

Oskari Alanko

P2p-suoratoiston keskeiset saatavuuden ongelmat

Tietotekniikan kandidaatintutkielma

23. maaliskuuta 2022

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Oskari Alanko

Yhteystiedot: `oskari.alanko@gmail.com`

Ohjaaja: Antti-Jussi Lakanen

Työn nimi: P2p-suoratoiston keskeiset saatavuuden ongelmat

Title in English: Relevant challenges of availability in p2p-streaming

Työ: Kandidaatintutkielma

Sivumäärä: 36+0

Tiivistelmä: Suoratoisto on keskeinen osa modernia mediamaisemaa. Suoratoistolähetysillä saattaa olla massiivisiakin katsojamääriä, jonka vuoksi suoratoistopalveluiden tarjoaminen vaatii huomattavasti resursseja. Näitä resurssitarpeita voidaan kuitenkin vähentää hyödyntämällä katsojien laitteiden edelleenlähetykskapasiteettia käyttämällä vertaisverkkoja (engl. *p2p-networks*). Katsojien, eli vertaisten, resurssien hyödyntämisessä on kuitenkin useita haasteita, jotka saattavat estää tai haitata sujuvan suoratoistopalvelun tarjoamista. Tämä tutkimus kartoittaa näitä p2p-suoratoiston kohtaamia haasteita katsojan palvelunlaadun näkökulmasta. Havaitaan, että suuri osa p2p-suoratoiston kohtaamista ongelmista liittyvät vertaisiin liittyvään ennalta-arvaamattomuuteen. Vertaisissa saattaa esiintyä keuhkoja verkkoyhteyksiä tai vertaisia saattaa poistua ilman varoitusta. Tästä johtuvat häiriöt yleensä esiintyvät suoratoiston katkeiluna, lähetyksen laadun heikkenemisenä ja lähetykseen liittymisen viiveenä. P2p-suoratoisto kohtaa myös flash crowd -ilmiön, jossa suuri määrä vertaisia pyrkii liittymään vertaisverkkoon nopeammin kuin vertaisverkko kykenee tehokkaasti liittämään. Hallitsematon flash crowd puolestaan johtaa liittymisien epäonnistumisiin ja pitkiin aloitusviiveisiin.

Avainsanat: peer-to-peer, p2p, suoratoisto, video, multicast, sovellustason multicast

Abstract: Live streaming is an important part of modern media landscape. Live streaming broadcasts can potentially have a massive amounts of concurrent viewers, because of this providing live streaming services takes considerable resources. However by using the forwarding capacity of viewers, by using peer-to-peer-networks, it is possible to lessen this resource requirement significantly. Using the resources of viewers, as in peers, has some challenges

and difficulties that might impede the service providers ability to deliver a smooth live streaming service. This paper surveys these challenges while focusing on problems that diminish the quality of service for viewers. Large part of p2p-streaming's issues are based on unpredictability of peers. Peers might have poor internet connections or leave the p2p-network unprompted. Issues caused by these peer unpredictabilities commonly manifest as interruptions in the broadcast, as poor video quality or as increased delay in joining the broadcast. Additionally p2p-streaming faces issues with a phenomena called flash crowd, where more peers try to join the network faster than the network can accomodate them. Unmanaged flash crowd leads to elongated join times and increases the likelihood of joining failing altogether.

Keywords: peer-to-peer, p2p, stream, video, multicast, application layer multicast

Kuviot

Kuvio 1. Havainnollistavia kuvia unicastista, IP-multicastista ja sovellustason multicastista mukaillen Chu ym. (2002).	8
Kuvio 2. Yksipuujärjestelmä.....	11
Kuvio 3. Monipuujärjestelmä	12

Sisällys

1	JOHDANTO	1
2	SUORATOISTO	3
	2.1 Suoratoisto palvelunäkökulmasta.....	3
	2.2 Suoratoisto sovellusnäkökulmasta	4
3	MULTICAST	6
	3.1 Verkkotason Multicast	6
	3.2 Multicast ilman verkkotason tukea	7
4	VERTAISVERKOT	10
	4.1 Vertaisverkkojen arkkitehtuuri	10
	4.2 Vertaisverkkojen rakenne	10
	4.2.1 Yksipuorakenne.....	11
	4.2.2 Monipuorakenne	12
	4.2.3 Mesh-pull	13
5	LÄHETYKSEN SAATAVUUS JA LAATU	15
	5.1 Vertaisverkkoon liittyminen	15
	5.2 Lähetyksen ylläpito	16
	5.3 Verkosta poistuminen	18
6	LÄHETYKSEN HÄIRIÖT	19
	6.1 Vertaisten vaihtuvuus.....	19
	6.2 Vertaisten heterogeenisuus	20
	6.3 Flash Crowd	22
7	YHTEENVETO.....	25
	LÄHTEET	27

1 Johdanto

Suoratoistossa (engl. *live streaming*) yksi lähettäjä tuottaa sisältöä reaaliaikaisesti sisältöä useille vastaanottajille reaaliaikaisesti. Esimerkiksi urheilutapahtuman yhtäaikainen katsojamäärä voi nousta satoihin tuhansiin (Guarnieri ym. 2021). Tapahtumien lisäksi käyttäjien tuottama sisältö palveluissa kuten YouTube ja Twitch on merkittävä osa nykyistä suoratoistoa.

Suurien käyttäjämäärien vuoksi suoratoiston palveluntarjoaja vaatii merkittävästi resursseja. Yksi keino vähentää palveluntarjoajan resurssien tarvetta on hyödyntää vastaanottajien edelleenlähetykskapasiteettia vertaisverkkojen (engl. *peer-to-peer networks, p2p-networks*) avulla. Vertaisverkoissa osallistuvat järjestelmät muodostavat päällysverkon muodostamalla yhteyksiä toistensa kanssa internetin välityksellä (Chu ym. 2002). Näin ne voivat vastaanottaa suoratoistoa toistensa kautta, näin vähentäen alkuperäisen lähettäjän kuormitusta.

P2p-suoratoisto kuitenkin kohtaa haasteita, kuten palvelunlaadun heikkenemistä, tilanteissa, joissa vertaisverkkojen rajoitteet törmäävät yhteen suoratoiston tarpeiden kanssa. Tutkimuksessa kartoitetaan näitä haasteita ja niiden vaikutuksia lähetykseen järjestelmän vakauden ja vastaanottajan palvelunlaadun näkökulmasta. Osa näistä haasteista johtaa vertaisverkkoarkkitehtuurin monimutkaisuudesta verrattuna asiakas-palvelin-arkkitehtuuriin. Vertaisverkkoihin liittymisprosessin, joka on kuvailtu artikkeleissa B. Li ym. (2008), Setton, Noh ja Girod (2006), Hei ym. (2007) ja Wu, Li ja Zhao (2012) monimutkaisuus johtaa pidempään viiveeseen vastaanottajan liittyessä lähetykseen.

Vertaisverkkoarkkitehtuurin monimutkaisuudesta johtuvien haasteiden lisäksi osa p2p-suoratoiston haasteista johtaa vertaisten ennalta-arvaamattomuudesta. Vastaanottajien edelleenlähetykskapasiteetissa saattaa olla suuriakin eroja riippuen laitteiden ja verkkoyhteyksien laadusta. Esimerkiksi monet protokollat, jotka toimivat hyvin homogeenisessä ympäristössä, suoriutuvat merkittävästi huonommin ympäristössä, joissa vertaisten verkkoyhteyksien laadun välillä merkittävää vaihtelua (Shen ym. 2011). Myös vertaisten ennalta-arvaamaton käyttäytyminen saattaa aiheuttaa ongelmia. Bo Li ym. (2008) mukaan vertaisten liittymisten ja poistumisten yhteisvaikutus on p2p-suoratoiston suunnittelun pääasiallinen haaste.

Luvussa 2 käydään läpi suoratoiston ominaisuuksia ja tutkitaan sen asettamia haasteita suoratoistoa tukeville järjestelmille. Luvussa 3 käsitellään ryhmälähetyksen eri muotoja suoratoistolähetysten näkökulmasta. Luku 4 käsittelee vertaisverkkoja, joissa hyödynnetään sovellustason suoratoiston esittelemää käsitettä vertaisten edelleenlähetyksen hyödyntämisestä. Luvussa 5 käydään läpi p2p-suoratoiston normaaliin toimintaan liittyviä rajoitteita ja luvussa 6 häiriötilanteiden vaikutuksia suoratoistoon. Luku 7 esittää yhteenvedon.

2 Suoratoisto

Tässä luvussa käydään läpi, mitä suoratoisto on ja millaisia teknisiä vaatimuksia se asettaa suoratoiston jakamiseen osallistuville järjestelmille. Suoratoistoa verrataan myös muihin ryhmän keskeistä tiedonjakoa hyödyntäviin sovelluksiin, kuten tiedostonjako ja konferenssisovellukset.

2.1 Suoratoisto palvelunäkökulmasta

Suoratoistossa ryhmä käyttäjiä vastaanottaa samanaikaisesti verkon kautta sisältöä yksittäiseltä lähettäjältä. Sisältö on useimmiten videota tai muuta multim mediasisältöä. Tässä tutkimuksessa termillä suoratoisto viitataan englanninkieliseen termiin live streaming, joka erotellaan termistä video-on-demand (VOD). VOD-systeemeissä käyttäjä voi toistaa mediaa riippumatta muista käyttäjistä ja vapaasti kelata videota eteen ja taaksepäin (Gelman ym. 1993). Suoratoistolähetyksiä seurataan reaaliaikaisesti, eivätkä ne tyypillisesti tue näitä ominaisuuksia. J. Liu ym. (2008) mukaan video-on-demand kuuluu erilliseen ongelmaan, koska vastaanottajat eivät lähtökohtaisesti vastaanota videota yhtäaikaaisesti.

Suoratoiston tekniset vaatimukset asettavat lähtökohtaisesti katsojat ja heidän käyttäytymisensä. Livelähetyksien katsojamäärät voivat mahdollisesti paisua hyvinkin suuriksi. Guarnieri ym. (2021) tutkivat jalkapallo-ottelulähetyksiä, joiden keskiarvoinen katselusessiomäärä oli yli 800000 per ottelu. Suosittujen otteluiden yhtäaikaainen katsojamäärä ylsi yli 600000 sessioon.

