Asiakas/palvelin ja vertaisverkko tiedostonjakojärjestelminä

Sami Hirvikallio

2.11.2006

Joensuun yliopisto
Tietojenkäsittelytiede
Pro gradu -tutkielma

Tiivistelmä

Tämä Pro gradu–tutkielma käsittelee asiakas/palvelin järjestelmiä ja vertaisverkkoja. Tutkielmassa pyritään tuomaan esiin järjestelmien eroja ja yhtäläisyyksiä. Tutkielma keskittyy tarkastelemaan järjestelmiä tiedostojen jakamisen näkökulmasta. Käsiteltäviä aiheita ovat toteutettavuus, verkonkuormitus ja turvallisuus. Tutkimusmenetelmänä on koota eri lähteistä vertailukelpoista tietoa ja verrata sitä näiden arkkitehtuurien kesken. Tutkielman tuloksena oli, että arkkitehtuurit ovat osittain samanlaisia, mutta eroavat kuitenkin merkittävästi toisistaan. Asiakas/palvelin järjestelmä on turvallisempi kuin vertaisverkko. Asiakas/palvelin järjestelmä on helpompi toteuttaa kuin vertaisverkko. Vertaisverkko on tiedostojen ja resurssien kannalta katsottuna ylivertainen verrattuna asiakas/palvelin järjestelmään.

ACM-luokat: (ACM Computing Classification System, 1998 version): C.0, C.2.1, C.2.4, C.2.m, C.4

Avainsanat: Asiakas/palvelin, hajautettu järjestelmä, toteutettavuus, turvallisuus, verkonkuormitus, vertaisverkko.

Sisällysluettelo

1 Johdanto

Tiedostonjakojärjestelmät ovat suunniteltu jakamaan tiedostoja. Ne ovat yleensä hajautettuja järjestelmiä. Hajautettu järjestelmä voidaan määritellä siten, että se on yhtenäiseltä vaikuttava järjestelmä, joka koostuu komponenteista, jotka ovat verkotetuissa tietokoneissa. Nämä tietokoneet kommunikoivat ja koordinoivat toimintaansa lähettämällä viestejä keskenään [6]. Tämä määritelmä johtaa hajautettujen järjestelmien tunnusmerkkeihin; komponenttien yhdenaikaisuus, yleismaailmallisen kellon puuttuminen ja komponenttien itsenäiset virheet. Tiedostojen ja resurssien jakaminen ovat hajautettujen järjestelmien pääasiallisia tarkoituksia. Asiakas/palvelin ja vertaisverkko ovat hajautettuja järjestelmiä. Tämän tutkielman tarkoitus on selvittää, miten asiakas/palvelin arkkitehtuuri ja vertaisverkko arkkitehtuuri eroavat toisistaan arkkitehtuurin, toteutettavuuden, verkonkuormituksen ja turvallisuuden osalta. Tutkielma keskittyy tarkastelemaan asiakas/palvelin ja vertaisverkko arkkitehtuureita tiedostojen jakamisen näkökulmasta. Tutkimusmenetelmänä on koota eri lähteistä vertailukelpoista tietoa ja verrata sitä näiden arkkitehtuurien kesken. Tutkielmassa käsitellään ensimmäiseksi luvussa 2 vertailtavat arkkitehtuurit, mm. historia ja erilaiset arkkitehtuurimallit. Seuraavaksi luvussa 3 käsitellään arkkitehtuurien toteutettavuutta, siinä esitellään hajautettujen järjestelmien toteutettavuuden haasteet ja verrataan niitä asiakas/palvelin ja vertaisverkko arkkitehtuureihin. Luvussa 4 käsitellään arkkitehtuurien verkonkuormitusta. Tämän jälkeen, luvussa 5 perehdytään arkkitehtuurien turvallisuuteen. Lopuksi luvussa 6 tehdään yhteenveto edellisistä luvuista.

2 Vertailtavat arkkitehtuurit

Tässä luvussa esitellään vertailtavat arkkitehtuurit. Ensin esitellään asiakas/palvelin arkkitehtuuri (client/server) ja tämän jälkeen esitellään vertaisverkko (peer-to-peer).

2.1 Asiakas/palvelin arkkitehtuuri

1990-luvulla asiakas/palvelin teknologia alkoi tulla tunnetuksi tietokoneiden käyttäjien keskuudessa. Tämä johtui laitteistotehojen ja ohjelmistojen kehityksestä. Aikanaan tällä teknologialla pyrittiin vähentämään kustannuksia, lisäämään hallittavuutta ja palveluita asiakkaille. Asiakas/palvelin systeemeissä on kolme komponenttia: asiakas (client), palvelin (server) ja verkko (network). Jokainen näistä komponenteista sisältää useita laitteisto- ja ohjelmistokomponentteja. Asiakas/palvelin systeemit rajoittuvat harvoin omaan verkkoonsa. Lisäksi saman verkon eri asiakkaat saattavat käyttää erilaisia ohjelmistoja, joilla ne käyttävät palvelimen palveluita. Palvelimet asiakas/palvelin systeemeissä saattavat myös käyttää eri käyttöjärjestelmiä ja tietokantasovelluksia. Asiakas/palvelin systeemin tulee olla kykenevä kommunikoimaan myös toisten asiakas/palvelin systeemien kanssa. On olemassa monen tyyppisiä asiakas/palvelin sovelluksia, vaihdellen yksinkertaisista monimutkaisiin ja jokaisella näillä on erilaiset vaatimukset jokaisen asiakas/palvelin systeemin komponentin suhteen. Asiakas/palvelin systeemien päätarkoitus on sallia jokaisen verkonsolmun olla tavoitettavissa ja sallia myös ohjelmistokomponenttien toimimisen yhdessä. Kun nämä ehdot täyttyvät, asiakas/palvelin ympäristö on hyvä ja näin se voi saavuttaa etuja, jotka voivat olla kustannuksien väheneminen, työn tehostuminen, joustavuus ja resurssien parempi käyttö [7].

Kuten aikaisemmin mainittiin, asiakas/palvelin systeemi koostuu kolmesta komponentista, jotka ovat asiakas, palvelin ja verkko. Jokainen komponentti koostuu useista laitteisto- ja ohjelmistokomponentista. Näiden osien välistä toimintaa on selvitetty kuvassa 2.1 [7].

Kuva 2.1: Asiakas/palvelin systeemin komponentit [7].

Asiakas: Asiakkaan laitteisto on yleensä pöytätietokone, jossa asiakasohjelmisto toimii. Asiakasohjelma muodostaa pyyntöjä ja lähettää ne verkko-ohjelmistolle. Tämä ohjelmisto lähettää pyynnöt palvelimelle, hyväksyy palvelimen vastaukset ja lähettää ne takaisin asiakkaalle. Asiakasohjelma voi tehdä jotain saadulle tiedolle, mutta pääasiassa asiakasohjelma siirtää tiedon vain asiakasohjelman tiedonesityskomponentille. Tiedonesityskomponentti muodostaa käyttöliittymän, jonka kanssa käyttäjä kommunikoi ja näkee saadut tulokset. Tieto esitetään yleensä graafisen käyttöliittymän (GUI) avulla. Asiakaslaitteistolla on myös käyttöjärjestelmä, jonka ei tarvitse välttämättä olla sama kuin palvelimella. Joissakin tapauksissa asiakas on itse asiassa palvelin, joka toimii kuin asiakas pyytäessään dataa toiselta palvelimelta. Tällöin palvelinta kutsutaan agentiksi (agent) [7].

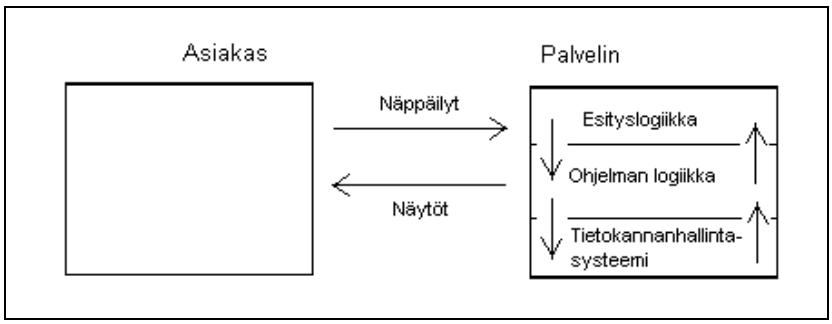
Palvelin: Palvelin on tietokone, joka suorittaa tiedonhallintaohjelmaa, joka on suunniteltu palvelintoimintoja varten. Verrattuna asiakaslaitteistoon, palvelimessa on yleensä enemmän muistia, kovalevytilaa ja suoritintehoa. Palvelimella on käyttöjärjestelmä, tiedonhallintajärjestelmä ja verkko-ohjelmisto. Käyttöjärjestelmän tulee kommunikoida luotettavasti verkko-ohjelmiston kanssa, sekä sen tulee olla myös tarpeeksi tehokas palvelin teknologialle. Hyvän palvelimen ominaisuuksia arvioidessa tulee huomioida palvelimen luotettavuus, eli miten usein tapahtuu virheitä. Myös palvelimen saatavissa olo on tärkeä ominaisuus, eli kuinka nopeasti palvelin on taas pystyssä kaatumisen jälkeen. Joustavuus ja skaalattavuus ovat myös tärkeää, eli miten helposti palvelin on päivitettävissä [7].

Verkko: Verkkolaitteisto koostuu kaapeleista, verkkokorteista ja muista laitteista, jotka yhdistävät palvelimen ja asiakkaat. Yhteyksien täytyy sallia palvelinten pääsyn toisille palvelimille ja tietenkin asiakkaiden pääsyn kaikkialle verkkoon. Verkko-ohjelmisto hoitaa

kommunikaation ja tietovuon (data flow) verkossa. Verkkokäyttöjärjestelmä hoitaa palvelimen syöteprosessit (input/output process).

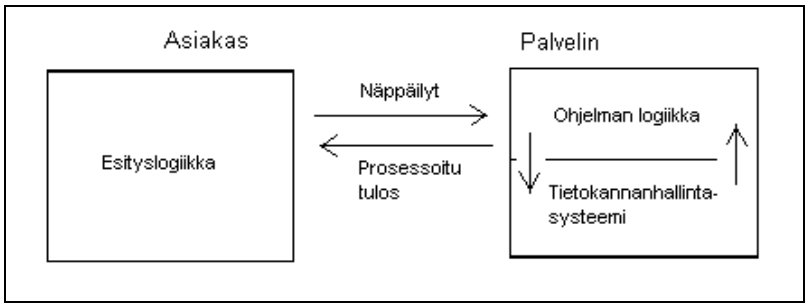
Asiakas/palvelin ohjelmistot voidaan kategorisoida kolmeen eri luokkaan sen perusteella, missä suurin osa tiedon prosessoinnista tapahtuu. Jokainen luokka vaatii erilaiset laitteisto- ja ohjelmistovalmiudet asiakkaalta, palvelimelta ja verkolta [7].

Palvelinperusteiset ohjelmat (host-based): Ovat perusmuoto asiakas/palvelin ohjelmistoista. Asiakaskoneella on vain esitystaso (presentation layer) ja kaikki muu ohjelmaan liittyvä hoidetaan palvelimella. Verrattuna muihin malleihin palvelinperusteinen ohjelma tarvitsee näin ollen kaikkein vähiten toimintoja asiakkaan koneella. Tätä mallia voi verrata tyhmiin päätteisiin (non-intelligent terminal) eli niin sanottuihin suppeisiin asiakkaisiin (thin client). Kuvassa 2.2 on selvitetty palvelinperustaisen ohjelman toiminta [7].



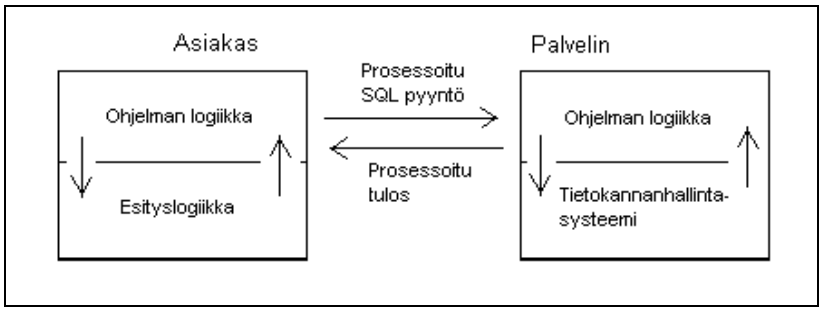
Kuva 2.2: Palvelinperusteinen prosessointi [7].

Asiakasperusteiset ohjelmat (client-based): Tässä mallissa kaikki ohjelman logiikka on asiakkaan vastuulla, pois lukien tiedon laillisuus rutiinit, jotka on koodattu palvelimen tiedonhallintasysteemiin (DBMS). Tällaiset ohjelmat on suunniteltu hyödyntämään uusinta teknologiaa ja arkkitehtuureja, sen sijaan, että ne simuloisivat vain palvelin/terminaali vuorovaikutuksia, kuten palvelinperusteiset ohjelmat ks. kuva 2.3. Tällöin puhutaan paksusta asiakkaasta (thick client) [7].



Kuva 2.3: Asiakasperusteinen prosessointi [7].

Yhteistoiminnalliset ohjelmat (cooperative): Tämä asiakas/palvelin ohjelmistojen kolmas luokka käyttää täysin yhteistoiminnallista lähestymistapaa. Todellisessa järjestelmässä kaikki komponentit ovat yhdenvertaisia ja voivat pyytää tai tarjota palveluita toisilleen. Tiedonprosessointi tapahtuu siellä, missä vapaita resursseja on tarjolla. Systeemi voi toimia asiakkaana toisille palvelimille ja palvelimena toisille asiakkaille. Tämä malli lähestyy vertaisverkkomallia. Tällöin puhutaan rikkaasta asiakkaasta (rich client). Yhteistoiminnallisen ohjelman toimintaperiaate on selvitetty kuvassa 2.4 [7].



Kuva 2.4: Yhteistoiminnallinen prosessointi [7].

Asiakas/palvelin systeemit ovat luonteeltaan modulaarisia. Ne sisältävät monia moduuleita tai komponentteja, jotka toimivat yhdessä. Eräs huomattava asia asiakas/palvelin arkkitehtuureissa on se, että niissä voidaan sekoittaa erilaisia asiakasohjelmia erilaisiin palvelinohjelmistoihin. Modulaarisuus on tärkeä näkökulma asiakas/palvelin arkkitehtuurissa, koska se kannustaa ohjelmistokehittäjiä kehittämään ohjelmansa siten, että ne toimivat myös muiden kehittämissä asiakas/palvelin ohjelmistoissa. Kuitenkin, tällä hetkellä yhteensopivuus on rajoittunutta, mutta ohjelmistokehittäjät ovat lisänneet kiinnostustaan tehdä ohjelmistaan yhteensopivia toisten ohjelmistojen kanssa. Tähän on päädytty, koska on ohjelmistokehittäjien edunmukaista, että heidän ohjelmansa toimivat

mahdollisimman monen muun asiakas/palvelin ohjelmiston kanssa. Asiakas/palvelin systeemeiden modulaarisuus tarjoaa näin ollen valinnanvaraa ohjelmistojen suhteen ihmisille, jotka käyttävät informaatiosysteemeitä [4].

Bochenski [4] määrittelee kymmenen kohdan lista, jolla voidaan määritellä ominaisuudet, jotka kuuluvat asiakas/palvelin systeemiin. Viisi ensimmäistä ovat perustavanlaatuisia ominaisuuksia ja loput viisi ovat vapaaehtoisia, mutta silti suotavia ominaisuuksia.

1. Asiakas/palvelin arkkitehtuuri sisältää asiakasprosessin ja palvelinprosessin, jotka voidaan erottaa toisistaan, mutta kuitenkin toimivat yhdessä
2. Asiakas ja palvelin voivat toimia samalla tai erillisillä tietokoneilla
3. Asiakasympäristö tai palvelinympäristö voidaan päivittää ilman, että tarvitsee päivittää toista ympäristöä
4. Palvelin pystyy palvelemaan useampaa asiakasta yhtä aikaa. Joissakin asiakas/palvelin systeemeissä asiakkaat voivat olla yhteydessä useampaan palvelimeen
5. Asiakas/palvelin systeemi sisältää jonkinlaisen verkottumismahdollisuuden
6. Merkittävin osa (joissakin tapauksissa kaikki) ohjelman logiikasta sijaitsee asiakkaan ohjelmassa
7. Asiakas aloittaa yleensä tapahtuman, ei palvelin
8. Käyttäjäystävällinen graafinen käyttöliittymä sijaitsee yleensä asiakkaanpuolella
9. Rakenteellisen hakukielen (SQL) mahdollisuus on ominaisuus, joka on lähes jokaisessa asiakas/palvelin systeemissä
10. Tietokantapalvelimen tulee tarjota tiedon turvaamiseen ja turvallisuuteen liittyvät palvelut

2.2 Vertaisverkko

Internet, joka kehitettiin 1960-luvun lopulla, pohjautui vertaisverkkoon. Alkuperäisen ARPANET:n tarkoitus oli jakaa laskentatehoa ympäri Yhdysvaltoja. Haasteena oli integroida erilaisia olemassa olevia ja tulevia verkkoja sellaiseen arkkitehtuuriin, joka sallisi jokaisen palvelimen olla yhdenvertainen. Ensimmäiset palvelimet ARPANET:ssä – UCLA, SRI, UCSB ja Utahin yliopisto, olivat itsenäisiä laskentakeskuksia, joilla oli samanarvoinen status. ARPANET yhdisti nämä, ei suhteessa isäntä/orja (master/slave), eikä suhteessa asiakas/palvelin (client/server) vaan yhdenvertaisiksi laskentayksiköiksi eli ensimmäiseksi vertaisverkoksi [17].

Aikainen Internet oli myös paljon avoimempi ja vapaampi kuin nykyinen Internet. Palomuureja ei tunnettu ennen 1980-lukua. Yleensäkin, mitkä tahansa kaksi tietokonetta Internetissä pystyivät lähettämään paketteja toisilleen. Internet oli tutkijoiden käyttämä apuväline. Tutkijoiden ei tarvinnut huolehtia turvallisuudesta, koska olivat tekemisissä vain toistensa kanssa. Protokollat ja systeemit olivat epämääräisiä ja tarpeeksi erikoistuneita ja näin ollen tietomurrot olivat harvinaisia ja harmittomia [17].

Internetin aikaisemmat jymysovellukset (killer application) olivat FTP ja Telnet, jotka olivat asiakas/palvelin sovelluksia. Telnet sovellus kirjautui palvelimelle sisään, jonka jälkeen voitiin käyttää palvelimella olevia ohjelmia. FTP sovellus lähetti ja vastaanotti tiedostoja tiedostopalvelimelta. Mutta, koska sovellukset olivat asiakas/palvelin sovelluksia, käyttötapaukset olivat symmetrisiä. Jokainen Internetin isäntä (host) pystyi ottamaan FTP tai Telnet yhteyden toiseen isäntään. Tästä seurasi se, että Internet mahdollisti mutkikkaampien systeemien kehittämisen, kuten Usenet ja DNS (domain name system), jotka käyttivät vertaisverkkopohjaisia kommunikaatiomenetelmiä. Nykyään Internet on yhä rajoitetumpi asiakas/palvelin pohjaisille ohjelmille, mutta vertaisverkkopohjaiset ohjelmat ovat jälleen yleistymässä. Uskotaan, että Internet on jälleen muuttumassa lähemmäksi sen alkuperäistä mallia [17].

Vertaisverkon edut verrattuna asiakas/palvelin arkkitehtuuriin ovat vähentynyt riippuvuus palvelimesta ja palvelimen tehtävien siirtäminen vertaisverkon jäsenille. Jotkut vertaisverkkomallit eivät sisällä ollenkaan palvelinta. Vertaisverkon käyttäjät voivat luoda

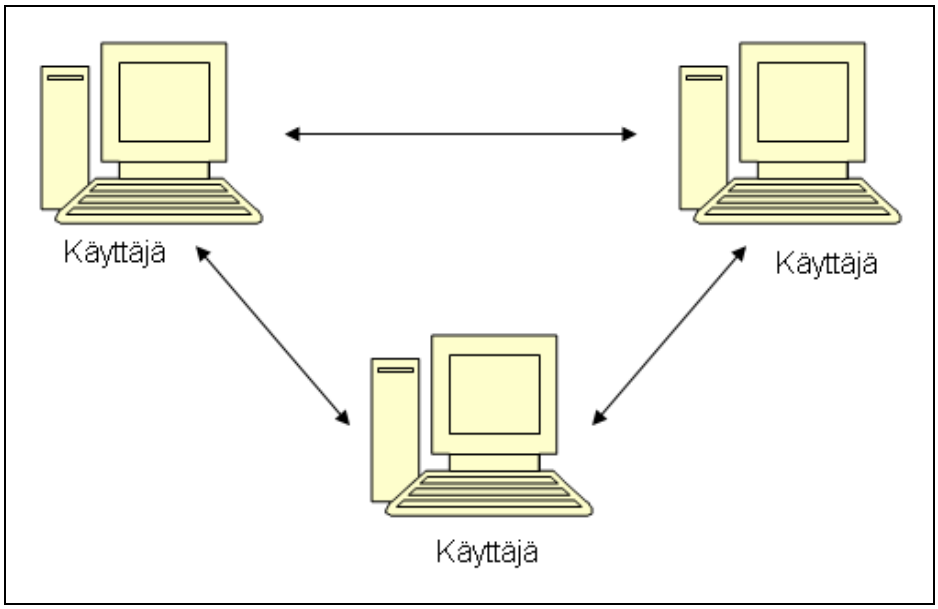
suoraan yhteyksiä toisiin käyttäjiin, ilman palvelimen apua. Vertaisverkkomalleissa käyttäjillä on enemmän hallinnan mahdollisuuksia kuin tyypillisissä asiakas/palvelin malleissa, joissa joudutaan noudattamaan tavanomaisia sääntöjä. Kuten asiakas/palvelin malleissa, vertaisverkoissa palvelin ei ole virheiden aiheuttajana. Vertaisverkkomalleissa, joissa on palvelin, palvelimen vastuut ja tehtävät on minimoitu [9].

