P2P tehnoloogia ülevaade

Meelis Roos
mroos@ut.ee

08.09.2009

Peer-to-peer süsteemid

- Liigitus
- Ründed
- Anonüümsus
- Puuräsi
- DHT
- Näide: Kazaa
- Näide: Kademlia
- Näide: Gnutella
- Näide: BitTorrent

Peer-to-peer süsteemide liigitus

- Eelajalugu: tsentraalsed süsteemid
- "Puhas" P2P
  - Klient on ise server
  - Pole keskset haldurit
  - Pole keskset ruuterit
  - Pole *single point of failure*'t
- Hübriid-P2P
  - Keskserver haldab klientide infot ja teeb otsinguid
  - Andmevahtus toimub reeglina otse klientide vahel

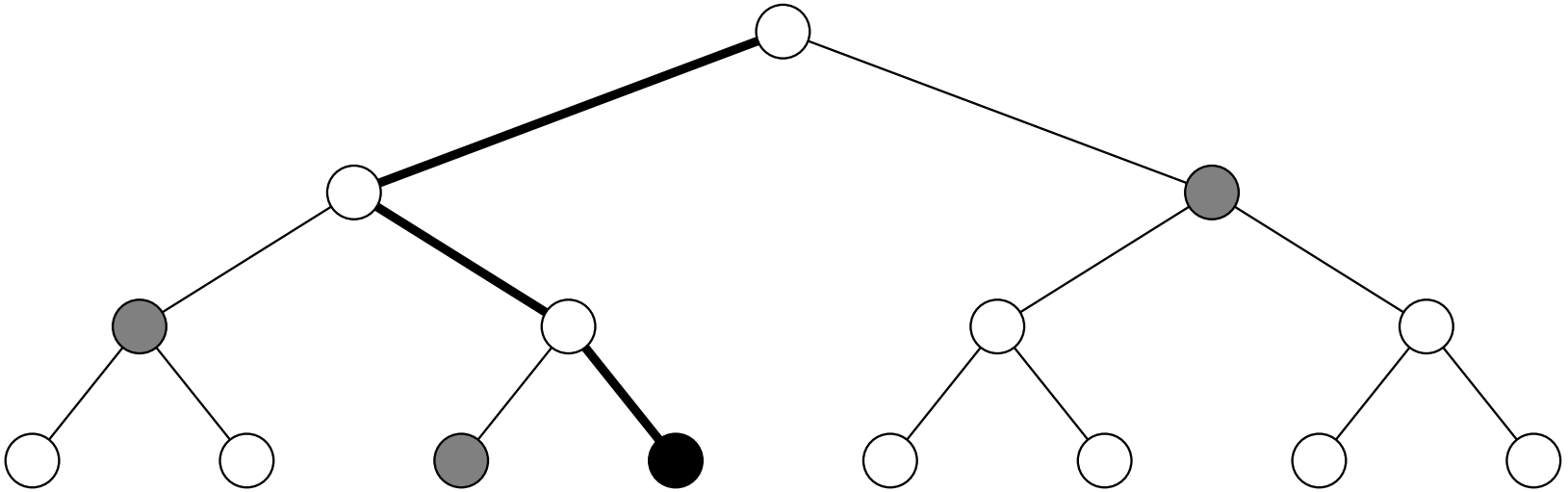
Ründed P2P vastu

- Mürgitamine otsingutulemuste tasemel
- Mürgitamine vigaste failijuppide tasemel
- Võrguressursside tarbimine ilma ise vastu andmata
- Kurivara (viiruste, Trooja hobuste jms) levitamine võrgus
- Kurivara (viiruste, Trooja hobuste jms) levitamine P2P tarkvaraga
- Teenusetõkestusründed
- Filtreerimine
- Identiteediründed
- Spämm võrgus

Anonümiseerivad P2P lahendused

- Kasutaja identiteet peidetakse teiste eest nii hästi kui saab
- Tegelikult kasutatakse reeglina mingeid pseudonüüme (kasvõi ajutisi)
- Kui ainult endale tõmbaks, saaks tõmbaja ja sisu kergesti kokku viia
- Seega edastame ka teistele
- Lisasaatmised liikluse analüüsi vältimiseks
- Efektiivsuse oluline langus
- Näiteks Freenet sisu levitamiseks, Tor pseudoanonüümseks ruutinguks

Puuräsi



- Failifragmentide autentsuse kontrolliks
- BT2, Gnutella, Open Content Network
- Tiger Tree hash, THEX
  - Sisuline lisavõimalus — tagavara-tracker

urn:tree:tiger:LWPNACQDBZRYXW3VHJVCJ64QBZNGHOHHHZWCLNQ

DHT

- *Distributed Hash Tables* — hajusad räsitabelid otsingu jaoks ilma floodimata
- Igal sõlmel on oma võti
- Igal infotükil on oma võti, otsing toimub võtmete järgi sarnase võtmega sõlmesid pidi
- Lihtne versioon: kõik sõlmed teavad kõiki sõlmi ning leiavad infotüki võtme järgi kohe õige sõlme, kelle tabelis kirje on
- Sõlmede lisandumine ja lahkumine
- Praktiline versioon: võtme otsing toimub samm-sammult ID-le lähenedes (logaritmiline sammude arv)
- Faili räsile lähedastesse sõlmedesse lisatakse viit publitseerivale sõlmele

