

Mathias Dahlstrøm

Streamingboomen

Et teknisk overblikk på hvordan underholdning strømmes til norske konsumenter i 2022

Bacheloroppgave i informasjonsbehandling

Veileder: Atle Nes

Mai 2022

Mathias Dahlstrøm

Streamingboomen

Et teknisk overblikk på hvordan underholdning strømmes til norske konsumenter i 2022

Bacheloroppgave i informasjonsbehandling
Veileder: Atle Nes
Mai 2022

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk



Kunnskap for en bedre verden

Innholdsfortegnelse

Introduksjon	5
Avgrensning	5
Forkunnskap	6
Bakgrunn	7
Begrepsavklaring	8
IPTV	8
Nettverksprotokoller ved strømming	10
Streaming-tilbyderne	11
Kundeforventninger	11
QoS / QoE	12
Adaptive bit rate	13
Content Delivery Network	15
Beskyttelse av innholdet	16
Streaming protokollene	18
Dynamic Adaptive Streaming over HTTP	18
HTTP Live Streaming	21
Microsoft Smooth Streaming	23
HTTP Dynamic Streaming og Real Time Messaging Protocol	25
Secure Reliable Transport	26
Common Media Application Format	27
Kodek	29
Advanced Video Coding	30
VP9	31
High Efficiency Video Coding	31
AV1	32
Versatile Video Codec	32

Utviklingen til streaming	34
Tidlig strømming	34
Plugins	35
Felles format	36
Hvor leder utviklingen?	37
Oppsummering	39
Videre forskning	41
Referanseliste	42

Introduksjon

Formålet med denne oppgaven er å kartlegge hvilke teknologier, prosesser og formater som danner grunnlaget for tilgjengeliggjøringen av strømmehold slik vi kjenner det i dag. Å streame eller strømme er gjennomgått en stor utvikling på de få årene konseptet har eksistert. I dag benyttes teknologien til så mangt, men forutsetningene for det markedet vi har i dag har ikke alltid funnets. Det har foregått en rivende utvikling innen ulike teknologier. Denne oppgaven har som mål å kartlegge hvilke ulike teknologier som er aktuelle og hvordan disse samkjøres for å kunne tilby tjenestene slik vi kjenner til de i dag.

Oppgaven vil dekke både hvordan de rent tekniske forutsetningene rundt streaming har blitt formet. Mål er å kunne gi et overblikk på hvordan de ulike teknologiene i dag benyttes til å levere strømmehold, ikke å skulle gå i detalj på den indre funksjonaliteten til hver enkelt teknologi. Ettersom jeg ved oppstart av oppgaven ikke nødvendigvis kjenner til alle disse teknologiene, vil jeg ta utgangspunkt i de tre hovedelementene jeg kjenner til at streaming er bygd opp av. Oppgaven vil derfor behandle hvordan den videotekniske kompresjonen og innpakningen av innholdet i ulike formater har bidratt og bidrar til streaming. Hvordan utviklingen og virkemåten bak de ulike streamingprotokollene påvirker tjenestene. Og også hvordan webteknologien for å tjene klientene med videoinnhold har formet seg og fungerer i dag for å effektivt kunne tilby strømmehold. Til sist er det også et mål med oppgaven å se de overordnede trendene i utviklingen som har foregått og kanskje også hvor teknologien sikter videre.

Avgrensning

Jeg har identifisert at det i dag finnes mange ulike former for streaming, benyttet både i undervisning, videokonferanser, undervisning livestreaming, tv og annen videounderholdning med mer til. Det er derfor gjort et bevisst valg i denne oppgaven å holde seg til teknologien bak underholdningstjenester som Netflix, Viaplay, Tv2 Play, Discovery+ og NRK TV med mer. Teknologiene som benyttes til streaming av underholdning er i stor grad de samme som benyttes og muliggjør andre former for streaming, og mye i oppgaven vil derfor også være gjeldende for andre type streaming uten at dette behandles direkte. Streaming til andre formål utgjør en voksende og stor andel av de kommersielle aktører, med ulike produkter som tjener sitt spesifikke formål. Denne delen av streaming har også vokst kraftig, og blitt et uoversiktlig marked med mange ulike interesser som ikke videre vil bli behandlet i denne oppgaven.

Forkunnskap

Denne oppgaven skal gi et overordnet blikk på hvordan strømmehold teknisk i dag deles med sluttbruker. Oppgaven er ment å både kunne gi et innblikk i de overordnede teknologiene forbundet med streaming. Både for de som allerede innehar en del teknisk kompetanse. Men målet er at det også skal være mulig å følge store deler av oppgaven for de med mindre teknisk innsikt som søker seg et overblikk over hvordan strømmehold deles. Det vil allikevel være en fordel å være kjent med oppbygningen av videofiler med både kodek og container. Og begreper som TCP/UDP og HTTP kan være nyttig å kjenne til hvordan fungerer overordnet når de benyttes i overføring av data på nett. Gjennom oppgaven forsøker jeg derfor å ha med korte introduksjoner til både videofiler og nettverksprotokoller for å gi nødvendig innsikt eller repetisjon ved lesningen av oppgaven.

Bakgrunn

Det har vært en rivende utvikling i distribusjonen av bevegelige bilder. Fra de første levende bildene ble vist for et publikum av brødrene Lumière, så til de første analoge tv-sendingene nesten et halvt århundre senere hvor levende bilder kan konsumeres i ens eget hjem fremfor tilpassede teater og kinoer. Teknologien videreutvikler seg, og sendingene går over til fargebilder og blir også med tiden allemannseie. Man finner en tv i tilnærmet hvert hjem hundre år etter Lumière for første gang viste frem sin nye teknologi så storslaget for et publikum i kjelleren på Grand Café i Paris. På 2000-tallet har også tv-signalene gjennomgått en digitaliseringsprosess og i dag leveres levende bilder i mange ulike former, formater og til utallige ulike enheter for konsument. Forbrukere stiller også helt andre krav, og er «/.../klokere, mer selvbevisste, mer utålmodige, kresne og selektive/.../» (Retail magasinet, 2019).

Utviklingen i den tekniske visningen har ikke stoppet opp, men heller blitt så tilgjengelig for allmennheten at den er blitt en mer forbruker og behovsstyrt utvikling. Teknologien finnes eller kan skapes for å levere medieinnhold til tilnærmet alle gjennom streaming. Du er ikke lengre avhengig av å møte opp i en kinosal, ha en kablet tv-tilkobling til en leverandør helt inn til huset eller i det heletatt ha en stasjonær tv i stuen. I dag stilles det mindre krav til forbrukerens utstyr og enhet, og signalet kan leveres tilnærmet hvor som helst, når som helst og på kryss av landegrenser. Vi kan hente medieinnholdet over internett enten kablet, trådløst eller over telenettet og spille av enten på en mobil, nettbrett, datamaskin, TV-boks eller TV, for å nevne noe.

I 2020 og de påfølgende årene med korna-pandemien har også betydelig påvirket strømmemarkedet. Både i forhold til nytte, bruk og folks forhold til streaming. I forkant av pandemien var et møte på jobb forutsatt tilknyttet både en tid og ikke minst et sted. To år etter vil man måtte presisere at et møte ikke bare skal holdes helt eller delvis digitalt. Nå er et møte like gjerne på et virtuelt møterom baserer på streaming gjennom tjenester som blant andre Teams, zoom eller Google Meet. Undervisning som til vanlig alltid har vært i klasserom eller forelesningssaler har plutselig blitt et digitalt møte eller en videostrøm man kan følge enten live eller når det passer en selv. Også Hollywood tilpasset seg hurtig en ny digital hverdag under pandemien, og satset stort på strømming. Den vanlige formelen for lansering av spillefilmer hvor man først har en runde på kino, og deretter etter litt tid tilgjengeliggjør

tittelen i strømmetjenestene slo sprekker. I 2020 hadde mange Hollywood-filmer premieren sin i ulike digitale strømmetjenester også uten kinovisning. (Grut, 2020)

Streaming i mange former har inntatt hverdagene våre, både unge som eldre i et tempo man aldri hadde sett for seg. Den store ryggraden i denne omstillingen er den digitale utviklingen som har foregått på området de siste årene, drevet av brukere som i stor grad er tvunget til å adaptere teknologien hurtigere enn noen kunne forventet. Dette har gått på grunn av den solide infrastrukturen som har tillatt den kraftige økningen i datatrafikken og serverkapasiteten som kreves av denne utviklingen. (Husby, 2020)