Vertaisverkkomallit voidaan jakaa seuraavasti:

- Puhdas vertaisverkko
- Vertaisverkko, jossa on yksinkertainen havaintopalvelin
- Vertaisverkko, jossa on havainto- ja hakupalvelin
- Vertaisverkko, jossa on havainto-, haku- ja sisältöpalvelin

Puhdas vertaisverkkomalli (pure peer-to-peer model) luottaa täysin asiakaskoneisiin (clients) eli se ei käytä lainkaan palvelimia (server). Kaikki kommunikaatio tapahtuu vertaisverkon jäsenten kesken ja näin ollen puhtaan vertaisverkon toimintaperiaate hajottaa perinteisen asiakas/palvelinperustaisen kommunikaatiomenetelmän. Tämä malli tarjoaa käyttäjien asiakas/palvelin funktioiden vaihdon. Kun käyttäjä pyytää tietoa, se toimii kuin asiakas ja milloin käyttäjä tarjoaa tietoa se olettaa olevansa palvelin [9].

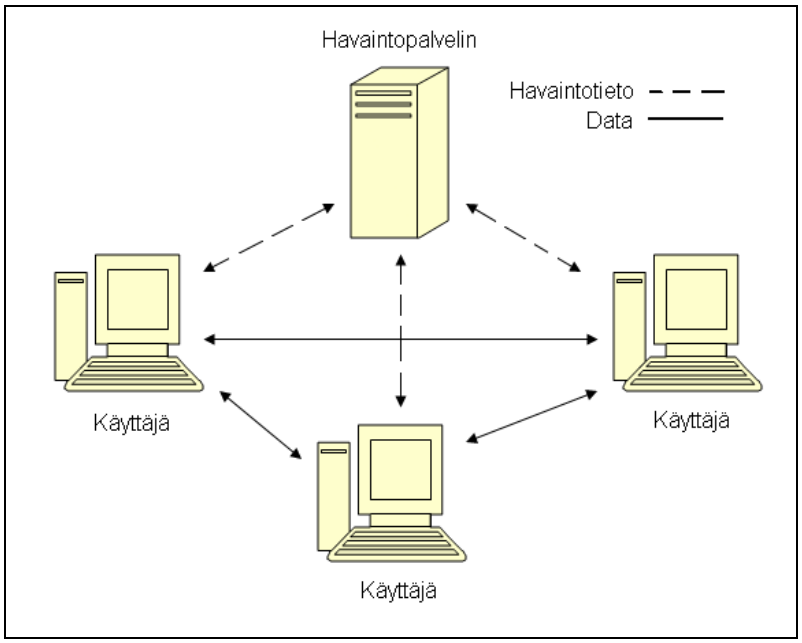
Ainut puhtaiden vertaisverkkomallien ongelma on muiden käyttäjien löytäminen vertaisverkosta, koska siinä ei ole minkäänlaista mekanismia, joka pitäisi yllä tietoa vertaisverkkoa käyttävistä käyttäjistä. Tästä seuraa se ongelma, että aina ei välttämättä tunneta tarpeeksi suurta vertaiskäyttäjä määrää, joka riittäisi kattamaan lähetetyn pyynnön. Näin ollen puhtaassa vertaisverkossa käyttäjien tulee löytää toisensa lähes manuaalisesti, mutta yleensä sovellukset, jotka perustuvat puhtaaseen vertaisverkkoon sisältävät jonkinlaisen verkottumisalgoritmin, joka huolehtii käyttäjien riittävästä löytymisestä. Verkottuminen tapahtuu dynaamisesti ja kiinteitä yhteyksiä luodaan vain jos siirretään dataa. Kuvassa 2.5 on hahmotelma puhtaasta vertaisverkkomallista [9].



Kuva 2.5: Puhdas vertaisverkko.

Vertaisverkko yksinkertaisella havaintopalvelimella (simple discovery server), tämän mallin nimi itsessään kuvaa hieman jo sen rakennetta. Tällaiset vertaisverkkomallit eivät oikeastaan sisällä palvelinta sen varsinaisessa merkityksessä. Saavuttaakseen hieman hallittavuutta, tähän malliin on lisätty palvelin ja hieman sen ominaisuuksia. Palvelimen rooli tässä mallissa on rajoitettu antamaan vain jo kytkeytyneiden jäsenten osoitteet, sellaisille jäsenille, jotka ovat juuri kytkeytymässä vertaisverkkoon. Tämä ominaisuus lisää mahdollisuuksia löytää suuremman määrän käyttäjiä vertaisverkosta. Tuleva jäsen huomioi palvelimen vain kirjautumalla palvelimelle sisään. Yhteyksien järjestäminen ja kommunikaatio jää vielä vertaisverkon jäsenten huoleksi. Ladatakseen vertaisverkon käyttäjien tarjoamaa dataa, käyttäjän täytyy lähestyä vertaisverkkoon kytkeytyneitä jäseniä, jokaista erikseen kunnes haluttu data löytyy ja tämä taas tekee prosessin aikaa vieväksi [9].

Tärkein saavutettu etu tässä mallissa on suuri määrä tavoitettavissa olevia käyttäjiä. Silti, jos palvelin kaatuu tai sen toiminta hidastuu, muiden käyttäjien jäljittäminen tulee hankalaksi ja näin ollen myös muut käyttäjät kärsivät tästä, mutta kuitenkin lähinnä vain uudet jäsenet. Kuvassa 2.6 on hahmotelma vertaisverkosta yksinkertaisella havaintopalvelimella [9].



Kuva 2.6: Vertaisverkko yksinkertaisella havaintopalvelimella.

Vertaisverkko havainto- ja hakupalvelimella (discovery and lookup server), Tässä mallissa palvelinta käytetään tarjoamaan listaa jo kytkeytyneistä käyttäjistä ja jokaisen käyttäjän tarjoamista resursseista. Tämä malli vähentää käyttäjien työmäärää, koska ei ole enää tarvetta etsiä haluttua tietoa kokeilemalla suoraan toisilta käyttäjiltä. Palvelin prosessoi haun ja paikallistaa käyttäjän, jolla on haettu sisältö ja tämän paikallistetun käyttäjän tiedot palvelin antaa haun tehneelle käyttäjälle. Näin ollen hakupolku lyhentyy. Tällaisessa mallissa palvelin käynnistää kommunikaation kahden käyttäjän välille [9].

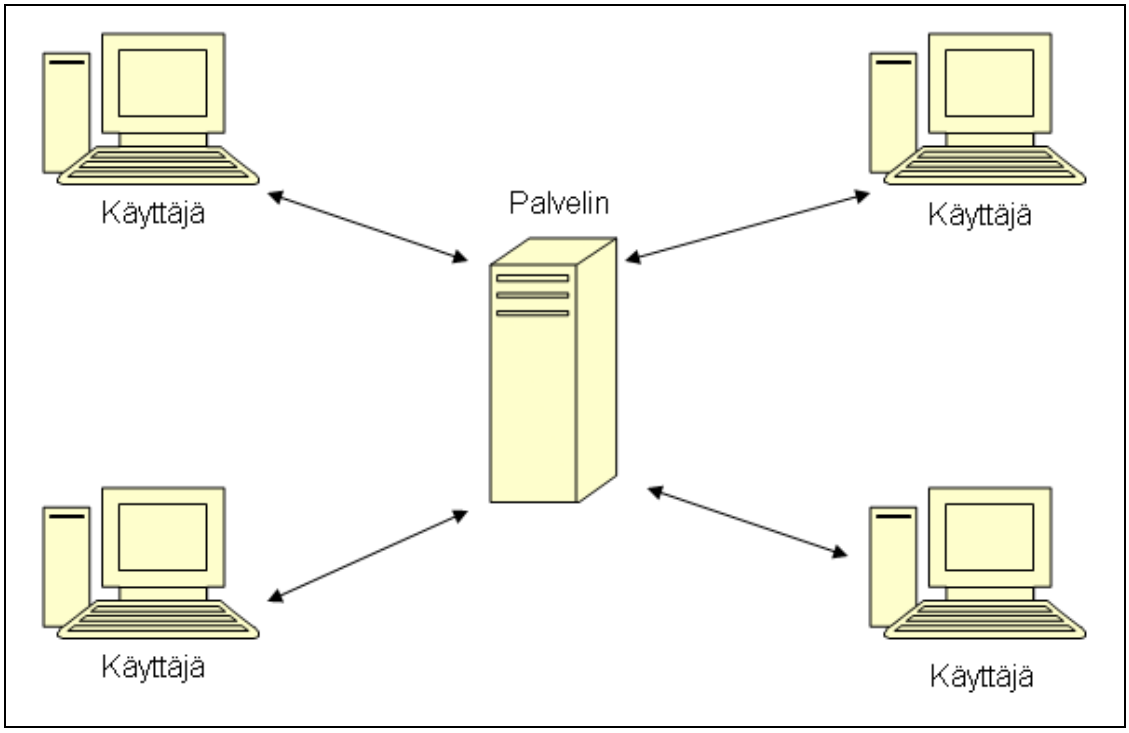
Tässä mallissa palvelin joutuu rasitukselle ja näin ollen on altis jossakin määrin hidastumiselle ja yleiselle suoritustason heikkenemiselle. Huolimatta palvelimen rasituksesta tämä vertaisverkkomalli on suosituin kehittäjien ja käyttäjien keskuudessa [9].

Vertaisverkko havainto-, haku- ja sisältöpalvelimella (discovery, lookup and content server), Tässä mallissa palvelin on määräävässä asemassa, kuten asiakas/palvelin arkkitehtuurissa (client/server). Kaikki mahdollinen toiminnallisuus on siirretty vertaisverkon jäseniltä palvelimelle. Vertaisverkon jäsenet eivät saa tässä mallissa ottaa yhteyttä suoraan toisiinsa. Kaikki resurssit on tallennettu tietokantaan, joka sijaitsee palvelimella. Kun käyttäjä hakee tietoa, sen sijaan, että käyttäjä kommunikoisi suoraan toisen käyttäjän kanssa, käyttäjä kysyy palvelimelta ja palvelin prosessoi pyynnön ja

näyttää tiedonlähteet. Tämän jälkeen käyttäjä valitsee haluamansa resurssit, jotka haluaa ladata itselleen. Seuraavaksi käyttäjä lähettää palvelimelle tiedonlatauspyynnön, jonka jälkeen palvelin lataa kyseisen datan itselleen vertaisverkon käyttäjältä, jolla kyseinen data on, jonka jälkeen latauspyynnön lähettänyt käyttäjä voi ladata datan itselleen palvelimelta. Tämä malli on kuitenkin vertaisverkkomalli, koska palvelin lataa itselleen haetun datan, joltain vertaisverkon käyttäjältä ja sen jälkeen lähettää ladatun tiedon, sitä hakeneelle käyttäjälle. Välityspalvelin on eräänlainen vertaisverkon jäsen, jonka kautta tieto kulkee. Kuvassa 2.7 on hahmotelma tällaisesta järjestelmästä [9].

Tämä malli on paras tiedon noutamiseen helposti verkon yli. Tämä malli tarjoaa myös käyttäjilleen hyvän turvallisuuden. Yhdenmukaisuus sisältöjen saatavuuteen ja tiedon luotettavuus ovat myös etuja [9].

Koska tällaiset mallit luottavat kokonaan palvelimeen, tästä seuraa se, että virheiden mahdollisuus, jotka vaikuttavat koko systeemiin kasvaa huomattavasti. Myös palvelimen hidastuminen on mahdollista jos palvelin saa liikaa palvelupyyntöjä kapasiteettiinsa nähden. Edellä mainittuja ongelmia ei tapahdu aikaisemmin esitellyissä malleissa, koska ne eivät tarvitse palvelimen toimintoja niin paljon kuin tämä malli [9].



Kuva 2.7: Vertaisverkko havainto-, haku- ja sisältöpalvelimella.

Seuraavaksi esitellään muutamia vertaisverkkosovelluksia. Tunnetuimpia vertaisverkko-ohjelmia ovat edesmennyt Napster ja nykyisin toimivat Kazaa [15], BitTorrent [3] ja eDonkey [10]. Jaetun laskennan tunnetuimpia ohjelmia ovat SETI@home [22] ja distributed.net RC5 salauksenpurku [8].

BitTorrent

BitTorrent [3] on tiedostojen siirtoon tarkoitettu vertaisverkkomainen järjestelmä. Kun perinteisesti tiedostojen jakamiseen tarvitaan palvelin ja paljon tiedonsiirtokapasiteettia, BitTorrent eliminoi tämän tarpeen sillä, että kaikki tiedostoa tai tiedostoja lataavat käyttäjät jakavat tiedostoja samanaikaisesti myös toisilleen.

BitTorrentilla jaettava tieto koostuu kahdesta osasta: varsinaisista jaettavista tiedostoista, sekä pienestä .torrent-tiedostopäätteisestä aputiedostosta, jonka tiedostojen jakaja luo asiakasohjelmansa avulla. Tiedon jakamiseen kuuluu yleensä kolme osapuolta: käyttäjät, seurantapalvelin eli trakkeri (engl. "tracker"), sekä WWW-palvelin joka levittää .torrent-tiedostoja. Yleensä nämä kaksi jälkimmäistä ovat yksi ja sama palvelin. Tiedonsiirto ilman varsinaista seurantapalvelinta on myös tullut mahdolliseksi laajennusten avulla.

Käyttäjä lataa .torrent-päätteisen tiedoston WWW-selaimellaan palvelimelta, ja BitTorrent-asiakasohjelma huolehtii sen käsittelystä. .torrent-tiedostosta löytyy varsinaisen trakkerin osoite, sekä tiedoston palasten tarkistussummat ja koot. Tämän jälkeen asiakasohjelma ottaa yhteyttä trakkeriin ja saa sitä kautta selville missä muita tiedostojen osia tarjoavat käyttäjät ovat, ja alkaa hakea heiltä tiedostojen osia.

WWW-palvelimia on tarpeen mukaan monenlaisia; mikään ei estä .torrent-tiedostojen laittamista normaaleille WWW-sivuille, mutta esimerkiksi julkiset BitTorrent-trackerit tarjoavat monipuolisempia kategorisointi- ja hakutoimintoja. Periaatteessa mikään ei estä muitakaan tapoja levittää .torrent-tiedostoja. Niitä voi levittää esimerkiksi IRCissä.

Tiedostojen lataajat tarvitsevat BitTorrent-asiakasohjelman ("client"). Koska BitTorrent on avoimen lähdekoodin ohjelma, erilaisia asiakasohjelmia on syntynyt useita. BitTorrentin tekijän virallinen asiakasohjelma on hyvin yksinkertainen, kun taas monipuolisemmissa ohjelmissa on erilaisia latausjono- ja nopeusrajoitusjärjestelmiä.

EDonkey2000

eDonkey2000 [10] on suosittu palvelinpohjainen vertaisverkko. Asiakasohjelmalla on lista palvelimista, joihin se yrittää ottaa yhteyttä yksitellen. Kun sopiva palvelin on löytynyt, asiakasohjelmalla voidaan suorittaa hakuja jaossa olevista tiedostoista ja siirtää niitä omalle koneelle.

Overnet [18] on eDonkeystä edelleen kehitetty ilman palvelimia toimiva vertaisverkko. Siinä jokainen tietokone on yhteydessä muutamaan muuhun tietokoneeseen. Jotkut koneista toimivat verkon 'solmukohtina' ja pitävät yllä normaalia isompaa tiedostolistaa hakutoimintojen nopeuttamiseksi.

eDonkey2000:n uusimmat versiot osaavat käyttää molempia vertaisverkkoja yhtä aikaa tai erikseen eikä Overnet-ohjelmaa enää kehitetä erikseen. eDonkey2000 myös varoittaa, jos ladattavan tiedoston nimi vaihtelee suuresti eri käyttäjien välillä. Tällöin ko. tiedoston sisältö on todennäköisesti aivan jotain muuta kuin mitä nimestä voisi päätellä.

eMule

eMule [11] on tiedostojen jakoon tarkoitettu vertaisverkko-ohjelma, joka toimii eDonkey-verkossa. Se tarjoaa uusia ominaisuuksia verrattuna alkuperäiseen eDonkey-asiakasohjelmaan. eMule on avoimen lähdekoodin ohjelmisto, joka on julkaistu GNU GPL -lisenssin alla.

eMulen pääominaisuuksia ovat

- ansiojärjestelmä, joka palkitsee lähettäjiä
- lähteiden vaihto, jonka avulla kaikkia tiedostoa jakavat käyttäjät löytyvät nopeasti
- automaattinen liikenteenpakkaus, joka säästää kaistaa
- automaattinen lähetysnopeudensäätely, joka etsii sopivan nopeuden, joka ei haittaa muuta liikennettä
- edistynyt rikkoutuneiden tiedostojen käsittely
- viestintäominaisuudet, kuten sisäänrakennettu IRC-asiakasohjelma ja yksityisviestit toisille käyttäjille sekä kaverilistat
- mahdollisuus käyttää omaa asiakasohjelmaansa verkon yli

Vertaisverkkojen yhteydessä puhutaan usein niiden väärinkäytöstä erityisesti piratismin osalta. Yleisön keskuudessa etenkin Napster sai negatiivista julkisuutta sen piirissä laittomasti levitettyjen musiikkitiedostojen takia. Myös lainsäätäjät ovat pohtineet, miten rikollinen toiminta verkoissa voidaan estää. Tietoverkoissa pätevät samat lait kuin verkkojen ulkopuolellakin. Musiikin levittäminen ilman tekijänoikeuskorvauksia on yhtä laitonta Internetissä kuin fyysisessä maailmassakin.

Vertaisverkkojen perusperiaate on jakaa laillista tietoa, tiedostoja ja muita resursseja verkon yli ilman suuria alkukustannuksia suhteellisen eristetyssä ympäristössä. Esimerkiksi lääketieteellisen tutkimuksissa vertaisverkkoja on hyödynnetty laskentakapasiteetin jakamisessa. Matemaattinen laskenta vaatii tietokoneelta huomattavan suurta kapasiteettia ja harvojen supertietokoneiden käyttö on erittäin kallista. Vertaisverkoissa samaa laskentaa voidaan tehdä tehokkaasti mutta edullisemmin kuin supertietokoneilla.

3 Toteutettavuuden haasteet

Tässä luvussa käsitellään vertailtavien arkkitehtuurien toteuttamisessa huomioon otettavia asioita. Käsiteltävät asiat ovat samankaltaisia ja näin ollen vertailukelpoisia. Toteutettavuudella tarkoitetaan sitä, miten helposti kyseinen järjestelmä on toteutettavissa ja mitä asioita tulee ottaa huomioon toteutuksessa.

3.1 Asiakas/palvelin arkkitehtuuri

Asiakas/palvelin arkkitehtuurin toteutettavuuden haasteet ovat samanlaisia kuin yleiset hajautetun järjestelmän haasteet. Seuraavaksi käydään läpi hajautetun järjestelmän ja asiakas/palvelin järjestelmän haasteet. Tämä luku perustuu kirjaan *Coulouris, G.: Distributed Systems, Concepts and Design* [6].

Heterogeenisyys

Internet koostuu heterogeenisestä joukosta tietokoneita ja verkkoja. Internet mahdollistaa käyttäjien pääsyn palveluihin ja se myös mahdollistaa erilaisten sovellusten suorittamisen. Heterogeenisyys eli monimuotoisuus koskee seuraavia asioita; verkot, tietokoneet, käyttöjärjestelmät ja ohjelmointikielet. Verkkojen protokollat eivät välttämättä ole yhteensopivat, esimerkiksi ethernet ja Internet verkot ovat yhteensopivat, mutta jos verkossa ei ole Internet protokollaa niin se tarvitsee implementaation sellaisesta. Myös data tyypit, kuten kokonaislukutyyppi (integer) vaihtelee laitteistoissa. Vaikka kaikkien käyttöjärjestelmien, jotka ovat Internetissä, täytyy implementoida Internet protokolla, niin ne eivät silti tarjoa samanlaista ohjelmointirajapintaa näille protokollille. Esimerkiksi viestien vaihtaminen eroaa UNIX ja Windows käyttöjärjestelmissä. Erilaiset ohjelmointikielet käyttävät erilaisia esitystapoja merkeille ja tietotyypeille. Näitä eroja täytyy huomioida, jotta ohjelmat toimivat keskenään. Ohjelmoijien ohjelmoimat ohjelmat eivät toimi keskenään jos ne eivät käytä standardeja. Asiakas/palvelin järjestelmissä heterogeenisyyden haasteet ovat vastaavia edellä mainittujen haasteiden kanssa.

