

---

# **MPEG-DASH Live Streaming**



Ammattikorkeakoulun opinnäytetyö

Tietotekniikan ko.

Riihimäki, kevät 2016

Juha Koski



RIIHIMÄKI  
Tietotekniikan ko.  
Mediajärjestelmät

---

<b>Tekijä</b>	Juha Koski	<b>Vuosi</b> 2016
<b>Työn nimi</b>	MPEG-DASH Live Streaming	

---

## TIIVISTELMÄ

Tämän opinnäytetyön tarkoituksena oli selvittää Otavamedia OMA Oy:lle MPEG-DASH-suoratoistotekniikan käytännön live toteutuksen mahdollisuudet ja samalla syventyä tekniikkaan teoriatasolla. Työn käytännön osuuden tavoitteena oli luoda ympäristö, jossa voidaan toteuttaa suoratoistoa MPEG-DASHin välityksellä ja testata tekniikkaa tulevaisuuden käyttöä varten. Opinnäytetyö toimisi tulevaisuudessa myös lähdemateriaalina, kun tarvitaan lisätietoa tekniikasta asiakkaalle.

Työn teoriaosuudessa käsitellään yleisesti suoratoistoa ja sen prosessia. Tarkemmin syvennytään MPEG-DASHiin ja selvitetään sen rakenne ja toimintaperiaatteet. Tekniikan sovellukset pohjautuvat vahvasti DASH-IF:n laatimiin ohjeisiin, jotka antavat suositukset tekniikan alkuvaiheen käyttöönottoon.

Käytännön osuudessa luotiin avoimeen lähdekoodiin perustuva järjestelmä, jolla voitiin tuottaa RTMP-streamista MPEG-DASH-muotoinen live-lähetys. Järjestelmä oli kamerasta mediapalvelimeen täysin omassa hallinnassa, joten pääsimme vaikuttamaan koko tuotantoketjuun ja näin paremmin ymmärtämään MPEG-DASHia tekniikkana.

Lopputuloksena havaittiin MPEG-DASHin olevan erittäin käyttökelpoinen ja hyvää laatua tuottava tekniikka, joka tulee pitkälti korvaamaan nykyisen HLS-suoratoistotekniikan. Tekniikan yleistymistä hidastaa lähinnä Applen päätös jättäytyä pois DASHin käyttöönotosta. Muilla päätelaitteilla ja sovelluksilla DASHilla on jo nyt tukeva asema ja se tulee olemaan avainasemassa etenkin 360-toteutuksissa ja HbbTV:n parissa.

**Avainsanat** Suoratoisto, live-lähetys, webcast

**Sivut** 44 s.

RIIHIMÄKI

Degree Programme in Information Technology  
Media Systems

---

**Author**

Juha Koski

**Year** 2016

**Subject of Bachelor's thesis**

MPEG-DASH Live Streaming

---

## ABSTRACT

The purpose of this thesis was to examine MPEG-DASH live streaming for Otavamedia OMA Oy. The main goal was to find out if it were possible to use MPEG-DASH as the main streaming standard for webcasts and also to provide a technical insight into the protocol itself. This was to be used as reference information for clients.

The technical part of this thesis contains general information on live streaming and explains the general workflow of producing a live broadcast over the internet. MPEG-DASH is studied more thoroughly and this thesis contains the guidelines for and early implementation of MPEG-DASH provided by DASH IF organization.

The empirical part consists of creating a lab environment based on open source technology. The system was able to create an MPEG-DASH from ingested RTMP-source.

In our testing process we found out that MPEG-DASH produces great quality with little effort on the encoding side. DASH will most likely replace existing streaming technologies, except HLS which will remain in Apple's products. DASH's on-demand-capabilites are already in extensive commercial use and it is and will be a major part in both 360-streams and in HbbTV.

**Keywords** Live streaming, webcast, broadcasting

**Pages** 44 p.

# SISÄLLYS

1	JOHDANTO.....	1
2	SANASTO.....	1
3	TEKNISET VAATIMUKSET .....	4
4	LIVE STREAMING.....	5
4.1	Suoratoisto.....	5
4.1.1	Adaptiivinen suoratoisto.....	5
4.1.2	RTMP .....	6
4.2	MPEG-DASH ja suoratoiston workflow.....	7
4.2.1	Tuotanto.....	7
4.2.2	Välitys.....	7
4.2.3	Toisto.....	8
4.3	Jakelu.....	8
5	KOODEKIT .....	8
5.1	H.264.....	10
5.1.1	Profiilit ja tasot .....	10
5.1.2	Entropiakoodaus .....	11
5.1.3	Enkoodaus .....	12
5.2	H.265.....	15
5.3	VP8 ja VP9.....	17
6	DASH - DYNAMIC ADAPTIVE STREAMING OVER HTTP.....	17
6.1	Enkoodaus ja rakenne.....	19
6.2	Profiilit .....	20
6.3	MPD – Media Presentation Description .....	21
6.4	MPD:n rakenne .....	22
6.4.1	Periods .....	22
6.4.2	Adaptation sets .....	22
6.4.3	Representations.....	23
6.4.4	Media Segments .....	23
6.5	MPEG-DASH Live .....	24
6.6	HBBTV .....	25
6.7	HTML5.....	25
6.8	MSE - Media Sources Extensions.....	27
7	PALVELUN TESTAUS .....	27
7.1	Nginx-konfiguraatio.....	29
7.2	Käytettävät sovellukset .....	31
7.2.1	DASH.js.....	32
7.2.2	Testiympäristön MPD .....	33
7.3	Tulokset.....	36
8	YHTEENVETO .....	38
	LÄHTEET .....	39



## 1 JOHDANTO

Opinnäytetyöni tarkoituksena on kartoittaa Otavamedia OMA Oy:lle MPEG-DASH-suoratoistotekniikan käytännön live-toteutuksen mahdollisuudet ja samalla syventyä tekniikkaan teoriatasolla. Työn käytännön osuuden tavoitteena on luoda ympäristö, jossa voidaan toteuttaa suoratoistoa MPEG-DASHin välityksellä ja testata tekniikkaa tulevaisuuden käyttöä varten. Opinnäytetyö toimisi tulevaisuudessa myös lähdemateriaalina, kun tarvitaan lisätietoa tekniikasta asiakkaalle.

MPEG-DASH on vielä nuori streaming-tekniikka ja sen yleistymistä vaikeuttavat heikko tuki yleisimmiltä CDN-toimijoilta ja selaimilta. Tällä hetkellä tekniikkaa tukevien yhtiöiden, lähinnä TV-kanaville tarkoitettujen, tuotteiden korkeat kustannukset ovat liikaa pienimuotoiseen testiympäristökäyttöön.

Webcast-palveluita tarjoavat yrityksen joutuvat tasapainoilemaan kustannusten, laitetuen ja laadun kanssa. Viime aikoina erityisesti Flash-tuen jatkuva kaventuminen eri laitteissa on vaikeuttanut toimintaa. Moni saattaa katsoa esim. webcast-lähetystä yrityksen tulosjulkistuksesta kännykällä tai huonojen yhteyksien päässä. Käytännössä olisi siis suuri tarve tekniikalle, joka on laadultaan hyvä ja mahdollisimman monelle laitteelle sopiva. Juuri tähän tarpeeseen luotiin MPEG-DASH.

DASHin tuki HEVC/H.265:lle on myös oleellinen, sillä se tarjoaa huomattavasti paremman laadun pienemmällä bittivirralla. H.265 tulee olemaan myös tulevaisuuden pakkausmuoto videolle, ja se tulee pitkälti korvaamaan nykyisen H.264:n videon suoratoistossa. Erityisen hyödyllinen se tulee olemaan, kun resoluutiot kasvavat ja tarve esim. 360-lähetysille kasvaa.

## 2 SANASTO

.MP4		Applen kehittämä säiliömuoto. Voi sisältää kuvaa ja ääntä. Enkoodattu H.264.
AAC	Advanced Audio Coding	Häviöllinen äänenpakkausmuoto. H.264:n audioformaatti.
Adaptiivinen Suoratoisto	Adaptive streaming	Päätelaitteen ja yhteyden mukaan laatua säätelevä suoratoistotekniikka.
Akamai		Pilvipalveluita tuottava yhtiö. Maailman suurimpia CDN-toimijoita.

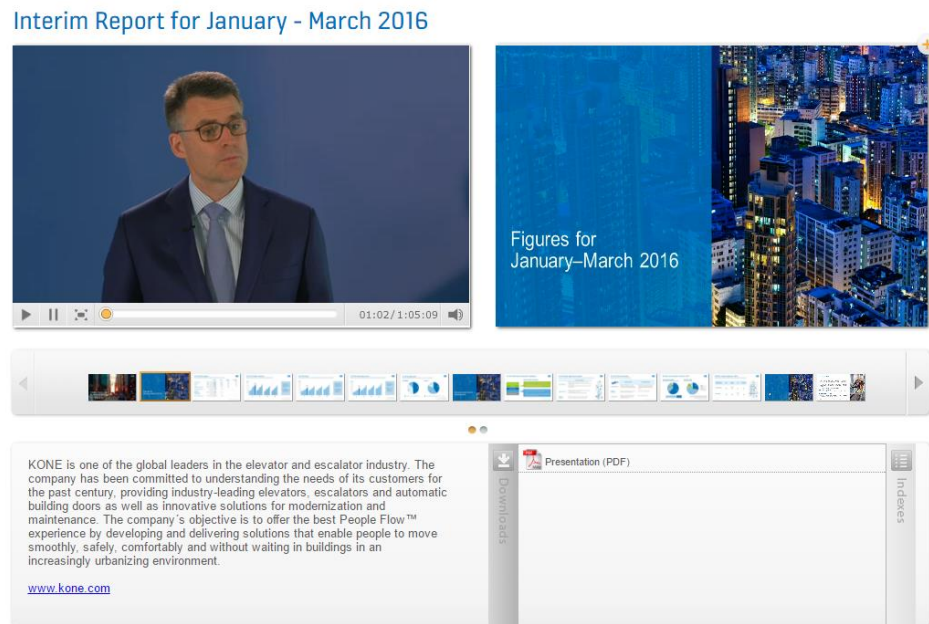
CDN	Content Delivery Network	Useita paikallisia palvelimia hyödyntävä sisällönjakoverkosto
Dekooderi	Decode	Purkaa enkooderilla pakatun datan käänteisellä prosessilla
Enkooderi	Encoder	Pakkaa dataa tietyn algoritmin mukaan.
Flash		Kehitysympäristö ja tekniikka multimediaesityksiin verkossa ja mobiililaitteissa
Frame		Yksittäinen videovirran kuva.
GOP	Group Of Pictures	Videonpakkauksen IPB-kuvien sarja
H.264		H.264, eli MPEG-4 AVC on videonpakkauksen standardi.
HbbTV	Hybrid Broadcast Broadband TV	Standardi, joka yhdistää television, laajakaistan ja IPTV:n ominaisuuksia
HEVC/H.265	High Efficiency Video Coding	HEVC, eli MPEG-4 part 10, on videonpakkauksen standardi. Tukee mm. jopa 8K-resoluutiota.
HLS	HTTP Live Streaming	Applen kehittämä live-esityksiin tarkoitettu suoratoistotekniikka.
HTTP	Hypertext Transfer Protocol	Selainten ja palvelinten käyttämä tiedonsiirto-protokolla.
iOS		Applen mobiililaitteiden käyttöjärjestelmä.
Koodekki	Codec	Algoritmi, joka pakkaa ja purkaa ääni- ja kuvasignaalia
MPD	Media Presentation Description	MPEG-DASHin soittolistarakenteen, XML-tiedosto
MPEG	Moving Picture Experts Group	Uusia videon ja äänen pakkaamiseen tarkoitettuja standardeja kehittävä organisaatio.
MPEG-DASH	MPEG Dynamic Adaptive Streaming over HTTP	Adaptiivinen suoratoistotekniikka, joka toimii HTTP:n yli.

MSE	Media Source Extensions	Javascript-kirjasto, joka mahdollistaa HTML5-videon
Nginx		Palvelinohjelmisto
Näytteistys		Monimutkaisen datan ilmaiseminen yksinkertaisemmin esim. vain kokonaislukujen avulla.
OGG		Avoimen standardin äänenpakkausmuoto. WebM:n ääniformaatti.
RTMP	Real Time Messaging Protocol	Tiedonsiirtoprotokolla äänen, videon ja datan siirtämiseen. Alun perin tarkoitettu Flash-playerin ja palvelimen väliseen tiedonsiirtoon.
Transkoodaus		Tiedoston enkoodausta pakatusta muodosta toiseen.
UHD	Ultra High Definition	4K ja 8K resoluution videokuva
WebM		Maksuton videotiedostomuoto
VOD	Video On Demand	Ladattava bittimuotoinen videotiedosto, esim. Youtuben video joka ei ole suora lähetyks.
VP8		WebM:n avoimeen lähdekoodiin perustuva videonpakkausmuoto.



### 3 TEKNISET VAATIMUKSET

Otavamedia OMA Oy:n tyypillinen webcast-toteutus on noin puolen tunnin suora lähetys esim. yrityksen osavuositiedotuksesta. Tilaisuus kuvataan usein yhdellä Sony PMW-EX3 kameralla. Kuva ja audio mikseristä viedään enkoodausta suorittavalle työasemalle. Laitteisto muodostaa medialähteistä RTMP-streamin, joka lähetetään Akamain serverille jakelua varten. Kuvassa 1 on esitetty tyypillisen webcast-toteutuksen näkymä asiakkaalle.



Kuva 1. Webcast-näkymä asiakkaalle (Otavamedia OMA Oy 2016)

Tulevaisuudessa Flash-tuki tulee vähenemään kaikilla alustoilla, ja tähän ongelmaan MPEG-DASH on erinomainen ratkaisu. Suuri osa netin suoratoistopalveluista tuottavista toimijoista on jo siirtynyt HTML5-pohjaisiin videoratkaisuihin ja selaimetkaan eivät oletuksena avaa Flash-pohjaista soittainta (Wolynski, Ribeiro 2016). Flashin vähenemisestä kertoo vahvasti myös se, että Google siirtyy 2017 täysin HTML5-pohjaiseen mainosten animointiin ja videoon (Hughes 2016). Applen laitteet eivät kuitenkaan tällä hetkellä MPEG-DASHia tue, joten jakelua varten tarvitaan edelleen ratkaisu, joka muuntaa vanhan RTMP-streamin sopivaan muotoon. Käytännössä siis muille laitteille MPEG-DASH ja Applen laitteille HLS.

Lopulta OMA Oy:n ideaalitalanne olisi Akamailta tarjottu palvelu, joka hoitaa koko prosessin. Tällä hetkellä tekniikka on niin uusi, että sovelluksia ei vielä ole tarjolla. Testikäyttöön rakennettiin siis alusta asti järjestelmä, jossa päästiin vaikuttamaan koko lähetysketjuun. Tavoitteena oli myös tuottaa mahdollisimman kevyt ja ”roudattava” toteutus.

## 4 LIVE STREAMING

### 4.1 Suoratoisto

Nykyisen kaltainen live streaming, eli suoratoisto, sai alkunsa jo 90-luvulla, kun Real Networks lähetti audio-muotoisen lähetyksen baseball-pelistä vuonna 1995. Real Networks olikin alkuaikoina hallitseva teknologia suoratoistossa, ja vuonna 2000 noin 80 prosenttia internetin suoratoistosta perustui Real-formaattiin. Realin toiminta perustui servereistä saataviin maksuihin, ja toiminta kuihtui nopeasti Microsoftin tultua markkinoille. Microsoft tarjosi tuotteita edullisesti tai jopa ilmaiseksi. Tästä Microsoft joutuikin maksamaan Realille 761 miljoonaa oikeuskiistan jälkeen.