Begrepsavklaring

I dagligtalen brukes først og fremst nettopp ordet «Streaming» som en fellesbenevnelse for å se film, tv eller serier på en mobil, datamaskin eller tv som henter innholdet over nett. Som en stor andel tekniske begreper er det et engelsk ord som direkte er blitt tatt i bruk i Norsk som et eget begrep. Skriftlig har vi det norske ordet strømming som er ment å være dekkende, gjennom sin beskrivelse i ordboken som «det å overføre og spille av lyd eller film løpende uten å lagre først» (Språkrådet og Universitetet i Bergen, 2022). Fokuset i denne oppgaven er i all hovedsak på nettopp dette aspektet ved streaming, men jeg vil allikevel påpeke en vesentlighets forskjell i begrepene der streaming ikke bare er en teknisk overføring eller avspilling av innhold, men også en aktivitet man kan gjøre, en beskrivelse av muligheten for interaktivitet i avspillingen og at det er mer enn kun lyd og bildeinnhold kan streames. Enten det er spill som streames gjennom en konsoll eller datamaskin, eller det er tilleggsinnhold som teksting av medieinnholdet. Denne oppgaven vil i all hovedsak omhandle teknologiene som benyttes for over-the-top (OTT) streaming tjenestene, det gjengs mann i gata som regel forbinder med streaming. OTT er tjenestene som leveres «på toppen av» og uavhengig av de tradisjonelle distributørene av medieinnhold som tv-tilbydere som holder blant annet kabel-TV og satellitt-TV. OTT leveres direkte til sluttkundene over internett til alle de enheter som har støtte og tilkobles av sluttkunden. Og det er ingen behov for sluttkunden å forholde seg til noe sendeskjema, da tjenesten er tilgjengelig hele døgnet, og når man selv ønsker. (Tariq, 2021)

IPTV

IPTV er en forkortelse for Internet Protocol Television som består av TV-relaterte tjenester som lineær-tv, video on demand og andre strømmetjenester til sluttbrukeren. (Masciopinto, Comesaña og Pérez-González, s. 34) Dette utgjorde globalt i 2017 mere enn en tredjedel av

det totale tv-titter markedet og Telia kan i sin tv-rapport med tall fra 2021 melde at hele 95 prosent i Norge ser på lineær-TV i løpet av måneden. Mens nærmere 80 prosent tar i bruk video on demand (VoD). (Telia, 2022) Den store fordelene til IPTV sammenliknet med ordinær lineær-TV er mulighet for å adressere signalet og interaktivitet der brukeren selv kan velge hva som skal sendes til enhver tid. (Masciopinto, Comesaña og Pérez-González, 2018, s. 35) Ettersom IP-baserte nettverk tillater toveiskommunikasjon kan brukeren potensielt velge kanaler og programmer når det måtte passe en selv, velge ønsket supplerende data som eventuell teksting og elektronisk programguide uten noen involvering fra tjenestetilbyderen. Men også ha tilgang på video on demand tjenester og selv velge ønsket tv-innholdet som til enhver tid leveres.

Hvordan skiller så streaming seg fra dagens tv-tjenester eller IPTV? Distinksjonen mellom TV og streaming har nok både begrepsmessig og basert på leveringingen av tjenestene blitt likere hverandre. Når TV levers gjennom IPTV kan man i dag kategorisere dette som en type streaming og teknologisk har IPTV teknologien banet vei for streaming slik at tjenestene også leveres veldig likt hverandre i dag. Forskjellene var større i de tidlige dagene til både streaming og IPTV, men i dag har de både teknisk og innholdsmessig i stor grad blitt mye av det samme. Når vi snakker om live eller lineær TV over IPTV skiller fortsatt tjenesten seg mulig mest fra streaming ettersom det benyttes en fast bit rate på innholdet som sendes. Dette ettersom man har kontroll over fremføringsveien og ikke skulle måtte ta det samme hensynet til en mulig internetttilkobling som ikke har hurtig nok linje inn til seg, eller en oppsamling av ip-pakker i ruterne til mottaker. I tillegg skal akkurat de samme pakkene på samme tid til mange personer, og derfor finnes det flere måter å transportere dette på, derav hvor pakkene dupliseres kun den siste biten ut til mottakere, og sparer mye trafikk ute i nettet.

Når større internettleverandører tilbyr IPTV, er som regel dette signalet beregnet inn i hastigheten som tilbys til adressen og man kan sikre en jevn strøm med fast bit rate og fast bildekvalitet helt frem til mottakerenheten. Jeg vil argumentere for at et av kjennetegnene ved streaming er at det nettopp har måttet tilpasse seg til å kunne levere uavhengig av å ha tilgang eller kontroll på leveransepunkter helt frem til forbruker. Her enten om forbruker har en dårlig internettkapasitet, om kapasiteten er brukt opp grunnet mye annen samtidig bruk eller om forbruker befinner seg utenfor fastnett på mobilnettet, og kanskje til og med hopper mellom basestasjoner. Allikevel ingen regler uten unntak, det er også i dag begynt å levere IPTV over trådløs tilkobling både WIFI og mobilnett, der også leveringingen av lineært signal må tilpasses leveringsmetoden. VoD gjennom IPTV leveres derimot med leveransemetode mer likt øvrig

streaming for de to i stor grad, og hvor forskjellen mellom IPTV og streaming blir tilnærmet visket ut. Den tekniske utviklingen har ført de to tjenestene nærmere hverandre nå enn noen gang tidligere.

Nettverksprotokoller ved strømming

Det er to viktige ulike typer av nettverksprotokoller som er aktuelle å kjenne til ved streaming. TCP og UDP hører til i transportlaget og er to fundamentalt ulike metoder å forespørre og transportere data over nettet, og derfor også viktig å kjenne forskjellen på til videre i denne oppgaven.

UDP er forbindelsesløs og det man kaller en upålitelig protokoll ettersom det kun sendes ut en forespørsel uten å være bekreftet en kommunikasjon mellom to tjenere. Det bare sendes data uten mulighet for kvittering eller å vite om det mottas, og overføringen av data foregår uten noen kvittering mellom to UDP-porter. Dataen som sendes må være uavhengige datapakker. UDP benyttes i sammenhenger hvor det er meningsløst å vente på kvitteringssvaret og hvor man ønsker å slippe oppkoblingstiden for en pålitelig forbindelse. UDP benyttes om noe foregår i sanntid og hvor det vil være meningsløst med kvittering ettersom det uansett ville vært for sent å sende de eventuelt manglende pakkene på nytt. UDP er noe brukt i kringkasting av liveinnhold som IPTV.

Den store forskjellen til TCP er at det opprettes en bekreftet forbindelse og at det kvitteres for sending og mottak mellom tjenere. TCP benytter også flytkontroll der målet er å ikke sende data raskere enn mottaker klarer å motta til enhver tid. Det er en verdi i TCP-pakkeheaderen som styrer flytkontroll. Formålet med flytkontroll er å bruke en gylden middelvei i kommunikasjonen mellom to maskiner der man hele veien forsøker å sende en håndterbar mengde pakker til mottaker, som man jevnt får kvitteringer tilbake på. Man oppnår et kompromiss mellom hastighet og sikkerhet, og dette er funksjon som ligger i grunnen og bidrar også i transport under streaming gjennom HTTP. HTTP er applikasjonslaget hvor forespørsler og respons på forespørsler utveksles mellom en klient og en tjener. Dette er hvordan vi laster inn nettsider, og det er også overordnet slik en strømmetjeneste kan tilby sluttbrukeren å etterspørre en video og bli tilsendt denne bit for bit.

Streaming-tilbyderne

I Norge har det kommet både internasjonale og nasjonale selskaper som drifter og tilbyr strømming av ulikt innhold. Innholdet de distribuerer er både internasjonale innkjøp, egenproduksjoner og lineær-TV kanaler. Ut mot sluttbrukere annonseres og fokuseres det i markedsføringen i hovedsak på medieinnholdet som tilbys for å differensiere de større aktørene. Selve den tekniske løsningen er således ikke produktet som selges, men i hovedsak en forventet leveransmåte for produktet som forbrukeren stiller store krav til. Kravene kan være bevisste krav brukerne stiller til tjenestene som om det finnes mulighet for å laste ned innhold for avspilling offline. Eller heller mer uskrevne og underbevisste krav som at tjenesten skal levere innhold i høy nok kvalitet raskt nok til at det ikke stopper opp i avspillingen.

Det er mange avanserte tekniske mekanismer for leveringen av innholdet og innholdet skal være av mangfold og interesse for brukerne. Kanskje er det ikke så rart at flere aktører har passet på å levere sin plattform felles for sine kunder innen hele Norden og kanskje andre deler av Europa også. En del felles medieinnhold er å finne for at tjenesten kan fremforhandle bedre avtaler, men også for at det koster å ha et oppdatert og teknisk godt rustet økosystem for å distribuere strømming ut til kundene. Netflix har justert sitt tilgjengelige innhold til å være et bibliotek på omtrent 4 000, mot tidligere et DVD-bibliotek på vel 100 000 titler. (Vorel, 2021)

Når de store internasjonale aktører med all sannsynlighet bruker samme tekniske plattform i alle landene de opererer i med kun geoblokkering og regions tilpasset visning av innholdet som forskjell, skaper det uten tvil press på mindre aktører om å skape og kunne tilby et sammenlignbart produkt som tilfredsstillende forbrukerens tekniske ønsker. Det finnes også stor konkurranse fra de største internasjonale aktørene i å distribuere det beste innholdet etter en økt globalisering i verden. Konkurransen innen markedssegmentet drives i stor grad av å hente og tilegne seg en stor brukermasse for å kunne sikre sin egen inntjening, som igjen brukes for å utvikle og holde på det beste innholdet.