Väliohjelma (middleware)

Termi väliohjelma (middleware) tarkoittaa ohjelmatasoa, joka tarjoaa ohjelmointiabstraktion, kuten myös heterogeenisyyden peittämisen verkoissa, laitteistoissa, käyttöjärjestelmissä ja ohjelmointikielissä. Esimerkki väliohjelmasta on CORBA [5] (Common Object Request Broker). Jotkin väliohjelmista tukevat vain yhtä ohjelmointikieltä, esimerkki tällaisesta on Java Remote Method Invocation (RMI) [14]. Väliohjelma tarjoaa yhtenäisen laskentamallin palvelimien ja hajautettujen järjestelmien ohjelmoijille. Mahdolliset mallit voivat tarjota esimerkiksi etäobjektin herättämisen (remote object invocation), etätapahtuman huomioimisen (remote event notification) ja etätietokantapääsyn (remote SQL access). Esimerkiksi CORBA tarjoaa etäobjektin herättämisen, joka sallii ohjelman objektin herättää toisella koneella olevan metodin. Asiakas/palvelin järjestelmissä väliohjelmien käyttö on tärkeässä asemassa ja niitä käytetään yleisesti.

Avoimuus

Hajautetun järjestelmän avoimuus on piirre, joka määrittelee, voidaanko systeemiä laajentaa tai muokata uudelleen. Hajautettujen järjestelmien avoimuus määritellään lähinnä siten, miten uusia resurssinjakopalveluita voidaan sisällyttää järjestelmään. Avoimuutta ei voida saavuttaa, elleivät spesifikaatiot ja dokumentaatiot ole ohjelmoijien saatavilla. Avoimien systeemien yhteinen piirre on, että niiden avainasemassa olevat rajapinnat ovat julkisia. Avoimet hajautetut järjestelmät perustuvat yhtenäiseen kommunikointimekanismiin ja rajapintojen julkistamiseen. Avoimet hajautetut järjestelmät voidaan muodostaa heterogeenisista laitteistoista ja ohjelmistoista, mutta jokainen niihin lisättävä uusi komponentti on testattava huolellisesti jotta systeemi toimisi oikein. Asiakas/palvelin järjestelmissä avoimuus ei välttämättä ole kovin tärkeässä asemassa. Esimerkkinä voidaan pitää Windows ja Linux käyttöjärjestelmiä; Windows on suljetun lähdekoodin käyttöjärjestelmä ja Linux on avoimen lähdekoodin käyttöjärjestelmä. Molemmille käyttöjärjestelmille on saatavana asiakas ja palvelin ohjelmistoja, mutta käyttöjärjestelmien luonteesta johtuen Linuxille on saatavilla enemmän avoimen lähdekoodin asiakas ja palvelin ohjelmistoja. Avoimuus on toteutuksen kannalta vapaavalintainen, sillä voidaan saavuttaa joitain etuja, mutta se ei välttämättä ole pakollista. Suljetut järjestelmät ovat yleensä maksullisia ja avoimet taas ilmaisia [19].

Turvallisuus

Suuri osa informaatiosta, joka on saatavilla hajautetuissa järjestelmissä, on arvokasta sen omistajalle. Näin ollen turvallisuus on avainasemassa. Asiakas/palvelin järjestelmien turvallisuutta on käsitelty luvussa 5.

Skaalautuvuus

Hajautetut järjestelmät toimivat tehokkaasti monissa eri mittakaavoissa, pienistä Intraneteistä Internetiin. Systeemiä luonnehditaan skaalautuvaksi, jos se pysyy toimintakuntoisena, vaikka käyttäjien tai resurssien määrä kasvaa huomattavasti. Skaalautuvan hajautetun järjestelmän suunnittelussa on erilaisia haasteita:

- *Fyysisten resurssien hinnan kontrollointi*: Kun resurssien tarve kasvaa, pitäisi olla mahdollista laajentaa systeemiä kohtuulliseen hintaan. Laitteistojen pitäisi olla helposti ja halvasti päivitettäviä. Laitteistoja hankittaessa tämä asia tulee ottaa

huomioon. Esimerkiksi kun käyttäjämäärät kasvavat, voi uusien palvelimien hankinta tulla kyseeseen.

- *Tehon häviön kontrollointi*: Palvelimilla olevan tiedon etsimiseen voidaan käyttää erilaisia algoritmeja. Tietoa voidaan myös säilyttää erilaisissa tietorakenteissa. Algoritmit, jotka käyttävät hierarkkisia tietorakenteita skaalautuvat paremmin kuin lineaarisia tietorakenteita käyttävät algoritmit.
- *Suorituskyvyn pullonkaulojen välttäminen*: Jaetut resurssit voivat muodostua pullonkauloiksi hajautetuissa järjestelmissä, kuten myös asiakas/palvelin järjestelmissä, sillä niitä voi käyttää yhtä aikaa monet käyttäjät. Tällaiseen tilanteeseen ratkaisuna voi olla välimuistin käyttö tai tiedon replikointi.

Yleensä ottaen käyttäjien ja resurssien määrän kasvaessa, systeemin tai ohjelmiston ei tulisi muuttua, mutta tämä on hankala tavoite saavuttaa. Skaalautuvuus on hallitseva teema hajautetuissa järjestelmissä, kuten myös asiakas/palvelin järjestelmissä.

Vikojen käsittely

Tietokonejärjestelmiin tulee vikoja ajoittain. Kun vika ilmenee laitteistossa tai ohjelmassa niin tämä saattaa johtaa tuottaa vääriä tuloksia tai toiminta saattaa loppua kokonaan. Hajautettujen järjestelmien ja asiakas/palvelin järjestelmien viat ovat yleensä osittaisia eli osa komponenteista voi kaatua, mutta osa jatkaa toimintaansa. Siksi vikojen käsittely hajautetuissa järjestelmissä on hankalaa. Seuraavaksi käsitellään muutamia tekniikoita, miten käsitellä vikoja:

- *Vikojen tunnistaminen*: Jotkin viat on mahdollista tunnistaa. Esimerkiksi, tarkistussummia (checksum) voidaan käyttää havaitsemaan viallinen data viesteissä tai tiedostoissa. Suurin haaste on hallita mahdollisten virheiden mahdollinen sattuminen, joita ei voida havaita, mutta jotka voivat sattua. Eli kyseessä on virheiden ennakointi.
- *Vikojen peittäminen*: Joitakin havaittuja vikoja voidaan peittää tai tehdä muuten vähemmän vakavimmiksi. Esimerkkeinä, viestejä voidaan lähettää uudelleen ja tiedostoja voidaan kirjoittaa kahdelle levylle ja näin toiselle levyllä on kunnossa olevaa dataa jos toiseen tulee vikaa.

- *Vikojen sietäminen*: Suurin osa Internetin palveluista tuottaa virheitä. Ei ole kovinkaan käytännöllistä, että nämä palvelut yrittäisivät havaita tai piilottaa kaikkia virheitä, koska kyseessä on laaja verkko, jossa on paljon komponentteja. Esimerkiksi, kun Internetselain ei saa yhteyttä palvelimeen, selain ei anna käyttäjän odottaa kovin pitkään, vaan sen sijaan ilmoittaa käyttäjälle, että palvelimeen ei saada yhteyttä. Eli käyttäjän on siedettävä mahdolliset virheet.
- *Virheistä toipuminen*: Virheistä toipumisella tarkoitetaan sitä, että kun virhe on tapahtunut, niin mahdollisesti kesken jäänyt toiminto voidaan jatkaa ja, että tieto on pysynyt tallessa eli sekin voidaan palauttaa.
- *Ylimäärä (redundancy)*: Palveluita voidaan tehdä sietämään vikoja paremmin käyttämällä ylimääräisiä komponentteja. Esimerkiksi; Internetissä olevien reitittimien välillä tulisi olla kaksi käytettävää reittiä. DNS systeemissä jokainen nimitietokanta tulisi olla vähintään kahdella palvelimella. Tietokannat tulisi replikoida useammalle palvelimelle, että tieto olisi varmasti tallessa.

Yhdenaikaisuus

Palvelut ja sovellukset tarjoavat resursseja, joita asiakkaat jakavat keskenään hajautetuissa järjestelmissä, kuten myös asiakas/palvelin järjestelmissä. Siksi on mahdollista, että useat asiakkaat yrittävät käsitellä samaa jaettua dataa yhtä aikaa. Jokaisen jaetun objektin tulisi olla vastuussa siitä, että se toimii oikein yhdenaikaisessa ympäristössä. Tämä koskee palvelimia, sekä sovellusten objekteja. Ollakseen toimiva objekti yhdenaikaisessa ympäristössä, sen tulee toimia siten, että data pysyy yhtenäisenä. Tämä voidaan toteuttaa esimerkiksi käyttämällä semaforia (semaphores), joita käytetäänkin useassa käyttöjärjestelmässä.

Tuntumattomuus (transparency)

Tuntumattomuus määritellään siten, että käyttäjä näkee vain koko systeemin, ei sen itsenäisiä hajautettuja komponentteja. Coulouris [6] määrittelee kahdeksan eri tuntumattomuuden muotoa.

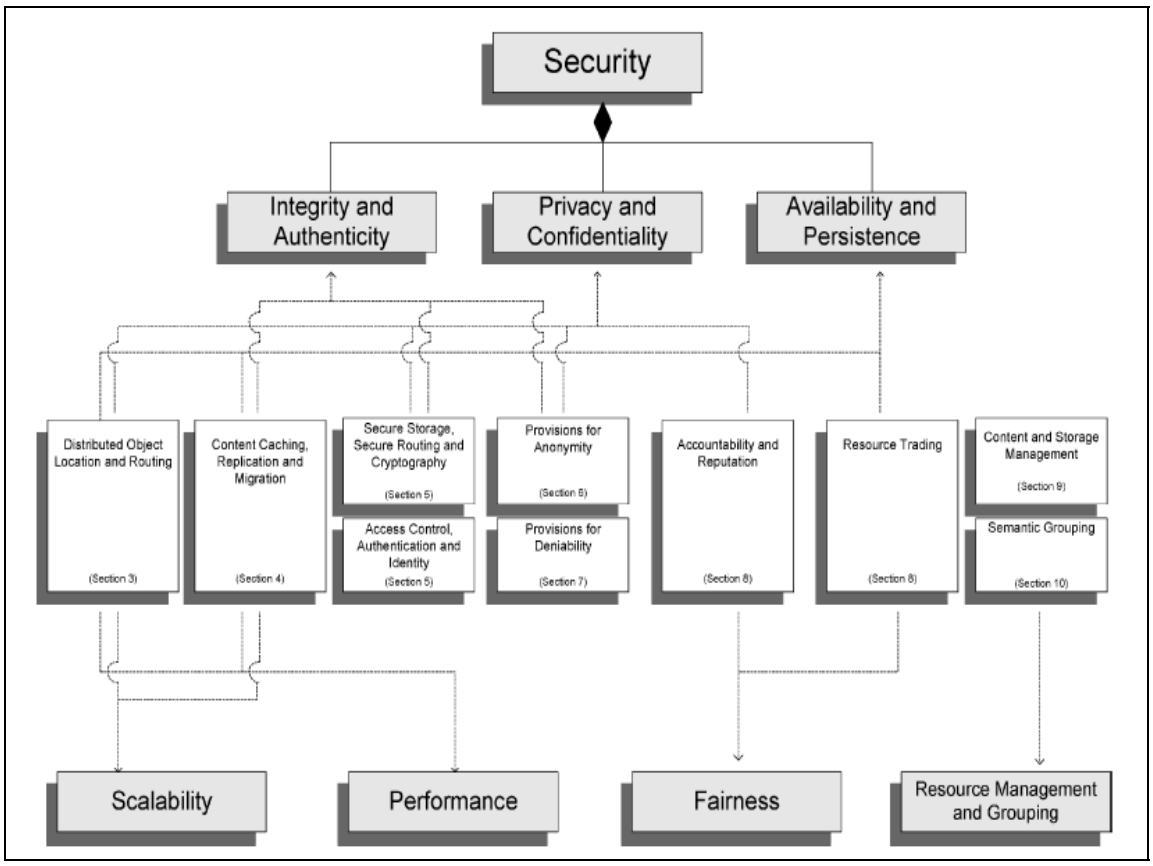
- *Pääsyn tuntumattomuus* mahdollistaa paikallisen ja etäresurssien käyttämisen käyttämällä identtisiä operaatioita

- *Paikallistamisen tuntumattomuus* mahdollistaa pääsyn resursseihin ilman tietoa siitä, missä resurssit fyysisesti sijaitsevat
- *Yhdenaikaisuuden tuntumattomuus* mahdollistaa useiden prosessien toimimisen hajautettujen resurssien kanssa yhtä aikaa, ilman että ne häiritsisivät toisiaan
- *Replikoinnin tuntumattomuus* mahdollistaa usean resurssi-instanssin (palvelimen, tietokannan, jne.) olla käytössä ja näin luotettavuus ja suorituskyky lisääntyy ilman, että käyttäjät tietävät tiedon olevan replikoitua
- *Virheiden tuntumattomuus* mahdollistaa virheiden piilottamisen ilman, että käyttäjien toiminta häiriintyy niistä
- *Liikkuvuuden tuntumattomuus* mahdollistaa resurssien ja asiakkaiden liikkuvuuden systeemissä ilman, että se vaikuttaisi käyttäjiin tai ohjelmiin
- *Suorituskyvyn tuntumattomuus* mahdollistaa systeemin uudelleen konfiguroinnin suorituskyvyn lisäystä varten, silloin kun työmäärä vaihtelee
- *Skaalautuvuuden tuntumattomuus* mahdollistaa systeemin ja sovellusten laajenemisen ilman, että systeemin rakenne tai ohjelmien algoritmit muuttuvat

Kaksi tärkeintä tuntumattomuudenlajia ovat paikallistamisen ja pääsyn tuntumattomuus, niiden olemassaolo tai niiden puutuminen vaikuttaa eniten hajautettujen resurssien toteuttamiseen. Niihin viitataan usein yhteisellä nimikkeellä, verkon tuntumattomuus.

3.2 Vertaisverkko

Vertaisverkkoja koskee samat toteutettavuuteen liittyvät asiat kuin asiakas/palvelin arkkitehtuurissa, mutta näiden lisäksi tulee huomioida myös seuraavaksi esiteltävät asiat. Tämä luku perustuu *Androutsellis-Theotokis, S. & al* tekemään tutkimukseen [2]. Kuvassa 3.1 nähdään miten erilaiset suunnittelussa huomioon otettavat asiat vaikuttavat vertaisverkkojen perusominaisuuksiin. Kuvan reunoilla (ylhäällä ja alhaalla) on kaikista tärkeimmät ominaisuudet, jotka vaikuttavat vertaisverkkojen toimintaan.



Kuva 3.1. *Hahmotelma siitä miten erilaiset asiat vaikuttavat vertaisverkkojen ominaispiirteisiin. [2]*

Kaikkein tärkein ominaisuus on turvallisuus, jota tarkastellaan seuraavasti:

Rehellisyys ja tarkkuus (integrity and authenticity): Ominaisuus, joka vartioi, että data on tarkkaa ja täydellistä. Mm. valtuuttamattomat henkilöt eivät voi vaihtaa dataa ja mahdolliset viholliset eivät voi tarjota väärennettyä tiedostoa pyydetyn tiedoston tilalle.

Yksityisyys ja luottamuksellisuus (privacy and confidentiality): Ominaisuus, jonka tarkoitus on varmistaa, että data on saatavilla vain siihen oikeutetuille tahoille. Kuten myös se, että on kontrolli siitä datasta, jota kerätään, käytetään ja kuinka sitä säilytetään.

Saatavuus ja pysyvyys (Availability and persistence): Takaa sen, että oikeutetut käyttäjät pääsevät halutessaan dataan käsiksi. Eli järjestelmän pitää olla vakaa, vaikka sattuisi virheitä.

Seuraavat ominaisuudet vaikuttavat myös turvallisuuteen epäsuorasti:

Skaalautuvuus (Scalability): Järjestelmän pitäminen suorituskykyisenä, riippumatta siitä miten paljon verkossa on käyttäjiä tai dokumentteja. Käyttäjien tai dokumenttien dramaattinen kasvu ei saa vaikuttaa lähes ollenkaan verkon toimivuuteen.

Suorituskyky (performance): Se aika, joka tarvitaan kun suoritetaan systeemin vaatimia toimintoja, tyypillisesti dokumenttien julkaisua, etsintää ja noutoa.

Oikeudenmukaisuus (fairness): Takaa sen, että käyttäjät tarjoavat, lataavat ja lähettävät tiedostoja oikeudenmukaisesti.

Resurssien hallinta mahdollisuudet (resource management capabilities): Vertaisverkot sisältävät tavallisimmillaan dokumenttien julkaisu-, etsintä- ja nouto mekanismit, mutta kehittyneemmät vertaisverkkosovellukset voivat näiden lisäksi sisältää myös dokumenttien editointi tai poisto mekanismin, säilytystilan hallinnoinnin ja metadatan operaatiot.

Semanttinen tiedon ryhmittely (semantic grouping of information): Tietoa voidaan ryhmitellä esimerkiksi sen sisällön, sen etäisyyden verkossa, paikallisuuden tai muiden siteiden mukaan.

Kuvassa 3.1 laatikot jotka ovat keskellä, edustavat päätöksiä, jotka vaikuttavat edellä mainittuihin ominaisuuksiin. Kuva 3.1 on UML muodossa ja näin ollen, esimerkiksi suorituskykyyn (performance) vaikuttavat tekijät ovat; hajautettu objektien paikallistaminen (distributed object location) ja ohjausmekanismi (routing mechanism), datan replikointi (data repclication), välimuistinkäyttö (caching) ja muuttoalgoritmit (migration algorithms). Seuraavaksi näitä suunnittelussa tehtäviä päätöksiä ja ominaisuuksia käsitellään hieman tarkemmin.

Hajautettu objektien paikallistaminen ja ohjaus (distributed object location and routing): Vertaisverkkojen toiminta luottaa vertaisverkon käyttäjien koneisiin (solmuihin) ja verkkoyhteyksiin niiden välillä (kaaret). Vertaisverkko muodostetaan fyysisen verkon päälle ja sitä kutsutaan päällysverkoksi (overlay network). Tämän päällysverkon topologia, rakenne, keskittymisen aste ja tiedonohjaus sekä paikannusmekanismit, jotka se luo

viesteille ja sisällölle ovat elintärkeitä vertaisverkkosysteemin toiminnan kannalta. Ne vaikuttavat vertaisverkon vikasietoisuuteen, itse ylläpitoon, virheisiin mukautumiseen, suorituskykyyn, skaalautuvuuteen ja turvallisuuteen. Toisin sanoen se vaikuttaa lähes kaikkeen vertaisverkossa. Vertaisverkkoja on käsitelty luvussa 2.

Sisällön välimuistinkäyttö, replikointi ja muutto (content caching, replication and migration): Vertaisverkko systeemit luottavat tiedon replikointiin. Tieto yritetään replikoida useammalle kuin yhdelle verkon solmulle. Näin saavutetaan parempi suorituskyky ja ehkäistään sensurointiyrityksiä. Replikointitavat voidaan kategorisoida seuraavasti.

- *Passiivisella replikoinnilla (passive replication)*, tarkoitetaan sitä, että tieto replikoidaan luonnostaan eli tämä tapahtuu automaattisesti kun vertaisverkon jäsenet lataavat toisiltaan dataa.
- *Välimuistiperustainen replikointi (cache-based replication)* tarkoittaa sitä, että kun tehdään haku verkossa, tieto kulkee verkon solujen kautta jättäen jälkensä niihin. Näin saavutetaan parempi tiedon saatavuus seuraaville hauille.
- *Aktiivinen replikointi (active replication)* tarkoittaa sitä, että aktiivinen tiedon replikointi ja tiedonmuuttometodit otetaan käyttöön, jotta saavutettaisiin parempi datan lokalisaatio, saatavuus sekä suorituskyky.
- *Introspektiivisellä replikoinnin hallinnalla (introspective replica management)* tarkoitetaan sitä, että tietoliikennettä tarkkaillaan ja luodaan kopioita datasta sinne missä sitä tarvitaan, eli vastataan kysyntään.
- *Dynaamisilla tiedon replikointi algoritmeilla (dynamic replica management)* tarkoitetaan sellaisia algoritmeja, jotka osaavat sijoittaa mahdollisimman vähän kopioita tiedosta säilyttäen kuitenkin verkon toiminnallisuuden. Tällaiset algoritmit ovat suunniteltu palvelemaan asiakkaitten latenssi aikojen pienentämistä varten ja vähentämään palvelinten kuormitusta.

