# Peer-to-peer (P2P) systémy a jejich bezpečnost

Ladislav Beránek

Katedra informatiky, Pedagogická fakulta, Jihočeská universita,
Jeronýmova 10, 37001 České Budějovice
beranek@pf.jcu.cz

**Abstrakt**

V současné době vzrůstá obliba různých peer-to-peer (dále P2P) systémů, zejména těch, které jsou vytvořené na infrastruktuře Internetu, na protokolech TCP/IP. Tyto systémy jsou stále oblíbenější, protože umožňují vytvoření kooperativních skupin uživatelů, anonymní sdílení souborů a případně i netradiční způsoby řešení dostupnosti informací či způsobů autentizace. Objevují se nové systémy P2P a jsou vytvářeny nástroje a knihovny umožňující programování těchto systémů. V následujícím textu budou popsány základní principy P2P systémů s cílem prezentovat typické vlastnosti peer-to-peer systémů, které mohou pomoci při programování P2P systémů, naznačit problematiku programování v jazyce Java na standardech JXTA a popsat některé bezpečnostní aspekty těchto systémů.

## 1. ZÁKLADNÍ PRINCIPY PEER-TO-PEER SYSTÉMŮ:

Základní principy sítí peer-to-peer se dají shrnout do několika bodů:

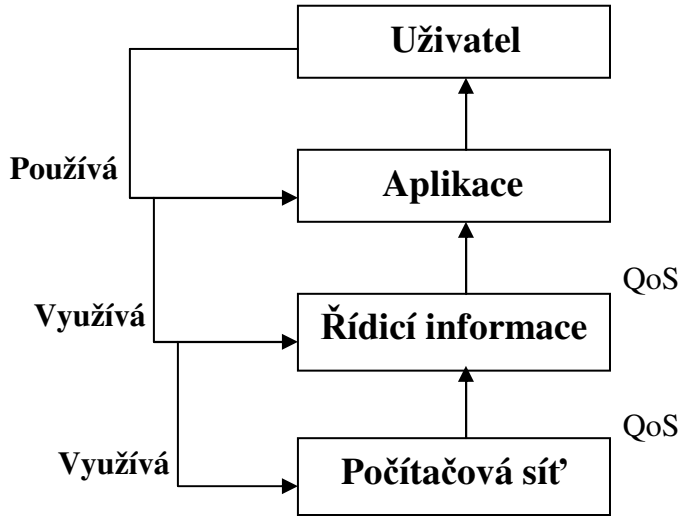
- princip sdílení zdrojů – každý účastník P2P systému (uzel) poskytuje některé zdroje ostatním uzlům (účastníkům) sítě peer-to-peer. Tyto zdroje mohou být fyzické jako je např. výpočetní výkon svého počítače nebo diskový prostor (resp. přístup k souborům na svém disku) nebo logické jako je např. určitý druh služby nebo specifická znalost,
- princip decentralizace – části systému nebo dokonce celý systém nejsou řízeny centrálním prvkem. Každý prvek systému funguje jako klient i server. Příkladem plně decentralizovaného systému je např. Gnutellea a Freenet. Způsob adresování klientů v P2P systémech je nezávislé na DNS systému, který je centralizovaný a hierarchický.
- princip samoorganizace – předchozí princip znamená, že v plně decentralizovaném systému P2P neexistuje centrální prvek, který by centrálně koordinoval aktivity nebo který by udržoval centrální databázi o celém P2P systému. Systém vykazuje samoorganizační chování, které je založeno na tom, jakou informaci mají jednotlivé uzly o okolních uzlech systému a jakým způsobem si tyto informace uzly vyměňují.

Kromě toho architektura a funkce systémů P2P někdy zahrnuje takové další hlediska jako je např. požadavek účastníků na to, aby zůstali anonymní nebo skutečnost, že uzly v P2P systémech jsou často nespolehlivé např. z hlediska jejich stálé dostupnosti apod.

