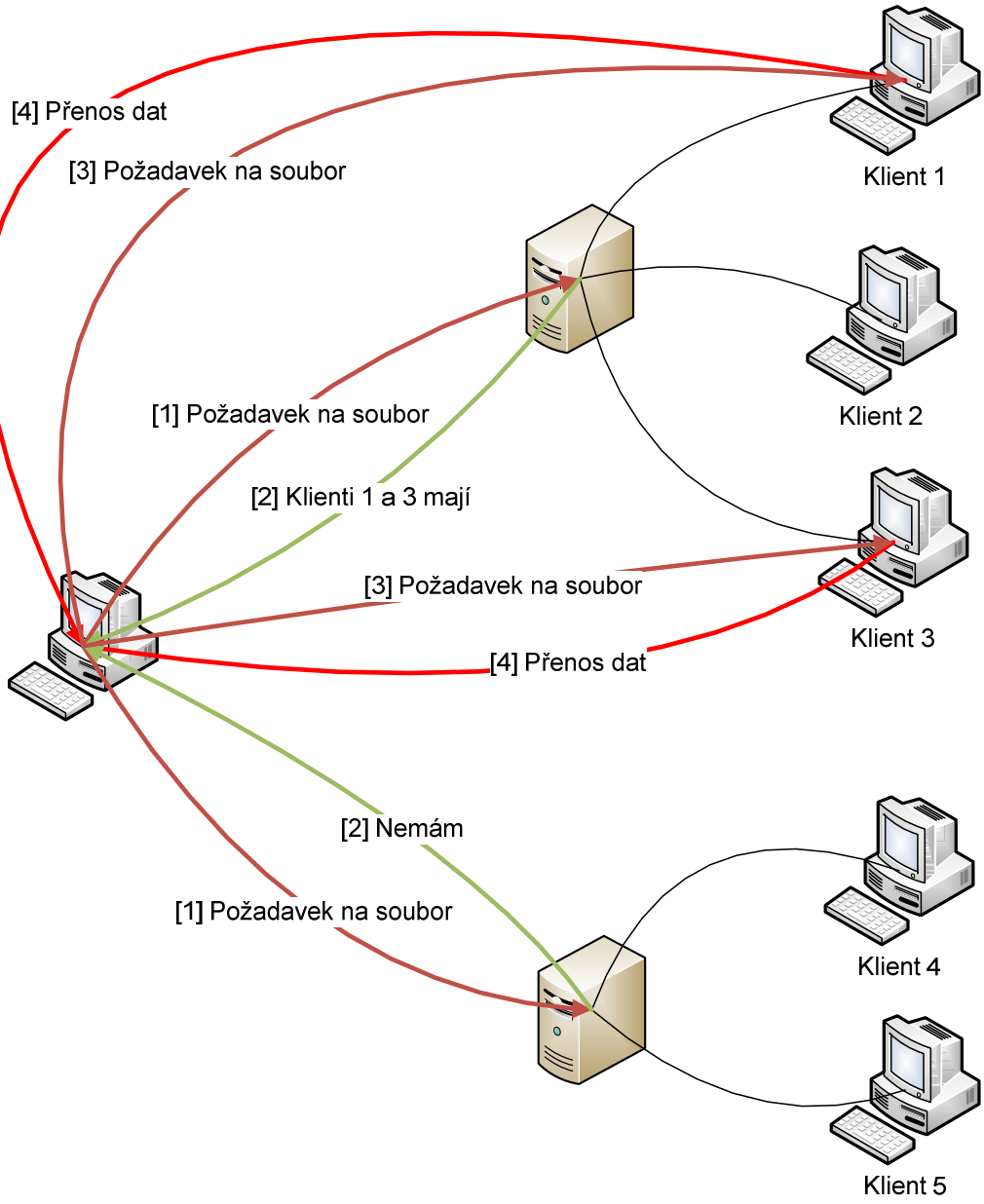
Jedná se o uzavřený protokol s binární formou packetů, ke kterému existuje neúplná neoficiální dokumentace a malé množství open source klientů. Navzdory tomu má KaZaA velké množství uživatelů.



Obrázek 3-1: KaZaA

3.3 Direct Connect

Když se chce uživatel připojit do této sítě, musí si z webu stáhnout HUBlist, který obsahuje adresy několika (v řádu stovek až tisíců) na sobě nezávislých HUBů. HUB je konkrétní server, který vyřizuje požadavky klientů. Každý klient může být připojen na více HUBů, nicméně s každým komunikuje zvlášť a HUBy si mezi sebou nevyměňují žádné informace. Po připojení na HUB pošle klient serveru informace o souborech, které sdílí. Při dotazu na soubor vrátí HUB klientovi seznam klientů, které daný soubor mohou poskytnout. Přenos souborů se děje v duchu P2P filozofie – přímo mezi klienty bez jakékoli účasti serveru. Některé implementace klientů dokáží stahovat různé části stejného souboru od více klientů zároveň.



Obrázek 3-2: Direct Connect

3.4 EDonkey(eMule)

Síť eMule se skládá z několika stovek eMule serverů a miliónů klientů. Každý server obsahuje centralizovaný seznam souborů svých klientů. Servery na síti eMule spolu nekomunikují, nicméně podobně jako u sítě Direct Connect může být klient připojen na více serverů. Seznam serverů je většinou součástí klienta, nicméně je možné další adresy serverů přidat později stáhnutím z webu. Vyřizování dotazů klienta a přenos souborů mezi klienty dodržuje běžné schéma centralizované sítě.

EMule servery slouží jako centralizované databáze souborů. Sami nenesou žádné soubory, ale v případě že ani jeden z klientů nemůže přijímat příchozí spojení, umožňují propojení klientů skrz server.

I přes to, že je specifikace této sítě uzavřená, existuje k ní kvalitní neoficiální dokumentace a dostatečné množství open source klientů. Tato síť ke komunikaci používá binární packety.

3.4.1 Komunikace klient – server

Klient ke komunikaci se serverem používá jedno TCP spojení. Po připojení server klientovi pošle jeho ID, které je platné po dobu spojení se serverem. V této síti se používají dva druhy ID. Jeden se jmenuje low ID a druhý high ID. Pokud je klient schopný přijímat příchozí spojení je mu přiděleno high ID. V opačném případě low ID. Po inicializaci spojení server obratem pošle informace o požadovaných souborech a klientech.

UDP spojení je používáno k periodickému zasílání stavu serveru klientům. Implementace UDP spojení není povinná.

3.4.2 Komunikace klient - klient

Ke spojení mezi klienty se používá TCP i UDP spojení.