Turvallinen tiedonsäilytys (secure storage): On olemassa paljon erilaisia tiedonsalakirjoitusalgoritmeja ja protokollia, joilla taataan tiedon turvallinen säilytys julkaistaessa ja säilyttäessä tietoa vertaisverkossa.

Turvallinen tiedonohjaus (secure routing): turvallisen tiedonohjauksen tarkoituksena on ratkaista ongelma, joka liittyy ilkeisiin verkonsolmuihin. Nämä solmut yrittävät korruptoida, tuhota, estää pääsyn tai jotka tarjoavat väärennöksiä datasta, jota liikutellaan verkonsolmuista toisiin.

Pääsynhallinta, todentaminen ja identiteetin hallinta (access control, authentication and indentity management): Asioita, jotka liittyvät pääsynhallintaan, todentamiseen ja identiteetin varmistamiseen yleensä väheksytään tai jätetään huomiotta vertaisverkoissa. Hajautetussa ympäristössä saman henkilön on helppo esiintyä useana eri henkilönä, erityisesti sellaisissa verkoissa, joissa on suuri verkonsolmujen vaihtuvuus.

Nimettömyys (provisions for anonymity): Nimettömyys on monessa vertaisverkossa tärkeä asia. Nimettömyyttä tarvitsevat yleensä sisällönjulkaisija, vertaisverkonsolmu, sisältö itse ja suoritettu haku.

Kiellettävyys (provisions of deniability): Kiellettävyys viittaa vertaisverkoissa siihen, miten jokainen käyttäjä voi kieltää tiedon siitä, mitä heidän koneelleen on tallennettu. Seurauksena tästä on, että käyttäjiä ei voida pitää vastuussa siitä tiedosta mitä heidän koneelleen on tallennettu. Tällä tiedolla viitataan siihen tietoon, mitä tarvitaan, että vertaisverkko toimii. Eli esimerkiksi erilaisia välitystietoja.

Seurattavuus ja maine (accountability and reputation): Toimivuus, suorituskyky ja saatavuus vertaisverkoissa luottavat vertaisverkon käyttäjien vapaaehtoiseen osallistumiseen. Juuri tämän takia on hyödyllistä kehittää jonkinlainen palkinto tai virikejärjestelmä, sekä myös jonkinlainen seurantasysteemi, joka rohkaisee vertaisverkonkäyttäjiä osallistumaan vertaisverkon toimintaan eli lähinnä jakamaan tiedostoja. Jos tällaisia rohkaisusysteemeitä ei ole niin se saattaa johtaa suorituskyvyn heikkenemiseen ja tiedostojen saatavuuden heikkenemiseen, pahimmillaan jopa verkon luhistumiseen. Esimerkkinä tällaisessa tapauksessa on niin sanottu vapaamatkustaja ongelma (free-ride), jossa käyttäjät vain lataavat tiedostoja, mutta eivät tarjoa yhtään tiedostoa. Maineeseen liittyvien mekanismien pääasiallinen tarkoitus on jakaa paikallisesti generoitu informaatio vertaisverkonkäyttäjän maineesta muiden vertaisverkonkäyttäjien kesken. Tämä tieto levitetään koko verkkoon ja näin syntyy globaali mainesysteemi, kun

jokaisen vertaisverkonjäsenen maine leviää. Prosessin aikana mainetietoja täytyy käsitellä turvallisesti.

Sisällön ja säilytystilanhallinta (content and storage management): Resursseja, joita vertaisverkot tyypillisesti käyttävät ovat sisältö (tiedostot), säilytystila (kiintolevyt) ja siirtokapasiteetti (kaistanleveys). Vertaisverkon perusoperaatiot ovat; tiedon lisäys, tiedon etsintä ja tiedon nouto. Näitä tarvitaan, että vertaisverkko olisi toimiva. Näiden lisäksi eri vertaisverkkosovellukset saattavat tarvita erilaisia resurssinhallintasysteemeitä, kuten sisällönpoistomekanismin, sisällönpäivitysmekanismin, versionhallinnan sisältöön, säilytystilanhallintamekanismin ja kaistankäyttöön liittyviä mekanismeja. Eri vertaisverkkosovellukset käyttävät näitä ominaisuuksia tarpeittensa mukaan.

Tiedon ryhmittely (semantic grouping of information): On olemassa paljon erilaisia mekanismeja, joilla tehdään niin sanottu päällysverkko (overlay network) fyysisen verkon päälle. Nämä mekanismit ottavat huomioon tiedon paikallisuuden ja etäisyyden fyysisessä verkossa. Näillä tiedonryhmittely mekanismeilla pyritään vähentämään kommunikaation kustannuksia.

3.3 Vertailu

Tässä kohdassa vertaillaan asiakas/palvelin arkkitehtuurin ja vertaisverkon eroja ja yhtäläisyyksiä toteutettavuuden kannalta. Vertailtavat asiat ovat edellä mainittuja ja arviot perustuvat edellä olevaan tekstiin ja myös omiin kokemuksiin ja tietoihin. Vertailtaville asioille annetaan pisteet, siitä miten tärkeä asia on toiminnallisuuden kannalta; 1 = ei tärkeä, 2 = melko tärkeä, 3 = tärkeä. Pisteitä annetaan myös siitä, miten helposti kyseinen ominaisuus voidaan toteuttaa, 1 = helppo, 2 = kohtalainen, 3 = vaikea.

Verrattava asia	*Asiakas/palvelin*		*Vertaisverkko*		
	Tärkeys	Toteutettavuus	Tärkeys	Toteutettavuus	Selitys
Heterogeenisyys	2	3	2	2	Heterogeenisyys on molemmissa järjestelmissä kohtalaisen suuressa asemassa

Verrattava asia	*Asiakas/palvelin*		*Vertaisverkko*		
	Tärkeys	Toteutettavuus	Tärkeys	Toteutettavuus	Selitys
Väliohjelmat	2	2	1	2	Väliohjelmia käytetään enimmäkseen asiakas/palvelin järjestelmissä
Avoimuus	1	1	1	1	Avoimuus on vapaavalintainen ominaisuus
Turvallisuus	3	3	3	3	Turvallisuus on avainasemassa molemmissa järjestelmissä. Se on myös hankala toteuttaa
Skaalautuvuus	3	3	3	3	Skaalautuvuus on tärkeä ominaisuus molemmissa järjestelmissä. Se on myös hankala toteuttaa
Vikojenkäsittely	2	2	2	2	Vikojen käsittelyä lisäämällä saadaan vakaampi järjestelmä ja toteuttamisen vaikeus on suhteellinen vikojen käsittelyn tarkkuuteen
Yhdenaikaisuus	2	2	1	2	Yhdenaikaisuus on asiakas/palvelin järjestelmissä olennainen osa. Vertaisverkossa yhdenaikaisuutta ei välttämättä tarvita
Yksityisyys ja luottamuksellisuus	3	2	3	3	Vertaisverkoissa yksityisyys ja luottamuksellisuus ovat tärkeitä piirteitä ja niiden toteuttaminen on hankalaa. Asiakas/palvelin järjestelmissäkin asia tulee ottaa huomioon, mutta niissä asia on helpompi toteuttaa, johtuen järjestelmän luonteesta
Tiedon saatavuus ja pysyvyys	3	2	3	2	Tiedon saatavuus ja pysyvyys ovat molemmissa järjestelmissä tärkeää ja toteuttaminen on suhteellisen hankalaa
Rehellisyys ja tarkkuus	2	1	3	2	Rehellisyys ja tarkkuus liittyvät lähinnä vertaisverkkoihin
Oikeudenmukaisuus	1	1	3	2	Oikeudenmukaisuus liittyy vertaisverkkoihin. Asiakas/palvelin järjestelmissä oikeudenmukaisuutta ei tarvita
Resurssienhallinta	2	2	2	2	Resurssien hallinta on kohtalaisen tärkeä ja yhtä vaikea toteuttaa

Verrattava asia	*Asiakas/palvelin*		*Vertaisverkko*		
	Tärkeys	Toteutettavuus	Tärkeys	Toteutettavuus	Selitys
					molemmissa järjestelmissä
Tiedon semanttinen ryhmittely	1	1	2	3	Tiedon semanttinen ryhmittely liittyy vertaisverkkoihin. Asiakas/palvelin järjestelmissä semanttista ryhmittelyä ei tarvita
Välimuistin käyttö	2	2	2	2	Välimuistin käyttö on molemmissa järjestelmissä kohtalaisen tärkeää ja yhtä vaikea toteuttaa
Tiedon replikointi	2	2	2	2	Tiedon replikointia käytetään molemmissa järjestelmissä ja se on kohtalaisen vaikea toteuttaa
Turvallinen tiedon säilytys	3	2	3	2	Tiedon turvallinen säilytys on molemmissa järjestelmissä olennaista ja sen toteuttaminen on yhtä vaikeaa
Pääsynhallinta	3	2	2	2	Pääsyn hallinta liittyy lähinnä asiakas/palvelin järjestelmiin. Vertaisverkoissa pääsynhallinta on vapaavalintainen, mutta sitä ei yleensä käytetä
Kiellettävyys	1	1	2	3	Kiellettävyys liittyy vertaisverkkoihin. Asiakas/palvelin järjestelmissä kiellettävyyden ongelmaa ei ole
Seurattavuus ja maine	1	2	2	3	Seurattavuus ja maine ovat vertaisverkkojen ongelma. Asiakas/palvelin järjestelmissä seurattavuutta voidaan joskus tarvita, mutta maineella ei ole merkitystä
Nimettömyys	1	1	2	2	Nimettömyys liittyy vertaisverkkoihin ja on kohtalaisen tärkeä ominaisuus ja suhteellisen hankala toteuttaa. Asiakas/palvelin järjestelmissä nimettömyys ei ole olennaista
Turvallinen tiedon ohjaus	2	2	2	2	Turvallinen tiedon ohjaus on molemmissa järjestelmissä kohtalaisen tärkeää ja suhteellisen hankala toteuttaa
Suorituskyky	3	3	3	3	Suorituskyky on molemmissa

Verrattava asia	*Asiakas/palvelin*		*Vertaisverkko*		
	Tärkeys	Toteutettavuus	Tärkeys	Toteutettavuus	Selitys
					järjestelmissä avainasemassa ja erittäin olennaista järjestelmän toimivuuden kannalta. Suorituskyvyn saavuttaminen on vaikea toteuttaa
Keskiarvo	***2.05***	***1.91***	***2.23***	***2.27***	Vertaisverkko on hieman vaikeampi toteuttaa ja siinnä on myös hieman vaativampi toteutettavuuden kannalta
3 määrä	***7/22***	***4/22***	***8/22***	***7/22***	
2 määrä	***9/22***	***12/22***	***11/22***	***14/22***	
1 määrä	***6/22***	***6/22***	***3/22***	***1/22***	

Luvuista voimme päätellä, että vertaisverkkojen toteuttaminen on hieman vaativampaa kuin asiakas/palvelin järjestelmien toteuttaminen. Luvuista nähdään myös, että vertaisverkossa on hieman enemmän vaativampia toteutus kohteita. Mutta nämä tulokset ovat vain suuntaa antavia ja ne ovat hyvin lähellä toisiin eli tämä on vain johtopäätös tuloksista.

4 Verkonkuormitus

Tässä luvussa vertaillaan neljän eri sisällöntoimitus-systeemin eroja verkonkuormituksen suhteen. HTTP verkkoliikennettä, Akamai [1] sisällöntoimitus verkkoa (content delivery network (CDN)), Kazaa [15] vertaisverkkosovellusta ja Gnutella [12] vertaisverkkosovellusta. Luku perustuu Washingtonin Yliopistossa (WU) tehtyyn tutkimukseen [21]. Siinä kerättiin tietoa kaikesta tietoliikenteestä, sekä lähtevästä, että saapuvasta. WU:ssa on yli 60 000 tietoverkkojen käyttäjää. Tutkimus kesti 9 päivää, jonka aikana tapahtui yli 500 miljoona tapahtumaa ja tietoa siirtyi yli 20 teratavua. Tutkimus antaa viitteitä siitä, miten eri sisällön toimitus systeemit eroavat toisistaan myös yleisesti ottaen. Huomioitavaa kuitenkin on, että tämä on vain yksi tutkimus yhden yliopiston osalta. Tutkimus on myös suhteellisen vanha, vuodelta 2002.

Tutkimuksessa asiakas/palvelinperustaista liikennettä edusti WWW-liikenne ja Akamai sisällöntoimitusverkko (CDN), vertaisverkkoperustaista liikennettä Kazaa ja Gnutella. Ylimmältä tasolta katsoen nämä järjestelmät omaavat saman roolin, eli ne jakavat sisältöä käyttäjille. Kuitenkin, näiden systeemien arkkitehtuurit eroavat toisistaan merkittävästi. Nämä erot vaikuttavat niiden suorituskykyyn, verkonkuormitukseen ja miten välimuistin käyttö voi näihin vaikuttaa.

WWW

Internetin perusarkkitehtuuri on yksinkertainen käyttäen HTTP-protokollaa [13]; käyttäjien koneilla toimivat asiakasohjelmat, mm. Internetselaimet, pyytävät objekteja Internetpalvelimilta. Internetobjektit ovat yleensä pieniä (5-10kt), mutta nämä objektit sisältävät paljon välitystietoa. Myös erittäin suuria objekteja on olemassa. WWW objektien määrä on huikea, niitä on miljardeja ja niitä tulee jatkuvasti lisää. Suurin osa WWW objekteista on staattisia, mutta myös dynaamiset objektit lisäävät suosiotaan jatkuvasti.

Sisällöntoimitusverkot

Sisällöntoimitusverkot ovat kokoelma palvelimia, jotka on sijoitettu strategisesti ympäri Internetiä. Sisällöntarjoajat, kuten Internetsivut ja Streamatun videon lähteet, tekevät sopimuksen kaupallisen CDN-tarjoajan kanssa. Kun sisältö on CDN-verkossa niin se replikoidaan palvelimille ja näin ollen sisältö on saatavilla hyvin. Suurimmat sisällöntoimitusverkot sisältävät tuhansia palvelimia, jotka on sijoitettu ympäri Internetiä ja näin ollen sisällöntoimitusverkot pystyvät toimimaan sekä palvelemaan suuria käyttäjämääriä. Sisällöntoimitusverkot ovat integroitu tiukasti olemassa olevan Internet arkkitehtuurin kanssa. Ne käytännössä vain ohjaavat käyttäjän lähimmälle palvelimelle. Sisällöntoimitusverkossa objektit ovat samanlaisia kuten HTTP-protokollassa. Sisällöntoimitusverkot tarjoavat yleensä staattista sisältöä, kuten kuvia, mainoksia tai media klippejä; sisällöntoimittajat määrittelevät itse sisältönsä.

Vertaisverkot

Kuten jo edellisissä luvuissa on todettu, vertaisverkonjäsenet toimivat asiakkaina ja palvelimina. Tiedosto jota ladataan, on yleensä myös muiden käyttäjien ladattavissa. Osallistuminen on täysin vapaaehtoista. Yleensä sisältö kärsii huonosta saatavuudesta ja suhteellisen huonoista verkkoyhteyksistä. Toisin kuin Internet- ja CDN-systeemeissä, pääsääntöinen käyttötarkoitus on ei-interaktiivinen, erä-tyylinen (batch-style) tiedostojenlataus. Vertaisverkot eroavat toisistaan siinä, miten tiedosto ladataan, kun tiedosto on ensin löydetty. Suurin osa systeemeistä luo suoranyhteyden tarjoajan ja lataajan välille. Latenssia parantavaa optimointia käyttävät systeemit lataavat tiedoston osina monilta eri käyttäjiltä yhtä aikaa.

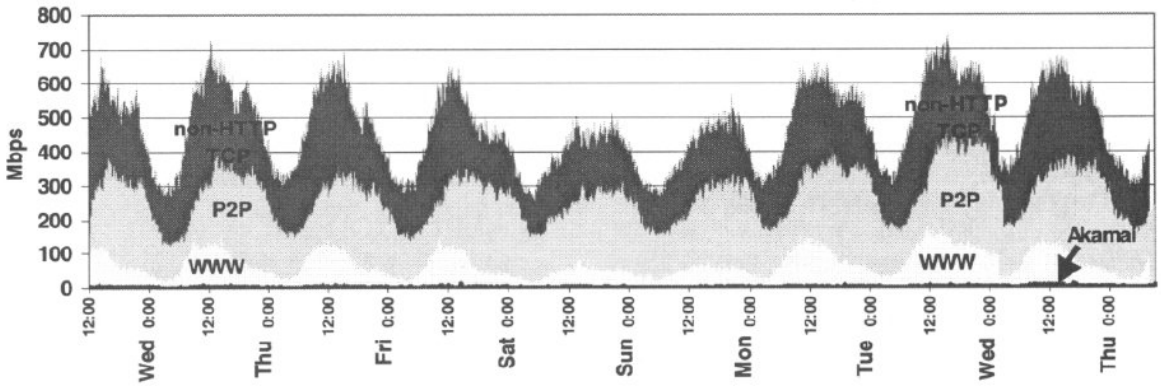
Korkeantason sisällön tunnusmerkit

Tämä osio esittelee korkeantason tunnusmerkkejä tutkimuksen keräämästä datasta. Taulukko 4.1 esittää yhteenvedon objektien liikenteestä. Taulukko erottelee neljä edellä mainittua sisällöntoimitussysteemiä ja edelleen erottelee datan, jonka UW:n asiakkaat pyytävät ulkoisilta palvelimilta (inbound) ja datan, jonka ulkoiset asiakkaat pyytävät UW:n palvelimilta (outbound). Huolimatta suuresta asiakkaiden määrästä Yliopiston on lähinnä datan tarjoaja kuin datan kuluttaja. HTTP dataa lähti Yliopiston palvelimilta yhteensä 16.65Tb, mutta saapui vain 3.44Tb. Vertaisverkko systeemit, etenkin Kazaa, vei suuren osan lähtevästä ja tulevasta datasta, huolimatta siitä, että Kazaa-asiakkaita oli sisäisessä ja ulkoisessa verkossa verrattain vähän. Tämä voidaan osittain lukea objektien koon ansioksi kun verrataan WWW ja vertaisverkko systeemejä.

Kuva 4.2 esittää TCP liikenteen kokonaismäärän molempiin suuntiin. Pienin kaistankuluttaja on Akamai, joka kuluttaa vain 0.2 % seuratusta TCP datasta. Gnutella kuluttaa 6.04 % ja WWW-liikenne on seuraavaksi suurin kuluttaen 14.3 % TCP liikenteestä. Kazaa on kaikista suurin kuluttaja, kuluttaen 36.9 %. Nämä neljä sisällönvälityssysteemiä kuluttavat yhteensä 57 % TCP liikenteestä, jättäen 43 % muille TCP perustaisille ohjelmille, mm. streamattu media ja sähköposti.