## 2. ARCHITEKTURA P2P SYSTÉMŮ

V minulosti celý Internet (Arpanet) fungoval jako P2P systém. Každý uživatel Internetu se mohl pomocí služby telnet spojit přímo s jiným uživatelem a pomocí služby ftp si od něho případně stáhnout požadované soubory. Současné systémy mobilních telefonů nebo některé systémy pro e-commerce (např. eBAY) nesou znaky P2P systémů. Největší popularitu však P2P systémy získaly existencí systémů pro sdílení souborů (zejména s hudebními skladbami) jako je Napster, Gnutella nebo Freenet a další.

Architekturu P2P systémů na infrastruktuře lze popsat následujícím schématem:



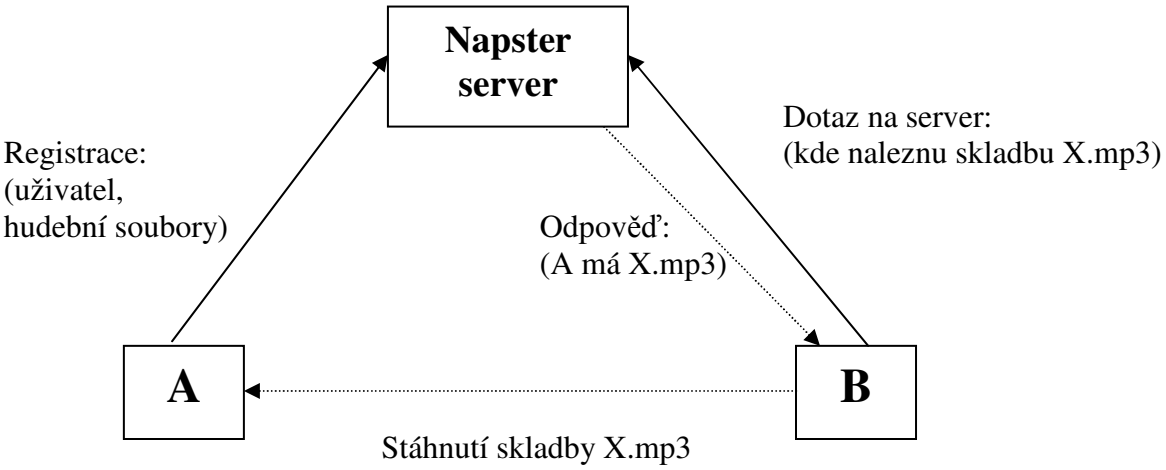

Obr. 1: Vrstvová architektura P2P systémů

Na úrovni síťové vrstvy (vrstva Počítačová síť) je Internet, který je svou podstatou P2P systémem. Každý prvek se může přímo spojit s jiným prvkem.
Vrstva obsahující řídicí informace je vrstvou, která se stará o informace o členech daného P2P systému. Je vytvářena adresáři a databázemi. Její konkrétní implementace může být buď centralizovaná nebo decentralizovaná.
Aplikační vrstva obsahuje logiku např. pro e-komerční systém nebo pro službu, která je pomocí P2P systému zajišťována. Opět může v konkrétní implementaci být centralizována nebo decentralizována.
Funkce aplikační vrstvy jsou přístupné jednotlivým uživatelům využívajícím služby systémů P2P.
Protokoly systémů P2P nejsou dosud standardizovány a proto se používají zpravidla proprietální protokoly. Např. AOL Messenger a ICQ používají proprietální protokol vyvinutý AOL. Kazaa, Morpheus a Music Sity využívají protokol FastTrack. Gnutella používá open source protokol gnutella.
V následujících odstavcích popíšeme nejznámější P2P systémy, jejich základní architekturu a fungování. Princip fungování dalších P2P systémů je podobný těmto popsaným systémům.