Microsoft hallitsi suoratoiston markkinoita 2000-luvun alkupuoliskon, aina Flashin yleistymiseen asti. Lopullisesti suoratoisto siirtyi Flash-aikakauden, kun Macromedia julkaisi vuonna 2005 VP6-koodekin, joka tarjosi aiempaan verrattuna hyvälaatuisen videokuvan. Flash oli aiemminkin sisältänyt videokomponentin, mutta laatu ei pärjännyt alkuaikoina Microsoftin Media Playerille.

Flash mullisti suoratoiston ja tarjosi ensimmäistä kertaa nettisivuille integroidun videon, joka toimi käytännössä kaikilla laitteilla. Microsoft julkaisi vielä oman tekniikkansa Silverlightin, mutta se ei koskaan noussut suureen suosioon flashiin verrattuna.

Vuonna 2010 julkaistiin iPad, joka ei tukenut ollenkaan Flashiä. Laitteen suuri suosio alkoi siirtämään suoratoiston painopistettä HTML5-pohjaisiin ratkaisuihin. HTML5 mahdollistaa median toistamisen selaimen omalla soittimella, eikä vaadi ulkoisia ohjelmistoja. (Ozer 2011.)

Flash on heikosti tuettu etenkin mobiililaitteissa, ja tietoturvastakin on paljastunut useita puutteita. Nykypäivänä HTML5-pohjaiselle suoratoistotekniikalle on suuri tarve. Tähän tarpeeseen pyrkii vastaamaan MPEG-DASH, joka laiteriippumattomasti yhdistelee adaptiivisen ja dynaamisen suoratoiston tekniikoita.

#### 4.1.1 Adaptiivinen suoratoisto

Adaptiivinen suoratoisto, eli Adaptive Bitrate Streaming (ABS), on vastaanottavan clientin suorituskyvyn mukaan mukautuva suoratoistotekniikka. Tyypillisesti HTTP:n yli toimiva tekniikka tarkkailee dynaamisesti prosessorin, muistin ja verkkoyhteyden laatua ja säätää kuvanlaadun mahdollisimman hyväksi niin, että video pyörii katkeamatta.

ABS-tekniikassa video enkoodataan usealle bittinopeudelle ja jokainen niistä segmentoidaan pieniin osiin. Tyypillisesti segmentin pituus vaihtelee 2-10 sekunnin välillä. Käyttäjän videosoitin voi sitten hyödyntää tätä vaihtamalla parhaaksi katsomansa laadun käyttäjän yhteyden mukaan. Eri laatuisten videoiden keyframe-väljen tulisi olla sama, jotta ero niiden välillä ei

kasvaisi liian suureksi. Palvelin vaihtaa laatua keyframen kohdalla, joten liian pitkä väli voi aiheuttaa viivettä palvelimen puolella. (Kapoor 2009.)

Segmenttien ja eri bit rate -vaihtoehtojen lista löytyy manifestitiedostosta. Kun käyttäjä avaa suoratoiston, kone pyytää ensin huonolaatuisimman segmentin. Mikäli soitin havaitsee, että suorituskyky riittää parempilaatuisen videon katsomiseen, pyytää se seuraavaksi korkeamman bit raten segmentin. Lopputuloksena on yleensä lyhyt bufferointiaika ja hyvä katselukokemus useilla eri nopeuden yhteyksillä.

Viime vuosina adaptiivinen suoratoisto on ollut markkinoilla neljän eri tekniikan muodossa:

HDS - Adobe HTTP Dynamic Streaming

HLS - Apple HTTP Live Streaming

Microsoft Smooth Streaming

MPEG-DASH – Dynamic Adaptive Streaming over HTTP

Näistä neljästä etenkin HLS on erittäin käytetty tekniikka live-lähetyksissä. Adaptiivisten suoratoistotekniikoiden joukosta HLS on käytetyin, noin 75 prosentin markkinaosuudella (Highwinds 2015). Tulevaisuudessa sen tukeutuminen Flashiin soitinkomponenttipuolella tulee kuitenkin vähentämään runsaasti sen käyttöä. Applen laitteissa se tulee pysymään todennäköisesti pitkään ainoana kunnolla tuettuna livelähetyksen muotona.

Edellä mainituista tekniikoista MPEG-DASH on ainoa, joka on kansainvälinen ISO-standardi. Tekniikan standardoimisella pyritään tarjoamaan alustariippumaton ratkaisu, joka ei ole kaupallisen yhtiön tuote. Se myös pohjautuu täysin HTML5-tekniikkaan, joka etenkin tulevaisuudessa tulee olemaan tuettu miltei jokaisessa laitteessa. (Encoding 2014.)

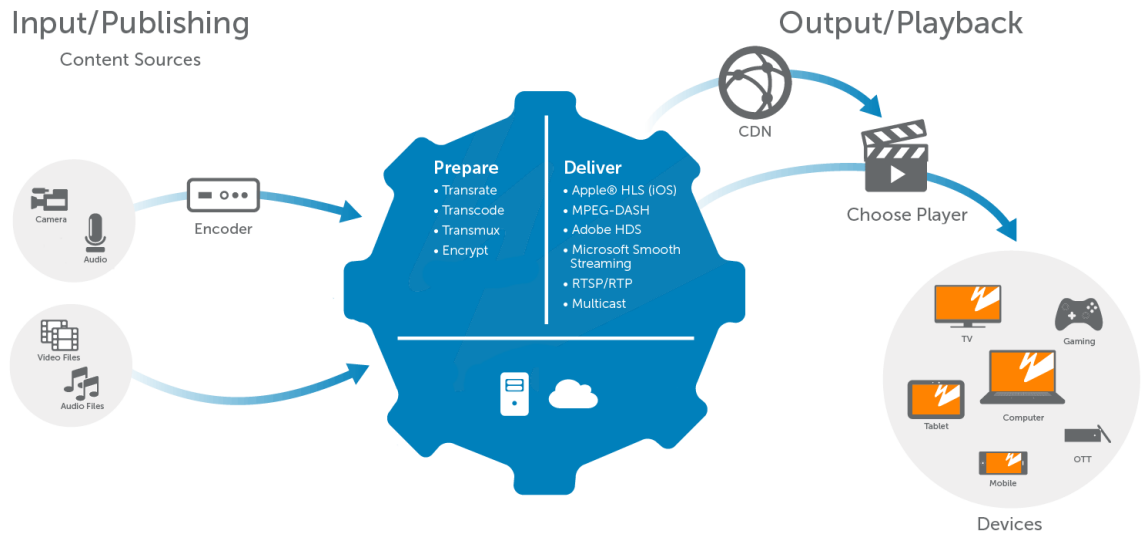
### 4.1.2 RTMP

RTMP, eli Real-Time Messaging Protocol, kehitettiin Flash-tekniikoiden väliseen tiedonsiirtoon. Nykyään RTMP toimii avoimena spesifikaationa, jolla siirtää Flash-yhteensopivaa videota, ääntä ja dataa. RTMP tukee videotiedostoissa MP4 ja FLV ja audiossa AAC ja MP3. Tekniikan suurimpana etuna voidaan pitää sen pientä viivettä, joka mahdollistaa suorat lähetykset..

Nykyään RTMP:tä käytetään usein siirtotekniikkana, kun suora lähetyksen koodataan CDN:lle jakeluun. Tämä onnistuu helposti esim. ilmaisella Flash Live Media Encoderilla. CDN transkoodaa saadun RTMP-lähetyksen ja lähettää käyttäen HLS, MPEG-DASH tai vastaavaa tekniikkaa. (Adobe 2016.)

## 4.2 MPEG-DASH ja suoratoiston workflow

Suoratoiston prosessi kamerasta loppukäyttäjälle voidaan jakaa kolmeen vaiheeseen. Ensin tarvitaan videokuva ja ääni kamerasta tai enkooderista, joka lähetetään serverille. Serveri paketoi ja striimaa pakatun median valitun tekniikan yli. Lopulta lähetys toistetaan selaimessa valitulla soittimella. Suoratoiston vaiheet on esitetty kuvassa 2.



Kuva 2. Suoratoiston vaiheet (Wowza 2011.)

### 4.2.1 Tuotanto

Suoratoistossa useat eri lähteet kelpaavat median tuottamiseen. Suorassa lähetyksessä tavallisinta on tuoda kuva kamerasta ja ääni mikseristä enkooderin kautta välityspalvelimelle. Videomikserillä lähetykseen voidaan tuoda myös valmiita videotiedostoja tai ääniraitaa. Välityspalvelimelle voidaan lähettää useilla eri tekniikoilla, joista yleisimmät ovat RTSP/RTP, MPEG-TS ja RTMP.

### 4.2.2 Välitys

Tavallisesti erillinen CDN-palvelu tarjoaa median välityksen loppukäyttäjille. Vastaanotettu RTMP-lähetys voidaan esimerkiksi muuntaa useaan muotoon, tarjoten mahdollisimman laajan laitetuken. Tulevaisuudessa todennäköisesti HLS iOS-laitteille, sekä MPEG-DASH kaikille muille. CDN-alan suuret toimijat, kuten Akamai, omistavat laajan ja suorituskykyisen jakeluverkon, joka pystyy tarjoamaan palvelua miljoonille käyttäjille samanaikaisesti.

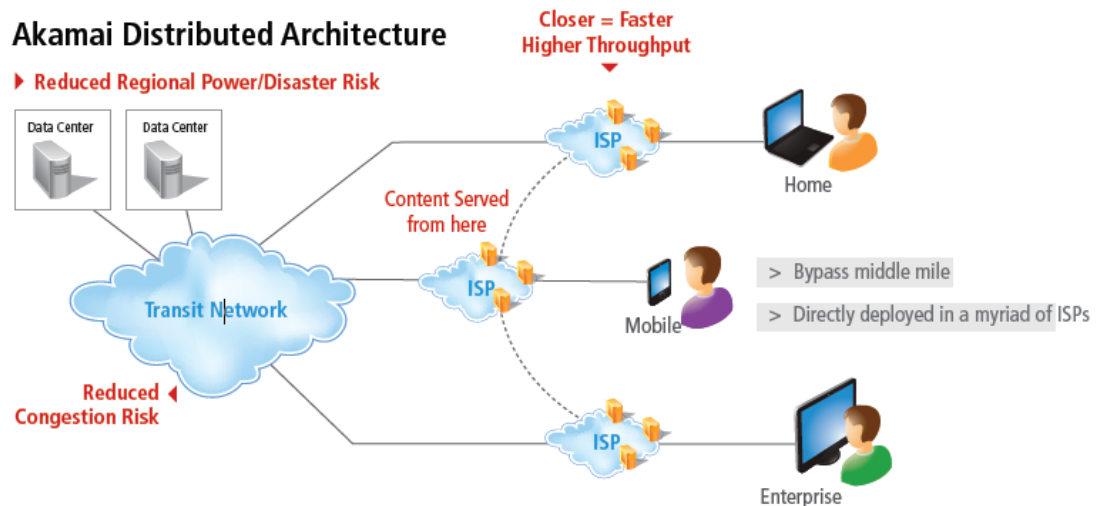
### 4.2.3 Toisto

Varsinainen median toistaminen voidaan suorittaa käyttäjän haluamalla tavalla, eikä soittimelle ole yleensä erityisiä vaatimuksia. Suoratoistolähetys voidaan avata suoraan esim. VLC Playerillä. Selaimessa käytetään usein live-lähetysten katsomiseen Flash-pohjaista soitinta, kuten JW Player (Wowza 2011). HTML5 toi selaimiin natiivivideosoitimen video-tagin muodossa, joka pitkälti syrjäyttää ulkoisten Flash-soitinten tarpeen selaimissa.

### 4.3 Jakelu

Otavamedia OMA Oy, kuten suurin osa muistakin internetin sisällöntuottajista, käyttää median jakeluun Akamain jakeluverkkoa. Akamai on pilvipalveluihin keskittynyt yhdysvaltalaisyhtiö, joka välittää 15-30 prosenttia kaikesta internetin liikenteestä (Tharakan 2015). Tyypillinen Akamain tarjoama palvelu on asiakkaalta vastaanotetun suoran RTMP-streamin transkoodaminen ja sen lähetys katselijoille halutussa muodossa.

Akamaille on tällä hetkellä markkinoiden laajin jakeluverkko ja sillä on eniten paikallisia palvelimia, noin 215000 (Akamai 2016). Palvelinten sijoittaminen lähelle loppukäyttäjiä parantaa luotettavuutta ja nopeutta. Last mile muodostuu mahdollisimman lyhyeksi ja internet-operaattorin palvelinlista löytyy Akamain oma palvelin. (Akamai 2014.) Akamain palvelinten sijoittelu on havainnollistettu kuvassa 3.



Kuva 3. Akamain palvelinrakenne (Akamai 2014.)

## 5 KOODEKIT

Videon pakkaukseen käytettävää algoritmia kutsutaan koodeksiksi. Termi codec muodostuu sanoista coder/decoder. Useimmiten koodekkeja käytetään äänen ja videon pakkaamiseen. Esimerkkeinä H.264, VP8, AAC, OGG yms. Näistä H.264 ja VP8 ovat videon pakkausmuotoja. AAC on H.264:n äänikoodekki ja OGG VP8:n.

MPEG-DASHille ei ole määritelty käytettäviä videokodekkeja, vaan periaatteessa mikä tahansa kelpaa. Käytännössä kuitenkin käytössä on H.264 lähes kaikissa tapauksissa. Tulevaisuudessa HEVC/H.265 tulee olemaan pääasiallinen enkoodausmenetelmä. Myös Googlen tarjoama VP9 tulee olemaan kilpaileva vaihtoehto näille tekniikoille.

Pakattu video on suoratoistotekniikoissa lähes pakollinen, sillä ilman tehokasta enkoodausta video vaatisi huomattavasti enemmän kaistaa lähetykseen. Nykyiset koodit pakkaavat videota käyttäjän tarpeen mukaan ja niillä voidaan saavuttaa esimerkiksi erittäin pienellä kaistatarpeella koko ruudulta katseltava video kohtuullisella kuvanlaadulla.

Käytännössä kaikki videonpakkaustekniikat noudattavat samoja periaatteita. Pakkaus tapahtuu etsimällä peräkkäisistä frameista samankaltaisuuksia. Siirrettävän datan tarve pienenee, kun ei tarvitse aina ilmaista koko framen tietoa uudelleen. Sen sijaan voidaan ilmoittaa vain tiettyjen kuva-alueiden liike kuvassa ja jättää esim. minimaalinen kirkkauden muutos huomiotta. Lopputuloksena ihmissilmä ei erota eroa laadussa.

Ihmissilmän heikkoutta hyödyntää erityisesti chroma subsampling. Tekniikassa enkoodataan kuvat näytteistämällä enemmän kirkkauden vaihtelua kuin värimuutoksia koska ihmisen silmä havainnoi helpommin kirkkauden muutokset.

Näytteistykseen laatu ilmaistaan yleensä muodossa J:a:b. Luku J ilmaisee vaakatasonäytteiden määrän, käytännössä aina 4. Luku a ilmaisee krominanssinäytteiden määrän J:n ilmaisemassa pikselirivissä. Viimeinen numero ilmaisee värierojen määrän kahden ensimmäisen rivin välillä.