Kundeforventninger

Det er ikke behov for å gå spesielt mange år tilbake i tid hvor kundenes forventninger til strømmeselskapene var mindre eller intet eksisterende i form av den tekniske kvaliteten og simplisiteten til å kunne strømme deres innhold. Den gang var det forbehold de mest interesserte å skulle strømme film og videoinnhold over internett. I dag er det gjennomgått

mange steg i utviklinger for hvor enkelt det er å spille av en videostream. Nå forventes det å være like tilgjengelig som TV, og at alle aldersgrupper, og enten de er spesielt teknisk kompetent eller ei skal kunne strømme. Selv statskanalen har gjort en satsning på strømming av innhold for å henge med i tiden, og fordi de mener det øker tilgjengeligheten i tråd med sitt samfunnsoppdrag.

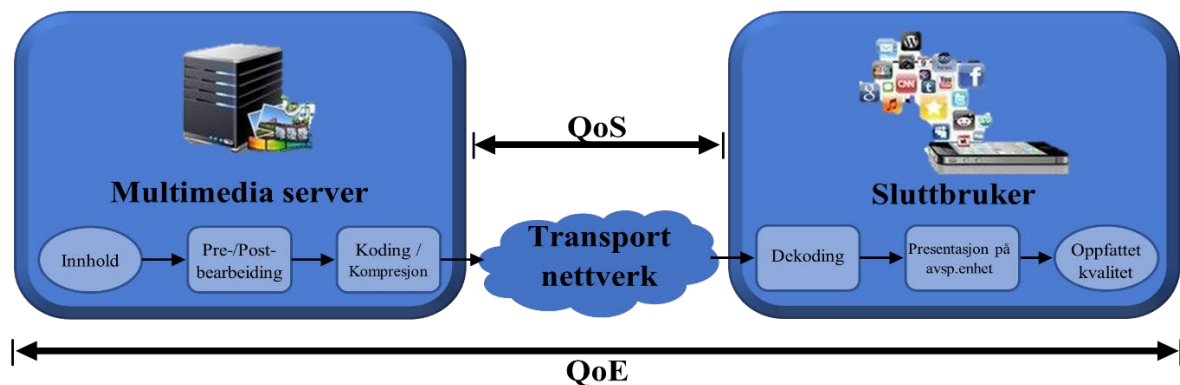
Det ligger også en forventning om at det skal være enkelt og tilgjengelig på mange ulike enheter, uten at det foreligger noe forståelse om at dette gjør det vesentlig mer komplekst for tilbyderne. Brukere forventer i dagens marked å kunne ha tilgang på OTT innhold gjennom app til både telefon, nettbrett og tv, samt en nettside man kan besøke for å strømme innhold direkte. (Bitmovin, 2021) Kvaliteten på strømmen av video skal helst også alltid være god og det skal være en sømløs opplevelse av produktet.

De fleste kan likevel leve med at kvaliteten dropper for en periode så lenge ikke underholdningen stopper helt opp. Selv om majoriteten ikke har et forhold til hva slags internettkapasitet som kreves for å hente inn en videostream i god kvalitet, så har de fleste en forventning om at innholdet ikke stopper opp. Og skulle det stoppe opp for å bufre, så vil det allikevel være uklart og likegyldig for de fleste om det skyldes internett eller bare en dårlig levering fra strømmetjenesten. Her har kanskje strømmetjenestene fått ufortjent kjeft i stua fra uvitende forbrukere som irriterer seg på at innholdet ikke spiller av som forventet, men kanskje skulle den kjeften egentlig blitt rettet mot en utdatert ruter med begrenset wifi som streamen forsøkes hentet inn på. Uavhengig er buffering for de fleste noe som hørte fortiden til, og som ikke skal oppleves med strømmetjenester sin QoE i 2022.

QoS / QoE

Quality of Service og Quality of Experience er to begreper som i stor grad har det samme fokuset på hvordan en tjeneste leverer. Og begge deler har stor innvirkning på kundeforventningene. QoS omfatter den tekniske leveransen fra server til enhet, og innebærer alle ledd i leveransen der imellom. Det omfatter blant annet om sluttbrukeren har lav hastighet på internett hjemme hos seg, eller en dårlig ruter som avspillerenheten akkurat står for langt unna. Og alle leddene fra medieinnholdet sendes fra server til det ankommer nettet til kunden og kundens enhet. Dette er forsøkt illustrert i figur 1 under, der QoS er markert som transporten i nettverket som leddet mellom server og avspillerenheten. I samme figur er QoE forsøkt markert der det innebærer hele løpet, ikke kun den tekniske transporten av innholdet. QoE handler om hele opplevelsen til sluttbruker. Alt fra om videoinnholdet har en god

komprimering og oppleves som av god kvalitet, til om avspillingen lagrer, stopper opp eller har dårlig kvalitet i avspillingsøyeblikket. Sluttbruker kan ha bra QoS, men samtidig dårlig QoE, eller vis versa, selv om de avhenger av hverandre. (Ozer, 2021) Det vil være en evigvarende diskusjon på nettopp hvordan man best måler disse to faktorene, men det er et forsøk på å få sammenliknbare tall som over tid kan si noe om hvordan tjenesten man leverer kan oppfattes av kundene.



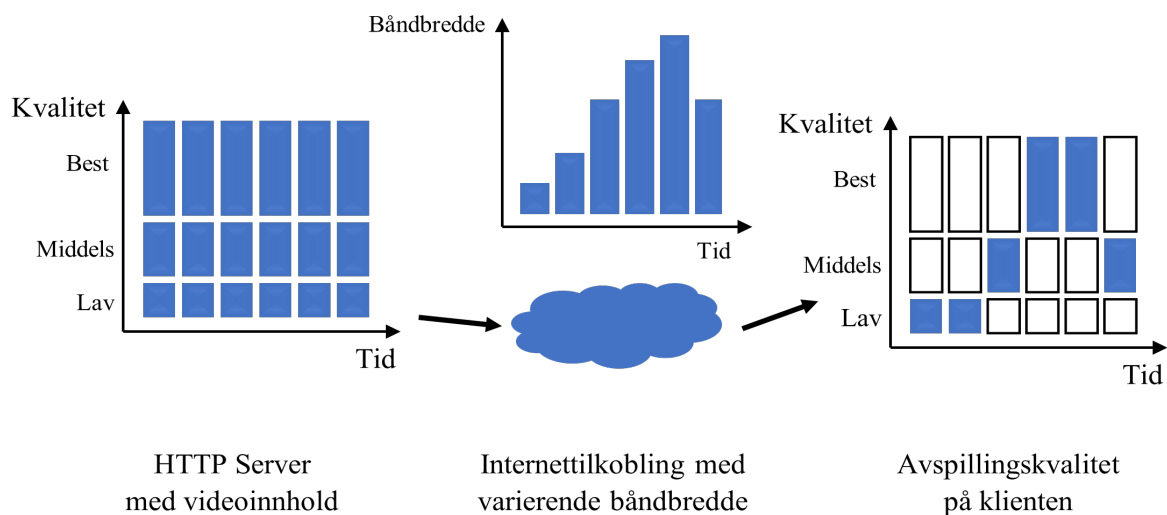
Figur 1 Illustrasjon av hva som omfattes av QoS og QoE

Når det kommer til ytelsesbergening og måling av dette, mener de som arbeider med streaming til at det er viktigst å ha fokus på hvilken bitrate som blir levert, buffertiden og buffer rate. Dette er resultatet som kom frem i 2022 rapporten til Bitmovin. Vi skal ikke mange år tilbake før den aller viktigste faktoren å måle var ansett som «video start up time», da dette virket å være av mest betydning for kundene. Dette hang igjen fra tidlig streaming, men så fort denne faktoren ble mestret var det naturlig å fokusere videre på andre mål som påvirker brukeropplevelsen. (Bitmovin, 2021) Det finnes mange ulike anbefalinger om hva man bør overvåke av avspillingsfeil for å kunne skape den beste opplevelsen for kundene, men ingen fasit da det i hovedsak er nødvendig å tilpasse til hver bruk og til hver type avspilling. Ikke minst er det like viktig hvordan man bruker dataene man får tilbake fra overvåkningen til å faktisk diagnostisere og håndtere utbedring av problemene som oppleves slik at opplevelsen av kvalitet kan forbedres.

