

hyväksymispäivä arvosana

arvostelija

Videoiden suoratoisto nyt ja lähitulevaisuudessa

Mikael Parvamo

Helsinki 05.05.2020

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen osasto

Tiedekunta/Osasto – Fakultet/Sektion – Faculty/Section		Koulutusohjelma – Studieprogram – Study Programme	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen maisteriohjelma	
Tekijä – Författare – Author			
Mikael Parvamo			
Työn nimi – Arbetets titel – Title			
Videoiden suoratoisto nyt ja lähitulevaisuudessa			
Ohjaajat – Handledare – Supervisors			
Jussi Kangasharju			
Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages	
	05.05.2020	64	
Tiivistelmä – Referat – Abstract			
<p>Tänä päivänä videoiden suoratoisto tuottaa suurimman osan Internetin IP-liikenteestä eikä tämän osuuden kasvulle näy pysähtymistä. Palveluiden kehittyessä uusimpien teknologioiden myötä, on myös käyttökokemuksen parantaminen noussut uuteen arvoon. Tämän edistämiseksi videoiden suoratoistossa etsitään jatkuvasti kehityskkeinoja mm. videon laadun parantamiseksi ja puskurointiviiveiden välttämiseksi.</p> <p>Suurimmat suoratoistopalveluiden tarjoajat ovat siirtyneet käyttämään, kehittyneitä, mukautuvia suoratoistomenetelmiä, joiden avulla mukautua dynaamisesti verkon muuttuviin olosuhteisiin ja käyttäjien tekemiin valintoihin. Näistä yleisimmin käytetty menetelmä on MPEG-DASH. Suoratoiston reaaliaikaisesta luonteesta johtuen myös virhetilanteita tulee välttää ja näistä palautuminen tulee tapahtua sekä nopeasti että tehokkaasti, sillä uudelleenlähetyksille harvoin on aikaa. Vaikka mukautuvat suoratoistomenetelmät lievittävätkin verkon ruuhkautumista, jatkuu myös verkon rasittuminen ennen näkemättömällä tavalla.</p> <p>Suoratoistopalveluiden käyttäjämäärien kohotessa satoihin miljooniin ja keskimääräisten katselutuntien noustessa niiden myötä, on syytä tarkastella, millaisista elementeistä suoratoiston tuottama verkkoliikenne muodostuu. Tämän tutkielman tarkoituksena onkin havainnollistaa kehityskaarta, joka on johtanut tänä päivänä vallitsevien suoratoistomenetelmien muovautumiseen ja tuoda esille sekä yhteiskunnallisia että teknologisia seikkoja, joiden vaikutus videoiden suoratoistossa tulee näkymään lähitulevaisuudessa. Tämän lisäksi tutkielmassa esitellään tämän hetkisiä suoratoistomenetelmiä sekä niiden toimintaa ja mukautumiskykyä. Näistä menetelmistä keskiöön on nostettu MPEG-DASH.</p> <p>The ACM Computing Classification System (CCS): Information systems → Information systems applications → Multimedia information → Multimedia streaming</p>			
Avainsanat – Nyckelord – Keywords			
Suoratoisto, mobiili-liikenne, suoratoistomenetelmät, mukautuminen, HTTP-suoratoisto, DASH			
Säilytyspaikka – Förvaringställe – Where deposited			
Muita tietoja – Övriga uppgifter – Additional information			
Erikoistumislinja / opintosuunta / seminaarin nimi / yms.			
Hajautetut järjestelmät ja tietoliikenne			

Sisältö

1 Johdanto	1
2 Suoratoistopalveluiden kulutus	4
2.1 Suoratoisto yleisesti	4
2.2 Johtavat videoiden suoratoistopalvelut	6
2.3 Videosuoratoiston käyttökokemus	8
2.4 Mobiilikulutus	9
3 Yleisimmät suoratoistomenetelmät ennen	11
3.1 Kuljetuskerroksen protokollat: TCP ja UDP	11
3.2 Ryhmälähetys (Multicasting)	14
3.3 Suoratoisto ennen	16
3.4 Sisällönjakeluverkko	18
4 HTTP-suoratoisto ja dynaaminen mukautuminen	20
4.1 HTTP-suoratoiston toimintaperiaatteet ja infrastruktuuri	20
4.2 Dynaamisesti mukautuvat suoratoistomenetelmät	22
4.3 MPEG-DASH	26
4.4 MPD-tiedosto	29
4.5 Mukautuminen muuttuviin olosuhteisiin	31
4.6 Virheiden hallinta ja palautuminen	35
5 Reaaliaikainen suoratoisto	39
5.1 Palvelumallin ero	39
5.2 Tekniset käytänteet ja käytännön toteutukset	41
6 Suoratoisto lähitulevaisuudessa	45
6.1 Lähitulevaisuuden näkymät	46
6.2 Reunalaskenta-avusteinen videoiden suoratoisto	48
6.3 Yhteiskunnan asettamat rajoitukset	52
8 Yhteenveto	56
9 Lähteet	58

1 Johdanto

Elämme tällä hetkellä videomateriaalin kulutuksen kannalta “kuuminta” aikaa. Viime vuosina videoiden toistaminen verkon välityksellä on noussut aivan uudelle tasolle. Tämän seurauksena Internet on kokenut ennennäkemätöntä räsitystä, jonka kasvulle ei näy loppua. Uutta videomateriaalia työnnetään verkkoon jatkuvasti ympäri maapalloa ja sen kulutusolosuhteet ovat äärimmäisen heterogeeniset. Alan suurimmat palveluntarjoajat, kuten Google ja Amazon, ovat levittäneet palvelinkeskuksiaan ympäri maapalloa ja pilvi on laskeutunut käyttäjien äärelle [Amazon 2020] [Google 2020]. Myös sosiaalinen media on noussut suureen asemaan videoiden suoratoiston kannalta. Miten tähän tilanteeseen päädyttiin ja kuinka kehitys tulee jatkumaan lähitulevaisuudessa?

Videoiden suoratoisto on palvelumuoto, jonka avulla käyttäjä voi toistaa videota lataamatta sitä ensin kokonaisuudessaan laitteelle, josta toistettavaa sisältöä katsoo. Ciscon tekemän ennusteen mukaan videomateriaalin tuottama liikenne tulee kattamaan n.82% kaikesta IP-liikenteestä vuoteen 2022 mennessä [Cisco 2019]. Kasvava tiedonsiirtomäärä tulee väistämättä asettamaan uusia haasteita ja rasitteita verkolle. Tämän lisäksi myös videoiden kulutustavat ovat murroksessa, minkä vuoksi videoiden suoratoisto joutuu vastaamaan vaikeisiin vaatimuksiin.

Suoratoistopalveluiden käyttäjät ovat yhä enenevässä määrin siirtyneet käyttämään palveluita mobiileilla alustoilla. Tämän seurauksena on noussut uusia haasteita videoliikenteen ohjaamiselle ja käyttäjien tavoittamiselle korkeasta palvelun laadusta tinkimättä. Myös käyttäjien laitteistokohtaiset eroavaisuudet asettavat palveluntarjoajille ehtoja käytettävien suoratoistomenetelmien puolesta.

90-luvun alussa kehittynyt videoiden levittäminen verkon välityksellä onkin siis kokenut huiman kehityksen ensiaskelistaan aina asteittaisesta latauksesta (progressive download) lähtien. Kehityksen aikana väistyneet protokollat ja ratkaisumallit ovat luoneet pohjan nykyisille malleille. Näiden kehityspisteiden seurauksena voimme tänä päivänä nauttia videoiden suoratoistosta lähes missä vain ja milloin vain.

Suurimmat suoratoistopalvelut ovat siirtyneet hyödyntämään mukautuvia suoratoistomenetelmiä, joiden avulla voidaan mukautua verkon muuttuviin olosuhteisiin tai käyttäjän tekemiin valintoihin suoratoiston aikana [Spiteri 2018]. Näihin muutoksiin

mukautuminen ja suoratoiston säätely niiden mukaisesti mahdollistaa korkeamman palvelun laadun käyttäjille.

Mukautuvia suoratoistomenetelmiä on lukuisia, mutta tällä hetkellä suosituin ja käytetyin niistä on MPEG-DASH (DASH). DASH on ISO:n hyväksymä suoratoistostandardi, jonka tuotannon ja ylläpidon taustalla oleva DASH-IF (DASH International Forum) pitää sisällään alan suurimpia tekijöitä, kuten Google, Akamai, Microsoft ja Netflix [DASH 2019]. Dynaamisen mukautumiskyvyn ja vankan kehittämistuen seurauksena DASH on omaksuttu moniin suurimmista suoratoistopalveluista.

Suoratoistopalveluiden kasvattaessa suosiotaan, myös kasvaneen käyttäjä- ja videomateriaalimäärän hallinta on tuonut mukanaan kustannuksia suoratoistopalveluiden tarjoajille. Resurssien kasvattaminen mm. palvelinkeskuksien lisäämisellä ja infrastruktuurin laajentamisella ovat omalta osaltaan lieventäneet verkon ruuhkautumista monin paikoin. Toisaalta, räjähdysmäisesti kasvanut videomateriaalin liikennöinti on herättänyt myös yhteiskunnan asettamaan rajoituksia verkkoliikennettä kohtaan. Monin paikoin myös verkko-operaattorit ovat alkaneet rajoittamaan kaistan käyttöä tietyiltä toimijoilta [Li 2019].

Suoratoisto elää teknologian suurimpien muutoksien mukana. Tämän seurauksena uudet ja modernit teknologiat, kuten 5G, ja uudet paketoitistandardit sekä esimerkiksi reunalaskennan hyödyntäminen tulevat näyttämään tien tulevaisuuden suoratoistoon. Väistyvätkö alan vallitsevat standardit uusien menetelmien edestä, vai kehittyvätkö ne entisestään nykyisellä pohjalla?

Tämän tutkielman tavoitteena on esitellä videoiden suoratoistoa yleiseltä kannalta. “Mistä tulemme? Missä olemme nyt?” ja “Minne olemme matkalla?”. Näiden kysymyksien saattelemana tutkielmassa pyritään havainnollistamaan videoiden suoratoiston kehityskaarta sekä esittelemään tällä hetkellä pinnalla olevat suoratoistomenetelmät ja niiden toimintamallit. Tutkielmassa pyritään myös esittämään tulevaisuuden kannalta tärkeitä näkökulmia suoratoiston kasvuun mm. yhteiskunnan asettamien rajoitteiden kautta.

Tutkielman toisessa luvussa perehdytään suoratoistoon yleisellä tasolla ja tutustutaan tämän hetken suurimpiin suoratoistopalveluihin. Näistä tarkasteluun nostetaan YouTube ja Netflix, jotka edustavat suurinta osaa videosuoratoiston liikennöinnistä. Luvussa esitellään myös suoratoiston käyttökokemukseen vaikuttavia tekijöitä ja havainnollistetaan muuttuneita kulutuskäytäntöjä mobiilikulutuksen näkökulmasta.

Tutkielman kolmannessa luvussa esitetään suoratoiston kannalta tärkeitä rakennuspalikoita, kuten kuljetuskerroksen protokollia ja ryhmälähetystä (multicasting). Tämän lisäksi luvussa esitellään suoratoiston kehityskaarta ja merkittävimpiä vaiheita 90-luvulta tähän päivään. Luvun lopussa perehdytään myös sisällönjakeluverkkojen tuomaan hyötyyn suoratoiston kannalta.

Neljännessä luvussa syvennyttään HTTP-suoratoistoon ja tästä kehittyneisiin mukautuviin suoratoistomenetelmiin. Mukautuvista suoratoistomenetelmistä käsitellään tarkemmin MPEG-DASH-nimistä (DASH) suoratoistostandardia, joka on noussut alan vallitsevaksi suoratoistomenetelmäksi. Luvussa käydään myös läpi mukautuvan suoratoistoprosessin etenemistä ja muuttuviin olosuhteisiin reagoimista DASH:n avulla. Luvun lopussa käsitellään muuttuvista olosuhteista johtuneiden virhetilojen hallintaa ja näistä palautumista.

Viides luku pyrkii havainnollistamaan reaaliaikaisen suoratoiston eroa tavalliseen VOD-suoratoistoon (Video On Demand). Luvussa käydään läpi palvelumallien selkeimmät erot ja esitellään, miten nämä eroavaisuudet ovat johtaneet erilaisiin teknisiin käytänteisiin reaaliaikaisessa suoratoistossa. Luvussa perehdytään lisäksi etenkin e-urheilun saralla suureksi nimeksi nousseen Twitch-palvelun toimintamalliin.

Viimeisessä luvussa käydään läpi suoratoiston lähitulevaisuuden näkymiä. Luvussa esitellään uusia teknologioita, joiden vaikutukset tulevat heijastumaan suoratoistopalveluiden käyttökokemukseen ja käytännön toteutuksiin. Tämän lisäksi luvussa tarkastellaan reunalaskennan mahdollista hyödyntämistä videoiden suoratoistossa ja pohditaan yhteiskunnan asettamien, verkon liikennöintiin kohdistuneiden, rajoitteiden vaikutusta suoratoistopalveluiden toimintaan. Näiden tarkasteltavien osa-alueiden avulla pyritään selvittämään, mihin suuntaan videoiden suoratoisto on kehittymässä tulevina vuosina.

2 Suoratoistopalveluiden kulutus

Suoratoistopalveluiden käyttö on ollut räjähdysmäisessä kasvussa viimevuosien aikana. Videosisällön kulutus on ollut suuressa murroksessa, kun kulutus on siirtynyt yhä enenevässä määrin mobiilialustoille. Tämä muutos on tuonut myös mukanaan uusia haasteita, joihin alan suurimmat palveluntuottajat ovat joutuneet vastaamaan käyttökokemuksen korkean tason ylläpitämiseksi.

Tässä kappaleessa esitetään suoratoiston toimintaperiaatteet yleisellä tasolla ja pyritään havainnollistamaan ongelmakohtia, joita modernit suoratoistopalvelut joutuvat haastamaan tällä hetkellä. Tämän lisäksi kappaleessa pureudutaan käyttökokemukseen ja siihen vaikuttaviin tekijöihin mm. mobiilin suoratoistokulutuksen näkökulmasta.

2.1 Suoratoisto yleisesti

Internetin kulutus on alkanut enenevässä määrin painottumaan mediasisällön siirtoon. Ciscon tekemän ennusteen mukaan videoiden siirto tulee kattamaan jopa 82% kaikesta IP-liikenteestä vuonna 2022 [Cisco 2019]. Videoiden tuotannon ja kulutuksen kasvu tulee asettamaan verkolle ennennäkemättömiä haasteita. Kuormituksen ja kulutuksen kasvussa verkossa, jota ei ole luotu kuljettamaan näin suuren kokoluokan tiedostoja pitkiä matkoja [Vakali 2003], on tärkeää tietää, mistä tämä kuormitus on pääasiassa peräisin.

Videoiden suoratoisto on nimensä mukaisesti halutun videosisällön suoraa toistamista. Suoratoisto mahdollistaa siis videosisällön toiston ennen kuin tiedostoa on kokonaisuudessaan ladattu verkosta. Käyttäjällä on mahdollisuus aloittaa videon katselu haluttuna hetkenä ja edetä katselussa ilman tahattomia viiveitä, ikään kuin videosisältöä suoritettaisiin suoraan käyttäjän omalta päätteeltä.

Suoratoistaminen rakentuu puskuroinnin (buffering) sekä videon latauksen ja toiston optimaaliselle suhteelle ja sen ylläpitämiselle vakaan tilan vaiheessa (steady state phase) [Rao 2011]. Puskurointivaiheessa videosoitin lataa toistettavaa videota puskuriin ennalta määritellyn määrän. Puskurin täytyttyä videon toisto voidaan käynnistää. Puskuriin kerätty video mahdollistaa videon samanaikaisen toistamisen ja uuden sisällön lataamisen.

Vakaan tilan vaiheessa uutta videosisältöä pyritään tuomaan puskuriin sen verran, että puskuri ei koskaan tyhjenisi toiston jatkuessa. Tämän vuoksi latauksen suhde toistoon

tulisi olla vähintään 1:1. Mikäli videon samanaikainen lataus on pienempi kuin puskurin tyhjenemisvauhti, tulee videon toisto katkeamaan uuden puskuroinnin ajaksi. Tällainen tahaton toiston keskeytyminen ei ole tavoiteltavaa, koska sillä on negatiivinen vaikutus palvelun käyttökokemukseen [Mok 2011]. Toisaalta, suhdeluvun nostaminen liian korkeaksi, eli lataamalla liikaa toistettavaa videota puskuriin, ei myöskään ole suotavaa.

Suoratoiston yhteydessä on muistettava, että toistettavan laitteiston muisti on rajallinen ja käyttäjä voi myös itse kontrolloida sisällön toistoa. On hyvin mahdollista, että käyttäjä aloittaa elokuvan katselun puhelimellaan, mutta päättääkin jättää elokuvan kesken, sillä elokuva ei olekaan hänen mieleisensä. Jos elokuva on ladattu kokonaisuudessaan käyttäjän laitteelle, on tämä aiheuttanut ylimääräistä, turhaa, kuormitusta verkolle. Tämän vuoksi olisikin syytä löytää sopiva suhdeluku latauksen ja toiston välille.

TCP:tä hyödyntävissä suoratoistomenetelmissä vakaan tilan vaiheen keskiarvoinen lataustaso saavutetaan siirtämällä videosisältöä yksi osio (block) kerrallaan [Rao 2011]. Kun osio on siirretty, jää siirtoon käytettävä TCP-yhteys odottamaan seuraavan siirron alkamista. Suoratoistaminen luo siis ikään kuin syklin, jolloin siirron aikana käytettävä TCP-yhteys aktivoituu ja deaktivoituu vuoron perään. (ON-OFF cycle)

Suoratoistomenetelmät voidaan jakaa ryhmiin sen perusteella, onko niissä havaittavissa ON-OFF-syklejä ja kuinka pitkiä ON-OFF-syklejä ne hyödyntävät [Rao 2011]. Tämän perusteella voidaan erottaa selkeästi kolme pääryhmää suoratoistomenetelmistä; *ei ON-OFF-syklejä*, *lyhyet ON-OFF-syklit* ja *pitkät ON-OFF-syklit*. Jokaisessa tavassa on omat hyötynsä ja haittansa. Onkin siis syytä löytää jokaiseen tilanteeseen sopiva menetelmä.

Suoratoistomenetelmät, joissa ei ole havaittavissa ON-OFF-syklejä, voidaan nähdä tavallisina tiedostosiirtoina. Toistettava videosisältö lähetetään kokonaisuudessaan käyttäjälle yhtenä yhtätoistoisena jaksone puskurointivaiheessa. Tämän menetelmän hyöty on sen yksinkertaisuudessa. Palvelimen puolella ei tarvitse säädellä lähetyksen jaksottamista, vaan kaikki osiot voidaan lähettää suoraan käyttäjälle. Huonona puolena tässä menetelmässä on sen piittaamattomuus verkon turhasta kuormittamisesta. Käyttäjä voi, aikaisemman esimerkin tavoin, päättääkin olla katsomatta jo kokonaan ladattua videosisältöä, jolloin verkko on kokenut tarpeettoman kuormituksen.

Lyhyitä ja pitkiä ON-OFF-syklejä hyödyntävät suoratoistomenetelmät eroavat toisistaan ainoastaan siirrettävien osioiden suuruuksilla [Rao 2011]. Määritelmän mukaan alle 2,5Mt:n kokoisia osuuksia siirrettäessä puhutaan lyhyistä ON-OFF-sykleistä, kun taas

puolestaan tätä suuremmat osuudet johtavat pitkiin ON-OFF-sykleihin. Lyhyitä ON-OFF-syklejä hyödyntävien menetelmien hyödyt ovat niiden kyvyssä reagoida käyttäjän tekemiin muutoksiin. Esimerkiksi käyttäjän lopetettua videosisällön katselun, voidaan lähetyksen lopettaa siihen pisteeseen, johon puskurointia oli suoritettu. Verkkoa ei siis kuormiteta tarpeettomasti. Toisaalta, pienten osioiden lähettäminen johtaa suuremman metatietomäärän lähettämiseen, joka lisää yleiskustannuksia (overhead). Pitkät ON-OFF-syklit lisäävät puolestaan potentiaalista verkon kuormitusta, mutta vähentävät lähetyksistä aiheutuvia yleiskustannuksia. Tärkeätä onkin siis valita tarvittava suoratoistotekniikka tapauskohtaisesti.

Yhä useammat yleisesti käytettävät suoratoistomenetelmät ovat alkaneet painottaa dynaamista mukautumista käyttäjän tekemiin päätöksiin ja verkon ruuhkautumiseen. Mukautuvat suoratoistomenetelmät ovatkin nousseet alan vallitseviksi menetelmiksi. Näitä menetelmiä tarkastellaan läheisemmin tämän tutkielman neljännessä luvussa.

2.2 Johtavat videoiden suoratoistopalvelut

Suoratoistopalveluista on muodostunut oleellinen osa tavallisen Internetin käyttäjän arkea. Palvelut, kuten Netflix, YouTube, HULU, Prime Video, Viaplay ja Twitch, tuottavat jatkuvasti uutta katseltavaa palveluiden käyttäjille. Tunnetuimpien sarjojen ja elokuvien tarjoamisen lisäksi, suoratoistopalveluiden tarjoajat ovat alkaneet myös tuottamaan omaa mediasisältöä suoratoistolevitykseen sekä ostamaan eksklusiivisia oikeuksia suosituimpiin tietueisiin ja yksityisten henkilöiden tuottamiin sisältöihin. Tässä kappaleessa tutustumme lähemmin Netflixiin ja Googlen omistamaan YouTubeen.

Nykypäivänä on mahdotonta puhua videoiden suoratoistosta mainitsematta alan suurimpiin ja vaikutusvaltaisimpiin palveluihin lukeutuvaa Netflixiä. Netflix perustettiin Yhdysvalloissa vuonna 1997, verkossa toimivaksi, elokuvien vuokrauspalveluksi [Netflix 2020]. Vuonna 1998 pystytetty verkkosivusto, www.netflix.com, lisäsi palveluitaan 2000-luvulla mm. tarjoamalla käyttäjilleen suositusjärjestelmää, jonka avulla löytää uutta katseltavaa.

Netflix alkoi muodostua nykyiselleen vuonna 2007, jolloin se siirtyi varsinaisen suoratoiston pariin. Lukuisten yhteistyökumppaneiden lyöttäytyttyä mukaan Netflixin toimintaan, laajeni se myös muille mantereille, kuten Eurooppaan [Netflix 2020 (1)].

Lisääntyneiden tilaajien ja suosion seurauksena Netflix julkaisi omaa tuotantoa mm. House of Cards-, Arrested Development- ja Orange is the New Black -ohjelmien muodossa.

Tämän vuoden 2012 tapahtuneen Eurooppaan laajentumisen jälkeen Netflix on noussut maailman suosituimmaksi elokuvien ja sarjojen suoratoistopalveluksi. Netflixille on kertynyt jo yli 158 miljoonaa maksavaa käyttäjää ja palvelu on käytettävissä yli 190 maassa ympäri maailmaa [Netflix 2020].

Toinen mainitsemisen arvoinen videoiden suoratoiston jättiläinen on Googlen omistama YouTube. YouTuben palvelumalli eroaa huomattavasti Netflixistä, sillä käyttäjät itse tuottavat palvelussa tarjottavan sisällön. Käytännössä kaikilla käyttäjillä on mahdollisuus tuottaa videosisältöä palveluun, jota muut käyttäjät voivat kuluttaa palvelun kautta.

YouTubella on kuukausittain noin kaksi miljardia kirjautunutta kävijää, jotka kuluttavat yhteensä noin miljardi tuntia videomateriaali palvelun kautta [YouTube 2020]. 70% tästä kaikesta videomateriaalista katsotaan mobiililaitteilta yli sadassa eri maassa. Palvelun levinneisyyden lisäksi YouTube tukee paikallisia käyttäjiään tarjoamalla palvelua 80:llä eri kielellä.

Käyttäjien tuottaman sisällön lisäksi YouTube tarjoaa myös maksullista elokuvien vuokrausta, pelisisällön reaaliaikaista jakamismahdollisuutta ja maksullista YouTube Music:n kautta toistettavaa musiikkia. Vaikka näiden suurimpien suoratoistopalveluiden palvelumallit eroavatkin toisistaan, on niiden toiminnalla myös paljon yhtäläisyyksiä.

Google ja Netflix kuuluvat molemmat DASH-IF:iin (DASH Industry Forum), jonka tehtävänä on kehittää standardoitua DASH-suoratoistomenetelmää muiden jäsenorganisaatioiden kanssa. Tämän vuoksi ei lienekään ihme, että kummatkin palvelut ovat omaksuneet kyseisen menetelmän oman palvelunsa toiminnan taustalle [Bitmovin 2019]. Tätä mukautuvaa suoratoistomenetelmää esitellään tarkemmin tämän tutkielman neljännessä luvussa.