4:4:4 on täydellä näytteistyksellä, eli chroma subsamplingia ei ole tehty.

4:2:2 on useiden ammattivideokameroiden käyttämä näytteistys, jossa kaksi chromanäytettä näytteistetään puolella luminanssinäytteiden määrästä. Esim. XDCAM HD422. Käytännössä lopputuloksessa ei voi huomata eroa. (Red 2016.) Yleisimmät näytteistystavat on esitelty kuvassa 4.

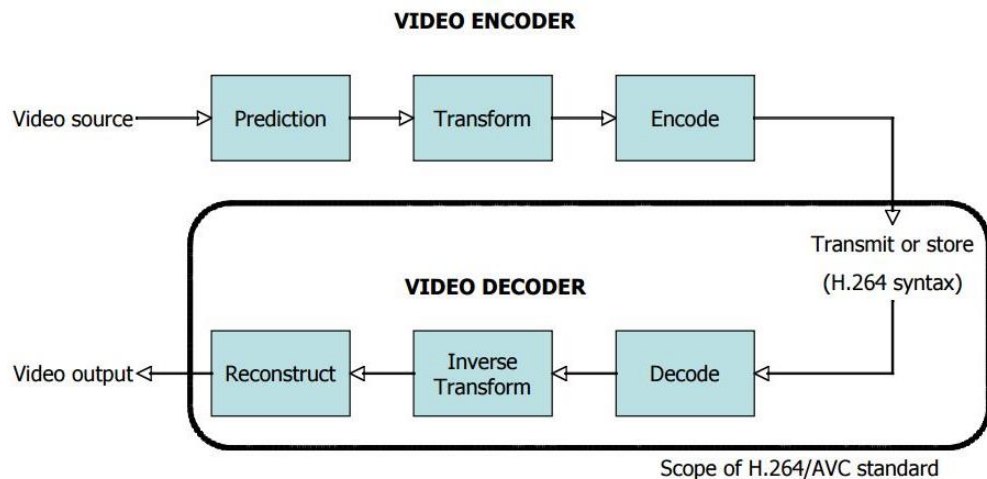


Kuva 4. Näytteistystavat (Wikipedia)

## 5.1 H.264

H.264, eli MPEG-4 AVC, on yleisimmin käytetty videon pakkaamiseen luotu standardi. Se on seuraaja MPEG-2- ja MPEG-4 Visual –standardeille ja parantaa näiden tehosuhdetta pakkauksen ja kuvanlaadun välillä. H.264 käytetään monen tyyppisessä videokäytössä nykypäivänä: suurin osa suoratoistettavasta videosta, HDTV ja Blu-Ray. Koodekkia käytetään myös harrastaja/puoliammattilaistason DSLR-kameroissa. H.264 sisältää enkooderin ja dekooderin. Enkooderi pakkaa videon ja dekooderi muuntaa pakatun videon takaisin pakkaamattomaan, eli katseltavaan, muotoon. (Ozer 2011.) H.264:n audioformaattina toimii AAC, eli MPEG-4 Part 3.

H.264:n standardia ylläpitää ITU-T(International Telecommunication Union), sekä ISO/IEC. Standardi määrittelee yleisesti dekodauksen prosessin, mutta enkoodauksen toteutus jää laitevalmistajan päätettäväksi. Käytännössä kuitenkin prosessi noudattaa useimmiten käänteistä dekodauksta (kuva 5).



Kuva 5. H.264 enkooderin ja dekooderin toiminta (Richardson 2007.)

## 5.1.1 Profiilit ja tasot

Vaikka H.264:n koodekit tulevat eri valmistajilta, on enkoodaukselle määritetty yleiset tekniikat ja suuntaviivat. Koodekki jakautuu profiileihin ja tasoihin (Levels). Profiilit ominaisuudet on lueteltu taulukossa 1. Yksinkertaisimmillaan profiilit määrittävät käytetyt tekniikat enkoodauksessa, kuten B-kuvien käytön, ja tasot määrittelevät yksityiskohdat, kuten maksimiresoluution ja bittinopeuden. (Ozer 2009.)

	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4 Predictive
I and P Slices	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B Slices	No	Yes	Yes	Yes	Yes	Yes	Yes
Multiple Reference Frames	Yes	Yes	Yes	Yes	Yes	Yes	Yes
In-Loop Deblocking Filter	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CAVLC Entropy Coding	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CABAC Entropy Coding	No	No	Yes	Yes	Yes	Yes	Yes
Interlaced Coding (PicAFF, MBAFF)	No	Yes	Yes	Yes	Yes	Yes	Yes
8x8 vs. 4x4 Transform Adaptivity	No	No	No	Yes	Yes	Yes	Yes
Quantization Scaling Matrices	No	No	No	Yes	Yes	Yes	Yes
Separate Cb and Cr OP control	No	No	No	Yes	Yes	Yes	Yes
Separate Color Plane Coding	No	No	No	No	No	No	Yes
Predictive Lossless Coding	No	No	No	No	No	No	Yes
	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4 Predictive

Taulukko 1. H.264:n profiilit (Ozer 2009.)

MPEG-DASHin kanssa voidaan periaatteessa käyttää mitä tahansa profiilia, mutta optimaaliseen laatuun ja toimivuuteen DASH-IF suosittelee käyttämään H.264:n Main- ja High-profiileja. (DASH IF 2015.)

### 5.1.2 Entropiakoodaus

Main- ja High-profiileissa videodatan pakkaamiseen on käytettävissä kaksi häviötöntä entropiakoodaukseen perustuvaa enkoodaustekniikkaa. Profiilista riippuen käytettävissä on joko CAVLC tai CABAC. Molemmat näistä ovat adaptiivisia enkoodaustekniikoita ja CABAC vaatii koneelta enemmän suorituskykyä. Samalla lopputuloksena on tehokkaampi pakkaus paremmalla laadulla. (Hermans 2012.)

Lopputuloksen laatu riippuu vahvasti myös lähteen laadusta, mutta yleisesti voidaan todeta CABACin olevan noin 5-15 prosenttia tehokkaampi kuin CAVLC. Erot ovat ihmissilmän havaittavissa etenkin HD-laadun videossa käyttäen erittäin matalia datamääriä. Esimerkkinä kuvassa 6 on enkoodattu 720p-video 800 kbps:

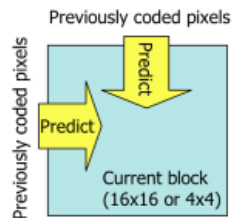


Kuva 6. 720p videotiedosto eri entropiakoodausmenettelyillä. (Ozer. 2009.)



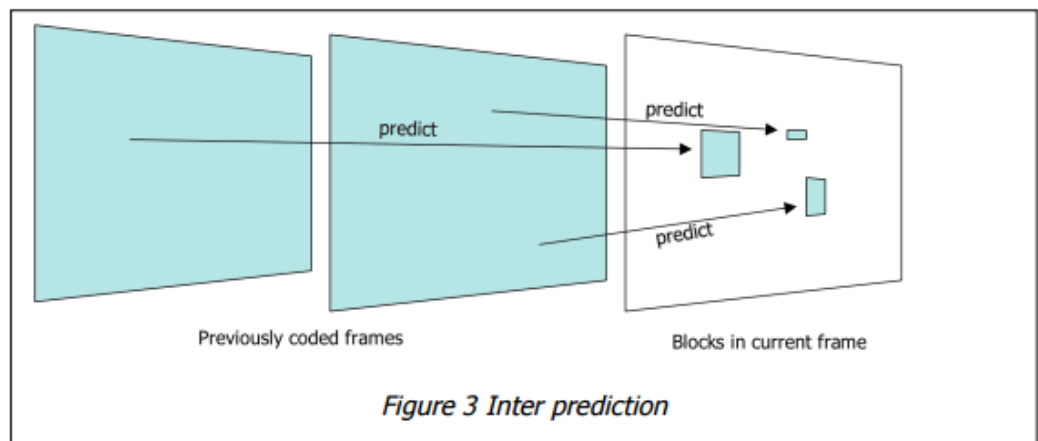
## 5.1.3 Enkoodaus

H.264-enkooderi käsittelee yhtä videon framea joukkona makroblokkeja. Enkooderi muodostaa makroblokista ennusteen joko nykyisestä kuvasta (intra), tai jo edellä muodostetuista ennusteista (inter). Intra-ennuste käyttää 16x16 tai 4x4 blokkeja (kuva 7).



Kuva 7. Intra-ennustaminen (Richardson 2011.)

Inter-ennustus käyttää vaihtelevia kokoja näiden välillä ja ennustaa pikselit samankaltaisilta alueilta edellisestä kuvasta (kuva 8).



Kuva 8. Inter-ennustaminen (Richardson 2011.)

Makroblokin pikselien arvot koodataan käyttämällä diskreettiä kosinimuunnosta (DCT). Esimerkkinä 8x8 pikselin kokoinen mustavalkokuva:



Jokaisen pikselin kirkkautta voidaan ilmaista lukuarvona 0 ja 255 välillä, eli seuraavasti:

120	108	90	75	69	73	82	89
127	115	97	81	75	79	88	95
134	122	105	89	83	87	96	103
137	125	107	92	86	90	99	106
131	119	101	86	80	83	93	100
117	105	87	72	65	59	78	85
100	88	70	55	49	53	62	69
89	77	59	44	38	42	51	58

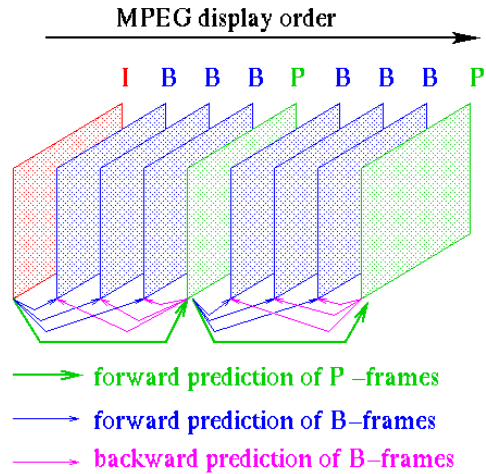
Normaalisti tieto pitäisi lähettää lukujonona oikealta vasemmalle ja ylhäältä alas. Tämä vaatisi 64 bittiä tallennustilaa. DCT:n jälkeen saadaan seuraavanlaiset arvot:

700	90	100	0	0	0	0
90	0	0	0	0	0	0
-89	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Kuvan tieto voidaan siis välittää huomattavasti pienemmällä bittimäärällä, kun tarvitsee ilmoittaa vain ensimmäiset luvut ja ilmoittaa, että loput ovat arvoltaan nolla. Dekooderi suorittaa käänteisen kosinimuunnoksen ja käytännössä saadaan tulokseksi sama kuva. Periaatteessa tällaisen pakkaustavan pitäisi olla häviötön, mutta käytännössä prosessi on painotettu keskiarvojen perusteella, joten tuloksena ei ole täsmälleen sama kuva. Ihmissilmä ei useimmiten pysty eroa kuitenkaan huomaamaan.

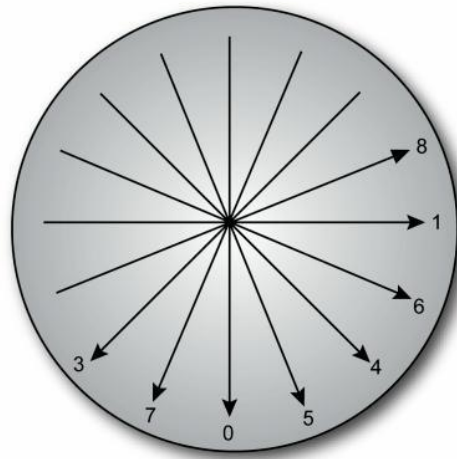
Enkoodausmenetelmästä huolimatta H.264:lle on tyypillistä, että kukin blokki tai frame voidaan ilmaista joko edeltävien tai seuraavien avulla. Inter-kuvien yhteydessä käytetään termiä liikearviointi, eli motion estimation. Jo pelkästään liikearvioinnilla saavutetaan suurin osa pakkauksen määrästä.

H.264:n keinot ennustamiseen ovat joustavampia kuin edellisten standardien ja se voi käyttää mitä tahansa 32 kuvan GOP:sta lähteenä sen referenssikuvulle. Intra-ennuste käyttää 16x16 sekä 4x4 kokoisia blokkikokoja ennustamiseen, ja inter-ennustus käyttää useita kokoja samalta väliltä. GOP-kuvasarjan rakenne ja järjestys on havainnollistettu kuvassa 9.



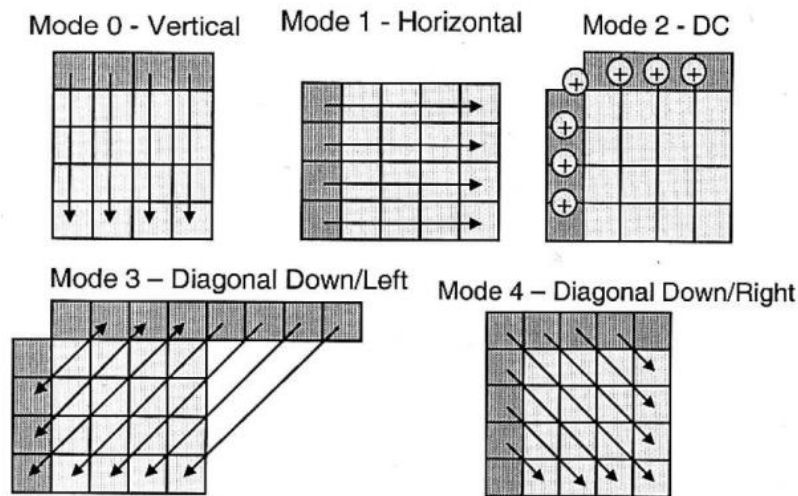
Kuva 9. Kuvasarjan rakenne (Anders)

H.264-kuvasarja muodostuu I-, P- ja B kuvista. I-frames ovat intra-koodattuja, eli ne voidaan muodostaa ilman viittausta muihin kuviin. Intra-kuvien ennustamisessa käytetään pohjatietona reunustavien pikselien keskiarvoa. Ennustelle on yhteensä 9 eri suuntaa, eli moodia, joita voidaan käyttää (kuva 10).



Kuva 10. Ennustevektorit (Vanam 2007.)

Näistä jokainen numero vastaa yhtä ennustemoodia, esim. kuten kuvassa 11:



Kuva 11. Ennustemoodit (Vanam 2007.)

P-frames ovat muodostettu edellisestä I- tai P-kuvasta. P-frame on siis mahdollista muodostaa ilman edellisen kuvan dataa. B-kuvat muodostetaan vertaamalla suuntaan tahansa, eli edellä/jälkeen oleva I- tai P-kuva. P- ja B-kuvat kuuluvat inter-ennusteen alle.