Adaptive bit rate

Adaptive bit rate (ABR) er en av grunnsteinene i streaming slik vi kjenner den i dag, og er spesielt viktig i forbindelse med QoE. ABR har tilnærmet direkte innvirkning på alle de viktigste faktorene nevnt som måles i forbindelse med å kartlegge QoE. ABR benyttes av flere ulike streaming konsepter, og som ordet tilsier så er det tilpasning av bit rate. Enkelt forklart styring av hvor mye data som kan sendes over nett til en enkelt lokasjon. Når en

bruker ønsker å strøemme en film så vil den ledige hastigheten på linjen hele veien frem til brukeren være uviss. ABR vil derfor ut fra responsen på mottatte pakker beregne hvor store filer som kan sendes til brukeren uten at strømmen stopper opp. Hvis bildet i en strømmesessjon stopper opp skyldes det at pakkene med segmentet av videoen ikke har nådd frem med sitt segment av filmen. Som oftest skal det flere datapakker til for å få et segment med seg og når pakkene ikke har rukket å nå frem til brukeren før det skulle blitt avspilt, kan det oppleves som kanskje noe av det mest irriterende i verden at filmen stoppet opp på en upassende plass. ABR har derfor somoftest algoritmer i avspilleren som beregner hele veien om hvor store mengder data som kan bli mottatt i tide. Og så sendes segmenter i ulik bildekvalitet basert på dette. Enten datamengden kan være større eller datamengden må være mindre, så sendes segmentene basert på informasjonen om overføringen.



Figur 2 Virkemåten til ABR

Dette er illustrert i figur 2 der medieinnholdet er delt opp i mange segmenter, derav segmenter basert på samme innhold i ulik kvalitet. Målingene av tilgjengelig båndbredde avgjør hvilke segmenter som blir etterspurt eller sendt til mottaker enheten. Avspillingen av strømmen kan så være av mediesegmenter med ulik datakvalitet (bitrate), men strømmen stopper aldri opp. Den derimot tilpasser kvaliteten adaptivt underveis ut fra tilgjengelig båndbredde og bitrate på mediesegmentene.

ABR baseres på at det kun sendes mindre segmenter av medieinnholdet av gangen, og kan medføre at en bruker plutselig kan oppleve varierende kvalitet på videostrømmen. Dårligere kvalitet vil i større grad skje med noen med liten datalinje og hvor forhold rundt denne brukeren i nettet har stor innvirkning på brukerens mottak av strømmen. Mens en med stabil

og rask datalinje i liten grad vil merke noe til denne typen innblanding fra et ABR-system ettersom datapakker og segmenter av høy kvalitet hele veien virker å nå frem til brukeren i tide. At pakker ikke når frem kan skyldes det meste som dekkes av QoS som trafikk enten på veien eller i brukerens eget nett etter hvor mye aktivitet som er der. Eller det kan skyldes trafikken til en server. For å i størst mulig grad kunne luke ut de punkter som strømmetjeneste tilbyder kan påvirke, har derfor en rekke større selskaper avtale om CDN for å bedre QoS på de punktene de selv kan påvirke.

Content Delivery Network

Strømming avhenger i dag i stor grad av Content Delivery Network (CDN) for å levere innholdet til strømmen. Selve CDN-et og dens funksjon er utenfor hva denne oppgaven tenker å dekke, men for å forstå hvordan strømming av ulikt innhold i stor grad foregår i dag, var det viktig å forstå hvilken rolle CDN har i dette.

CDN kan best beskrives som en mellomlagring eller cache for internett. Det blir ofte referert til også som web-akselerasjon. I likhet med hvordan en datamaskin passer på å holde sine mest brukte programkoder, og datalokasjoner lett tilgjengelig for mellomlagring, tilbys den samme tjenesten for internett. Sammenliknet med de fleste andre former for innhold distribuert på nett, er videostreamer mye større og derfor også vanskeligere å levere og kan i stor grad nyttiggjøre seg av CDN. (Ozer, 2019¹)

Dagens mye benyttede metode er derfor å bygge et nettverk av serverer med det mye etterspurt innhold. Gjennom å ha mange servere å fordele trafikken på så unngår man også single point of failure, som har vært et problem gjennom en sentralisert nettverksarkitektur. (Fati og Sumari, 2018, s.11-12) Dette er sett på som en moderne måte å distribuere innhold på, der man allokere duplikater av innholdet mellom flere servere for å få bedre ytelse og skalerbarheten til tjenestene. De finnes flere ulike strategier for å tjene denne oppgaven på best mulig vis, med ulik struktur i lagdeling og hvordan forespørsler og filene fordeles på de tilgjengelige serverene.

Filmer og serier lagres og distribueres slik på et vis til forbruker gjennom et nettverk av lokasjoner som inneholder det mest etterspurte innhold. For å unngå blant annet lang transporter av strømmefilene fra en server på andre siden av verden, har flere av de største internasjonale selskapene avtaler med CDN-er som cacher opp på de mest etterspurte dataene i sentralt plasserte mer lokale servere nærmere kundene. Serverne er gjerne lokalisert nært nettverksleverandørenes større tilkoblingspunkter, og gir kortere transport til forbrukeren. Den

samme dataen trenger ikke å bli sendt duplisert til ulike adresser over mye av samme rute, men blir heller styrt fra en nærrere lokasjon hvor den tildeles forbrukeren. Dette gjør ofte de største organisasjonene for å optimalisere leveransen i ulike regioner. Som et eksempel har Telenor (2020) sin CDN-tjeneste ifølge deres egen tjenestebeskrivelse flere lokasjoner i Norge og en i Sverige.

Beskyttelse av innholdet

Når man arbeider med distribusjon av åndsverk, er det vanlig og en nødvendighet å beskytte innholdet man tilbyr for strømming. Både å skulle beskytte mot piratvirksomhet, men også i forhold til tilordning av tilgangsrettighet på å hente en strøm av innholdet. Ifølge Telia (2020) er populært benyttede metoder for å beskytte seg mot piratvirksomhet fra åpne nettsteder. Det ene er å benytte en token-mekanisme. Dette er oftest i benyttelse på livestreaming der det genereres en nøkkel for en sesjon eller tilkobling mot et domene. Skulle man prøve å dele innholdet eller hente strømmen utenfor dette domenet, så vil innholdet ikke kunne spille av. Det er også populært brukt for å forhindre hot-linking (direkte lenking) til beskyttet innhold. Den andre måten å beskytte multimedieinnhold som streames på er å benytte DRM.

Digital rights management (DRM) er en måte å beskytte opphavsretten til digital media. I de fleste tilfeller vil det tilsi at det finnes kode som forbyr kopiering eller begrenser antallet ganger eller antallet enheter det gis tilgang til et produkt. DRM innbefatter restriksjon av brukers muligheter for å redigere og lagre innhold, dele innhold, sette utløpsdato på tilgang til innhold og tildele tilgang til innhold basert på ip-adresse, lokasjon eller enhet. (Groot, 2018)

DRM har sine fordeler og kan bidra til å minske piratvirksomhet på opphavsberettiget materiale som video. Det er likevel ikke umulig å omgå eller fjerne DRM funksjonaliteten fra en fil, men allikevel kan DRM bidra gjennom å lære brukere om opphavsrett og åndsverk. Funksjonaliteten kan gi tydelige retningslinjer for brukeren om hva de kan og ikke kan gjøre. For streaming kan det bety for eksempel ikke tillate bruker å spille av en strøm ettersom bruker befinner seg utenfor tillatt geolokasjon og med medfølgende forklarende feilmelding. (Groot, 2018) Gjennom en feilmelding kan bruker bli gjort bevisst på at det ikke er mulig grunnet rettigheter, i stedet for at bruker får spilt av i brudd på rettighetene man kanskje ikke var klar over. Eller noe så vitalt som å sikre at det kun er betalende kunder som har tilgang på å strømme innholdet som etterspørres. Betaling for innhold er hvordan produsentene skal kunne sikre sitt eget inntektsgrunnlag og betale for en produksjon eller de kommende produksjonene av innhold.

Bitmovin (2021) kan rapportere at andelen som velger å ikke implementerer DRM i sitt multimedieinnhold reduseres år for år. De siste årene spesielt har det kommet en betraktelig nedgang fra 40% i 2019 til 20% i 2021. Det gjenspeiler industriens anerkjennelse av viktigheten av å beskytte opphavsrettighet materiale. Det er økning i både de som bruker et kommersielt DRM produkt, og de som har in-house DRM løsninger. De kan også melde om at piratøkosystemet rundt video skaper over en milliard dollar i året i fortjeneste årlig bare i USA. Noe som gir grunnlaget for flere og flere til å ta i bruk DRM løsninger som er støttet av streaming protokollene for å sikre innholdet etter beste evne.