### 2.1 Napster

Z technického hlediska je Napster velmi jednoduchý centralizovaný P2P systém. Centrální server Napsteru udržuje centrální databázi hudebních skladeb ve formátu MP3/WMA, které nabízí účastníci připojení do systému. Účastníci se přihlašují k serveru a posílají seznam skladeb, které nabízejí. Noví uživatelé musí nejprve zřídit účet u centrálního serveru Napsteru. Každý uživatel může poslat dotaz na vyhledání serveru Napster a obdrží seznam uživatelů, kteří nabízejí poptávané skladby. Poptávající si může vybrat z tohoto seznamu toho, od něhož se bude snažit získat hledanou skladbu. Poté mu pošle požadavek a

stáhne si od něho požadovaný soubor s hudební skladbou. Napsater tedy není čistý P2P systém, ale kombinace architektury klient/server a P2P. Schéma systému je vidět z následujícího obrázku.




Obr. 2: schéma P2P systému Napster

Protokol, který Napster používá ve svých interakcích nebyl nikdy publikován. Existují definice protokolu, avšak ne příliš podrobně popsané, jako např. [3]. Z dostupných údajů však lze soudit, že protokol je komplikovaný a nepříliš konzistentní.

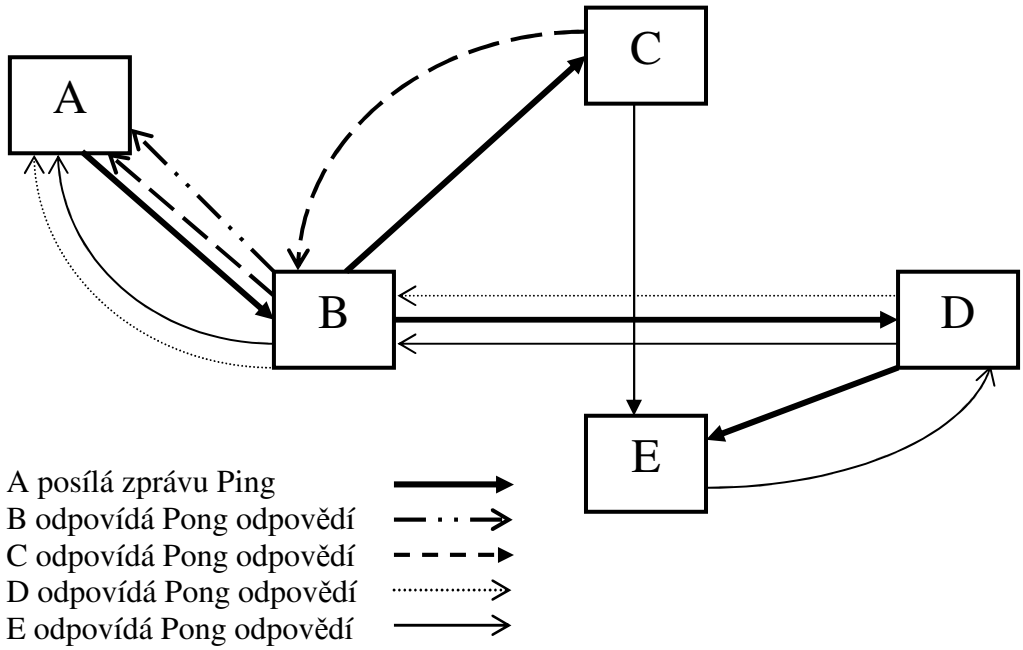
### 2.2 Gnutella

Z technického hlediska je Gnutella decentralizovaný systém pro sdílení souborů, jehož účastníci vytvářejí virtuální síť pomocí protokolu Gnutella, což je jednoduchý protokol pro vyhledávání distribuovaných souborů. Na začátku se však musí ten, kdo má zájem o účast v systému, spojit s někým, kde je již v systému Gnutella zaregistrovaný. Tato registrace ale není součástí standarního protokolu Gnutelly. Protokol samotný sestává z několika základních zpráv:

- ping – oznamuje dostupnost a testuje dostupnost ostatních účastníků. Nenese žádnou informaci,
- pong – odpověď na Ping. Nese informaci o IP adrese a čísle portu odpovídajících účastníků, dále informaci o počtu a celkové velikosti sdílených souborů v kB,
- Query – dotaz při vyhledávání,
- QueryHit – vracejí účastníci, kteří mají požadovaný soubor. Nese informaci o jejich IP adrese, čísle portu, dále nese informaci o výsledku vyhledávání,
- Push – požadavek na stáhnutí požadovaného souboru. Nese informaci o identifikaci účastníka, který má požadovaný soubor, index požadovaného souboru, IP adresu a číslo portu účastníka, kam se má soubor poslat.