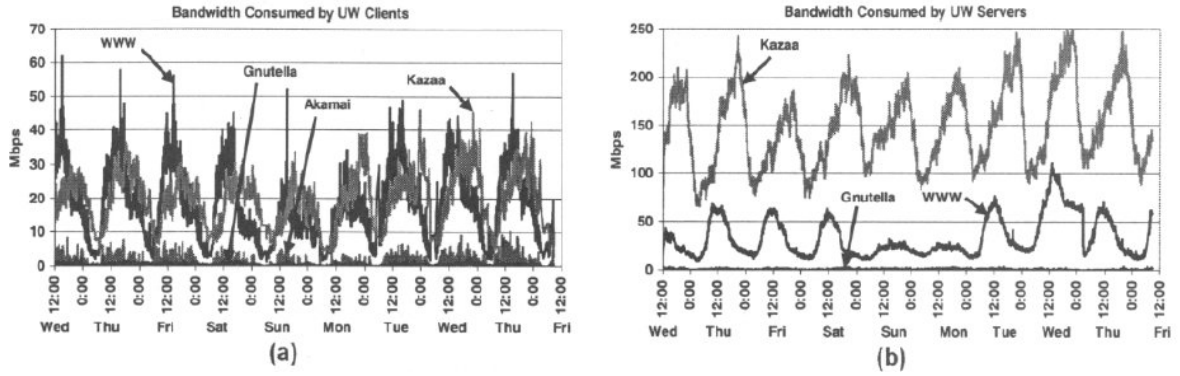
	WWW		Akamai		Kazaa		Gnutella	
	inbound	outbound	inbound	outbound	inbound	outbound	inbound	outbound
HTTP transactions	329,072,253	73,001,891	33,486,508	N/A	11,140,861	19,190,902	1,576,048	1,321,999
unique objects	72,818,997	3,412,647	1,558,852	N/A	111,437	166,442	5,274	2,092
clients	39,285	1,231,308	34,801	N/A	4,644	611,005	2,151	25,336
servers	403,087	9,821	350	N/A	281,026	3,888	20,582	412
bytes transferred	1.51 TB	3.02 TB	64.79 GB	N/A	1.78 TB	13.57 TB	28.76 GB	60.38 GB
median object size	1,976 B	4,646 B	2,001 B	N/A	3.75 MB	3.67 MB	4.26 MB	4.08 MB
mean object size	24,687 B	82,385 B	12,936 B	N/A	27.78 MB	19.07 MB	19.16 MB	9.78 MB

Taulukko 4.1 HTTP jäljityksen yhteenveto: *Inbound viittaa liikenteeseen Internetistä UW:n asiakkaille ja outbound viittaa liikenteeseen UW:n palvelimilta Internet asiakkaille. Jäljityksen kesto oli 9 päivää. [21]*



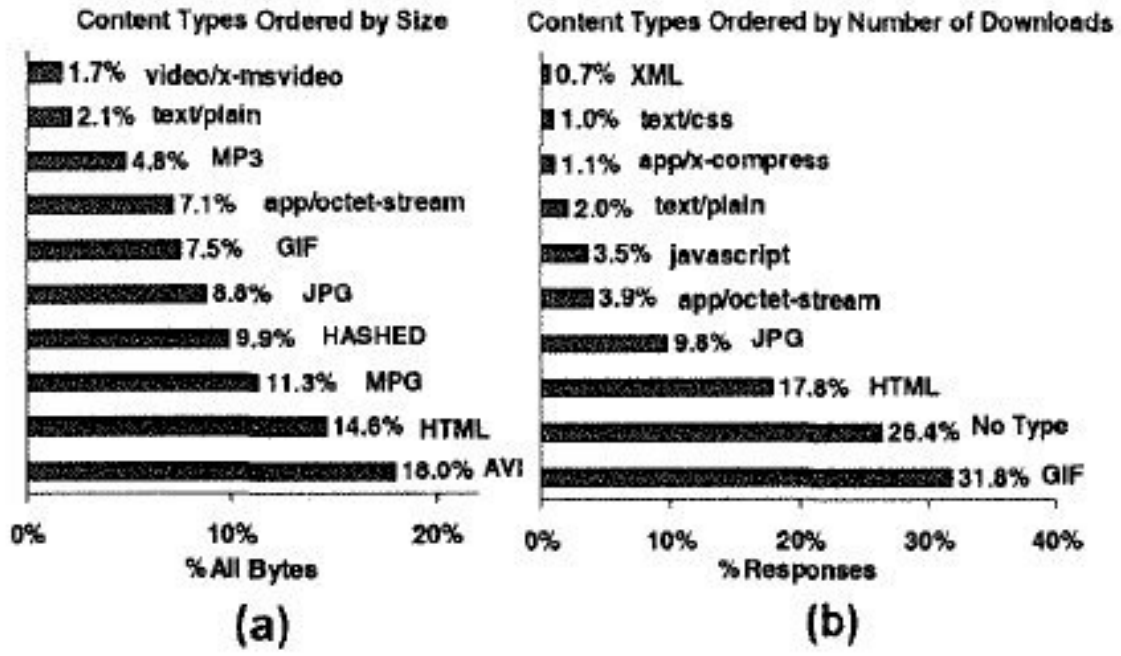
Kuva 4.2 TCP kaistankäyttö: *Kokonaismääräinen TCP kaistankäyttö HTTP liikenteenä jaoteltuina eri sisällöntoimitussysteemien perusteella. Jokainen käyrä on kumulatiivinen, joka tarkoittaa sitä, että klo 12.00 ensimmäisenä keskiviikkona, Akamai kulutti noin 10Mbps, WWW kulutti noin 100Mbps, P2P kulutti noin 200Mbps ja ei-HTTP TCP liikenne kulutti noin 300Mbps, yhteenlaskettuna yhteensä 610Mbps [21].*

Kuvassa 4.3 kohdissa a ja b näkyvät lähtevän ja tulevan liikenteen kaistankäyttö. A) kuvaajassa nähdään, että WWW- ja Kazaa-liikenteessä on tietyt syklit, mutta ne eivät ole mitenkään toisistaan riippuvaisia, koska WWW-liikenteen piikki on keskellä päivää, mutta Kazaan liikennepiikki on myöhään illalla.

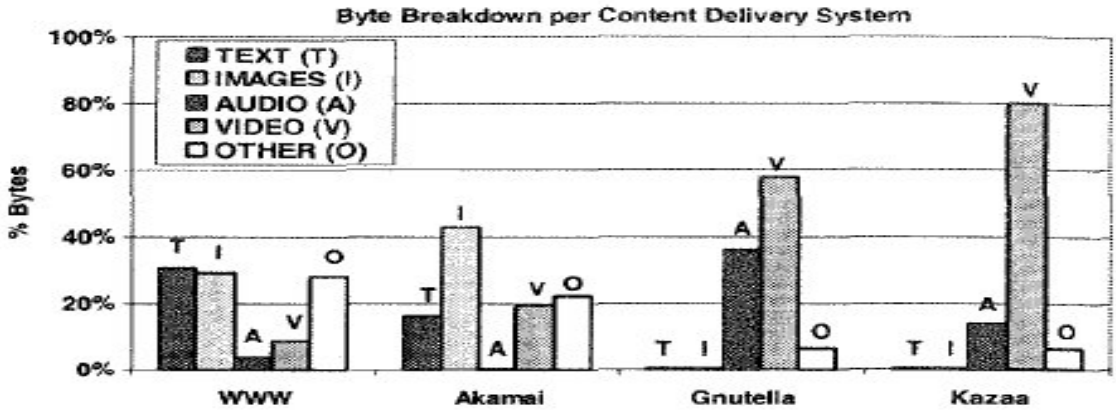


Kuva 4.3 UW:n asiakas ja palvelin TCP verkonkuormitus: *verkonkuormitus ajan suhteen (a) WWW ja P2P lataukset UW:n asiakkailta, (b) WWW ja P2P lähetykset UW:n palvelimilta [21].*

Kuvassa 4.4 nähdään kymmenen suosituinta tiedostotyyppiä, joita UW:n asiakkaat hakivat. Kohdassa (a) tiedostotyypit on jaoteltu sen mukaan miten monta tavua niitä ladattiin ja kohdassa (b) miten monta kertaa kyseistä tiedostomuotoa ladattiin. GIF ja JPEG kuvat ottavat 42% osuuden latausten kappale määrästä, mutta eivät kuitenkaan vie kuin 16.3% ladatun tiedon koosta. Toisaalta, AVI ja MPG videot, jotka vievät ladatuista tavuista 29.3%, eivät kuitenkaan vie kuin 0.41% latausten kappale määrästä. HTML on merkittävä, sillä se vie 14.6% tavuista ja 17.8% kappale määrästä. Jos näitä lukuja verrataan samojen tutkijoiden tekemään samanlaiseen tutkimukseen [23], joka tehtiin vuonna 1999 niin huomataan seuraavat muutokset liikenteessä. Tavujen suhteellisessa määrässä on tapahtunut vähenemistä HTML:n osalta 43% ja GIF/JPG:n osalta 59%. Samaan aikaan taas AVI/MPG:n osuus on kasvanut lähes 400% ja MP3:n lähes 300%. Kuvassa 4.5 nähdään sisällön tyypit jaoteltuina järjestelmien mukaan. Kuvasta nähdään, että WWW-liikenteen suurimmat sisältötyypit ovat teksti ja kuvat, Akamaissa suurin on kuvien määrä, Gnutellassa video ja audio, sekä Kazaassa huomattavin on video. Nämä korkeantason datan tunnusmerkit paljastavat huomattavat muutokset sisällöntoimitus systeemien Internet käytössä, tarkasteltuna UW:n kannalta. Ensinnäkin, HTTP-liikenne on muuttunut rajusti viime vuosien aikana, koska P2P-liikenne on siirtynyt hallitsemaan WWW-liikennettä. Toiseksi, vaikka UW on iso verkkotiedostojen julkaisija, P2P-liikenne tekee yliopistosta vieläkin isomman tiedon tarjoajan. Lopuksi, ladattavien tiedostotyyppien suhde on muuttunut. Video ja audio tiedostot vievät huomattavimman osan, toisin kuin aikaisemmin, huolimatta siitä, että niillä on vähäinen pyyntömäärä.



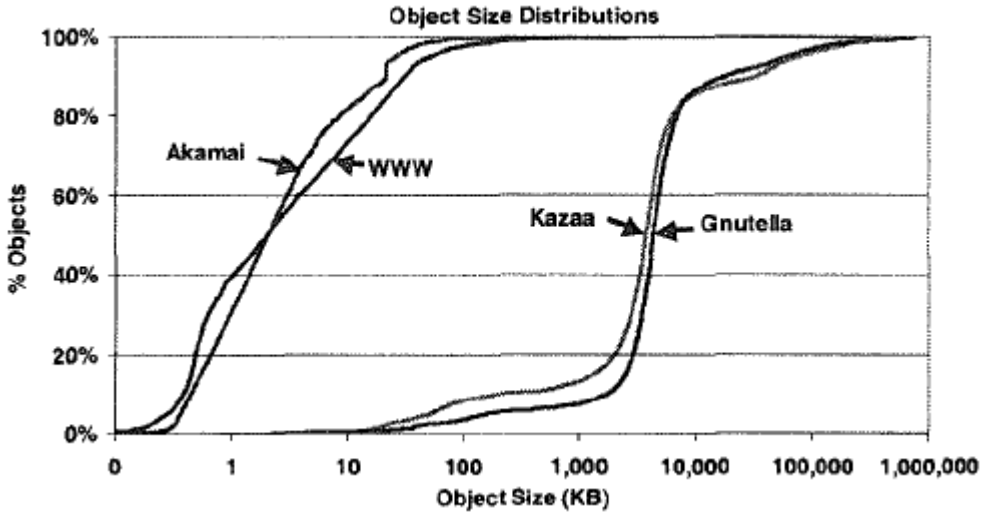
Kuva 4.4 UW:n asiakkaiden latausten sisältö tyypit: *histogrammi 10 suosituimmasta sisältötyypistä, joita UW:n asiakkaat lataavat. Järjestyksessä (a) koon mukaan (b) latausten määrän mukaan [21].*



Kuva 4.5 Ladattujen tavujen määrä objektin tyypin mukaan: *jokaisen systeemin ladattujen tavujen määrä, eriteltyinä sisällön mukaan [21].*

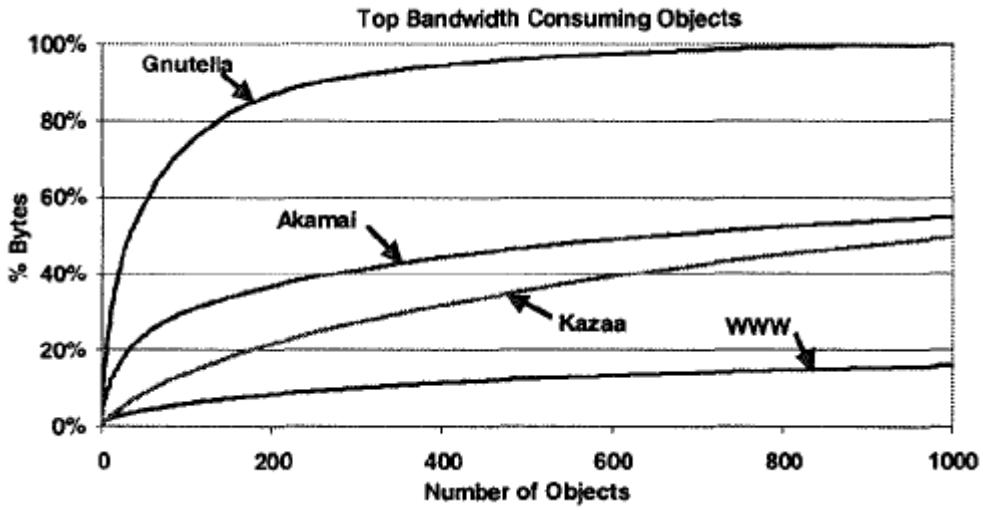
Tarkemmat sisällön tunnusmerkit

Kuten aikaisemmin on todettu, objektien koko eroaa huomattavasti P2P-liikenteen ja WWW-liikenteen osalta. Kuva 4.6 osoittaa tämän hyvin. Akamai ja WWW objektien koko on verrattain sama, mediaani objektin koko on 2Kt. Kazaan ja Gnutellan kuvaaja on huomattavasti erilainen edellisistä, mediaani objektin koko on suurin piirtein 4mt.



Kuva 4.6 Objektien koon hajonta. *(CDF cumulative distribution function) [21].*

Kuva 4.7 esittää kumulatiivisen tavujen hajonnan tuhannen eniten kaistaa vievän objektin osalta, jaoteltuina jokaisen sisällöntoimitus systeemin mukaan. Akamain käyrä kohoaa jyrkästi, 34 objektia vastaa 20% Akamain siirtämistä tavuista. Kazaan osalta nähdään, että suhteellisen pieni osa objekteista vastaa suurimmasta osasta siirrettyjä tavuja. 1000 suosituinta Kazaan objektia vastaa 50% siirretyistä tavuista. WWW-liikenteen osalta käyrä näyttää suhteellisen tasaiselta, 1000 suosituinta objektia vastaa vain 16% siirretyistä tavuista.



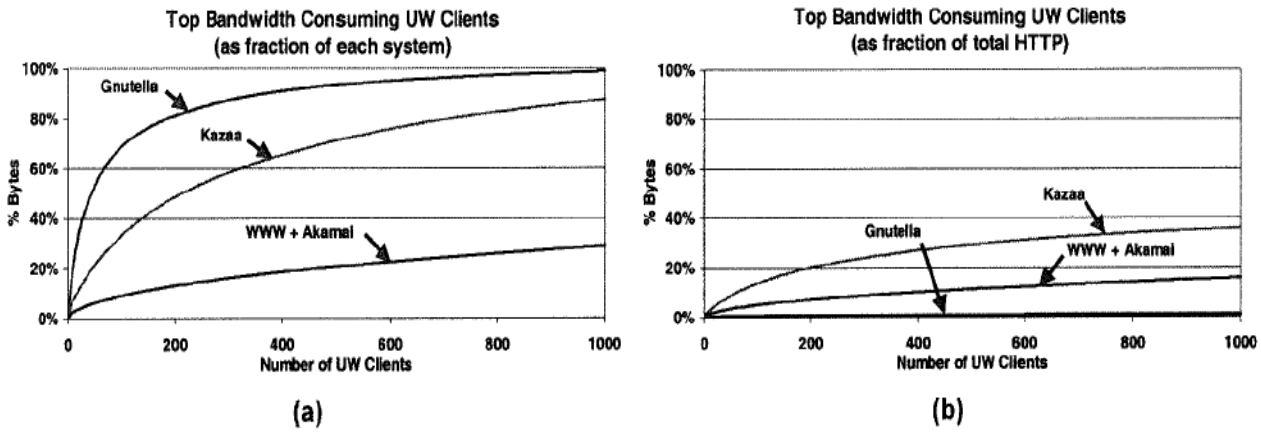
Kuva 4.7 Kaistaa kuluttavimmat objektit: *CDF tavuista, joita UW:n asiakkaat noutivat 1000 eniten kaistaa kuluttavan objektin osalta [21].*

Kuva 4.8 erittelee 10 eniten kaistaa vievää objektia WWW:n, Akamain ja Kazaan osalta. Nämä 10 objektia ovat vastuussa 1.9%, 25%, 4.9% systeemien kokonaisliikenteestä. WWW:n osalta näemme, että 10 eniten kaistaa kuluttavinta objektia ovat sekoitus erittäin suosituista pienistä objekteista (objektit 1,2 ja 4) ja suhteellisen ei-suosituista isoista objekteista (esim. objekti 3). Suurinta kaistankuluttaja objekti 1. pyydettiin useasti. Akamaissa kahdeksan kymmenestä objektista ovat suuria ja ei niin suosittuja, kaksi kolmesta pahimmasta kaistankuluttajasta ovat pieniä ja suosittuja. Kazaan sisääntuleva liikenne on lähes yhtenäistä, kaikki kymmenen eniten kaistaa kuluttavaa objektia ovat isoja (700MB) ja niitä on pyydetty kymmenestä kahteenkymmeneen kertaan. Kun verrataan Kazaan liikennettä sisään (inbound) ja ulos (outbound), huomataan useita eroja. Objektit ovat saman kokoisia, mutta UW lähettää näitä objekteja enemmän kuin ottaa vastaan. Pieni määrä UW:n asiakkaista pyytää isoja objekteja vain muutamalta ulkoiselta palvelimelta, mutta lähes kolmekymmentäkertainen määrä ulkoisia asiakkaita pyytää saman kokoisia objekteja vain kouralliselta UW:n palvelimia. Tämä johtaa yli kymmenkertaiseen kaistain kulutukseen. Huomioitavaa tässä on, että Kazaan suosituimmat tiedostot ovat kooltaan noin 700Mt kokoisia. Tämä viittaa siihen, että ne ovat pakattuja elokuvia, jotka tyypillisesti ovat juuri tuon kokoisia.

	WWW (inbound)			Akamai			Kazaa (inbound)				Kazaa (outbound)			
	object size (MB)	GB consumed	# requests	object size (MB)	GB consumed	# requests	object size (MB)	GB consumed	# clients	# servers	object size (MB)	GB consumed	# clients	# servers
1	0.009	12.29	1,412,104	22.37	4.72	218	694.39	8.14	20	164	696.92	119.01	397	1
2	0.002	6.88	3,007,720	0.07	2.37	45,399	702.17	6.44	14	91	699.28	110.56	1000	4
3	333	6.83	21	0.11	1.64	68,202	690.34	6.13	22	83	699.09	78.76	390	10
4	0.005	6.82	1,412,105	9.16	1.59	2,222	775.66	5.67	16	105	700.86	73.30	558	2
5	2.23	3.17	1,457	13.78	1.31	107	698.13	4.70	14	74	634.25	64.99	540	1
6	0.02	2.69	126,625	82.03	1.14	23	712.97	4.69	17	120	690.34	64.97	533	10
7	0.02	2.69	122,453	21.05	1.01	50	715.61	4.49	13	71	690.34	54.90	447	16
8	0.03	1.92	56,842	16.75	1.00	324	579.13	4.30	14	158	699.75	49.47	171	2
9	0.01	1.91	143,780	15.84	0.95	68	617.99	4.12	12	94	696.42	43.35	384	14
10	0.04	1.86	47,676	15.12	0.80	57	167.18	3.83	39	247	662.69	42.28	151	2

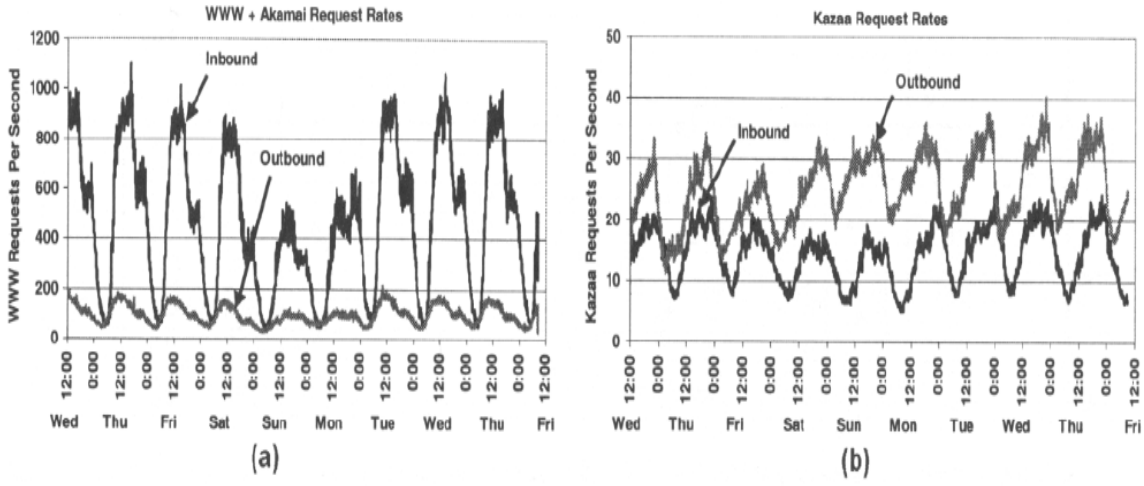
Kuva 4.8 10 eniten kaistaa kuluttavaa objektia: *Taulukossa on selvitetty objektin koko, kuinka paljon tavuja kulutettu ja pyyntöjen lukumäärä, 10 eniten kaistaa kuluttavan objektin osalta, jokaisesta systeemistä. Kazaan osalta pyyntöjen sijaan käytetään asiakkaat ja palvelimet, jotka osallistuivat objektin siirtoon [21].*

Seuraavaksi tutkimuksessa keskityttiin siihen kuka on vastuussa kaistain kulutuksesta, koska aikaisemmin käsiteltiin mikä on vastuussa kaistan kulutuksessa. Eli tutkimus keskittyi seuraavaksi asiakkaisiin. Koska WWW- ja Akamai-liikennettä ei voida erottaa toisistaan, ne yhdistettiin. Kuva 4.9 kohta (a) näyttää kumulatiivisen tavujen hajonnan, jotka on ladannut 1000 eniten kaistaa kuluttavinta UW:n asiakasta, jaoteltuina eri sisällöntoimitus systeemein. Kuvan kohdasta (a) nähdään, että pieni määrä asiakkaita on vastuussa suuresta osasta liikennettä. Kuvan kohta (b) näyttää suuremman kuvan tilanteesta, koska se tarkastelee liikennettä osana http-liikennettä. Gnutellan asiakkaat eivät vie lähes yhtään http-liikenteestä, toisaalta taas Kazaan 200 pahinta kaistankuluttajaa vievät 20% kokonais HTTP-liikenteestä ja verraten 200 WWW + Akamai pahinta kaistankuluttajaa vievät 7% kokonais HTTP-liikenteestä.