Käyttäjät saattavat toimia myös tavoilla, joita ei kohdata merkittävästi esimerkiksi tiedostonjaoissa tai konferenssisovelluksissa. Käyttäjät liittyvät ja poistuvat lähetyksestä vahvasti sen sisällön mukaan. Guarnieri ym. (2021) havaitsivat, että jalkapallo-ottelulähetyksien aikana suuri määrä käyttäjiä liittyi lähetykseen nopeasti pelin alkaessa ja pelin päättyttyä suuri osa käyttäjistä poistui hyvin lyhyen aikavälin sisällä. Lisäksi väliajan aikana merkittävä määrä käyttäjiä poistui lähetyksestä ja palasi väliajan päättyessä. Myös Hei ym. (2007) havaitsivat omassa tutkimuksessaan merkittävän määrän poistuvia käyttäjiä suosittujen ohjelmien päättyessä. Tämä tarkoittaa, että suoratoiston katsojat usein liittyvät lähetykseen ja poistu-

vat siitä mahdollisesti suurissakin ryppäissä. Guarnieri ym. (2021) tutkimuksessa suosittujen jalkapallo-otteluiden mediaani saapumistiheys oli 357 katsojaa sekunnissa, ja saapumistiheyden korkein kymmenys oli 469 uutta katsojaa sekunnissa.

Käyttäjien yhteys- ja laitetyyppi voi myös ennustaa ja määrittää käyttäjien toimintaa suoratoiston kanssa. On esimerkiksi myös havaittu ilmiö lyhyistä tai erittäin lyhyistä sessioista. Z. Li ym. (2015) mukaan merkittävä määrä käyttäjistä hylkäsi lähetyksen ennen kuin lähetyksen toistoa voitiin aloittaa. Mobiililaitteet ja mobiiliyhteydet lisäsivät todennäköisyyttä sille, että käyttäjä hylkäsi session ennen toiston alkua verrattuna ei-mobiili laitteisiin tai WiFi-yhteyksiin (Z. Li ym. 2015). Myös Guarnieri ym. (2021) havaitsivat mobiilikäyttäjien sessioiden olevan merkittävästi lyhyempiä kuin ei-mobiilikäyttäjillä. He kuitenkin arvoivat tämän johtuvan osittain siitä, että Brasiliassa mobiiliyhteydet ovat kalliita ja niissä on tiukat latausrajoitukset. Heidän tutkimuksessaan ei otettu kantaa ennen toistoa hylättyihin sessioihin.

2.2 Suoratoisto sovellusnäkökulmasta

Suoratoistolla on useita merkittäviä piirteitä, jotka määrittävät suoratoistosovelluksille asetettavia vaatimuksia. Yksi suoratoiston merkittävistä piirteistä on edellä mainittu systeemin suuri koko. Suoratoistolla on myös vaatimuksia reaaliaikaisuuden suhteen, data tulee toimittaa nopeasti ja jatkuvasti. Lievää datan myöhästymistä voidaan lievittää puskuroinnilla, mutta kokonaan myöhästynyt tai puuttuva data voi aiheuttaa epätoivottua toiston katkeilua. Videon suoratoisto tarvitsee myös paljon kaistaa laadukkaaseen lähetyksen ylläpitoon. Jos kaistaa ei ole tarpeeksi, olisi videolähetyksen laadun pystyttävä laskemaan tasaisesti mukautuakseen eri laatuksille yhteyksille. (J. Liu ym. 2008)

Suoratoisto on pohjimmiltaan datan jakoa ryhmälle vastaanottajia, joten suoratoiston tarpeita voidaan verrata muihin samankaltaisiin järjestelmiin. Näitä ovat esimerkiksi konferenssisovellukset ja tiedostonjakosovellukset.

Ääni- ja videokonferenssisovellukset, kuten Zoom ja Jitsi, ovat hyvin interaktiivisia, joten viive osallistujien välillä on hyvin tärkeä pitää matalana. Suoratoistossa katsojien vuorovaikutus on usein paljon epäsuorempaa, koska he eivät suoraan osallistu lähetykseen, joten pie-

ni viive on merkityksellisempää konferenssisovelluksissa. Lisäksi, toisin kuin suoratoistossa, konferenssisovelluksissa kuka tahansa osallistujista saattaa tuottaa dataa muille osallistujille. (J. Liu ym. 2008) Kuitenkin videokonferenssien koko jää usein huomattavasti pienemmäksi kuin suoratoistolähetysten (J. Liu ym. 2008). Suurien katsojamäärien ylläpitäminen tuottaa suoratoistossa haasteita, joita konferenssisovelluksien ei tarvitse kohdata samassa mittakaavassa.

Peer-to-peer -tiedostonjaolla voi olla suoratoiston tavoin valtava yhtäaikainen käyttäjämäärä. Kuitenkin p2p-tiedostonjaolla ei ole samanlaisia vaatimuksia reaaliaikaisuuden tai kaistan suhteen. Tiedostonjaossa ei esimerkiksi ole tärkeää, missä järjestyksessä data otetaan vastaan tai onko datan vastaanotossa katkoksia. (J. Liu ym. 2008) Suoratoistosovelluksissa sen sijaan katkeamaton toisto on tärkeää. Hitaasti jaettu tiedosto on yhtä hyödyllinen vastaanottajalle, mutta suoratoistossa toistosta myöhästynyt data ei ole enää käytettävissä.

3 Multicast

Tässä luvussa esitellään multicast-reititysjärjestelmää. Luvussa käsitellään IP-multicastia verrattuna unicastiin ja sen hyötyjä suoratoiston näkökulmasta. IP-multicast ei ole kuitenkaan käytettävissä internetin välityksellä yleisesti. Tämän vuoksi tutustutaan myös sovellustason multicasttiin.

3.1 Verkkotason Multicast

Verkkotason multicast, eli IP-multicast on yksi monista IP-verkoissa käytettävistä reititysjärjestelmistä. Reititysjärjestelmät määrittelevät, miten pakettikytketyssä verkossa paketti kulkee verkon läpi. Reititysjärjestelmiä on useita, kuten broadcast, anycast tai geocast, mutta tässä työssä keskitytään IP-multicastiin ja sen eroihin unicastista.

Unicast on yhdeltä yhdelle (engl. *one-to-one*) -reititysjärjestelmä. Paketilla on siis yksi määränpää, joka IP-verkoissa tunnustetaan yksilöllisellä IP-osoitteella. Esimerkiksi tyypillinen verkkosivuilla vierailu tapahtuu yleensä muodostamalla unicast-yhteys palvelimeen. Multicast on yhdeltä monelle (engl. *one-to-many*) -järjestelmä. Paketilla voi olla siis useita vastaanottajia. IP-multicastissa multicast-ryhmät tunnustetaan ryhmän IP-osoitteen perusteella, johon lähetetyt paketit toimitetaan ryhmän jäsenille (Deering ja Cheriton 1990). Toisin kuin unicast-lähetyksessä, missä IP-osoite osoittaa yksittäiseen laitteeseen verkossa, multicast-ryhmän osoite ei osoita mihinkään konkreettiseen verkon laitteeseen. Multicast-ryhmässä voi olla mielivaltainen määrä jäseniä, ryhmä voi olla myös tyhjä (Deering 1989).

Multicastin käyttäminen ryhmälle lähettäessä on lähettäjälle tehokkaampaa kuin joukko unicast-yhteyksiä (Deering ja Cheriton 1990). Unicastia hyödyntäessä lähettäjän on lähetettävä sama data jokaiselle vastaanottajalle erikseen, siinä missä IP-multicastia käyttäessä sama data lähetetään vain kerran. IP-multicastia ei kuitenkaan voi mielekkäästi hyödyntää suoratoiston tarpeisiin, koska se ei ole yleisesti käytettävissä internetin kautta. Diot ym. (2000) kuvailevat IP-multicastin käyttöönottoon liittyviä ongelmia ja haasteita. IP-multicast on toteutettu OSI-mallin verkkotasolla osana Internet-protokollaa. Tällöin multicastin toiminnallisuus on toteutettu multicast-reitittimien avulla. Kuitenkin multicast voidaan toteuttaa myös

ilman verkkotason tukea hyödyntäen vastaanottajien edelleenlähetyskapasiteettia.

3.2 Multicast ilman verkkotason tukea

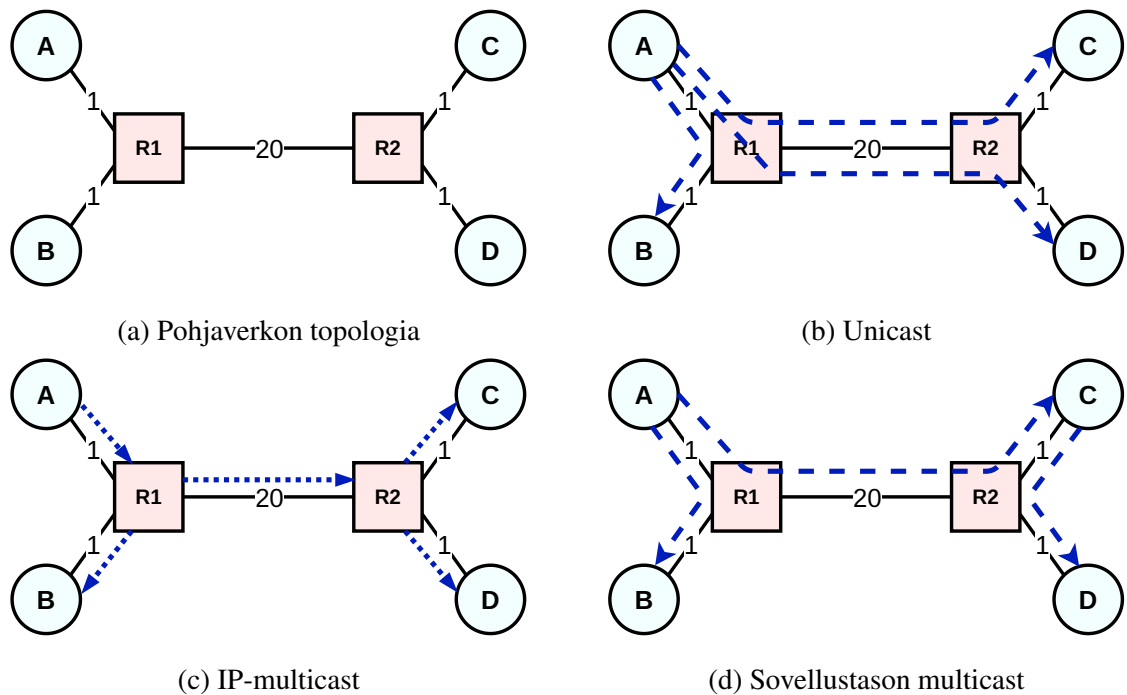
OSI-mallin sovellustasolla toteutettua multicastia kutsutaan sovellustason multicastiksi (engl. *application-layer multicast*). Sovellustason multicastissa lähetystä vastaanottavat järjestelmät hoitavat datan edelleenlähettykseen liittyvän toiminnallisuuden (Banerjee, Bhattacharjee ja Kommareddy 2002). Tämä tarkoittaa, että kun vastaanottaja on vastaanottanut osan lähetyksestä, se voi puolestaan lähettää saman lähettyksen jollekin toiselle halukkaalle vastaanottajalle. Näin voidaan vähentää kuormitusta alkuperäiseltä lähettäjältä.