TCP spojení je použito pro přenos souborů. Tento přenos může proběhnout pouze v případě dvou klientů s high ID nebo jednoho klienta s high ID a druhého s low ID. Dva klienti s low ID mezi sebou nemohou přenést soubor – ani jeden z nich nemůže přijímat příchozí spojení.

UDP spojení je použito k výměně informací o serverech. Implementace tohoto druhu spojení není povinná.

V této síti je možné, aby klient stahoval soubory po částech. Díky tomu může každou část stahovaného souboru stáhnout od jiného klienta. Obdobně může jiný klient stáhnout pouze část jednoho souboru, který je sdílen původním klientem.

3.4.3 Identifikace souborů a kontrola konzistence souborů

K jednoznačné identifikaci souborů slouží File hash, což je 128 bitová sekvence vypočítaná klientem na základě obsahu daného souboru. Tato sekvence se počítá MD4 algoritmem. Soubor je dále rozdělen na 180kb bloky a pro každý spočítán hash SHA1 algoritmem.

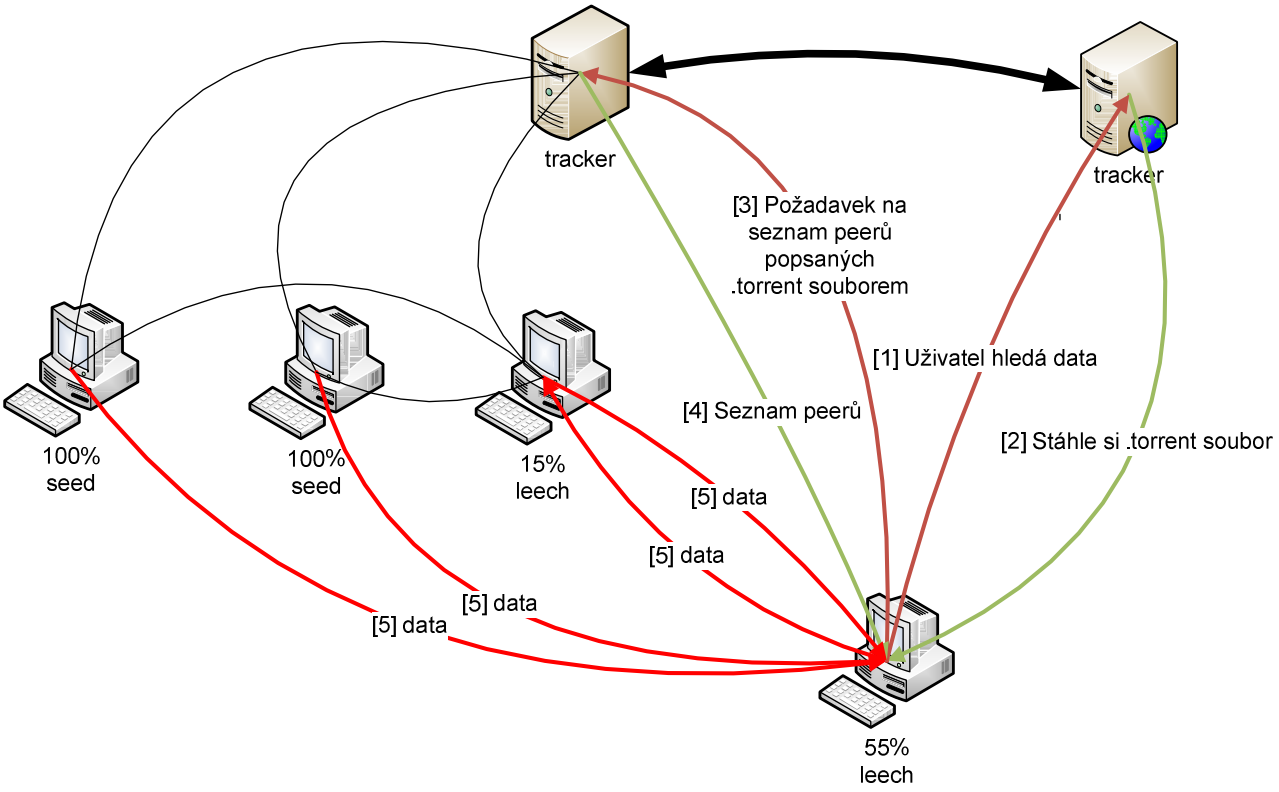
3.5 BitTorrent

BitTorrent je protokol pro přenos obrovských množství dat. Jeho topologie se skládá z jednoho serveru (tracker) a klientů (peer), kteří se dále dělí na dva typy (leech, seed). Tracker musí obsahovat

dvě části. Jedna z částí trackeru je webové rozhraní, kde uživatelé vyhledávají data pomocí klíčových slov. Tato data jsou popsána soubory typu .torrent, které obsahují kontrolní součty jednotlivých částí souborů a adresu trackeru. Když má uživatel zájem o nějaká data, musí si stáhnout .torrent soubor a načíst ho pomocí jeho BitTorrent klienta. BitTorrent klient se připojí na tracker jehož adresu nalezne v .torrent souboru. Druhá část trackeru je démon obsluhující požadavky klientů. Po připojení na tracker klient obdrží seznam peerů, které sdílí požadovaná data popsaná daným .torrent souborem. Peer je typu seed v případě, že sdílí kompletní data popsaná .torrent souborem, v jiném případě je typu leech. Každý peer komunikuje s ostatními peery a vyměňují si data bez ohledu na to, zda je typu seed nebo leech. Při prvotní distribuci dat mezi peery posílá seed každému leechu jinou část souboru. Díky tomu si klienti typu leech mohou vyměňovat data i mezi sebou a dochází tak k efektivnímu síření dat.

Každý klient má možnost sdílet nová unikátní data. Uživatel v takovém případě musí pomocí klienta vytvořit torrent soubor a nahrát ho na tracker. Tento mechanismus má tu výhodu, že pokud jsou sdílené soubory uložené u klienta později napadené virem (změní se jejich kontrolní součet), nelze je už použít pro sdílení. Nicméně pokud originální soubory vir obsahovaly, samotný BitTorrent protokol neobsahuje žádné mechanismy pro jeho odhalení.

Dnes je jedná o nejoblíbenější protokol. Existuje tisíce veřejných nebo privátních trackerů. Jeho celkový provoz představuje asi 35% z celkového datového toku po internetu



Obrázek 3-3: BitTorrent

4 Návrh systému pro procházení P2P sítí

Systém pro procházení P2P sítí má za úkol podle klíčových slov vyhledat a stahovat soubory z P2P sítí. Tyto soubory otestovat a ponechat ty soubory, které mohou být nakažené škodlivým kódem. Grafický návrh systému je zobrazen na obrázku Obrázek 4-1.