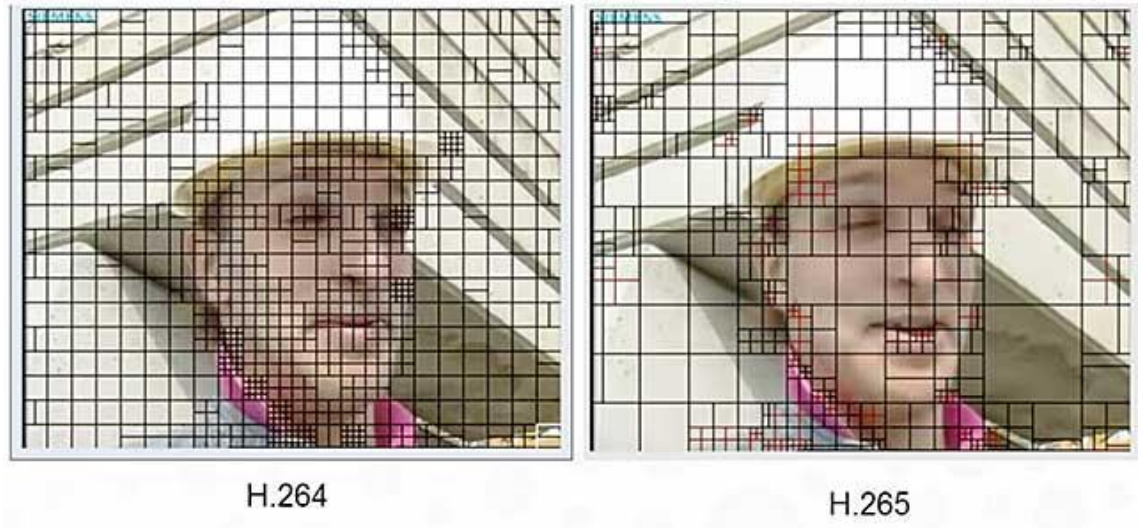
Koska B-kuva vaatii molemmista suunnista ennustamisen, tarvitsee dekooderille lähettää kuvat eri järjestyksessä kuin ne katsotaan. Jokaisessa frame:ssä tulee siis olla mukana järjestysnumero, jotta ne voidaan dekodauksen jälkeen asettaa katseltavaan muotoon. (Anders.)

## 5.2 H.265

H.265/High Efficiency Video Coding (HEVC) on H.264:n seuraaja. Kuten edeltäjänsä, sitä kehittää ISO/IEC MPEG ja ITU-T Video Coding Experts Group. MPEG-tekniikoille tyypillisesti tavoitteena on parantaa pakkauksen tehosuhdetta 50 prosenttia edelliseen tekniikkaan verrattuna.

HEVC käyttää edellisten tekniikoiden tapaan I-, B- ja P-kuvia ja käyttää sekä inter- että intraframe-pakkausta. H.265 sisältää kuitenkin lukuisia etuja aiempaan verrattuna.

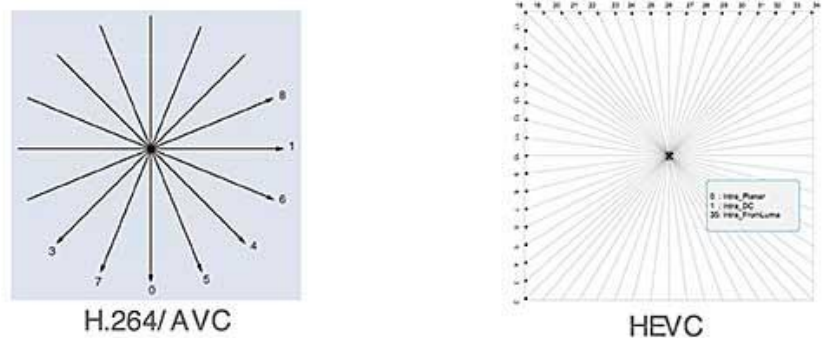
Kiinteiden 16x16px makroblokkien sijaan HEVC käyttää ns. coding tree blokkeja (CTB), joiden maksimikoko on 64x64 pikseliä. Isommat blokkikoot tuovat pakkaustehoa etenkin isojen framekokojen kanssa, kuten 4K-videossa. Etenkin suuret ja yksinkertaiset alueet on huomattavasti tehokkaampaa ilmaista suuren makroblokin avulla (kuva 12).



Kuva 12. H.264 ja H.265 makroblokkien erot (Ozer, J. 2013.)

HEVC taipuu myös aiempaa joustavampaan intra-ennusteeseen. H.264 pystyi ennustamaan liikettä yhdeksään eri suuntaan, kun taas HEVC vastaava ennustemäärä on 35 (kuva 13). Luonnollisesti suurempi ennustemäärä vaatii myös enemmän aikaa enkoodaukselta. (Ozer 2013.)

- **Luma: 35 prediction directions (33 + Planar + DC)**



Kuva 13. H.265:n laajennettu ennustaminen (Ozer, J. 2013.)

HEVC osaa hyödyntää moniydinprosessoria jakamalla kuvan useaan yhtä monta CTU:ta sisältävään osaan. Tämä mahdollistaa samalla myös sen, että esim. 4K-kuvasta voi katsella zoomattua aluetta ilman muiden alueiden laskeamista. Käytännössä tätä voisi hyödyntää 360-toteutuksissa, joille on jatkuvasti kasvava kysyntä. Kokonaisuudessaan HEVC on toiminnaltaan huomattavasti monimutkaisempi kuin H.264 ja enkoodaus vaatii koneelta runsaasti tehoa. (Crijns 2014.)

HEVC jakautuu kahteen profiiliin: Main ja Main 10. Molemmat profiilit tukevat 4:2:0 näytteistystä, mutta Main 10 tukee lisäksi 10-bitin värisyvyyttä. Tällä värisyvyydellä saavutetaan ITU-T:n määrittelemän UHD-standardin Rec. 2020 vaatimat väriarvot. H.264 vaatii huonosti tuetun laajennuksen 10-bitin väreihin, mutta HEVC:n pitäisi tukea sitä oletuksena.

HEVC:n yleistymistä jarruttavat monet asiat. Tällä hetkellä ei ole esim. laajalti saatavilla olevaa ilmaista soitinta. Adobe on ilmoittanut tukevansa HEVC-toistoa omassa maksullisessa Primetime-ohjelmistossaan, mutta ilmaiseen Flash Playeriin ei ole luvattu tukea. DivX-player ja VLC tukevat laajennusten myötä tekniikkaa, mutta käyttäjämäärät eivät ole lähelläkään laajamittaisen käytön mahdollistamista. (Ozer 2014.)

MPEG-DASHin kannalta ei ole mitään estettä HEVC:n yleistymiselle. Standardi on tällä hetkellä painottunut vahvasti H.264:n käyttöön, mutta on muistettava DASHin olevan täysin koodekkiriippumaton. Tulevaisuudessa HEVC-lähetykset onnistuvat DASHin alla mainiosti, kunhan vastaanotettava laite tukee sen toistamista.

### 5.3 VP8 ja VP9

WebM on Googlen HTML5-pohjainen vastine kaupalliselle H.264 ja h.265-koodekeille. Se perustuu avoimeen lähdekoodiin ja on täysin ilmainen. Videokoodekkina toimivat VP8 ja tuleva VP9 ja äänen pakkausmuotona on Ogg Vorbis. Laadullisesti etenkin VP9 pääsee varsin lähelle HEVC-pakkausta, mutta etenkin rajussa pakkauksessa ja pienellä bitratella HEVC vie voiton (Bultje 2015). WebM on erityisen hyödyllinen toimijoille, jotka eivät pysty maksamaan lisenssimaksuja, joita saattaa tulla MPEG:n tekniikoille. Tässä työssä keskityttiin kuitenkin H.264 ja HEVC:n käyttöön, sillä tulevaisuuden käyttökin tulisi nojautumaan edelleen vahvasti niihin pakkaustehonsa vuoksi.

## 6 DASH - DYNAMIC ADAPTIVE STREAMING OVER HTTP

DASH, eli Dynamic Adaptive Streaming over HTTP, on suoratoistotekniikka, joka yhdistelee dynaamisen ja adaptiivisen suoratoiston ominaisuuksia. DASH eroaa edeltäjistään siinä, että se on täysin riippumaton ja irrallaan kaupallisista toimijoista, toisin kuin Apple HLS, Adobe HDS ja Microsoft Smooth Streaming. Edellä mainitut tekniikat käyttävät DASHin tavoin HTTP:tä välitysprotokollana, mutta jokainen käyttää erilaista manifestitiedostoa ja segmentin muotoa. Streamia vastaanottavan laitteen tulee siis tukea kunkin tekniikan client-protokollaa.

Suoratoistotekniikoiden sekava tilanne sai MPEG:n käynnistämään uuden standardin tutkimustyöt vuonna 2009 ja vuonna 2012 julkaistiin ensimmäinen MPEG-DASHin ISO-standardi. HTTP-striimauksen standardi helpottaa huomattavasti palvelinten ja laitteiden välistä kommunikointia, kun kaikki perustuu yhteen ja samaan perustekniikkaan. DASH-tekniikan kehitystä ja yleistymistä tukemaan perustettiin DASH Industry Forum. DASH IF muodostuu tekniikan kehitykseen osallistuneista toimijoista, kuten esim. Microsoft, Apple, Netflix, Qualcomm, Ericsson, Samsung ja Dolby. (Dash IF.)

MPEG-DASH sisältää useita hyödyllisiä ominaisuuksia, jotka tekevät siitä erityisen kiinnostavan kilpailijoihin verrattuna. Aikaisemmista tekniikoista poiketen DASH tukee huomattavasti laajemmin erilaisia client-kokoonpanoja, sillä se ei vaadi ulkoisten laajennusten asentamista. HTTP-siirtotekniikan ansiosta tiedonsiirto tapahtuu täysin jo olemassa olevan netti-infrastruktuurin yli ja tukee nykyisiä tekniikoita ja toimijoita sisältäen mm. CDN:t, välimuistitekniikat, palomuurit ja NAT:t. DASH on myös täysin koodekkiriippumaton, joten webcast-toimijat voivat itse valita haluamansa tekniikat soittimiin ja lähetyksiin. Suoratoistotekniikoiden ominaisuuksia on vertailtu taulukossa 2.

Feature	Adobe HDS	Apple HLS	Microsoft Smooth	MPEG-DASH
Deployment on Ordinary HTTP Servers		✓		✓
Official International Standard (e.g., ISO/IEC MPEG)				✓
Multiple Audio Channels (e.g., Languages, Comments, etc.)		✓	✓	✓
Flexible Content Protection with Common Encryption (DRM)	✓	✓	✓	✓
Closed Captions / Subtitles	✓	✓	✓	✓
Efficient Ad Insertion				✓
Fast Channel Switching	✓		✓	✓
Protocol Support's multiple CDNs in parallel				✓
HTML5 Support				✓
Support in HbbTV (version 1.5)				✓
HEVC Ready (UHD/4K)				✓
Agnostic to Video Codecs				✓
Agnostic to Audio Codecs				✓
ISO Base Media File Format Segments	✓		✓	✓
MPEG-2 TS Segments		✓		✓
Segment Format Extensions beyond MPEG				✓
Support for multiplexed (Audio + Video) Content	✓	✓		✓
Support for non-multiplexed (separate Audio, Video) Content		✓	✓	✓
Definition of Quality Metrics				✓
Client Logging & Reporting				✓
Client Failover				✓
Remove and add Quality Levels during Streaming				✓
Multiple Video Views				✓
Efficient Trick Modes				✓

Taulukko 2. Suoratoistotekniikoiden vertailu (Mueller 2015.)

DASHin parhaita puolia on sen varsin väljä standardointi, mutta se on samalla tekniikan yleistymisen suurimpia haasteita. Esimerkiksi koodekkiriippumattomuus on suuri etu uusien tekniikoiden ilmestyessä markkinoille, mutta enkooderien ja soitinten valmistajille tämä aiheuttaa vaikeuksia. Valmistajan pitää valita itse tuettu koodekki, tekstityksen muoto ja tuettu DRM-tekniikka laitteelleen. DASH IF julkaisikin valmistajien toimintaa helpottamaan yleiset ohjeet, joilla tämän hetken DASH-toisto olisi mahdollisimman sujuvaa.

Oleellisimmat ohjeen kohdat ovat:

- Videon enkoodaus tapahtuu AVC/H.264 Main ja High-profiileilla. Tämän ohella voidaan tarjota myös muita koodekkeja vaihtoehtona.
- Audio-koodekki on HE-AAC v2.
- Segmentin muoto perustuu fragmentoituun ISOBMFF (MP4).
- MPEG-DASHin ondemand- ja live-profiilit ovat tuettuja.
- Segmentit ovat keyframe-painotteisesti jaettu tasaisesti Representationin alueelle ja ne ovat kestoiltaan suunnilleen samanmittaisia.
- Käytetty DRM-muoto on Common Encryption.
- Tekstityksen muoto on TTML-pohjainen SMPTE-TT-formaatti. (DASH IF 2015.)

Tekniikka on siis tällä hetkellä erittäin painottunut H.264:n käyttöön, mutta koodekki voisi olla mikä tahansa muukin. Samalla myös siirtotekniikan protokolla voi vaihdella. Tähän päädyttiin pitkälti siksi, että Apple voisi käyttää MPEG-2 Transport Streamia ja Adobe, Microsoft ja muut MP4 Elementary Streamia. (Siglin 2012.)

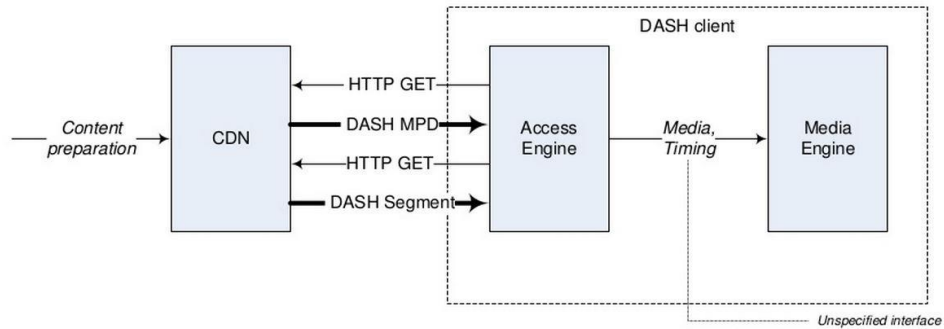
Dashin yleistymistä varjostaa MPEG LA:n laatima patenttiohjelma, jonka toteutuminen käytännössä tarkoittaisi lisenssimaksuja tekniikan käytölle. Tämä voisi pysäyttää selaintuen kasvamisen, etenkin voittoa tavoittelemattoman Mozilla Firefoxin kohdalla. DASHin kehityksessä mukana olleet Google, Microsoft, Cisco ja Qualcomm haluaisivat kaikki rojaltivapaan tekniikan, joten toivoa on. Pahimmillaan rojaltrimaksut voivat pysäyttää MPEG-DASHin yleistymisen täysin. (Ozer 2015.)

### 6.1 Enkoodaus ja rakenne

DASH-suoratoistossa media enkoodataan monta kertaa eri bit ratelle. Yksittäinen enkoodaus on nimeltään Representation, joka jaetaan edelleen useaksi Media Segmentiksi. Suoratoistossa mediaa toistava laite, eli Client, pyytää segmenttejä järjestyksessä yksittäiseltä Representationilta, johon sisältyy usein myös Initialisation Segment, joka toimii otsikkokenttänä (header). Se voi sisältää tietoa mm. enkoodauksesta ja kuvakoosta. (Perrott 2013.)

MPEG-DASH -serveri tarjoaa client-soittimille listan saatavilla olevista segmenteistä URL-osoitteina, jotka on lueteltu Media Presentation Description-tiedostossa (MPD). Client voi siis pyytää järjestyksessä palasen kerrallaan, jolloin lopputuloksena on katkeamaton ja yhteyden laatuun reagoiva median esitys (kuva 14). (Wowza 2016.)