On kuitenkin huomioitava, että alan suurimpien tekijöiden lyöttäytyessä yhteen, yhteisen suoratoistomenetelmän kehittämiseksi, määrittää tämä menetelmä valtaosan videomateriaalin levityksestä. Näin ollen, alan suurimpien tekijöiden käyttämään suoratoistomenetelmään tehdyillä muutoksilla on mittavat vaikutukset koko Internetin IP-liikenteeseen.

2.3 Videosuoratoiston käyttökokemus

Alalla, joka nojaa voimakkaasti käyttäjien määrään, ja jossa on suurta kilpailua, on käyttäjäkokemuksella erittäin suuri merkitys menestyksen kannalta. Tämä pätee myös videoiden suoratoistoon, jonka kasvu on ollut räjähdysmäistä viimeisen kymmenen vuoden aikana. Esimerkiksi Netflixin käyttäjämäärä on kasvanut jopa n. 125 miljoonalla käyttäjällä vuodesta 2012, jolloin Netflix aloitti toimintansa Euroopan mantereella [Netflix 2020 (1)].

Kuluttajakokemuksen merkitys on kasvanut viime vuosina enenevässä määrin sovellusten ja verkkosivustojen parissa. Mikäli käyttökokemukseen jää huomattavasti parannettavan varaa, ei käyttäjä todennäköisesti palaa palvelun pariin, jos vaihtoehtoisia palveluita on tarjolla [Bocchi 2016]. Tällä on puolestaan selkeät vaikutuksen yhtiön tulokseen ja menestykseen. Onkin siis tärkeää tiedostaa suoratoistopalvelun mahdolliset ongelmakohdat, jotta käyttäjille voidaan tarjota parasta mahdollista palvelua.

Videoiden suoratoistossa suurimmat vaikutukset käyttökokemuksen kannalta ilmenee puskuroinnin aikana [Mok 2011]. Videon laadun heikkenemisellä tai kuvassa ilmenneillä häiriöillä ei ollut niin suurta vaikutusta käyttökokemuksen kannalta kuin keskeytyvällä toistolla. Näitä puskuroinnin aiheuttamia vaikutuksia käyttökokemukseen voidaan jakaa kolmeen eri osioon: Alustuspuskurointiaika, keskiarvoinen puskuroinnin kesto ja uudelleenpuskuroinnin toistumiskerrat [Mok 2011]. Näiden metriikkojen avulla on mahdollista mitata, millainen palvelutaso kyseisellä suoratoistopalvelulla on.

Alustuspuskurointiaika kuvastaa palvelulla kuluvaa aikaa, joka menee videon toiston käynnistämisestä videon toiston alkamiseen. Keskiarvoinen puskuroinnin kesto kuvastaa puolestaan keskimääräistä aikaa, joka kuluu yhden puskurointitapahtuman suorittamiseen. Uudelleenpuskuroinnin toistumiskerrat kertovat puolestaan, miten usein uudelleenpuskurointeja tapahtuu toiston aikana.

Tehdyn tutkimuksen mukaan uudelleenpuskuroinnin toistumisajalla oli suurimmat vaikutukset käyttökokemukseen [Mok 2011]. Tutkimuksen mukaan käyttäjät olivat valmiita odottamaan pidemmän ajan toistettavan sisällön alkamista, mutta useasti toistuvat uudelleenpuskuroinnit tekivät katselukokemuksesta selkeästi huonomman.

Näihin ongelmakohtiin ovat myös pureutuneet tämän hetkisen trendin suosituimmat suoratoistomenetelmät, jotka ovat mukautuvia suoratoistomenetelmiä. Mukautuvat

suoratoistomenetelmät pyrkivät välttämään puskurointiviiveitä, pienentämällä toistettavan sisällön laatua, mikäli verkon vallitsevat olosuhteet eivät mahdollista korkealaatuisen sisällön viiveetöntä toistamista. Toisaalta, mukautuvien suoratoistomenetelmien toimintaan kuuluu myös laadun parantaminen, mikäli tämä ei aiheuta puskurointiviiveitä videon toistamisessa. Näin ollen käyttäjille tarjotaan parasta mahdollista palvelua vallitsevat olosuhteet huomioiden.

2.4 Mobiilikulutus

Videomateriaalin kulutus on laajalti siirtynyt mobiililaitteille. Videoiden suoratoistopalvelut, kuten Netflix ja YouTube toimivat myös mobiiliapplikaatioilla, joiden avulla videomateriaali on saatavissa missä vain ja milloin vain. Ciscon tekemän ennusteen mukaan vuoden 2022 loppuun mennessä mobiilidatalla kulkeva liikenne kattaa jo viidenneksen kaikesta IP-liikenteestä [Cisco 2019]. Sama trendi on ollut havaittavissa myös suoratoistopalveluiden julkaisemissa tiedotteissa.

YouTube on ilmoittanut julkaisemassaan tiedotteessa, että n. 70%:n osuus palvelun suoratoistoista tapahtuu mobiililaitteilla [YouTube 2020]. Kokonaisuudessaan tämä kattaa siis lähes 700 miljoonaa tuntia katselu-aikaa päivittäin YouTuben palvelusta. Myös Netflix on ilmoittanut, että vain n. 15% palvelun pitkäaikaisista katsojista käyttää palvelua tietokoneen välityksellä [Kafka 2018]. Suurin osa Netflixin katseluista tapahtuukin television ja mobiililaitteiden kautta. Videoiden suoratoisto on siis siirtymässä kohti mobiilia kulutusta, mutta palvelumalli ei kuitenkaan ole ongelmaton.

Mobiililaitteet tuovat mukanaan suuren määrän ongelmakohtia, jotka on ratkaistava sekä sovellus- että laitteistotasolla. Mobiililaitteet hyödyntävät eri käyttöjärjestelmiä, minkä vuoksi palveluiden tarjonnan on toimittava usealla eri toiminta-alustalla. Käyttöjärjestelmien eroavaisuudet tuovat mukanaan sovelluskehityksellisiä seikkoja, jotka on syytä ottaa huomioon sovellusten kehitysvaiheessa. Suoratoistopalveluiden sovellukset ja niiden käytettävyys eivät kuitenkaan ole tämän tutkielman kannalta merkittäviä, joten niitä ei käsitellä tässä tutkielmassa.

Mobiililaitteiden laitteistolliset ominaisuudet poikkeavat myös toisistaan merkittävästi. Puhelimet ja tabletit tarjoavat käyttäjilleen moninaisen valikoiman laitteiston puolesta. Laitteiden pienen koon vuoksi myös muistin ja akun kapasiteetti ovat selkeästi

pienemmät kuin esimerkiksi kuluttajien tietokoneissa. Suoratoistopalveluiden onkin pyrittävä tarjoamaan parasta mahdollista palvelua vaihtelevalla laitteistolla ja päästävä yli laitteiston tuomista rajoitteista. Tämän lisäksi toiston on jatkuttava ilman suurempia katkoksia, vaikka käyttäjän maantieteellinen sijainti vaihtuu. Näiden ongelmien ratkaisemiseksi on hyödynnettävä käytettävissä olevaa teknologiaa ja tekniikoita mahdollisimman tehokkaasti.

Suoratoistopalveluiden yksi tärkeimmistä tavoitteista on maksimoida käyttäjien käyttökokemuksen laatu. Suoratoiston on tapahduttava ilman katkoksia, vaikka yhden pullonkaulalinkin ympärille olisikin sijoittunut suuri määrä käyttäjiä [Mehrabi 2019]. Tämän lisäksi mobiilikäyttäjien verkko-olosuhteet ovat jatkuvasti muutoksessa käyttäjien liikkumassa paikasta toiseen. Jotta palvelu toimisi kuitenkin parhaimmalla mahdollisella tavalla valtaosalle käyttäjistä, ovat suoratoistopalvelut siirtyneet käyttämään mukautuvia suoratoistomenetelmiä. Näiden menetelmien, kuten HLS (HTTP Live Streaming) ja DASH (Dynamic Adaptive Streaming over HTTP), avulla pystytään reagoimaan verkon muuttuviin olosuhteisiin ja toimimaan sen mukaisesti joko heikentäen tai parantaen suoratoistettavan sisällön laatua [Mehrabi 2019]. Jotta verkon muuttuviin olosuhteisiin voidaan reagoida, tulee käyttäjän myös pysyä tavoitettavissa, vaikka käyttäjän maantieteellinen sijainti muuttuisikin. Tämä tapahtuu mobiili-IP:n (Mobile IP) avulla.

Mobiili-IP mahdollistaa käyttäjän tavoittamisen, vaikka käyttäjän sijainti verkolla muuttuisikin kesken yhteyden [Nada 2006]. Mobiili-IP:n vahvuus on siinä, että käyttäjän ei tarvitse muuttaa pitkäaikaista IP-osoitettaan, vaikka tämän sijainti verkolla vaihtuisikin. Näin ollen käyttäjällä on kaksi IP-osoitetta, joiden avulla toimia verkossa. Yksi osoite tunnistautumista varten, ja toinen osoite reititystä varten.

Mobiili-IP perustuu seuraaviin toiminnallisuuksiin: välittäjän löytö (agent discovery), rekisteröityminen (registration) ja tunnelointi (tunneling) [Nada 2006]. Näiden avulla käyttäjä tunnistaa muuttuneen verkon, rekisteröityy vieraaseen verkkoon ja ilmoittaa tästä kotiverkon välittäjälle, joka tunneloi saapuvat segmentit vieraassa verkossa olevalle käyttäjälle.

Välittäjän löytämiseksi mobiilikäyttäjä kuuntelee vierailtavaa verkkoa. Verkkoon välitetyn välittäjämainonnan (agent advertisement) saapuessa mobiilikäyttäjälle, tunnistaa käyttäjä verkon, johon kuuluu sillä hetkellä. Mikäli käyttäjä kuuluu vieraaseen verkkoon, saa tämä väliaikaisen "vierasosoitteen" (care-of address), joka välitetään kotiverkon

välittäjälle (home agent). Vierasoitteen rekisteröinnin jälkeen kotiverkon välittäjä ryhtyy välittämään saapuvia lähetyksiä liikkeellä olevalle mobiilikäyttäjälle tämän vierasoitteen perusteella.

Lähetyksien välitys mobiilikäyttäjälle tapahtuu tunneloinnin avulla. Tunneloinnissa kotiverkon välittäjä kapseloi välitettävän lähetyksen mobiilikäyttäjän rekisteröimällä IP-osoitteella ja lähettää se vierailtavan verkon välittäjälle. Tämän vastaanotettua lähetyksen, kapselointi puretaan ja lähetys välitetään vierailevalle mobiilikäyttäjälle. Lähetyksen välitys verkkovierailevalle mobiilikäyttäjälle tapahtuu tavallisen IP-reitityksen avulla.

3 Yleisimmät suoratoistomenetelmät ennen

Valtaosa verkon liikenteestä tänä päivänä on mediasisällön siirtoa. Tätä liikennöintiä varten kehitetyt menetelmät ovat pääasiassa siirtyneet dynaamiseen mukautumiseen verkon olosuhteiden tai käyttäjän valintojen perusteella. Aina ei kuitenkaan näin ole ollut.

Videoiden suoratoiston historian aikana, aina 90-luvun alusta tähän päivään saakka, on ollut tiettyjä suoratoistomenetelmiä, jotka ovat luoneet pohjan nykyisille menetelmille. Jotta voimme tunnistaa nykyisten menetelmien hyvyiden ja mahdolliset ongelmakohdat, on vertailukohdan vuoksi syytä tutustua tarkemmin jo väistyneisiin menetelmiin ja ratkaisumalleihin. Tässä kappaleessa pyritäänkin esittelemään verkon infrastruktuurin kannalta tärkeitä elementtejä kuten käytettäviä kuljetuskerroksen protokollia ja sisällönjake-luverkkoja sekä aikaisemmin vallinneita suoratoistomenetelmiä.

3.1 Kuljetuskerroksen protokollat: TCP ja UDP

Internetin monimuotoisuuden vuoksi ei ole vain yhtä ratkaisua siirtää jaettavaa sisältöä eri toimijoiden välillä. IP (Internet Protocol) toimii vankkana pohjana, jonka päälle on rakennettu useita eri mahdollisuuksia sisällön siirtämiseen OSI-mallin (Open Systems Interconnection Reference Model) seuraavalla kerroksella eli kuljetuskerroksella. Näistä tärkeimpiä ja käytetyimpiä ovat TCP (Transmission Control Protocol) ja UDP (User Datagram Protocol) [Apostolopoulos 2003].

Kuljetuskerroksen protokollat tarjoavat suoratoistopalveluille, sekä muille verkon sisälönjakajille, toimijoiden välisiä kuljetuspalveluita [Wu 2001]. Tämän lisäksi kuljetuskerroksen protokollat tuovat mukanaan erilaisia “lisäpalveluita” ja ominaisuuksia, jotka soveltuvat erilaisiin käyttötarkoituksiin. Onkin siis syytä tarkastella näitä protokollia hie- man tarkemmin, jotta nähdään, miten suoratoistomenetelmät ovat kokeneet muutoksia myös käytettävien kuljetuskerroksen protokollien myötä. Tämän lisäksi pystymme tar- kemmin punnitsemaan videoiden suoratoiston tärkeimpiä ominaisuuksia ja analysoi- maan, kuinka käytettävät protokollat tukevat niitä.

TCP ja UDP ovat luonteeltaan varsin erilaisia. Molemmat protokollat toimivat toki kul- jetuskerroksella ja mahdollistavat sisällön siirtämisen lähettäjältä vastaanottajalle. Tämän lisäksi sekä TCP että UDP pystyvät erottelemaan saapuvan sisällön eri applikaatioiden välillä (multiplexing) ja molemmat protokollat tarjoavat virheiden hallintaa tarkistussum- man (checksum) muodossa. Käytännössä siis bittivirheet tunnistetaan ennen sisällön vä- littämistä asiakasohjelmalle. Virheellinen datagrammi (UDP) tai segmentti (TCP) pудо- tetaan saapuessaan vastaanottajalle, mutta tähän yhtäläisyydet sitten loppuvatkin. Kor- ruptoituneen sisällön tapauksessa TCP reagoi “kadonneeseen” segmenttiin ilmaisemalla sisällön puuttumisesta lähettäjälle. Ilmoituksen saavuttua segmentti toimitetaan uudel- leenlähetyksen yhteydessä. UDP ei takaa sisällön toimitusta toimijoiden välillä, joten kor- ruptoitunutta datagrammia ei toimiteta uudelleen.

TCP on luotettava kuljetuskerroksen protokolla, joka takaa toimitettavan sisällön siirron toimijoiden välillä [Wu 2001]. Sisällön kuljetus toteutetaan yhteydellisen tavuvirran (byte-stream) välityksellä [Apostolopoulos 2003]. Tämän toimijoiden välisen yhteyden sekä TCP:n “toimitustakuun” seurauksena jokainen toimittamaton, vaurioitunut, tuplaan- tunut, väärässä järjestyksessä toimitettu tai korruptoitunut segmentti uudelleenlähetetään vastaanottajalle [IETF 2020]. Käytännössä tämä tarkoittaa sitä, että jokainen vastaan- otettu segmentti kuitataan vastaanotetuksi lähettäjälle lähettämällä ACK-viesti (acknow- ledgement). Tämän viestin yhteydessä ilmoitetaan myös seuraavaksi odotetun segmentin numero, jotta lähettäjä on tietoinen järjestyksessä saapuneista segmenteistä. Mikäli lähet- täjä ei vastaanota ACK-viestiä lähetetystä segmentistä, toimitettu sisältö lähetetään uu- delleen uudelleenlähetyksajastimen mennessä umpeen. Tarpeeton uudelleenlähetyks voi pa- himmassa tapauksessa kuitenkin tuottaa siirtoikkunan (transfer window) verran ylimää- räistä kuormitusta jo ennestään ruuhkautuneelle verkolle.

Verkon ollessa ruuhkautuneessa tilassa, TCP:n käyttäminen voi aiheuttaa lisäkuormitusta verkolle, verrattuna UDP:hen, joka ei takaa sisällön toimitusta vastaanottajalle. Mikäli lähetyksen aikana osa datagrammeista katoaa tai vaurioituu, ei niitä uudelleenlähetetä vastaanottajalle. Tämän lisäksi yhtäkään datagrammia ei vastaanottajan tarvitse kuitata vastaanotetuksi, sillä lähettäjä ei kuitenkaan reagoisi näihin ilmoituksiin. Verkolle ei siis aiheudu lisäkuormitusta ACK-viestien ja uudelleenlähetyksien seurauksena.

UDP on yhteydetön kuljetusprotokolla, joka toimii käytännössä rajapintana (interface) IP:lle (Internet Protocol) [Apostolopoulos 2003]. Lähettäjän tehtävänä on ainoastaan lähettää sisältö vastaanottajalle yhden kerran. Tämän UDP:n ominaisuuden seurauksena se on huomattavasti kevyempi kuin TCP ja vaatii sekä vastaanottajalta että palvelimelta selvästi vähemmän kuin TCP. UDP tarjoaa myös enemmän liikkumavaraa korkeammilla tasoilla, sillä sovellus voi itse määrittää, kuinka hallinnoida ruuhkautumista (congestion control) tai paineentasausta (flow control). Nämä toiminnallisuudet on implementoitu TCP:hen jo valmiiksi, mutta UDP ei tällaisia ominaisuuksia tarjoa. Onkin siis sovelluksen vastuulla määrittää, kuinka tällaisissa tilanteissa toimitaan.

Videoiden suoratoisto on pitkään nojannut UDP:hen, sillä monelta osin se sopii paremmin suoratoiston kuljetusprotokollaksi kuin TCP. Se on kevyt palvelimelle, se tarjoaa lisävapauksia sovelluksen kehitykseen ja se ei vaadi yhteyttä osapuolien välille. Lisäksi TCP:n tarjoama uudelleenlähetykset on monelta osin turhaa videoiden suoratoistossa, joka on aikakriittinen palvelu. Aikakriittisyys on huomattavissa etenkin reaaliaikaisissa suoratoistopalveluissa, joissa ei ole aikaa uudelleenlähetyksille. Tällaisissa tapauksissa ei ole syytä toimittaa uudelleen sisältöä, jota ei keretä edes toistamaan asiakasohjelmassa. Yhden kuvan puuttuminen suoratoistettavassa videossa, jota toistetaan esimerkiksi 60 kuvaa/sekunti kuvataajuudella, ei ole suurta merkitystä käyttökokemuksen kannalta. Voisi sanoa, että on parempi jättää tällöin sisältö uudelleenlähettämättä ja säästää samalla verkkoa turhalta kuormitukselta.

UDP:n viive toimituksissa on myös ennustettavampi kuin TCP:n kohdalla [Apostolopoulos 2003]. TCP:n viive voi vaihdella 1-3:n suuntaviiveen verran, mikäli segmentti saadaan toimitettua yhden uudelleenlähetyksen aikana. UDP-datagrammien viive on puolestaan aina yhdensuuntaisen viiveen verran.

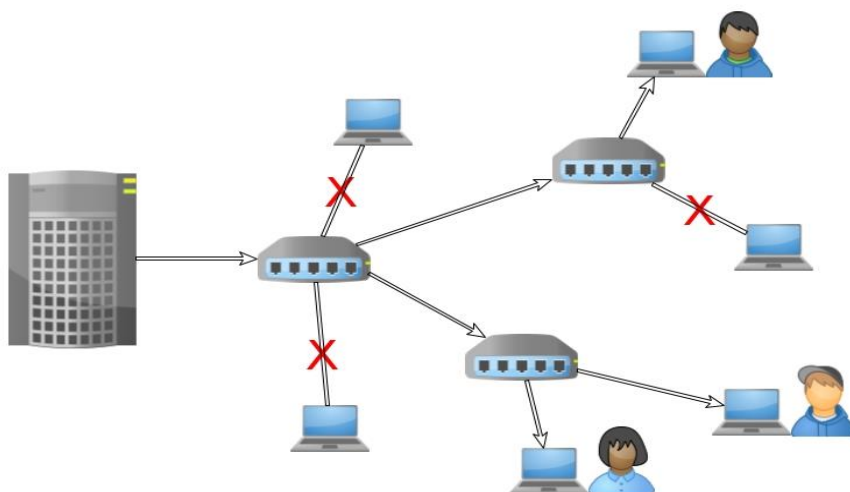
UDP on toiminut pitkään vallitsevana kuljetusprotokollana median suoratoistossa edellä mainittujen ominaisuuksien johdosta. Viime vuosina on kuitenkin ollut selkeä trendi

siirtyä TCP:n puolelle [Microsoft 2010]. Tämä johtuu siitä, että median suoratoisto on enenevässä määrin siirtynyt HTTP-suoratoistoon, joka toimii TCP:n päällä. HTTP-suoratoiston suosion kasvettua TCP:stä on tullut vallitseva kuljetusprotokolla suurimpien suoratoistopalveluiden keskuudessa [DASH 2019]. Siirtyminen HTTP-suoratoistoon ja TCP:hen on kuitenkin tuonut paljon hyötyä suoratoistopalveluille. Esimerkiksi olemassa olevan arkkitehtuurin hyödyntäminen, taloudelliset syyt ja käyttöönoton helppous ovat olleet avainasemassa tässä muutoksessa. Näistä HTTP-suoratoiston hyödyistä tarkemmin tutkielman neljännessä luvussa.

3.2 Ryhmälähetys (*Multicasting*)

Internetin alkuperäinen malli soveltuu parhaiten täsmälähetykseen (unicast) esimerkiksi sähköpostien tai tiedostosiirtojen yhteydessä. Tämä sisällönjakotapa ei kuitenkaan sovellu ideaalisesti suuriin tiedostolähetyksiin monille vastaanottajille [Wu 2001]. Tällaista jakoa tukemaan on olemassa ryhmälähetys (multicast).

IP-ryhmälähetys (IP multicast) on tapa välittää lähetettävää sisältöä yhdeltä lähettäjältä usealle vastaanottajalle. Tämä sopii hyvin tilanteisiin, joissa saman sisällön vastaanottajia on samanaikaisesti merkittävä määrä. IP-ryhmälähetyksen avulla vältetään ylimääräinen verkon kuormittaminen sekä lähettäjän että vastaanottajan päässä [Cisco 2000].



(Kuva 3.1: Esimerkki ryhmälähetyksen haarautumisesta verkossa)

IP-ryhmälähetys perustuu ryhmiin, joihin vastaanottajat voivat liittyä, mikäli he ovat kiinnostuneita jaetun sisällön vastaanottamisesta [Cisco 2000]. Näihin ryhmiin kuuluvat jäsenet voivat olla missä tahansa osassa verkkoa, joten ryhmälähetystä ei ole rajattu maantieteellisesti tai fyysisesti. Vastaanottaja voi liittyä haluamaansa ryhmään lähettämällä IGMP-viestin (Internet Group Management Protocol) verkkoon, jolla ilmoittaa kiinnostuksensa viestissä ilmoitetun ryhmän IP-osoitteeseen tulevista paketeista. Näin ollen verkossa on tieto, että tästä ryhmälähetyksen puun “oksasta” ja tietyistä “lehdestä” löytyy vastaanottaja lähetettävälle medialle. Verkon jäsen ei voi vastaanottaa tietyn ryhmän paketteja kuulumatta siihen itse. Verkon jäsenet lähettävät tietyin väliajoin viestejä verkon ryhmälähetys-reitittimelle selvittääkseen, mitä ryhmiä tietyssä aliverkossa on tarjolla.

Tavallisesti sisältö toimitetaan vastaanottajalle tämän MAC-osoitteen (Media Access Control) perusteella [Cisco 2000]. Ryhmälähetykseen ei kuitenkaan voida määrittellä vastaanottajan MAC-osoitetta, sillä lähetys toimitetaan usealle vastaanottajalle. Tämän vuoksi MAC-osoitteessa on ensimmäisen oktetin 0-paikalla oleva bitti, joka kertoo, onko sisältö täsmälähetys vai yleinen lähetys joko ryhmälähetysten tai yleislähetysten (broadcast) muodossa.

Ryhmälähetys käyttää UDP:tä kuljetuskerroksen protokollana. TCP olisi luotettavampi protokolla siirtoon ja tarjoaisi parempaa verkon ruuhkautumisen hallinnointia, mutta aiheuttaisi liikaa kuormitusta verkolle. UDP:n avulla vältytään tarpeettomista ACK- ja NACK-viesteistä (Negative Acknowledgement) vastaanottajien osalta, sillä kadonnutta tai korruptoitunutta sisältöä ei voida uudelleenlähettää jokaiselle käyttäjälle.