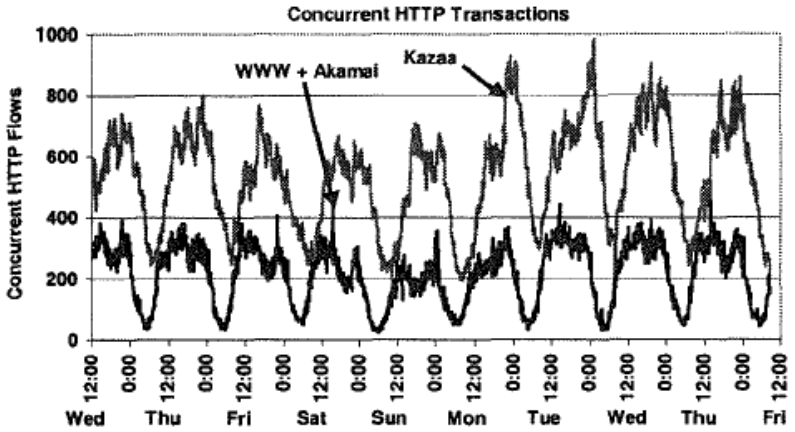
Kuva 4.9 Eniten kaistaa kuluttavat UW:n asiakkaat: *CDF ladatuista tavuista tuhannen eniten kaistaa kuluttavan UW:n asiakkaan osalta (a) osana jokaista systeemiä (b) osana koko HTTP liikennettä [21].*

Kuvassa 4.10 nähdään pyyntöjen määrät. Kohdassa (a) WWW + Akamai ja kohdassa (b) Kazaan sisään tulevien ja ulos lähtevien pyyntöjen määrä. Huomattavaa on, että kohdassa (a) y-akseli on huomattavasti isompi. Kazaan ulkoa tulevien pyyntöjen määrän huippu on 40 pyyntöä sekunnissa ja sisältä lähtevien pyyntöjen huippu 23 pyyntöä sekunnissa. WWW + Akamai sisältä lähtevien pyyntöjen huippu on 1100 pyyntöä sekunnissa ja ulkoa tulevien pyyntöjen huippu 200 pyyntöä sekunnissa (ei sisällä Akamaita). Eli Kazaalla on pienempi pyyntöjen määrä, mutta suurempi objektien koko.



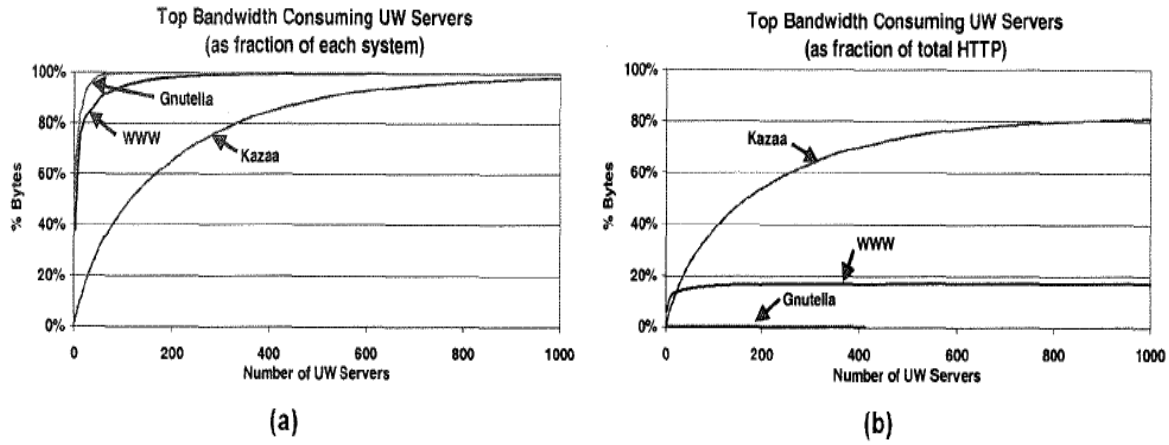
Kuva 4.10 Pyyntöjen määrät eri aikoina: *sisään ja ulos tulevien HTTP pyyntöjen määrät (a) WWW + Akamai (b) Kazaa [21].*

Yllättävin tulos järjestelmien objektien koossa ja pyyntöjen määrässä näkyy kuvassa 4.11, jossa nähdään yhdenaikaiset HTTP-toiminnot ajan suhteen. Huolimatta WWW+Akamain suuremmasta pyyntömäärästä Kazaalla on yhtäaikaisia yhteyksiä auki kaksi kertaa enemmän kuin WWW+Akamailla. Kazaa tekee vain 23 pyyntöä sekunnissa, mutta sillä on silti lähes 1000 avointa pyyntöä käynnissä, johtuen Kazaan pitkistä latausajoista. WWW pyynnön keston mediaani on 120ms kun taas Kazaan mediaani pyyntö kestää 130 sekuntia.



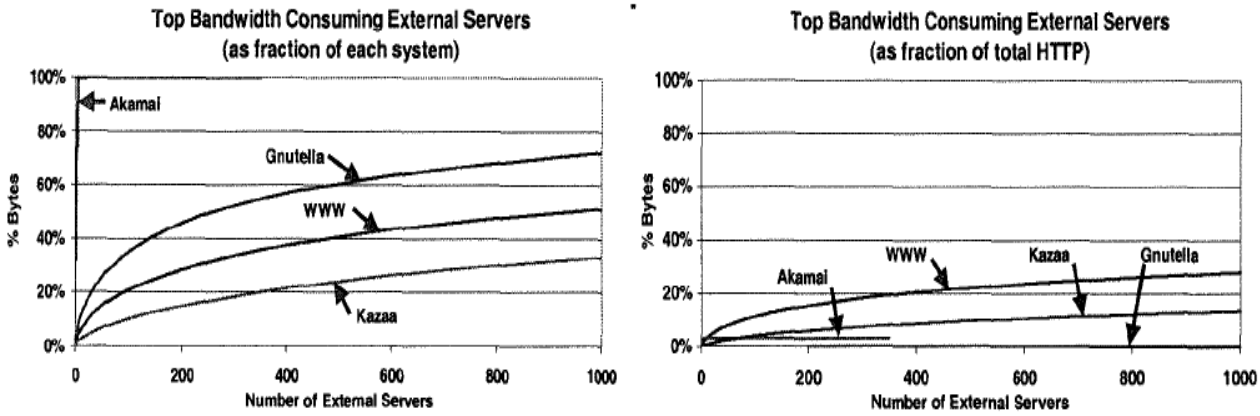
Kuva 4.11 Yhdenaikaiset HTTP toiminnot: *yhdenaikaiset HTTP toiminnot UW:n asiakkailla [21].*

Seuraavaksi tutkimus keskittyi palvelimiin eli objektien ja tavujen tarjoajiin. Kuva 4.12 näyttää CDF:n tavuista, joita UW:n sisäiset palvelimet ovat siirtäneet ulkoisille asiakkaille. Gnutellalla on vähiten sisältöä tarjoavia lähteitä ja kaikki nuo tavut on tarjonnut 10 palvelinta. WWW-käyrä on kohtuullisen jyrkkä, tässä tapauksessa kampuksella on useita isoja palvelimia, jotka tarjoavat dokumentteja Internetiin, 80% WWW-liikenteestä keskittyy 20:lle sisäiselle palvelimelle. Kazaan käyrä kohoaa paljon hillitymmin, 334 käyttäjää tarjosi 80% tavuista. Kohdassa (b) nähdään UW:n palvelimilta lähtevä data osana kokonais HTTP liikennettä. Kazaan sisäisten palvelimien vähyys verrattuna kokonaiskuvaan näkyy kuvaajasta hyvin. Kuvasta näkyy taas, että pieni määrä WWW-palvelimia tekee työn, vaikka tämä on vain pieni osa ulos lähtevää kokonaisliikennettä. WWW-palvelimet tarjoavat 20% kokonais HTTP-liikenteestä ja käyrä nousee erittäin hitaasti siitä lähtien. Kuvaajasta nähdään myös, että 170 Kazaa vertaista (peer) on vastuussa yli 50% kokonais HTTP-liikenteestä. 400 eniten lähettävää Kazaa vertaista on muodostavat yhteensä 70% kokonais HTTP-liikenteestä.



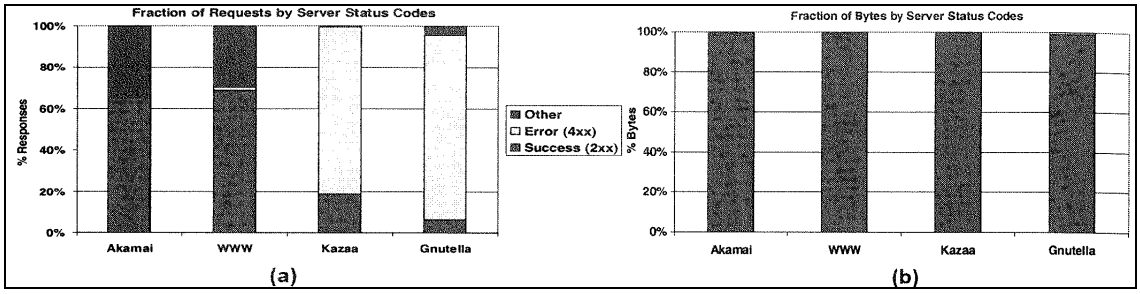
Kuva 4.12 Eniten kaistaa kuluttavat UW:n sisäiset palvelimet: : *CDF-tavuista tuhannen eniten kaistaa kuluttavan UW:n sisäisen palvelimen osalta (a) osana jokaista systeemiä (b) osana koko HTTP-liikennettä [21].*

Kuvassa 4.13 nähdään ulkoisilta palvelimilta UW:n sisäisille asiakkaille tulevan datan määrä eriteltyinä systeemein (a) ja osana kokonais HTTP-liikennettä (b) tuhannen eniten tarjoavan palvelimen osalta. Kohdassa (a) nähdään, että Kazaan käyrä nousee erittäin hitaasti, 600/281026 vertaista tarjoaa 26% datasta UW:n vertaisille. WWW nousee ensin jyrkästi, mutta tasoittuu sitten, 938/400000 ulkoista palvelinta tarjoaa 50% datasta UW:n asiakkaille. Tulos on sinänsä mielenkiintoinen, sillä WWW-asiakkaat hakevat tietoa tietyn osoitteen mukaan (URL) ja pieni määrä WWW-palvelimia on erittäin suosittuja. Vertaisverkko systeemit taas ovat erilaisia, asiakkaat hakevat dokumentteja nimen mukaan, ei palvelimien. Nämä dokumentit sijaitsevat monella eri vertaisella. Vertaisverkon tarkoitus on jakaa työ vertaisten kesken. Kazaassa isot tiedostot ladataan pieninä osina eri käyttäjiltä ja näin saavutetaan vertaisverkon suoma etu. Voisi olettaa, että palvelimien rasitus Kazaassa olisi jakautunut paljon tasaisemmin vertaisten kesken, verrattuna WWW-palvelimiin. Kuvan kohdassa (a) asia ei ole näin. Kohdassa (b) nähdään kaistan kulutus osana kokonais HTTP-liikennettä. 500 ulkoista Kazaa vertaista tarjoaa 10% kokonaismäärästä, kun taas 500 WWW-palvelinta tarjoaa dataa 22% kokonaismäärästä. Akamai ja Gnutella tarjoavat merkityksettömän osan.



Kuva 4.13 Eniten kaistaa kuluttavat UW:n ulkoiset palvelimet: : *CDF tavuista tuhannen eniten kaistaa kuluttavan UW:n ulkoisen palvelimen osalta (a) osana jokaista systeemiä (b) osana koko HTTP liikennettä [21].*

Kuvassa 4.14 nähdään vastauskoodit, jotka ulkoiset palvelimet palauttivat, jaoteltuina sisällöntoimitus systeemien mukaan. Kohdassa (a) nähdään, että WWW ja Akamai pyynnöistä onnistuivat noin 70%. Kuitenkin vain alle 20% P2P systeemien pyynnöistä olivat onnistuneita, yleisin vastauskoodi oli ”Service unavailable”. Kohdassa (b) nähdään vain, että kaikki HTTP tavut, jotka on siirretty WWW, Akamai ja P2P järjestelmissä ovat onnistuneita. Vaikka suurin osa P2P pyynnöistä ovat hylättyjä, kuitenkin se on pieni haitta verrattuna hyödyllisen tiedon lataamisen määrään.



Kuva 4.14 Palvelimien tilakoodit: *vastauskoodit, jotka ulkoiset palvelimet palauttivat; (a) pyynnöt jaoteltuina vastauskoodien mukaan. (b) tavut jaoteltuina vastauskoodin mukaan [21].*

Yhteenveto

Tutkimus päätyi seuraaviin tuloksiin: Vertaisverkkoliikenne vastasi ¾ HTTP-liikenteestä heidän tutkimusympäristössään. Kazaa vastasi yksin 36.9% TCP-liikenteestä, verrattuna WWW liikenteen 14.3%. Vertaisverkkoliikenteen mediaani objektien koko oli kolme kertaa isompia kuin WWW-objektit. Juuri objektien koko on merkittävä P2P-liikenteen suuruuden kannalta katsottuna, ei niinkään suuri käyttäjä määrä. Pieni määrä P2P-käyttäjiä vastasi isosta osasta kaistankäyttöä. 200 eniten kaistaa vievää Kazaa asiakasta vastasi 50% Kazaalla ladatuista tavuista ja lähes 27% kokonais HTTP-liikenteestä, joka tuli UW:n sisälle. Vaikka P2P pyyntöjen määrä on vähäinen niin lataukset kestävät pitkään, kolme kertaa pitempään kuin WWW-lataukset, tämä johtaa useisiin yhtäaikaisiin avonaisiin P2P-yhteyksiin. WWW pyyntöjen määrä on kaksinkertainen verrattuna P2P pyyntöihin, mutta yhtäaikaisia avonaisia yhteyksiä on P2P:ssa kaksi kertaa enemmän kuin WWW:ssa. Vaikka vertaisverkkojen suunnittelu tähtää työtaakan jakamiseen skaalautuvuuden avulla niin silti tutkimus osoittaa, että pieni määrä palvelimia vastaa suurimmasta osasta työtaakkaa. Tutkimuksessa mitattiin, että vain 600/281026 UW:n ulkopuolisista Kazaa vertaisista tarjosi 26% UW:n sisälle tulleista tavuista. Jokainen P2P asiakas luo merkittävän kaistankulutuksen molempiin suuntiin, Kazaan osalta lähetyksiä on enemmän kuin latauksia. Kazaan asiakkaan kaistan tarve on 90kertainen verrattuna WWW asiakkaaseen. Omat johtopäätökseni tutkimuksesta ovat seuraavanlaisia: Vertaisverkoissa lataajia on enemmän kuin datan tarjoajia. Vertaisverkkojen luonteeseen kuuluen, niissä liikkuu isompia tiedostoja kuin esimerkiksi Internetissä. Tämä johtuu siitä, että Internetissä ei tarjota kovinkaan paljon isoja tiedostoja. Kuten aikaisemmin mainitsin niin isot tiedostot ovat yleensä elokuvia, jotka ovat yleensä noin 700Mt kokoisia, sekä myös pelejä, jotka voivat olla vieläkin isompia. Elokuvat ja pelit, jotka ovat vertaisverkossa jaossa, loukkaavat tekijänoikeuksia ja näin ollen ovat laittomia kopioita. Vertaisverkkojen sisällön valvominen on hankalampaa kuin Internetin WWW-sivujen. Jos WWW-sivuja ei valvottaisi, niin silloin myös Internetissä olisi yleisesti jaossa isoja tiedostoja, kuten elokuvia ja pelejä (laittomia kopioita) ja näin Internet-liikenne olisi samankaltaista kuin vertaisverkkojen. Jos tiedostoa halutaan levittää nopeasti ja tehokkaasti niin se kannattaa laittaa vertaisverkkoon. Tällä hetkellä tehokkain vertaisverkko sovellus on Bittorrent. Jos kyseessä on laillinen tiedosto, niin myös Internet soveltuu sen levittämiseen, mutta tällöin palvelimen kuormitus kasvaa huomattavasti.

5 Turvallisuus

Turvallisuus on tärkeä elementti asiakas/palvelin ja vertaisverkko järjestelmissä. Seuraavaksi käydään läpi turvallisuutta yleisesti ja tämän jälkeen käydään läpi turvallisuuteen liittyvät asiat molempien järjestelmien osalta. Seuraavat yleiset turvallisuuteen liittyvät asiat koskevat molempia järjestelmiä. Tämä johdanto turvallisuuteen perustuu kirjaan *Saari, Juhani: Tietoturvallisuuden Käsikirja* [20].

Suurten tietojenkäsittelyjärjestelmien suunnittelijat ja käyttäjät ovat jo kauan tienneet turvallisuuden ja yksityisyyden tarpeellisuudesta. Niiden ihmisten lukumäärä, jotka huolehtivat näistä asioista, oli suhteellisen pieni. Tilanne on kuitenkin muuttunut tietokoneiden määrän voimakkaan lisääntymisen myötä. Tietokoneiden ja verkkojen turvallisuudesta on kerrottu paljon, mutta huomiota ovat saaneet vähemmän ne ongelmat, jotka ovat tulleet näiden mukana. Mitä useammat ihmiset käyttävät tietokoneita ja tietoverkkoja, sitä tärkeämpää on ymmärtää niihin liittyvät ongelmat ja suojaamisen tarpeellisuus.

Tietokoneiden kehittyessä myös niihin kohdistuvat rikokset kehittyvät. Tietokonerikolliset ovat kiinnittäneet huomionsa rahanarvoisen tiedon ja elektronisen rahan käsittelytekniikkaan. Tietokoneisiin liittyvästä rikollisuudesta on käytetty erilaisia nimityksiä: tietokoneväärinkäytös, tietokonepetos, tietokonepohjainen rikos, tietokoneavusteinen rikos, tietokonerikos. Donn B. Parker SRI Internationalista Kaliforniasta esitti 1970-luvulla määritelmät, joita on lainattu kirjallisuudessa siitä lähtien.

Tietokoneväärinkäytös: ”Mikä tahansa tarkoituksellinen teko, johon liityy tietokone ja jossa yksi tai useampi rikollinen hyötyy ja jossa yksi tai useampi uhri kärsii tai voisi kärsiä vahingon.”[20]

Tietokonerikos: ”Yleinen sanonta, jota käytetään määrittelemään laitonta tietokoneen käyttöä; kuitenkin se merkitsee tietokoneen suoranaista mukanaoloa rikosta suorittaessa.”[20]

Tietokonepohjainen rikos: ”Laajempi ilmaus mille tahansa laittomalle teolle, jossa tietokoneteknologian tunteminen on olennaista menestyksellisen kanteen loppuunsaattamiseksi.”[20]

Erilaisia määrittelyjä on kuitenkin paljon, lähes yhtä paljon kuin määrittelijöitä. Yleisimmäksi on noussut tietokonerikos, koska se on yksinkertaisin ja kattavin.

Tietokonerikostyypit voidaan jakaa seitsemään eri luokkaan eli kuuteen tavallisimpaan tietokonerikostyyppiin ja lisäksi muihin rikostyyppeihin.