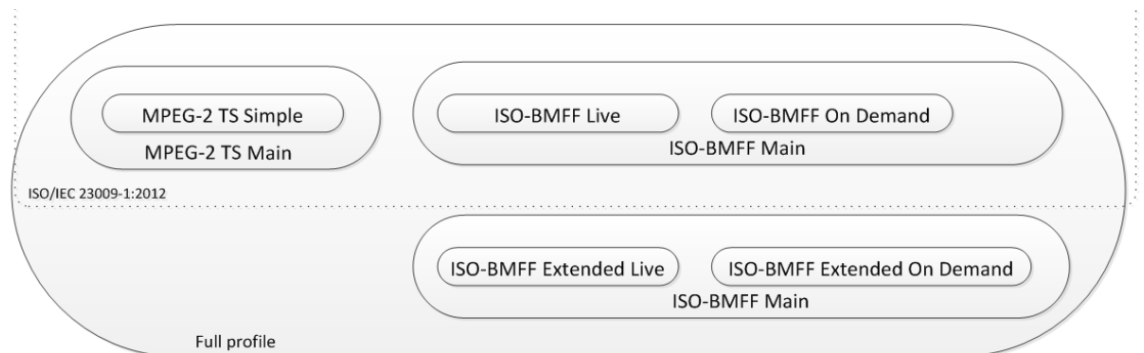




Kuva 14. MPEG-DASHin toimintamalli (Giladi 2011)

## 6.2 Profiilit

DASHin oleellisin profiili on ISO Base Media File Format Live (ISO-BMFF Live). Profiili on ominaisuuksiltaan kaikista monipuolisin ja tukee myös on-demand – toistoa. ISO Base-profiili ei ole kuitenkaan tarpeeksi tarkka, jotta voitaisiin perustaa suoraan clientit ja sovellukset suoratoistoa varten. Tähän tarkoitukseen perustettiin mm. Dash-IF, joka laati jo edellä mainitut ohjeistukset DASHin alkuvaiheen käyttöön. MPEG-DASHin profiilien rakenne on esitetty kuvassa 15.



Kuva 15. MPEG-DASH profiilit (Akamai)

Yksinkertaisempi On-demand profiili tarjoaa tuen on-demand -sisällön toistamiseen. Profiili vaatii jokaisen representationin esittämistä yhtenä segmenttinä niin, että alisegmentit on jaettu tasaisesti rep:ien kesken. Tämä sallii skaalautuvan ja tehokkaan HTTP-palvelimen käytön ja yksinkertaistaa laadun sujuvaa vaihtamista.

Live-profiili on optimoitu suoriin lähetyksiin ja sillä voidaan saavuttaa muutaman sekunnin viive. Segmentit koostuvat lyhyistä videofragmenteista ja jokaista niistä voidaan pyytää välittömästi niiden ilmestymisen jälkeen. Live-profiililla voidaan sujuvasti myös hoitaa on-demand -toiminnot

MPEG-DASH vaatii H.264-tiedostosta fragmentoitua muotoa, eli videoleike on jaettu osioihin, jotka voidaan sitten jakaa MPD:n kautta. On huomattava, että vaikka BMFF-spesifikaatio on aina tukenut fragmentointia, ei

sen käyttö ole ollut oleellista, kuin vasta adaptiivisen suoratoistotekniikan yleistyttyä. (Perrott 2015.)

### 6.3 MPD – Media Presentation Description

Kaikki HTTP-pohjaiset adaptiivisen suoratoiston tekniikat koostuvat kahdesta komponentista, eli varsinaisesta A/V-streamista, sekä manifestitiedostosta, joka sisältää streamien nimet ja osoitteet. DASHin A/V-stream on nimeltään Media Presentation ja manifestitiedosto Media Presentation Description. MPD on metadataa sisältävä XML-dokumentti, jolla DASH Client pystyy rakentamaan HTTP-URL:t, joilla pääsee käsiksi suoratoistoon vaadittaviin segmentteihin. (ISO/IEC 2014.) MPD sisältää tiedot media segmenteistä, niiden suhteista ja tarvittavat tiedot niiden välillä vaihtamiseksi.

DASHin manifest-tiedosto on yksinkertainen ja kompakti. MPD on myös fragmentoitu, joten se voidaan jakaa useaan osaan ja ladata useassa vaiheessa eri kohteista (kuva 16). Sama sisältö voi esimerkiksi olla saatavilla myös useasta URL:sta, esimerkiksi eri serveri tai CDN. Tämä tuo joustavuutta käyttäjälle ja sulavamman katselukokemuksen.

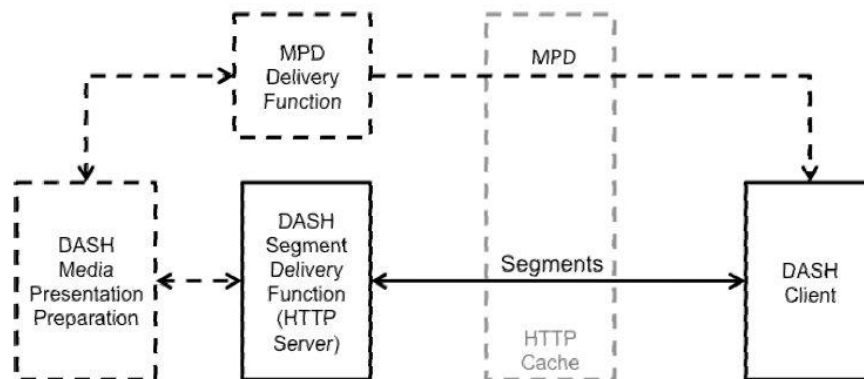


Figure 1 — Example system for DASH formats

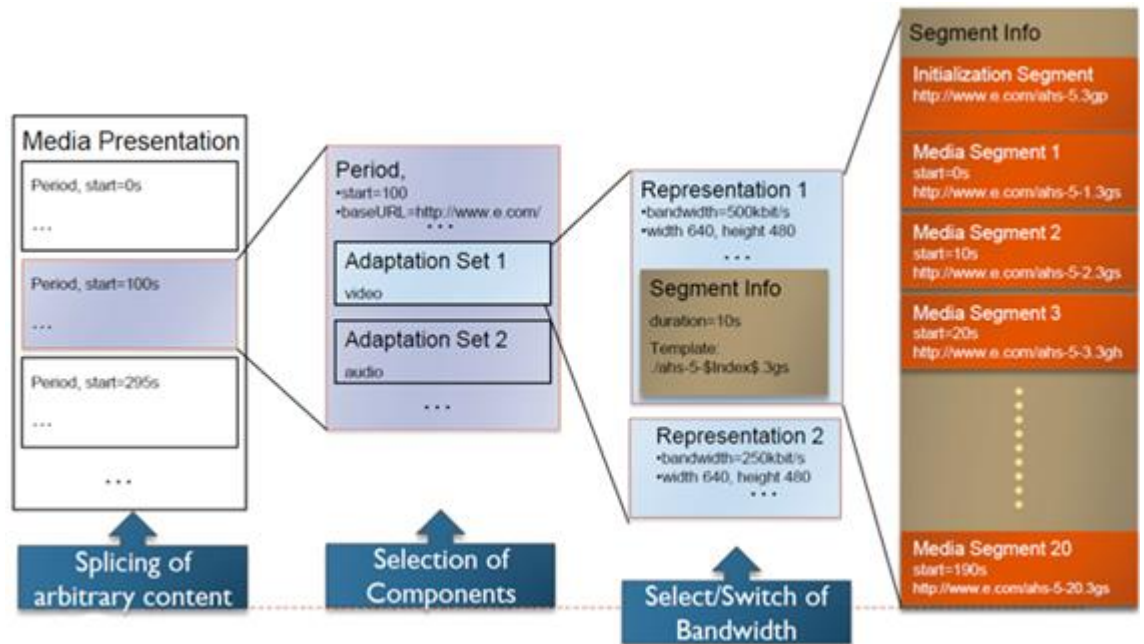
Kuva 16. DASHin ja MPD toiminta. (ISO-standardi)

MPD tarjoaa käyttäjälle riittävän informaation striimien valintaan ja vaihtamiseen. Saman lähetyksen sisällä voidaan esimerkiksi valita erikielinen audio, kamerakulma, tekstitys halutulla kielellä ja eri bitratet samasta kamerakulmasta. DASH tukee myös monipuolisesti DRM-tekniikoita ja mainosten upottamista sekä on-demand, että live-lähetyksiin (Malkiel 2015). Mainosformaateista tuettuja ovat ainakin VAST 3.0 ja VPAID 2.0. (Tavakoli 2015.)

## 6.4 MPD:n rakenne

MPD jakautuu neljään eri tasoon (kuva 17):

1. Periods
2. Adaptation sets
3. Representations → SubRepresentations
4. Segments → Media Segments & Index Segments



Kuva 17. MPD:n rakenne (Streamingmedia.com 2011)

## 6.4.1 Periods

MPD määrittää videosekvenssin yhdellä tai useammalla peräkkäisellä periodilla, jotka pilkkovat sisällön alusta loppuun (Ozer 2011). Periods kuvaavat lähetyksen sisältöä ja kertovat aloitusajan ja keston. MPD:ssä voidaan käyttää myös useampaa periodia jaotellaan chapterit tai erottamaan mainokset varsinaisesta ohjelmasisällöstä. (Long 2015.)

## 6.4.2 Adaptation sets

Adaptation sets sisältää mediavirran tai joukon useita mediavirtoja. Lähettykseen riittää jopa yksi adaptation set, mutta usein jaotellaan video ja ääniraidat erikseen. Näin esim. eri kielisiä audioraitoja ei tarvitse kaikkia ladata kerralla. Adaptation set voi myös sisältää teksitykset tai metadatan. (Long 2015.)

### 6.4.3 Representations

Representations-kerros tarjoaa yhdelle adaptation setille saman sisällön enkoodattuna usealla eri tavalla. Tyypillisesti tarjotaan muutama eri kuvakoko ja bit rate. Media voidaan myös enkoodata usealla koodekilla, jotta saavutetaan mahdollisimman laaja tuki eri laitteille. Tyypillisesti representation valitaan automaattisesti, mutta käyttäjälle tarjotaan mahdollisuus vaihtaa resoluutiota niin halutessaan. (Long 2015.)

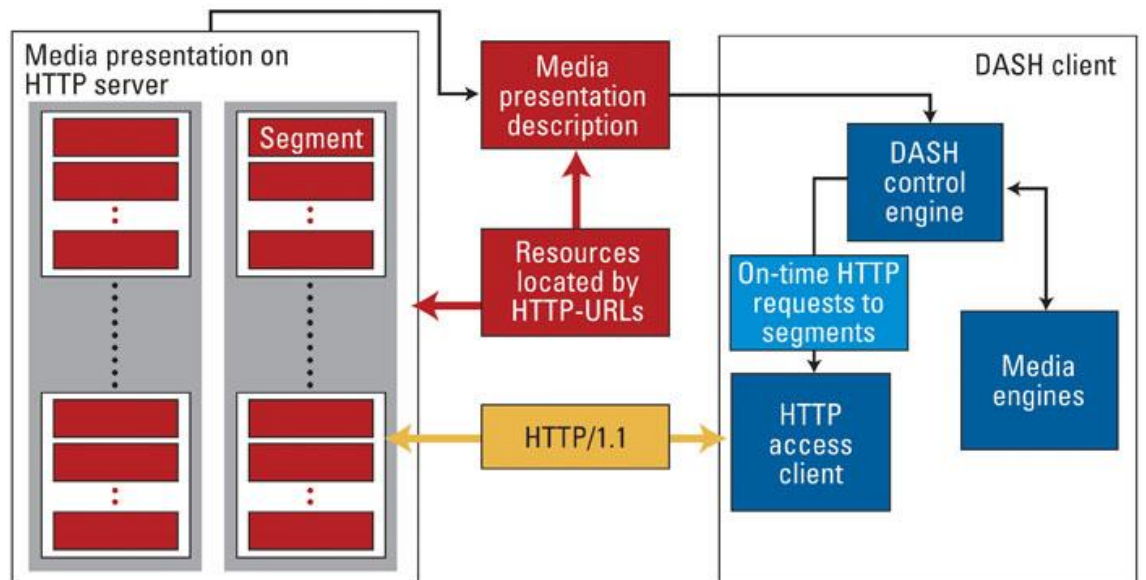
### 6.4.4 Media Segments

Mediasegmentit ovat varsinaista toistettavia mediatiedostoja. Niiden sijainti voidaan ilmaista URL-listana tai yhtenä SegmentTemplatena. Livelähettyksen kannalta oleellista on SegmentTimelist, josta client saa nopeasti haettua viimeisimmän segmentin. Viisi segmenttiä sisältävä Representation voidaan ilmaista esim. näin:

```
<Representation id="720p" bandwidth="3200000" width="1280" height="720">
  <BaseURL>720p/</BaseURL>
  <!-- Timescale is "ticks per second", so each segment is 1 minute long -->
  <SegmentList timescale="90000" duration="5400000">
    <RepresentationIndex sourceURL="representation-index.sidx"/>
    <SegmentURL media="segment-1.ts"/>
    <SegmentURL media="segment-2.ts"/>
    <SegmentURL media="segment-3.ts"/>
    <SegmentURL media="segment-4.ts"/>
    <SegmentURL media="segment-5.ts"/>
  </SegmentList>
</Representation>
```

Useimmiten live-lähetyksessä segmentit ovat erillisiä tiedostoja ja on-demandissa bittialueita yhdestä tiedostosta. (Long 2015.)

MPD-tiedosto voidaan toimittaa HTTP:n yli, sähköpostilla, broadcastina tai millä tahansa muulla tavalla. MPD:stä DASH-client saa tiedon mm. aika-koodeista, mediatyypeistä, resoluutioista, kaistanleveydestä ja vaadituista DRM-tekniikoista. Client valitsee sopivan enkoodatun vaihtoehdon ja aloittaa suoratoiston käyttäen HTTP:n GET-pyyntöä. (Encoding 2016.) MPD:n ansiosta ei tule päällekkäistä latausta bit raten vaihdon kohdalla ja materiaalit voivat olla erillään MPD:n sijainnista, eli eri palvelimella (kuva 18).



Red blocks are MPEG-DASH's scope

Kuva 18. MPEG-DASHin server-client -toiminta (TVTechnology 2012)

## 6.5 MPEG-DASH Live

Live-toteutus ei juurikaan eroa tavallisesta DASH-suoratoistosta, mutta muutamalla asetuksella katselukokemuksesta saa mahdollisimman sujuvan. AVC-koodatun videon sekvenssin pituuden tulisi olla noin kaksi sekuntia, sillä se määrittää myös videosegmentin pituuden. Kahden sekunnin videosegmentit ovat optimaalisia DASH-suoratoistoon. Lyhyt segmentin kesto mahdollistaa myös suhteellisen lyhyen latenssiajan videon saapuessa live-enkooderille ja toistavalle soittimelle.

Varsinainen latenssiaika määrittyy soittimen bufferointilogiikan ja nettiyh-teyden laadun mukaan. Tyypillinen viive vaihtelee muutamasta sekunnista puoleen minuuttiin. CDN-toimijan täytyy tarjota soittimelle MPD:t reaaliajassa, eikä niitä voida luoda etukäteen. Liian aikaisin pyydetty MPD aiheuttaa selaimessa HTTP 404-virheen ja toisto keskeytyy. Ilman reaaliaikaista MPD-tarjontaa täytyy luottaa soittimen kellodataan. Virheiden välttämiseksi vaaditaan tavallista pidempää viivettä toistoon, jotta suurikaan kellovirhe ei haittaisi.