Systém bude složený z několika částí

- P2PRobot - bezobslužné řízení celého systému
- klient – změna stavu běžícího systému
- démon – vyhledávání a stahování souborů z P2P sítí
- tester - spouštěč MZ-PE testu

Jednotlivé části spolu komunikují pomocí TCP/IP protokolu. Díky tomu může každá z částí systému běžet na samostatném stroji.

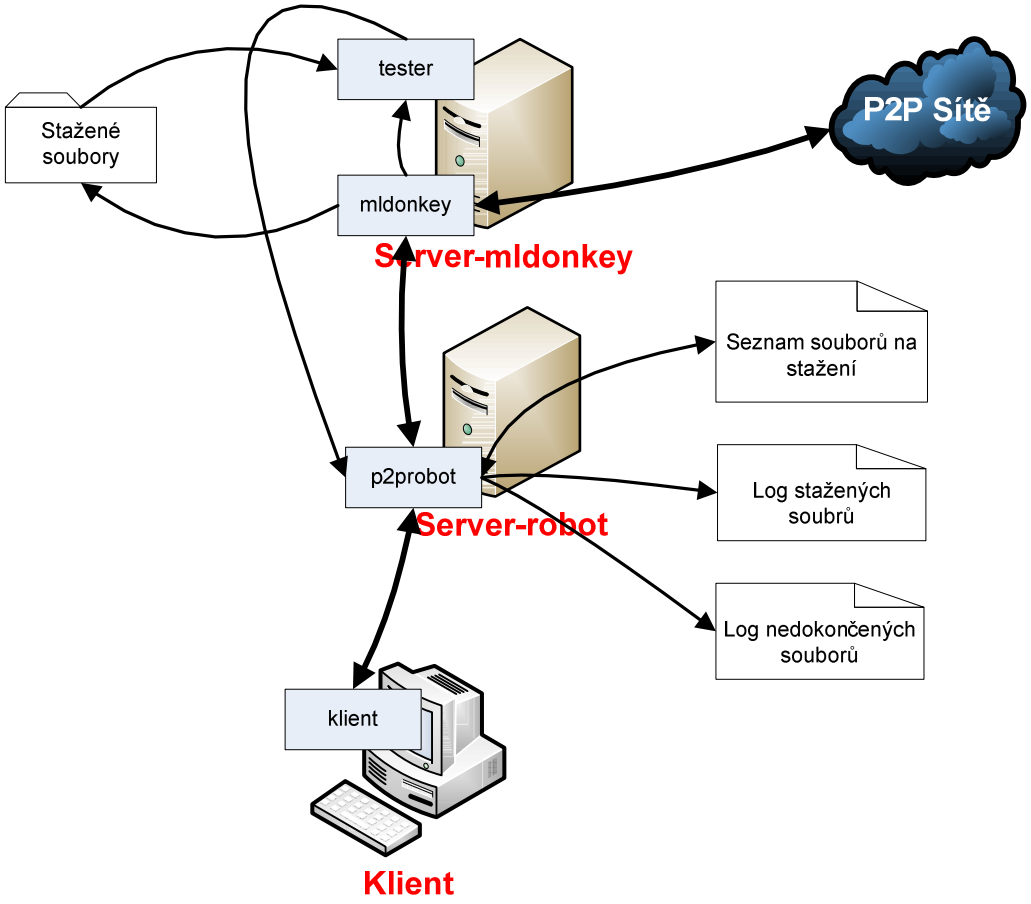
Jako démon vyhledávající a stahující soubory z P2P sítí bude použita aplikace Mldonkey, která podporuje několik P2P sítí. Je možné ji ovládat ručně přes telnet konzoli nebo pomocí GUI protokolu.

Všechny ostatní části (P2PRobot, klient, tester) bude nutné naimplementovat. Jako implementační jazyk bude použit jazyk C.

Tester je implementací MZ-PE testu (více viz podkapitola 5.3.1).

P2PRobot bude jádrem celého systému. Bude komunikovat se všemi ostatními prvky systému. Jeho úkolem je řídit démona Mldonkey, zpracovávat výsledky testeru a naslouchat klientovi, pomocí kterého je možné manuálně měnit stav systému za běhu.

Cílovým operačním systémem by měl být Windows XP, Linux nebo jejich kombinace.



Obrázek 4-1: Systém na procházení P2P sítí

5 Implementace systému

5.1 Mldonkey

Mldonkey je Open Source projekt. V našem systému zastává funkci démona, který stahuje soubory z P2P sítí.

Mldonkey může běžet na operačních systémech Linux, Solaris, MacOSX, MorphOS a Windows. V našem řešení je Mldonkey otestováno na systémech Windows XP a Linux.

5.1.1 Ovládání přes telnet konzoli

Ačkoliv je Mldonkey démon řízen pomocí P2PRobota zcela automaticky pomocí GUI protokolu (viz podkapitola 5.1.2), je možné řídit Mldonkey i ručně přes konzoli pomocí textových příkazů. V této podkapitole jsou popsány ty nejužitečnější z nich.

Tabulka 5-1: Telnet příkazy

auth <user> <password>	Přihlášení jako uživatel <user> s heslem <password>

<vm>	Seznam serverů, se kterými je navázáno spojení
n <ip>	Připojení k novému serveru
s <query>	Hledání podle klíčových slov
Vs	Zobrazit všechna hledání
vr <num>	Zobrazit konkrétní hledání
d <num>	Stáhnout soubor
dllink <link>	Stáhnout soubor pomocí ed2k odkazu
vd <num>	Zobrazit informace o stahovaném souboru
pause <num>	Pozastavit stahování souboru
cancel <num>	Zrušit stahování konkrétního souboru
cancel all	Zrušit stahování všech souborů
q	Opustit telnet
kill	Vypnout démona
passwd <noveheslo>	Změnit heslo aktuálního uživatele

5.1.2 Ovládání přes GUI

Pro manuální ovládání Mldonkey existuje i několik GUI aplikací, které používají mldokey GUI protokol. Ačkoliv pro běh systému není ovládání pomocí GUI nezbytné, je vhodné ho použít pro ladění chybových stavů a získání více informací o stavu systému.

Jako optimální řešení pro náš systém jsme vybrali jednoduchou aplikaci jMoule, s jejíž pomocí je, po připojení k Mldonkey, možné systém připojit k dalším eMule serverům, ručně stahovat nové soubory, popř. rušit probíhající stahování souborů.

Jediná nevýhoda jMoule je, že pracuje pouze s jedním uživatelem – admin. Není možné tohoto uživatele změnit. Nicméně v našem systému nám toto omezení nevadí.

Aplikace jMoule je Open Source projekt napsaný v Javě. Naleznete ji v příloze A v umístění „install/jMoule“ a je možné ji, po instalaci Java prostředí, spustit na operačních systémech Windows XP i Linux.