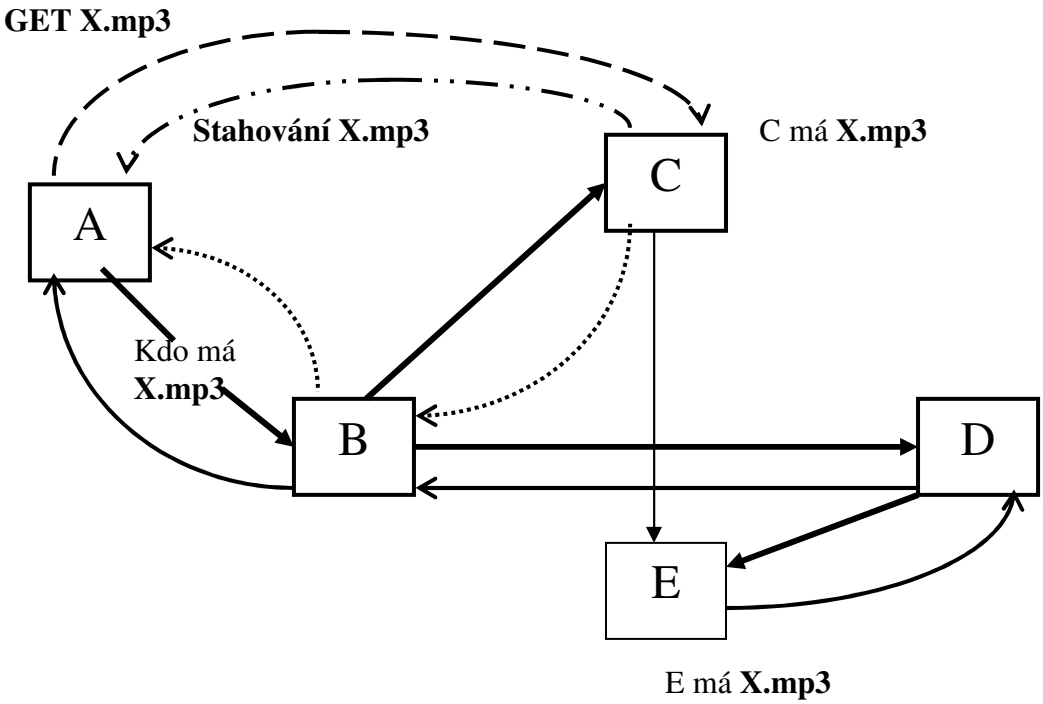
Tyto zprávy jsou předávány všem účastníkům. Používá se přitom zjednodušený mechanismus všesměrového vysílání. Poté, co uživatel přijme zprávu, sníží hodnotu TTL pole zprávy (pole

time-to-live). Jestliže uživatel přijme zprávu s dotazem Query, zkontroluje nejdříve své lokální úložiště souborů a jestliže najde poptávaný soubor, odpoví zprávou QueryHit. Současně uživatel, pokud je TTL větší než 0, přepošle zprávu všem dalším uživatelům P2P sítě, které zná. Odpověď je posílána stejnou cestou, jakou byl poslán dotaz.
Zjednodušené navazování spojení v systému Gnutelly vypadá následovně:
Účastník A kontaktuje účastníka B a posílá zprávu Ping. B odpovídá zprávou Pong a přeposílá zprávu Ping dalším účastníkům C a D. Tito účastníci odpovídají zprávou Pong a přeposílají Ping … Po nějaké době zná A počet aktivních účastníků.