IP-ryhmälähetysten sisällön jako tapahtuu puu muodostelmassa. Yksinkertaisimmillaan puu on muodoltaan lähdepuu (source tree), jossa puun juuri on sisällön jakajassa [Cisco 2000]. Ryhmään kuuluvat ryhmälähetysreitittimet muodostavat keskenään puun, jonka oksia pitkin lähetettävä sisältö kuljetetaan puun lehdille. Sisältö lähtee lähettäjältä tämän oman aliverkon ryhmälähetysreitittimelle, josta lähetys ohjataan eteenpäin muille kiinnostuneille ryhmälähetysreitittimille. Mikäli kyseisen reitittimen aliverkossa on kiinnostunut vastaanottaja, toimitetaan sisältö myös tälle vastaanottajalle. Käytännössä jokainen ryhmälähetysreititin tietää siis, onko tätä korkeammilla oksilla kiinnostuneita vastaanottajia, joiden perusteella sisältöä ohjataan oikeisiin portteihin.

Toinen tapa muodostaa IP-ryhmälähetysten puu, on sopia “jakopiste” (rendezvouspoint), jonne sisältö ohjataan, ja josta sisältöä myös haetaan [Cisco 2000]. Puun juuri

muodostuukin jakopisteeseen. Tällaisen muodostelman nimi on jakopuu (Shared tree). Jakopuun hyöty yksinkertaiseen lähdepuuhun nähden on, että jakopisteen ollessa keskellä verkkoa, on se yhtäläisen lähellä vastaanottajia. Toisaalta, sisällön jakajasta vastaanottajalle ei reitti välttämättä ole kuitenkaan optimaalisin (vrt. lähdepuu). Esimerkiksi sisällön jakaja saattaa olla sijoittuneena lähelle vastaanottajaa, jolloin tämän vastaanottajan kannalta lyhin reitti ei vastaa ollenkaan todellista reittiä, jonka sisältö joutuu kulkemaan. Tämä aiheuttaa myös ylimääräistä kuormitusta verkolle.

Vaikka IP-ryhmälähetys vaikuttaisikin olevan täydellinen ratkaisu multimediasisällön lähettämiseen verkon yli, on siinä omat ongelmakohtansa. IP-ryhmälähetyksen ongelmakohtiin lukeutuvat mm. skaalautuvuus (scalability) ja korkeampien kerroksien toimintojen, kuten virheidenhallinta (error control) ja ruuhkahallinta (congestion control), käyttöönotto ja näiden toiminnallisuuksien tuki [Wu 2001]. Näiden ongelmakohtien ylipääsemiseksi onkin kehitetty myös mm. sovellustason ryhmälähetystä (application-level multicast).

Sovellustason ryhmälähetys rakentuu kerrosverkkomalliin. Tämän vuoksi itsenäiset palveluntarjoajat voivat rakentaa omia ryhmälähetysverkkoja. Näitä erillisiä verkkoja voidaan liittää toisiinsa sovellustasolla muodostaen suuremman, yhtenäisen, verkon [Wu 2001]. Tällaisen ratkaisun avulla reititys voidaan suorittaa sovellustasolla, dynaamisesti optimoituen verkon olosuhteiden muutoksiin. Sovellustasoisen ryhmälähetyksen hyötyinä on mm. verkon ylimääräisen kuormituksen välttäminen, sisällönjaon skaalautuvuuden parantaminen sekä verkon ruuhkaantumisen hallinnointi.

3.3 Suoratoisto ennen

Videoiden suoratoisto on ollut kasvujohteisessa kehityksessä 90-luvun alusta asti. Näiden vuosien saatossa se on muotoutunut suurimmaksi tekijäksi verkon liikenteen kannalta. Tämän suhteellisen lyhyen historian aikana käydyt vaiheet ovat johdattaneet videoiden suoratoiston pisteeseen, jossa se tällä hetkellä on. Vallitsevien menetelmien muutokset ovat johtaneet videoiden suoratoiston tilaan, jossa miljoonat ihmiset ympäri maailmaa voivat hakeutua suurimpien palveluiden ääreen, haluamanaan ajankohtana ja nauttia niiden tarjoamista palveluista parhaalla mahdollisella laadulla.

Aivan 1990-luvun alussa käytetty TCP:n päällä toimiva HTTP asteittainen lataus

(progressive download) alkoi väistyä mediasisällön siirtoprotokollana uusien protokollien tieltä. Asteittainen lataus ei varsinaisesti toimi suoratoistona, sillä sisältö ladataan kokonaisuudessaan käyttäjän laitteelle, josta sisältöä toistetaan. Uudet protokollat puolestaan mahdollistivat sisällön toistamisen suoratoiston muodossa. Näihin protokolleihin luokutuivat mm. RTP (Real-time Transport Protocol) ja tätä hallinnoiva RTCP (RTP Control Protocol).

RTP:n käyttö ei vaadi kaistan varaamista siirtoa varten [Li 2013]. Suurena hyötynä RTP mahdollistaa median siirron yhteydessä myös tietopakettien katoamisen havaitsemisen ja siirtoviiveen vaihtelun (jitter) tasaamisen. Näiden ominaisuuksien avulla on mahdollista kontrolloida lähetyksen etenemistä vastaanottajalle suotuisammaksi.

UDP:n päällä toimiva RTP nousikin pian erittäin suosituksi siirtoprotokollaksi, sillä se tarjosi nopean ja kontrolloitavan päästä-päähän yhteyden lähettäjän ja vastaanottajan välille [Li 2013]. RTP ei kuitenkaan itsessään pysty tunnistamaan pakettien katoamista tai tasaamaan siirtoviiveen vaihtelua. Tätä varten RTP tarvitsee siirtoa kontrolloivan protokollan, RTCP:n [Li 2013]. RTCP, joka toimii myös UDP:n päällä, tarkkailee siirron etenemistä. RTCP:n tärkeimpänä tehtävänä on useiden samanaikaisten siirtovirtojen (stream) synkronointi. Tämän synkronoinnin yhteydessä RTCP myös tekee siirtoviiveiden tasausta, jotta jokainen paketti on oikeassa paikassa oikeaan aikaan vastaanottajan puolella. Tämän protokollan omaksui myös käyttöön suuret kaupalliset tuotteet ja palvelut, kuten QuickTime [Li 2013].

RTP:n kanssa samoihin aikoihin kehitettiin myös nimenomaan multimedian siirtoa varten oma protokollansa, RTSP (Real Time Streaming Protocol). RTSP:n avulla pystyttiin myös mahdollistamaan käyttäjille tuttuja toiminnallisuksia videoiden toistosta, kuten kelausmahdollisuus ja videon pysäytys sekä toiston jatkaminen [Li 2013]. Käyttäjille pyrittiin tarjoamaan mahdollisimman tuttu toiminnallisuus oikeasta elämästä myös videoiden suoratoistossa. Nykyään nämä ominaisuudet tuntuvat varsin triviaaleilta nykypäivän suoratoistopalveluissa. Olisikin vaikea kuvitella miljoonien käyttäjien käyttämiä palveluita ilman mahdollisuutta etsiä haluttua kohtaa toistettavasta sisällöstä ja jatkaa toistoa käyttäjän haluamasta kohdasta.

Microsoftin hyödyntäessä RTSP:n hybridimallia oman Smooth Streaming -implementaation myötä, Macromedia oli kehittämässä toista protokollaa, RTMP:tä (Real-Time Messaging Protocol). RTMP on tilallinen suoratoistoprotokolla, joka ylläpitää yhteyttä

lähettäjän ja vastaanottajan välillä koko suoratoiston ajan [Cisco 2011]. Siirtoprosessin ajan palvelin ylläpitää käyttäjän ja siirron tilaa. Yhteys käyttäjän ja palvelimen välillä puretaan vasta, kun siirto on suoritettu loppuun tai toisen osapuolen katkaistessa yhteyden.

RTMP tarjoaa kuitenkin hyötyjä esimerkiksi HTTP:hen nähden, jota yleisimmät suoratoistoprotokollat tänä päivänä tukevat. Esimerkiksi RTMP tukee ryhmälähetys-ominaisuutta (multicast), jota HTTP ei tue [Cisco 2011]. Tämän seurauksena palvelut ja organisaatiot, joille ryhmälähetys tuo tarvittua hyötyä suoratoistoratkaisuissa, on RTMP varteenotettava valinta suoratoistoprotokollaa valittaessa. RTMP-protokollan välittämät tietopaketit eivät myöskään ole yhtä alttiita paikalliselle tallentamiselle välittävien osapuolien osalta kuin HTTP:n yhteydessä. Näin ollen RTMP tarjoaa tässä mielessä turvallisempaa sisällön siirtoa kuin HTTP:tä hyödyntävät suoratoistomenetelmät. Tämä on huomattavissa esimerkiksi RTMPE:n (RTMP Encryption) yhteydessä, joka hyödyntää Adobe:n kehittämää reaaliaikaista kryptografiaa suojaussessaan [Cisco 2011].

RTMP:n yksi suurimmista hyödyistä on kuitenkin tilatiedot, joita se ylläpitää suoratoiston yhteydessä [Cisco 2011]. Näiden tilatietojen avulla palvelimen ja käyttäjän välisen kommunikaation tila on jatkuvasti tiedossa. HTTP on puolestaan tilaton protokolla, minkä vuoksi näitä tietoja ei ole saatavilla. HTTP:llä on kuitenkin suuri etu sen yleisessä käytössä, minkä vuoksi sen liikennöinti on vapaampaa palomuurien ja osoitteenmuunnosta suorittavien laitteiden (NAT:ien) läpi. Tämän lisäksi HTTP-liikennöinti on halvempaa, sillä sisällönjakeluverkon solmut ovat tavanomaisesti veloittaneet enemmän RTMP-suoratoiston liikennöinnistä [Cisco 2011]. Tämä perustuu siihen, että RTMP on ollut yleisesti Flash-videoiden käytössä, minkä vuoksi tällaisesta liikenteestä on helpompi verottaa. HTTP-liikenne sisältää myös "tavallista" verkkoliikennettä, minkä vuoksi sen liikennöinti on varmempaa ja halvempaa. Muun muassa tästä syystä HTTP-pohjaiset protokollat ovat nousseet vallitseviksi suoratoistoprotokolliksi.

3.4 Sisällönjakeluverkko

Internetin käyttötarkoitus on muuttunut täysin siitä, mihin se alkuaan suunniteltiin [Vakali 2003]. Yliopistojen ja tieteellisten yhteisöiden kommunikaatiokanavasta on muotoutunut verkko, jonka resursseista suuri osa käytetään videomateriaalin kuljetukseen eri

osapuolien välillä. Voidaankin siis sanoa, että Internetin arkkitehtuuria ei ole valmistettu kantamaan niin raskasta liikennettä kuin se tänä päivänä joutuu kantamaan. Tämän käytötarkoituksen muuttumisen tuoman rasitteen lievittämiseksi on olemassa olevan arkkitehtuurin päälle luotu toinen verkko, sisällönjakeluverkko.

Vuonna 1998 Internetin tueksi pystytetty sisällönjakeluverkko tarjoaa useita hyötyjä verkon käyttäjille, niin yksityistoimijoille kuin mediasisältöä tarjoaville toimijoillekin. Näihin hyötyihin lukeutuvat mm. parempi sisällön saatavuus, sisällön nopeampi toimitus sekä pienempi kaistan kulutus [Vakali 2003]. Näiden ominaisuuksien lisäksi sisällönjakeluverkot parantavat verkon skaalautuvuutta ja mahdollistavat sisällön kulutuksen suurissa käyttäjäpiikeissä.

Sisällön jakeluverkko koostuu solmuista (CDN-solmu), jotka on levitetty alla olevan arkkitehtuurin päälle omaksi verkokseen. Mitä hajanaisemmin solmut on sijoitettu, sitä suurempi kattavuus niiden avulla saadaan. Toisaalta solmujen sijoittamisessa tulee ottaa huomioon myös niihin aiheutuva kuormitus. Tämän vuoksi esimerkiksi tiheästi asutuilla alueilla tulee olla paremmat palveluresurssit kuin haja-asutusalueilla.

CDN-solmut nojaavat välimuistiin, jota ne rakentavat verkossa liikkuvan sisällön avulla [Vakali 2003]. Muodostetun välimuistin avulla CDN-solmut voivat tarjota paremman palvelunopeuden käyttäjille ja vähentää alkuperäisen sisällöntarjoajan palvelimien kuormitusta. Palvelimen kuormitus pienenee, sillä videomateriaali on saatavilla lähellä palvelevasta CDN-solmusta alkuperäisen palvelimen sijaan. Tämä myös pienentää verkon kaistan kulutusta. Sisällön ollessa saatavilla lähellä palvelevasta CDN-solmusta, ei sitä tarvitse kuljettaa verkon halki. CDN-solmujen välimuistin avulla myös käyttökokemus paranee, sillä alkuperäisen sisällöntarjoajan palvelimen kaatuessa, sisältö on vielä mahdollisesti toistettavissa CDN-solmujen välimuistista.

4 HTTP-suoratoisto ja dynaaminen mukautuminen

HTTP-suoratoisto on noussut vallitsevaksi suoratoistomenetelmäksi. Tämän kehityksen saattamina suoratoistopalvelut ovat kuitenkin siirtyneet tavanomaisesta HTTP-suoratoistosta mukautuviin suoratoistomenetelmiin. Mukautuvat suoratoistomenetelmät tarjoavat palveluille kyvyn reagoida sekä verkon olosuhteiden että käyttäjän tekemien muutoksien vaikutuksiin suoratoiston kannalta. Dynaamisen mukautumisen myötä on pystytty takaamaan parempi käyttökokemus suoratoistopalveluiden käyttäjille.

Tämä luku esittelee HTTP-suoratoiston keskeisimpiä toimintaperiaatteita sekä tästä johdettuja mukautuvia suoratoistomenetelmiä. Luvun keskiössä on MPEG-DASH -suoratoistomenetelmä, joka on noussut yhdeksi suurimmista ja käytetyimmistä suoratoistomenetelmistä tämän päivän suoratoistopalveluissa. Lisäksi tässä luvussa tarkastellaan yhden suurimman suoratoistopalvelun, Netflixin, toimintamallia sekä syvennyttään verkon olosuhteiden tuottamien virhetilojen hallintaan ja niistä palautumiseen.

4.1 HTTP-suoratoiston toimintaperiaatteet ja infrastruktuuri

HTTP-suoratoisto on nimensä mukaisesti Hypertext Transfer Protocol -protokollaa (HTTP) hyödyntävä suoratoistomenetelmä. HTTP on tilaton (stateless), sovelluserroksen protokolla, joka mahdollistaa verkossa olevan sisällön siirtämiseen mm. www-palvelimien ja verkkoselainten välillä [Fielding 1999].

HTTP toimii pyyntöjen (request) ja vastausten (response) avulla. Yksinkertaisimmillaan käyttäjä voi siis pyytää palvelimelta haluttua sisältöä ilmoittamalla pyyntötyypin (esim. GET), halutun sisällön sijainnin eli URI:n (Uniform Resource Identifier) sekä käytettävän HTTP-version [Fielding 1999]. Tämän lisäksi pyyntö sisältää mm. käyttäjää tarkentavia tietoja. Pyyntöä käsiteltyä, palvelin vastaa käyttäjälle pyynnön onnistumista kuvaavalla koodilla, metatiedoilla ja pyydetyllä sisällöllä. Tämän jälkeen HTTP-yhteys osapuolten välillä katkeaa. Mikäli käyttäjä kuitenkin tarvitsee jatkoa toimitettuun sisältöön, voi tämä muodostaa uuden HTTP-yhteyden palvelimeen ja tehdä samat toimenpiteet uudelleen.

Tämä HTTP-protokollasta omaksuttu toimintamalli poikkeaa huomattavasti aikaisemmin vallitsevista suoratoistomenetelmistä, kuten RTSP (Real-Time Streaming Protocol) ja Adobe:n kehittämästä RTMP:stä (Real-Time Messaging Protocol). Näissä kyseisissä suoratoistotekniikoissa yhteys käyttäjän ja palvelimen välillä pysyy ylhäällä, kunnes käyttäjä

katkaisee yhteyden. Tämän yhteyden aikana käyttäjä vastaanottaa halutun sisällön yhtäjaksoisena virtana (stream) sisällölle omistautuneelta palvelimelta joko UDP:n tai TCP:n välityksellä [Stockhammer 2011]. Palvelin on jatkuvasti tietoinen käyttäjän tilasta, ja toimii tämän tilatiedon mukaisesti.

HTTP-protokolla on puolestaan tilaton, minkä seurauksena jokainen pyyntö tapahtuu yksilöllisenä tapahtumana. Tämän lisäksi tavallisten HTTP-palvelimien palvelussa asiakkaan pyyntöä videomateriaalin toimituksesta, tapahtuu tämä asteittaisen latauksen (progressive download) muodossa [Stockhammer 2011].

Asteittaisen latauksen käyttökokemus on käyttäjän näkökulmasta varsin samanlainen kuin oikeassa suoratoistossa, mutta pinnan alla erot ovat kuitenkin huomattavat. Asteittaisen latauksen yhteydessä katsottava videomateriaali ladataan kokonaisuudessaan käyttäjän laitteelle. Tämä voi mahdollisesti aiheuttaa turhaa kuormitusta verkolle, mikäli käyttäjä ei päädykään katsomaan ladattua sisältöä kokonaisuudessaan [Stockhammer 2011]. Asteittainen lataus ei myöskään ole olosuhteisiin mukautuva menetelmä, minkä vuoksi käyttökokemus voi jäädä heikoksi mahdollisten puskurointiviiveiden johdosta. Tässä ei suinkaan ole asteittaisen latauksen kaikki huonot puolet.

Asteittaisessa latauksessa sisältö ladataan kokonaisuudessaan, jokainen videosisällön osa omalla vuorollaan. Mikäli käyttäjä tahtoi esimerkiksi siirtyä eteenpäin videolla, joutuisi tämä odottaa, että videon lataus on yltänyt haluttuun kohtaan asti. Tästä seuraa pitkä puskurointiviive, jolla on huomattava vaikutus käyttökokemukseen [Mok 2011].

Videomateriaalin lataamisessa käyttäjän laitteelle on myös se huono puoli, että suoratoistopalveluiden kulutuksen siirtyessä enemmän ja enemmän mobiililaitteille, ei tallennuskapasiteetti välttämättä riitä halutun videosisällön tallentamiseen. Tästä huolimatta HTTP-suoratoistaminen on käytössä monissa suoratoistopalveluissa. Syy tähän löytyy HTTP-suoratoiston tuomista hyödyistä.

Yksi suurimmista hyödyistä, joita HTTP-suoratoisto tarjoaa palveluntarjoajille, on infrastruktuuri, jonka päälle HTTP-liikenne on rakennettu. Olemassa olevan infrastruktuurin hyödyntäminen vähentää palvelun käynnistämisen ja ylläpidon kustannuksia [Stockhammer 2011]. HTTP-protokollan käyttö verkossa olevan sisällön siirtämiseen on jo vakiinnuttanut asemansa sen yksinkertaisuudellaan ja luotettavuudellaan.

HTTP-pohjaiset sisällönjakeluverkot vähentävät myös kotipalvelimien (origin server) kuormitusta ja tarjoavat pienempää viivettä käytettävyydessä [Stockhammer 2011].

Olemassaolevan sisällönjakeluverkon lisäksi HTTP:n käyttö mahdollistaa olemassa olevien HTTP-palvelimien, -välimuistien (cache) ja välityspalvelimien (proxy server) hyödyntämisen. Tällainen infrastruktuuri tuo mukanaan suuria etuja mm. palvelun saatavuuteen (availability), viiveen (latency) pienentämiseen ja skaalautuvuuteen (scalability). Sen lisäksi, että HTTP mahdollistaa olemassa olevan infrastruktuurin hyödyntämisen, pystyy se välttämään verkossa olevien NAT:iien (Network Address Translation) ja palomuurien (firewall) läpikäyntiongelmia (traversal issue). HTTP-liikenne pystyy siis kulkemaan verkon läpi varsin vaivattomasti, minkä seurauksena palvelun saatavuus ei heikene. Tämä perustuu siihen, että HTTP käyttää kiinteää porttinumeroa 80, kun taas mediaprotokollat ovat tavanomaisesti käyttäneet UDP-pistokkeita (socket) erikoisilla porttinumeroilla. Internet on rakennettu HTTP:tä silmällä pitäen, joten “uusien” mediaprotokollien optimaalinen käyttö voisi vaatia muutoksia olemassa olevaan infrastruktuuriin, mikä tulee varsin kalliiksi.

HTTP-suoratoisto on siis varsin looginen valinta suoratoistopalveluiden tarjoajille. Sen käyttöönotto ja operointi on taloudellisesti kannattava vaihtoehto, sillä olemassa oleva infrastruktuuri tarjoaa apua mm. palvelimien kuormitukseen. Olemassa olevan infrastruktuurin hyödyntäminen mahdollistaa myös paremman skaalautuvuuden ja saatavuuden palvelulle. Tämän lisäksi palvelimien kuormitus vähenee, kun suoratoiston kontrollointi voidaan siirtää kokonaan käyttäjän vastuulle [Stockhammer 2011]. Käyttäjä voi kontrolloida suoratoistoa avaamalla TCP-yhteyksiä useille eri HTTP-palvelimille, joista sisältöä voidaan ladata.

Monet suurimmista palveluntarjoajista ovat kuitenkin jättäneet taakseen “tavallisen” HTTP-suoratoiston ja asteittaisen latauksen. Uudeksi vallitsevaksi trendiksi on noussut mukautuvien suoratoistomenetelmien käyttö, joista käytetyimmäksi on noussut MPEG-DASH.

4.2 Dynaamisesti mukautuvat suoratoistomenetelmät

Nykyaikaiset videoiden suoratoistopalvelut nojaavat dynaamisesti mukautuviin algoritmeihin [Spiteri 2018]. Näiden algoritmien avulla palvelut pystyvät tarjoamaan käyttäjilleen sujuvan suoratoistokokemuksen ilman ylimääräisiä viivästyksiä toiston aikana. Algoritmia suoritettaessa, videosoitin pystyy säätelemään, minkä suuruisella bittivirralla soitin vastaanottaa toistettavan sisällön videosegmenttejä.

Bittivirran suuruuteen vaikuttavat vallitsevien verkon olosuhteiden lisäksi myös käyttäjän tekemät valinnat esimerkiksi videon laadun suhteen [Sodagar 2011]. Näiden ominaisuuksien perusteella, soitin säätää bittivirran halutulle tasolle ja pyrkii ylläpitämään parasta mahdollista laatua. Parhaan mahdollisen käyttökokemuksen tarjoamiseksi, dynaamisesti mukautuvien algoritmien tulisi ylläpitää mahdollisimman suurta bittivirtaa ilman puskurointiviiveitä [Spiteri 2018]. Toiston aikana tapahtuvat laatumuutokset vaikuttavat myös negatiivisesti suoratoistopalvelun käyttökokemukseen.

Mukautuvat suoratoistoalgoritmit hyödyntävät mallia, jossa videosisältö on jaettu segmentteihin. Jokainen segmentti pitää sisällään määritellyn, noin muutaman sekunnin mittaisen, osan toistettavasta videosta. Palvelevalla palvelimella on useita eri laatuversioita jokaisesta videosegmentistä, joiden avulla palvelun saatavuus pysyy korkealla laitteistosta tai verkon tilasta huolimatta. Esimerkiksi verkon ruuhkautuessa, videosoitin voi pyytää heikompaa laatua olevia segmenttejä, jotta videon toisto ei keskeydy. Ruuhkan hälvennyttyä soitin voi pyrkiä jälleen nostamaan laatua korkeammaksi. Mukautuvat suoratoistopalvelut voivat tehdä tarvittaessa muutoksia toistettavan laadun suhteen jokaisen segmentin kohdalla. Laadun vaihtelu jokaisen segmentin kohdalla ei ole kuitenkaan suotavaa, sillä se heikentää katselukokemusta [Mok 2011]. Liialliset laatumuutokset saavat palvelun vaikuttamaan epävakaa. Jotta suurimmilta laatuvaihteluilta ja toiston viivästyksiltä kuitenkin vältyttäisiin, hyödyntävät soittimet puskureita.

Puskurin avulla videosoitin pystyy varastoimaan määritellyn puskurikoon verran videosegmenttejä varastoon [Spiteri 2018]. Soitin pyytää palvelimelta tulevia segmenttejä ennen toistoa ja varastoi ne puskuriin odottamaan toistovuoroa. Mikäli puskuri kerkeää tyhjentymään eikä uusia videosegmenttejä ole toistettavaksi, keskeytyy videon toisto puskuroinnin ajaksi. Soitin jatkaa videon toistoa, kun uusi segmentti on vastaanotettu. Tätä kaavaa jatketaan, kunnes video on toistettu kokonaisuudessaan.