Streaming protokollene

Streaming protokoller er en standardisert metode å levere ulike mediefiler over nettet. Fellesnevneren er at det sendes segmenter av innholdet som blir satt sammen igjen på andre siden. (Krings, 2021) Protokollene støtter ulike streaming formater som fungerer som container for videoinnholdet i streaming.

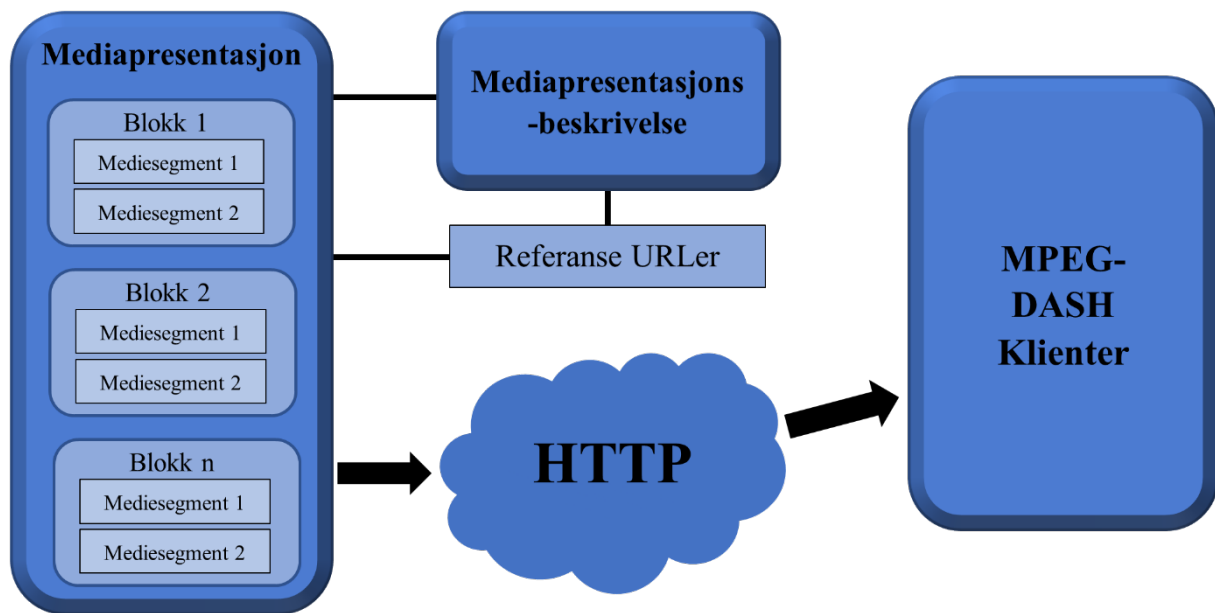
Alle teknologiene som benytter seg av HTTP adaptive bitrate teknologi fungerer på tilnærmet samme måte. Det hentes en manifest fil over segmentene, som så benyttes for finne frem til korrekt segment av ønsket video. Avspilleren forespør til en HTTP-server og så strømmes medieinnholdssegmentet til avspilleren. Det er avspiller styrt og serveren og nettverket belastet kun med trafikk av de etterspurte segmentene. Dette bringer oss videre til et viktig aspekt, ettersom både avsender og mottaker må støtte protokollen, formatet og kodeken for at strømmen skal fungere. Dette er del av grunnen for at standardisering er viktig, ettersom hver enhet som skal kunne spille av må tilpasses til hvordan innholdet leveres. Det finnes noen større ulike protokoller som i all hovedsak er hva som benyttes og arbeides for å skape støtte for. De store forskjellene mellom disse ligger i hvordan segmentene blir funnet frem til, hvordan segmentene spilles av, og om teknologien er proprietære. Videre skal vi se på oppbygningen og hvordan de viktigste protokollene i dagens marked fungerer.

Dynamic Adaptive Streaming over HTTP

MPEG-DASH eller Dynamic Adaptive Streaming over HTTP er en av de største protokollene på markedet. Den kom først ut i 2012, men har senere blitt revidert og omfavnet av store deler av streaming markedet. Protokollen benytter standard HTTP web servere og reduserer slik både kostnad og teknisk kompleksitet i implementeringen hvis man sammenlikner med andre alternative metoder for streaming, spesielt der egne spesialiserte strømmeservere har måttet benyttes. MPEG-DASH er en ABR protokoll som tillater deteksjon av båndbredden i seerens internettilkobling og leverer dermed hele veien best mulig kvalitet på videoen til enhver tid. Dette reduserer også buffer og løfter brukeropplevelsen. MPEG-DASH er også en åpen standard som ikke er kontrollert av kun et selskap. (Krings, 2021)

MPEG-DASH er designet slik at den ikke avhenger av en spesiell kodek eller er kodekavhengig. Og støtter derfor en rekke av kodekene som er aktuelle i dag. Det benyttes Media Presentation description (MPD) som manifestfil for MPEG-DASH. Manifestfilen består av ulik informasjon om kodek benyttet, lyd filer, videofiler, bitrate, oppløsning, undertekster, ABR informasjon og også DRM informasjon. Media presentasjonen og media

presentasjonsbeskrivelsen gjør det mulig for avspilleren å finne frem korrekt rekkefølge og mediasegment, samt velge hvilken strømmekvalitet som bør benyttes ved ulike båndvidde slik. Media presentasjonsbeskrivelsen inneholder URL til alle mediasegmentene som ved MPEG-DASH er kodet som MPEG 2 Transport Stream i filtypen .ts. Disse segmentene leveres så over TCP til avspilleren, så det er kun HTTP og TCP som benyttes for overføringen av MPEG-DASH innhold. Strukturen på leveransen til avspillerenheten er illustrert i figur 3.



Figur 3 Arkitekturen bak MPEG-DASH

Mediapresentasjonsbeskrivelsen (MPD) med sine referanser er kjernen i funksjonaliteten hvor viktig informasjon fremstilles gjennom en hierarkisk struktur av typen xml. I filen finner man periods som er det øverste nivået av elementer i mpd-en, der en periode beskriver en del av innholdet med starttid og lengde som i figur 4. Det kan benyttes flere perioder for å beskrive scener, kapitler eller for å skille reklame fra programinnhold. (Long, 2015) Det er i hver periode igjen adaptation sets som beskriver tilgjengelig lyd og bilde for perioden. All lyd og bilde for perioden kan legges i et adaptation set, men for å redusere båndbredde kan også strømmene deles i ulike adaptation sets, der lyd kan finnes for flere språk og undertekster. (Long, 2015)

```

<MPD>

  <Period id= "0" duration="PT59.52s">

    <AdaptationSet mimeType="video/mp4" codecs="avc1.4D401F">

      <BaseURL>video/</BaseURL>

      <Representation id="1080p" bandwidth="3200000">

        <BaseURL>1080p/</BaseURL>

        <SegmentList>

          <SegmentURL media="segment-1.ts"/>

          <SegmentURL media="segment-2.ts"/>

        </SegmentList>

      </Representation>

    <AdaptationSet>

  </Period>

</MPD>

```

Figur 4 Eksempel på strukturen i en MPEG-DASH mediapresentasjonsbeskrivelse

Representations lar så det samme innholdet i et adaptation set bli tilgjengelig kodet på ulik måte, og det er her ulike skjermstørrelser og båndbredder har tilgang på ulik størrelse på innholdet. Ulike kodeker kan tilbys til ulike enheter ut fra støtte for å dekode de ulike, eller ha høyere bitrate og bildekvalitet til de enheter som støtter dette når båndbredden er god nok. Det kan også være lurt med ulike kodeker slik at enheter kan spare strøm på å spille av eldre kodeker med hardware støtte, heller enn å spille av den nyere kodeken som må dekodes i software på enheten. Representasjonen velges vanligvis automatisk av avspiller, men kan i noen tilfeller velges preferanser på fra bruker, som for eksempel låse til høyere oppløsning selv om det eventuelt må stoppe opp for å bufre. (Long, 2015)

BaseURL i figur 4 beskriver så lokasjonen til de faktiske mediefilene som avspillerklienten vanligvis spiller av segmentene en etter en. Representasjon kan gjøre det mer avansert enn som så med mulighet for å hente ulike filer til ulike enheter og båndbredder som tillater at det skiftes mellom disse. Segmentene har en starttid og varighet som kan beskrives både tidligere i MPD-en eller for hvert segment gjennom SegmentTimeline. Og hvert segment kan være en

egen fil eller de kan være en byterange innenfor en fil. Avspilleren benytter så MPD-en for å finne frem til, etterspørre og spille av de korrekte segmentene for klienten. (Garrison, 2021)