Obr. 3: Gnutella – navazování spojení

Postup přeposílání inicializačních zpráv Ping/Pong popsaný nahoře může být použit i v případě odesílání dotazů. Když účastník obdrží odpověď QueryHit, nejprve se pokusí spojit s účastníkem uvedeným v QueryHIt a pomocí zjednodušené metody http GET se snaží získat soubor. Jestliže hledaný účastník je za firewallem, může poslat Push zprávu (stejnou cestou, kterou održel QueryHit) účastníku za firewallem. Zpráva Push specifikuje, kde účastník za firewallem může kontaktovat iniciátora dotazu pomocí „pasivní“ GET session. Jestliže oba účastníci jsou za firewallem, není stáhnutí souboru možné.




Obr. 4: Gnutella - vyhledávací mechanismus

**2.3 Další P2P systémy**

Freenet je P2P systém určený pro publikaci, sdílení a hledání souborů dat. Jeho cílem je poskytnutí infrastruktury, která chrání anonymitu autorů a čtenářů. Architektura je navržena tak, aby nebylo možné snadno zjistit původ souboru nebo místo původu dotazu při hledání souborů dat. Vzhledem k tomu, že data jsou šifrována, a to jak uložená data tak i posílaná, je obtížné vůbec zjistit, co se v souborech skrývá. Vedle tohoto aspektu ochrany anonymity má Freenet ještě jeden zajímavý koncept: adaptivní způsob směrovacího schématu pro efektivní směrovací dotaz na fyzické umístění účastníků, kteří s největší pravděpodobností nabízejí poptávaný soubor. Freenet udržuje směrovací tabulky, které jsou dynamicky měněny při hledání a stahování dat, aby se zlepšila účinnost hledání. Freenet také používá dynamické replikace populárnějších souborů tak, že tyto soubory migrují mezi účastníky a mohou být tak nalezeny rychleji a s větší pravděpodobností. Protokol Freenet nevyžaduje centrální server jako Napsater a ve srovnání s Gnutellou nepoužívá méně efektivního mechanismu broadcástů. Existuje další množství P2P systémů sítí, např. komerčního charakteru FastTrack nebo výzkumné peer-to-peer sítě např. Chord, Pastry a P-Grid a další jako např. ICQ aj. Jak již bylo řečeno principy jejich základního fungování se příliš neliší od principů systému popsaných v předcházejících odstavcích. Tyto sítě mohou garantovat odpověď na dotaz nebo mohou vytvářet samo-organizovanou překryvnou síť. Obsahují adresování dle hašovaného obsahu souborů, adaptivní způsoby směrování, dynamické cachování, replikace a automatický failover, multicástové kaskády pro skupinovou komunikaci, šifrování aj.

## 3. PROGRAMOVÁNÍ P2P SYSTÉMŮ

IT komunita rozeznala potenciál P2P systémů a již v roce 2000 se pod vedením Intelu vytvořila skupina s cílem vytvořit nový standard pro vyvíjející se trh P2P. Také firma SUN se snaží o vytvoření standardů P2P systémů. Jedná se o projekt JXTA, jehož cílem je navržení minimálního standardu P2P systému, který může být modifikován podle účelu, ke kterému P2P systém bude navrhován. Podrobně je projekt JXTA popsán např. v [5]. Jedná se tedy o projekt pro vytvoření standardů peer-to-peer sítí, které by umožnily snadné propojování zařízení (mobilní telefony, PDA, PC k serveru na síti) za účelem komunikace a spolupráce.
Na projekt JXTA navazuje projekt vytváření P2P socketů [8]. P2P sockety, které se snažíme používat k vytváření našich P2P aplikací, jsou právě určeny pro snadné vytváření P2P aplikací. Jsou založeny na standardu JXTA a nativně umožňují průchod NAT nebo firewaly, aniž se programátor musí dopodrobna touto problematikou zabývat zabývat. Používají porty používané pro webové servere nebo webové služby.
P2P sokety pracují na základě JXTA peer-to-peer sítě. Ta zahrnuje webový server (Jetty), servlet a JSP engine (Jetty a JAsper) která umožňuje, aby P2P klienty mohly používat existující servlety a JSP. Dále využívají XML-RPC klienta a server (Apache XML-RPC) pro dosažení a využití P2P koncových bodů XML-RPC, HTTP/1.1 klienta (Apache Common HTTP-Client), který může kontaktovat P2P webový server. P2P sokety také zavádějí implementaci java.net.Socket a java.net.ServerSocket které mohou vytvářet na síti JXTA něco jako distribuovaný uživatelsky přívětivý nezabezpečený DNS systém.
Pomocí těchto soketů na JXTA se P2P aplikace dají programovat. Například chceme-li naprogramovat, aby člen P2P sítě inzerovat jednoduchou službu „www.jcu.programovani“, která může být dosažena jinými účastníky P2P sítě, vytvoříme následující kód:

