P2P hálózatok

Párhuzamos és Elosztott Rendszerek II

Készítette: Dr. Mileff Péter
Miskolci Egyetem Általános Informatikai Tanszék

1. Bevezetés

Napjainkban megkérdőjelezhetetlen az elosztott rendszerek jelentőssége. Az elosztott rendszereken, mint általános csoporton belül is a közismert többrétegű architektúrák mellett egyre inkább tért nyernek az *egyenrangú résztvevők együttműködésén alapuló*, úgynevezett Peer-to-Peer (P2P) rendszerek.

A peer-to-peer vagy P2P paradigma lényege, hogy az informatikai hálózat végpontjai közvetlenül egymással kommunikálnak, központi kitüntetett csomópont nélkül. A peer-to-peer fogalom két hasonló, de célját tekintve mégis eltérő fogalomkört is takar: a számítógépek egyenrangú technológiai szintű kapcsolódási módját egy helyi hálózaton, valamint valamilyen célból közvetlenül kapcsolódó szoftver megoldások működési elvét. A közvetlen kapcsolat hibatűrőbb felépítést, skálázhatóságot jelent. Hátrányai: a nehezebb adminisztráció, az erőforrások pazarló használata, a nehezebb megvalósíthatóság.

A P2P hálózatok az ügyfél-kiszolgáló kapcsolathoz képest jelentősen eltérő módon működnek: a szerepek nincsenek előre meghatározva; többnyire követelmény is, hogy az összes résztvevő képes legyen valamilyen erőforrást a rendszer egésze számára elérhetővé tenni viszonzásképp az általa igénybevett szolgáltatásokért. Az így megosztható erőforrások általában a következő három kategóriába sorolhatók: *fájlok*, *számítási kapacitás*, *felhasználói jelenlét* (legegyszerűbb esetben csevegés).

2. Peer-to-Peer rendszerek felépítése

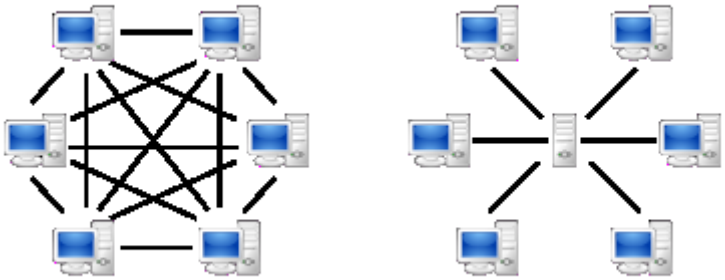
A P2P rendszerek az aktuális feladattól függetlenül egy úgynevezett **overlay** („fölöttes”) network létrehozásával működnek. Erre azért van szükség, mert ezek a rendszerek alapvetően szomszédossági alapon működnek: hogy ne legyen szükség kitűntetett szerepű résztvevőkre, ugyanakkor elegendő legyen, hogy minden node csak korlátos számú másikkal tartson kapcsolatot, az üzenetek továbbítása logikai pont-pont kapcsolatokon keresztül történik.

Ez a működési mód igazából nem újdonság: az „ős-Internet” pontosan ugyanígy működött, minden csomópont útvonalválasztóként is funkcionált és az ügyfél-kiszolgáló szerepek sem különültek el. A mai Internet már erősen specializált, ez a három szerep tipikusan szeparáltan valósul meg. Emiatt, és azon tény miatt, hogy az Internet közösségének jelentős hányada nem képezi részét egy adott P2P rendszernek, szükséges a saját logikai hálózat kialakítása saját üzenettovábbítási mechanizmusokkal.

A saját útvonalválasztó és üzenettovábbító alrendszer szükségessége természetesen munka, viszont egyben szabadságot is jelent: az adott applikáció számára protokollszinten biztosíthatunk támogatást (ez például a tartalom szerinti route-olás lehetőségét jelenti), illetve a megcélzott hardver eszközök és átviteli közeg ismeretében dönthetünk, hogy milyen routing algoritmust alkalmazunk (egy adott algoritmus számításigénye és az általa generált forgalom a gyakorlati tapasztalatok alapján fordítottan arányosnak tekinthető).