Sovellustason multicastissa vertaisjärjestelmät hoitavat kaikki multicast-operaatiot, kuten reitityksen, jäsenten hallinnan ja pakettien monistamisen. Vertaiset kommunikoivat luomalla päällysverkon (engl. *overlay network*) pohjaverkon päälle muodostamalla unicast-yhteyksiä keskenään. Koska sovellustason multicast tarvitsee vain unicast-yhteyksiä, se ei vaadi laajennettua toiminnallisuutta pohjaverkolta, toisin kuin IP-multicast. Tämän vuoksi sovellustason multicast voidaan ottaa käyttöön verkoissa, joissa IP-multicastin hyödyntäminen on mahdotonta. (Chu ym. 2002) Tämä tekee sovellustason multicastista monipuolisen työkalun ryhmälähettyksien näkökulmasta.

Kuvio 1 kuvaa havainnollistavasti eri ryhmälähettysmuotojen eroja. Solmut R1 ja R2 kuvaavat reitittämiä ja solmut A, B, C ja D ovat lähettykseen osallistuvia järjestelmiä. Solmu A lähettää vastaanottajille B, C ja D. Linkeissä esiintyvät numerot kuvaavat kunkin linkin kustannusfunktiota.

Kuvio 1b esittää tilanteen, jossa A lähettää paketin unicast-lähettyksellä jokaiselle vastaanottajalle erikseen. Tällöin A joutuu lähettämään datan kerran jokaista vastaanottajaa kohden. Sama data kulkee myös monta kertaa samojen linkkien kautta. Data kulkee A-R1 linkin kautta kolmesti ja kaksi kertaa hyvin kalliin R1-R2 linkin kautta. Kuitenkin jokainen yksittäinen vastaanottaja saa lähettyksen halvinta reittiä. Reitin solmusta A solmuun D kustannus on 22.

Kuviossa 1c IP-multicast-tapauksessa solmu A joutuu lähettämään vain yhden paketin. Koska reitittimet hoitavat datan uudelleenlähettykset, lähetty ei joudu kulkemaan samoja link-



Kuvio 1: Havainnollistavia kuvia unicastista, IP-multicastista ja sovellustason multicastista mukaillen Chu ym. (2002).

kejä useampaan kertaan. Huomion arvoista on myös, että samoin kuin unicast-tapauksessa, jokainen vastaanottaja saa lähetyksen halvinta reittiä pitkin. Reitin A-D kustannus on yhä 22.

Kuviossa 1d kuvattu sovellustason multicast mahdollistaa, että solmun A tarvitsee lähettää data vain osalle vastaanottajista. Tämä vähentää liikennettä lähellä lähettäjä olevissa linkeissä. Älykkäästi rakennetuissa päällysverkoissa voidaan välttää käyttämästä kalliita linkejä, kuten R1-R2, moneen kertaan, vaikka sen takana olisi useampia vastaanottajia. Kuitenkin sovellustason multicastissa joudutaan käyttämään samoja linkejä moneen kertaan lähellä solmuja, jotka edelleenlähettävät dataa. Koska solmu C saa datan solmulta A ja lähettää sen solmulle D, linkki R2-C on kuljettava useampaan kertaan. Sovellustason multicast ei myöskään pysty tarjoamaan jokaiselle vastaanottajalle unicastiin verrattavaa reittiä. Reitti A-D koostuu reiteistä A-C ja C-D, joten reitin kustannus päättyy olemaan 22+2.

Sovellustason multicastissa osa solmuista on yhteydessä verkkoon toisten solmujen kautta, joten esimerkin solmun C poistuminen irrottaisi myös solmun D verkosta. Tässä tilanteessa

solmun D olisi kiinnittädyttävä verkkoon uudelleen. IP-multicastiin verrattuna sovellustason multicastilla on muutamia välttämättömiä rajoituksia. Osa solmuista joutuu lähettämään saman datan useampaan kertaan, että verkko ei muutu jonoksi. Muiden solmujen kautta datan saaminen on myös välttämättä hitaampaa kuin suoraan alkuperäiseltä lähettäjältä, koska data kulkee epäsuoraa reittiä.

4 Vertaisverkot

Tässä luvussa tutustutaan vertaisverkkoihin. Suoratoiston yhteydessä vertaisverkkojen avulla voidaan hyödyntää sovellustason multicastin esittelemää käsitettä vastaanottajien edelleenlähetyskapasiteetin hyödyntämisestä. Vertaisverkkojen yleisen arkkitehtuurin lisäksi tutustutaan suoratoistoon käytettävien vertaisverkkojen yleisimpiin rakenteisiin: tree-push ja mesh-pull.

4.1 Vertaisverkkojen arkkitehtuuri

Suurin osa internetissä käytettävissä olevista palveluista hyödyntää asiakas-palvelin-arkkitehtuuria (engl. *client-server architecture*), jossa asiakas ja palvelinroolit ovat selkeästi erillään. Vertaisverkkoarkkitehtuureissa (engl. *peer-to-peer*, *p2p*) verkkoon osallistujat toimivat yhtä aikaa palvelimina ja asiakkaina toisilleen. Schollmeier (2001) mukaan tätä palvelimen ja asiakkaan yhteisroolia kutsutaan server-rooliksi. Suoratoistosovelluksissa server-rooli esiintyy vastaanottajien edelleenlähetyskapasiteetin hyödyntämisen kautta. Vastaanottajat voivat näin vastaanottaa lähetystä olemalla yhteydessä muihin vastaanottajiin alkuperäisen palvelimen sijaan. Baochun Li ym. (2013) mukaan p2p-suoratoistojärjestelmän on mahdollista kasvattaa resursseja kysynnän kasvaessa, koska vastaanottajat lähettävät videota edelleen. Vertaisten resurssien hyödyntäminen vähentää keskeisen palvelimen raskautusta, joten palvelin kykenee palvelemaan enemmän asiakkaita ilman lisättyjä resursseja (Xu ym. 2012). Myös Liu, Guo ja Liang (2008) ja B. Li ym. (2008) tuovat esille vertaisverkkosovelluksien itseskaalautuvuuden kysynnän mukaan etuna asiakas-palvelin-arkkitehtuureihin.

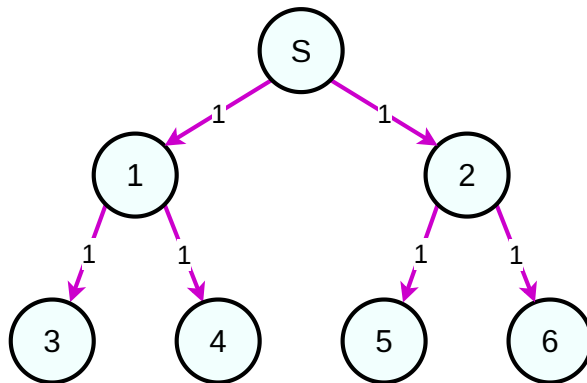
4.2 Vertaisverkkojen rakenne

Vertaisverkkoarkkitehtuurissa verkoilla esiintyy erilaisia topologioita. Sovelluksen käyttötarkoitus määrittää sopivimman topologian vertaisverkolle. Epäsopiva topologia saattaa rajoittaa tai haitata sovelluksen tehokkuutta tai käyttötarkoituksia. Suoratoistossa on hyvä huomioda, miten hyvin informaatio leviää suurissakin verkossa ja kuinka haastavaa verkon rakennetta on ylläpitää vertaisten liittyessä ja poistuessa. Suoratoistossa on käytetty pääasiassa

kahdenlaisia rakenteita. Shen ym. (2011) mukaan suoratoistoon käytettävien vertaisverkkojen topologiat jakautuvat karkeasti tree-push ja mesh-pull rakenteisiin. Nämä esitellään seuraavaksi.

4.2.1 Yksipuurakenne

Tree-push topologia perustuu vertaisten järjestämiseen puurakenteeseen. Solmulla on yksi vanhempi-solmu, joka puskee sille lähetystä. Vanhempi voi olla lähettävä palvelin tai toinen server-solmu. Solmu saattaa myös toimia vanhempana muille solmuille. Mitä enemmän resursseja ja edeenlähetykskapasiteettia solmulla on, sitä enemmän lapsisolmuja se pystyy tukemaan. Solmuja, joilla ei ole yhtään lapsisolmuja, kutsutaan lehtisolmuiksi. Puujärjestelmät voi jakaa kahteen luokkaan: yksipuujärjestelmät ja monipuujärjestelmät.



Kuvio 2: Yksipuujärjestelmä

Kuvio 2 kuvaa yksinkertaista yksipuujärjestelmää, jossa on yksi palvelin (S) ja kuusi numeroitua vastaanottajaa 1-6. Nuolissa esiintyvät numerot kuvaavat abstraktia tiedonsiirtomäärää per aikayksikkö. Solmun on vastaanotettava dataa nopeudella yksi per aikayksikkö, jotta se voi ylläpitää jatkuvaa toistoa.

Lähetysrasite on jakaantunut useammalle solmulle, joten yksipuujärjestelmä vähentää alkuperäisen lähettäjän kuormitusta. Kuitenkin hyvin nopeasti kohdataan yksipuujärjestelmän ongelmia, jotka muut rakenteet hoitavat paremmin.