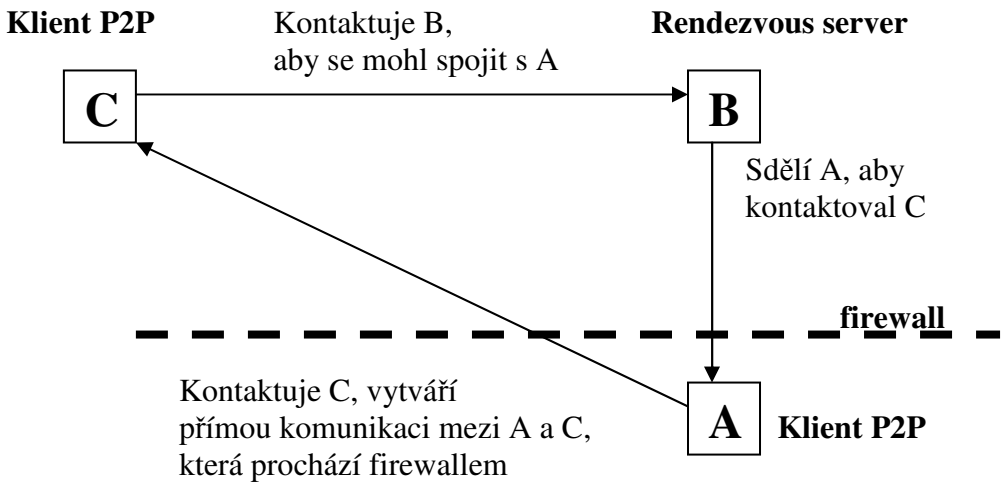
```
// člen sítě vytvoří službu jménem "www.jcu.programovani" na virtuálním portu 100
java.net.ServerSocket server = new P2PServerSocket("www.jcu.programovani ", 100);
// na portu 100 čeká na spojení s klientem
java.net.Socket client = server.accept();
// klient je akceptován; nyní probíhá komunikace
InputStream in = client.getInputStream();
OutputStream out = client.getOutputStream();
// aplikace může nyní provádět specifické funkce
```

Strana klienta je také tak jednoduchá

```
// vytvoř spojení se členem nabízející službu na " www.jcu.programovani " na virtuálním portu 100
java.net.Socket socket = new P2PSocket("www.jcu.programovani ", 100);
// nyní začni komunikovat
InputStream in = socket.getInputStream();
OutputStream out = socket.getOutputStream();
```

Účastníci sítě mohou nyní spolu komunikovat a využívat služby, i v případě, že jsou nedosažitelní za NAT nebo firewallem. Mohou publikovat své služby na P2P distribuovaném jmenném systému, aniž by k tomu potřebovaly nakonfigurovaný DNS server.
Zde byla zmíněna jednoduchá ukázka, programování P2P aplikací pomocí P2P soketů je samozřejmě složitější. Nicméně stojí za to, se tím zabývat, problematika je to velmi zajímavá a může mít spousty aplikací.

Co se týče principu obejití ochranných síťových prvků: projekt JXTA, jak již bylo naznačeno, podporuje činnost rendezvouz serverů nebo "web" proxy. Takový server může jednak pomáhat jiným uzlům, aby mohly iniciovat spojení a jednak pracovat jako proxy, která směruje informace mezi členy P2P systému.
P2P sokety jsou založeny na použití HTTP protokolu, který používá jako standardní nebo pro tunelování svého vlastního protokolu. Spojení protokolem http firewally zpravidla povolují. Pokud je však firewall dobře nakonfigurovaný, neumožní však spojení pomocí HTTP protokolu, které je iniciováno zvenku. P2P sítě JXTA však toto umějí obejít. Přístup je založen na veřejně adresovatelném uzlu nazývaném rendezvous server, s kterým se účastnící chránění firewallem již mohou spojit. Takový uzel funguje jako prostředník, který umožní uzlu uvnitř sítě chráněné firewallem komunikaci s uzlem vně této sítě.