A P2P rendszerekben az útvonalválasztás ill. a protokoll nem választható el az alkalmazástól, hiszen egy file-sharing alkalmazás esetében elegendő, ha legalább egy példányt megtalálunk egy adott állományból, míg egy csevegés-jellegű (chat, whiteboard, vagy bármilyen applikáció-megosztás) együttműködés esetében értelemszerűen szükséges, hogy az összes résztvevőhöz eljussanak az események. Ezen okok miatt általános protokollszintű optimalizációról **nem** beszélhetünk.

A P2P már az Internet születésénél is jelen volt. Gazdaságossági és technikai okokból a lapos hierarchia folyamatosan kezdett központosodni, míg az Internet széleskörű elterjedésének idejére már túlnyomórészt a szerver-kliens architektúra volt jellemző. A peer-to-peer rendszerek reneszánszát a sávszélesség növekedésével és a tömörítési algoritmusok (és az ehhez szükséges processzorok) javulásával berobbanó fájlcserélő alkalmazások hozták el.



1. ábra. Peer-to-Peer alapú hálózat (bal) és szerver alapú hálózat (jobb) felépítése

2.1 A Napster és a szerzői jogi problémák

Első fájlcserélő klasszikusként a Napster jelent meg, és hihetetlen sebességgel tett szert több millió felhasználóra. Architektúráját tekinteve még hibrid, központi szerverfarm kapcsolja össze a felhasználóktól jött file-kéréseket a file felajánlásokkal, de maga az adatcsere a felhasználók között már a szervertől függetlenül zajlik. A nagy népszerűség mellett sok per is indult a rendszer ellen a szerzői jogok tulajdonosai, ez végül a Napster bezárását eredményezte. A fájlcserélők és ezzel a peer-to-peer szoftverek terjedésének ezzel azonban már nem lehetett gátat vetni.

2.2 A P2P hálózatok csoportosítása

A peer-to-peer hálózatokat több szempont alapján lehet csoportosítani. Elsőként beszélhetünk arról, hogy a P2P hálózatok csoportosíthatók **a felhasználási célok** alapján:

- Fájlmegosztás.

- Telefon.
- Videó és audio média „sztrimelése” (streaming).
- Fórumok.

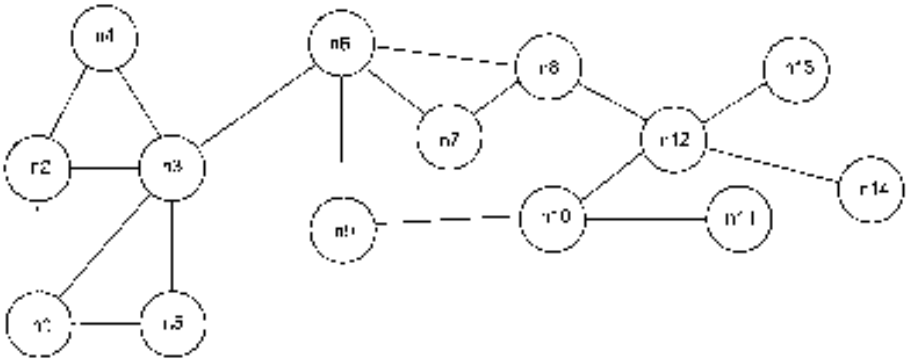
Egy másik csoportosítás lehet a P2P hálózat centralizáltságának foka is. Beszélhetünk **tiszta** én **nem tiszta** P2P hálózatokról egyaránt. A tiszta P2P hálózatok jellemzői:

- A Peer-ek egyenlőként tevékenykednek a kliens és szerver szerepkörök egybeolvasztásával.
- Nincs központi szerver, amely menedzselné a hálózatot.
- Nincs központi router.

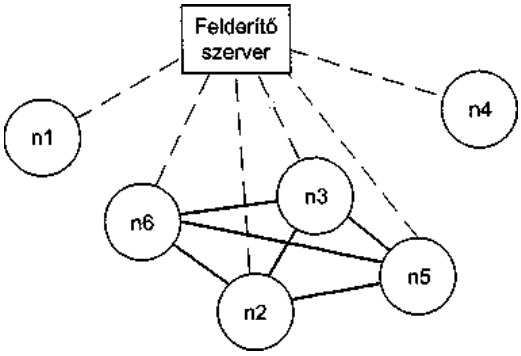
Példa: fájlmegosztás.