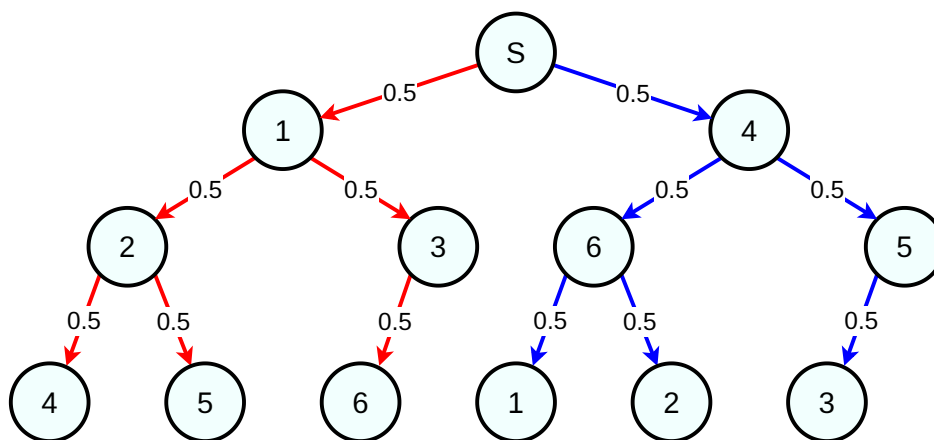
Shen ym. (2011) mukaan yksipuujärjestelmissä lehtisolmujen lähetyskapasiteettia ei hyödynnetä. Tämän voi myös havaita kuvioista 2. Solmut 3, 4, 5 ja 6 eivät lähetä kellekään, joten

niiden edelleenlähetyksenopeus on nolla. Kolmella lähettävällä solmulla S, 1 ja 2 edelleenlähetyksenopeus on kaksi, eli kaksinkertainen siihen nähden mitä ne itse vastaanottavat. Sisäiset solmut siis kompensoivat lehtisolmuja. Oikeissa käyttötapauksissa tilanne saattaa olla vielä vinoutuneempi, koska matalat ja leveät puut ovat suoratoistolle suotavampia. Liu, Guo ja Liang (2008) mukaan puun syvyys lisää viivettä solmuille, jotka ovat kauempana alkuperäisestä lähettäjistä. Lisäksi, mitä kauempana solmu on alkuperäisestä lähettäjistä sitä todennäköisempää on, että sen vanhempisolmuissa esiintyy hitaita yhteyksiä (Magharei, Rejaie ja Guo 2007). Matalissa ja leveissä puissa lehtisolmujen suhde sisäisiin solmuihin on suurempi.

Shen ym. (2011) mukaan puurakenteen ylläpitäminen saattaa olla raskasta ja tämän vuoksi puun uudelleenrakentaminen saattaa olla hidasta, kun vertaisia liittyy tai poistuu. Tämä puolestaan saattaa lisätä häiriöitä lähetyksessä poistuvien solmujen lapsille.

4.2.2 Monipuurakenne

Monipuujärjestelmissä lähetyksen pilkkotaan useampaan alilähetykseen ja jokaiselle alilähetykselle muodostetaan oma puu (Liu, Guo ja Liang 2008). Toteutuksesta riippuen vertainen liittyy kaikkiin tai osaan puista. Esimerkiksi Magharei, Rejaie ja Guo (2007) kuvailemassa systeemissä, joka käyttää Multiple Description Coding (MDC) -koodausmenetelmää, vertainen liittyy useampaan puuhun, jos sillä on tarpeeksi latauskapasiteettia. Magharei, Rejaie ja Guo (2007) mukaan vertaiset ovat sisäisiä solmuja vain yhdessä puussa, jotta vertaisten resursseja voidaan hyödyntää tehokkaasti,



Kuvio 3: Monipuujärjestelmä

Kuviossa 3 kuvataan samaa yhden lähettäjän ja kuuden vertaisen tilannetta kuin kuvioissa 2, mutta vertaiset on järjestetty kahteen puuhun. Vertaiset, jotka ovat sisäisiä vertaisia toisessa puussa, ovat lehtisolmuja toisessa.

Kuten kuvioista huomataan, jakamalla vertaiset kahteen puuhun lähetyksitasoa voidaan jakaa huomattavasti tasaisemmin solmujen kesken. Kaikki solmut vastaanottavat lähetystä yhä yksi per aikayksikkö, mutta myös lähettävät maksimissaan yksi per aikayksikkö, toisin kuin yksipuujärjestelmäsimerkissä. Jos vertaisten maksimilähetyksikapasiteetti olisi ollut yksi per aikayksikkö yksipuujärjestelmässä, niin puun olisi täytynyt muuttua ketjuksi.

Koodausmenetelmästä riippuen monipuujärjestelmä voi olla myös kestävämpi vertaisten poistussa järjestelmästä. Esimerkiksi MDC-koodausta käytettäessä video pystytään toistamaan, vaikka toisen puun data puuttuisi kokonaan. Tällöin yhdessä puussa tapahtuvat häiriöt eivät välttämättä katkaise lähetystä toisin kuin yksipuujärjestelmässä. Nämä hyödyt kuitenkin riippuvat vahvasti käytetystä koodausmenetelmästä.

Monipuukurakenteella on kuitenkin joitain merkittäviä heikkouksia. Liu, Guo ja Liang (2008) mukaan puun ylläpito vaikeutuu, kun vertaisia liittyy ja poistuu usein. Ramzan, Park ja Izquierdo (2012) ja Shen ym. (2011) mukaan monipuukurakenteen monimutkaisuus lisää sen ylläpidon haasteellisuutta. Tämä saattaa lisätä vertaisten kokemia häiriöitä puuta muokattaessa.

4.2.3 Mesh-pull

Mesh-pohjaisissa järjestelmissä vertaiset muodostavat satunnaisesti järjestäytyneen päällysverkon, jossa jokainen vertainen pyrkii muodostamaan naapuriyhteyksiä muihin vertaisiin (Magharei, Rejaie ja Guo 2007). Vertaiset myös muodostavat ja katkaisevat yhteyksiä toisiin vertaisiin dynaamisesti (Liu, Guo ja Liang 2008). Tämä tarkoittaa, että toisin kuin puurakenteessa, mesh-pohjaisilla verkoilla ei ole tiukasti määriteltyä rakennetta, ja niiden rakenne saattaa muuttua, vaikka vertaisia ei liittyisi tai poistuisi. Tämä rakenne on jokseenkin samankaltainen kuin tiedostonjakoon käytettävillä vertaisverkoilla, kuten BitTorrent.

Mesh-verkoissa vertaisilla ei ole määrättyjä vanhempia, vaan vertainen voi vastaanottaa videota useammalta naapurilta, joten vaikka yksi naapuri poistuisi, vertainen voi yhä vastaan-

nottaa muilta naapureiltaan (Liu, Guo ja Liang 2008). Puupohjaisissa verkoissa kunkin datapaketin kuljetusreitti määritellään verkon rakennusvaiheessa, kun taas mesh-pohjaisissa verkoissa paketin kuljetusreitti muokkautuu jatkuvasti saatavilla olevien resurssien mukaan (Magharei, Rejaie ja Guo 2007). Pakettien muuttuvien kuljetusreittien vuoksi puupohjausten verkkojen push-strategian käyttö vaikeutuu. Potentiaalisessa mesh-push-systeemissä, jossa vertaiset automaattisesti lähettävät saamansa datan edelleen naapureilleen, vertaiselle saateen lähetetään sama data moneen kertaan usean eri naapurin toimesta (Liu, Guo ja Liang 2008). Tämä selkeästi tuhlaa vertaisten edelleenlähetyksen kapasiteettia. Push-strategian sijaan mesh-verkot pääasiallisesti hyödyntävät pull-systeemiä. Siinä videolähetyksen on jaettu pienempiin palasiin (engl. *chunk*). Vertaiset lähettävät ajoittain puskurikarttojaan (engl. *buffer maps*) naapureilleen kertoakseen, mitä videon palasia niillä on. Näiden puskurikarttojen perusteella vertaiset päättävät, keneltä naapurilta ne pyytävät mitään videon palasia. Nämä päätökset tehdään aikataulutusalgoritmeja (engl. *scheduling algorithms*) hyödyntäen (Bideh, Akbari ja Sheshjavani 2016)

Magharei, Rejaie ja Guo (2007) mukaan mesh-pull-systeemit pärjäävät paremmin monissa tilanteissa verrattuna puupohjaisiin systeemeihin. Heidän mukaansa tämä johtuu pääasiassa siitä, että mesh-pull-pohjaisissa järjestelmissä paketit kykenevät kulkemaan useita polkuja pitkin, näin minimoiden huonojen yhteyksien vaikutusta muuhin verkkoon, kun taas puupohjaisissa järjestelmissä solmun huono yhteys vaikuttaa kaikkiin sen jälkeläisiin. Mesh-pohjaiset systeemit myös pärjäävät paremmin vertaisten vaihtuvuuden ollessa korkea (Magharei, Rejaie ja Guo 2007). Kaupallisissa ratkaisuissa mesh-pull-tekniikat ovatkin päätyneet suosittumiksi kuin tree-push-tekniikat (Shen ym. 2011).

5 Lähetyksen saatavuus ja laatu

Tässä luvussa käydään läpi suoratoistoverkaisverkon normaalia toimintaa ja arvioidaan sen kykyä ylläpitää laadukasta lähetystä. Luvussa arvioidaan myös, kuinka saavutettavissa p2p-suoratoistosovelluspalvelu on potentiaalisille vastaanottajille.

5.1 Vertaisverkkoon liittyminen

Kun katsoja haluaa vastaanottaa lähetystä, hänen on ensin liityttävä vertaisverkkoon. Koska katsojan tulee mahdollisesti ottaa yhteys ennalta tuntemattomiin vertaisiin, liittymisprosessi on monimutkaisempi kuin yhteyden ottaminen yhteen tunnettuun IP-osoitteeseen asiakaspalvelin-arkkitehtuurissa. Vertaisverkoissa on usein bootstrapping-solmu, joka ylläpitää osoitusta tai kokonaista listaa aktiivisista solmuista vertaisverkoissa. Liittyvä solmu ottaa yhteyden bootstrapping-solmuun ja saa listan solmuista, joihin se voi pyrkiä muodostamaan yhteyden. Tämänkaltainen systeemi on kuvailtu muun muassa artikkeleissa B. Li ym. (2008), Setton, Noh ja Girod (2006), Hei ym. (2007) ja Wu, Li ja Zhao (2012)

Uuden solmun saatua listan verkon aktiivisista solmuista, sen on päätettävä vertaisten valikointiprotokollan avulla, mihin solmuihin se pyrkii ottamaan yhteyden. Zhang ja Hassanein (2012) mukaan vertaisten valinta voi perustua satunnaisuuteen, vertaisten työmäärään tai vertaisten väliseen kaistaan tai viiveeseen. Vertaisten valinnassa on mahdollisesti hyödyllistä priorisoida yhteyksiä vertaisiin, jotka ovat lähellä pohjaverkon topologiassa. Shen ym. (2011) mukaan datan määrän kasvaessa, kaukaisten naapureiden tuottama raskaus saattaa aiheuttaa haasteita pohjaverkon resurssien hallinnassa.