Obr. 5: Vytváření spojení mezi servery pomocí rendezvous serveru

## 4. BEZPEČNOST P2P SYSTÉMŮ

Současné P2P systémy nemají v sobě zabudovány bezpečnostní mechanismy, které se standardně používají v architektuře klient/server. P2P systémy určené pro malé skupiny lidí, kteří se většinou znají mezi sebou, se sice mohou bez těchto mechanismů obejít, ale v současné době se P2P systémy rozšířily v prostředí Internetu, kde je mohou využívat i miliony anonymních uživatelů. Přesto bezpečnostní mechanismy implementované v P2P systémech zůstávají omezené, přestože na druhé straně v sobě P2P systémy zahrnují stále více sofistikovaných funkcí, jako je například zajištění šířky pásma při stahování souborů pro daného uživatele a další funkce.
Výsledkem uvedené situace je to, že provozování P2P systémů s sebou nese zvýšená bezpečnostní rizika a to zejména, jsou-li P2P systémy používány v rámci podnikových počítačových sítí.
Problematiku bezpečnosti P2P systémů můžeme rozdělit do dvou základních oblastí [9]:

- Bezpečnostní rizika vyplývající z architektury a fungování P2P systémů (tj. zejména situace, kdy pracovní stanice účastníka P2P systému se stává v P2P systému klientem a zároveň serverem, a to nezabezpečeným serverem). Důsledkem je zejména možnost úniku privátních dat ze sdílených disků uživatelů P2P systémů, nízká obranyschopnost proti škodlivému software (malware), jako jsou viry, červi, trojské koně a software typu spyware, a proti útokům typu DoS (Denial of Services).

- Bezpečnostní rizika vyplývající z bezpečnostních nedostatků jednotlivých P2P aplikací nebo z činnosti uživatelů, kteří z největší míry z neznalosti, z podcenění nebezpečí v důsledku a nízkého bezpečnostního povědomí instalují P2P klienty na podnikové počítače. P2P systémy mohou být provozovány tak, aby obcházely firewally, antivirovou kontrolu a umožnili anonymitu v rámci P2P systému (anonymní přístup do systému, nemožnost sledování provozu). Z toho plynou další hrozby specifické pro P2P systémy jako jsou možnost odkrytí IP nebo MAC adres, plýtvání přenosovými kapacitami, možnost prozrazení privátních dat a další.

## 4.1 Přehled bezpečnostních rizik P2P systémů

### 4.1.1 Škodlivý software

Současné P2P systémy umožňují snadnou distribuci škodlivého software (malware), jako jsou viry, červi, trojské koně a další. Většina P2P systémů totiž vytváří přímé tunely a tak obejdou zabezpečení, kterou poskytují NAT a proxy servery. Je zřizováno přímé spojení na uživatelské úrovni. To je daleko nebezpečnější než e-mail, protože spojení je přímé a živé. Může tak dojít k snadnému zavlečení nejen virů, červů, trojských koní ale i software, které umožní odposlouchávání dat (spyware), například obchodní data nebo přístupová hesla a které může být i obtížné odstranit.
Jako příklad snadné distribuce škodlivého software na P2P systémech lze uvést systém stahování souborů na P2P systému FreeNet. Zde je poptávaný soubor kopírován postupně na stanice účastníků systémů, až se dostane na stanici účastníka, který daný soubor požadoval. Nelze si představit lepší mechanismus pro šíření škodlivého software.

### 4.1.2 Riziko – prozrazení dat

Jedno z největších rizik provozu P2P systémů na podnikových počítačích je krádež dat. Uživatelé většiny P2P systémů musejí nabídnout data (např. hudební soubory) nebo určitou část své diskové kapacity do P2P systému. Nebezpečí z takového sdílení diskových kapacit a dat je zřejmé.