MPEG-DASHin live-profiili sopii niin livetuohtantoon, kuin VOD-toistoonkin. Tavallinen On-Demand profiili on rajattu VOD-lähetykseen, eikä se tue dynaamista mainosten sijoittamista streamiin. Käytännössä CDN voi tarjota kaikki DASH-palvelunsa käyttäen Live-profiilia. (Hughes 2014.)

### 6.6 HBBTV

MPEG-DASHin yksi tulevaisuuden käyttötarkoituksista on HbbTV, eli Hybrid Broadcast Broadband TV. Standardi yhdistää tv-lähetyksen ja laajakaistan tuomalla TV-lähetyksiin lisäominaisuuksia, kuten laajennettua ohjelmakarttaa, uutispalveluita yms. Tekniikkaa voidaan kutsua myös ns. ”punaisen napin” toiminnallisuudeksi. (Digita 2016.)

Suomessa Digita on aloittanut pilottihankkeen mm. Ylen ja Nelosen kanssa, jossa Areenan ja Ruudun sisältöä tarjotaan HbbTV:n yli. Käytännössä ohjelmaoppaasta voi katsoa esim. leffatrailereilta ja aiemmin näytettyjä ohjelmia. Digitan HbbTV-lähetykset perustuvat versioon 1.5 ja MPEG-DASHin pohjalle. (DigitalTVEurope 2015.)

Standardin uusin versio, eli 2.0, tarjoaa parannuksia useilla osa-alueilla, kuten:

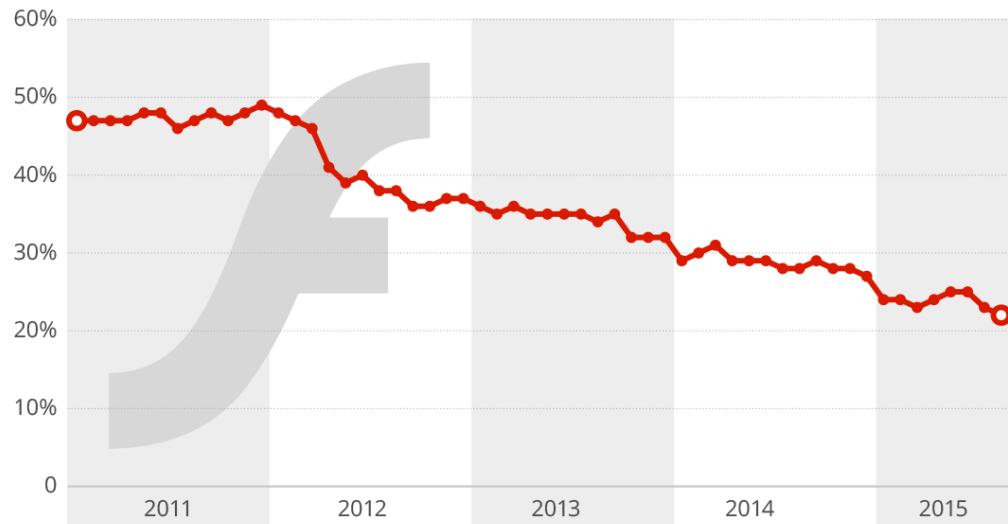
- Sujuvampi käyttökokemus laitteesta riippumatta
- Tuki UHD-laadulle HEVC:n myötä
- Median synkrointi päätelaitteiden välillä
- Tukee laajemmin mainostekniikoita
- Mobiililaitteesta voidaan käynnistää TV:n HbbTv-sisältöä

Uusin versio rantautuu todennäköisesti ensin Iso-Britanniaan, ja ensimmäiset tekniikkaa tukevat laitteet alkavat tulla markkinoille vuoden 2016 aikana. Tällä hetkellä Euroopassa myytävistä uusista televisiosta suurin osa tukee HbbTV-tekniikkaa. (HbbTV 2015.)

### 6.7 HTML5

HTML5 on HTML-kuvauskielen uusin versio. Uudistus on kuitenkin sen verran merkittävä, että HTML5 on muodostunut kattotermiksi useille eri tekniikoille. HTML5:stä puhuttaessa tarkoitetaan yleensä sovellusten ajamista kokonaan selaimessa ilman ulkoisia ohjelmistoja, kuten Flash Player. HTML5 nojaa vahvasti JavaScriptiin ja CSS-kuvauskieleen sovellusten toteutuksessa. Merkittävänä uudistuksena HTML5 toi myös canvas- ja videoelementit, jotka mahdollistavat videon toistamisen sekä live-lähetyksen katsomisen suoraan selaimen vakiosoitinkomponentilla. Soitinkomponentti riippuu selaimesta ja näin jokaisella selaimella on hieman eri videoformaattit tuettuna.

HTML5 on erityisen tarpeellinen mobiililaitteissa, joissa ulkoisten ohjelmien tuki on heikkoa ja asentaminen on vaivalloista. Flash-tuki alkaa pikkuhiljaa kadota etenkin mobiilipuolella (kuva 19), joten median toistaminen ilman erillisen toistimen asentamista on erittäin hyödyllistä.



Kuva 19. Vähintään yhden Flash-pyynnön lähettävien nettisivujen määrän lasku. Mukaan on laskettu myös mainokset. (Rosoff 2015.)

DASH-toiston kannalta oleellinen osa HTML5-tekniikkaa on sen tarjoama video-tag. Aiemmin tätä ei ollut tarjolla HTML-tekniikoissa. Yksinkertaisimmillaan video-tagia käytetään videomateriaalin sijoittamiseen natiivisti HTML-sivulle ilman ylimääräisiä komponentteja. Videolähde osoitetaan soittimelle <source>-elementillä. Videolähteitä voi olla useita, ja selain valitsee niistä laitteelle sopivimman.

Video-tagia voi konfiguroida useilla eri tavoilla. Normaalit HTML:n ns. global attributes-asetukset toimivat myös, kuten id ja hidden. Soittimelle oleellisia asetuksia ovat esimerkiksi:

Controls – Käyttäjälle tarjotaan videon toistamisen säädöt, kuten play ja äänenvoimakkuuden säätö.

Loop – Videon lopussa palataan takaisin alkuun ja aloitetaan toisto uudelleen

Preload – Voidaan kertoa selaimelle, että kannattaako videosta ladata jotain ennen toistoa. Esim. metadata tai koko video ilman että käyttäjä edes käynnistää toistoa.

Poster – URL-sijainti kuvalle, joka näkyy playerissä ennen videon toistoa.

Src – Videolähteen URL-osoite

Esimerkiksi tässä ladataan video usealla eri lähteellä. Video-tagille on määritetty leveys 480 pikseliä, controls ja poster-kuva:

```
<video width="480" controls poster=https://archive.org/download/webmvp8.gif>  
<source src="https://archive.org/download/webmvp8_512kb.mp4" type="video/mp4">  
<source src="https://archive.org/download/webmvp8.ogv" type="video/ogg">  
<source src="https://archive.org/download/webmvp8.webm" type="video/webm">  
Your browser doesn't support HTML5 video tag.  
</video>
```

(Mozilla 2016.)

### 6.8 MSE - Media Sources Extensions

Tällä hetkellä selaimessa DASHia toistava soitin vaatii selaimelta MSE-tukea. MSE, eli Media Source Extensions API, mahdollistaa selaimelle median suoratoiston perustuen puhtaasti JavaScriptiin. Käytännössä siis saadaan DASH-median suoratoisto HTML5-pohjalle. Laitteelta ei vaadita erilisiä asennuksia, kuten aikaisempi HLS:n vaatima Flash-tuki.

Nykyään on tarjolla myös hls.js, jolla pystytään tukemaan myös vanhaa HLS-tekniikkaa HTML5:n päällä. Se muuntaa MPEG-2 Transport Streamin ISO BMFF (MP4)-fragmenteiksi ja näin HTML5-yhteensopivaksi. Hls.js on integroitu jo tarjolla oleviin soittimiinkin, kuten Flowplayer ja Videojs (Github). Laaja MSE-tuki on vasta tulevaisuudessa todellisuutta, mutta jo tällä hetkellä IE11 ja Chrome kattavat n. puolet käytössä olevista selaimista, mahdollistaen uusimmat videotekniikat monelle käyttäjälle. (Protalinski 2015.)

## 7 PALVELUN TESTAUS

MPEG-DASH on vasta nuori ja kehittyvä tekniikka, joten toimivan ja kohtuuhintaisen DASH-ympäristön löytäminen tällä hetkellä on haastavaa. Kaupallisten toimijoiden tuotteissa on muutama tarjolla, mutta niiden kustannukset ovat melko korkeat. Akamai tarjosi syksyn aikana vain kahdelle tuotteelle DASH-tuen omassa ympäristössään. Ensimmäinen oli Elemental Liven pilvipalvelu, jonka kustannukset olivat liian suuret näin pieneen kokeiluun. Palvelu onkin tarkoitettu lähinnä suurille toimijoille TV-lähetysten parissa. Toinen tuettu ratkaisu oli Cisco AnyRes 9.5.1 -tekniikka, joka löytyy fyysisestä enkooderista (kuva 20).



Kuva 20. Cisco AnyRes Live 9500 UHD Encoder (Cisco)



Kumpikaan ratkaisu ei tarjonnut helpotusta tilanteeseen, jossa olisi tarkoitus hyödyntää olemassaolevaa RTMP-streamia tukevaa kalustoa, jolla lähetetään RTMP-stream CDN:n palvelimelle, josta se jaetaan haluttuun muotoon, kuten HLS ja MPEG-DASH. Akamai ei pystynyt tarjoamaan ollenkaan transkoodausta rtmp:stä DASH-muotoon, ainakaan lähiaikoina. Tarvittiin siis vaihtoehtoinen tapa tutkia DASHin toimivuutta webcast-tuotannossa, joten alettiin kartoittamaan ilmaisia vaihtoehtoja.

Lopulta päädyttiin toteuttamaan avoimeen lähdekoodiin perustuva järjestelmä, joka olisi alusta asti itse rakennettu. Näin päästiin tutkimaan DASHin suoratoistoa paremmin vaihe vaiheelta, ja saatiin parempi kuva koko prosessin toiminnasta. Kaikki osat olisivat omassa hallinnassa, eikä tuntematon stream-palvelin pilvipalvelussa. Lisäksi saatiin eliminoitua kaikki suorituskykyyn vaikuttavat ylimääräiset osat/sovellukset kun koko järjestelmä asennettiin tyhjään koneeseen kevyen käyttöjärjestelmän päälle.

Tulevaisuudessa ymmärrys tekniikasta auttaa webcast-keikoilla vikatilanteiden selvittämisessä esim. kaupallisten toimijoiden tuotteiden kanssa. Vika on helpompi paikallistaa, kun on kokemusta koko MPEG-DASH-lähetyksen prosessista. On siis kyettävä kertomaan asiakkaalle varsin monipuolisestikin MPEG-DASHin edut ja toiminnot muihin tekniikoihin verrattuna. Lopulta tarkoituksena on päästä täysin eroon Flash-pohjaisista ratkaisuksista ja siirtyä HTML5-pohjaiseen soittimeen ja webcast-näkymään.

Toimiva ratkaisu oli vaikea löytää. Yksittäisiä toimintoja hoitavia ohjelmia löytyi kyllä, mutta lopputuloksena olisi syntynyt useista erillisistä osista koostunut järjestelmä. Tarjolla olevien DASH-tekniikkaan liittyviä ilmaisohjelmien toiminnot vaihtelevat MP4-tiedoston fragmentoinnista enkoodaukseen. Usein kuitenkin tuli seinä vastaan, kun Live-profiilia ei tuettu ollenkaan, ainakaan ilmaisversiossa.

Toimivimmaksi ratkaisuksi löytyi Nginx-palvelin, jonka RTMP-moduuli hoitaisi kaiken tarvittavan. Selvitystyön jälkeen päädyttiin mahdollisimman yksinkertaiseen ja mukautuvaan järjestelmään, eli Ubuntu Serverin päälle asennettuun Nginx-serveriin. Nginx on perusversioltaan ilmainen HTTP-, kuormantasaus-, sähköposti ja yleinen TCP/UDP-palvelin. Huhtikuussa 2016 Nginx:n pohjalla toimi n. 26% netin suosituimmista sivuista. (Nginx.)

Varsinaisen alustana toimii Shuttle XPC-tietokone (kuva 21), joka tarjoaa erittäin kompaktissa muodossa riittävästi suorituskykyä ja ominaisuuksia. Koneeseen on asennettu kevyt Ubuntu Server 15.04 ilman graafista käyttöliittymää. Mediaserverinä toimii Nginx, joka tarjoaa todella monipuoliset ominaisuudet, eikä kuormita konetta juuri yhtään. Se tarjoaa myös projektin kannalta tärkeän rtmp-moduulin, jonka toiminnallisuus mahdollistaa rtmp-streamin kääntämisen ja lähettämisen DASH-streamiksi.



Kuva 21. Shuttle XPC-tietokone (Shuttle)

### 7.1 Nginx-konfiguraatio

Ensin täytyy määritellä Nginx:lle työskentelevien prosessien määrä. Tämä tapahtuu `worker_processes`-arvolla. Oletuksena suositellaan yhtä per prosessorin ydin. Useampi ei vahingoita järjestelmää, mutta voi tuottaa ylimääräisiä ja turhia toimintoja taustalle (Kavon 2014). Lisäksi on määriteltävä lukumäärä käyttäjille, joille Nginx tarjoaa palvelua. Tässä tapauksessa käytettiin oletusarvoa 1024. Konfiguraatiossa näkyy myös virhelokin nimeäminen:

```
worker_processes 1;

error_log logs/error.log debug;

events {
    worker_connections 1024;
}
```

Seuraavaksi määritellään kuinka Nginx käsittelee HTTP-liikennettä. Serveille osoitetaan `mime.types`-tiedosto, joka ilmaisee sisällön tiedostomuodot. Octet-stream on binääritiedosto, joka tyypillisesti tulee avata sovelluksella, kuten taulukko-ohjelmalla tai tekstinkäsittelyssä. Tiedostotyypistä riippumatta octet stream ei tyypillisesti ole avattavissa suoraan selaimessa. (Indiana university 2015.)

Näiden lisäksi määritellään aika timeoutille clientin suhteen:

```
http {
    include mime.types;
    default_type application/octet-stream;

    sendfile on;
    keepalive_timeout 65;
```

Varsinaisen serverin konfiguraatiossa määritellään serverin nimi, sekä tiedostopolut HTML-tiedostoille, DASH-medialle ja soitinkomponentille.

```
server {
    location /dash {
        root /tmp;
        add_header Cache-Control no-cache;
    }

    location /dash.js {
        root /home/webcast/www;
    }

    location / {
        root /home/webcast/www;
    }
}
```

RTMP-moduulille on omat asetukset. Tässä määritellään serverille vielä muutama lisäasetus, sekä käynnistetään halutut moduulin sovellukset. Tässä käytettiin vain dash-sovellusta live-tilassa, mutta samaan aikaan olisi voitu esim. tallentaa suoratoistolähetys tai luoda rinnakkainen HLS-lähetys. Serveri asetetaan kuuntelemaan porttia 1935, johon lähetetään RTMP-stream moduulin käännettäväksi.

```
rtmp {
    server {
        listen 1935;
        ping 30s;
        notify_method get;
```

RTMP-moduulin dash-sovellus asetetaan live-tilaan, eli sovellus luo lähetysten havaittuaan väliaikaisen mpd-tiedoston, jonka voi sitten selaimessa avata. MPD-tiedostolle ja dash-fragmenteille määritellään sijainti.