Vertaisverkkojen kasvukyky on merkittävä ja yksi p2p-suoratoiston vahvuuksista. Esimerkiksi Zhang ja Hassanein (2012) mukaan p2p-suoratoistopalvelu PPSstream on raportoinut pystyneensä ylläpitämään lähetystä kymmenien miljoonien katsojien kanssa vuonna 2010. Tämä antaa ymmärtää, että vertaisverkon kasvava koko ei ole merkittävä rajoittava tekijä p2p-suoratoistolle. Vertaisverkot sen sijaan todennäköisesti hyötyvät suurista käyttäjämääristä. F. Liu ym. (2012) mukaan systeemi kykenee kasvamaan sitä nopeammin, mitä suurempi vanhojen vertaisten suhde on liittyviin vertaisiin. Sen sijaan vertaisverkko ei pysty

kuitenkaan kasvamaan loputtoman nopeasti. Liian nopea kasvuvauhti voi johtaa flash crowd -ilmiöön, josta kerrotaan enemmän osiossa 6.3

Aloitusviiveen pituus on usein esille tuotu ongelma p2p-suoratoistosovelluksissa, se mainitaan muun muassa artikkeleissa J. Liu ym. (2008), Ghaderzadeh, Kargahi ja Reshadi (2018) ja Liu (2010). Aloitusviiveeksi määritellään aika, joka kuluu solmun yrityksestä liittyä verkkoon siihen, että videon toisto voi alkaa. Tämä koostuu kahdesta komponentista: vertaisverkkoon liittymisestä aiheutuvasta viiveestä ja videon puskuroinnista aiheutuvasta viiveestä. Pienempi määrä puskurointia voi nopeuttaa aloitusviivettä, mutta se voi vaikeuttaa toiston sulavaa jatkumista (Bo Li ym. 2008). Puskurointi on tärkeä keino viiveen ja kaistan muutoksien kompensoinnissa (Liu 2010).

Aloitusviivettä ja sen vähentämistä on tutkittu huomattavasti. Seuraavaksi esitellään muutamia kokeellisia tuloksia aloitusviiveistä p2p-suoratoistojärjestelmissä. On hyvä huomata, että tulokset eivät välttämättä ole suoraan vertailtavissa. Tutkimusasetelmat saattavat poiketa toisistaan esimerkiksi oletetun palvelinkapasiteetin, alustavan puskuroinnin määrän, vaaditun tiedonsiirtonopeuden, oletetun vertaisten jakokapasiteetin ja vertaisten vaihtuvuuden mukaan.

Hei ym. (2007) mittauksien mukaan suoratoistopalvelu PPLive saavutti noin 10-20 sekunnin aloitusviiveen suosituilla kanavilla, kun taas vähemmän suosituilla kanavilla se saattoi venyä kahteen minuuttiin. Bideh, Akbari ja Sheshjavani (2016) saavuttivat simulaatioissaan alle 20 sekunnin aloitusviiveen noin 80 % vertaisista. Yin ym. (2009) mukaan LiveSky-systeemin aloitusviive oli noin 15 sekuntia. B. Li ym. (2008) mukaan Coolstreaming saavutti alle 5 sekunnin aloitusviiveen hyödyntäen lisättyä palvelinkapasiteettia (24 palvelinta). Tästä voi huomata, että p2p-suoratoistojärjestelmien aloitusviive on keskimäärin suurempi kuin nykyisten CDN-pohjaisten järjestelmien, kuten Twitch ja YouTube.

5.2 Lähetyksen ylläpito

Kun vertainen on liittynyt vertaisverkkoon ja vastaanottanut tarpeeksi videota aloittaakseen toiston, se on onnistuneesti liittynyt lähetykseen. Lähetyksen ylläpidon aikana vertainen vastaanottaa videota ja lähettää sitä muille vertaisille. Lähetyksen vastaanottamis- ja edelleen-

lähetyksen prosessi riippuvat vahvasti vertaisverkon topologiasta.

Tärkeä osa mesh-pohjaisia suoratoistosysteemejä ovat palasten aikataulutusalgoritmit, joiden avulla vertaiset organisoivat, miltä naapurilta pyytää mitään palasta ja missä järjestyksessä. Tämä on tarpeellista, koska mesh-pohjaisissa verkoissa ei ole selkeitä lapsi-vanhempisuhteita. Optimaalinen aikataulutus on NP-vaikea ongelma (Zhang ym. 2005). Tämän vuoksi on hyödynnettävä yksinkertaisempia heuristiikoita tai algoritmeja, joita voidaan realistisesti ajaa p2p-suoratoiston viiveherkässä ympäristössä.

Bideh, Akbari ja Sheshjavani (2016) mukaan yksi pääasiallinen syy mesh-pohjaisen p2p-suoratoiston häiriöihin ovat aikataulutusalgoritmit, jotka eivät ota huomioon videolähetysten luonnetta. Videolähetys on vastustuskykyinen datan menetykselle verrattuna esimerkiksi tiedostonsiirtoon. Videon kuvilla (engl. *frame*) on tyypillisesti eritasoinen tärkeys videon kokonaisvaltaisen laadun kannalta, verraten MPEG-standardien mukaisesti Group of Pictures -rakenteen I-kuviin, P-kuviin ja B-kuviin. Videon toistamista voidaan jatkaa mielekkäästi, vaikka joitain vähemmän tärkeitä kuvia puuttuisi. Bideh, Akbari ja Sheshjavani (2016) mukaan lähes kaikki aikataulutusalgoritmit suoriutuvat yhtä hyvin, kun resursseja, kuten kaistaa, on tarpeeksi. Kuitenkin videolaatujen kasvaessa resursseja ei ole koskaan yltäkyläisesti, joten älykkäät aikataulutusalgoritmit, jotka hyödyntävät rajallisia resursseja tehokkaasti, ovat tarpeellisia (Bideh, Akbari ja Sheshjavani 2016). Hyödyntäen paranneltuja lähettäjän ja vastaanottajan aikataulutusalgoritmeja Bideh, Akbari ja Sheshjavani (2016) pystyivät vähentämään aloitusviivettä ja kuvakatoa (engl. *frame loss*).

Huomionarvoinen viive suoratoistossa on toistoviive, joka viittaa siihen, kuinka paljon vertainen on jäljessä verrattuna alkuperäiseen lähettäjään. P2p-suoratoiston viive juontaa samoista syistä kuin muidenkin sovellustason multicastin kaltaisten tekniikoiden hyödyntävien sovelluksien. Vertainen saa datan usein muiden vertaisten kautta, ja se ei voi lähettää dataa eteenpäin ennen kuin se on itse vastaanottanut sen. Suoratoistossa toistoviive ei ole yhtä haitallista kuin esimerkiksi konferenssisovelluksissa. Kuitenkin lyhyemmät toistoviiveet ovat suoratoistolle toivottava ominaisuus, koska se helpottaa mahdollista katsojien vuorovaikutusta lähetyksen kanssa. On myös havaittu, että vertaisten toistoviive voi olla hyvinkin poikkeava. Hei ym. (2007) mukaan vertaisten toistoviiveet voivat olla minuuttienkin päässä toisistaan. Heidän mukaansa tämä saattaa estää näitä vertaisia tarjoamasta toisilleen hyödyll-

listä dataa.

Lähetyksen aikana vertaiset tekevät yhteistyötä lähetyksen jakamisessa kaikille vertaisille. Yhteistyöhaluttomat vertaiset saattavat haitata muun verkon toimintaa (Ramzan, Park ja Izquierdo 2012). Yksi keino vähentää yhteistyöhaluttomia vertaisia on ylläpitää joitain insentiivimekanismeja, jotka palkitsevat vertaisia verkon kannalta edullisesta toiminnasta. J. Liu ym. (2008) mukaan BitTorrentin kaltainen *tit-for-tat*-insentiivi ei suoraan toimi suoratoistossa sen reaaliaikaisen luonteen vuoksi. Heidän mukaansa tehokkaan insentiivisysteemin kehitys on vielä avoin kysymys.

5.3 Verkosta poistuminen

P2p-suoratoistossa hyödynnetään vertaisverkon solmujen edelleenlähetyksen kapasiteettia. Jos solmu poistuu tai muusta syystä lakkaa lähettämästä, sen lapsi- tai naapurivertaiset lakkaavat saamasta lähetystä, joten verkko on korjattava, että toistoa voidaan jatkaa (J. Liu ym. 2008). Kuitenkaan suoratoistojärjestelmä ei pysty hallitsemaan osallistuvien solmujen käyttäytymistä. Se ei pysty pakottamaan solmuja olemaan irrottautumatta tai noudattamaan tiettyä irrottautumisprotokollaa. Solmu voi esimerkiksi vain kaatua. Tämän vuoksi vertaisverkon pitää pystyä tehokkaasti toipumaan mielivaltaisten solmujen äkillisistä poistumisista.

Poistuvien vertaisten lapsivertaiset ja naapurit joutuvat etsimään uusia vertaisia, joihin muodostaa yhteys. Tällä on hyvin samankaltaisia vaikutuksia verkkoon kuin uusien vertaisten saapumisella (Bo Li ym. 2008). Puupohjaisissa verkoissa lähellä puun juurta olevien solmujen poistuminen saattaa häiritä suuren määrän solmujen datan saantia (J. Liu ym. 2008). Mesh-pohjaisissa verkoissa useat solmut voivat tarjota saman datan, joten solmun muut naapurit voivat kompensoida menetettyä naapuria (J. Liu ym. 2008). Verkon korjaaminen ei kuitenkaan ole välitön toimenpide. Puskurointi on merkittävä osa solmujen sietokykyä vertaisten poistumista vastaan. Vertaisen tulisi puskuroida dataa ainakin sen verran, että toistoa on mahdollista jatkaa katkeamattomana verkon korjaamisen ajan (Ramzan, Park ja Izquierdo 2012).