Mezi citlivá data patří ale i adresy stanic účastníků. Jak již bylo uvedeno, většina P2P systémů je navržena tak, aby bylo možné obcházet firewally, antivirovou kontrolu, vytvářejí přímé tunely, čímž obejdou zabezpečení, kterou poskytují NAT a proxy servery. P2P systémy umožňují získat IP nebo MAC adresy účastníků nebo i údaje o rychlosti připojení účastníků. Například Gnutella umožňuje zobrazit IP adresu stanice, ze které si účastník stahuje požadovaný soubor. To je dobrý výchozí bod pro potenciálního útočníka.

### 4.1.3 Útoky typu DoS

Každý účastník P2P aplikace při své činnosti v rámci P2P systému potřebuje určitou šířku pásma. P2P systémy jako jsou Napster, Gnutella a další jsou využívány pro získání souborů, hudebních, obrazových textových a dalších. Tyto soubory formátu MP3, AVI, MPG, JPG, GIF a další jsou většinou velmi velké, desítky i stovky MB. Již samo stahování souborů několika účastníky může způsobit problémy na počítačové síti nehledě na cílené útoky typu DoS, ke kterým je P2P systém vzhledem ke své architektuře citlivý.

### 4.1.4 Bezpečnostní rizika jednotlivých aplikací

U většiny současných P2P systémů jsou bezpečnostní funkce slabé, testování na bezpečnost aplikace se neprovádí nebo se provádí povrchně. Např. některé P2P systémy jsou

navrženy tak, aby maximalizovaly sdílení diskových kapacit účastníků. Zatímco dřívější verze P2P systému Kazaa pouze vybízely k většímu sdílení kapacity lokálního disku, ale bez defalutní volby, současné verze vytvářejí při instalaci klienta systému překryvné okno požadující nastavení sdílení diskové kapacita, přičemž defaultně je nastaveno sdílení celého disku. Nepozorný uživatel se tak může dostat do situace, že celý jeho disk bude sdílený v P2P systému. Jaké z toho nebezpečí hrozí pro jeho data je zřejmé. Podobných bezpečnostních slabin nebo „opomenutí“ tvůrce P2P systémů by se dalo nalézt více.
Problematika zabezpečení P2P aplikací však není jednoduchá a nedostatky mohou vyplývat i z nedokonalosti použité platformy. Například aplikace vytvářené pomocí zmíněných P2P soketů na JXTA síti mají některé bezpečnostní nedostatky, které vyplývají z toho, že vývoj těchto soketů je v určitém stadiu vývoje:

- snadný spoofing jmen účastníků P2P sítě a IP adres (neexistuje žádný mechanismus, který by jednoznačně přiřadil specifickému účastníku jméno nebo IP adresu),
- síť je náchylná k útokům typu DoS, kdy účastník zaplaví síť požadavky nebo pokusy o vytváření server soketů,
- P2P sokety v současné době ještě neodpovídají JVM Security Manager architektuře. Jakmile je účastník připojen k P2P síti, jiní účastníci mohou využít nedostatků vrstvy P2P soketů k jeho kompromitaci.

**LITERATURA**

[1] A. Oram et. All. Peer-to-peer: Harnessing the Power of Disruptive Technologies. O’Reilly&Association, March 2001
[2] Napster homepage. http://www.napster.com
[3] Napster protocol specification, April 7 2001. http://opennap.sourceforge.net/napster.txt
[4] Gnutella protocol specification, *www.gnutella.com*
[5] Sun Microsystems' JXTA – *platform.jxta.org*
[6] The Peer-to-Peer Working Group – *www.p2pwg.org*
[7] p2psockets.jxta.org/docs/tutorials
[8] www.onjava.com/pub/a/onjava/2003/12/03/p2psockets.htm
[9] Beránek, L.: Peer-to-peer sítě a jejich bezpečnost II., DSM, 2005, č.1, roč. IX, s. 34-36