5.1.3 GUI protokol

Je binární protokol, pomocí kterého P2PRobot ovládá démona Mldonkey.

5.1.3.1 Formát packetu

Tabulka 5-2: Formát packetu

Hlavička	Opcode	Data
Int32	Int16	Proměnná velikost

Hlavička je velikost zbývajících dat (Opcode + Data) v bytech. Opcode je jednoznačné označení typu packetu. Následují data, jejichž typ je závislý na Opcode. Všechna data jsou přenášena v little-endian.

5.1.3.2 Základní datové typy

Tabulka 5-3: Základní datové typy

Int8	Integer o velikosti jeden bajt
Int16	Integer o velikosti dva bajty
Int32	Integer o velikosti čtyři bajty
Int64	Integer o velikosti osum bajtů

5.1.3.3 Složené datové typy

String

Skládá se z hlavičky (int16), která udává velikost řetězce, který následuje bezprostředně za hlavičkou. Řetězec není ukončen žádným ukončovacím znakem.

Float

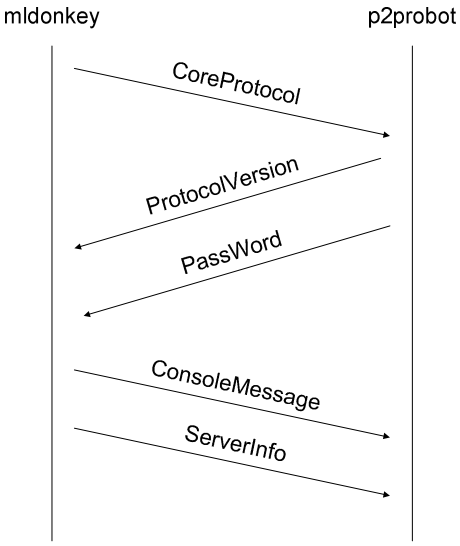
Čísla s pohyblivou desetinnou čárkou jsou přes GUI protokol přenášeny jako typ String. A to ve tvaru, který je dán regulárním výrazem [0-9]+ \.[0-9]+

List of

Je seznam položek. Skládá se z hlavičky (int16), která udává počet položek. Za hlavičkou následují jednotlivé položky bez oddělovače.

5.1.3.4 Inicializace spojení

Následující schéma popisuje inicializaci spojení mezi démonem Mldonkey a P2PRobotem.



Obrázek 5-1: Inicializace spojení

Core protocol

Je první zpráva, kterou Mldonkey posílá novému klientovi, používající GUI protokol. Součástí této zprávy je podporovaná verze protokolu.

Protocol version

Tato zpráva dává Mldonkey najevo, jaké verzi protokolu rozumí. Je to první zpráva, kterou P2PRobot Mldonkey posílá.

Password

Obsahuje přihlašovací jméno a heslo.

Console Message

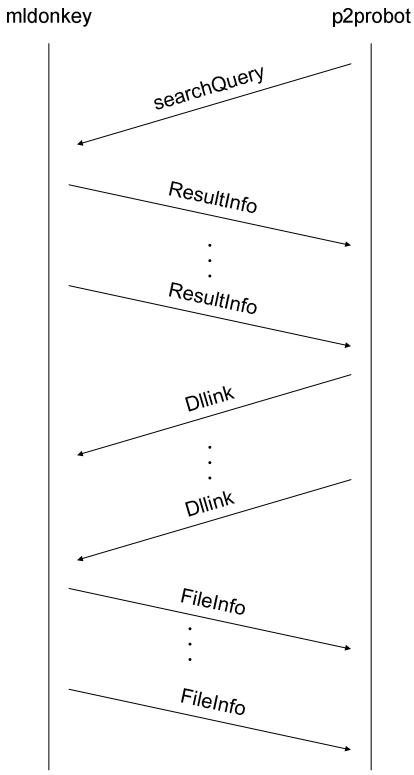
Obsahuje text nastavený správcem démona Mldonkey.

Server Info

Obsahuje informace o serverech, ke kterým je démon Mldonkey připojen.

5.1.3.5 Proces hledání a stahování souborů

Následující schéma popisuje komunikaci mezi démonem Mldonkey a P2PRobotem během hledání a stahování souborů.



Obrázek 5-2: Hledání a stahování souborů

Search query

Hledání souborů na P2P síti podle klíčových slov, minimální a maximální velikosti a typu souboru.

Result Info

Je odpovědí Mldonkey na zprávu „search query“. Pro každý nalezený soubor Mldonkey pošle přes GUI protokol P2PRobotu jednu zprávu Result Info, která obsahuje informace o souboru jako: identifikační číslo výsledku, P2P síť, jméno, jednoznačný identifikační odkaz ve tvaru ED2K URL, velikost souboru, formát a další.

Dllink

Stažení souboru podle URL. P2PRobot posílá URL typu ed2k.

File info

Nese informaci o souboru, který se právě stahuje nebo byl stažen. Z této zprávy využívá P2PRobot tyto informace: identifikátor souboru, jméno souboru, velikost souboru, velikost stažených dat, dostupnost, aktuální rychlost stahování.

5.1.3.6 Další zprávy

Pause download

Pozastaví stahování souboru. Tato zpráva je poslána ve chvíli, kdy je staženo prvních 1024B. Pokud MZPE test (viz podkapitola 5.3.1) vyhodnotí soubor jako nebezpečný, pokračuje se ve stahování souboru zasláním zprávy Resume download.

Resume download

Pokračování ve stahování souboru.

Remove download

Zrušení stahování souboru.

Console Command

Pomocí této zprávy je možné poslat Mldonkey příkaz, který je rozpoznán telnet konzolí mldokey. P2PRobot používá tuto zprávu ke spuštění MZPE testu (viz podkapitola 5.3.1).