6 Lähetyksen häiriöt

Tässä luvussa käsitellään vertaisverkkojen toimintaa ja laadukkaan lähetyksen ylläpitoa vaikeuttavia tekijöitä. Käsiteltävät aiheet ovat vertaisten vaihtuvuus, vertaisten heterogeenisyys ja flash crowd -ilmiö.

6.1 Vertaisten vaihtuvuus

Vertaisten liittymisien ja poistumisien yhteisvaikutusta kutsutaan vertaisten vaihtuvuudeksi (engl. *peer churn*) (Stutzbach ja Rejaie 2006). Vertaisten vaihtuvuus saattaa olla suuri, vaikka vertaisten kokonaismäärä ei muuttuisi paljon, jos vertaisia liittyy ja poistuu suunnilleen yhtä paljon. Merkittävä vertaisten vaihtuvuus aiheuttaa räsitystä vertaisverkolle ja saattaa häiritä lähetystä. Bo Li ym. (2008) mukaan vertaisten vaihtuvuus on p2p-suoratoiston suunnittelun pääasiallinen haaste.

Yin ym. (2009) mukaan vertaisten vaihtuvuus saattaa aiheuttaa toiston katkeilua, esimerkiksi vertaisen vanhemman poistuessa vertaisverkosta. Katkeilut datan toimituksessa ovat ehkä ilmeisin vertaisten vaihtuvuuden vaikutus, mutta vertaisten vaihtuvuudella voi olla muitakin merkittäviä negatiivisia vaikutuksia. Setton, Baccichet ja Girod (2008) mukaan vertaisten vaihtuvuus voi vääristää verkon rakennetta epätehokkaaseen muotoon. Heidän mukaansa tämä on erityisen näkyvää suurissa poistumistapahtumissa, jonka seurauksena joukko matalan edelleenlähetyksyvyn vertaisia saattaa liittyä lähelle alkuperäistä palvelinta ja ikään kuin tukkia vertaisverkon. Tämän vuoksi verkon rakenteen jatkuva ylläpitäminen on tärkeää (Setton, Baccichet ja Girod 2008). Vertaisten vaihtuvuudella on myös vaikutusta järjestelmän tehokkuuteen, esimerkiksi Bo Li ym. (2007) mukaan vertaisten vaihtuvuus oli suurin vaikuttava tekijä CoolStreaming-systeemin tehokkuuteen.

Suunniteluratkaisuilla voidaan vaikuttaa siihen, kuinka sietokykyinen vertaisverkko on vertaisten vaihtumiselle. Puupohjaiset rakenteet ovat herkempiä vertaisten vaihtuvuudelle kuin mesh-pohjaiset, koska puun sisäisen vertaisen poistuessa se katkaisee lähetyksen koko alapuolta, kun taas mesh-pohjaisissa vertaisilla on useita naapureita, joilta se voi vastaanottaa lähetystä, jos yksi naapureista poistuu verkosta (Liu, Guo ja Liang 2008). Tätä tukevat Mag-

harei, Rejaie ja Guo (2007), joiden mukaan mesh-pohjaiset rakenteet pärjäävät huomattavasti paremmin massapoistumisilmiön vääristämässä verkossa.

Vertaisten vaihtuvuuden vaikutuksien suuruutta on kuitenkin vaikea arvioida yleisellä tasolla, koska vaikutukset ovat hyvin riippuvaisia niiden kontekstista. Magharei, Rejaie ja Guo (2007) mukaan vertaisten vaihtuvuuden vaikutus lähestyksen laatuun on lopulta monen asian, kuten palautumismekaniikoiden luonteen, vaihtuvuuden luonteen ja vertaisten puskuroinnin määrän, summa. Yin ym. (2009) havaitsivat myös että CDN:ien hyödyntäminen vähentää vertaisten vaihtuvuuden vaikutuksia, koska vaihtuvuudesta kärsivät vertaiset voivat tarvittaessa hyödyntää CDN-palvelinkapasiteettia, kunnes ne voivat jälleen hyödyntää vertaisverkkoa. Tämä on teema, joka toistuu monien p2p-suoratoistosysteemien kohtaamien ongelmien kanssa. Tekijät, kuten puskuroinnin määrä ja serverikapasiteetti, vaikuttavat vertaisverkon kohtaamiin ongelmiin, mutta ne ovat kompromisseja, jotka eivät aina välttämättä ole toivottuja tai saatavilla.

6.2 Vertaisten heterogeenisuus

Internetin käyttäjissä on valtavasti eroja laitteiden ja verkkoyhteyksien saralla. Osa käyttää tehokkaita pöytäkoneita ja nopeita langallisia yhteyksiä, ja osa käyttää heikompia mobiililaitteita epävakaaammilla langattomilla yhteyksillä. Tämän internetin käyttäjien moninaisuuden vuoksi p2p-suoratoistojärjestelmä ei voi tehdä oletuksia vertaisten lataus- tai edelleenlähetykskapasiteetista. Näin ollen kaikkia vertaisia ei voi kohdella samalla tavalla ja kaikki vertaiset eivät ole yhtä hyödyllisiä. Shen ym. (2011) mukaan monet protokollat, jotka teoreettisesti toimivat hyvin homogeenisissa ympäristöissä, suoriutuvat merkittävästi huonommin heterogeenisissä ympäristöissä.

Vertaisten heterogeenisen latauskapasiteetin vuoksi suoratoistosysteemi ei voi tukea vain yhdenlaista lähetyslaatuja. Vastaanottajat, joilla on hidas yhteys, eivät välttämättä kykene lataamaan liian korkealaatuista lähetystä ja suuren latauskapasiteetin vastaanottajat saattavat olla tyytymättömiä matalalaatuisen lähetykseen (J. Liu ym. 2008). Näitä ongelmia on pyritty ratkaisemaan esimerkiksi käyttämällä erilaisia videon koodaustekniikoita, kuten Scalable Video Coding (SVC) ja Multiple Description Coding (MDC). Zhang ja Hassanein (2012) mu-

kaan näissä monikerroksisissa koodaustekniikoissa (engl. *multiple layer coding*) video jaetaan kerroksiin. Video voidaan toistaa, vaikka osa kerroksista puuttuisi, mutta se heikentää kuvan laatua (Zhang ja Hassanein 2012). SVC jakaa videon pohjakerrokseen ja useampaan ehostuskerrokseen. Pohjakerros voidaan purkaa itsekseen, ja se tarjoaa karkean kuvanlaadun. Pohjakerroksen kuvanlaatua voidaan parantaa ehostuskerroksien avulla, mutta niitä ei voi purkaa ilman pohjakerrosta. MDC jakaa videon useampaan tasa-arvoiseen kerrokseen, jotka pystytään purkamaan itsekseen karkealla kuvanlaadulla. Kuvanlaatu paranee, mitä useampi kerros on saatavilla. (Xu ym. 2012) MDC:llä on kuitenkin merkittävä tehokkuuskustannus, eikä sitä usein hyödynnetä reaali-aikaisessa maailmassa (Zhang ja Hassanein 2012). Nämä koodaustekniikat mahdollistavat useamman lähetyslaadun tukemisen yhtäaikaaisesti.

Vertaisten lähetyksen kapasiteetti saattaa poiketa myös huomattavasti niiden latauskapasiteetista. Edelleenlähetyksmäärät on helppo suhteuttaa latausmääriin. Vertaiset edelleenlähettävät vähemmän, yhtä paljon tai enemmän kuin ne lataavat. Hei ym. (2007) käyttävät näistä vertaisista termejä *sinks*, *forwarders* ja *amplifiers*.

Hei ym. (2007) mukaan jotkut vertaiset eivät tarjoa lähes yhtään edelleenlähetyksen kapasiteettia. Zhang ja Hassanein (2012) mukaan jotkut vertaiset saattavat olla muun verkon saavuttamattomissa, koska ne ovat palomuurin tai NAT:in (Network Address Translator) takana. Tämän vuoksi p2p-suoratoiston olisi hyödynnettävä NAT traversal -tekniikoita hyödyntääkseen isompaa osaa verkosta. Kuitenkin p2p-suoratoistossa mikään tietty yhteys ei ole välttämätön, kuten esimerkiksi p2p-konferenssisovelluksissa tai p2p-yhteyksiä hyödyntävissä videopeleissä.

Suoratoiston luonteen vuoksi suuren kapasiteetin vertaisilla latausmäärillä ja lähetyksmäärillä saattaa olla suurikin ero. Hei ym. (2007) mukaan keskimääräiset latausmäärät seuraavat vahvasti toistoon vaadittavia latausmääriä, toisin kuin BitTorrentissa latausmäärät eivät kasva latauskapasiteetin kanssa mahdollisimman suureksi. Alkuperäisen lähettäjän lähetyksen nopeus asettaa teoreettisen maksimin vertaisten latausmäärälle, jos ei oteta huomioon potentiaalista redundanssia. Tätä vastoin edelleenlähetyksmäärillä ei ole minkäänlaista ylärajaa, joten vertainen voi palvella niin montaa vertaista, kuin sillä riittää resursseja ja lähetyksen kapasiteettia. Hei ym. (2007) mukaan suuren edelleenlähetyksen kapasiteetin vertaisilla edelleenlähetyksmäärät voivat olla valtavasti suurempia kuin latausmäärät. Esimerkkinä he antavat vertaisen, jonka

edelleenlähetysmäärät olivat kymmenenkertaiset latausmääriin verrattuna. Shen ym. (2011) mukaan vertaiset, jotka pysyvät systeemissä kauemmin ja joilla on korkea edelleenlähetyksen kapasiteetti, ovat toivottavia vertaisverkolle, koska sellaiset vertaiset parantavat verkon skaalautuvuutta. Insentiivimekanismit saattavat rohkaista vertaisia tarjoamaan runsaammin edelleenlähetyksen kapasiteettiaan.