Mukautuvaa bittivirtaa hyödyntävät algoritmit voidaan jakaa kolmeen kategoriaan: Läpisyöttöön nojaaviin -, puskurointiin nojaaviin - ja näiden risteymäalgoritmeihin [Spiteri 2018]. Yleisesti ottaen mukautuvat algoritmit toimivat samojen raamien sisällä, mutta kategorisointi perustuu siihen, missä algoritmin painotus on.

Läpisyöttöön nojaavat algoritmit perustuvat verkon palvelutason arviointiin [Spiteri 2018]. Tämän arvion perusteella tehdään päätös seuraavan segmentin lataukseen käytettävästä bittivirrasta. Arvio suoritetaan jokaisen segmentin kohdalla, joten jokainen

segmentti on parasta mahdollista laatua. Tällaista menetelmää hyödyntäviä algoritmeja ovat mm. Squad, Festive ja PANDA. Kuten aikaisemmin mainittu, jokaisen segmentin kohdalla tehty arvio voi johtaa tilanteeseen, jossa katselukokemus kärsii liiallisten laatu-muutoksien johdosta.

Puskurointiin nojaavat algoritmit tarkastelevat puskurin yleistä laatutasoa ja tekevät tämän pohjalta päätöksen seuraavan segmentin bittivirrasta [Spiteri 2018]. Puskuri toimii ikään kuin otoksena verkon tilasta. Mikäli valtaosa puskurin segmenteistä on korkealaatuisia, on verkon läpisyöttö kapasiteetti todennäköisesti myös korkealla. Jos puskurissa olevat segmentit ovat puolestaan heikkolaatuisia, on verkko todennäköisesti kykenemätön siirtämään korkean bittivirran segmenttejä palvelimelta soittimelle. Puskurin suuruudesta riippuen tällaiset algoritmit ovat todennäköisesti hitaita reagoimaan verkon muutoksiin. Esimerkiksi yllättävät verkon ruuhkautumiset (flash crowds) voivat aiheuttaa tilanteen, jossa puskuuri on täynnä korkean bittivirran segmenttejä, jolloin tehdyn arvion mukaan verkon läpisyöttö on varsin korkealla tasolla. Näin ollen soitin pyytää korkean bittivirran videosegmenttiä. Pahimmassa tapauksessa videosegmentin siirron aikana puskuuri tyhjenee, jonka jälkeen vasta soitin reagoi verkon heikkoon läpisyöttöön.

Hybridi-algoritmit puolestaan nojaavat kumpaankin edellä mainittuun tekniikkaan [Spiteri 2018]. Näiden algoritmien toiminta perustuu siis sekä verkon analyysiin että puskuuriin varastoitujen videosegmenttien bittivirtojen yleiseen tasoon. Näiden pohjalta tehdään päätös ladattavien segmenttien bittivirran suhteen. Tällaisia algoritmeja ovat mm. ELASTIC ja MPC.

Videopalveluiden tarjoajilla ja media-alan toimijoilla on kuitenkin vaatimuksia käytettävälle mukautuville suoratoisto algoritmeille [Spiteri 2018]. Näihin vaatimuksiin kuuluvat videoiden toistaminen mahdollisimman korkealla bittivirralla ilman suuria muutoksia laadussa, puskuuriinviiveiden välttäminen, toistolaadun vakaus, nopea reagointikyky sekä verkon että käyttäjän aiheuttamiin muutoksiin ja mahdollisimman pieni viive aikakriittisissä lähetyksissä (live streaming).

Suosittu mukautuvat suoratoistomenetelmät ovat tietotekniikka-alan isoimpien tekijöiden nimissä. Näihin lukeutuvat mm. Microsoft Smooth Streaming, Apple HTTP Live Streaming sekä Adobe HTTP Dynamic Streaming. Näiden tekijöiden mukanaolo mediasisällön suoratoistomenetelmien kehityksessä ei liene yllätys, ottaen huomioon palvelut, joita ne tarjoavat kuluttajilleen.

Microsoft Smooth Streaming on implementoitu MP4-tiedoston ympärille, joka nojaa H.264 videokoodekkiin [Zambelli 2009]. H.264-videokoodekki on implementoitu suureen osaan Microsoftin palveluista, minkä vuoksi tämän videokoodekin tukeminen myös suoratoistopalvelussa on looginen valinta. Microsoft Smooth Streaming:ssä MP4-tiedosto siirretään käyttäjälle laatikoissa (box), jotka ovat pieniä palasia kokonaisesta MP4-tiedostosta. Tämä Microsoftin versio HTTP-suoratoistosta toimii kuin suoratoisto, mutta on toteutettu oikeastaan asteittaisen latauksen (progressive download) ympärille. Jokainen ladattava osuus on kuin yksittäinen, lyhyt, MP4-tiedosto, joka suoritetaan sen saapessa soittimelle. Soitin hakee seuraavia toistettavia segmenttejä yhdistämällä toistokohdan ajan manifestitiedoston URL-tietoihin [Fisher 2014].

Apple HTTP Live Streaming toimii pääasiassa samalla tavalla kuin Microsoft Smooth Streaming -suoratoistomenetelmä. Yhtenä huomattavimpana erona aikaisemmassa versiossa oli kuitenkin tähän suoratoistomenetelmään implementoitu multipleksointi [Apple 2016]. Multipleksoinnissa kaikki siirrettävät mediakanavat käytännössä yhdistyvät yhdeksi bittivirraksi, minkä seurauksena esimerkiksi puhuttua kieltä vaihdettaessa jokainen bufferoitu videosegmentti tuli toimittaa käyttäjälle uudelleen, toisella kielellä varustettuna. Tämän menetelmän hyöty on kuitenkin sen ”heikkoudessa”; soittimen ei tarvitse huolehtia kuin yhdestä virrasta ja toistaa vastaanotetut .mov-tiedostot. Apple siirtyi kuitenkin tukemaan segmentoituja MP4-tiedostoja vuonna 2016 tulleen muutoksen myötä.

Adobe HTTP Dynamic Streaming suorittaa Microsoft Smooth Streamingin tavoin hajotettuja MP4-tiedostoja. Adoben suoratoistomenetelmässä puolestaan manifestitiedosto ladataan uudestaan jokaisen suoritettavan segmentin kohdalla, kuten Applen HTTP Live Streamingissäkin. Tämän tiedon avulla, soitin pystyy hakemaan seuraavaa toistettavaa mediakokonaisuutta. Vaikka videoiden metatiedot ovat pieniä verrattuna videotiedoston kokoon, lisää tämä yleiskustannuksia siirrolle [Fisher 2014]. Adobe HTTP Dynamic Streaming käyttää edelleenkin multipleksointia, minkä seurauksena useaa äänikanavaa ei tueta tämän suoratoistomenetelmän puolesta. Tämän lisäksi Adobe HTTP Dynamic Streaming nojaa Flash-teknologiaan, joka on alkanut menettää suosiotaan tietoturvaheikkouksiensa puolesta. Esimerkiksi Google Chrome on luopumassa kokonaan Flash-teknologian tukemisesta, minkä vuoksi tämän suoratoistomenetelmän tulevaisuus, tukevan infrastruktuurin kannalta, on vaakalaudalla [Laforge 2017].

4.3 MPEG-DASH

Vuonna 2012 ISO hyväksyi uuden standardin mukautuvalle suoratoistolle, MPEG-DASH eli Dynamic Adaptive Streaming over HTTP (DASH). DASH on HTTP:n yli toimiva mukautuva suoratoisto standardi, joka hyödyntää TCP:tä. DASH:llä on erittäin suuri potentiaali kasvaa suurimmaksi suoratoistomenetelmäksi, sillä DASH-soittimen (dash.js) kehityksestä ja ylläpidosta vastuussa oleva päätäntäelin, DASH-IF (DASH Industry Forum) koostuu alan suurimmista tekijöistä, kuten Akamaista, Googlesta, Netflixistä, Microsoftista, Hulusta, Nokiasta, Samsungista ja monista muista [DASH 2019]. Tästä syystä DASH:stä onkin tullut yksi alan standardeista ja se on implementoitu moniin suurimpiin videoiden suoratoistopalveluihin, kuten Netflixiin, Huluun ja YouTubeen.

DASH tukee median suoratoistoa tavallisilta verkkopalvelimilta, joilla ei ole erityisiä ominaisuuksia [DASH 2014]. Pyynnöt voidaan suorittaa siis tavallisilla HTTP-pyyntöillä. Koska sisältö on määritelty täysin XML- ja binäärimuodoissa, voidaan sitä säilyttää olemassa olevan verkkoinfrastruktuurin HTTP-välimuisteissa. Toimijoiden välinen tiedonsiirto muistuttaa siis hyvin paljon tavallista verkkomateriaalin siirtoa. Käyttäjä pyytää saamansa Media Presentation Description -tiedoston (MPD) perusteella, haluttua materiaalia oikealta palvelimelta.

DASH:in sisältö koostuu median esitystavoista (media presentation) [DASH 2014]. Nämä esitystavat rakentuvat puolestaan mediasisällöstä (media content) tai sen ajallisista jaksoista (period). Nämä jaksot puolestaan rakentuvat mediakomponenteista (media component), joita ovat esimerkiksi erilaatuiset videot, ääniraidat sekä tekstitykset. Jokaisella mediakomponentilla on oma tyyppinsä, jonka mukaan se voidaan jaotella esimerkiksi videoksi tai ääniraidaksi.

Saman mediasisällön ominaisuudet periytyvät siihen kuuluville mediakomponenteille [DASH 2014]. Esimerkiksi jokainen erilaatuinen video-osuus kuuluu tietyn mediasisällön tiettyyn jaksoon. Tämä metatieto on välttämättömyys jokaiselle mediakomponentille, jotta DASH:n olennainen toiminnallisuus, eli mukautuminen muutoksiin olisi mahdollista. Ilman yhdistäviä metatietoja olisi mahdotonta vaihtaa kahden eri videolaadun välillä ilman suurempia katkoksia. Tämän metatiedon perusteella asiakasohjelma voi valita esimerkiksi olosuhteisiin sopivan videolaadun MPD-tiedostosta. MPD-tiedostosta kuitenkin tarkemmin myöhemmällä aliluvulla.

DASH määrittelee kaksi erilaista aikajanaa mediasisällön suoratoiston aikana [DASH 2014]. Ensimmäinen määritellyistä aikajanoista on mediakomponenttien määrittelemä yhteinen aikajana. Tähän kuuluvat jokaisen mediaesityksen kuuluvat mediakomponentit. Jokaisen komponentin esitysaika on kirjattuna globaaliin aikajanaan, jotta vaihdos erilaisuisen sisällön välillä voidaan suorittaa saumattomasti ja tehokkaasti. Tätä aikajanaa kutsutaan mediaesityksen aikajanaksi. (Media Presentation Timeline).

Toinen aikajana kertoo puolestaan asiakasohjelmalle kunkin segmentin saatavuusajat MPD-tiedostossa ilmoitetusta URL-osoitteesta. Tämän perusteella asiakasohjelma voi valikoida haettavan sisällön sen perusteella, mikä palvelin palvelee lähimpänä käyttäjää kunkin aikana. Tämän segmentin saatavuusajan (Segment Availability Time) perusteella asiakasohjelma voi välttää segmenttipyyntöjä palvelimelta, jolla kyseistä segmenttiä ei enää ole saatavilla.

MPEG-DASH:n suoratoistossa välitetään mediasisältö mediasegmenteissä [DASH 2014]. Mediasegmentit voivat olla pituudeltaan eri mittaisia ja segmenttien pituus vaihtelee usein suoratoistettavan sisällön perusteella. Mediasisällössä, jonka suoratoisto vaatii pientä viivettä, käytetään pääsääntöisesti lyhyempiä segmenttejä. VOD-sisällön (Video On Demand) suoratoistossa voidaan puolestaan hyödyntää pidempiä segmenttejä. Lyhyitä mediasegmenttejä käytetään esimerkiksi reaaliaikaisen materiaalin suoratoistossa.

DASH:n mediasegmentit ovat pieniä objekteja, joita käsitellään yksittäisinä, omina ja riippumattomina entiteetteinä [DASH 2014]. DASH ei tue kuitenkaan segmenttien pituuksien muokkausta ajan kuluessa. Yhden segmentin pituuden muuttaminen vaikuttaisi muiden segmenttien alkamisajankohtiin. Tämä toisi synkronoinnin kannalta ongelmia suoratoiston yhteydessä. Pahimmassa tapauksessa yhden segmentin pituuden muokkaaminen vaikuttaisi kaikkien tulevien suoratoistossa siirrettävien mediasegmenttien metatietoihin.

DASH-suoratoiston mediasegmentit sisältävät täsmällisen tiedon segmentin toistoajasta [DASH 2014]. Tämän tiedon perusteella on mahdollista synkronoida suoratoiston ääni-, kuva- ja tekstityksraidat. Näin ollen verkon olosuhteiden muuttuessa on mahdollista vaihtaa toistettavan sisällön laatua ilman synkronointiongelmia.

DASH-suoratoisto toimii sekä VOD- että reaaliaikaisen videosisällön suoratoistamiseen [DASH 2014]. Näiden kahden eri sisällön suoratoistossa on kuitenkin huomattavia eroja,

jotka juontavat juurensa suoratoistettavan sisällön perimmäiseen olemukseen, videon valmiuteen.

Reaaliaikaisen sisällön MPD-tiedosto on dynaaminen, mikä poikkeaa tavallisesta MPD-tiedoston tyypistä, joka on staattinen [DASH 2014]. MPD-tiedoston dynaamisuus johtuu siitä, että reaaliaikaista sisältöä ei ole etukäteen tuotettu. Näin ollen käyttäjän ladattavat mediasegmentit tulee lisätä dynaamisesti MPD-tiedostoon. Tavanomaisesti videoiden suoratoiston yhteydessä MPD-tiedostoon on kirjattu valmiiksi ladattavat mediasegmentit ja URL:t, joista segmentit ovat ladattavissa. Dynaamiseen MPD-tiedostoon voidaan puolestaan määrittää, millaisella päivitysjaksoituksella uusien segmenttien tiedot päivitetään. Mikäli sisältö ei ole toistettavissa reaaliaikaisen lähetyksen jälkeen, voidaan vanhojen segmenttien tiedot poistaa, kun ne eivät enää ole saatavilla. Tätä MPD-tiedoston dynaamisuutta tukee myös HTTP 1.1:n mukana saapunut tuki lohkotetulle lähetykselle (chunked transfer encoding). Myös DASH tukee tätä ominaisuutta tarkoin määritellyistä segmenteistään huolimatta.

Lohkotettu lähetys mahdollistaa pienemmän viiveen lähettäjältä vastaanottajalle [DASH 2014]. Tämän avulla lähettäjän on mahdollista pakata lähetettäviä segmenttejä ennaltamääritellyn kokosiin lohkoihin ja välittää nämä lohkot vastaanottajalle [HTTPWatch 2020]. Tämä on hyödyllistä tilanteissa, joissa vastaanottajan pyynnön vastaukselle ei ole ennaltamääriteltyä kokoa. Näin ollen vastaanottajan soitin voi toistaa sisältöä, jota ei ole kokonaan ladattu vastaanottajalle [Akamai 2020]. Jokainen lähetettävä segmentti jaetaan siis lohkoihin, jotka voidaan välittää yksi kerrallaan vastaanottajalle koodauksen (encoding) jälkeen. Toistettava sisältö saadaan siis nopeammin jakoon, ilman suurempaa viivettä suuremman segmentin koodauksesta ja siirrosta. Tiedostonsiirto keskeytyy vasta, kun vastaanottaja vastaanottaa tyhjän lohkon, jonka koko on 0 tavua. Näin ollen yhteys lähettäjän ja vastaanottajan yllä pysyy käynnissä koko sen ajan kuin dynaamisesti tuotettua sisältöä on lähetettäväksi. Tämä on etenkin hyödyllistä reaaliaikaisen suoratoiston yhteydessä, jolloin toistettavan sisällön koosta ei ole tietoa ennalta.

4.4 MPD-tiedosto

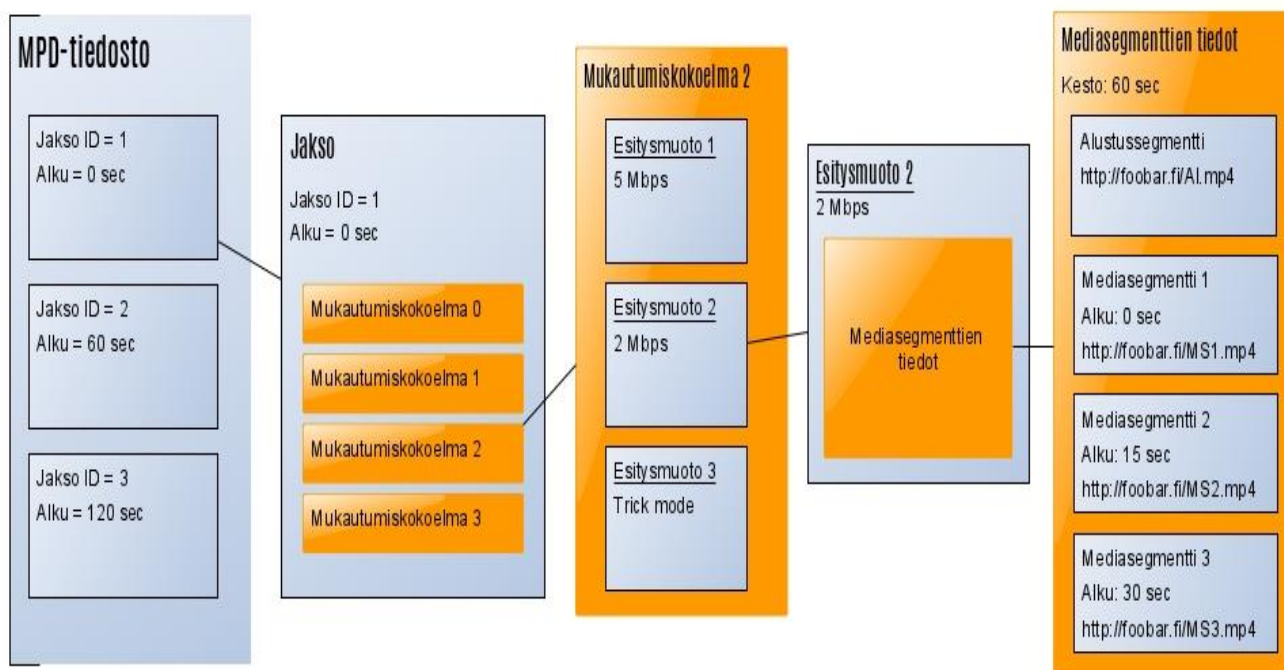
Suoratoiston aikana asiakasohjelma voi esimerkiksi verkon olosuhteiden tai käyttäjän valintojen perusteella joko heikentää tai parantaa videon laatua. Käyttäjä voi lisäksi valita useiden eri tekstityksien välillä tai vaihtaa puhuttua kieltä. Näitä toiminnallisuuksia ja niiden hallinnointia varten suoratoistaminen vaihtelevan bittivirran välityksellä vaatii manifestitiedoston, eli DASH:n tapauksessa MPD-tiedoston (Media Presentation Description) [Sodagar 2011].

Ennen suoratoiston käynnistymistä palvelin lähettää asiakasohjelmalle MPD:n, joka sisältää tiedot käytettävistä ääni- ja videosisällön laatuvalikoista, puhutuista kielistä ja tekstityksistä [Sodagar 2011]. Tiedot on kirjattu XML-dokumenttiin, hierarkkiseen tietorakenteeseen, jonka perusteella käyttäjäohjelma saa haltuunsa tarvittavat tiedot sisällön lataamiseen ja toistamiseen. MPD:n avulla käyttäjäohjelma oppii muun muassa, millaiseen tiedostomuotoon sisältö on tallennettu, millaisiin segmentteihin tiedostot on jaettu ja mistä sijainneista eri laatuiset sisällöt löytyvät. MPD voidaan välittää asiakasohjelmalle HTTP-yhteyden muodostuksen yhteydessä tai esimerkiksi sähköpostin välityksellä.

MPD-tiedoston saatuaan asiakasohjelma jäsentelee vastaanotetun tiedon. MPD:n tarjoaman tiedon, käyttäjän tekemien valintojen ja verkon olosuhteiden perusteella, MPD ohjaa asiakasohjelman oikeiden lähteiden äärelle. Näin asiakasohjelma voi käynnistää sisällön lataamisen tehtyjen valintojen ja vallitsevien olosuhteiden mukaisesti.

MPD-tiedoston hierarkia muodostuu jaksoista (periods), mukautumiskokoelmista (adaptation set), esitysmuodoista (representation) ja segmenteistä (segment) [Sodagar 2011]. MPD:hen kirjatut jaksot ovat ajallisia osuuksia toistettavasta mediasisällöstä. Jokaisella jaksolla on oma paikkansa toistojärjestyksessä. Näin ollen jokaisella jaksolla on oma alku- ja loppuaikansa sekä määritelty kesto. Tämän lisäksi jokainen jakso on jaettu sisäisesti yhteen tai useampaan mukautumiskokoelmaan. Tätä hierarkiaa havainnollistaa kuva 4.1.

Mukautumiskokoelma pitää sisällään jonkin auditiivisen tai visuaalisen median esitysmuotojen tiedot [Sodagar 2011]. Tämän tiedon avulla asiakasohjelma voi valita esimerkiksi heikomman esityslaadun toistettavalle jaksolle, mikäli olosuhteet tätä vaativat. Mukautumiskokoelman avulla asiakasohjelma näkee, millaisia esitysmuotoja kyseiselle jaksolle on tarjolla, kunkin median osalta. Mukautumiskokoelmat voivat pitää sisällään myös trick mode-esitysmuodon tiedot.



(Kuva 4.1: Esimerkki MPD-tiedoston sisäisestä hierarkiasta)

Trick mode on esitysmuoto, joka mahdollistaa katsottavan median toistokohdan siirtämistä haluttuun kohtaan esikatselukuvien perusteella. Trick mode -esitysmuoto pitää sisällään tietyn jakson esikatselukuvan, joka näyttää käyttäjälle, missä kohdassa toistokohta kulloinkin on. Trick mode ei sisällä äänimediaa, joten tämä on varsin kevyt esitysmuoto. Trick mode tarjoaa ainoastaan visuaalisen vastineen aikaisemmista mediatyypeistä tutulle kelausominaisuudelle.

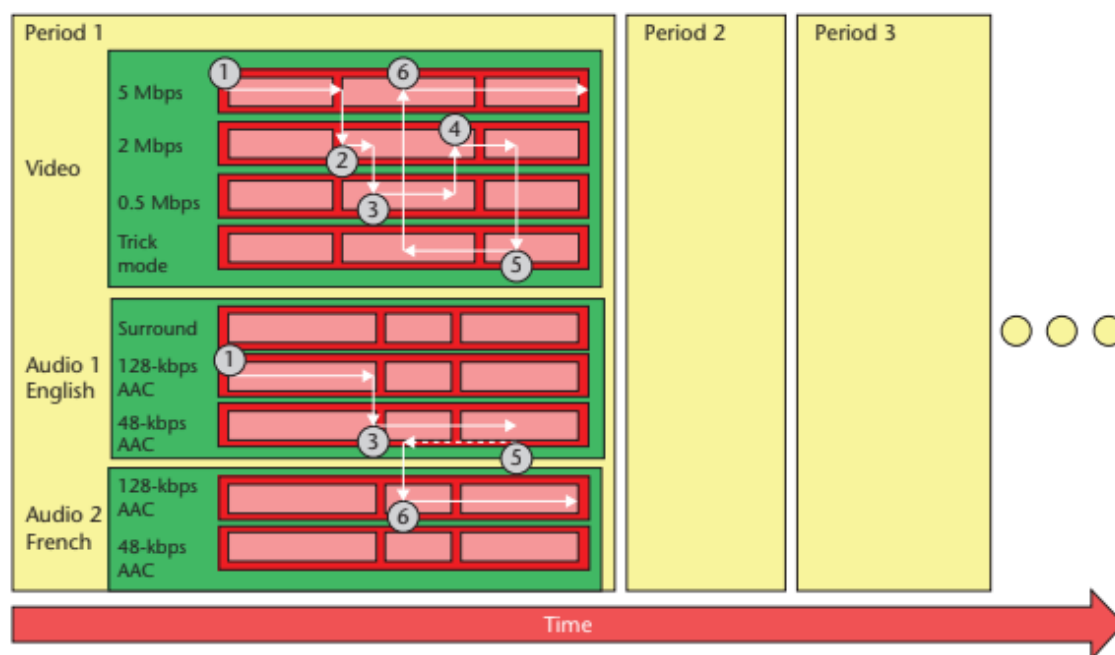
Sopivan esitysmuodon kautta asiakasohjelma pääsee käsiksi kyseisen esitysmuodon segmentteihin. Segmentit sisältävät tiedot pienestä osasta kokonaista jaksoa, kyseisen median valitulla laadulla. Segmentin tiedot sisältävät URL:n, jonka avulla asiakasohjelma voi ladata kyseisen segmentin valitulla laadulla. Segmentin lataus tapahtuu tavallisella HTTP GET -pyynnöllä, joka välitetään palvelun HTTP-serverille [Sodagar 2011].