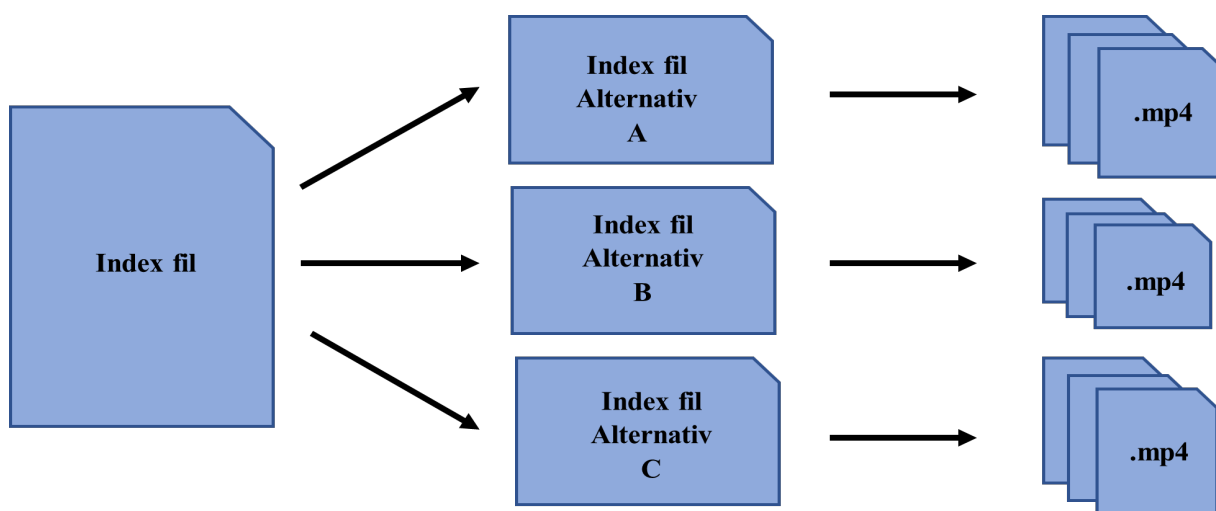
Totalt sett er MPEG-DASH en effektiv og fleksibel protokoll som gjennom HTTP leverer til alle enheter som støtter MPEG-DASH, og kan levere en bred kombinasjon av multimedieinnhold. Den aller største styrken er i at det er utviklet bred støtte for protokollen i dagens avspillerenheter, og at det benyttes HTTP servere i leveranse av innholdet, heller enn de dyrere dedikerte streaming-serverne.

HTTP Live Streaming

HLS eller HTTP Live Streaming er Apple sin streaming protokoll som ble lansert sommeren 2009 og er i dag også en av de mest benyttede protokollene i streaming. Protokollen er under stadig utvikling ettersom Apple regelmessig legger til funksjonalitet. (Krings, 2021)

Protokollen kan levere både bilde og lydinnhold til en bruker over nett til et vidt antall ulike enheter, og er av den første protokollen som har fokus på og kom for å løse problemet med varierende båndbredde frem til brukeren og tillate at enheten også kunne bytte nettverk underveis. Den kan nytte seg av ordinære webservere og CDN for kostnadseffektiv leveranse. (Apple, u.å²)

HLS protokollen benytter også et manifest for at avspiller skal finne frem til korrekt korrekte videosegmenter å hente. Det er mulig å benytte manifestet som en multivariant spilleliste, som det kalles i HLS. Illustrert i figur 5, der benyttes det første manifestet som en oversikt over manifestene til spillelister av ulike varianter av det samme innholdet. (Apple, u.å.¹) Det er derfor avspilleren sin oppgave å holde styr på endringer i båndbredde og skulle endringene at man bør hoppe på en lavere kvalitet strøm, så finner avspiller frem i hovedmanifest til korrekt båndbredde manifest, og så neste videosegment med lavere bitrate. Det hele skal kunne gå til ubemerket for bruker.



Figur 5 HLS Manifest hierarki

Manifestfilen brukt i HLS er av typen EXTM3U, som er en extended m3u fil, og det lagres som regel som en m3u8 fil. Filen er en tekstfil ofte referert til som en spilleliste hvor det er definert ulike tagger for å fremstille innholdet. Det er en EXT-X-STREAM-INF-tag som benyttes i spillelisten for å indikere at det neste i filen er en URL til en annen spilleliste. Som en del av tag-en er det et krav å oppgi parameteren «bandwidth» som er den øvre grenseverdien for total bitrate for mediefilene per sekund som illustrert i figur 6. For spillelister for video trenger man parameteren «resolution» der videoinnholdets oppløsning oppgis, og «codec» der alle kodeker som er nødvendig for å dekode strømmen oppgis som en liste skriver Apple (u.å.¹) i sin dokumentasjon.

```
#EXTM3U

#EXT-X-STREAM-INF: BANDWIDTH=150000, RESOLUTION=416x234, CODECS="avc1.42e00a, mp4a.40.2"
http://example.com/low/index.m3u8

#EXT-X-STREAM-INF: BANDWIDTH=640000, RESOLUTION=416x234, CODECS="avc1.42e00a, mp4a.40.2"
http://example.com/high/index.m3u8
```

Figur 6 Eksempel på HLS manifestfil som refererer til en lav- og høykvalitets spilleliste

Selve spillelisten som refererer til innholdet er strukturert på samme måte, og hvor alle segmentene som trengs for å få et komplett innholds strøm vil ligge tilgjengelig. Filtypen og formatet for hvert segment må være segmenter av fMP4 format eller MPEG2 Transport Stream med kodek AVC/H.264 eller HEVC/H.265. (Apple, u.å.³) Dette formatet støtter også innebygde undertekster. Videosegmentene henvises til fra manifestet eller spillelisten gjennom en EXTINF-tag som trenger at lengde attributtet defineres med antallet sekunder varigheten av segmentet er, etterfulgt av URL-sti til segmentet enten som en full URL eller en

relativ sti til filen fra manifestet som i figur 7. Det er viktig at det fremkommer i manifestet om det er en live-stream manifest, eller manifest med en gitt lengde. Dette defineres av taggen EXT-X-PLAYLIST-TYPE, der det enten kan defineres som «event» eller «VoD». Spillelisten avsluttes etter taggen EXT-X-ENDLIST som er forventet i manifestet om det ikke er live.

```
#EXTM3U
#EXT-X-PLAYLIST-TYPE:VOD
#EXTINF:10.0,
fileSequenceA.ts
#EXTINF:10.0,
fileSequenceB.ts
#EXTINF:9.0,
fileSequenceC.ts
#EXT-X-ENDLIST
```

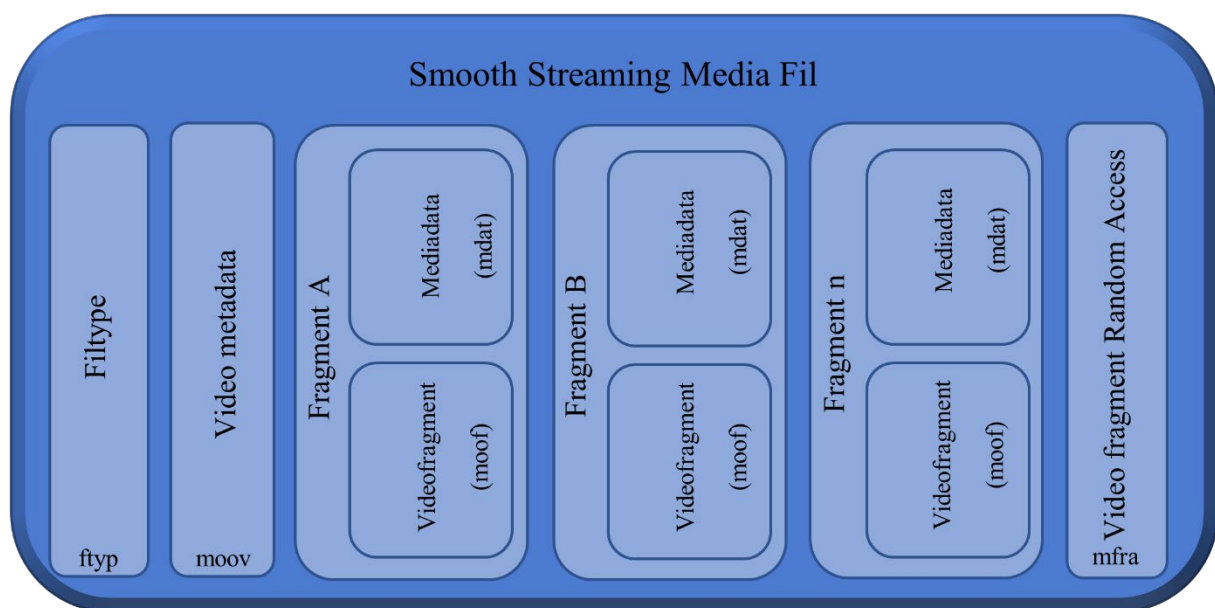
Figur 7 Eksempel på HLS spilleliste for VoD innhold

HLS støtter gjennom dette ikke innebygd DRM, men det finnes en del løsninger der ute hvor det utnyttes muligheten for å kryptere data og benyttes autentifisering gjennom HTTPS for å sikre innholdet. Det er også bygget flere større tredjeparts DRM løsninger til HLS.