6.3 Flash Crowd

Flash crowd on ilmiö, jossa suuri määrä vertaisia liittyy systeemiin hyvin lyhyen ajan sisällä (J. Liu ym. 2008). Tämä on haastavaa p2p-suoratoistolle, koska sen täytyy nopeasti liittää uusia vertaisia vertaisverkkoon merkittävästi uhraamatta vanhojen tai uusien vertaisten videon laatua (J. Liu ym. 2008). Flash crowd on merkittävä ongelma nimenomaan suoratoiston näkökulmasta. Muunkaltaiset p2p-sovellukset eivät koe samankaltaista vertaisten liittymiskäyttäytymistä tai ovat vähemmän herkkiä flash crowdin aiheuttamille häiriöille (Bo Li ym. 2008). Kuten osiossa 2.1 mainittiin, vertaiset liittyvät ja poistuvat lähetyksestä vahvasti sisällön mukaan. Esimerkiksi ohjelmien alut kokevat potentiaalisesti suuria liittymisnopeuksia. Voi tapahtua myös flash crowdia vastakkainen ilmiö, missä valtava määrä vertaisia poistuu, jolloin systeemin pitää korjata verkko nopeasti minimoidakseen lähetyksen häiriöt (J. Liu ym. 2008).

Flash crowd aiheuttaa verkolle häiriöitä pääasiassa kahdella tavalla: huonontamalla verkon rakennetta lisäämällä siihen vertaisia, jotka eivät heti kykene osallistumaan jakoprosessiin ja aiheuttamalla resurssikilpailua vertaisten kesken. Bo Li ym. (2008) mukaan uudet vertaiset kykenevät vain kuluttamaan muiden vertaisten edelleenlähetyksen kapasiteettia, koska niillä ei ole vielä riittävästi videosisältöä, jota jakaa muille vertaisille. Tämä johtaa siihen, että vertaiset saattavat joutua kilpailemaan verkon resursseista keskenään (Bo Li ym. 2008). Flash crowdin tapahtuessa joukko uusia vertaisia saattaa vaatia enemmän edelleenlähetyksen kapasiteettia kuin mitä vanhoilla vertaisilla on tarjota, mutta eivät kykene vielä käyttämään toistensa edelleenlähetyksen kapasiteettia, koska niillä ei ole tarpeeksi edelleenlähettävää videota. Myös F. Liu ym. (2012) mukaan uusien vertaisten vaikeus kerätä tarpeeksi kaistaa on flash crowdin keskeisin ongelma.

Resurssien puute saattaa johtaa yhteyksien muodostamisen epäonnistumisiin vertaisten välillä, riittämättömään vanhempisolmujen edelleenlähetyskapasiteettiin tai hyvin pitkiin aloitusviiveisiin (Bo Li ym. 2008). Nämä häiriöt heijastuvat käyttäjiin palvelun laadun heikkenemisenä tai lähetykseen liittymisen epäonnistumisena. Flash crowd lisää vertaisen liittymisen epäonnistumisen todennäköisyyttä merkittävästi (Bo Li ym. 2007). Bo Li ym. (2008) mukaan riittämätön lähetyksen laatu johtaa joko käyttäjän lähtemiseen tai yritykseen liittyä systeemiin uudelleen. Systeemin näkökulmasta uudelleenliittyvät vertaiset lisäävät liittymisnopeutta (Bo Li ym. 2008).

Flash crowdin aiheuttamat häiriöt saattavat luoda uusia häiriöitä. Baochun Li ym. (2013) mukaan vertainen saattaa poistua systeemistä, jos videon laatu ei ole tarpeeksi hyvä. Bo Li ym. (2008) mukaan lyhyet sessiot ja vertaisten uudelleenliittymiset lisääntyvät merkittävästi flash crowdin aikana. Lyhyet sessiot ja vertaisten runsas poistuminen lisäävät vertaisten vaihtuvuuden määrää, näin aiheuttaen lisää häiriöitä vertaisverkolle.

Yleensä käytännössä flash crowdin vaikutuksia p2p-suoratoistojärjestelmissä pyritään hillitsemään lisäämällä palvelinkapasiteettia tai hyödyntämällä Content Delivery Netwerkeja (F. Liu ym. 2012). Yin ym. (2009) mukaan CDN-p2p-hybridisysteemi pystyi ylläpitämään hyvää palvelunlaatua myös flash crowdin aikana. Kuitenkin F. Liu ym. (2012) arvioivat, että suuri palvelinkapasiteetti on tarpeellista vain alkupään kasvuvaiheessa ja p2p-systeemi kykenee ylläpitämään itseään, kun tarpeeksi suuri koko on saavutettu. Lisäksi heidän mukaansa lisätty palvelinkapasiteetti parantaa systeemin skaalautumiskykyä, mutta hyödyt pienenevät serverikapasiteetin lisääntyessä edelleen.

F. Liu ym. (2012) mukaan pelkkä järjestelmän suuri edelleenlähetyskapasiteetti ei riitä varmistamaan hyvää skaalautumista ja tehokkuutta p2p-suoratoistosysteemeissä, koska sopivien vertaisten löytyminen vie aikaa ja vertaiset joutuvat kilpailemaan keskenään. Vertaisverkko skaalautuu sitä paremmin, mitä enemmän olemassa olevia vertaisia on verrattuna uusiin vertaisiin. Siksi systeemi kasvaa nopeammin flash crowdin aikana, kun se on onnistunut kasvamaan jo jonkin verran. (F. Liu ym. 2012)

F. Liu ym. (2012) mukaan flash crowdin vaikutuksia voidaan lieventää tietoisesti rajoittamalla vertaisten liittymisnopeutta. Heidän mukaansa vertaisten liittymisnopeuden kasvaes-

sa niin suureksi, että verkon suorituskyky alkaa kärsiä, osan liittyvistä vertaisista liittymistä voidaan viivyttää. Heidän mukaansa verkko kasvaa nopeimmin, kun liittyvien vertaisten suhde on sopiva vanhoihin vertaisiin. Suhteen ylittyessä verkon kasvunopeus hidastuu merkittävästi. Hidastettujenkin vertaisten liittymisnopeus kompensoituu verkon paremman kasvunopeuden kautta. (F. Liu ym. 2012).

7 Yhteenveto

P2p-suoratoiston keskeiset teknologiset haasteet liittyvät vahvasti vertaisiin liittyvään ennalta-arvaamattomuuteen. Vertaiset saattavat poiketa toisistaan ominaisuuksiltaan, kuten edelleenlähetyskapasiteetin osalta, ennalta-arvaamattomasti. Yksittäisten vertaisten käyttäytymistä ei myöskään pystytä ennustamaan ennakkoon. Vertaisia voi liittyä tai poistua milloin vain ilman varoitusta. Nämä ennalta-arvaamattomuuden elementit aiheuttavat häiriöitä vertaisverkolle, jonka toiminta perustuu siihen, että suuri määrä vertaisia toimii yhdessä. Tämä yhteistoiminnan periaate myös heikentää tehokkuutta jonkin verran yksittäisen käyttäjän näkökulmasta. Esimerkiksi vertaisverkkoon liittyminen on hitaampaa kuin yhteyden muodostaminen yksittäiseen palvelimeen, ja datan saaminen vertaisten kautta on hitaampaa kuin palvelimelta suoraan. Nämä kompromissit kuitenkin mahdollistavat pienemmän resurssien kulutuksen palveluntarjoajalle.

P2p-suoratoisto kohtaa kuitenkin muitakin ongelmia, jotka eivät suoraan liity teknisiin suorituskyvyn ongelmiin. Esimerkiksi Guarnieri ym. (2021) ottavat esille Brasilian kalliit mobiiliyhteydet ja niiden tiukat latausrajoitukset. Tällaiset ISP:den asettamat rajoitukset saattavat merkittävästikin haitata suoratoistoverkkkojen toimintaa ja vuorovaikuttaa negatiivisesti vertaisverkkojen teknisten haasteiden kanssa.

Baochun Li ym. (2013) mukaan vertaisverkkoihin perustuvan suoratoiston suosio kaupallisissa tuotteissa ja tutkimuksessa on syrjäytynyt pilvipohjaisten teknologioiden tieltä. Pilvipohjaiset ratkaisut ovat Netflixin VOD-palveluiden tarjoajille taloudellisempia ja kykenevät käsittelemään äkillisetkin palveluntarpeen vaihtelut (Baochun Li ym. 2013). P2p-suoratoiston sijaan joutuu mahdollisesti ponnistelemaan pärjätäkseen nopeasti vaihtelevan palveluntarpeen kanssa.

Van der Sar (2014) mukaan Spotify, joka oli aikanaan suurimpia p2p-verkkoja internetissä, siirtyi pois p2p-teknologioista. Tämän perusteella voidaan arvella, että p2p-teknologiat eivät välttämättä ole laatu- ja kustannustehokkuudeltaan paras ratkaisu vakiintuneille kaupallisille sovelluksille kuten Spotify. Tästä saattaa nousta kysymys, että mikä on vertaisverkkojen käyttötapaus nykyään. Kuitenkin Van der Sar (2014) mukaan vertaisverkkojen hyödyntämi-

nen salli Spotifyn alkutaipaleellaan vähentää palvelinkustannuksia. Vertaisverkot saattavat olla siis hyödyllisimpiä sellaisille palveluntarjoajille, joilla on rajatusti resursseja käytössään. Esimerkkejä tällaisista palveluntarjoajista ovat esimerkiksi tuoreet kaupalliset toimijat, ei-kaupalliset toimijat ja hajautetun sosiaalisen verkon palveluntarjoajat. Hajautetun tai federoidun sosiaalisen verkon palvelut tuotetaan useamman palveluntarjoajan toimesta, jotka voivat kommunikoida keskenään, samaan tapaan kuin sähköpostin palveluntarjoajat (“What is Mastodon?”, n.d.). Koska federoituneiden sosiaalisten verkkojen ympäristössä kuka tahansa voi olla palveluntarjoaja, saattaa ympäristöstä löytyä hyvin pientenkin resurssien toimijoita. Vertaisverkkoihin perustuvien suoratoistotekniikoiden hyödyntäminen voisi antaa melko pienillekin toimijoille mahdollisuuden tarjota suoratoistopalveluita.

Jatkotutkimuksissa olisi hyvä paikantaa p2p-suoratoiston ongelmia, jotka eivät suoraan liity teknisiin haasteisiin. Viimeaikaista tutkimusta ei ole kovin paljon, ja internet sekä siihen liitetyt laitteet ovat muuttuneet merkittävästi viimeisen kymmenen vuoden aikana. Myös vertaisverkkojen tietoturvanäkökulmaa olisi hyvä käsitellä viimeaikaisesta perspektiivistä. Lisäksi käyttötarkoitusten, joissa p2p-suoratoistolla on etuja pilvipohjaisiin ratkaisuihin verrattuna, kartoittaminen olisi mielekästä jatkotutkimuksissa, koska vertaisverkot ovat jääneet pilven varjoon viime vuosina.