5.2 P2PRobot

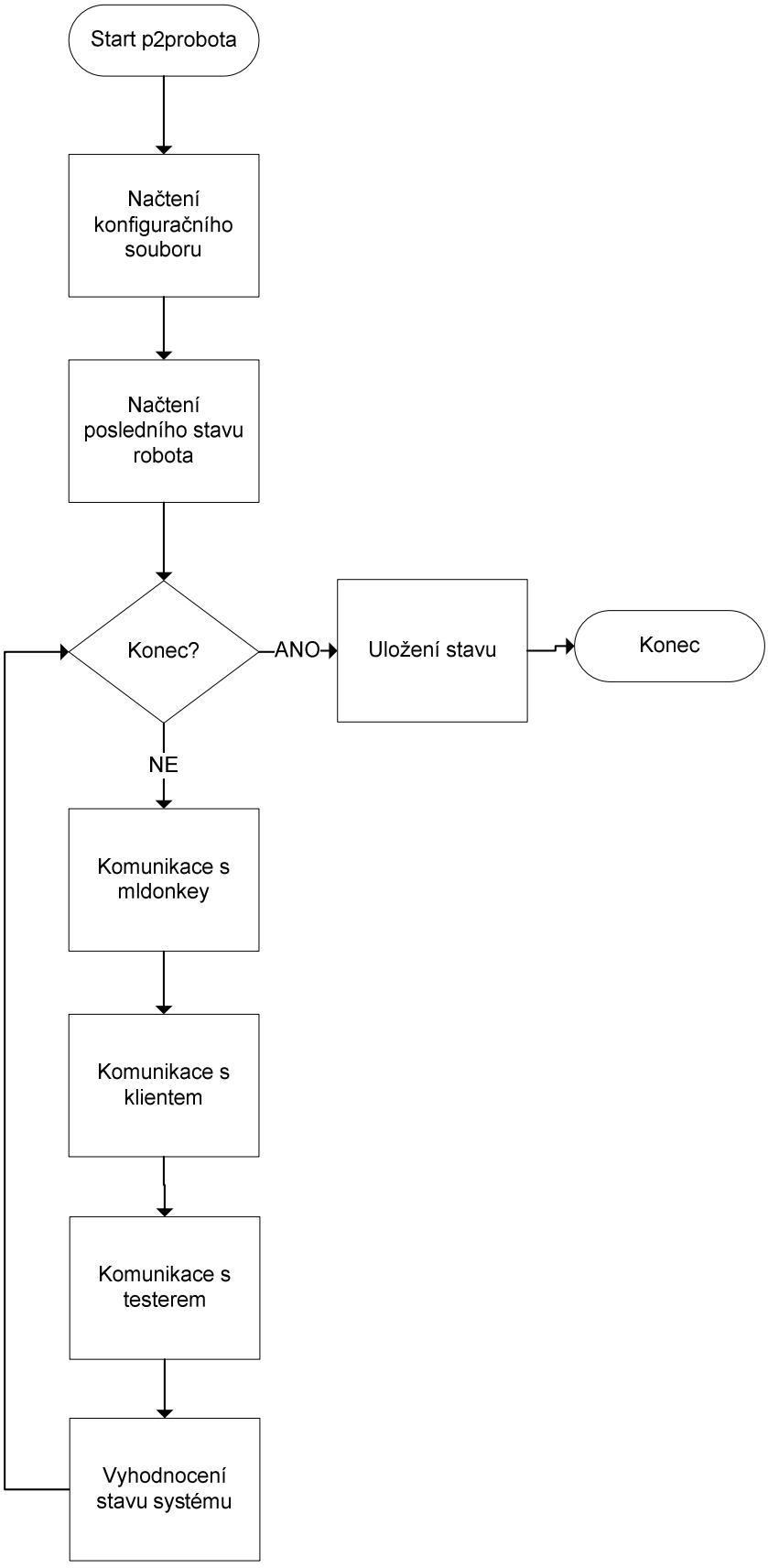
P2PRobot je řídící část systému. Tento robot se stará o řízení démona Mldonkey. Chování robota je možné ovlivnit pomocí konfiguračního souboru a řídícího rozhraní (viz podkapitola 5.6.2.3).

Tato aplikace je implementována v jazyce C. Kromě standardní knihovny jazyka C a BSD socketů nevyužívá žádné další knihovny.

5.2.1 Komunikace P2PRobota s ostatními prvky systému

Všechna komunikace je implementována pomocí BSD sockets. Při startu P2PRobota je vytvořen jeden klientský socket pro komunikaci s Mldonkey. Pak další dva serverové, jeden pro komunikaci s klientem a druhý pro přijímání zpráv od testeru.

5.2.2 Grafické znázornění běhu programu



Obrázek 5-3: Běh programu

Načtení konfiguračního souboru

Konfigurační proměnné jsou načteny z konfiguračního souboru, jehož cesta se zadává jako první parametr příkazové řádky při spuštění P2PRobota. Při chybné struktuře konfiguračního souboru se P2P robot pokusí přeskočit chybu a pokračovat na další položce. V případě nenalezení konfiguračního souboru je nastavení načteno z výchozího konfiguračního souboru defaults.ini.

Načtení posledního stavu robota

V tomto kroku jsou načteny uložené ED2K URL odkazy na soubory nacházející se na P2P sítích (viz podkapitola 5.5.2).

Komunikace s Mldonkey

V každém cyklu hlavní smyčky programu se P2PRobot pokusí o přijetí zprávy od Mldonkey.
V případě nové příchozí zprávy je podle identifikačního čísla zprávy – opcode (viz podkapitola 5.1.3) – rozpoznán typ zprávy. Zpráva je dále v podobě struktury t_message předána příslušné funkci, která rozpozná data nesená zprávou.

Komunikace s testerem

Ve chvíli kdy P2PRobot zjistí, že již bylo staženo aspoň 1024B z nějakého souboru, je stahování tohoto souboru pozastaveno zprávou „pause download“ (viz podkapitola 5.1.3) a zprávou „console command“ je spuštěn MZPE test, který rozhodne o tom, zda se bude pokračovat ve stahování souboru nebo zda bude stahování ukončeno a soubor smazán. Výsledek testu je P2PRobotovi zaslán přes TCP/IP protokol připojením na serverový socket P2PRobota.

Komunikace s klientem

Pomocí klientské konzole lze zjistit stav systému, upravit konfigurační hodnoty nebo ukončit běh systému (viz podkapitola 5.6.2.3).

Vyhodnocení stavu systému

Na základě dat přijatých od Mldonkey nebo od klienta je vyhodnocen nový stav celého systému a v případě potřeby jsou odeslány zprávy démonovi Mldonkey nebo informace klientovi.

Uložení stavu

Tímto se rozumí uložení konfiguračního souboru a všech ED2K odkazů.

5.3 Tester

Součástí systému je tester, který je spouštěn P2PRobotem přes Mldonkey démona. Tester ověřuje, zda je soubor spustitelný a výsledek posílá zpět P2PRobotovi.

5.3.1 MZ-PE test

Tímto testem se zjišťuje, zda se jedná o spustitelný soubor v systémech Windows NT. Tento test je implementovaný v jazyce C

5.3.1.1 Ověření spustitelnosti souboru

Všechny soubory, které jsou spustitelné v systémech Windows NT, jsou formátu Portable Executable. Tento test je proveden po stažení prvních 1024B.