Microsoft Smooth Streaming

MSS eller Microsoft Smooth Streaming er utviklet av Microsoft i 2008 for å møte de behovene som fantes i tidlig adaptiv bitrate streaming. Den er kjent for å være både kostnadseffektiv og optimert for å være effektiv og redusere buffering. Det kreves Silverlight som er en Microsoft proprietært rammeverk teknologi som gir god kontroll over avspillingen og som bidrar til MSS sin støtte for DRM, (Krings, 2021) Smooth Streaming skal kreve lite av sluttbrukers båndbredde og datakraft, men sluttbruker må ha installert Silverlight for å kunne motta medieinnholdet fra serveren. Smooth streaming nyttiggjør seg i likhet med de to andre nevnte HTTP streaming protokollene av ordinære CDN løsninger for kostnads og effektiv distribusjon av innholdet, men man er avhengig av Microsoft sin streaming programvare på serveren som distribuerer mediesegmentene til MSS. Og CDNene må bruke Microsoft-produkter, som begrenser enkelheten noe.

MSS fungerer noe ulikt fra MPEG-DASH og HLS ved at det finnes en server manifestfil i formatet ISM på XML form, og det er i tillegg en klient manifestfil ISMC. Klientmanifestfilen inneholder en oversikt over de ulike representasjonene og segment beskrivelser som tilhører videostrømmen. Den inneholder all informasjon som trengs for å forespørre og tolke innkommende data som kodek, bitrate og oppløsning som tilhører hvert enkelt segment av det ulike tilgjengelige innholdet. Servermanifestfilen inneholder oversikt over og linker til relasjonene mellom ulike representasjoner og segmenter. Der er også oversikten over filnavn og hvor hvert fragment finnes er lagret på serveren. Servermanifestet brukes til å oversette HTTP forespørselen til mediefiler, og videreformidler fragmentene til klienten på forespørsel. På serveren er hele representasjoner lagret i samme MP4 fil i ISMV formatet. Filen er satt sammen av selvstendige fragmenter som er hva som finnes frem til og sendes til klienten som i figur 8. Hver representasjon har derfor videofiler lagret i ulike bitrate som servermanifestet kan henvise til etter hva som blir forespurt fra klienten. (Mueller, 2015)



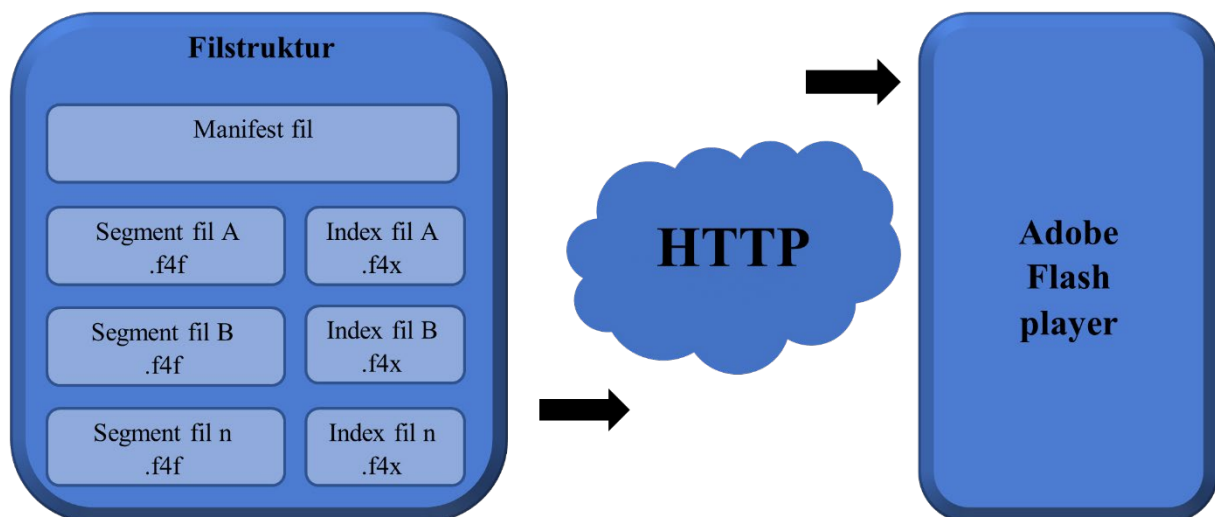
Figur 8 Strukturen i Smooth Streaming

Det finnes i dag mange avspillere og tjenester som har innebygd støtte for Silverlight, men de største nettleserne har ikke lengre mulighet for Silverlight. Det er i skrivende stund både synkende støtte og bruk av MSS, og Microsoft har i all hovedsak kuttet supporten til Silverlight løsningen som trengs til avspilling høsten 2021. (Microsoft, u.å.) Dette har ført med seg at denne teknologien i hovedsak brukes der Smooth Streaming allerede er implementert, og det ikke lengre utvikles nye løsninger hvor det implementeres MSS. Annet enn støtteløsninger for å opprettholde støtte til de eksisterende løsningene som fortsatt

nyttiggjør seg MSS. Mange store aktører har fortsatt dette som en produkt, spesielt under livestreaming. Produktet er fortsatt noe brukt i dag, og vil være det i flere år fremover også og var derfor naturlig å ta med noe kortfattet i denne sammenhengen.

HTTP Dynamic Streaming og Real Time Messaging Protocol

HTTP Dynamic Streaming eller HDS er Adobe sin måte å levere en strøm av bilder til Adobe sine spillere. De benytter i denne protokollen fragmentert MP4 segmenter med video, og disse segmentene er originalt pakket i F4F format. Arkitekturen er illustrert i figur 9. Fra segmentene identifiseres og sendes bare fragmentene som trengs, og det benyttes URL for å identifisere hvilke fragmenter som skal sendes fra indeksene. Manifestfilen brukt i HDS er av typen F4M som er et format for å beskrive komplekse media hvor det kan være både flere lydspor, bitrates og som innholdsbeskyttelse. (fileinfo.com, u.å) Også i HDS er det mulighet for å beskytte innholdet og kryptere og benytte autentisering for å hente innholdet til avspilleren, gjennom Adobe protection. HDS fungerer i stor grad i likhet med de andre nevnte protokollene, men differensierer seg ved at det må benyttes Adobe sin avspiller for å kunne spille av innholdet. Dette vil si Adobe Flash som ikke lengre er støttet, og derfor benyttes også HDS i mye mindre grad i dag enn tidligere. (Swani, u.å.)



Figur 9 Arkitekturen bak HDS

RTMP eller Real time Messaging Protocol ble i utgangspunktet utviklet for å levere video til Adobe Flash Player, men dette bruksområdet er blitt mye utdatert med avviklingen av Adobe Flash. RTMP benyttes fortsatt mye i dag men da for å transportere datastrøm fra encoderen og hvertfall til webserver, men den kan også brukes til transport helt til videospiller. Det finnes mange rimelige og også open source encodere som støtter RTMP, og protokollen er veldig

raskt, noe som gjør det til en veldig aktuell protokoll innen streaming fortsatt i dag. (Krings, 2021) Det er allikevel verdt å merke seg at protokollen ikke benytter leveransesystemet som er designet for å levere til Flash video player, men at denne funksjonaliteten er en del av den samme protokollen. I tillegg til standardutgaven av RTMP, finner også flere variasjoner av protokollen som benyttes til litt differensierte oppgaver innen livestreaming.