Létezik számtalan hibrid P2P rendszer is, amelyek a nem tiszta P2P hálózatokat képviselik. Legfontosabb jellemzőik:

- Létezik egy központi szerver, amely információkat tárol az egyes peer-ekről.
- Az egyes peer-ek felelősek a rendelkezésre álló erőforrások hostolásáért (amennyiben a központi szerver nem rendelkezik vele). A szerver ismeri a peer-ek által megosztásra szánt erőforrásokat. Valamint a peer-ek felelősek azért is, hogy mely erőforrásokat teszik osztottá a többi peer felé.



2. ábra. Decentralizált P2P architektúra



3. ábra. Egy félig centralizált P2P hálózat felépítése

Példák:

- Centralizált P2P hálózat: Napster.
- Decentralizált P2P hálózat: KaZaA
- Strukturált P2P hálózat: CAN
- Nem strukturált P2P hálózat: Gnutella

Indypeer a p2p hálózatokat az alábbi generációkba sorolja:

- **Első generációs hálózatok:** közös jellemzőjük, hogy a kereséshez központi szervert használnak. Ez azt jelenti, hogy a kliensnek csatlakoznia kell egy megadott szerverhez (a szerverek egyikéhez). Az ilyen p2p hálózatok előnye, hogy a szerver egy globális listát tarthat fent egy adott időben rendelkezésre álló fájlokról. Jogi okokból megszűnt hálózatok a Napster és az Audiogalaxy, ma is működő hálózatok a Direct Connect, a Bittorrent (bt) és a Soulseek. (A Napster eredeti formája szűnt meg, ma azonos néven egy fizetős szolgáltatás érhető el.) Bár Indypeer a Bittorrentet a harmadik generációsok közé sorolja, a letöltést vezérlő tracker szerverfunkciója miatt mi az első generációsak között említjük. A Bittorrent fő hátrányai, hogy nem lehet keresni.

- **Második generációs hálózatok (hibrid p2p):** a második generációs hálózatok a legismertebb típusba tartoznak, annak ellenére, hogy a harmadik generációs hálózatok gyorsan fejlődnek. Közös jellemzőjük, hogy a hálózati infrastuktúrát (pl. keresés) a szerverfunkciót is ellátó szupercsomópontok tartják fent. Mivel ezek száma magas, és a hálózat globális, nem lehet jogi vagy technikai eszközökkel ellehetetleníteni őket. Hátrányuk, hogy a keresés nem mindig kimerítő. Ilyen például a régi eDonkey (ed2k) és a Fasttrack.

- **Harmadik generációs hálózatok (tiszta p2p):** működésük teljesen elosztott, minden csomópont egyenrangú, tehát a kliens és szerver szerepek keverednek. A hálózati sávszélesség és a CPU-sebesség általános növekedésével vált lehetővé ilyen hálózatok kialakítása. Ezzel vált lehetővé, hogy a kliensek fokozatosan szerver szerepeket (Pl. a rendelkezésre álló fájlok listájának fenntartása, és terjesztése) is átvegyenek. Ilyen például a Freenet, az Overnet, a WinMX, a Shareaza, és az XOR Kademlia (kadMule). Régi példa a Gnutella, ennek 7 mélységben elárasztó keresési algoritmusa nagy sávszélességet igényel.

3. Fájl megosztás megvalósított Peer-to-Peer rendszerekben

Napjaink legelterjedtebb P2P rendszerei azon rendszerek, amelyek célja alapvetően a fájl megosztás. Több fejlődési irány, megoldás alakult ki a felhasználók közötti dokumentumok megosztására.

3.1 Centralizált, hibrid

Egy hibrid P2P rendszer valójában nem teljesen felel meg a P2P rendszer definíciójának. Bár minden csomópont egyenrangú, a felhasználók dokumentumainak attribútumai (fájlnév, milyen IP-vel rendelkező gépen található, minőség, szerző, cím, stb.) egy központi szerveren vagy szerver csoporton vannak tárolva. Az egyik első P2P rendszer, a Napster is ilyen architektúrájú, az ilyen rendszerek a P2P rendszerek legelső generációjához tartozik.