Lisäksi käsketään unohtamaan lähetys 10 sekunnin jälkeen kun RTMP-lähetys loppuu. Ilman tätä Nginx ei osaa ottaa vastaan uutta lähetystä ilman serverin uudelleenkäynnistämistä.

```
application dash {
    live on;
    drop_idle_publisher 10s;
    dash on;
    dash_path /tmp/dash;
}
}
```

Näiden asetusten jälkeen Nginx vaatii vain konfiguraation uudelleenlataamisen ja palvelu toimii ongelmitta.

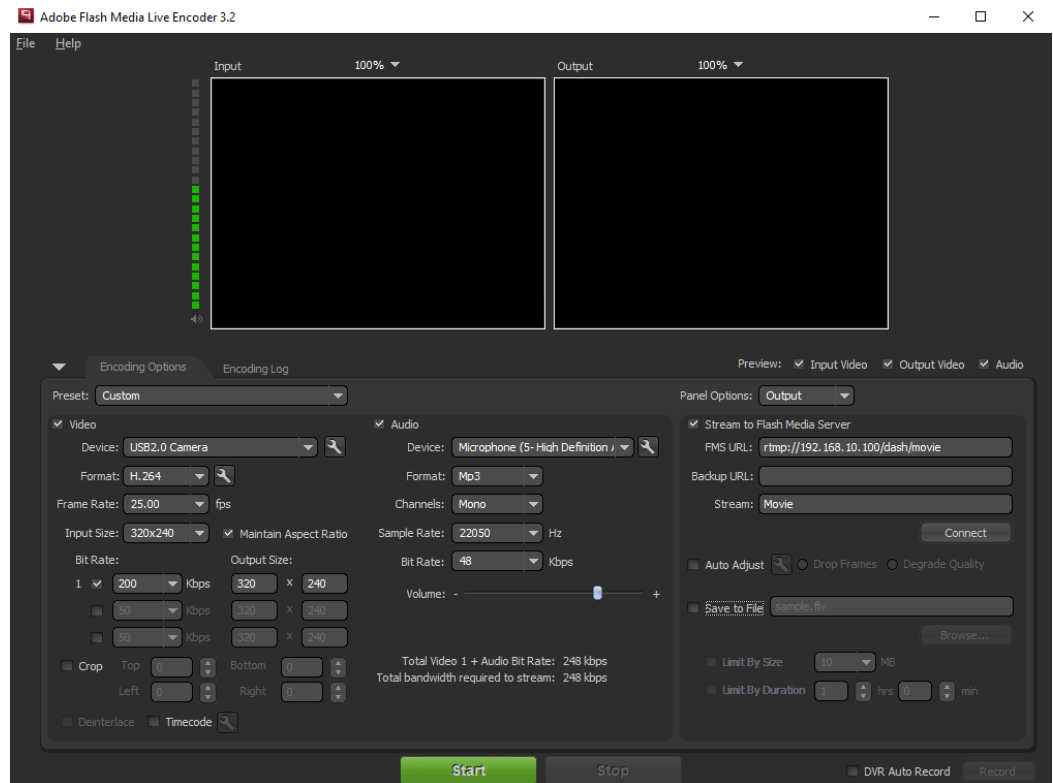
### 7.2 Käytettävät sovellukset

Enkooderina voidaan käyttää mitä tahansa ohjelmaa, kunhan Nginx:lle lähetetään RTMP-stream porttiin 1935. Ennen varsinaista testiä lähetin serverille RTMP-streamin enkoodaamalla valmiista videoleikkeestä FFmpegillä. FFmpeg on komentorivipohjainen kokoelma ohjelmistoja, joilla voidaan tehdä äänelle ja videolle lähes mitä tahansa leikkauksesta enkoodaukseen. Videoleikkeenä toimi Blender Corporationin lyhyt animaatiotraileri Sintel\_trailer. Käytetty FFmpeg-komento, jossa flash-videota lähetetään Nginx:n IP-osoitteeseen:

```
ffmpeg -re -i sintel_trailer-480p.mp4 -c:v libx264 -profile:v \
baseline -c:a aac -strict -2 -ar 44100 -ac 2 -f flv \
rtmp://192.168.10.100/myapp/movie
```

Noin kymmenen sekunnin jälkeen Nginx oli prosessoinut lähetetyn videon ja se oli katsottavissa DASH-muotoisena selaimessa.

Varsinainen labraympäristön testi tehtiin käyttäen Adobe'n ilmaista Flash Media Live Encoderia. Sony PMW-EX3 –kamerasta lähetettiin 720p-videokuva kaapparikortin kautta Live Encoderille, joka sitten lähetti sen RTMP-muotoisena Nginx:lle (kuva 22).



Kuva 22. Adobe Flash Media Live Encoder

### 7.2.1 DASH.js

Projektin varsinainen soitinkomponentti on Dash.js. Dash.js on Javascript-kirjasto, joka muuntaa DASH-streamin HTML5-videosoitimen MSE:n ymmärtämään muotoon. Muiden selaimessa toimivien soittimien tuki DASHin livelähetyksiin on vielä erittäin niukkaa. Muutama toimija tarjoaa ominaisuutta maksua vastaan, mutta tähän projektiin tarvittiin ilmainen ja avoimeen lähdekoodiin perustuva ratkaisu. DASH IF on perustanut tarkoitusta varten dash.js-projektin, jonka tarkoituksena on luoda avoimen lähdekoodin JavaScript-kirjasto DASHin toistamiseen.

Dash.js voidaan ladata ilman asennusta ja projektissa videota toistettiin yksinkertaisen nettisivun kautta, jossa oli pelkkä videoistin. Varsinainen soitinkomponentti upotettuna HTML5-sivulle on varsin yksinkertainen. Videon toistamiseen ei vaadita muuta kuin perus HTML5-sivupohja, jossa ladataan dash.js, sekä kerrotaan toistimelle MPD-tiedoston sijainti.

Videosoitin sijoitetaan divin sisään ja määritetään, että soittimen säädöt ovat näkyvillä käyttäjille:

```
<div>
  <video id="videoPlayer" controls="true"></video>
</div>
```

Varsinainen dash.js-soitin voidaan ladata joko netistä tai paikallisesti, kuten tässä on tehty. Suunnitellussa labraympäristössä ei ollut tarkoitus päästä internetiin, joten soitin oli ladattava suoraan palvelimen HTML-kansiosta:

```
<script src="/dash.js-live/dash.all.js"></script>
```

Lopuksi soittimelle täytyy vielä kertoa MPD:n sijainti. Tiedosto löytyy striimin aikana väliaikaisena tiedostona RTMP-moduulille määritellystä dash-kansiosta. MPD:n nimi riippuu lähetettävän striimin nimestä. Tässä käytimme aina striimin nimenä ”movie”.

```
<script>
(function(){
  var url =
    "http://192.168.10.49/dash/movie.mpd";
  var context = new Dash.di.DashContext();
  var player = new MediaPlayer(context);
  player.startup();
  player.attachView(document.querySelector("#videoPlayer"));
  player.attachSource(url);
})();
</script>
</body>
</html>
```

### 7.2.2 Testiympäristön MPD

Kun Nginx:n RTMP-moduuli vastaanottaa saapuvan RTMP-lähetyksen, se kääntää sen DASHiksi ja luo temp-kansioon väliaikaisen mpd-tiedoston streamin nimellä. Moduuli vastaanottaa saapuvan RTMP-lähetyksen ja kirjoittaa siitä saadut RTMP-urilit MPD-tiedostoon. Tiedosto päivitetään aina kun uusi palanen lähetystä on tarjolla. Segmentti luodaan siten, että jokaisen segmentin alussa on keyframe ja segmentin pituus saadaan kertomalla keyframien määrä frame ratella. Esim. neljän sekunnin keyframe-välillä saadaan 24fps videossa segmentin pituudeksi  $4 \cdot 24 = 96$ (framea). (Bitdash 2014.)

Tässä esimerkkinä luotu movie.mpd, kun lähetettiin yhtä 720p RTMP-streamia Nginx-palvelimelle ja siitä edelleen clientille. Alussa määritellään mm. käytetty profiili, lähetyksen alkamisaika yms.:

```
<?xml version="1.0"?>
<MPD
  type="dynamic"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  availabilityStartTime="2015-12-07T16:01:26+02:00"
  availabilityEndTime="2015-12-07T16:02:02+02:00"
  minimumUpdatePeriod="PT5S"
  minBufferTime="PT5S"
  timeShiftBufferDepth="PT0H0M0.00S"
  suggestedPresentationDelay="PT10S"
  profiles="urn:hbbtv:dash:profile:isoff-live:2012,urn:mpeg:dash:profile:isoff-live:2011"
  xmlns:xsi="http://www.w3.org/2011/XMLSchema-instance"
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011 DASH-MPD.xsd">
  <Period start="PT0S" id="dash">
```

Lähetys on dynaaminen, eli sillä ei ole ennaltamääritettyä kestoja ja väliin voidaan upottaa esim. mainoksia. Lähetykselle määritellään aloitusaika ja siihen mennessä saatujen palasten mukaan laskettu loppuaika. Nämä tiedot päivittyvät lähetyksen edessä. Päivitysten väli ja bufferointiaika on asetettu viiteen sekuntiin. MinimumUpdatePeriod määrittää myös sen, että lähetys on saatavilla vähintään sen loppumiseen asti. Profiiliksi määritetään live.

Adaptation Set määrittää resoluution ja frame raten:

```
<AdaptationSet
  id="1"
  segmentAlignment="true"
  maxWidth="1280"
  maxHeight="720"
  maxFrameRate="25">
```

Representation määrittelee käytetyt koodekit ja tiedostotyyppin:

```
<Representation
  id="movie_H264"
  mimeType="video/mp4"
  codecs="avc1.42001f"
  width="1280"
  height="720"
  frameRate="25"
  sar="1:1"
  startWithSAP="1"
  bandwidth="1500000">
```

Seuraavaksi näkyy live-lähetykselle tyypillinen Segment Template, jossa lähetys on pilkottu useaan osaan, joita client sitten järjestyksessä pyytää. Osia luodaan dynaamisesti lähetysten jatkuessa.

```
<SegmentTemplate
  presentationTimeOffset="0"
  timescale="1000"
  media="movie-$Time$.m4v"
  initialization="movie-init.m4v">

  <SegmentTimeline>
    <S t="101960" d="6000"/>
    <S t="107960" d="6000"/>
    <S t="113960" d="6000"/>
    <S t="119960" d="6000"/>
    <S t="125960" d="6000"/>
    <S t="131960" d="6000"/>
  </SegmentTimeline>
</SegmentTemplate>
</Representation>
```

DASHille tyypillisesti lähetys on jaettu erillisiin Adaptation Set-osiin, joista ensimmäisessä lähetetään video ja toisessa audio. Videon tapaan määritellään järjestyksessä tiedostotyytit, koodekit ja segmenttien ajat.

```
</AdaptationSet>

<AdaptationSet
  id="2"
  segmentAlignment="true">

  <AudioChannelConfiguration
    schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011"
    value="1"/>
  <Representation
    id="movie_AAC"
    mimeType="audio/mp4"
    codecs="mp4a.40.2"
    audioSamplingRate="48000"
    startWithSAP="1"
    bandwidth="96000">
    <SegmentTemplate
      presentationTimeOffset="0"
      timescale="1000"
      media="movie-$Time$.m4a"
      initialization="movie-init.m4a">
      <SegmentTimeline>
        <S t="101960" d="6000"/>
        <S t="107960" d="6000"/>
        <S t="113960" d="6000"/>
        <S t="119960" d="6000"/>
        <S t="125960" d="6000"/>
        <S t="131960" d="6000"/>
      </SegmentTimeline>
    </SegmentTemplate>
  </Representation>
</AdaptationSet>
</Period>
</MPD>
```



Lähetyksen jälkeen MPD-tiedosto ei ole enää saatavilla. Konfiguraatiossa määritelty `drop_idle_publisher 10s`; lopettaa saman lähetyksen kuuntelun palvelimen puolella, ja Nginx on valmis vastaanottamaan uuden RTMP-streamin ja luomaan sille uuden väliaikaisen MPD-tiedoston.

Lähetyksestä olisi voinut samalla luoda myös muita versioita, kuten HLS-lähetyksen. Ominaisuuden saa yksinkertaisesti käyttöön lisäämällä konfiguraatioon:

```
application hls {  
    live on;  
    hls on;  
    hls_path /var/www/hlsLive  
}
```

### 7.3 Tulokset

Lähetetty 720p-video oli katsottavissa noin 20-30 sekunnin kuluttua RTMP-streamin lähetyksen alkamisesta. Videokuvan laatu oli erinomainen ja minkäänlaista pätkimistä ei ollut havaittavissa. Uuden lähetyksen aloittamisessa oli aluksi ongelmaa, mutta ongelma korjaantui `drop_idle_publisher`-asetuksella. Tämän jälkeen lähetykset sujuivat taas ongelmitta.

Videolähetystä pääsi katsomaan samassa verkossa olevilla laitteilla. Sisäverkkotestin jälkeen järjestelmää testattiin myös julkisen-ip:n yli ilman ongelmia. Windows-tietokoneilla lähetystä pystyi katsomaan Chrome- ja IE11-selaimella. Applen laitteet eivät odotetusti toistaneet lähetystä. Sen sijaan pienenä yllätyksenä vanhempi Lumia 925 toisti lähetyksen mainiosti. Tähän mennessä Lumiaan ei ollut saatu tarjottua live-lähetyksiä käytettävillä tekniikoilla. Android-pohjaisilla puhelimilla toisto onnistui ongelmitta Chrome-selaimella.

Testiympäristöllä voitaisiin muodostaa myös HLS-stream ja luoda fallback DASHista siihen, mikäli selain ei DASHia tue. Soittimena voitaisiin käyttää hls.js-pohjaista HTML5-natiivisoitinta tai erillistä soitinta, kuten Flowplayer, joka sekkin pohjautuu hls.js-toiminnallisuuteen.

Kokonaisuudessaan DASH-tuki on esitelty vielä taulukossa 3. Käytännössä kaikki uudemmat päätelaitteet Applen iOS-laitteita lukuunottamatta pystyvät näyttämään DASH-sisältöä. Mukana on myös tekniikkaa tukevat älytelevisiot.

Device	DASH HTML5	DASH Flash	DASH Native
Chrome >= 30	✓	✓	
Chrome < 30		✓	
Safari 8	✓	✓	
Safari < 8		✓	
Firefox >= 42	✓	✓	
Edge	✓	✓	
IE 11 Windows 8	✓	✓	
IE <= 11		✓	
HbbTV 1.5			✓
HbbTV 2.0			✓
Samsung Smart TV 2012+			✓
LG Smart TV 2012+			✓
Sony TV 2012+			✓
Philips NetTV 4.1+			✓
Panasonic Vierra 2013+			✓
ChromeCast	✓		✓
Android >= 4.0 with Chrome	✓		
Windows Phone 8	✓		
iOS			
Apple TV			

Taulukko 3. Selainten ja päätelaitteiden DASH-tuki (Lederer 2015.)