1. *Aineelliseen omaisuuteen kohdistuva kavallus, petos ja varkaus*: Yleisin tekotapa kavalluksissa, petoksissa ja varkauksissa on ns. ”data diddling”, eli syöttötietojen muuttaminen ennen syöttöä tai syötön yhteydessä tietokoneelle. Esimerkkejä ovat dokumenttien muuttaminen tai väärentäminen, tallennusmedioitten vaihtaminen ja perustietojen virheellinen syöttäminen. Varkaus voi sisältää kaiken materiaalisen omaisuuden, mukaan luettuna tallennusmediat. Tallennusmediat ovat houkuttelevia kohteita, koska ne voivat sisältää henkilötietoja, taloudellisia tietoja tai liikesalaisuuksia. Eräs tavallisimmista tietokonerikoksista on petos. Tekaistuja henkilötietoja, sopimuksia, tilejä ja yrityksiä voidaan usein sisällyttää helpommin tietojenkäsittelyjärjestelmään kuin manuaalijärjestelmään, koska tietoja tietojenkäsittelyjärjestelmissä ei valvota jatkuvasti tietojen syöttämisen jälkeen ja koska ohjelmissa olevat kontrollit ovat usein puutteellisia.
2. *Sabotaasi ja vandalismi*: Tietokoneet, oheislaitteet ja tiedot voidaan fyysisesti hävittää tai tuhota polttamalla, räjäyttämällä tai muilla keinoilla. Historia tuntee useita tällaisia tapauksia, mm. 1960- ja 1970-luvuilla Yhdysvalloissa tapahtui lukuisia sabotaasi- ja vandalismitekoja, joissa hyökkäyksien kohteina olivat tietokonekeskukset. Tällaiset vakavat rikokset aiheuttavat suurta vahinkoa yrityksille ja yhteiskunnalle.
3. *Tietojen automaattinen hävittäminen*: Tallennusmedioille tallennetut tiedot voidaan poistaa tai muuttaa tai niihin voidaan vaikuttaa muilla tavoin. ”Looginen pommi” on ohjelma, joka määrittelee tietyn ajankohdan tai olosuhteen, milloin ko. rikollinen teko toteutetaan tietokonejärjestelmässä. Looginen pommi voidaan ohjelmoida esimerkiksi käyttämällä ns. ”Troijan hevonen” -menetelmää, jolloin ohjelman käskyillä on salainen sijoituspaikka tietokonejärjestelmässä ja näin

tietokone suorittaa asianmukaisesti ohjelmoidut toiminnot, mutta tiettyjen edellytysten puitteissa laittomiksi muuttuvat toiminnot.

4. *Tietojen varastaminen*: Tietokonejärjestelmään varastoitua tietoa voidaan varastaa ilman, että tietoa otetaan haltuun fyysisesti. Tietokoneiden välisen tai tietokoneiden ja päätteiden välisen tietoliikenteen kautta rikolliset voivat päästä käsiksi informaatioon ja näin edelleen lähettää sen omiin päätteisiinsä sekä varastaa informaatiota lukemalla, kirjoittamalla muistiin ja tulostamalla päätteillään. Eräs menetelmä informaation saamiseksi on toisena henkilönä esiintyminen eli ”impersonointi”, tässä tapauksessa kytkeydytään tietokonejärjestelmään teeskentelemällä laillista tietojärjestelmän käyttäjää. Tällaisen rikoksen suorittamiseen tarvitaan jollakin tapaa hankittu käyttäjätunnus ja salasana. Tärkeää puuttuvaa tietoa voi saada myös puhelimitse, esiintymällä toisena henkilönä. Tätä sanotaan ”spoofing” menetelmäksi. Myös eräs mielenkiintoinen menetelmä on elektroninen ”siivellä olo” eli ns. ”piggy-packing”. Tässä menetelmässä piilossa oleva laite on kytketty samaan linjaan kuin laillisen käyttäjän pääte. Piilotettua päätettä käytetään silloin, kun laillinen pääte ei ole käytössä.
5. *Tietokonepalvelujen varastaminen*: Tietokonepalveluja tai tietokoneaikaa voi käyttää käyttäjä, joka käyttää palveluja hyväkseen yli valtuuksiensa tai ohi sopimuksen. Viimeksi mainitut tapaukset voivat aiheuttaa suuria vahinkoja. Joissakin tapauksissa rikolliset ovat käyttäneet vuosien mittaan tietokoneaikaa jopa satojentuhansien eurojen edestä.
6. *Tietojen korjaaminen ja muuttaminen*: Sen lisäksi, että tietoja voidaan tuhota, niitä voidaan myös korjata ja muuttaa käytettäväksi laittomissa tapahtumissa. Jotkut tapaukset ovat paljastaneet, että opiskelijat ovat päätteiltään kyenneet muuttamaan omiaan ja toisten arvosanoja.
7. *Muita rikostyyppejä*: ”Superzapping” on IBM-koneissa käytetty macro-apuohjelma. Tietokonekeskukset, joiden käyttöympäristö on turvallinen, tarvitsevat ”hätätilan sattuessa, särje lasi”-ohjelman, jolla voi ohittaa kaikki kontrollit. Tämä ominaisuus tekee systeemin haavoittuvaksi, ellei valvonta ole riittävän tehokasta. ”Scavenging” tapahtuu esimerkiksi roskakoreja kaivelemalla, roskakoreista saattaa löytyä tärkeää tietoa rikollista toimintaa harjoitettaessa. ”Wiretapping” on luvatonta tiedon sieppaamista tietoliikennelinjalta. Tämä tapahtuu siten, että kytkeydytään linjalle itse. ”Trap door” on tavallisesti mekanismi, joka on jollakin tapaa saatu liitettyä esimerkiksi käyttöjärjestelmään. Ohjelmoijat saattavat sijoittaa järjestelmään

erilaisia virheen havainnointi- tai poistamisapukeinoja, jotka aikaansaavat katkoja ohjelmien käskyihin. Näin ohjelmiin voidaan sijoittaa lisäkäskyjä, joilla käyttöjärjestelmää voidaan hämätä. ”Asynchronous Attack” perustuu käyttöjärjestelmän epätahtiseen toimintaan. Useimmat tietokoneiden käyttöjärjestelmät toimivat epätahdissa tarjotessaan järjestelmäpalveluja samanaikaisesti ajettaville ohjelmille. Käyttöjärjestelmä varastoi ohjelmien pyyntöjä ja suorittaa ne katsomallaan ajalla. Sen sijaan, että järjestelmä suorittaisi pyynnöt siinä järjestyksessä kuin ne on vastaanotettu, järjestelmä suorittaa ne epätahdissa, käytettävissä olevien järjestelmäresurssien mukaan. Menetelmä vaatii käyttöjärjestelmien hyvää tuntemista. Tämänkaltaisen hyökkäyksen monimutkaisuudella on myös ehkäisevä vaikutus. ”Data Leakage” on tietojen tai kopioiden ottamista fyysisesti järjestelmästä. Arkaluontoista tietoa voidaan piilottaa vaarattomilta näyttäviin tulosteisiin. Varastettava tieto voi olla naamioituna eripituisissa tulosteen riveissä, sanojen tai numeroiden määrässä riviä kohti, välimerkkien paikoissa, jne. Tämä keino on monimutkainen eikä ole kovin käyttökelpoinen, koska on olemassa helpompiakin keinoja vastaavan tiedon hankkimiseksi. On kuitenkin aina muistettava se tosiseikka, että rikosten tekotapoja tulee tarkastella tekijän näkökohdista.

5.1 Asiakas/palvelin arkkitehtuuri

Asiakas/palvelin systeemeissä turvallisuus on tärkeässä asemassa, koska asiakas/palvelin systeemit ovat luonteeltaan modulaarisia. Tämä tarkoittaa sitä, että turvallisuudesta joudutaan huolehtimaan useammassa kuin yhdessä paikassa. Tämä luku perustuu kirjaan, *Bochenski, B., Implementing production quality client/server system* [4].

Pöytätietokoneen eli ns. asiakaskoneen turvallisuuteen tulee kiinnittää huomiota myös asiakas/palvelin systeemeissä. Jokaisella käyttäjällä, joka käyttää asiakaskonetta tulee olla henkilökohtainen tunnus ja salasana, jonka avulla käyttäjä tunnistetaan ja näin voidaan seurata käyttäjän toimintaa palvelimella. Yleensä asiakaskoneella ei säilytetä arkaluontoista dataa jos siinä ei ole minkäänlaisia turvallisuustoimenpiteitä, mutta jos

asiakaskoneella joudutaan säilyttämään arkaluontoista dataa, siinä tulee olla turvallisuus toimenpiteet.

Sovellusturvallisuudella voidaan viitata asiakas/palvelin systeemeissä asiakkaanpuoleiseen osa ohjelmaan tai kokonaiseen sovellukseen, joka on kehitetty asiakas/palvelin systeemiin. Sovellus turvallisuudella voi olla monia tarkoituksia. Sillä voidaan viitata esimerkiksi, tekstinkäsittelyohjelmiin, taulukkolaskentaohjelmiin ja sähköpostiin. Turvallisuuden taso edellä mainituissa sovelluksissa vaihtelee järjestelmän, kyseessä olevan datan ja sovelluksen käyttötarkoituksen mukaan. Tekstinkäsittelyohjelmat tarvitsevat harvoin turvallisuus toimenpiteitä, mutta toisaalta sähköpostiohjelmat tarvitsee turvallisuus toimenpiteitä lähes aina.

Verkon turvallisuuteen liittyy aikaisemmin esiin tulleet asiat. Asiakas/palvelin systeemeissä tieto sijaitsee yleensä palvelimella, johon asiakas sovellus käyttää verkon kautta. Näin ollen turvallisuus toimenpiteet, jotka pätevät dataan yleisesti, mutta eritoten palvelimilla sijaitsevaan dataan, pätevät myös verkotettuun dataan. Datan turvaaminen tietoliikenteessä on myös tärkeää. Liikkuva data voidaan turvata salakirjoittamalla se.

Fyysinen turvallisuus: Fyysinen turvallisuus on perusasia puhuttaessa turvallisuudesta. Se on myös kaikkein yksinkertaisin asia. Palvelin ja verkkolaitteisto voidaan turvata sijoittamalla ne suljettuun lukolliseen huoneeseen. Tällaisessa huoneessa voidaan turvallisuutta lisätä sijoittamalla laitteisto lukollisiin kaappeihin. Näin pyritään estämään mahdolliset laitevarkaudet ja asiattomien henkilöiden laitteisiin käsiksi pääsy.

Salasanat: Salasanat ovat tärkeimpiä asioita puhuttaessa asiakas/palvelin systeemeiden turvallisuudesta. Samat yleiset säännöt salasanoihin liittyen pätevät myös asiakas/palvelin systeemeissä. Esimerkiksi käyttäjän ei tulisi valita salasanakseen jotain itsestään selvää sanaa, kuten omaa nimeä, puolison nimeä, lapsensa nimeä, tai omaa syntymäpäiväänsä. Yleensä henkilö joka yrittää murtautua tietokoneelle tai tietojärjestelmään, yrittää arvata juuri tällaisia yksinkertaisia ja helppoja salasanoja. Tällainen henkilö on yleensä töissä samassa yrityksessä ja tuntee kohdehenkilön. Toisaalta salasanat saattavat selvitä jopa työpöydällä olevista valokuvista, joka tuo esiin uhan. Käyttäjän ei tulisi ikinä kirjoittaa salasanojaan mihinkään, esimerkiksi paperille. Tämä on yleinen ongelma ja siitä hyötyvät henkilöt, jotka osaavat etsiä salasanoja työpöydältä. Yleensä henkilöt jotka vastaavat

turvallisuudesta, kehittävät jonkinlaisen järjestelmän, joka vaatii salasanan vaihtoa tietyin väliajoin. Tämä aika saattaa vaihdella suuresti, tunneista vuosiin.

Salasanoihin liittyy myös muut tunnistamismetodit (authentication). Tällaisia metodeja voivat olla esimerkiksi erilaiset fyysiset tunnistuskortit (secure cards), silmän tunnistus ja sormenjälki tunnistus [5].

Varmuuskopiot ja tiedon palauttaminen: Varmuuskopiointi ja tiedon palauttaminen ovat olennaisia komponentteja jokaisessa tietokonejärjestelmässä. Varmuuskopiot voivat palauttaa kadonneen tiedon, joka on voinut kadota esimerkiksi tunkeutujan, ohjelmisto tai laitevian, käyttäjän virheen tai jonkin fyysisen tapahtuman, kuten tulipalon takia. Varmuuskopiointi ja tiedon palauttamisproseduureista löytyy runsaasti kirjallisuutta. Monet näistä proseduureista soveltuvat myös asiakas/palvelin systeemeihin. On olemassa myös paljon valmiita ohjelmistoja, joilla voidaan hoitaa varmuuskopiointi ja tiedon palautus. Jotkut ohjelmistot ovat yhteensopivia vain tiettyjen laitteistojen ja ohjelmien kanssa, toiset ohjelmistot taas soveltuvat yleiseen käyttöön.

Systeemin dokumentoinnin tulisi kertoa kuinka ja milloin varmuuskopiot tehdään. Varmuuskopiointiin ja tiedon palauttamiseen tulisi määrätä tietty henkilö tai henkilöstö. Henkilöstö tulee tarpeen vaatiessa kouluttaa tehtäviinsä. Jos useampi henkilö on vastuussa varmuuskopioista, tulee kehittää jonkinlainen järjestelmä, joka huolehtii siitä, että varmuuskopiot todella tehdään ajallaan. Jos varmuuskopioinnista huolehtii ohjelmisto, tulee sen toimivuus tarkistaa säännöllisesti. Jos varmuuskopiot sisältävät arkaluontoista dataa, voidaan ne salakirjoittaa (encrypt). Varmuuskopiot tulisi säilyttää jossakin turvallisessa paikassa, mielellään jossain toisessa paikassa, esimerkiksi poissa yrityksen tiloista.

5.2 Vertaisverkko

Vertaisverkkoteknologioita täytyy pitää turvallisuuden kannalta ”vapaamielisinä” siinä mielessä, että ne tarjoavat periytyvästi suuren määrän yksilöllistä vapautta ja ne ovat suunniteltu tarjoamaan helposti suoran yhteyden käyttäjien välille. Lyhyesti, vertaisverkkosovelluksilla on taipumus jättää, joko aktiivisesti tai tahattomasti huomiotta

yleiset turvallisuus toimenpiteet, kuten palomuurit ja filtterit. Näin ollen tieto voi kulkea asetettujen rajojen läpi ilman lupaa tai apua tai niin että kukaan ei tiedä sitä, että tieto liikkuu. Tämä luku perustuu kirjaan, *Leuf, B*: *Peer to Peer: Collaboration and Sharing over the Internet* [16].

Ohjelmien käyttömukavuus: Eräs käyttömukavuuteen liittyvä ominaisuus jonka kanssa tulee olla varovainen, on automaattinen kiintolevyjen sisällön läpikäyminen, joka etsii jaettavia tiedostoja. Tämä ominaisuus voi myös sisältää rekursiivisen hakemistojen läpikäyntiominaisuuden. Näin ollen on helppo vahingossa jakaa mahdollisesti salaisia tai muuten henkilökohtaisia tiedostoja tahtomattaan. Ratkaisu tämänkaltaisiin ongelmiin on valinnainen jaettavien tiedostojen etsiminen, selkeät asetukset, selkeät ilmoitukset ja mahdollisuus määrittää ”turvallisia” tiedostotyyppejä ja jättää pois tietynlaisia tiedostokategorioita. Edellä mainitut ovat haluttuja ominaisuuksia vertaisverkkosovelluksissa.

Joissakin vertaisverkkosovelluksissa oleva käyttömukavuuteen liittyvä ominaisuus on automaattinen vertaisverkkosovelluksen päivittäminen. Tällainen ominaisuus tarkoittaa sitä, että vertaisverkkosovellus voi päättää itse tarvitseeko se päivitystä. Eli jos vertaisverkkosovelluksesta on saatavilla uusia versioita tai siihen on saatavilla uusia komponentteja, niin sovellus voi käynnistää käyttäjälle yleensä näkymättömän tiedoston nouto ja asennus prosessin. Tällainen ominaisuus muodostaa potentiaalisen turvallisuus riskin. Valvomaton vertaisverkkosovellus voi jatkuvasti ”kehittyä” ilman, että käyttäjä tietää siitä. Tämä ominaisuus on kieltämättä arvokas jos vertaisverkkosovellus kehittyy nopeasti, koska tämä ominaisuus takaa sen, että jokaisella käyttäjällä on nopeasti uusin versio ohjelmasta. Yleensä tällainen päivitys prosessi käynnistää sovelluksen uudestaan, jotta uusi versio voi asentaa itsensä, tämä häiritsee vertaisverkkosovelluksen yhteyttä vertaisverkkoon. Monissa tapauksissa se merkitsee sitä, että käyttäjä näkee täysin uuden verkkotopologian kun sovellus käynnisty uudestaan ja saa yhteyden vertaisverkkoon. Vertaisverkkosovelluksen automaattiseen päivittämiseen liittyvä riski on järjestelmän epävakaus, joka yleensä johtuu uuden version epävakaudesta, tämä koskee erityisesti sellaisia järjestelmiä, joita ei valvota jatkuvasti.

Merkille pantavaa on, että tiedostojen läpikäynti ja automaattinen sovelluksen päivitys ovat yleensä oletusarvoisesti päällä. Asiaan liittyvät asetukset ovat yleensä hankalia löytää,

kuten esimerkiksi se kuinka usein sovellus tarkastaa onko uusia versioita saatavilla. Yleensä molemmat asetukset ovat kuitenkin käyttäjän muokattavissa.

Eräs vakava ongelma on, että vertaisverkkosovelluksen päivitys voi kaatua siten, että vertaisverkkosovellus huijataan hakemaan ja suorittamaan vahingollinen ohjelma sen sijaan, että vertaisverkkosovellus päivittyisi normaalisti. Vahingollinen ohjelma voi tehdä erilaisia tuhoisia aktiviteetteja koneelle ja koneelta, jolle se on saanut asennettua itsensä. Tällainen ohjelma on yleensä ohittanut virustutkat ja palomuurin, koska se on yleensä vertaisverkkosovelluksen lapsiprosessi.

Kommunikaatio: Kommunikaatiossa, yleensä ottaen vertaisverkkosovelluksien välisessä viestinnässä, huomio on käyttäjän identiteetissä. Seuraavaksi käydään läpi muutamia kommunikaatioon liittyviä asioita, ongelmia ja mahdollisia ratkaisuja näihin ongelmiin.

Vertaisverkkosovelluksen käyttäjän identiteetin todentamisen ongelmana on, että monet vertaisverkkosovellukset eivät välitä kuka käyttäjä on tai olettavat, että käyttäjä on kuka väittääkin olevansa. Mahdollinen ratkaisu kyseiseen ongelmaan on esimerkiksi ulkoinen mekanismi, joka voi olla esimerkiksi palveluun kirjautuminen. Myös oikeanlaisen sovelluksen valinta auttaa tähän ongelmaan.

Läsnäolon varmentamisen ongelmana on, että käyttäjä ei välttämättä aina ole läsnä, vaikka vertaisverkkosovellus olisi päällä. Tietyt vertaisverkkosovellukset vaativat käyttäjän läsnäoloa toimiakseen, koska tietyt sovellukset vaativat tiedonsiirto varmennuksen käyttäjältä. Myös kytkeytyneen sovelluksen tunnistaminen on ongelma. Ratkaisuna on parempi käyttäjien valitseminen vertaisverkkoon ja automaattinen aktiivisuuden tunnistaminen sovelluksessa. Myös käyttäjien loukkaamattomuus on ongelma. Loukkaamattomuudella tarkoitetaan sitä, että käyttäjä on turvassa erimuotoisilta häirintä yrityksiltä. Loukkaamattomuuteen ratkaisuna on vertaisverkkosovelluksessa käyttäjälle mahdollinen näkymättömyyden päälle valinta. Tämä tarkoittaa sitä, että halutessaan käyttäjä voi olla muille käyttäjille näkymätön. Tällä estetään joskus ilmenevää vertaisverkkosovellusten käyttäjiin kohdistuvaa erimuotoista häirintää ja näin se voidaan estää.

Joissakin vertaisverkkosovelluksissa on käyttäjälista. Tämä on lista, josta näkee esimerkiksi henkilökohtaiset vertaisverkon käyttäjät, joiden kanssa yleensä toimitaan. Ongelmana on käyttäjälistan loukkaamattomuus. Ratkaisuna on mahdollisuus varmistaa käyttäjälistan yksityisyys ja muiden käyttäjälistojen pois sulkeminen.

Identiteetin varmistaminen vertaisverkoissa voidaan jakaa seuraaviin lajeihin, jotka ovat nousevassa järjestyksessä turvallisuuteen nähden:

1. Varmistamista ei ole, systeemi hyväksyy käyttäjän automaattisesti
2. Implisiittinen, sovelluksessa on mahdollisuus suorittaa identiteetin varmistava osa
3. Paikalliseen koneeseen, sovellukseen tai paikallisverkkoon perustuva sisäänkirjautuminen eli paikallinen salasana on pakollinen
4. Vertaisverkkopalvelimeen/palveluun sisään kirjautuminen, vertaisverkko tunnistaa käyttäjän
5. Ulkoinen turvallinen identiteetin varmennus, jonka välittäjänä on keskitetty vertaisverkkopalvelu eli jokin palvelin, jossa on salattu allekirjoitus

Tiedon jakamisessa mielenkiinto ei ole niinkään identiteetissä vaan päämielenkiinto on yleensä staattisessa sisällössä, jota voidaan ladata vertaisverkon avulla. Tämä ei tarkoita sitä, että kaikki vertaisverkot pitäisivät identiteetin tunnistamista epäolennaisena, koska se määräytyy ulkoisista tekijöistä ja vertaisverkon laadusta.

Henkilökohtainen identiteetti tai henkilöiden tunnistaminen on tärkeää jos vertaisverkkoyhteisö on jollakin tavalla suljettu tai jos sisällön jakaminen tarvitsee henkilöltä henkilölle nimenomaan tarkoitetun hyväksynnän. Muutoin, mielenkiinto on asiakaan/palvelimen rooleilla ja mahdollisella luottamussysteemin kehittämisellä, joka perustuu verkonsolmun identiteettiin ja kuinka se käsittelee epäluotettavia, häiritseviä ja vihamielisiä verkonkäyttäjiä.