MPD:n rakenteen vuoksi sovituskokoelman sisältämä tieto sisältää ainoastaan esitysmuotovaihtoehdot jollekin medialle tai trick mode:lle. Esimerkiksi jaksoa toistettaessa on mahdollista, että äänenlaatua ei tarvitse pienentää verkon olosuhteiden muuttuessa. Voidaan tulla tilanteeseen, jossa videon laadun heikentämisellä saadaan tarvittavasti vapautettua verkon resursseja, jolloin ääntä voidaan suoratoistaa paremmalla laadulla.

4.5 Mukautuminen muuttuviin olosuhteisiin

Mukautuvien suoratoistomenetelmien tuoma dynamiikka tarjoaa käyttäjille paremman käyttökokemuksen mm. lyhyempien ja harventuneiden puskurointiviiveidenmuodossa. Mitä tällainen sopeutuminen pitää käytännössä sisällään? Millaiset tekijät vaikuttavat suoratoiston olosuhteisiin? Muun muassa näitä kysymyksiä tarkastellaan Netflix-nimisen suoratoistopalvelun näkökulmasta.

Netflix-palvelun suoratoisto tapahtuu käyttäen DASH-suoratoistomenetelmää [Bitmovin 2019]. Tämä jakaantuu käytännössä kolmeen osaan: sisällön lähetyks, välitys ja vastaanotto. Jokaisessa tämän ketjun vaiheessa on mahdollisuus tapahtua olosuhteiden muutoksia, joiden seurauksena suoratoistomenetelmän on reagoitava ja mukautettava toimintaansa. Yksinkertaisimmillaan videon suoratoisto etenee mukautuvassa suoratoistomenetelmässä kuvan (4.2) esittämällä tavalla [Sodagar 2011].



(Kuva 4.2: Mukautuminen muuttuviin olosuhteisiin ajan edetessä [Sodagar 2011])

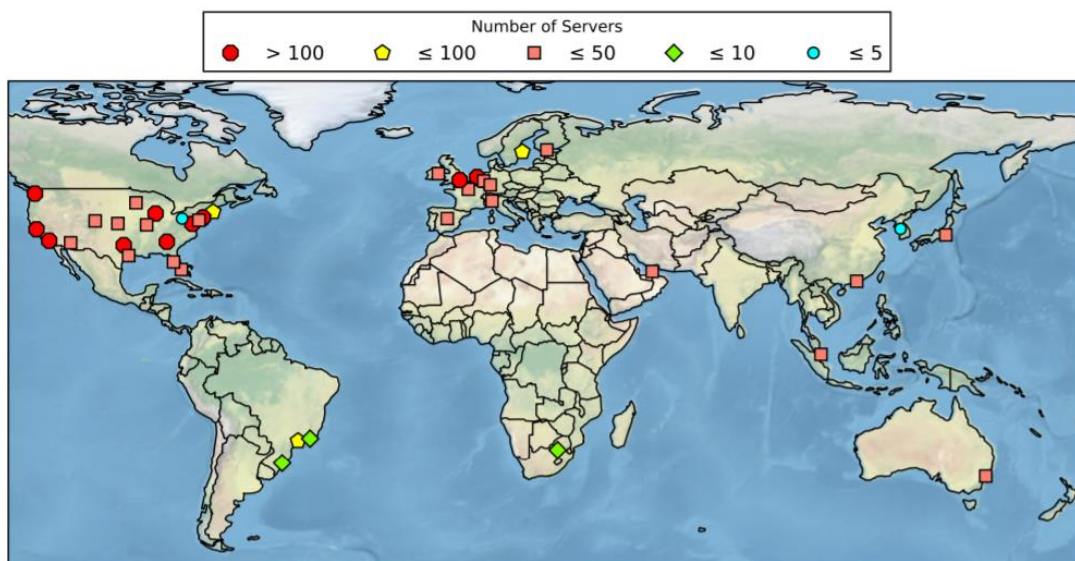
Kuvasta 4.2 voidaan nähdä seuraavat vaiheet:

1. Käyttäjä aloittaa suoratoistettavan videosisällön katsomisen laadulla, jonka välittämiseen vaaditaan 5 Mbps:n suuruista kaistaa. Äänenlaatu on myös korkeimmalla tasolla, englanninkielisenä.

2. Verkon käytettävästä kaistasta johtuen videon laatu pudotetaan heikompaan laatuun, jotta suoratoisto voi jatkua ilman puskurointiviiveistä johtuneita katkoksia.
3. Laitteen tarkkaillessa verkon tilaa huomataan, että käytettävissä oleva kaista onkin oletettua pienempi, minkä seurauksena videon ja äänen laatua pudotetaan entisestään. Suoratoistettavaa videomateriaalia välitetään katsojalle 0,5 Mbps siirtonopeudella ja äänen siirtonopeus tiputetaan 48-kbps-tasolle.
4. Käytettävissä oleva kaista kasvaa, jolloin videon laatua voidaan nostaa seuraavalle tasolle. Suoratoistoa jatketaan 2 Mbps:n siirtonopeudella.
5. Käyttäjä päättää siirtää toistokohdan taaksepäin, hyödyntäen trick mode -nimistä ominaisuutta. Tämän lisäksi katsoja päättää vaihtaa katsottavan sisällön puhutun kielen englannista ranskaan.
6. Suoratoistoa jatketaan parhaalla mahdollisella kuvan- ja äänenlaadulla, ranskan-kielisenä.

Netflix varastoi suoratoistettavan materiaalin oman sisällönjakeluverkkonsa solmuihin [Böttger 2018]. [Böttger 2018]:ssä kuvaillun tutkimuksen yhteydessä huomattiin, että Netflix käyttää pyyntöjen ohjauksissa verkkotunnuksia, jotka ovat liian yksityiskohtaisia ollakseen peräisin yleisesti käytetyn sisällönjakeluverkon DNS-pohjaisesta (Domain Name Service) ohjaustavasta. Netflixin käyttämät verkkotunnukset pitävät sisällään mm. tietoja palvelimien fyysisistä sijainneista.

Netflix on levittänyt palvelimensa keskitetysti tietyille alueille. Tavallisen verkkoliikenteen palvelimet on suotuisampaa levittää kattamaan mahdollisimman suurta aluetta, mutta videosuoratoiston kuormittavan luonteen vuoksi tämä ei ole kannattavaa [Böttger 2018]. Videosuoratoiston palvelutason ylläpitämiseksi on kannattavampaa sijoittaa palvelimet keskitetysti suurien käyttäjämäärien ääreen, kuten Netflix on myös tehnyt. Tätä levittäytymistä havainnollistaa kuva 4.3, jossa nähdään Netflixin ylläpitämien palvelimien sijainnit ja määrät IX-pisteissä (Internet Exchange Point). Nämä ovat fyysisiä sijainteja, joissa Internetin palveluntarjoajat vaihtavat verkon liikennöitävää sisältöä palveluntarjoajalta toiselle.



(Kuva 4.3: Netflixin ylläpitämien palvelimien sijainnit ja lukumäärät IX-pisteissä)

Netflixin suuruiselle palveluntarjoajalle ei pitäisi muodostua pullonkauloja yksittäisille palvelimille käytössä olevan kaistan suuruuden sekä heidän käyttämiensä kuormantausmekanismien johdosta. Pienempien palveluntarjoajien puolesta lähetysvaiheessa voi tapahtua odottamattomia ruuhkautumisia, joiden seurauksena lähettäjä voi joutua rajoittamaan lähetettävän videosisällön laatua. On myös mahdollista, että sisältöä hallussa pitävä palvelin kaatuu kesken lähetyksen, minkä seurauksena sisällön välitys tulee ohjata toiselle palvelimelle, jonka käytettävä kaistan leveys ei ole alkuperäisen palvelimen tasolla. Muissa kuin virhe- tai ruuhkautumistilanteissa, ei lähettäjän tulisi heikentää lähetettävän sisällön laatua. Laadun heikkous heijastuu palvelun käyttökokemuksessa negatiivisesti [Mok 2011].

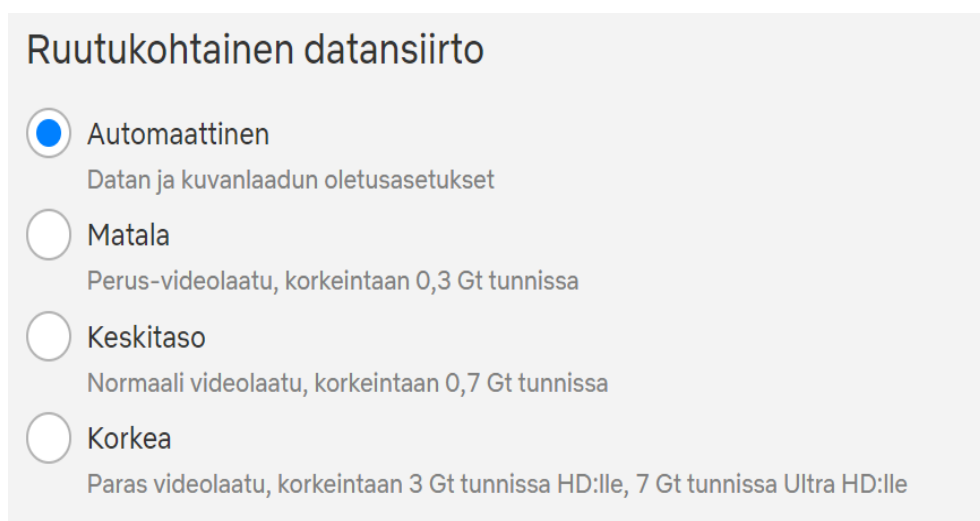
Toistettavan sisällön välitys käynnistyy käyttäjän puolen DASH-asiakasohjelman (client) ottaessa yhteyden palvelimeen ja pyytäen tältä MPD-tiedostoa, jonka avulla päästä mediasegmentteihin käsiksi. Palvelin välittää asiakasohjelmalle tämän pyytämän MPD-tiedoston, jonka asiakasohjelma jäsentää (parse). Tämän jäsenneilyn tiedon avulla DASH-asiakasohjelma voi aloittaa suoratoiston lataamalla mediasegmenttejä MPD-tiedostossa ilmoitetuista osoitteista. Asiakasohjelman tehtäväksi jää media segmenttien sijoittaminen puskuriin ja verkon viiveen vaihtelun laskeminen säännöllisesti [Kesavan 2015]. Tämä viiveen vaihtelun tarkkailu tapahtuu mukautumismoduulin avulla.

Mukautumismoduulin tehtäviin kuuluu viiveenvaihtelun tarkastelun lisäksi oikean

laatuisen esitysmuodon valitseminen seuraaville segmenteille sekä tämän valitun esitysmuodon lataamisen käynnistys palvelimelta [Kesavan 2015]. Tämä prosessi jatkaa toimimista niin pitkään kunnes viimeinen mediasegmentti on välitetty vastaanottajalle ja yhteys palvelimeen voidaan purkaa.

Verkko on jatkuvasti “elävä” toimija, joten yksittäisten solmujen kaatumiseen ja uusien reittien muodostamiseen on varauduttava. Tällaiset vikatilat heijastuvat myös suoratoiston laatuun, sillä uuden reitin muodostaminen voi johtaa esimerkiksi käytettävissä olevan kaistan heikkenemiseen. Myös siirron aikana ilmentyvät ruuhkautumiset vaikuttavat mukautuvan suoratoistomekanismin toimintaan. Kaistanleveyden kavetessa, pyritään lähettävän sisällön laatua pienentää tasolle, joka vastaa kaistanleveyttä. Mikäli siirrettävän sisällön kuljettaminen vaaditussa ajassa lähettäjältä vastaanottajalle ei ole mahdollista kaiventuneen kaistan takia, heikennetään laatua niin paljon, että tämä aikavaatimus voidaan toteuttaa. Mukautumismoduuli pitää huolen, että suoratoisto pysyy jatkuvana tekemällä edellä mainittua verkon viiveen tarkkailua.

Netflix tarjoaa käyttäjilleen mahdollisuuden vaikuttaa suoratoistettavan sisällön laatuun, puhuttuun kieleen ja tekstitykseen. Puhutun kielen ja tekstityksen katsoja voi vaihtaa kesken toiston, mutta siirtonopeuden muuttaminen tapahtuu tilin asetusten kautta (ks. kuva 4.4). Tästä johtuen kuvanlaatua ei voi käyttäjän toimesta vaihtaa kesken suoratoiston, sillä asetuksiin siirtyminen katkaisee suoratoiston.



(Kuva 4.4: Netflixin tarjoamat asetukset kuvanlaadun vaihtamiseksi [Netflix 2020 (2)])

Käyttäjän tehdessä muutoksia suoratoiston aikana esimerkiksi vaihtamalla toistettavan sisällön puhuttua kieltä tai muuttamalla tekstityksiä, reagoi DASH-asiakasohjelman mukautumismoduuli näihin muutoksiin. Mukautumismoduulin pyytäessä palvelimelta ääniraita- tai tekstitysegmenttejä, sovelletaan näitä käyttäjän tekemiä valintoja pyyntöjen yhteydessä. Tämän kaltainen mukautuminen jatkuu aina, kunnes viimeinen segmentti suoratoistettavasta sisällöstä on toimitettu käyttäjälle ja yhteys käyttäjän ja palvelimen välillä voidaan katkaista.

4.6 Virheiden hallinta ja palautuminen

Videoiden suoratoisto on verkkoa kuormittava prosessi. Siirrettävät tiedostot ovat suuria kokonaisuuksia, jotka välitetään verkon läpi lähteestä käyttäjälle kaiken muun verkossa olevan liikenteen joukossa. Verkon ruuhkautumisesta tai muista verkossa olevista epävakauksista johtuen siirrettäviä videosegmenttejä tai osia ääniraidoista voi esimerkiksi kadota matkan varrella. Videoiden suoratoiston luonteeseen kuitenkin kuuluu jaettavan sisällön toistaminen sitä mukaa, kun toistettavaa materiaalia on saatu kokonaisuudessaan siirrettyä ennalta määritelty osuus lähettäjältä vastaanottajalle. Näin ollen kadonneiden segmenttien uudelleen lähettämiseen ei kaikissa tapauksissa ole aikaa. Millaisilla keinoilla voidaan tällaisista virhetilanteista siis palautua niin, että palvelun käyttökokemus ei heikkene merkittävästi?

Internet ei tunnetusti pysty takaamaan pakettien välitystä lähettäjältä vastaanottajalle. Tämän ominaisuuden seurauksena Internet voi tarjota vain parhaan yrityksensä pakettien välittämiseksi. Internet ei anna takeita kaistan suuruudesta (bandwidth), viiveen vaihte- lusta (delay jitter) tai pakettien katoamismäärästä (loss rate) [Apostolopoulos 2003]. Siirrettävien pakettien hävikki (packet loss) onkin väistämätöntä Internetin luonteen omaisesti [Li 2013]. Tämän vuoksi palveluiden onkin tärkeää pystyä tarjoamaan virheiden hallintaa (error control), jonka avulla pystytään varmistamaan sulava käyttökokemus palvelun käyttäjille virhetilanteiden sattuessa.

Virheiden hallintaa voidaan harjoittaa usealla eri tavalla videoiden suoratoistossa. Näihin lukeutuvat mm. täydentävä virheiden korjaus (Forward Error Correction), virhevakaa koodaus (Error-Resilient Encoding), virheiden peittäminen (Error Concealment) ja viive- rajoitettu uudelleenlähetyt (Delay-Constrained Retransmission) [Li 2013].

Täydentävä virheiden korjaus on yleisesti käytetty virheidenhallintamekanismi, joka perustuu täydentävän tiedon paketoimiseen ja lähettämiseen muihin paketteihin jaettuna. Käytännössä lähettäjä lähettää jokaisen yksittäisen paketin sellaisenaan, mutta tämän lisäksi jokaiseen pakettiin on lisätty ennalta määritelty osuus jonkin toisen paketin sisältämästä mediasisällöstä. Näin ollen, jos alkuperäinen paketti katoaa matkan varrella, voidaan sen sisältämä mediasisältö koota muiden pakettien sisältämän informaation avulla [Li 2013].

Täydentävässä virheiden korjauksessa on omat heikkoutensa. Esimerkiksi erittäin epävakaisissa verkkoolosuhteissa tai äkillisissä verkko ruuhkautumisissa on hyvin todennäköistä, että alkuperäisen paketin kadotessa myös täydentävää mediasisältöä katoaa jonkin toisen paketin katoamisen myötä. Näin ollen kadonnutta sisältöä ei saada rakennettua täydentävän tiedon avulla. Tämän lisäksi täydentävä virheidenkorjaus lisää siirron yleiskustannuksia (overhead), mikä aiheuttaa lisäkuormitusta myös verkolle. Liiallisen lisäkuormituksen välttämiseksi olisikin syytä löytää optimaalinen suhde ylimääräisen sisällön ja alkuperäisen mediasisällön välille kussakin paketissa. Puskurointi viiveiden ollessa kaikista haitallisinta käyttökokemuksen kannalta [Mok 2011] voi kuormitusta välttää lisäämällä ylimääräisenä kulkeva mediasisältö heikommalla laadulla. Paketin kadotessa mediasisältö voidaan tällöin rakentaa ja näyttää käyttäjälle heikommalla laadulla. Näin ollen myös mediasisällön yleiskustannukset pienenevät huomattavasti, täydentävän mediasisällön laadusta riippuen.

Virhevakaan koodauksen tarkoituksena on lisätä paketoitun videon kestävyttä tietyn tyyppisille virheille jo koodaus vaiheessa. Ilmenevät virheet ovat riippuvaisia käytettävän ratkaisun arkkitehtuurista. Näistä virheistä yleisimpiä ovat kuitenkin bittivirran synkronisaation menetys sekä väärään tilaan ajautuminen ja virheiden levittäminen [Apostolopoulos 2003].

Bittivirran synkronisaation menetys viittaa tapahtumaan, jossa bittivirran prosessoinnissa tapahtuva virhe saa dekooderin menettämään tiedon bitin vastaamista parametreista [Apostolopoulos 2003]. Esimerkiksi tilanteessa, jossa määrittelemättömän pitkässä metatieto-kentässä ilmenee bittivirhe voi johtaa suuriinkin ongelmiin. Sen lisäksi, että metatieto-kenttä tulkitaan väärin, myös kaikki seuraavat bitit, jotka sisältävät jaettavan mediasisällön, ovat jokainen väärässä kohdassa. Tämä korjaantuu vasta siinä vaiheessa, kun saavutetaan seuraava synkronointipiste. Tällaisen pienen virheen ilmeneminen kriittisessä vaiheessa voi siis johtaa suuren tietomäärän katoamiseen tai korruptoitumiseen.

Onkin siis erittäin tärkeää havaita synkronoinnin virhe mahdollisimman nopeasti sen tapahtumisesta ja eristää sen vaikutukset, jotta virheen vaikutukset jäävät mahdollisimman pieniksi ja välttyään suuren tietomäärän korruptoitumiselta.

Uudelleensynkronoimiseen on olemassa useita keinoja, joiden avulla saadaan tehokkaasti tuloksia aikaiseksi [Apostolopoulos 2003]. Helpoin ratkaisu on käyttää uniikkeja uudelleensynkronointimerkkejä. Dekooderin on helppo synkronoida prosessointi säännöllisin väliajoin. Toinen vaihtoehto on käyttää RVLC-koodeja (Reversible Variable Length Code), joiden prosessointi poikkeaa tavallisista VLC-koodeista (Variable Length Codes) siten, että ne ovat uniikkeja sekä oikein päin että väärin päin luettuina. Näin ollen, jos virhe sattuu bittivirran synkronoinnissa, niin synkronointikohdassa dekooderin on mahdollista lukea bittivirtaa väärinpäin. Siispä, kun väärinpäin dekodatussa bittivirrassa ilmenee uudelleen virhe, tiedetään, missä kohdassa virhe on tapahtunut. RVLC-koodien käyttäminen vaikuttaa kuitenkin negatiivisesti suorituskykyyn, minkä vuoksi VLC-koodit ovat yleisemmässä käytössä [Apostolopoulos 2003].

Näiden lisäksi yleisesti käytettyjä keinoja ovat mm. tiedon jaottelu (data partitioning) ja sovellustason kehystys (application-level framing) [Apostolopoulos 2003]. Tiedon jaottelu perustuu todennäköisyyksiin, joiden perusteella synkronointipisteen jälkeiset bitit puretaan todennäköisemmin oikein kuin synkronointipisteestä kauempana olevat bitit. Tämän seurauksena tärkeä tieto tulisi sijoittaa heti synkronointipisteen jälkeen, jotta tämä tieto ei korruptoituisi. Sovellustason kehystys on puolestaan keino, jota käytetään mm. MPEG-4-mediatiedostojen yhteydessä. Tiedosto jaetaan sovellustason paketteihin, jotka ovat jäsennettävissä ja purettavissa yksittäisinä entiteetteinä. Näin ollen virheet eivät leviä näiden pakettien ulkopuolelle.

Virheiden peittäminen (Error Concealment) on puolestaan virheidenhallintamekanismi, jolla pyritään peittämään virhetilanteen aiheuttamat puutokset toistettavassa videossa. Tämä tapahtuu luomalla synteettistä sisältöä puuttuvan kuvan tai sen osan tilalle [Li 2013]. Etenkin reaaliaikaisen suoratoiston yhteydessä ei ole kannattavaa odottaa korvaavan sisällön uudelleenlähettämistä lähettäjältä, joten puuttuvan sisällön luonti reaaliajassa on tällaisessa tilanteessa toimivampi ratkaisu.

Tapoja synteettisen sisällön luomiseen on monia [Salama 2000]. Yksinkertaisimmillaan kadonneen kuvan voi korvata aikaisemmalla kuvalla, jolloin sama kuva näytetään useammin kuin kerran. Tavallisen elokuvan kuvanpäivitysnopeuden ollessa 24 kuvaa

sekunnissa, ei yksittäisen kuvan tuplaaminen tuone suurta heikennystä käyttökokemukseen. Tämä ei kuitenkaan luo siisteintä jälkeä, minkä vuoksi on kehitetty useita algoritmeja, joiden avulla voidaan laskea ja arvioida puuttuvan sisällön pikselit, jotta kuvan puuttuvat osat voidaan korjata mahdollisimman tarkasti alkuperäisiksi.

Virheiden peittäminen voi olla joko aktiivista tai passiivista [Salama 2000]. Mikäli virheiden peittäminen tapahtuu aktiivisesti, tehdään se siirron yhteydessä, hyödyntäen ylimääräistä informaatiota, joka välitetään siirrettävän sisällön yhteydessä. Aktiivisella virheiden peittämisellä on mahdollista korjata puuttuvat osat täydellisesti, mikäli kadonneet osat eivät ole järin mittavia. Toisaalta verkon kuormituksen välttämiseksi merkittävän sisällön siirtonopeutta tulee pienentää, jotta vältytään verkon ylimääräiseltä kuormittamiselta. Passiivinen virheiden peittäminen tapahtuu puolestaan vasta vastaanottajan puolella jälkiprosessointina. Tämä perustuu makrolohkojen (macro block) ja liikevektoreiden (motion vector) analyysiin.

Makrolohkot ovat tavallisesti 16x16-alueen kokoisia pikselilohkoja. Näiden avulla ja vertaamalla niitä korruptoituneen kuvan viereisiin kuviin, voidaan päätellä, kuinka pikselit ovat todennäköisesti olleet värittyneitä puuttuvissa kohdissa. Passiivisella virheiden peittämisellä ei saada rekonstruoitua täysin alkuperäistä vastaavaa kuvaa, mutta algoritmista riippuen on mahdollista päästä hyvinkin lähelle alkuperäistä kuvaa [Salama 2000]. Nämä perustuvat pitkälti erilaisiin tapoihin löytää mahdollisimman tarkka liikevektori, joiden avulla tulkita makrolohkojen pikseleiden mukautumista puuttuneessa kuvassa.