Ezen architektúra előnyének mondható, hogy a szerver kiszolgáló-képességének határáig gyors, a keresés az összes dokumentumban történik, viszonylag kis hálózati forgalmat igényel. E sok előnyből azt gondolhatnánk, hogy mindenki ezt az architektúrát választja. De egy ilyen, félig centralizált rendszer hátránya, hogy a dokumentumok adatait tároló szerver miatt rendelkezik a centralizált rendszerek minden hátrányával, vagyis az egész rendszer működése megszűnik a szerver hibája, vagy megszűntetése miatt (mint például a Napster esetében, jogi okok miatt).

3.2 Teljesen elosztott, homogén

A P2P rendszerek következő generációja a teljesen elosztott, homogén, „igazi peer-to-peer" rendszer. Leggyakoribb példája a Gnutella hálózat. Az ilyen rendszerben valóban minden résztvevő (csomópont) egyforma: a csomópontok teljesen azonos szerepűek, ugyanazt a feladatot látják el. Egy ilyen rendszerben a dokumentumok adatai nem egy központi szerveren találhatók, hanem ténylegesen egy adott résztvevő tudja csak, hogy milyen attribútumokkal rendelkeznek az általa tárolt dokumentumok.

A keresés tehát megnehezedik: egy adott fájl keresése a hálózatban már nem csupán egy központi szerverhez való adatküldésre korlátozódik. Az egyik csomópont megkap egy kérést (keresést), megvizsgálja, hogy nála megtalálható-e az adott dokumentum, majd ha nem, tovább küldi az általa ismert résztvevőknek, akik szintén ezeket a műveleteket hajtják végre a kérésen.

Az előnyök és hátrányok már láthatók: ebben a rendszerben egy résztvevő hibából, vagy más okokból történő leállása, leállítása nem okoz problémát. Azonban a keresés lassabb lesz, és ami a legfontosabb, nagy hálózati forgalmat generál egy keresés. Ha egy adott kérés bizonyos mélységű továbbküldözgetésével szeretnénk elérni egy adott dokumentum megtalálásának valószínűségét, akkor figyelembe kell vennünk, hogy az általunk használt résztvevő megengedheti-e magának azt a hálózati forgalmat, ami ehhez szükséges.

Sok rendszer született, ami ezt az architektúrát választja, és még több, ami alapvetően nem ezen az architektúrán alapul, de támogatja az ilyen protokollt. A legtipikusabb példák: LimeWire, Shareaza, Morpheus.

3.3 Félig elosztott, hierarchikus

Manapság a gyakorlatban (fájlmegosztásra) használt rendszerek jelentős része hierarchikus szerkezetű. Ilyen rendszereknél bár nem minden csomópont azonos, de azonos viselkedési típusok, szerepek közül választhatnak. Egy nagyobb hálózati forgalmat, vagy nagyobb számítási kapacitást kiszolgálni képes csomópont kiválaszthatja a neki megfelelő szerepet, amivel a többlet erőforrásai kihasználhatóvá válnak, míg a kisebb teljesítményű résztvevők sem kényszerülnek olyan feladatok ellátására, amire nem képesek.

Az ilyen rendszerek jelentős része mindössze kétszintű: a kisebb képességű résztvevők közvetlenül nem vesznek részt a keresésben, hanem dokumentumaik információit odaadják egy nagyobb képességű résztvevőnek, akivel közvetlen kapcsolatban állnak. Egy adott attribútumokkal rendelkező dokumentum iránti kérést szintén ehhez a nagyobb képességű

csomóponthoz intézi. A nagyobb képességű résztvevő (super-node) a többi társához továbbítja a kérést, akik mind információval rendelkeznek a hozzájuk kapcsolódó kisebb képességű résztvevők dokumentumairól is.

Ez az architektúra tehát az előző két típusú rendszerek előnyeit próbálja ötvözni. Talán ennek köszönhető, hogy a legtöbb mai rendszer kétszintű: a talán legtöbbek által ismert Kazaa, és Grokster társa, illetve korábban a Morpheus rendszer is (mindannyian a FastTrack nevű rendszert használják), valamint a szintén közkedvelt WinMX is, de a Gnutella 0.6 vagy 2 protokollt használó számos rendszer (Shareaza, LimeWire) is ebbe a kategóriába tartozik.