Palomuurit ja tunnelit: Suurin osa vertaisverkkosovelluksista ja protokollista ovat kehittyneet toimimaan palomuurien takana, joten olemassa oleva palomuuri ei välttämättä estä virtuaalisen vertaisverkon toimintaa. On yleensä helppoa asentaa vertaisverkkosovellus toimimaan ”turvallisen” alueen sisälle, esimerkiksi palomuurin sisäpuolelle. Näin ollen vertaisverkkosovellus yhdistyy helposti vertaisverkkoon, joka toimii yleisessä Internetissä. Käyttäjä ei yleensä tajua, että hänen asentamansa sovellus

tarjoaa tunnelin (tunnel) ”turvalliselle” alueelle ja näin ollen muodostaa vakavan uhan palomuurin toimintaperiaatteelle.
Koska vertaisverkoilla ei ole mitään yhtenäistä toimintaperiaatetta, arkaluontoinen sisältö, jota on pidetty turvallisessa paikassa palomuurin takana voi tahtomattaan tulla helposti ulkopuolisten saataville. Useassa tapauksessa turvallisuusmurto tapahtuu kumpaankin suuntaan, johtuen tästä tunneliefektistä.

Sisällön pysyvyys: Oletamme yleensä, että varastoitu sisältö pysyy saatavilla ikuisesti, huolimatta kokemuksista, jolloin henkilökohtainen tietokone tai yrityksen palvelin kaatuu. Varmuuskopiot voivat yleensä palauttaa hävinneen tiedon. Vertaisverkon sisällön pysyvyys riippuu käytettävästä tekniikasta. Puhtaassa vertaisverkossa, kuten Gnutella:ssa [12] on satunnainen sisällön kopiointi, joka kattaa suurimman osan tiedon varmuuskopioinnista täydelliseen sisällön pysyvyyteen. Kriittisintä on löytää vertaisverkosta sellainen käyttäjä, jolla on vaadittava sisältö ja kapasiteettia tarjota se. Hajautettu tiedontallennus sisältää yleensä sisäänrakennetut tiedon replikointi- ja virheenkorjaus toiminnallisuudet, jotka kompensoivat normaalien varmuuskopioiden puutetta. Jotkut tallennusjärjestelmät tarjoavat automaattisen tiedon replikoinnin, tämä ei vain paranna suorituskykyä vaan myös takaa paremman tiedon pysyvyyden.

5.3 Vertailu

Tässä kohdassa verrataan asiakas/palvelin arkkitehtuurin ja vertaisverkon eroja sekä yhtäläisyyksiä turvallisuuden kannalta katsottuna. Vertailtavat asiat ovat edellä mainittuja ja niiden tärkeys ja riskin suuruus pohjautuvat edellä olevaan tekstiin. Asioille annetaan pisteitä systeemin turvallisuuden tärkeyden näkökulmasta seuraavalla tavalla.

Tärkeys, systeemin turvallisuuden kannalta: 1 = Ei tärkeä, 2 = Melko Tärkeä, 3 = Tärkeä
Pisteitä annetaan myös verrattavan asian aiheuttaman turvallisuus riskin kannalta katsottuna. Turvallisuus riskin suuruus: 1 = Ei vaikutusta, 2 = Pieni, 3 = Suuri
Lopuksi lasketaan tärkeyden ja riskin keskiarvo ja näin saadaan luku, jolla selviää ero.

Verrattava asia	*Asiakas/palvelin*		*Vertaisverkko*		
	Tärkeys	Riski	Tärkeys	Riski	Selitys
Käyttäjätunnus ja salasana	3	3	1	2	Käyttäjätunnus ja salasanat ovat tärkeitä lähinnä asiakas/palvelin järjestelmissä. Vertaisverkoissa käyttäjätunnusta ja salasanaa ei yleensä tarvita
Fyysinen turvallisuus	3	3	2	2	Fyysinen turvallisuus on hieman tärkeämpää asiakas/palvelin järjestelmissä
Varmuuskopiointi	3	3	1	1	Varmuuskopiointi on avainasemassa asiakas/palvelin järjestelmissä
Tiedonsalakirjoittaminen	2	2	1	1	Tiedonsalakirjoittamisella saavutetaan tiettyjä etuja, mutta sen käyttö on vapaavalintaista
Ohjelmien oletus asetukset	1	1	3	3	Ohjelmien oletus asetusten tarkastaminen on vertaisverkkosovelluksissa tärkeää turvallisuuden kannalta
Ohjelmien automaattinen päivittäminen	1	1	3	3	Ohjelmien automaattinen päivittäminen on vertaisverkkosovelluksia koskeva turvallisuus riski
Käyttäjän identiteetin todentaminen	2	2	2	2	Käyttäjän identiteetin todentaminen, koskee molempia järjestelmiä ja on kohtalaisen tärkeää
Käyttäjän läsnäolon varmentaminen	2	2	2	2	Käyttäjän läsnäolon varmentaminen on molemmissa järjestelmissä kohtalaisen tärkeää
Kytkeytyneen sovelluksen tunnistaminen	2	2	2	2	Kytkeytyneen sovelluksen tunnistaminen on kohtalaisen tärkeää molemmissa järjestelmissä

Verrattava asia	*Asiakas/palvelin*		*Vertaisverkko*		
	Tärkeys	Riski	Tärkeys	Riski	Selitys
Palomuurin ohittaminen tiedostamatta	1	1	3	3	Palomuurin ohittaminen tiedostamatta koskee vertaisverkkosovelluksia
Tunnelin luominen tiedostamatta	1	1	3	3	Tunnelin luominen tiedostamatta koskee vertaisverkkosovelluksia
Sisällön pysyvyys	2	2	2	2	Sisällön pysyvyys on kohtalaisen tärkeää molemmissa järjestelmissä
Näkymättömyys muilta käyttäjiltä tarvittaessa	1	1	2	2	Näkymättömyys muilta käyttäjiltä koskee vertaisverkkoja
Keskiarvo	***2.15***	***1.84***	***2.08***	***2.15***	Asiakas/palvelin järjestelmissä on pienempi riskien suuruus, mutta niissä on enemmän turvallisuuden kannalta tärkeitä asioita
3 määrä	*3/13*	*3/13*	*4/13*	*4/13*	
2 määrä	***5/13***	***5/13***	***6/13***	***7/13***	
1 määrä	***5/13***	*3/13*	*3/13*	*2/13*	

Luvuista voimme päätellä, että asiakas/palvelin arkkitehtuurissa on pienempi riskien suuruus kuin vertaisverkossa. Mutta taasen vertaisverkossa on vähemmän tärkeitä asioita, jotka pitää ottaa huomioon turvallisuuden kannalta. Loppujen lopuksi järjestelmät ovat aikalailla samanlaiset turvallisuuden kannalta katsottuna, näiden lukujen perusteella.

6 Yhteenveto

Tässä luvussa pyritään yhdistämään edelliset luvut siten, että asiakas/palvelin järjestelmien ja vertaisverkkojen erot ja yhtäläisyydet tulevat ilmi.

Molempia järjestelmiä voidaan peilata Internetin kehitykseen. Ensimmäinen vertaisverkko, ARPAnet, kehitettiin 1960-luvulla, kun Yhdysvaltain puolustusministeriö teki aloitteen muutaman tutkimuslaitoksen yhdistämisestä tietoverkolla toisiinsa. Tämä täysin hajautettu vertaisverkko oli nykyisen internetin esiaste. 1980-luvun lopulla verkkoon liitettiin lisää yliopistoja ja pian myös yrityksiä ja yhteisöjä. 1990-luvun aikana Internet on mullistanut perinteisen tiedonsiirron lisäksi kaupankäynnin, poliittisen vaikuttamisen ja koko sosiaalisen elämän.

Uusien käyttäjäryhmien tarpeet ovat muokanneet verkon rakennetta. Tavallinen Internetkäyttäjä lähinnä hakee verkosta tietoa ja tiedostoja, eikä tiedon jakaminen enää ole ensisijaista. Tämän takia Internetiä on kehitetty epäsymmetriseksi verkoksi, jossa on hierarkkinen asiakas/palvelin-arkkitehtuuri.

Asiakas/palvelin-tietojärjestelmä on hajautettu niin, että osa järjestelmästä toimii tavallisilla työasemilla ja osa (esim. tietokanta) isommassa koneessa eli palvelimessa. Järjestely vähentää verkon yli kulkevan tiedon kokoa, sillä tietoa ei siirretä palvelimelta käyttäjälle tiedon käsittelyä varten. Internetissä siirtyvät vain tiedon käsittelypyynnöt ja niistä saadut tulokset. Vastaavasti palvelimen kuormitus kevenee, koska sen ei tarvitse huolehtia käyttöliittymän ylläpidosta.

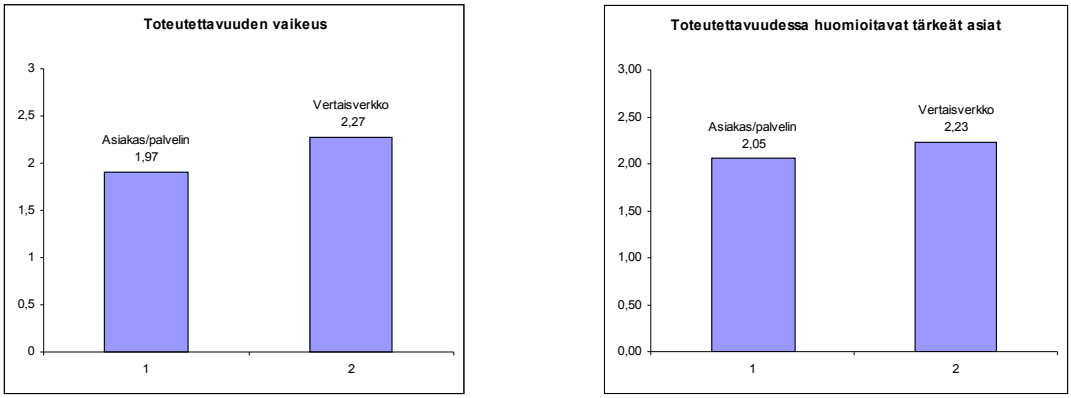
Vertaisverkossa kaikki työasemat toimivat siis myös palvelimina. Ne voivat jakaa omat levyasemansa ja oheislaitteensa muiden verkossa olevien käyttöön. Jako tapahtuu taustaajona niin, että työasemaa voidaan samaan aikaan käyttää myös muihin toimintoihin, tosin koneen toiminta saattaa hidastua. Vertaisverkoissa tietoliikenteen määrä on suurempaa kuin normaalissa asiakas/palvelin-tietojärjestelmissä. Lisäksi yhden asiakaskoneen tiedonkäsittelykapasiteetilta vaaditaan enemmän.

Verkkojen kasvaessa myös vertaisverkkojen arkkitehtuuri on muuttumassa. Uuden sukupolven vertaisverkko-ohjelmistot noudattavat jo hierarkkista mallia. Pääperiaate on

niissäkin sama kuin hajautetussa mallissa, eli kaikki työasemat voivat olla keskenään yhteydessä toisiinsa, eikä tieto kulje keskitetyn palvelimen kautta. Hierarkkisessa vertaisverkossa on kuitenkin palvelimia, jotka toimivat puhelinluettelon tapaisina indeksointivarastoina. Vastaavalla logiikalla toimii nykyinen palvelinnimijärjestelmä (DNS).

Vertaisverkot ja asiakas/palvelin järjestelmät koostuvat molemmat pääsääntöisesti kolmesta komponentista, asiakkaasta, palvelimesta ja verkosta. Vertaisverkot ovat kehittyneempi versio asiakas/palvelin järjestelmistä ja näin ollen ne sisältävät samoja ominaisuuksia ja toimintoja, kuin tavanomaiset asiakas/palvelin järjestelmät, mutta lisäävät niihin uusia ominaisuuksia. Vertaisverkoissa palvelimen roolia hoitavat asiakkaat, pois lukien erinäiset apupalvelimet, kuten havaintopalvelin. Tästä johtuen, luvussa 2 esitelty kymmenen kohdan lista asioista, jotka kuuluvat hyvään asiakas/palvelin järjestelmään, sopivat myös vertaisverkkoon.

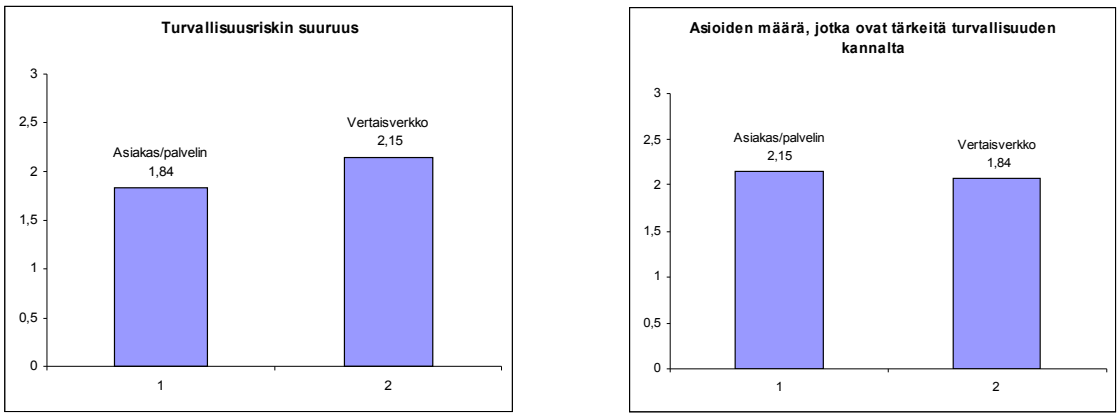
Toteutettavuuden kannalta katsottuna, molemmat järjestelmät omaavat haastavia osa-alueita. Haastavimpina osa-alueina voidaan molemmissa pitää, skaalautuvuutta, turvallisuutta, heterogeenisyyttä ja suorituskykyä. Asiakas/palvelin järjestelmissä täytyy näiden lisäksi huomioida yhdenaikaisuus, sekä pääsynhallinta. Vertaisverkoissa huomionarvoinen ominaisuus on oikeudenmukaisuus. Kuvassa 6.1 on nähtävillä asiakas/palvelin järjestelmien ja vertaisverkkojen erot toteutettavuuden vaikeuden asteessa, sekä toteutettavuudessa huomioitavien asioitten määrä. Kuvaajista nähdään, että vertaisverkko on hieman vaikeampi toteuttaa ja vertaisverkossa on myös enemmän huomioimisen arvoisia asioita toteutettavuuden kannalta. Kuvaajat on tehty luvussa 3 esitetystä taulukosta. Huomion arvoista on, että molemmat ovat kuitenkin lähestulkoon samanarvoisia toteutettavuuden kannalta katsottuna.



Kuva 6.1 *Toteutettavuuden vaikeus ja toteutettavuudessa huomioitavat tärkeät asiat.*

Verkonkuormituksen osalta voidaan todeta, että vertaisverkossa liikkuu isompia tiedostoja, johtuen vertaisverkkojen luonteesta. Eli vertaisverkoissa liikkuu paljon laitonta materiaalia, nämä tiedostot ovat kooltaan suuria. Vertaisverkon asiakas luo paljon liikennettä molempiin suuntiin, kun taas asiakas/palvelin järjestelmissä palvelimet ovat kuormitettuja. Molemmissa järjestelmissä verkonkuormitusta pyritään jakamaan replikoimalla tietoa verkon jäsenten kesken. Vertaisverkon ongelmana tiedonreplikoinnissa on niin sanottu vapaamatkustajaongelma, jossa vertaisverkon jäsen ei lähetä tietoa. Yleensä ottaen tietoliikenne kasvaa Internetissä ja myös vertaisverkoissa ja näin ollen myös laitteistojen ja ohjelmistojen on kehityttävä vastaamaan tarvetta.

Perustavanlaatuiset turvallisuuskysymykset asiakas/palvelin järjestelmissä ja vertaisverkoissa ovat samat. Nämä aiheet on esitelty luvun 5 alussa. Yhtäläisyyksiä molemmissa järjestelmissä on tärkeyden kannalta fyysinen turvallisuus, käyttäjän identiteetin todentaminen, käyttäjän läsnäolon varmentaminen, kytkeytyneen sovelluksen tunnistaminen ja sisällön pysyvyys. Asiakas/palvelin järjestelmissä näitten lisäksi tärkeää on käyttäjätunnukset ja salasanat, tiedon salakirjoittaminen, sekä varmuuskopiointi. Vertaisverkoissa tärkeää turvallisuuden kannalta on edellisten lisäksi ohjelmien oletus asetukset, ohjelmien automaattinen päivitys, näkymättömyys muilta käyttäjiltä, palomuurin ohittaminen tiedostamatta, sekä tunnelin luominen tiedostamatta. Kuvassa 6.2 on kuvaajat, jotka pyrkivät esittämään turvallisuuden kannalta tärkeiden asioiden määrän molemmissa järjestelmissä, sekä mahdollisen turvallisuusriskin suuruuden. Molemmissa järjestelmissä huomioitavia asioita on lähes tulkoon saman verran, mutta vertaisverkossa turvallisuusriskin suuruus on jonkin verran suurempi. Kuvaajat on laadittu luvussa 5 esitetystä taulukosta.



Kuva 6.2 Tärkeiden a*sioiden määrä turvallisuuden kannalta ja turvallisuusriskin suuruus.*

Tästä tutkielmasta voidaan tehdä seuraavanlaiset johtopäätökset:

- Turvallisuuden kannalta katsottuna asiakas/palvelin järjestelmä on turvallisempi kuin vertaisverkko
- Toteutettavuuden kannalta katsottuna asiakas/palvelin järjestelmä on helpompi toteuttaa kuin vertaisverkko
- Tiedostojen ja resurssien jaon tehokkuuden kannalta katsottuna vertaisverkko on ylivertainen verrattuna asiakas/palvelin järjestelmään

Viitteet

[1] Akamai. Internet WWW-sivu, URL: http://www.akamai.com (30.8.2006)

[2] Androutsellis-Theotokis, S. & al: *A Survey of Peer-to-Peer Content Distribution Technologies*. ACM Computing Surveys, Vol 36, No. 4, s. 335-371.

[3] BitTorrent. Internet WWW-sivu, URL: http://www.slyck.com/bt.php (12.10.2006)

[4] Bochenski, B.: *Implementing Production-quality Client/Server Systems*. John Wiley & Sons, Inc., Yhdysvallat, 1994.

[5] CORBA. Internet WWW-sivu, URL: http://www.corba.org/ (12.10.2006)

[6] Coulouris, G.: *Distributed Systems, Concepts and Design, Fourth Edition*. Addison-Wesley, Yhdysvallat, 2005.

[7] Dewire, D.: *Client/Server Computing*. McGraw-Hill, Inc., Yhdysvallat, 1993.

[8] Distributed.net RC5. Internet WWW-sivu http://www.distributed.net/rc5/ (12.10.2006)

[9] Dreamtech Software Team: *Peer-to-Peer Application Development: Cracking th Code*, Hungry Minds, Inc., Yhdysvallat, 2002.

[10] eDonkey2000. Internet WWW-sivu, URL: http://www.slyck.com/edonkey2k.php (12.10.2006)

[11] eMule. Internet WWW-sivu, URL: http://www.emule-project.net (12.10.2006)

[12] Gnutella. Internet WWW-sivu, URL: http://www.gnutella.com (30.8.2006)

[13] HTTP. Internet WWW-sivu, URL: http://www.w3.org/Protocols (30.8.2006)

[14] Java RMI. Internet WWW-sivu, URL: http://java.sun.com/products/jdk/rmi/ (12.10.2006)

[15] Kazaa. Internet WWW-sivu, URL: http://www.kazaa.com (30.8.2006)

[16] Leuf, B.: *Peer to Peer: Collaboration and Sharing over the Internet*. Addison Wesley, Yhdysvallat, 2002.

[17] Oram, A.: *Peer to Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly & Assosiates, Inc., 2001.

[18] Overnet. Internet WWW-sivu, URL: http://www.slyck.com (12.10.2006)

[19] Robert Frances Group: *Total Cost of Ownership for Linux Web Servers in the Enterprise.* WWW-sivu, URL: http://www.rfgonline.com/subsforum/LinuxTCO.pdf (31.10.2006)

[20] Saari, J.: *Tietoturvallisuuden Käsikirja*. Otava, 1988.

[21] Sariou, S. & al: *An Analysis of Internet Content Delivery Systems*. ACM SIGOPS Operating System Review, Volume 36 Issue SI, 2002, s. 315 – 327.

[22] Seti@home. Internet WWW-sivu, URL: http://setiathome.berkeley.edu (12.10.2006)
[23] Wolman, A. & al:. *Organizationbased analysis of web-object sharing and caching. In Proc. of the 2nd USENIX Conf. on Internet Technologies and Systems,* Oct. 1999.