Suoratoistopalvelusta ja tilanteesta riippuen on mahdollista harjoittaa virheiden hallintaa myös uudelleenlähetyksen muodossa. Uudelleenlähetystä ei yleisesti ottaen lueta virheidenhallintamekanismien joukkoon, mutta tietyissä olosuhteissa tämä on kuitenkin käytökelpoinen toimenpide. Käytettävyys riippuu paljon suoratoistopalvelun hyödyntämästä puskurista ja hyväksyttävästä viiveestä viiverajoitetussa uudelleenlähetyksessä. Mikäli yhdensuuntainen viive lähettäjän ja vastaanottajan välillä on tarpeeksi pieni, voi uudelleenlähetyksellä toimia korjauskeinona korruptoituneen tai kadonneen sisällön paikkaamiseksi [Li 2013]. Reaaliaikaisessa suoratoistossa uudelleenlähetyksellä ei ole vaihtoehtona, sillä uudelleenlähetyksellä videotiedosto ei todennäköisesti tule kerkeämään toistopaikalleen ennen toiston etenemistä seuraavaan segmenttiin. VOD-pohjaisessa suoratoistossa viiverajoitettu uudelleenlähetyksellä voi kuitenkin soveltua virheidenhallintamekanismiksi.

5 Reaaliaikainen suoratoisto

Reaaliaikaisten lähetyksen suoratoisto on vakiinnuttanut asemaansa katsottuna media-muotona jatkuvalla kehityksellä viime vuosien aikana. Lähetysten tuottaminen on alkanut hivuttautua yrityksiltä ja järjestöiltä myös tavallisten verkon käyttäjien pariin. Tätä on edesauttanut mm. striimaussovellusten kehitys, verkon kapasiteetin kasvu, audio- ja videolaitteiston halpeneminen sekä ihmisten kiinnostuksen kasvu mm. videopelejä ja suoratoistettavia tapahtumia kohtaan. Lähes jokaisen on mahdollista tuottaa reaaliaikaista videomateriaalia verkkoon esimerkiksi mobiililaitteen avulla.

Tämän seurauksena suurimmat reaaliaikaista videomateriaalia jakavat palvelut, kuten Twitch ja YouTube Gaming sekä Facebook Live ja muut sosiaalisen median tarjoamat suoratoistopalvelut, ovat olleet kovassa kasvussa viime vuosien aikana. Tähän vaikuttaa varmasti myös palveluiden eroava käyttötarkoitus ja käyttökokemus suhteessa tavalliseen VOD-sisältöiseen videomateriaaliin. Katsoja voi kokea tilanteen yhdessä muiden ihmisten kanssa. Oli kyseessä sitten yksittäisen henkilön tuottama striimi tai suuren urheilukanavan jakama urheilutapahtuma, on aikakriittiset tapahtumat parempi myös kokea mahdollisimman pian niiden tapahduttua.

VOD-tyyppisen ja reaaliaikaisen videomateriaalin suoratoistot eroavat palvelumalliensa puolesta toisistaan merkittävästi. Nämä palvelumalleissa esiintyvät erot johtavat myös eroihin käytettävien tekniikoiden ja mekanismien käytössä. Tästä syystä on syytä tarkastella tarkemmin, miten reaaliaikainen suoratoisto poikkeaa tyypillisen VOD-sisällön suoratoistosta.

5.1 *Palvelumallin ero*

Reaaliaikaisen suoratoiston sekä tavalliseen VOD-sisältöiseen videomateriaalin suoratoiston palvelumallien erot ovat johtaneet erilaisiin ratkaisuihin myös käytettävien mekaniikojen ja tekniikojen osalta. Reaaliaikaisen suoratoiston aikakriittisyydestä johtuen on selkeitä valintoja, joiden avulla voidaan taata mahdollisimman pieni viive tuotannon ja toiston välillä. Reaaliaikaisen suoratoiston suurin haaste onkin mahdollisimman pienen viiveen saavuttamisessa [Spiteri 2018].

Reaaliaikaisesti tuotettua videomateriaalia ei ole koodattu ennalta (pre-encode). Tästä johtuen koodauksen tulee tapahtua joko sovelluksen tai laitteiston toimesta lähettäjän laitteella. Tällaista reaaliaikaisesti jaettavaa sisältöä ovat esimerkiksi interaktiiviset

ryhmäpelit, videokonferenssit ja eri palveluiden tarjoamat suorat lähetykset [Bettahar 2005]. Tavallinen VOD-sisältöinen videomateriaali on koodattu levitystä varten jo ennakkoon, joten tähän ei kulu prosessointi aikaa eikä muitakaan resursseja lähetyksenvaiheessa.

Reaaliaikaisissa suoratoistoissa hyödynnetään usein myös lohkotettua lähetystä, joka on kuvattu tarkemmin tämän tutkielman kappaleessa 4.3. Lohkotetun lähetysten avulla on mahdollista pienentää viivettä lähettäjän ja vastaanottajan välillä. Lähetettävät segmentit jaetaan pienempiin lohkoihin, joiden toistaminen erillisinä video-osuuksina on mahdollista. Näin ollen pienempien segmenttien koodaus tuotantovaiheessa nopeutuu ja vältetään kokonaisen segmentin koodauksen tuottamalta ylimääräiseltä viiveeltä. Lohkotettu lähetys mahdollistaa myös yhteyden ylläpitämisen vaihtuvan mittaisen suoratoiston ajan, sillä yhteys katkaistaan vasta, kun lopetusta merkkäava lohko on vastaanotettu. Lohkotetun lähetysten hyödyt tulevat esille etenkin aikakriittisessä ja dynaamisesti tuotetun sisällön jaon yhteydessä. Tavallisen VOD-sisällön suoratoistossa on mahdollista määrittellä yhteyden ylläpito siirrettävän videon kapasiteetin mukaan. Lisäksi suoratoiston alussa käytetyn puskurointiajan pidentämisellä on mahdollista varata puskurivara, joka kestää pidempien segmenttien lähetysten ajan lähettäjältä vastaanottajalle.

Eri suoratoistomenetelmät ja standardit tarjoavat myös erilaisia ratkaisuja ja keinoja pienentää suoratoiston viivettä. Esimerkiksi DASH-suoratoistossa lähetettävät videosegmentit voivat olla pituuksiltaan toisistaan poikkeavia. Lyhyemmät videosegmentit ovat nopeampia koodata ja niiden prosessointi vie vähemmän aikaa. Tästä syystä esimerkiksi DASH:n kanssa on syytä käyttää lyhyempiä videosegmenttejä, kun on kyse aikakriittisestä suoratoistosta [DASH 2014]. Pidemmät videosegmentit toimivat puolestaan paremmin VOD-sisällön välittämisessä ja toistamisessa.

Reaaliaikaisen suoratoiston luonteen vuoksi on syytä valita dynaamisesti mukautuva suoratoisto menetelmä, jossa ei ole suurta puskurivaraa. Suurelle puskurivaralle ei pitäisi olla käyttöä, sillä suuri puskurivara indikoi pidempää viivettä tuottamisen ja toistamisen välillä. Tämä onkin yksi suurimmista haasteista aikakriittisen suoratoiston kanssa [Spiteri 2018]. Pieni puskurijättää erittäin vähän varaa virheille. Muutaman videosegmentin mittainen puskurijättää ei mahdollista virheidenhallinnassa uudelleenlähetystä. Huonosti optimoitu mukautuva suoratoistomenetelmä todennäköisesti aiheuttaisi puskurin tyhjenty-
misen ja tuottaisi puskurointiviivettä, jolla on suuri vaikutus käyttökokemukseen.

Pienen puskurikoon lisäksi segmenttien ennakkoon hakeminen on käytännössä

mahdotonta reaaliaikaisen suoratoiston yhteydessä, sillä sisältö ei todennäköisesti ole vielä levityksessä. Levitykseen kertynyt videomateriaali on kirjattuna tiedostoon, joka ohjaa käyttäjän laitteen hakemaan pyydettyä videosegmenttiä oikealta palvelimelta. DASH hyödyntää reaaliaikaisen suoratoiston yhteydessä dynaamista MPD-tiedostoa, joka muuttuu sisällön tuotannon yhteydessä [DASH 2014]. Mikäli pääsy vanhentuneeseen materiaaliin halutaan evätä, voidaan se poistaa halutun ajan kuluessa MPD-tiedostosta.

Kuten huomaamme, näistä päätöksistä kaikki tukee toistaan. Dynaaminen MPD-tiedosto mahdollistaa segmenttien levittämisen sitä mukaa, kun videomateriaalia tuotetaan. Tuotettu videomateriaali on jaettu lyhyisiin segmentteihin, jotta niiden prosessointi ja siirtäminen on nopeampaa. Segmentit pilkotaan vielä lohkoihin, jotta segmenttien välitys käyttäjälle olisi mahdollisimman nopeaa. Käyttäjän puskuri on tämän lisäksi pieni, sillä suurempi puskuri mahdollistaisi videon varastoitumisen puskuriin, jolloin viive tuotannon ja toiston välillä voi kasvaa huomaamattomasti.

5.2 Tekniset käytänteet ja käytännön toteutukset

Käyttäjien tuottamaa sisältöä tarjoavat palvelut, kuten Twitch, YouTube Live ja Mixer, ovat kasvattaneet suosiotaan etenkin e-urheilun myötä. Kyseisistä palveluista onkin tullut varteenotettava kilpailija tavanomaisille kaapelitelevisiolähetysille [Pires 2015]. Myös sosiaalisen median alustat, kuten Facebook, ovat alkaneet tarjota mahdollisuutta suorien lähetyksien tuottamiseksi ja jakamiseksi.

Eri palveluiden käyttötarkoitukset saattavat vaihdella toisistaan, mutta monilta palveluilta löytyy kuitenkin myös paljon yhteistä. Tämän vuoksi tutkielman kannalta ei ole oleellista tutustua jokaiseen suoratoistopalveluun, joka tarjoaa mahdollisuutta oman sisällön levittämiseen reaaliajassa. Tässä kappaleessa syvennytäänkin tarkemmin Amazonin omistamaan Twitch-nimiseen palveluun ja pyritään havainnollistamaan tämän esimerkin avulla, millaisia teknisiä ratkaisuja palveluun on upotettu, millaisia teknisiä vaikutusmahdollisuuksia palvelu tarjoaa sekä lähetyksen lähettäjän että katsojan näkökulmasta, ja mitkä ovat tämän kokoluokan palvelun suurimmat ongelmat tekniseltä näkökannalta.

Twitch on Amazonin omistama suoratoistoalusta, jota hyödyntäen käyttäjät voivat jakaa tuottamaansa sisältöä palveluun. Pääasiassa pelisisällön jakamiselle rakennettu palvelu

mahdollistaa katsojien osallistumisen pelikokemuksen jakamiseen ja kommunikointiin muiden ihmisten kanssa. Kaikki tämä tapahtuu kanavalla, jonka jokainen voi itselleen perustaa.

Jokaisella Twitch-käyttäjällä on oma kanava, joka luodaan palveluun rekisteröitymisen yhteydessä. Käyttäjän on mahdollista jakaa suoraa lähetystä omasta toiminnastaan yksityisen ja kanavakohtaisen avaimen avulla omalle kanavalleen. Tätä avainta hyödyntäen kuka tahansa voi lähettää jaettavaa sisältöä kyseiselle kanavalle omalta striimaussoveltukseltaan (streaming software).

Twitch-palvelun käyttäjät voivat katsoa kaikkia kanavia ja tukea kanavien toimintaa joko seuraamalla kanavaa tai tilaamalla kanavan. Kanavan tilaaminen on maksullinen toimenpide, joka tarjoaa käyttäjälle erityisiä oikeuksia kanavan parissa. Tilaajille suodut erityisoikeudet vaihtelevat kanavakohtaisesti, mutta yleisimpinä etuuksina ovat kanavalle luotujen hymiöiden avaaminen käyttöä varten. Mitä enemmän katsojia, seuraajia ja tilaajia kanavalla on, sitä enemmän Twitch on myös valmis käyttämään resursseja kanavan lähetksen tukemiseksi [Twitch 2020].

Twitch-kanavat voidaan jakaa kolmeen pääkategoriaan “pienet kanavat”, “jäsen-kanavat” (affiliates) ja “kumppani-kanavat” (partners). Pienimmät kanavat saavat vähiten etuuksia, sillä niillä on vähiten katselukertoja, samanaikaisia katselijoita ja seuraajia. Jäsen-kanavilla on puolestaan vähintään 50 seuraajaa, useampia yhtäaikaisia katselijoita ja sen puolesta myös pieniä etuuksia, kuten kanavan tilausmahdollisuus [Twitch 2020 (1)]. Suurimmilla kanavilla, eli kumppaneilla on puolestaan suurimmat etuudet, sillä näillä kanavilla on eniten seuraajia, huomattava määrä yhtäaikaisia katsojia ja suuri määrä katselukertoja. Tämän vuoksi myös Twitch on valmis käyttämään resurssejaan näiden kumppanien tukemiseksi.

Pienimmillä Twitch-kanavilla on mahdollisuus jakaa ainoastaan “raakaa” (raw) videomateriaalia, jota ei ole koodattu eri laatuiseksi. Mikäli kanavan omistaja päättää esimerkiksi jakaa 1080p-resoluutiolla tuotettua sisältöä kanavallaan, tulee katsojien myös katsoa jaettava sisältöä kyseisellä laadulla. Twitch ei siis käytä resurssejaan videon koodaamiseksi eri laatuisiin pienimpien kanavien kohdalla. Mukautuvan suoratoiston implementointi Twitchin suuruiseen palveluun jokaiselle käyttäjälle koituisi varsin kalliiksi palvelun tarjoajalle [Pires 2014]. Käytännössä jokaisen samanaikaisen käyttäjän tuottama videomateriaali tulisi koodata eri laatuisiin. Tämä kasvattaisi palvelun kuluja merkittävästi ja

kuluttaisi palvelun laskentakapasiteettia huomattavasti. Toisaalta tällaisen mahdollistaminen toisi myös paljon hyötyjä palvelun käyttökokemuksen puolesta. Korkealaatuisen videon katsominen voi aiheuttaa ongelmia käyttäjille, jotka katsovat lähetystä vanhoilla mobiililaitteilla tai heikolla verkkoyhteydellä. Mukautuvan suoratoiston mahdollistaminen videomateriaalin kulutukseen toisi siis myös huomattavaa pienennystä kaistan kuormitukseen, joka mahdollistaisi sisällön toistamisen heikommillakin verkkoyhteyksillä.

Twitch määrittelee lähetyksen korkeimmaksi mahdolliseksi bittinopeudeksi (bitrate) 6000kbps [Twitch 2020 (2)]. Twitchin keräämien tietojen mukaan useat lähetykset käyttävät kuitenkin huomattavasti pienempää bittinopeutta lähetyksen jakamiseksi. Korkeampi bittinopeus mahdollistaa parempilaatuisen sisällön jakamisen, mutta aina se ei kuitenkaan ole välttämätöntä. Suuri bittinopeus voi aiheuttaa myös hankaluuksia videon suoritusvaiheessa, sillä kaikki laitteet eivät välttämättä tue korkeaa bittinopeutta. Lisäksi yhteyden mahdolliset heikentymiset ovat havaittavissa helpommin korkean bittivirran lähetyksissä, sillä pienetkin muutokset voivat estää korkean bittinopeuden lähetyksen sujuvan jakamisen. Videosisällössä, jossa ei tapahdu nopeita muutoksia, pienempikin bittinopeus riittää laadukkaan kuvan jakamiseksi. Esimerkiksi 1500kbps bittinopeudella on mahdollista jakaa laadukasta videota HD-laadulla (high definition) sisällöstä, jossa ei ole nopeita muutoksia [Twitch 2020 (2)].

Käyttäjien tekemien bittinopeuksien pienennyksellä on suuria vaikutuksia isojen palveluntarjoajien tiedonsiirtomäärissä. Twitch kattaa n. 4,2% kaikesta videoiden suoratoistosta maailmanlaajuisesti. Tämä määrä on ollut suuressa kasvussa e-urheilun myötä [Sandvine 2019]. Jaettavan sisällön bittinopeuden pienentäminen neljännekseen tuo suuria muutoksia myös verkon kuormitukseen. Mikäli tätä samaa muutosta sovelletaan muihin reaaliaikaista suoratoistoa tarjoaviin palveluihin, kuten YouTube Liveen, voidaan verkon kuormitustilaa pienentää merkittävästi, vaikuttamatta suuresti suoratoistettavan sisällön laatuun.

Twitch on varsin tavanomainen suoratoistopalvelu (Over-The-Top (OTT) service), jossa median välitys käyttäjälle tapahtuu yhdeltä yhdelle -yhteydellä (unicast) [Pires 2014]. Twitchin tarjoamien käyttäjämäärien ja siirtonopeuksien perusteella on laskettu, että mediasisältöä välitetään palvelulta korkeimmillaan noin 2Tbps siirtonopeudella. Pienin siirtonopeus Twitchin jokapäiväisessä tiedonsiirrossa on noin 400Mbps.

Twitchille lähetetty "raakavideo" koodataan käyttäjän striimauslaitteella ja lähetetään

Twitchin palvelinkeskukselle (data center) levitystä varten [Pires 2014]. Mikäli lähetystä tekevä kanava on kumppani-kanava, koodataan lähetetty video eri laatuiseksi, jotta käyttäjän on mahdollista valita haluamansa ja tämän yhteyksille tai laitteelleen soveltuva toistolaatu. Myös mukautuva suoratoistoalgoritmi alkaa tässä tapauksessa toimimaan ja välittämään oletuksena parasta mahdollista laatua käyttäjäkohtaisesti. Mikäli kanava ei kuitenkaan ole tarpeeksi iso ja Twitchillä ei ole resursseja jaettavaksi pienemmille kanaville, välitetään raakavideo sellaisenaan Twitchin palvelimilta käyttäjille.

[Pires 2014]:n tekemien päätelmien mukaan vain murto-osalla lähetystä tekevästä kanavista tulisi olla mukautuvan bittinopeuden suoratoisto käytössä lähetyksissään. Tästä on hyötyä ainoastaan suurille kanaville, joiden lähetykset ovat korkeaa laatua, ja joiden välitys Twitchille tapahtuu korkealla bittinopeudella. Mukautuvasta bittinopeudesta on tällöin kaksi mainittua hyötyä.

Ensinnäkin mukautuva bittivirta mahdollistaa sisällön toiston kaikille käyttäjille. Käyttäjät, joiden yhteys ei tue suurta bittivirtaa, on mahdollista välittää sisältöä pienemmällä bittivirralla, jolloin välityksen puskurointiviiveiltä ja katkoksilta lähetyksessä [Pires 2014]. Katsojan lataama sisältö välitetään pienemmällä bittinopeudella kuin lähetetty raakavideo, jolloin katsoja välttyy ylimääräiseltä kuormitukselta. Tämä parantaa palvelun käyttökokemusta ja antaa kanavalle potentiaalisia, uusia, katsojia. Toiseksi, mukautuva bittinopeus suoratoistossa pienentää yleistä kuormitusta palvelun kaistalla ja mahdollistaa pienemmän kaistanleveyden palveluntarjoajalle. Ilman mukautuvaa bittinopeutta suoratoistossa Twitchin tulisi välittää vain raakavideota katsojille, jolloin sekä palveluntarjoaja että katsoja voisivat kokea palvelun laadun heikkenemistä.

Reaaliaikaisen suoratoiston yhteydessä palvelun eräs ongelma on päätös kanavista, joiden lähetyksiä voidaan resurssien mukaan koodata eri laatuiseksi. Mikäli palvelun taso heikenee käyttäjillä esimerkiksi puskurointiviiveiden johdosta, voi palvelu menettää potentiaalisia katsojia [Pires 2014]. Näin ollen Twitchin kaltaisten palveluiden tulee tunnistaa tilanne, jolloin raakavideon koodauksesta toiseen laatuun on enemmän hyötyä kuin haittaa. Potentiaalisten katsojien määrän täytyy olla tarpeeksi suuri, jotta sen tuomat hyödyt ylittävät resurssien käytöstä aiheutuvat kulut. Kanavien katsojamäärät elävät lähetyksien aikana, joten resurssien käyttö tietyn kanavan kohdalla voi olla ongelmallista. Muutokset käytettävään prosessiin vaikuttaa jokaisen katsojan manifestitiedostoon, sillä palvelun käyttäjien tulee olla tietoisia käytettävissä olevista laatuversioista, jotta optimaalinen hyöty mukautuvasta bittinopeudesta on saavutettavissa.

Mukautuvan bittinopeuden suoratoistossa on kaksi lähestymistapaa prosessin muutokseen [Pires 2014]. Suoratoistopalvelu voi päättää lähetyksen sisällön koodaamisesta toiseen laatuun joko lähetyksen käynnistysvaiheessa tai lähetyksen edetessä. Kummassakin mallissa on omat hyötynsä ja haittansa.

Käynnistysvaiheessa tehty päätös koodauksesta vaatii ennakkointia palvelun puolesta. Katsojamäärän on syytä ylittää tietylle tasolle, jotta prosessin muutoksen hyötysuhde on palvelun kannalta positiivinen. Tästä johtuen tämä malli on ongelmallinen, sillä ennakkointi on aina varsin epävarmaa. Tämän vuoksi esimerkiksi Twitch tarjoaa mukautuvan bittivirran suoratoistoa kumppani-kanaville, joiden asema palvelussa on jo varsin vakiintunut. Lähetyksen edetessä tehty päätös on puolestaan yleisesti toimiva ratkaisu, sillä sen avulla pystytään reagoimaan yllättäviinkin katsojapiikkeihin eikä resursseja mene käytännössä hukkaan, mikäli muutoksen hyötysuhde ei tue prosessin muutosta. Tämä malli vaatii kuitenkin aktiivista seuraamista katsojamääristä, joten oikeanlaisen mallin rakentaminen palvelulle voi olla hankalaa.

6 Suoratoisto lähitulevaisuudessa

Suoratoisto elää teknologian suurimpien muutoksien mukana. Esimerkiksi uudet langattoman teknologian kehitykset, kuten 5G, tulevat tarjoamaan käyttökokemuksen parannusta mobiileille käyttäjille, joiden osuus tämän päivän suoratoistosta on merkittävän suuri. Käyttäjien siirtyessä mobiileille alustoille, tulee myös verkon mukautua tarjoamaan parannuskeinoja palveluiden käyttökokemuksille. Eräänä ratkaisuna on esitetty esimerkiksi reunalaskennan hyödyntämistä suoratoistopalveluiden yhteydessä.

Tämän kappaleen tarkoituksena on havainnollistaa suoratoistopalveluiden tulevaisuuden näkymiä lähivuosina uusien teknologioiden astuessa esiin. Kappaleessa tarkastellaan myös suurimpien palveluntarjoajien mukautumista räjähdysmäiseen käyttäjä- ja videomäärän kasvuun. Lisäksi kappaleessa esitetään suoratoiston vaikutuksia yhteiskuntaan ja yhteiskunnan tekemiä asetuksia suoratoiston toiminnan kehitykselle.

6.1 Lähitulevaisuuden näkymät

Videoiden suoratoiston lähitulevaisuus näyttää varsin lupaavalta tällä hetkellä. Suoratoistopalveluiden käyttäjämäärät ovat jatkuvassa nousussa ja videomateriaalia tuotetaan katseltavaksi jatkuvalla syötteellä. Kysyntää ja tarjontaa on paljon, mutta mitä voimme odottaa videoiden suoratoistolta tulevaisuudessa? Millaiset teknologiamuutokset tulevat vaikuttamaan suoratoiston kehitykseen ja kuinka suurimmat palveluntarjoajat valmistautuvat videomateriaalin jatkuvaan kasvuun?

Tällä hetkellä yksi kuumimmista aiheista mobiilissa verkkoteknologiassa on 5G. 5G-datayhteys on seuraavan sukupolven verkkoteknologiaratkaisu, joka mahdollistaa huomattavasti leveämmän verkon kaistan ja siirtonopeuden. Tämä perustuu taajuuteen, jolla 5G-verkko operoi [IEEE 2020].

5G-verkon myötä langaton teknologia siirtyy hyödyntämään millimetriaaltoja. Tämä tarkoittaa sitä, että 5G-yhteydessä käytettävän taajuuden (30-300Ghz) aallonpituus, jolla tietoa siirretään, on vain n.1-10mm. Tällä hetkellä älylaitteita palvelevat aallot ovat pituudeltaan jopa kymmeniä senttimetrejä. Aallonpituuden pienentämisen eli palvelutaajuuden kasvatuksen seurauksena on mahdollista palvella verkon mobiileja käyttäjiä uskomattomalla nopeudella aikaisempaan verrattuna. Tämän hetkinen 4G-verkko palvelee käyttäjiä 1-100Mbps siirtonopeudella, kun taas 5G:n myötä tavoitteena on saavuttaa jopa 100Mbps-1Gbps vaihteluväli [Andrews 2014]. Taajuuden nostaminen on yksinkertaiselta kuulostava muutos, mutta käytännön kannalta se sisältää paljon haasteita.