Pro ověření spustitelnosti je nutné nejprve načíst hlavičku PE souboru a zkontrolovat příznak „Magic number“ (viz tabulka Tabulka 5-4: Formát PE hlavičky). Ten musí obsahovat hodnotu 0x54AD, což odpovídá ASCII znakům „MZ“. Dále je důležitá hodnota ukazatele na PE příznak, která se nachází na bajtech 60-63 hlavičky PE souboru. Po načtení této hodnoty tester zkontroluje, zda se na daném místě v paměti opravdu nachází příznak PE. Pokud ano, s největší pravděpodobností se jedná o spustitelný soubor v systémech Windows NT.

Tabulka 5-4: Formát PE hlavičky

Velikost v bajtech	Název	Poznámka
2	Magic number	MZ příznak
2	Bytes on last page of file	
2	Pages in file	
2	Relocations	
2	Size of header in paragraphs	
2	Minimum extra paragraphs needed	
2	Maximum extra paragraphs needed	
2	Initial (relative) SS value	
2	Initial SP value	
2	Checksum	
2	Initial IP value	
2	Initial (relative) CS value	
2	File address of relocation table	
2	Overlay number	
8	Reserved words	
2	OEM identifier (for e_oeminfo)	
2	OEM information; e_oemid specific	
20	Reserved words	
4	File address of new exe header	Ukazatel na PE příznak

5.4 Vstupy systému

Primárním vstupem systému jsou klíčová slova, podle kterých se vyhledávají a stahují soubory. Přesněji řečeno taková klíčová slova, která ve většině případů vedou k nalezení a stažení souboru nakaženého nějakým typem viru.

Soubory nakažené škodlivým kódem se nejčastěji vyskytují v jedné těchto oblastí:

- Spustitelné soubory, které klamou svým názvem a příponou (nejčastěji pornografická videa)
- Upravené spouštěcí soubory komerčních aplikací (cracky)

5.5 Výstupy systému

5.5.1 Stažené soubory

Jsou hlavním výstupem systému. Jejich umístění je určeno nastavením aplikace Mldonkey. Ve výchozím nastavení se tyto soubory ukládají do adresáře incoming v umístění, že kterého je spuštěna aplikace Mldonkey.

5.5.2 Seznamy ED2K odkazů

Jsou soubory obsahující ED2K odkazy na soubory nalezené pomocí zadaných klíčových slov, nacházející se na některé z P2P sítí.

Formát seznamu ED2K odkazů

Každý odkaz je na samostatném řádku a má následující formát

```
Název souboru|Velikost|MD4HASH|
Beatles (The) - Ticket To Ride.gp3|6559|F33659DB7B3E1E2D5526C0293D29DD36|
```

Tabulka 5-5: Seznamy ED2K odkazů použité P2PRobotem

dllink.txt	Seznam odkazů, které ještě nebyly zpracovány. Tyto odkazy jsou výsledkem vyhledávání na P2P sítích na základě klíčových slov
log_deleted.txt	Smazané soubory v průběhu stahování
log_downloaded.txt	Stáhnuté soubory
log_previewed.txt	Soubory, z kterých bylo staženo aspoň 1024B

5.6 Instalace a nastavení systému

5.6.1 Mldonkey

5.6.1.1 Instalace

Instalace pomocí instalačního balíku na operačních systémech Windows NT

Instalátor MLDonkey je možné získat na webové prezentaci projektu MLDonkey nebo v příloze A v umístění „*install\mldonkey\win32*“. Spusťte instalátor a program nainstalujte. Tento návod používá umístění *„C:\Program Files\mldonkey“*. MLDonkey se spouští souborem run_mldonkey.exe, který se nachází přímo v místě instalace. Tímto spuštěním jsou v instalační složce vytvořeny nezbytné soubory.

Instalace pomocí balíčkovacího systému linuxové distribuce

Přihlaste se jako uživatel s oprávněním instalovat nové balíčky. Vyhledejte balíček se jménem podobným „Mldonkey-server“ a nainstalujte.

Postup v systému debian:

apt-get install Mldonkey-server

Kompilace ze zdrojového kódu na systémech Linux

Pokud se aplikace MLDonkey nenachází v balíčkovacím systému vašeho operačního systému, budete ji muset zkompilovat ze zdrojového kódu. Ten naleznete na webové prezentaci projektu MLDonkey nebo v příloze A v umístění *„install/mldonkey/linux/“*. Výstupní binární soubor „mlnet“ je možné spustit z jakéhokoli umístění. Konfigurační soubory ukládá do umístění „~/.mldonkey/”.

5.6.1.2 První spuštění

Operační systémy Windows NT

Démon se spouští příkazem run_mldonkey.exe v instalačním adresáři aplikace.

Operační systémy Linux

Pokud jste nastavili spuštění Mldonkey ihned po startu, démon by již měl běžet. Je možné to zkontrolovat příkazem:

```
ps aux | grep mlnet
```

V opačném případě je možné démona spustit následovně:

```
mlnet > /dev/null 2>&1 &
```

Nebo pokud chcete zachytit případná chybová hlášení při běhu démona:

```
screen -dmS mldonkey mlnet
```

5.6.1.3 Nastavení

Po spuštění začne Mldonkey podle výchozího nastavení naslouchat na IP adrese 127.0.0.1:4000 a 127.0.0.1:4001. Ze stejného stroje, na jakém běží Mldonkey, se připojíme pomocí telnet klienta na adresu 127.0.0.1:4000. Po navázání spojení se přihlásíme výchozím uživatelem admin příkazem:

```
auth admin ""
```

Nyní máme přístup ke všemu nastavení. Změníme heslo výchozího uživatele na „admin“ příkazem:

```
passwd admin
```

Pokud potřebujeme povolit přístup z jiného stroje než na, kterém běží Mldonkey démon, připojíme se k běžícímu démonovi, přes telnet konzoli a zadáme příkaz:

```
set allowed_ips "127.0.0.1 192.168.2.0/24"
```

Tímto povolíme přístup ke konzoli z adresy 127.0.0.1 a sítě 192.168.2.0/24. Stejným způsobem je možné přidat jednotlivé další adresy či sítě, ale vždy je nutné uvést jejich kompletní výčet.

Dále je nutné povolit možnost spustit pomocí Mldonkey jakýkoli příkaz:

```
set allow_any_command true
```

Nastavení je možné si zkontrolovat příkazem:

```
options security
```

Nakonec nastavení uložíme příkazem `save` a ukončíme démona příkazem `kill`. Při dalším spuštění bude již možné se k démonovi připojit z nově nastavených IP adres.