3.4 Elosztott indexelés

A legelső, centralizált változat azért volt gyorsabb (és kisebb erőforrás-igényű) a második, decentralizált változatnál, mert nem kellett az összes résztvevőt végigkérdezni, rendelkezik-e az adott dokumentummal, hanem egy központi szerver tudott erre a kérdésre válaszolni. A központi szerver ilyen szerepét azonban rábízhatjuk a többi résztvevőre is. Egy megfelelően szervezett struktúra majdnem hasonlóan gyors lehet, míg hibákra és egyéb tényezőkre érzéketlen marad az elosztott tárolás következményeként.

Az indexek legtöbbször bináris formába alakítás után egy fa, vagy egy gyűrű struktúrában tárolhatók, ahol az index utáni keresés logaritmikus mértékű a résztvevők számához képest, ami hatékonynak mondható. Az ilyen rendszerek azonban többnyire egyszerre csak egy kritérium (például fájlév) indexelését támogatják, ami nem elég flexibilis.

Ezen a problémán például az előző architektúrákkal való kombinálással lehet segíteni (például Gridella). Szintén ezen a problémán próbálnak megoldást javasolni olyan struktúrák, ahol több, hierarchikus felépítésű index-szerkezet létezik (például „Content-based aggregation network”). Szintén jellemző ezen kívül a keresések gyorsítása olyan módon, hogy a P2P struktúra figyelembe veszi az egyes résztvevők földrajzi elhelyezkedését, vagy a hálózati késleltetést (például Grapes).

Ezeknek a rendszereknek tehát rendkívüli előnyük a teljes elosztott működés melletti gyors, kis hálózati forgalmat generáló keresés a dokumentumok között, míg hátrányul sorolhatjuk fel a csökkent mértékű flexibilitáson kívül a meglehetősen bonyolult felépítést is. Egy index fa, vagy gyűrű, esetleg egyéb, elhelyezkedésen alapuló struktúra felépítése és karbantartása legtöbbször nem egyszerű algoritmusokat igényel. Külön probléma lehet – a résztvevők folyamatos változása (kilépés, belépés) miatt – a rendszer nem stabil állapotban lévő teljesítményére vonatkozó állítások megfogalmazása is.

Az ilyen rendszerek sok előnye, ám sok problémája miatt többféle megoldás is született, ami elosztott indexelést alkalmaz. Példaként sorolható fel: Gridella (P-Grid fa), Chord, CAN, Pastry, Tapestry, Freenet, Grapes, stb.

3.5 Homogén Peer-to-Peer architektúra felépítése

Vizsgálataink alanyául az egyik legelterjedtebb kevéssé számításigényes protokollt, a Gnutella-t választottuk. File-sharing protokollról van szó, de ez a gyakorlatban csak kérések és válaszok továbbítását jelenti, a tényleges átvitel az overlay network-ön kívül történik.

3.5.1 A Gnutella protokoll

A Gnutella a kevéssé számításigényes protokollok közé tartozik, a P2P hálózaton belül broadcast-ot valósít meg. Az egyes kérések által generált forgalmat az úgynevezett TTL (time to live, hátralevő élettartam) paraméterrel korlátozza, ami általában 7-ről indul, és minden továbbításkor csökken eggyel (mikor eléri a 0-t, a csomag nem kerül továbbításra). Az egyszer már feldolgozott csomagok újraprocesszálásának (és ami még lényegesebb: újratovábbításának) elkerülése érdekében általános módszer az utoljára látott n csomag azonosítójának megőrzése. A Gnutella ezt az információt a forrás összeköttetéssel együtt tárolja, és a válaszok back-route-olására is felhasználja.

A Gnutella kérések ugyanis nem tartalmaznak IP címeket, csak a válaszokat. Ez egyfajta – bár megkérdőjelezhető hasznosságú – névtelenséget biztosít a résztvevők számára.