5G-verkko mahdollistaa suuren siirtonopeuden käyttäjille, mutta tämän siirtonopeuden saavuttaminen on itsessään haaste [Andrews 2014]. Jotta 5G-verkon tavoitenopeudet saavutettaisiin, tulee muun teknologian ja infrastruktuurin tukea moninkertaisia nopeuksia. Esimerkiksi tukiasemien palvelukattavuutta tulee pienentää merkittävästi ja tukiasemien tiheyttä lisätä, jotta signaalin voimakkuus ylittää tavoitenopeuksien saavuttamiseksi. Tämän lisäksi 5G-verkon suunnitellut käyttötarkoitukset mm. virtuaalitodellisuuden ja robotiikan saralla vaativat nykyistä pienemmän viiveen. Tämän viiveen saavuttaminen vaatii myös muiden teknologioiden kehitystä mm. verkon kerroksien osalta. Verkon viive on riippuvainen muustakin kuin siirtonopeudesta. Myös 5G-verkon kustannukset tulevat olemaan moninkertaiset 4G-verkkoon nähden.

5G-verkon hyödyntämät millimetriaallot mahdollistavat leveämmän kaistan ja suuremman siirtonopeuden, mutta näillä millimetriaalloilla on myös yksi suuri heikkous

nykyisiin 4G-teknologian hyödyntämiin aaltoihin nähden. Niiden kantavuus on huomattavasti lyhyempi [Andrews 2014]. Suurella aallon pituudella on se hyöty, että se kulkee helpommin rakennusten ja muiden esteiden läpi, kun taas pienet millimetriaallot eivät näitä pysty ohittamaan. Tämän seurauksena tukiasemien määrää joudutaan nostamaan huomattavasti ja niiden tiheyttä kasvattamaan merkittävästi. Esimerkiksi aikaisempia langattomia teknologioita tukeneet tukiasemat ovat pystyneet palvelemaan jopa neliökilometrien kokoisia alueita. 5G-verkon tukiasemien palvelualue putoaa vain satojen metrien säteelle [Andrews 2014]. Tämän seurauksena tukiasemat joudutaan myös linkittämään välityskanavien avulla ydinverkkoon, joiden ylläpito tulee lisäämään huomattavasti kuluja 5G-verkolle.

Suoratoiston kannalta 5G-verkko pystyy tuomaan huomattavia laatuparannuksia. Suuremman siirtonopeuden myötä ja leveämmän kaistan avulla käyttäjien palvelukapasiteetti kasvaa merkittävästi ja käyttökokemuksen voidaan olettaa nousevan suoratoistopalveluissa. Tämän 5G-teknologiaan siirtymisen seurauksena myös videoiden laatu voidaan nostaa UHD-tasolle (Ultra High Definition) [Nightingale 2018]. Kun tämä yhdistetään H.265 pakkausstandardiin, joka on HD-sisällössä paljon käytettyä H.264 pakkausstandardia tehokkaampi, voidaan esimerkiksi 4K-laatuista videomateriaalia välittää pienemmässä muodossa kuin aiemmin. H.265 pakkausstandardi mahdollistaa siis videosisällön paketoimisen entistäkin pienempään tilaan aiheuttamatta vahinkoa toistettavalle videomateriaalille [Nightingale 2018]. Tämä pakkausstandardi lievittää verkon kuormitusta UHD-materiaalin siirrossa, joka tulee varmasti yleistymään 5G-verkon leviämisen myötä.

Suoratoistopalveluiden jatkuvan kasvun ja tulevan siirtymisen UHD-laatuiseen videon pääsääntöiseen levitykseen, tulevat aiheuttamaan haasteita suurimmille palveluntarjoajille, kuten Googlelle ja Amazonille. Kuinka näiden toimijoiden tulisi hallita jatkuvasti heidän palveluihinsa työnnettävää videosisältöä?

Pakkausstandardien kehittyminen tuo mukanaan suuren lievityksen myös suoratoistopalveluntarjoajilta vaadittuun tallennuskapasiteettiin. Volyymi, jolla uusia videoita julkaistaan esimerkiksi YouTubessa, on kuitenkin niin suuri, että tämä ei yksinään auta suuren videomäärän hallinnoinnissa. Tästä syystä alan suurimmat toimijat joutuvat jatkuvasti lisäämään palvelinkeskuksia ympäri maailmaa, joilla palvella kaikkia käyttäjiään, ja joihin säilöä julkaistavaa videomateriaalia.

Tällä hetkellä Googlella ja Amazonilla on omistuksessaan yhteensä 42 palvelinkeskusta ympäri maailmaa. Tämä määrä tulee kuitenkin yhtiöiden ilmoittamien suunnitelmien mukaisesti kasvamaan lähivuosien aikana. Kuvassa 6.1 havainnollistetut palvelinkeskuksat on levitetty myös suurimpien käyttäjämassojen läheisyyteen, jotta palvelimet sijaitsisivat mahdollisimman lähellä käyttäjiä. Tällaisella levittäytymisellä ja maantieteellisesti taktisella palvelinkeskuksien sijoitetuilla pystytään tarjoamaan mahdollisimman hyvää palvelua käyttäjille.

Tulevaisuuden teknologiat, kuten uudet pakkausstandardit ja langattomat verkkoteknologiat, tulevat mahdollistamaan korkeamman palvelutason tarjoamisen palveluiden käyttäjille. Tämän lisäksi videoliikenteen lisääntyessä ja käyttäjien tukeutuessa enenevin määrin mobiililaitteisiin, tulee tulevaisuuden ratkaisujen palvelulla suuresti toisistaan poikkeavia laitteita. IoT-laitteet tulevat tuomaan oman haasteensa m2m-kommunikaation (machine-to-machine) muodossa ja kasvattamaan palveltavien laitteiden heterogeenisyyttä entisestään [Porambage 2018]. Palvelutason noustessa, myös vaatimukset kasvavat, joten 5G-verkolle suunnitellut käyttötarkoitukset tulevat puskemaan verkkoteknologiaa nopeampia ja tehokkaampia ratkaisuja kohti. Virtuaalitodellisuus ja internetin välityksellä toimivat kosketeltavat käyttöjärjestelmät (tactile internet) vaativat lähes olemattoman pientä viivettä [Andrews 2014]. Suoratoiston puolella 5G-verkkojen hyödyntäminen tulee näkymään etenkin korkeamman videolaadun muodossa, sekä sulavammassa käyttäjäkokemuksessa mm. harventuneiden puskurointiviiveiden puolesta.

6.2 Reunalaskenta-avusteinen videoiden suoratoisto

Mobiilipalveluiden käyttökokemus on vahvasti yhteydessä verkon olosuhteisiin. Käyttäjämäärän kasvaessa myös käytettävissä oleva kaista jaetaan pullonkaulalinkkien osalta. Tämä voi aiheuttaa mm. pitkiä puskurointiviiveitä ja laadun pienenemistä, jolloin myös palvelun käyttökokemus jää varsin heikolle tasolle [Mehrabi 2019].

Pullonkaulalinkkien aiheuttaman ruuhkautumisen takia verkon resursseja ei myöskään pystytä hyödyntämään optimaalisesti ja reilusti jokaisen käyttäjän kesken [Mehrabi 2019]. Tämän takia viime aikoina onkin pyritty löytämään keino, jonka avulla voidaan jakaa yksittäisten linkkien kuormitusta verkossa. Ratkaisun tulisi edistää videoiden suoratoiston käyttökokemusta vähentämällä puskurointiviiveitä, pienentämällä niiden kestoa

sekä jakamalla verkon resursseja tasaisesti jokaiselle käyttäjälle. Tällaista ratkaisua tarjoaa mm. reunalaskenta-avusteinen (Edge-Computing), mobiili, videoiden suoratoisto [Mehrabi 2019].

Reunalaskennan tarkoituksena on tuoda pilven laskentateho ja säilytystila käyttäjän läheisyyteen, nopeampaa prosessointia varten [Mehrabi 2019]. Verkon materiaalin säilytys verkon reunalla lievittää myös laajalti verkon kuormitusta. Koska säilötty videosisältö on kopioitu verkon reunalle käyttäjän läheisyyteen, on sen toistaminen ystävällisempää verkolle. Saman videomateriaalin ei tarvitse kulkea useaan kertaan koko verkon läpi, kuluttaen väliin jäävien linkkien palvelukapasiteettia. Käyttäjälle koituva hyöty sisällön läheisestä säilytyksestä ei kuitenkaan tule sen nopeasta saatavuudesta, vaan parantuneesta palvelun laadusta, jonka tämä mahdollistaa.

Moniliittynäisen reunalaskennan (Multi-access edge computing, MEC) tarkoituksena on laajentaa pilven laskentakykyä radioliityntäverkon reunalle, mikä mahdollistaa yhteyden radioverkon resursseihin äärimmäisen pienellä viiveellä [Porambage 2018]. Tämän lisäksi MEC on mahdollistanut myös verkon sisällön säilömisen radioliityntäverkkoon (Radio Access Network) [Mehrabi 2019]. Radioliityntäverkot ovat aivan verkon reunalla, loppukäyttäjän ja ydinverkon välissä olevia verkkoja. Näiden verkkojen liittäminen reunalaskentaan on mahdollistanut vastaamisen kasvaneeseen kysyntään reaaliaikaisten palveluiden osalta [Mehrabi 2019]. Nämä reaaliaikaiset palvelut ovat erityisen hyödyllisiä IoT-laitteiden toiminnan ja kontrolloinnin kannalta, mutta ne mahdollistavat myös koordinoitua ohjaamista videoiden mobiilin suoratoiston yhteydessä.

Mobiiliin videoiden suoratoistoon on kehitetty ratkaisu, jonka avulla voidaan saavuttaa monelta osin parempia tuloksia kuin tavallisen DASH-protokollan avulla [Mehrabi 2019]. Parannusta on havaittavissa mm. verkon hyödyntämisen ja reilun kapasiteetin jaon muodossa. Kehitetty ratkaisu tuo jaettavan videosisällön käyttäjien läheisyyteen verkon reunalle, jolloin verkon kuormitus jää alhaisemmalle tasolle. Tämän seurauksena myös palveluiden käyttökokemus paranee.

[Mehrabi 2019]:ssä esitetyn ratkaisun pohjana on MEC-ympäristön mahdollistamien radioliityntäverkon reaaliaikaisten palveluiden avulla toimivat koordinaattorit. Näiden koordinaattorien tehtävänä on huolehtia säännöllisin väliajoin käyttäjien ja reunal palvelimien (edge-server) kartoituksesta sekä huolehtia jokaisen käyttäjän bittinopeudesta. Näitä toimintoja ylläpidetään MEC-ympäristön mahdollistamien radioliityntäverkon

kommunikaatiotoiminnoilla, joiden avulla informaation välitys eri toimijoiden välillä on äärimmäisen nopeaa, sekä mobiilikäyttäjien tarjoamien sovelluskerroksen tietojen perusteella.

Koordinaattorien tekemän käyttäjien ja reunapalvelimien kartoituksen tarkoituksena on jakaa reunapalvelimien ja tukiasemien kuormitusta palvelukyvyyn optimoimiseksi [Mehrabi 2019]. Tämän seurauksena jokainen mobiilikäyttäjä ohjataan tälle optimaaliselle palvelimelle. Palvelin valitaan jokaisen käyttäjän kohdalla sen kuormitustason mukaan niin, että tämän optimoinnin seurauksena käyttäjän suoritusteho (throughput) saa maksimaalisen arvon. Jokainen käyttäjä pysyy samalla palvelimella niin pitkään kuin tällä on lohkon (chunk) lataus kesken. Tämän jälkeen käyttäjä voidaan ohjata uudelle palvelimelle, mikäli verkon olosuhteissa on tapahtunut muutoksia, joiden seurauksena palvelimen vaihdosta koituu käyttäjälle hyötyä. Tällä tavoin pystytään parantamaan palvelun käyttökokemusta.

Koordinaattorien tekemällä bittinopeuden optimoimisella pyritään jakamaan palvelimien resursseja jokaiselle käyttäjälle reilulla tavalla [Mehrabi 2019]. Bittinopeuden tulee olla kuitenkin niin korkea kuin mahdollista, jotta käyttäjä voi nauttia valitsemastaan sisällöstä mahdollisemman korkealla laadulla. Toisaalta liian suuri laatu voi johtaa käyttökokemuksen heikkenemiseen esimerkiksi puskurointiviiveiden myötä tai epätasaiseen resurssien jakoon eri käyttäjien välillä.

Tavallisessa DASH-ratkaisussa, käyttäjä voi itse valita parhaan mahdollisen bittinopeuden, joka tämän resursseilla on mahdollista ylläpitää. Tämä voi johtaa epäreiluun tilanteeseen, sillä useamman käyttäjän välinen kilpailu saman aikaikkunan resursseista saattaa heikentää yhden käyttäjän bittinopeutta, muiden viedessä aikaikkunan resurssit omaan käyttöön [Mehrabi 2019]. Esiitetty ratkaisu bittinopeuden reiluksi optimoimiseksi perustuu kuitenkin bittinopeuden valintaan käyttäjän puolesta. Valinta jokaisen käyttäjän kohdalla perustuu käyttäjän linkin laatuun ja aikaikkunan resurssien tasaiseen jakoon. Bittinopeus pyritään asettamaan siten, että bittinopeuksien hajonta pysyy mahdollisimman pienenä. Jokaiselle käyttäjälle valitaan paras mahdollinen bittinopeus, joka ei poikkea liikaa saman aikaikkunan jakavien käyttäjien keskiarvoisesta bittinopeudesta. Päättännän siirtyessä järjestelmälle, ei yksittäinen käyttäjä voi valita ahneesti suurinta laatua, heikentäen näin muiden käyttäjien käyttökokemusta.

Ratkaisu ottaa huomioon neljä kriteeriä, joiden avulla määrittää käyttökokemusta. Videon

bittinopeus (video bitrate), käynnistysviive (startup delay), seisahtumissuhde (stalling ratio) sekä bittinopeuden vaihtelu (bitrate switching) [Mehrabi 2019]. Näiden kriteerien mukaisesti tehtävä käyttökokemuksen optimointi johtaa oikean bittinopeuden valintaan, joka suoritetaan reilusti jokaisen palvelimeen yhdistyneen käyttäjän kesken.

Ratkaisun mallin mukaan tukiasemat ja niihin yhdistyneet reunapalvelimet toimivat yksittäisinä toimijoina, jolloin tietoja ei tarvitse välittää tukiasemien kesken. Näin ollen järjestelmän palvelimilla ei ole laajempaa käsitystä verkosta tai sen tilasta. Oletuksena onkin, että jokainen videolohko on monistettu jokaiselle reunapalvelimelle eri laatusina versiona, jolloin käyttäjien on mahdollista saada haluamansa videolohko läheiseltä palvelimelta. Mikäli lohkoa ei löydy palvelimelta, näiden rajallisen tallennuskapasiteetin vuoksi, joudutaan se hakemaan kauempaa, toiselta palvelimelta [Mehrabi 2019].

Palvelun koosta riippuen tämä toteutus voi tuoda lisähyötyä videoiden suoratoistoon. Kokonaisvaltaiseksi infrastruktuuriksi esimerkiksi YouTube'n suuruiselle palvelulle tällaisesta reunalaskenta-avusteisesta toteutuksesta ei väistämättäkään ole. Yksinkertaisesti jokainen palvelin ei voi pitää sisällään YouTube'n kokoisen palvelun kaikkia videoita. Näin ollen hauissa joudutaan nojaamaan jatkuvasti kauempana oleviin palvelimiin. Kun reunapalvelimien osumat jäävät tarpeeksi alhaisiksi, ajaudutaan käytännössä nykyisen kaltaiseen tilanteeseen, jossa suurin osa videoista haetaan pilvestä, reunan sijaan. Toisaalta, jos reunapalvelimet voisivat pitää sisällään esimerkiksi kopiot videosta, jotka ovat suuressa suosiossa päivän aikana, voi tämä tuoda selkeää apua verkon kuormitukseen.

Suuri osa YouTube'n sisältämistä videoista ei saa lähes ollenkaan päivittäisiä katselukertoja. Tämän vuoksi on mielestäni perusteltua sanoa, että niiden kopioiminen jokaiselle reunapalvelimelle on täysin turhaa. YouTube-videoita voidaan myös asettaa yksityisiksi, jolloin käyttäjä on ainoa, joka voi videota katsoa. Tällaisia videoita on siis turhaa kopioida palvelimille, joita käyttäjä ei tule itse ikinä käyttämään. Tällainen yltiömäinen kopioiminen johtaa vain hukattuihin resursseihin. Lisäksi videosisällön paikallisuus voi rajoittaa sen tarvetta tulla levitettyksi muissa maissa sijaitseville reunapalvelimille. Mikäli jaettu videosisältö kuitenkin on suuressa kulutuksessa ympäri verkkoa, on tällaisen ratkaisun hyödyntäminen palvelun kannalta varsin kannattavaa. Sisällön jakoon ei ilmesty pullonkaulalinkkejä, sillä käyttäjät ovat yhteydessä lähimpänä palvelemaan reunapalvelimeen, keskitetyn palvelimen sijaan. Näin ollen ruuhkautuminen verkossa vähenee, ja keskeiset linkit välttyvät ylimääräiseltä kuormitukselta. Verkon läpi suuntautuvia massiivisia videotiedostojen siirtoja ei tarvitse enää välittää yhdeltä yhdelle -yhteyden avulla.

Paikallinen reunapalvelin on kykenevä palvelemaan useaa käyttäjää paikallisesti [Mehrabi 2019].

6.3 Yhteiskunnan asettamat rajoitukset

Suurimmat palveluiden tarjoajat, kuten Netflix, Google, Amazon ja Facebook käyttävät erittäin suuren osan verkon kaistasta. Netflix kuluttaa 12,6% verkon latausliikenteestä (downstream traffic) ja YouTube puolestaan 8,7% [Sandvine 2019]. Verkon kaistan kulutuksen jakautuessa varsin epätasaisesti lukuisten tarjolla olevien palveluiden kesken, onkin herännyt kysymys siitä, tulisiko näitä suurimpia kaistan kuluttajia veloittaa tai rajoittaa tämän seurauksena. Tällä hetkellä videomateriaalin kattaessa n. 60% kaikesta Internetin liikenteestä ja Ciscon tekemän ennusteen mukaan tämän noustessa jopa 82%:iin vuoteen 2022 mennessä, on tämä varsin ajankohtainen ja tärkeä asia nostaa esille [Sandvine 2019] [Cisco 2019].

Verkon neutraalius (net neutrality) on periaate, jonka mukaan hallituksien ja internetpalveluntarjoajien tulisi kohdella jokaista verkon toimijaa ja palvelua yhdenvertaisesti [Madhvapaty 2014]. Verkon liikennettä ei tulisi siis rajoittaa sen tyypistä, lähettäjistä tai vastaanottajasta riippuen. Kaiken liikenteen tulisi voida kulkea vapaasti verkossa ilman liikenteen hidastamista tai estämistä. Suurimpien palveluiden kuitenkin omiessa ison osa verkon kapasiteetista, on tähän pyritty tekemään muutoksia.

Vuonna 2018 Yhdysvalloissa kumottiin verkon neutraaliutta tukeva päätös, minkä seurauksena internetpalveluntarjoajien oli mahdollista rajoittaa verkossa kulkevaa liikennettä tai laskuttaa verkon toimijoita tämän liikenteen välittämisestä. Ääripään toimenä liikenne pystyttiin jopa estämään kokonaisuudessaan, mikäli se oli tietyn tyyppistä tai tietyn palvelun tarjoamaa. Käytännössä tämä johti tilanteeseen, jossa internetpalveluntarjoajien nähtiin olevan kykeneviä alkaa veloittamaan palvelunlaadusta käyttäjiltään. Mikäli käyttäjä haluaisi käyttää tiettyä palvelua, tulisi tämän maksaa enemmän verkkolas-kussaan [Collins 2018]. Tämän verkon neutraaliuden kumoutumisen myötä nousi myös huoli siitä, että suurimmat palvelut voivat vain ostaa itselleen nopeaa palvelua, kun taas tavalliset käyttäjät joutuvat tyytymään hitaampaan kaistaan. Pahimmillaan tilanne voi johtaa pienten yritysten kaatumiseen, suurien tekijöiden vallatessa verkon kapasiteetin lähes kokonaan rahallisella kompensatiolla.

Vuonna 2019 Yhdysvalloissa on ajaututtu asemaan, jossa osavaltiot voivat itse päättää,

kuinka toteuttaa verkon neutraaliutta. Tämän seurauksena eri osavaltioissa on erilaiset käytänteet verkon neutraaliuden toteuttamiseksi. Tästä on kuitenkin seurannut tilanne, jossa internetpalveluidentarjoajat ovat alkaneet rajoittamaan (throttling) suosituimpien suoratoistopalveluiden liikennettä [Li 2019].

Tehdyn tutkimuksen mukaan esimerkiksi AT&T operaattori Yhdysvalloissa on rajoittanut Netflixin bittinopeutta 1,5Mb/s:iin 70% tutkituista tapauksista ja YouTuben liikennettä puolestaan 74%:ssa tutkituista tapauksista [Li 2019]. Nämä luvut vaihtelevat merkittävästi eri operaattoreiden välillä. Tärkeintä on kuitenkin tiedostaa se, että videomateriaalin levityksen noustua johtavaksi liikenteen tuottajaksi verkossa, ovat operaattorit ruvenneet asettamaan palvelunopeuksien rajoituksia tietyille palveluille eri suhteissa. Esimerkiksi tehdyn tutkimuksen mukaan AT&T ei ole rajoittanut ollenkaan Amazon Primen liikennettä. Verkon neutraaliuden poistamisesta koituvia seurauksia aletaankin nyt näkemään enenevässä määrin. Loppukäyttäjien näkökulmasta tämä on varsin huolestuttavaa, sillä internetpalveluntarjoajalla on nyt mahdollisuus vaikuttaa palveluiden toimimiseen ja eri palveluiden suosimiseen. Näin ollen teleoperaattorit pystyvät vaikuttamaan suuresti siihen, mitä palvelua heidän käyttäjänsä käyttävät.

Verkon neutraalius ei ole ainoa suoratoistopalveluihin liittyvä asia, johon yhteiskunta on ottanut kantaa aktiivisesti viime aikoina. Videoiden levitykseen liittyy myös tekijänoikeudet, joiden suojelemiseksi on EU:ssa tehty lakialoite, jonka artikla 17 (ent. artikla 13) määrittää verkon materiaalin tarkemman tekijänoikeuksien valvomisen ja suojelemisen [EU 2020]. Käytännössä tämä tarkoittaa sitä, että palveluiden tulee taata, että levitetystä materiaalista ei ole tekijänoikeuksia rikkovaa materiaalia. Tekijänoikeudellista mediaa ei saa käyttää, näyttää tai soittaa jaettavissa videoissa. Tämän seurauksena, myös videopeljä ja e-urheilua käsittelevän materiaalin asema tulee muuttumaan monien palveluiden osalta viimeistään vuoden 2021 kesäkuuhun mennessä, jolloin EU-jäsenmaiden tulee asettaa maakohtaiset lait, jotka tukevat tätä EU:n päätöstä [Google 2019]. Tämän seurauksena esimerkiksi peliyhtiöiden tulee määritellä tekijänoikeuksiin, kuinka materiaalia kyseisestä pelistä saa jakaa verkossa. Lisäksi peliyhtiöt voivat jatkossa tehdä sopimuksia eri suoratoistopalveluiden kanssa, jotta niiden asema palvelukentällä paranee kyseistä peliä käsittelevän videomateriaalin myötä. Esimerkiksi peliyhtiö voi asettaa rajoituksen, jonka mukaan peliä käsittelevää sisältöä ei voi levittää tiettyjen palveluiden kautta. Tällaiset päätökset voivat nostaa joidenkin suoratoistoalustojen asemaa palvelukentällä ja puolestaan laskea toisten asemaa.

Tekijänoikeuksien puolustaminen tuo etuja sisällön tekijän näkökulmasta, mutta palveluiden kuluttajien näkökulmasta tämä on rajoittava tekijä, jonka avulla pystytään vaikuttamaan käyttäjien käyttämiin palveluihin. Sisällön rajoituksessa vain tietyille palvelulle, on tällä palvelulla monopoliasema jaettavaan sisältöön. Googlen tekemän ilmoituksen mukaan maakohtaiset rajoitukset vuoteen 2021 mennessä, tulevat näyttämään, kuinka YouTube tulee toimimaan kyseisissä maissa jatkossa [Google 2019]. On mahdollista, että automaattisen tunnistusalgoritmin heikkouksista johtuen EU:n vaatimaa ylläpitoa ei voida toteuttaa YouTuben suuruudessa palvelussa, joten palvelua ei voida tarjota maihin, joiden lainsäädäntöä YouTube rikkoisi. Myös palvelut, jotka perustuvat videopelien suoratoistolle, kuten Twitch ja Mixer, saattavat joutua muuttamaan palvelunsa toimintaa, mikäli peliyhtiöt eivät salli omien peliensä sisältöä lähetettäväksi palvelun kautta.