Lähteet

- Banerjee, Suman, Bobby Bhattacharjee ja Christopher Kommareddy. 2002. “Scalable Application Layer Multicast”. Teoksessa *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 205–217. SIGCOMM '02. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery. ISBN: 158113570X. <https://doi.org/10.1145/633025.633045>.
- Bideh, Minoo Kargar, Behzad Akbari ja Abdollah Ghaffari Sheshjavani. 2016. “Adaptive content-and-deadline aware chunk scheduling in mesh-based P2P video streaming”. *Peer-to-Peer Networking and Applications* 9, numero 2 (maaliskuu): 436–448. ISSN: 1936-6442, 1936-6450. <https://doi.org/10.1007/s12083-015-0355-x>.
- Chu, Yang-hua, S.G. Rao, S. Seshan ja Hui Zhang. 2002. “A case for end system multicast”. *IEEE Journal on Selected Areas in Communications* 20 (8): 1456–1471. <https://doi.org/10.1109/JSAC.2002.803066>.
- Deering, Dr. Steve E. 1989. *Host extensions for IP multicasting*. RFC 1112, 1112, elokuu. <https://doi.org/10.17487/RFC1112>. <https://rfc-editor.org/rfc/rfc1112.txt>.
- Deering, Stephen E., ja David R. Cheriton. 1990. “Multicast Routing in Datagram Internetworks and Extended LANs”. *ACM Trans. Comput. Syst.* (New York, NY, USA) 8, numero 2 (toukokuu): 85–110. ISSN: 0734-2071. <https://doi.org/10.1145/78952.78953>.
- Diot, C., B.N. Levine, B. Lyles, H. Kassem ja D. Balensiefen. 2000. “Deployment issues for the IP multicast service and architecture”. *IEEE Network* 14 (1): 78–88. <https://doi.org/10.1109/65.819174>.
- Gelman, A. D., H. Kobrinski, L. S. Smoot, S. B. Weinstein, M. Fortier ja D. Lemay. 1993. “A store-and-forward architecture for video-on-demand service”. *Canadian Journal of Electrical and Computer Engineering* 18 (1): 37–40. <https://doi.org/10.1109/CJECE.1993.6591636>.

- Ghaderzadeh, Abdulbaghi, Mehdi Kargahi ja Midia Reshadi. 2018. “ReDePoly: reducing delays in multi-channel P2P live streaming systems using distributed intelligence”. *Telecommunication Systems* 67, numero 2 (helmikuu): 231–246. ISSN: 1572-9451. <https://doi.org/10.1007/s11235-017-0336-x>.
- Guarnieri, Thiago, Idilio Drago, Ítalo Cunha, Breno Almeida, Jussara M. Almeida ja Alex B. Vieira. 2021. “Modeling large-scale live video streaming client behavior”. *Multimedia Systems* (huhtikuu). ISSN: 1432-1882. <https://doi.org/10.1007/s00530-021-00788-4>.
- Hei, Xiaojun, Chao Liang, Jian Liang, Yong Liu ja Keith W. Ross. 2007. “A Measurement Study of a Large-Scale P2P IPTV System”. *IEEE Transactions on Multimedia* 9 (8): 1672–1687. <https://doi.org/10.1109/TMM.2007.907451>.
- Li, B., S. Xie, Y. Qu, G. Y. Keung, C. Lin, J. Liu ja X. Zhang. 2008. “Inside the New Coolstreaming: Principles, Measurements and Performance Implications”. Teoksessa *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, 1031–1039. <https://doi.org/10.1109/INFOCOM.2008.157>.
- Li, Baochun, Zhi Wang, Jiangchuan Liu ja Wenwu Zhu. 2013. “Two Decades of Internet Video Streaming: A Retrospective View”. *ACM Trans. Multimedia Comput. Commun. Appl.* (New York, NY, USA) 9, numero 1s (lokakuu). ISSN: 1551-6857. <https://doi.org/10.1145/2505805>.
- Li, Bo, Gabriel Y. Keung, Susu Xie, Fangming Liu, Ye Sun ja Hao Yin. 2008. “An Empirical Study of Flash Crowd Dynamics in a P2P-Based Live Video Streaming System”. Teoksessa *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, 1–5. <https://doi.org/10.1109/GLOCOM.2008.ECP.339>.
- Li, Bo, Susu Xie, Gabriel Y. Keung, Jiangchuan Liu, Ion Stoica, Hui Zhang ja Xinyan Zhang. 2007. “An Empirical Study of the Coolstreaming+ System”. *IEEE Journal on Selected Areas in Communications* 25 (9): 1627–1639. <https://doi.org/10.1109/JSAC.2007.071203>.

- Li, Zhenyu, Gaogang Xie, Mohamed Ali Kaafar ja Kave Salamatian. 2015. “User Behavior Characterization of a Large-Scale Mobile Live Streaming System”. Teoksessa *Proceedings of the 24th International Conference on World Wide Web*, 307–313. WWW '15 Companion. Florence, Italy: Association for Computing Machinery. ISBN: 9781450334730. <https://doi.org/10.1145/2740908.2743054>.
- Liu, Fangming, Bo Li, Lili Zhong, Baochun Li, Hai Jin ja Xiaofei Liao. 2012. “Flash Crowd in P2P Live Streaming Systems: Fundamental Characteristics and Design Implications”. *IEEE Transactions on Parallel and Distributed Systems* 23 (7): 1227–1239. <https://doi.org/10.1109/TPDS.2011.283>.
- Liu, Jiangchuan, Sanjay G. Rao, Bo Li ja Hui Zhang. 2008. “Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast”. *Proceedings of the IEEE* 96 (1): 11–24. <https://doi.org/10.1109/JPROC.2007.909921>.
- Liu, Yong. 2010. “Delay Bounds of Chunk-Based Peer-to-Peer Video Streaming”. *IEEE/ACM Transactions on Networking* 18 (4): 1195–1206. <https://doi.org/10.1109/TNET.2009.2038155>.
- Liu, Yong, Yang Guo ja Chao Liang. 2008. “A survey on peer-to-peer video streaming systems”. *Peer-to-Peer Networking and Applications* 1, numero 1 (maaliskuu): 18–28. ISSN: 1936-6450. <https://doi.org/10.1007/s12083-007-0006-y>.
- Magharei, N., R. Rejaie ja Y. Guo. 2007. “Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches”. Teoksessa *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, 1424–1432. <https://doi.org/10.1109/INFCOM.2007.168>.
- Ramzan, Naeem, Hyunggon Park ja Ebroul Izquierdo. 2012. “Video streaming over P2P networks: Challenges and opportunities”. *ADVANCES IN 2D/3D VIDEO STREAMING OVER P2P NETWORKS, Signal Processing: Image Communication* 27 (5): 401–411. ISSN: 0923-5965. <https://doi.org/https://doi.org/10.1016/j.image.2012.02.004>.
- Schollmeier, R. 2001. “A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications”. Teoksessa *Proceedings First International Conference on Peer-to-Peer Computing*, 101–102. <https://doi.org/10.1109/P2P.2001.990434>.

Setton, Eric, Pierpaolo Baccichet ja Bernd Girod. 2008. "Peer-to-Peer Live Multicast: A Video Perspective". *Proceedings of the IEEE* 96 (1): 25–38. <https://doi.org/10.1109/JPROC.2007.909925>.

Setton, Eric, Jeonghun Noh ja Bernd Girod. 2006. "Low Latency Video Streaming Over Peer-To-Peer Networks". Teoksessa *2006 IEEE International Conference on Multimedia and Expo*, 569–572. <https://doi.org/10.1109/ICME.2006.262472>.

Shen, Zhijie, Jun Luo, Roger Zimmermann ja Athanasios V. Vasilakos. 2011. "Peer-to-Peer Media Streaming: Insights and New Developments". *Proceedings of the IEEE* 99 (12): 2089–2109. <https://doi.org/10.1109/JPROC.2011.2165330>.

Stutzbach, Daniel, ja Reza Rejaie. 2006. "Understanding Churn in Peer-to-Peer Networks". Teoksessa *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, 189–202. IMC '06. Rio de Janeiro, Brazil: Association for Computing Machinery. ISBN: 1595935614. <https://doi.org/10.1145/1177080.1177105>.

Van der Sar, Ernesto. 2014. "Spotify Starts Shutting Down Its Massive P2P Network". Viitattu 26. helmikuuta 2022. <https://torrentfreak.com/spotify-starts-shutting-down-its-massive-p2p-network-140416/>.

"What is Mastodon?" n.d. Viitattu 2. maaliskuuta 2022. <https://docs.joinmastodon.org/>.

Wu, Chuan, Baochun Li ja Shuqiao Zhao. 2012. "Diagnosing Network-Wide P2P Live Streaming Inefficiencies". *ACM Trans. Multimedia Comput. Commun. Appl.* (New York, NY, USA) 8, numero 1S (helmikuu). ISSN: 1551-6857. <https://doi.org/10.1145/2089085.2089090>.

Xu, Yuanyuan, Ce Zhu, Wenjun Zeng ja Xue Jun Li. 2012. "Multiple description coded video streaming in peer-to-peer networks". *ADVANCES IN 2D/3D VIDEO STREAMING OVER P2P NETWORKS, Signal Processing: Image Communication* 27 (5): 412–429. ISSN: 0923-5965. <https://doi.org/https://doi.org/10.1016/j.image.2012.02.005>.

Yin, Hao, Xuening Liu, Tongyu Zhan, Vyas Sekar, Feng Qiu, Chuang Lin, Hui Zhang ja Bo Li. 2009. “Design and Deployment of a Hybrid CDN-P2P System for Live Video Streaming: Experiences with LiveSky”. Teoksessa *Proceedings of the 17th ACM International Conference on Multimedia*, 25–34. MM '09. Beijing, China: Association for Computing Machinery. ISBN: 9781605586083. <https://doi.org/10.1145/1631272.1631279>.

Zhang, Xiangyang, ja Hossam Hassanein. 2012. “A survey of peer-to-peer live video streaming schemes – An algorithmic perspective”. *Computer Networks* 56, numero 15 (lokakuu): 3548–3579. ISSN: 1389-1286. <https://doi.org/10.1016/j.comnet.2012.06.013>.

Zhang, Xinyan, Jiangchuan Liu, Bo Li ja Y-SP Yum. 2005. “CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming”. Teoksessa *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*. 3:2102–2111. IEEE.