A mai Internet közönsége érthető okokból bizalmatlan. Ennek az egyik leggyakoribb megnyilvánulása a tűzfalak használata, amelyek igen erőteljesen elhatárolják a kiszolgálókat és az ügyfeleket. Ez egy olyan alapvető probléma, amellyel valamilyen módon minden P2P rendszernek számolnia kell: kapcsolatlétrehozás szempontjából a tűzfalak egyirányúak. Az overlay network szempontjából tehát szükséges olyan node-ok jelenléte, amelyek képesek kapcsolatokat fogadni, nincsenek tűzfal mögött. A nagyobb problémát a fájlátvitel támogatása jelenti; ezt a Gnutella akkor bírja megoldani, ha a két node közül legfeljebb az egyiket védi firewall. Ugyanis – ha a szerver szerepet betöltő csomópont van tűzfal mögött – lehetőség van az overlay network-ön belül egy úgynevezett Push Request továbbítására, melynek hatására a fájlátvitelt a szerver fogja megindítani, így a tűzfal – rendszeradminisztrátorok által tipikusan nem támogatott – átkonfigurálására sincs szükség, és mégis létrejöhet az áttöltés.

Topológiák

A Gnutella specifikáció a legteljesebb mértékben nyitva hagyja a hálózatépítés és egyáltalán, a logikai hálózatban megvalósítandó célszerű topológia kérdését, vizsgálatainkat elsősorban ebben a témakörben végeztük.

Négy alap topológiát hasonlítunk össze az általunk definiált kritériumok alapján.

- Random Mesh: teljesen véletlenszerű gráf, tehát megengedjük izolált node-ok jelenlétét is
- Connected Mesh: található a gráfban olyan részfa, amely minden csomópontot tartalmaz
- Semi-Random Mesh: véletlenszerű gráf, de minden csomópont rendelkezik legalább egy kapcsolattal
- Connected Stars: szoftver szempontból továbbra is él a homogenitás, de az erőteljesebb hardverrel rendelkező résztvevők felismerése segítségével létrehozható egy alhálózat, amelyen a többi csomópont csupán 1-1 kapcsolaton át, levélként függ

4. További alkalmazási területei

A peer-to-peer hálózatok másik jelentős alkalmazási területe a telefonprogramok, például a **Skype**. Ezek a peer-to-peer hálózatok által olcsón biztosított infrastruktúrát használják ki, hogy olcsón szolgáltassanak telefonos kapcsolatot. A peer-to-peer hálózat miatt jelentősebb

befektett tőke nélkül tudnak olyan szolgáltatást nyújtani, amelyek a korábban óriási pénzekből létrehozott infrastruktúra megtérülése miatt maradtak drágák.

A **Skype** készítői a telefonprogram fejlesztése során felismerték a peer-to-peer hálózatokban rejlő lehetőségeket, és 2006 decemberében kísérleti jelleggel beindították **Joost** néven az egyik internetes televíziós szolgáltatást.

P2P TV

A P2P TV elnevezés a peer-to-peer szoftver alkalmazásra utal, amely lehetővé teszi a videós tartalmak újraelosztását valós időben a P2P hálózaton. Ezek a videós tartalmak legtöbbször TV csatornák a világ minden tájáról, de egyéb forrásokból is eredhetnek. Különösen annak köszöneti népszerűségét, hogy a P2P TV modell a segítségével további költségmegtakarításokat lehet véghezvinni a TV szolgáltató részéről. Ez lehetővé teszi, hogy a világ bármely csatornája globális szinten elérhető legyen.

A P2P TV alapelvét a broadcatching technológia határozza meg. Ennek a technológiának a segítségével a digitális tartalmak letöltése az interneten keresztül az RSS-en keresztül véghezvihető. Az általános alapelv, a technikai megoldás mögött egy automatizált mechanizmus, ahogy a különböző web feed-ek összegyűjtésre kerülnek, majd letölthető tartalommá válnak a felhasználó részére.

Minden felhasználó, amikor letölt egy videót, egyidejűleg fel is tölti azt más felhasználók számára, így hozzájárulván a teljes sávszélességhez. A videó letöltésének állapota általában mindig néhány perccel késleltetett a forrásvideóhoz képest. A csatornák videóminősége attól függ, hogy hány ember nézi éppen. Minél több felhasználó nézi egyidejűleg, annál jobb minőségben lehet a videót lejátszani.

Forrásmunkák:

1. P2P: **https://nws.niif.hu/ncd2003/docs/ehu/EHU-73.htm, 2008.**
2. Andy Oram (edited by): Peer-to-Peer – Harnessing the Power of Disruptive Technologies. O'Reilly, 2001.
3. P2PTV: http://www.e-businessportal.hu/e-business/hirek/p2p-tv, 2008.