EU:n asettamat säädökset tulevatkin muokkaamaan paljon suurimpien palveluiden palvelumallia EU:n alueella. Vuoden 2021 kesäkuuhun mennessä tiedetään tarkemmin, miten verkossa toimivat suurimmat palvelut ottavat nämä haasteet vastaan ottaen huomioon, että jatkuvasti lisääntyvä videomateriaali myös lisää kuluja ylläpidon puolelta.

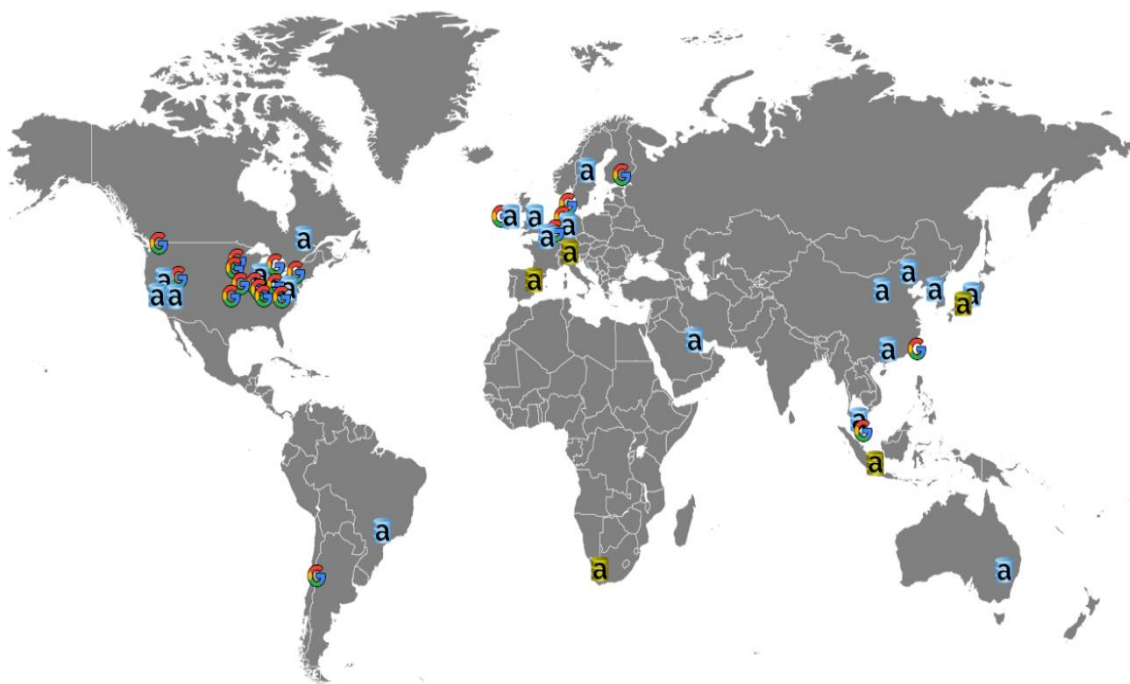
Videomateriaalin levityksen vallatessa verkon liikennettä ja maailman kääntäessä katsetta ilmastonmuutosta kohti, on useasti noussut esille myös verkon aiheuttamien päästöjen määrä ja näiden vaikutukset ilmastonmuutokseen. Vuonna 2017 ICT-alan (The Information and Communication Technology) ja palvelinkeskusten tuottamat hiilidioksidipäästöt olivat 2% kaikista maailman hiilidioksidipäästöistä [Avgerinou 2017]. Tämä päästömäärä on yhtä suuri kuin koko maailman lentoliikenteen tuottamien päästöjen yhteenlaskettu määrä.

Suoratoistettavan videomateriaalin määrän kasvaessa kovaa vauhtia joka minuutti, tulee jatkuvasti myös tarvetta tallentaa jaettava sisältö palvelimille, joista se on noudettavissa tarvittaessa. Palvelimien tallennuskapasiteetti on kuitenkin rajallinen, minkä vuoksi tarvitaan jatkuvasti suurempaa palvelinmäärää vastaamaan videoiden kasvaneeseen kysyntään. Tämän vuoksi pystytetään uusia palvelinkeskuksia ja lisätään palvelimia, jotta kasvun ei tarvitsisi tulla päätökseen. Jokainen palvelinkeskus ja niiden jokainen palvelin kuluttaa kuitenkin energiaa.

Ilmastonmuutos on tällä hetkellä yksi puhutuimmista aiheista maailmalla. Tämän seurauksena myös YK on reagoinut säädöksillä ilmastonmuutoksen vaikutuksien minimoimiseksi. Vuonna 2015 allekirjoitettu Pariisin ilmastopöytäkirjan mukaan Maan

keskilämpötilan nousu pyritään pitämään alle kahdessa celsiusasteessa [Avgerinou 2017]. Vuonna 2023 tehtävässä kokonaistarkastelussa pyritään selvittämään, miten hyvin tässä tavoitteessa on pysytty [YM 2015]. Tämän ja muiden ilmastopöytäkirjojen sekä maakohtaisen lainsäädännön seurauksena myös yritykset ja niiden omistamat palvelinkeskukset joutuvat rajaamaan tuottamansa päästöt sovitulle tasolle. Tämä näkyy myös päästöistä aiheutuviin kuluihin sekä käytettävän energian hinnoittelussa.

Internetin suurimpiin palveluntarjoajiin lukeutuvat Amazon ja Google ylläpitävät lukuisia palvelinkeskustoja ympäri maailmaa [Amazon 2020] [Google 2020]. Tämä palvelinkeskustojen levittyminen on esitetty kuvassa 6.1. Laaja-alaisen palveluiden tarjoaminen vaatii jatkuvaa ylläpitoa ja skaalautumista olosuhteiden mukaan. Tämä näkyy myös esimerkiksi Amazonin ilmoittamissa suunnitelmissa uusien palvelinkeskustojen pystyttämiseksi ympäri maailmaa.



(Kuva 6.1: Googlen ja Amazonin olemassa olevat ja suunnitellut palvelinkeskukset.)

Amazon on ilmoittanut sivuillaan lisäävänsä palvelinkeskustoja mm. Indonesiaan, Italiaan, Etelä-Afrikkaan, Espanjaan ja Japaniin [Amazon 2020] (Kuva 6.1 keltaisella taustalla olevat Amazon-palvelimet). Amazonin tarjoamien palveluiden tuottaman tiedon säilyttäminen ja palveluiden käyttäjäkunnan hajanaisuus asettaa kuitenkin omia

haasteita yhtiön toimintaan ja palvelinkeskusten ylläpitoon lähitulevaisuudessa, sillä maakohtaiset lainsäädännöt ja asetukset voivat poiketa toisistaan merkittävästi.

7 Yhteenveto

Videoiden suoratoisto on noussut aivan uudelle tasolle viime vuosien aikana. Aikaisemmin käytetyt suoratoistomenetelmät ja latauskäytänteet ovat vaihtuneet dynaamisesti mukautuviin suoratoistomenetelmiin, joiden avulla pyritään takaamaan korkealaatuinen ja luotettava käyttökokemus palveluiden käyttäjille. Kulutuskäytänteiden murroksessa tämä ei kuitenkaan ole ollut helppo tehtävä, sillä käyttäjien mobiliteetti on luonut uusia haasteita mm. toistolaitteiden heterogeenisyydellä ja puskurointivaran vaihtelevuuksilla. Palveluiden on pystyttävä tarjoamaan korkea saatavuus missä vain ja milloin vain.

Suoratoistoalan suurimmat tekijät, kuten YouTube ja Netflix ovat siirtyneet käyttämään DASH-suoratoistostandardia, jonka kehityksen taustalla on lukuisia tietotekniikka-alan suurimpia tekijöitä, kuten Google, Netflix, Microsoft ja Akamai [Bitmovin 2019] [DASH 2019]. Tämän suoratoistomenetelmän avulla pystytään reagoimaan verkon olosuhteiden muutoksiin tai käyttäjien tekemiin valintoihin suoratoiston aikana. Suoratoistettavan sisällön laatua heikentämällä tai parantamalla pystytään reagoimaan käytettävän kaistan leveyteen ja tarjoamaan sujuva käyttökokemus mm. ilman liiallisia puskurointiviiveitä.

Videoiden suoratoiston tulevaisuus näyttää tällä hetkellä varsin kirkkaalta ja suoratoisto elää uusimpien teknologioiden mukana. Langattoman teknologian ensiaskeleet mm. 5G-datayhteyden parissa tulevat edesauttamaan suoratoistopalveluiden käyttökokemusta suuremman kaistanleveyden myötä. Myös reunalaskennan hyödyntäminen voi tarjota pienempää vasteaikaa paikallisesti tarjottavien videoiden kulutukseen ja auttaa liiallisen ruuhkautumisen ennaltaehkäisyssä. Se, kuinka suuri hyöty tästä loppuen lopuksi tulee tavalliseen pilveen verrattuna, jää nähtäväksi. Suurimmat suoratoistopalveluiden tarjoajat jatkavat kuitenkin levittäytymistä ympäri maapalloa, laajentaen näin pilven ulottuvuutta. Kun pilvi laskeutuu tarpeeksi lähelle käyttäjiä, katoaa reunalaskennan potentiaalinen hyöty monilta osin.

Pilven liikkua palvelinkeskuksien myötä lähemmäs ja lähemmäs käyttäjiä, myös palveluiden ylläpitoressurssien tarve nousee. Tämän seurauksena yhteiskunta on havahtunut

kulutuksen kasvuun ja mm. palvelinkeskuksien tuottaman lämpöenergian määrään. Tätä kasvua ja verkon laajamittaista kulutusta rauhoittamaan on tehty kansainvälisiä ja kansallisia säädöksiä, joiden avulla on pyritty tasoittamaan verkon kulutusta eri toimijoiden välillä. Tämän seurauksena myös verkon neutraliteettia on rajoitettu kansallisilla mittapuilla.

Tutkielmassa on esitelty suoratoiston kannalta tärkeimpiä tekijöitä sekä tärkeimpiä kehityskaareen kuuluneita suoratoistomenetelmiä. Näiden avulla on havainnollistettu looginen jatkumo aiempien suoratoistoratkaisujen ja nykyisen suoratoiston tilan välille. Tämän lisäksi tutkielmassa on esitelty tällä hetkellä vallitsevat mukautuvat suoratoistomenetelmät, joista keskiöön on nostettu yleisesti käytössä oleva DASH-suoratoistostandardi.

Suoratoiston tulevaisuutta on pyritty analysoimaan eri näkökulmista. Teknologista kehitystä on havainnollistettu esittelemällä nousevia, langattomia, teknologioita ja pakkausstandardeja sekä reunalaskennan hyödyntämistä videoiden suoratoistossa. Myös yhteiskunnallinen näkökulma on tuotu esiin mm. kansainvälisesti asetettujen rajoitusten ja kansallisten päätösten muodossa. Tämän lisäksi tutkielmassa on esitetty suurimpien suoratoistopalveluiden kykyä ja tapaa reagoida kasvaneeseen kysyntään ja tarjontaan.

Kaiken kaikkiaan suoratoiston tulevaisuus näyttää varsin valoisalta. Jatkuvasti kasvava kysyntä ja tarjonta ruokkivat toinen toisiaan ja uudet teknologiat mahdollistavat entistäkin sujuvamman suoratoiston käyttäjien laitteisiin ympäri maailmaa. Vaikka yhteiskunnallisia rajoituksia tehdäänkin suoratoiston liikennöintiin, ei sen kasvulle näy loppua. Vain aika näyttää, minne nousevat vaatimukset tulevat johdattamaan videoiden suoratoiston lähitulevaisuudessa.

8 Lähteet

- [Akamai 2020] “Chunked Transfer Encoding.” Akamai Technologies. Accessed January 21, 2020. <https://learn.akamai.com/en-us/webhelp/media-services-live/media-services-live-encoder-compatibility-testing-and-qualification-guide-v4.0/GUID-A7D10A31-F4BC-49DD-92B2-8A5BA409BAFE.html>.
- [Amazon 2020] “Debt Information for Teens: Tips for a Successful Financial Life, Including Facts about the Economy and Personal Finances, Money Management, Interest Rates, Loans, Credit Cards, Predatory Lending Practices, and Resolving Debt-Related Problems.” Amazon. Omnigraphics, 2018. <https://aws.amazon.com/about-aws/global-infrastructure/>.
- [Andrews 2014] J. G. Andrews et al., "What Will 5G Be?," in *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065-1082, June 2014.
- [Apple 2016] Apple Inc. “What's New in HTTP Live Streaming - WWDC 2016 - Videos.” *Apple Developer*, <https://developer.apple.com/videos/play/wwdc2016/504/>.
- [Apostolopoulos 2003] Apostolopoulos, J.G. & Tan, W.-T & Wee, S.J.. (2003). "Video streaming: Concepts, algorithms, and systems."
- [Avgerinou 2017] Avgerinou, Maria, Paolo Bertoldi, and Luca Castellazzi. “Trends in Data Centre Energy Consumption under the European Code of Conduct for Data Centre Energy Efficiency.” *Energies* 10, no. 10 (2017): 1470. <https://doi.org/10.3390/en10101470>.
- [Bettahar 2005] Bettahar, Hatem. (2005). Tutorial on multicast video streaming techniques.
- [Bitmovin 2019] “Why YouTube & Netflix Use MPEG-DASH in HTML5.” Bitmovin, October 16, 2019. <https://bitmovin.com/status-mpeg-dash-today-youtube-netflix-use-html5-beyond/>.

- [Bocchi 2016] Bocchi, Enrico, Luca De Cicco, and Dario Rossi. “Measuring the Quality of Experience of Web Users.” *Proceedings of the 2016 Workshop on - Internet-QoE 16*, 2016.
<https://doi.org/10.1145/2940136.2940138>.
- [Böttger 2018] Böttger, Timm, Felix Cuadrado, Gareth Tyson, Ignacio Castro, and Steve Uhlig. “Open Connect Everywhere.” *ACM SIGCOMM Computer Communication Review* 48, no. 1 (2018): 28–34.
<https://doi.org/10.1145/3211852.3211857>.
- [Cisco 2000] *Internetworking Technologies Handbook*. Cisco, 2000.
- [Cisco 2011] “HTTP versus RTMP: Which Way to Go and Why?” Accessed April 7, 2020. https://www.bogotobogo.com/VideoStreaming/Files/Video_streaming_etc/HTTPvsRTMP.pdf.
- [Cisco 2019] VNI Global Fixed and Mobile Internet Traffic Forecasts. (2019, November 6). Retrieved from: <https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html#~complete-forecast>.
- [Collins 2018] Collins, Keith. “Net Neutrality Has Officially Been Repealed. Here's How That Could Affect You.” *The New York Times*. The New York Times, June 11, 2018. <https://www.nytimes.com/2018/06/11/technology/net-neutrality-repeal.html>.
- [DASH 2014] ISO/IEC 23009-1:2014(E) Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats
- [DASH 2019] “DASH Industry Forum: Catalyzing the Adoption of MPEG.” *DASH*, <https://dashif.org/members/>.
- [EU 2020] Council of the European Union: Proposal for a DIRECTIVE OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on copyright in the Digital Single Market - Agreed negotiating mandate. Accessed April 28, 2020. <https://www.consilium.europa.eu/media/35373/st09134-en18.pdf>

- [Fielding 1999] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). Hypertext Transfer Protocol -- HTTP/1.1. doi: 10.17487/rfc2616
- [Fisher 2014] Fisher, Y. "Paper - *An Overview Of HTTP Adaptive Streaming Protocols For TV Everywhere Delivery* - NCTA Technical Papers." *Tech Paper Database*, <https://www.nctatechnicalpapers.com/Paper/2014/2014-an-overview-of-http-adaptive-streaming-protocols-for-tv-everywhere-delivery>.
- [Google 2019] "Updates on Article 17 (Formerly Article 13) - YouTube Community." Google. Google. Accessed March 12, 2020. <https://support.google.com/youtube/thread/17592587?hl=en>.
- [Google 2020] "Discover Our Data Center Locations." Google. Google. Accessed March 12, 2020. <https://www.google.com/about/datacenters/locations/>.
- [HTTPWatch 2020] "HTTP Chunked Encoding." HttpWatch. Accessed January 21, 2020. <https://www.httpwatch.com/httpgallery/chunked/>.
- [IEEE 2020] "Full Page Reload." IEEE Spectrum: Technology, Engineering, and Science News. Accessed April 7, 2020. <https://spectrum.ieee.org/video/telecom/wireless/5g-bytes-millimeter-waves-explained>.
- [IETF 2020] Transmission Control Protocol. (n.d.). Retrieved January 6, 2020, from <https://tools.ietf.org/html/rfc793>.
- [Kafka 2018] Kafka, Peter. "70 Percent of Netflix Viewing Happens on TVs." Vox. Vox, March 8, 2018. <https://www.vox.com/2018/3/7/17094610/netflix-70-percent-tv-viewing-statistics>.
- [Kesavan 2015] Kesavan, Selvaraj & Jayakumar, J.. (2015). Dynamic adaptive streaming over HTTP (DASH)-comprehensive study and rate adaptation performance analysis. *International Journal of Soft Computing*. 10. 261-273.
- [Laforge 2017] Laforge, Anthony. "Saying Goodbye to Flash in Chrome." *Google*,

Google, 25 July 2017, <https://www.blog.google/products/chrome/saying-goodbye-flash-chrome/>.

- [Li 2013] Li, Baochun, Zhi Wang, Jiangchuan Liu, and Wenwu Zhu. “Two Decades of Internet Video Streaming.” *ACM Transactions on Multimedia Computing, Communications, and Applications* 9, no. 1s (January 2013): 1–20. <https://doi.org/10.1145/2505805>.
- [Li 2019] Li, Fangfan, Arian Akhavan Niaki, David Choffnes, Phillipa Gill, and Alan Mislove. “A Large-Scale Analysis of Deployed Traffic Differentiation Practices.” *Proceedings of the ACM Special Interest Group on Data Communication - SIGCOMM 19*, 2019. <https://doi.org/10.1145/3341302.3342092>.
- [Madhvapaty 2014] Madhvapaty, Mr. Havish, and Ms. Shiva Goyal. “Net Neutrality – A Look at the Future of Internet.” *IOSR Journal of Computer Engineering* 16, no. 4 (2014): 71–77. <https://doi.org/10.9790/0661-16427177>.
- [Mehrabi 2019] Mehrabi, Abbas, Matti Siekkinen, and Antti Yla-Jaaski. “Edge Computing Assisted Adaptive Mobile Video Streaming.” *IEEE Transactions on Mobile Computing* 18, no. 4 (January 2019): 787–800. <https://doi.org/10.1109/tmc.2018.2850026>.
- [Microsoft 2010] “Integrating Microsoft IIS Smooth Streaming.” Silverlight Recipes, 2010, pp. 949–985., doi:10.1007/978-1-4302-3034-2_11.
- [Mok 2011] Mok, R. K. P., Chan, E. W. W., & Chang, R. K. C. (2011). Measuring the quality of experience of HTTP video streaming. *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*. doi: 10.1109/inm.2011.5990550
- [Nada 2006] Nada, Fayza A. “On Using Mobile IP Protocols.” *Journal of Computer Science* 2, no. 2 (January 2006): 211–17. <https://doi.org/10.3844/jcssp.2006.211.217>.
- [Netflix 2020] “About Netflix.” Netflix Media Center. Accessed January 21, 2020. <https://media.netflix.com/en/about-netflix>.

- [Netflix 2020 (1)] “Quarterly Earnings.” Netflix. Accessed January 21, 2020. <https://www.netflixinvestor.com/financials/quarterly-earnings/default.aspx>.
- [Netflix 2020 (2)] “Sign In.” Netflix. Accessed April 28, 2020. <https://www.netflix.com/settings/playback/>.
- [Nightingale 2018] Nightingale, James, Pablo Salva-Garcia, Jose M. Alcaraz Calero, and Qi Wang. “5G-QoE: QoE Modelling for Ultra-HD Video Streaming in 5G Networks.” *IEEE Transactions on Broadcasting* 64, no. 2 (2018):621–34. <https://doi.org/10.1109/tbc.2018.2816786>.
- [Pires 2014] Pires, Karine, and Gwendal Simon. “DASH in Twitch.” *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming - VideoNext 14*, 2014. <https://doi.org/10.1145/2676652.2676657>.
- [Pires 2015] Pires, Karine, and Gwendal Simon. “YouTube Live and Twitch.” *Proceedings of the 6th ACM Multimedia Systems Conference on - MMSys 15*, 2015. <https://doi.org/10.1145/2713168.2713195>.
- [Porambage 2018] Porambage, Pawani, Jude Okwuibe, Madhusanka Liyanage, Mika Ylianttila, and Tarik Taleb. “Survey on Multi-Access Edge Computing for Internet of Things Realization.” *IEEE Communications Surveys & Tutorials* 20, no. 4 (2018): 2961–91. <https://doi.org/10.1109/comst.2018.2849509>.
- [Rao 2011] Rao, A., Legout, A., Lim, Y.-S., Towsley, D., Barakat, C., & Dabbous, W. (2011). Network characteristics of video streaming traffic. Proceedings of the Seventh Conference on Emerging Networking EXperiments and Technologies on - CoNEXT 11. doi: 10.1145/2079296.2079321
- [Salama 2000] Salama, P., N.b. Shroff, and E.j. Delp. “Error Concealment in MPEG Video Streams over ATM Networks.” *IEEE Journal on*

- Selected Areas in Communications* 18, no. 6 (2000): 1129–44.
<https://doi.org/10.1109/49.848263>.
- [Sandvine 2019] “The Global Internet Phenomena Report - Sandvine.” Accessed February 25, 2020.
[https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/Internet Phenomena/Internet Phenomena Report Q32019 20190910.pdf](https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/Internet%20Phenomena/Internet%20Phenomena%20Report%20Q32019%20190910.pdf).
- [Sodagar 2011] Sodagar, I. (2011). The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE Multimedia*, 18(4), 62–67. doi: 10.1109/mmul.2011.71
- [Spiteri 2018] Spiteri, K., Sitaraman, R., & Sparacio, D. (2018). From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player. *Proceedings of the 9th ACM Multimedia Systems Conference on - MMSys 18*. doi: 10.1145/3204949.3204953
- [Stockhammer 2011] Stockhammer, T. (2011). Dynamic adaptive streaming over HTTP -- Design Principles and Standards. *Proceedings of the Second Annual ACM Conference on Multimedia Systems - MMSys 11*. doi: 10.1145/1943552.1943572
- [Twitch 2020] “Following.” Twitch.tv. Accessed February 25, 2020.
<https://www.twitch.tv/p/partners/>.
- [Twitch 2020 (1)] “Twitch Affiliate Program.” Twitch Affiliate. Accessed February 25, 2020. <https://affiliate.twitch.tv/>.
- [Twitch 2020 (2)] Customer Support. Accessed February 25, 2020.
https://help.twitch.tv/s/article/guide-to-broadcast-health-and-using-twitch-inspector?language=en_US.
- [Vakali 2003] Vakali, A., and G. Pallis. “Content Delivery Networks: Status and Trends.” *IEEE Internet Computing*, vol. 7, no. 6, 2003, pp. 68–74., doi:10.1109/mic.2003.1250586.
- [Wu 2001] Wu, Dapeng, et al. “Streaming Video over the Internet: Approaches and Directions.” *IEEE Transactions on Circuits and*

Systems for Video Technology, vol. 11, no. 3, 2001, pp. 282–300.,
doi:10.1109/76.911156.

- [YM 2015] “Pariisin Ilmastopimus 2015: Ympäristöministeriö.” 2015 | ympäristöministeriö. Accessed March 12, 2020.
https://www.ymp.fi/fi-FI/Ymparisto/Ilmasto_ja_ilma/Ilmastomuutoksen_hillitseminen/Kansainvaliset_ilmastoneuvottelut/Pariisin_ilmastopimus.
- [YouTube 2020] YouTube. YouTube. Accessed January 21, 2020.
<https://www.youtube.com/intl/en-GB/about/press/>.
- [Zambelli 2009] Zambelli, A. *Smooth Streaming Technical Overview - Bogotobogo.com*.https://www.bogotobogo.com/VideoStreaming/Files/iis8/IIS_Smooth_Streaming_Technical_Overview.pdf.