Kazaa

- Fasttrack võrk
- Hübriidvõrk supernodedega, supernoded suhtlevad omavahel
- Kergesti haavatav räsi
- Käigu pealt kokku lepitav krüpto
- Andmevahetus üle HTTP
- Sisseehitatud supernodede nimekiri

eDonkey

- Hübriidvõrk ca miljoni kliendi serveritega otsingute jaoks
- Lokaalne ja globaalne otsing
- MD4 räsi faili identifikaatoriks (nimest sõltumatu)
- High ID/low ID
- Libaserverid filtreeringu ja kasutajate jälgimisega

Overnet

- eDonkey järeltulija
- Detsentraliseeritud
- Kasutab Kademlia algoritmi
- Näiteks Storm botnet kasutab oma suhtluseks modifitseeritud eDonkey/Overnet protokolli (oma krüpto, osade kaupa rentimine)

Kademlia

- Nn. ülekattevõrk TCP/IP peal (igal sõlmel on oma ID)
- Kaugus võrgus on ID-de Hammingi kaugus
- Suhtluseks UDP
- Otsimiseks *distributed hash table* (DHT)

eMule

- Töötas esialglselt eDonkey võrgu baasil
- + Krediidisüsteem
- + Pakkimine
- Hiljem oma Kad võrgus ka
- Kad on Kademlia protokolliga võrk, mille enamus servereid on ka "päris" Kademlia võrgus

Gnutella protokoll

- Avatud protokoll + mitmeid erinevaid kliente
- Protokoll on aja jooksul oluliselt arenenud
- Failide transport otse klientide vahel HTTP-ga
- NAT taguste klientide käest failide saamiseks *push* päringud
- Otsingute jaoks oma protokoll, TCP ja uuemal ajal ka UDP
- Ootejärjekorrad failide tõmbamisel
- Upload-download suht lõdvalt seotud
- Gnutella2 on omaette ja sõltumatu protokoll Shareaza tegijatelt

Failide tõmbamise paralleliseerimine

- Otsingu tulemuseks oli esialgu masin ja failinimi koos metainfoga, sealt tõmmati
- Sama nime ja pikkusega faili fragmente võis mitmest kohast korraga tõmmata (*swarming*)
- Töökindlama failide identifitseerimise jaoks toodi sisse SHA-1 räsi faili identifikaatorina
  - urn:sha1:DH7EPRIOKFFXCZJGDA7TEBAR7SYYWQO3
- Ülekatmine erinevatest allikatest tõmbamisel

Gnutella otsing

- Algselt *flooding* — päringutega üle ujutamine
- Igal kliendil on ühendused mingi hulga teiste klientidega, päring saadetakse kõigile naabritele
- Iga saadud päringu kohta tehakse kohalik otsing ning lisaks saadetakse päring edasi naabritele, vähendades TTL
- Hiljutiste nähtud päringute cache duplikaatide vältimiseks
- Töötab väikese võrguga, suure võrgu jaoks ei skaleeru
- Aeglase (modemi)ühendusega kliendid aeglusatavad kogu võrgu tööd

Gnutella ultranoded

- Tänapäeval töötav lahendus skaleerimismurede vastu
- Kiiremad sõlmed valitakse teiste päringuid vahendama
- Kui enne oli klient ühenduses 10-20 naabriga, siis nüüd 3-4 ultranodega
- Ultranoded vahendavad otsinguid omavahel, aga mitte lehtede kaudu
- Ultranodeks valitakse avastatud omaduste kaudu
- Skaleerub üsna mõistlikult, aga garantiisid pole ja protokoll on *ad-hoc*

Gnutella — muud

- Iga klient peab meeles nimekirja võrgus viimati nähtud sõlmede aadressidest, et järgmisel käivitumisel kuhugi ühenduda oleks
- Samuti ultranodede nimekiri
- Konkreetse faili pakkujate otsimine sha1 ja DHT abil
- Gwebcache - mitmed veebiserverid sõlmede nimekirja pidamiseks ja uutele ning kaua eemal olnud klientidele levitamiseks (ajalooline)
- Teiste nodede leidmine DHT abil (gwebcache asemel)
- IPv6 tugi
- Hosti brausimise päringud

BitTorrent

- Failide efektiivseks levitamiseks paljude klientide vahel
  - Üleslaadimine sisuliselt kohustuslik
- Otsing puudub (leidub veebiteenuseid)
- Tracker — teab, kellel missugused tükid on, vahendab infot selle kohta
- Kliendid tõmbavad tükke üksteiselt ja teatavad olemasolevtest tükkidest trackerile
- Uuemal ajal DHT trackeri puudumisel töötamiseks, *ad-hoc* krüpto