5.6.2 P2PRobot

5.6.2.1 Kompilace

Tento program je nutné zkompilovat ze zdrojového kódu, který naleznete ve složce src/

Operační systémy Windows NT

Z webové prezentace projektu MinGW získejte instalátor balíku MinGW pro platformu Win32 a binární i zdrojovou část knihovny mingw-regex . Tyto balíky lze také najít v příloze A v umístění *„install\MinGW\“*. Při instalaci zaškrtněte „g++ compiler“ a „MinGW Make“. Kompilátor nainstalujte do umístění *„C:\Program Files\MinGW“*. Do proměnné PATH systému přidejte cestu *„%ProgramFiles%\MinGW\bin“*. Knihovnu libgnurx-0.dll zkopírujte z balíku mingw-libgnurx-bin do umístění *„C:\Program Files\MinGW\bin\libgnurx-0.dll“ a „C:\Program Files\MinGW\lib\libgnurx.dll“*. Její hlavičku „regex.h“ zkopírujte z balíku mingw-libgnurx-src do umístění *„C:\Program Files\MinGW\include\regex.h“*. Nakonec restartujte operační systém.

Zdrojové kódy jsou umístěny v příloze A ve složce src. Z příkazového řádku ve složce src zadejte příkaz *"mingw32-make windows"*. Výstupem jsou tři soubory: robot.exe, robot_client.exe, tester.exe. Soubor tester.exe zkopírujte do instalačního umístění aplikace MLDonkey („*C:\Program Files\mldonkey\")*.

Operační systémy Linux

Pro kompilaci zdrojových kódů je nutné mít nainstalované tyto nástroje: GNU make, gcc, libc-dev. Pokud váš operační system používá balíčkovací system, je pravděpodobné, že tyto nástroje naleznete pod meta balíčkem build-essential. Z příkazového řádku zadejte ve složce "src/" příkaz "make". Výstupem jsou soubory: robot, robot_client, tester. Soubor tester zkopírujte do umístění „*~/.mldonkey/"*.

5.6.2.2 Nastavení

MLDonkey běží na jiném stroji než P2PRobot

V konfiguračním souboru config.ini změňte položku MLD_ADDR na hostname nebo IP adresu stroje, kde běží MLDonkey. Položku ROBOT_IP změňte na IP adresu nebo hostname P2PRobota z pohledu stroje, na kterém běží MLDonkey. Stroj, na kterém běží MLDonkey musí mít pro porty 4000 a 4001 povolenou příchozí komunikaci. To stejné platí pro stroj, kde běží P2PRobot a port 5000 a 5001.

MLDonkey běží na jiném typu systému

V takovém případě je nutné zkompilovat program tester pro příslušný systém – viz kapitola o kompilaci na daném systému. Dále je nutné v konfiguračním souboru config.ini změnit nastavení položky OS. Možné volby jsou 'posix' nebo 'win32'.

5.6.2.3 Řídící rozhranní

Chování systému lze upravit pomocí klienta robot_client.exe (platforma Win32) nebo robot_client (linux), který se spouští příkazem „robot_klient.exe adresa port". Adresou se myslí IP adresa nebo hostname P2PRobota, jako port zadejte hodnotu 5001.

Tabulka 5-6: Příkazy řídícího rozhranní P2PRobota

Volba	**Popis**
GET STATUS	Zobraz stav robota
GET OPTIONS	Zobraz volby a jejich hodnoty
SET OPTION_NAME VALUE	Nastav volbu OPTION_NAME na hodnotu

	VALUE
DOWNLOAD „klicove slova“ MIN_SIZE MAX_SIZE	Započni stahování souborů vyhovujícím zadaným klíčovým slovům a zadané minimální a maximální velikosti
CLEAN DOWNLOAD QUEUE	Zruš frontu souborů čekajících na stáhnutí
CANCEL DOWNLOADS	Zruš všechny probíhající přenosy souborů
SAVE SETTINGS	Ulož nastavení do konfiguračního souboru
LOAD SETTINGS	Načti nastavení z konfiguračního souboru
LOAD DEFAULTS	Načti nastavení z konfiguračního souboru defaults.ini
EXIT	Ukonči klienta
KILL	Ukonči robota a klienta
SHUTDOWN SYSTÉM	Ukonči MLDonkey, robota a klienta

Tabulka 5-7: Volby příkazu SET

Volba	**Rozsah hodnot**	**Popis**
SAVE_INTERVAL	time_t	Počet sekund udávající délku intervalu mezi zapisováním do souborů.
STARTUP_DELAY	time_t	Doba v sekundách po kterou robot po inicializaci spojení s MLDonkey nebude odesílat zprávy.
PREVIEW_LIMIT	int32	Maximální počet souborů stahovaných na prohlídnutí
DOWNLOAD_LIMIT	int32	Maximální počet souborů pro kompletní stáhnutí
DOWNLOAD_DELAY	time_t	Doba mezi dvoumi po sobě jdoucími požadavky na stáhnutí souboru
MIN_BYTES_PER_SEC	int32	Průměrná minimální rychlost stahování soboru. Slouží k nalezení souborů, jejichž stahování je tak pomalé, že se nevyplatí pokračovat.
TIMEOUT_RESERVE	time_t	Při nedodržení průměrné

		minimální rychlosti stahování soboru systém před zrušením stahování počká po dobu TIMEOUT_RESERVE
PREVIEW_SIZE	int32	Počet stažených bytů, které stačí k vykonání MZPE testu
LOGIN	Řetězec znaků	Login pro přihlášení do démona MLDonkey
PASS	Řetězec znaků	Heslo pro přihlášení do démona MLDonkey
DLLINK_FILE_PATH	Řetězec znaků	Cesta k souboru, kam se ukládá fronta souborů na stažení
LOG_DELETED	Řetězec znaků	Cesta k logu nedokončených souborů
LOG_PREVIEWED	Řetězec znaků	Cesta k logu prohlíhnutých souborů, které neprošli MZPE testem
LOG_DOWNLOADED	Řetězec znaků	Cesta k logu dokončených souborů
MLD_ADDR	IPv4 adresa nebo platný hostname	IPv4 adresa démona MLDonkey
MLD_PORT	uint16	Port, na kterém MLDonkey přijímá příchozí spojení
ROBOT_ADDR	IPv4 adresa nebo platný hostname	IPv4 adresa P2PRobota z pohledu démona MLDonkey
MLD_HOST	Řetězec „posix“ nebo „win32“	Informace o systému, na kterém běží MLDonkey

Všechny cesty k souborům jsou relativní vzhledem k umístění, ze kterého se spouští P2PRobot.

5.7 Spuštění systému

Operační systémy Windows NT

Nejdříve spusťte MLDonkey pomocí run_mldonkey.exe. Následně můžete spustit z libovolného umístění P2PRobota. P2PRobot se spouští příkazem příkazem „robot.exe config.ini“. Je možné zadat cestu na jiný konfigurační soubor. Pokud nezadáte cestu, robot načte konfigurační soubor defaults.ini. Pokud nenalezne ani ten, použije výchozí hodnoty.