## 8 YHTEENVETO

Opinnäytetyön käytännön testien perusteella MPEG-DASH on erittäin toimiva ja käyttökelpoinen tekniikka tulevaisuutta varten. Tekniikan laiteriippumattomuus on erityisen hyödyllistä webcast-käytössä, kun katsojat ovat ympäri maailmaa vaihtelevien päätelaitteiden ja yhteyksien päässä. Tekniikan yleistymistä varjostaa lähinnä Applen pysyminen HLS:n parissa. HLS.js:n mahdollistama fallback tarjoaa mahdollistaa vain yhden soittimen tarjoamisen asiakkaalle, ilman ulkoisten sovellusten asentamista. MPEG-DASHin standardi antaa melko vapaat kädet suoratoistolähetysten toteuttamiseen ja tarjoaa yksinkertaisessa paketissa laajan tuen eri päätelaitteille.

Käytännön osuudessa Nginx tarjosi hyvän pohjan tekniikan kokeiluun ja voitiin huomata MPEG-DASHin olevan jo nyt hyvin toimiva ja yksinkertainen lähetystekniikka suoratoistoon. Tulevaisuuden käyttöä varten tullaan kuitenkin tarvitsemaan Akamain palvelu, jotta voidaan edelleen lähettää nykyisen laitteiston tukema RTMP-stream transkoodausta varten.

Kaupalliset toimijat kehittävät jatkuvasti uusia tapoja välittää videota ja taapahtumia yleisölle, ja DASH tulee olemaan avainasemassa käytännössä kaikessa. HbbTV nojaa vahvasti DASHin avoimeen standardiin ja sen tarjoamaan HEVC-tukeen. Tämä tuki on äärimmäisen tärkeä 360-toteutuksissa, jossa kaistanarve kasvaa suureksi laajan videokuvan myötä.

## LÄHTEET

Adobe. 2011. Real-Time Messaging Protocol (RTMP) specification. Viitattu 25.2.2016  
<http://www.adobe.com/devnet/rtmp.html>

Akamai. 2014. Adaptive Media Delivery. PDF-tiedosto. Viitattu 20.2.2016.

Akamai 2016. Seminaari-esitys 03/2016. Viitattu 1.5.2016.

Anders, J. MPEG Video Compression Technique. Viitattu 24.2.2016.  
[https://vsr.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg\\_tech.html](https://vsr.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg_tech.html)

Bitdash. 2014. MPEG-DASH CONTENT GENERATION USING MP4BOX AND X264. Viitattu 5.5.2016  
<http://www.dash-player.com/blog/2014/11/mpeg-dash-content-generation-using-mp4box-and-x264/>

Bultje, R. 2015. VP9 Encoding/decoding performance vs HEVC/H.264. Viitattu 3.5.2016  
<https://blogs.gnome.org/rbultje/2015/09/28/vp9-encodingdecoding-performance-vs-hevch-264/>

Crijns, K. 2014. H.265 a.k.a. HEVC: Videocodec of the future. Viitattu 15.2.2016  
<http://us.hardware.info/reviews/5145/6/h265-aka-hevc-videocodec-of-the-future-profiles-and-levels>

Dash IF. 2015. About. Viitattu 19.10.2015.  
<http://dashif.org/about/>

Digita. 2016. Hybridi-tv:ssä yhdistyvät antenni-tv:n ja netin parhaat puolet. Viitattu 3.5.2016  
<http://www.digita.fi/hybriditv>

DigitalTVEurope. 2015. HbbTV 1.5 version of OTT TV service Ruutu TV now on Digita. Viitattu 3.5.2016.  
<http://www.digitaltveurope.net/412051/hbbtv-1-5-version-of-ott-tv-service-ruutu-tv-now-on-digita/>

Encoding. 2014. What is Adaptive Bitrate Streaming? Knowledge base. Viitattu 7.1.2016.  
<http://help.encoding.com/knowledge-base/article/what-is-adaptive-bitrate-streaming/>

Encoding. 2016. MPEG-DASH – An Overview. Viitattu 23.2.2016.  
<http://www.encoding.com/mpeg-dash/>

Github. Hls.js. Viitattu 2.5.2016  
<https://github.com/dailymotion/hls.js>

- HbbTv. 2015. HbbTv Specification FAQ. Viitattu 3.5.2016  
<https://www.hbbtv.org/wp-content/uploads/2015/07/HbbTV-Specification-2.0-FAQ.pdf>
- Hermans, A. 2012. H.264/MPEG-4 Advanced Video Coding. Seminar Report. PDF-tiedosto. Viitattu 23.2.2016.
- Highwinds. 2015. Utilizing HLS for Delivering Streaming Video. Viitattu 1.5.2016.  
<https://www.highwinds.com/blog/hls/>
- Hughes. 2014. DASH Live Streaming with Azure Media Service. Microsoft Azure. Viitattu 20.2.2016.  
<https://azure.microsoft.com/en-us/blog/dash-live-streaming-with-azure-media-service/>
- Hughes N. 2016. Google to ban Adobe Flash-based display ads, go 100% HTML5. Apple Insider. Articles. Viitattu 1.5.2016.  
<http://appleinsider.com/articles/16/02/10/google-to-ban-adobe-flash-based-display-ads-go-100-html5>
- Indiana University. 2016. What is an "application/octet-stream" MIME attachment, and how can I see it? Knowledge base. Viitattu 5.5.2016.  
<https://kb.iu.edu/d/agtj>
- ISO/IEC. 2014. 23009-1. Information technology — Dynamic adaptive streaming over HTTP (DASH) – Part 1 Media presentation description and segment formats Viitattu 25.2.2016
- Kapoor, A. 2009. Dynamic streaming on demand with Flash Media Server 3.5. Adobe Developer Connection. Viitattu 1.5.2016.  
[http://www.adobe.com/devnet/adobe-media-server/articles/dynstream\\_on\\_demand.html](http://www.adobe.com/devnet/adobe-media-server/articles/dynstream_on_demand.html)
- Kavon, A. 2014. How To Optimize Nginx Configuration. DigitalOcean. Viitattu 25.2.2015.  
<https://www.digitalocean.com/community/tutorials/how-to-optimize-nginx-configuration>
- Long, B. 2015. The Structure of an MPEG-DASH MPD. Viitattu 25.2.2015.  
<https://www.brendanlong.com/the-structure-of-an-mpeg-dash-mpd.html>
- Malkiel, S. 2015. H.264 and HEVC MPEG DASH LIVE STREAMING. Viitattu 7.1.2016.  
<https://www.linkedin.com/pulse/h264-hevc-mpeg-dash-live-streaming-sharon-malkiel>
- Mozilla. 2016. <video>. Mozilla Developer Network. HTML. Viitattu 15.1.2016.  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video>

Nginx. About. Viitattu 5.5.2016

<http://nginx.org/en/>

Ozer, J. 2009. Encoding Options for H.264 Video. Adobe Developer Connection. Viitattu 24.1.2016.

[http://www.adobe.com/devnet/adobe-media-server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html)

Ozer, J. 2011. What is H.264? Streaming Media. Articles. Viitattu 19.10.2015.

<http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx>

Ozer, J. 2011. What Is MPEG-DASH? Streaming Media. Articles. Viitattu 25.2.2016.

<http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-MPEG-DASH-79041.aspx>

Ozer, J. 2011. What Is Streaming? Streaming Media. Articles. Viitattu 9.9.2015.

<http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=74052>

Ozer, J. 2013. What is HEVC (H.265?) Streaming Media. Articles. Viitattu 23.2.2016.

[http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-HEVC-\(H.265\)-87765.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-HEVC-(H.265)-87765.aspx)

Ozer, J. 2014 HEVC: Are We There Yet? Streaming Media. Articles. Viitattu 7.2.2016.

<http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/HEVC-Are-We-There-Yet-99363.aspx>

Ozer, J. 2015. An Unhappy Surprise: MPEG LA Is Forming a Patent Pool for DASH. Streaming Media. Articles. Viitattu 26.12.2016.

<http://www.streamingmedia.com/Articles/News/Online-Video-News/An-Unhappy-Surprise-MPEG-LA-Is-Forming-a-Patent-Pool-for-DASH-105419.aspx>

Perrott, S. 2015. MPEG-DASH Test Streams. BBC. Research & Development. Blog. Viitattu 5.11.2015.

<http://www.bbc.co.uk/rd/blog/2013/09/mpeg-dash-test-streams>

Protalinski, E. 2016. Chrome passes 25% Market Share, IE and Firefox Slip. Viitattu 5.1.2016.

<http://venturebeat.com/2015/05/01/chrome-passes-25-market-share-ie-and-firefox-slip/>

Red. 2016. Chroma Subsampling Techniques. Learn. Viitattu 26.1.2016.

<http://www.red.com/learn/red-101/video-chroma-subsampling>

Richardson, I. 2011. H.264 Video Compression: An Overview. Viitattu 20.11.2015.

<http://www.slideshare.net/vcodex/h264-video-compression-an-overview>

Siglin, T. 2012. When Will H.265 HEVC Arrive and What Will it Mean for MPEG DASH. Streaming Media. Articles. Viitattu 3.1.2016.

<http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/When-Will-H.265-HEVC-Arrive-and-What-Will-it-Mean-for-MPEG-DASH-84481.aspx>

Tavakoli, S. 2016. Publishers still need flash. Viitattu 5.5.2016.

<http://adexchanger.com/tv-and-video/publishers-still-need-flash/>

Tharakan. 2015. Strong dollar hurts Akamai's profit forecast, shares fall. Reuters. Articles. Viitattu 20.2.2016.

<http://www.reuters.com/article/us-akamai-tech-results-idUSKBN0NJ2IV20150428>

Vanam, R. 2007. Motion Estimation and Intra Frame Prediction in H.264/AVC/Encoder. Viitattu 19.2.2016.

<https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahul-large.pdf>

Wolynski S., Ribeiro F. 2016. Flash-Free Video in 2016. New York Times. Blogs. Viitattu 1.5.2016.

<http://open.blogs.nytimes.com/2016/02/08/flash-free-video-in-2016/>

Wowza. 2011. What is Streaming? Resources. Viitattu 12.10.2015.

<https://www.wowza.com/resources/what-is-streaming>

Wowza. 2016. How to do MPEG-DASH streaming. Streaming engine. Tutorials. Viitattu 12.10.2015.

<https://www.wowza.com/forums/content.php?508-How-to-do-MPEG-DASH-streaming>

KUVA 1: Wowza. 2011. What is Streaming? Resources. Viitattu 12.10.2015.

<https://www.wowza.com/resources/what-is-streaming>

KUVA 2: Kone Interim Report for January. 2016. Kuvakaappaus webcast-lähetyksestä. Viitattu 5.5.2016

[http://qsb.webcast.fi/k/kone/kone\\_2016\\_0421\\_q1/#/webcast](http://qsb.webcast.fi/k/kone/kone_2016_0421_q1/#/webcast)

KUVA 3: Akamai. 2014. Adaptive Media Delivery. PDF-tiedosto. Viitattu 20.2.2016.

KUVA 4: Chroma subsampling. Wikipedia. Viitattu 1.5.2016

[https://en.wikipedia.org/wiki/Chroma\\_subsampling](https://en.wikipedia.org/wiki/Chroma_subsampling)

KUVA 5: Richardson, I. 2011. White Paper: An Overview of H.264 Advanced Video Coding Viitattu 20.11.2015

<http://www.slideshare.net/vcodex/h264-video-compression-an-overview>

KUVA 6: Ozer, J. 2009. 720p videotiedosto eri entropiakoodausmenetelyillä. Viitattu 26.12.2015

[http://www.adobe.com/devnet/adobe-media-server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html)

KUVA 7: Richardson, I. 2011. White Paper: An Overview of H.264 Advanced Video Coding Viitattu 20.11.2015

<http://www.slideshare.net/vcodex/h264-video-compression-an-overview>

KUVA 8: Richardson, I. 2011. White Paper: An Overview of H.264 Advanced Video Coding Viitattu 20.11.2015

<http://www.slideshare.net/vcodex/h264-video-compression-an-overview>

KUVA 9: Anders, J. MPEG Video Compression Technique. Viitattu 24.2.2016.

[https://vsr.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg\\_tech.html](https://vsr.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg_tech.html)

KUVA 10: Vanam, R. 2007. Motion Estimation and Intra Frame Prediction in H.264/AVC/Encoder. Viitattu 19.2.2016.

<https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahul-large.pdf>

KUVA 11: Vanam, R. 2007. Motion Estimation and Intra Frame Prediction in H.264/AVC/Encoder. Viitattu 19.2.2016.

<https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahul-large.pdf>

KUVA 12: Ozer, J. 2013. What is HEVC (H.265?) Streaming Media. Articles. Viitattu 23.2.2016.

[http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-HEVC-\(H.265\)-87765.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-HEVC-(H.265)-87765.aspx)

KUVA 13: Ozer, J. 2013. What is HEVC (H.265?) Streaming Media. Articles. Viitattu 23.2.2016.

[http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-HEVC-\(H.265\)-87765.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-HEVC-(H.265)-87765.aspx)

KUVA 14: Giladi, A. 2011. Using DASH and MPEG-2 TS. Viitattu 13.2.2016 .

<http://www.slideshare.net/agiladi/alex-giladi-dashtsr3>

KUVA 15: Akamai. 2015. On the Road To 4K Streaming With MPEG-DASH. Viitattu 15.2.2016. PDF-tiedosto.

KUVA 16: ISO/IEC. 2014. 23009-1. Information technology — Dynamic adaptive streaming over HTTP (DASH) – Part 1 Media presentation description and segment formats Viitattu 25.2.2016



KUVA 17: Streaming Media West. 2011. MPEG-DASH: Driving The Growth Of Streaming Using The New HTTP Standard. Viitattu 20.1.2016  
<http://www.streamingmedia.com/Conferences/West2011/docs/SMWest2011-MPEG-Dash.pdf>

KUVA 18: TVTechnology. Mobile Video Delivery. Viitattu 3.5.2016  
<http://www.tvtechnology.com/cable-satellite-iptv/0149/mobile-video-delivery/235555>

KUVA 19: Rosoff, M. 2015. Slowly but surely, web sites are abandoning Flash. Business Insider UK. Viitattu 2.5.2016  
<http://uk.businessinsider.com/flash-usage-decline-2015-9?r=US&IR=T>

KUVA 20: Cisco. 2015. Cisco AnyRes Live 9500 UHD Encoder Data Sheet. Viitattu 26.2.2016  
<http://www.cisco.com/c/en/us/products/collateral/video/videoscape-anyres-live/datasheet-c78-734319.html>

KUVA 21: Shuttle. Tuotesivu.  
<http://www.shuttle.eu/products/slim/ds81/overview/>

KUVA 22: Adobe Flash Media Live Encoder. Kuvakaappaus. Viitattu 10.2.2016

TAULUKKO 1: Ozer, J. 2009. Viitattu 26.12.2015  
[http://www.adobe.com/devnet/adobe-media-server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html)

TAULUKKO 2: Mueller, C. 2015. MPEG-DASH vs. Apple HLS vs. Microsoft Smooth Streaming vs. Adobe HDS. Bitmovin. Viitattu 22.2.2016  
<https://www.bitcodin.com/blog/2015/03/mpeg-dash-vs-apple-hls-vs-microsoft-smooth-streaming-vs-adobe-hds/>

TAULUKKO 3: Lederer, S. MPEG-DASH browser support and device compability. Bitmovin. Viitattu 2.5.2016  
<https://bitmovin.com/mpeg-dash-browser-support-device-compatibility/>