Pokud P2PRobot při spuštění spadne s návratovým kódem -1073741515, znamená to, že systém nemohl najít potřebné knihovny pro správný běh. Tento problém odstraníte zkopírováním knihovny libgnurx-0.dll do stejného adresáře, ze kterého spouštíte P2PRobota.

V případě nečekaného pádu operačního systému je zapotřebí před spuštěním aplikace smazat soubor mlnet.exe.pid, který se nachází ve složce instalace démona Mldonkey.

Zdali spuštění démona proběhlo v pořádku, je možné zkontrolovat v konzolovém okně, které se objeví po spuštění příkazu run_mldonkey.exe. Případné chybové stavy je možné vyřešit pomocí dokumentace Mldonkey, která se nachází na oficiální webové prezentaci Mldonkey.

Chovnání systému je možné měnit pomočí řídící konzole, která se spouští souborem robot_client.exe.

Operační systémy Linux

Nejdříve spusťte MLDonkey souborem mlnet. P2PRobot se spouští příkazem „robot config.ini“. Chovnání systému je možné měnit pomočí řídící konzole, která se spouští souborem robot_client.

5.8 Používání systému

Po spuštění systému se připojte na řídící rozhranní pomocí souboru robot_client(.exe). Příkazy řídícího rozhranní jsou popsány v kapitole 5.6.2.3. Příkazy je možné také zobrazit příkazem „HELP“.

Příkazem „GET OPTIONS“ zkontrolujte nastavení systému. Pokud potřebujete změnit nastavení systému, použijte příkaz SET. Příkazem DOWNLOAD (syntaxe příkazu viz kapitola 5.6.2.3) je započnuto vyhledávání a stahování souborů. Průběh stahování souborů je možné sledovat pomocí GUI klienta jMoule (viz kapitola 5.1.2) nebo přímo přes telnet konzoli démona Mldonkey, příkazem „vd“ (viz kapitola 5.1.1).

Klienta ukončíte příkazem „EXIT“. Pokud chcete zrušit frontu souborů, které čekají na stáhnutí, použijte příkaz „CLEAN DOWNLOAD QUEUE“. Všechny stahované a stáhnuté soubory můžete odstranit příkazem „CANCEL DOWNLOADS“. Systém ukončíte příkazem „SHUTDOWN SYSTEM“.

Na operačních systémech s jádrem Linux naleznete stažené soubory ve složce „~/.mldonkey/incoming/files“. Na operačních systémech Windows NT se složka „incoming“ nachází v umístění instalace démona Mldonkey.

Rozšíření P2P sítě eDonkey „KAD“ používá pro hodnocení uživatelů bodový systém. Díky tomuto systému může, při prvním přihlášení do sítě, trvat několik hodin, než je započato stahování prvního souboru. S každým stáhnutým souborem se tento interval snižuje.

6 Závěr

Výsledkem práce je analýza typů P2P sítí a protokolů, které používají. Jako nejvhodnější kandidát pro automatické odhalování nebezpečných souborů byl vybrán protokol eDonkey. Po důkladné analýze tohoto protokolu bylo zjištěno, že jeho implementace je nad časový rámec bakalářské práce. Proto byl použit open source démon MLDonkey. Předmětem implementace se tedy stal P2PRobot řídící navržený systém pro procházení P2P sítí. Bylo nutné se důkladně seznámit s GUI protokolem démona MLDonkey a tento protokol implementovat. Díky tomu, že systém musí být schopný rozlišit, zda soubor může být potenciálně nebezpečný, bylo potřeba nastudovat problematiku testování spustitelnosti souborů na platformě Windows NT a tento test implementovat. Všechny části systému spolu komunikují pomocí protokolu TCP. Bylo tedy nutné důkladně nastudovat a implementovat jak klientskou, tak serverovou část síťové komunikace. Vzhledem k tomu, že podmínkou zadavatele byla aplikace, běžící na Windows NT a operačních systémech s jádrem Linux, bylo třeba nastudovat problematiku multiplatformního programování a tyto znalosti použít při implementaci.

Výsledný systém je otestován na systémech Windows XP a Debian, splňuje všechny požadavky zadavatele a pro zhodnocení jeho přínosu je třeba delšího časového období. Implementace tohoto systému má rozsah 212kB zdrojových kódů.

Literatura

[1] Hewlett-Packard Company, BSD Sockets Interface Programmer's Guide, Edition 6, 1997.
196 stran.

[2] Yoram Kulbak and Danny Bickson, The eMule Protocol Specification, The Hebrew University of Jerusalem, Israel, 2006
68 stran.

[3] Jian Liang, Rakesh Kumar, Keith W. Ross, Understanding KaZaA, Polytechnic University Brooklyn, NY 11201
7 stran.

[4] Andrew Kalafut, Abhinav Acharya, Minaxi Gupta, A Study of Malware in PeertoPeer Networks, University Bloomington, Indiana, USA
6 stran.

[5] 6 Most Widely Used Free P2P File Sharing Applications, 26. Listopadu 2007
URL: http://www.bizzntech.com/2007/11/26/6-most-widely-used-free-p2p-file-sharing-applications

[6] P2P Search Engines, Březen 2003
URL: http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p8.html

[7] How Gnutella Works, 2004
URL: http://computer.howstuffworks.com/file-sharing.htm/printable

[8] Position on Peer-to-Peer File Sharing, 2005

URL: http://www.umkc.edu/is/security/p2p_explanation.asp

[9] List of P2P protocols, 2009

URL: http://protocolinfo.org/wiki/List_of_P2P_protocols

[10] Known P2P Viruses, 2006

URL: http://www.kazaa.com/us/help/known_virus.htm

[11] MLDonkey Project Wiki, 2009

URL: http://mldonkey.sourceforge.net/

[12] Tiny PE, 2009

URL: http://www.phreedom.org/solar/code/tinype/

[13] TCP klient v MS Windows, 2003

URL: http://www.builder.cz/art/cpp/tcp_klient_windows.html

[14] Minimalist GNU for Windows, 2009

URL: http://www.mingw.org/

[15] pDonkey, 2009

URL: http://sourceforge.net/projects/pdonkey/

Přílohy

A. DVD obsahující
   - Zdrojové kódy projektu P2PRobot - složka „src“
   - Binární soubory potřebné ke spuštění démona MLDonkey na systémech Windows NT – složka „install/mldonkey/win32“
   - Zdrojové soubory projektu MLDonkey potřebné ke kompilaci spustitelného souboru na systémech s jádrem Linux – složka „install/mldonkey/linux“
   - Konfigurační soubory – složka „src“
   - Binární soubory potřebné ke kompilaci P2PRobota na systémech Windows NT – složka „install/mingw“
   - Uživatelská příručka programu – soubor instalace.pdf
   - Zdrojový text této zprávy – soubor zprava.pdf a zprava.doc