Secure Reliable Transport

SRT eller Secure Reliable Transport er en relativt ny streaming protokoll som er open source. Navnet kommer av nettopp hva de arbeider mot å oppnå med protokollen, å tilby både sikkerhet og lav forsinkelse i protokollen. Ulempen i protokollen ligger for øyeblikket i kompatibilitet, noe som ikke står på protokollen i seg selv, men heller på både hard og mykvarer som ennå blitt utviklet for å støtte protokollen. Haivision som har utviklet protokollen har startet SRT alliansen bestående av en gruppe selskaper som er dedikert til å arbeide frem SRT innen streaming, og for øyeblikket er derfor å benytte produkter fra selskaper i alliansen beste fremgangsmåte for å benytte SRT: Det er også viktig å merke seg at SRT er media og innhold agnostisk, som vil si at alle video og lyd kodeker og pakkeformater er støttet av SRT. (Krings, 2021) Det er verdt å nevne at denne protokollen for øyeblikket i motsetning til de andre protokollene per i dag i liten grad sikter på å levere innhold ut til sluttbruker. Dette er en protokoll som i dag i hovedsak har støtte for punkt-til-punkt overføring og benyttes i liveproduksjoner der man skal sende data fra for eksempel en livebuss til et studio. Den benytter allikevel mye av de samme funksjonene som adaptiv bitrate og Transport Stream for å frakte medieinnholdet i nettverk. (Haivision, 2020)



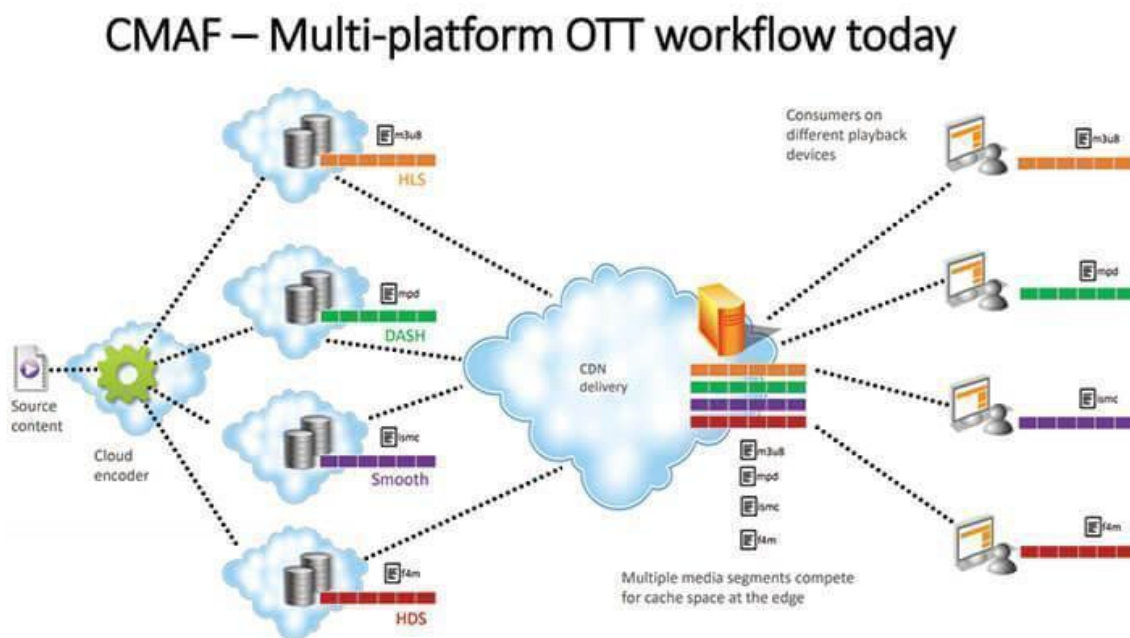
Figur 10 Illustrasjon av SRT i bruk (Haivision, 2020)

Denne protokollen sin bruksområde i dag blir i utgangspunktet utenfor omfanget av denne oppgaven, men det er allikevel valgt å ta med da denne dukker opp som en mulig fremtidig kandidat til å dekke flere aktuelle bruksområder. TCP som er brukt i de tidligere nevnte protokollene gjør at man kan sikre innholdet og tilpasse bitrate til hver enkelt mottaker. Selv om SRT har sikkerheten og kan tilpasse innholdets bitrate, gjøres dette i hovedsak med tanke på å sende ut informasjon. Det går ikke kommunikasjon tilbake fra hver enkelt mottaker av

pakker om hvilken bitrate som er ønskelig, og gjennom sin nyttiggjøring av UDP er dette et lite tenkelig bruksområde. Man vil mest sannsynlig være avhengig av utstyr som kan motta innholdet og at det tilrettelegges for at korrekt port er åpen i brannmuren, samt en gitt minimums bitrate for å tillatte mottak av disse pakkene. Det vil derfor være mest aktuelt for distributører og tjenesteleverandører der de kan ha kontroll på leveransen tilnærmet helt ut mot sluttbrukeren, og mulig som et alternativ eller supplement til IPTV i første omgang.

Common Media Application Format

Common Media Application Format eller CMAF er et nytt format som er ment å simplifisere leveransen av HTTP-basert strømming. Det er et forsøk fra bransjen om å nå en felles standard format for å simplifisere leveransen av videoinnhold. Overordnet fungerer CMAF slik at man benytter og lenker til de samme medieinnholdssegmentene i alle protokollene, og at det utarbeides et manifest for hver protokoll som er tilpasset CMAF-strukturen på medieinnholdet. Dette gjør det mulig for avspillere å hente den typen manifest som passer ønsket avspillingsenhet, men samtidig for CDN-et å holde et sett med medieinnholdssegmenter i ulik kvalitet som alle manifestene linker til, i stedet for et individuelt sett for hver protokoll som skal støttes.



Figur 11 Illustrerer arbeidsflyten for OTT i dag (Demir, 2020)

Figuren viser dagens situasjon illustrert ved benyttelse av fire store streaming formater, og hvordan arbeidsflyten er i dag. For å kunne støtte absolutt flest mulig enheter for avspilling, er man avhengig av at innholdet leveres i ulike formater. Figuren viser hvordan et kildemateriale

i form av en film som kodes gjennom ulike kodeker og tilpasses til de ulike strømmeformatene. Her holder et CDN fire forskjellige strømmeformater av nøyaktig den samme kilden, kun for at flest mulig skal kunne ha støtte for avspilling av innholdet.

CMAF er i motsetning til Dynamic Adaptive Streaming over HTTP (DASH) og HTTP Live Stream (HLS) ikke et presentasjonsformat. CMAF er et containerformat som støttes av de tidligere nevnte presentasjonsformatene DASH, HLS, HDS og MSS. Når man benytter CMAF vil man få en spilleliste tilpasset hver av de ønskede presentasjonsformatene, og et felles sett med video og lydfiler som alle spillelistene uavhengig av format linker til. (Ozer, 2020) Dette gjør CMAF unikt ettersom man kan spare koding av innholdet til hvert enkelt format, man sparer server og CDN kapasitet på å slippe å holde ulike formater, og man beholder støtten for de fleste avspillere. Teoretisk sett skal man kunne kutte koding og lagringskostnaden med syttifem prosent og gjøre cachingen av filene mer effektiv.

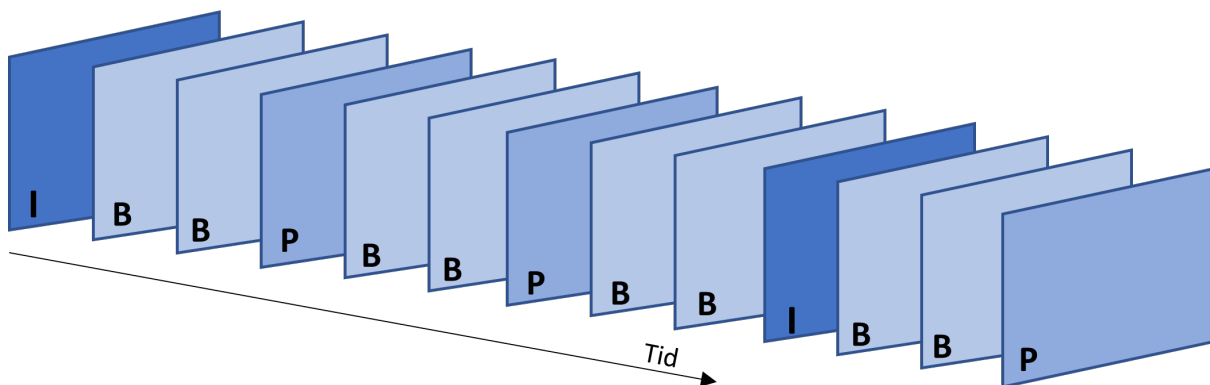
Ulempen for CMAF ligger i at det for øyeblikket i stor grad fungerer som enda et format man må forholde seg til og som krever støtte. For selv om de aller fleste nye enheter støtter å få innholdet levert gjennom HLS eller DASH i en CMAF-fil, så gjør ofte ikke eldre enheter dette. Det vil si at i en god stund også fremover, må bransjen leve med enda ett format. Den store fordelen som ligger i å benytte dette formatet ligger i at det er fremtidstenkende og dannes en støtte rundt å få samlet bransjen under et format. De nye enhetene som støtter og bruker CMAF har en stor gevinst i at filene i DNS-et lagres lett tilgjengelig, ettersom de blir så hyppig etterspurt fra avspillere som etterspør de cachede CMAF filene.

Kodek

Kodek er ulike metoder som benyttes for å komprimere og fremstille multimedieinnhold på. Kodeken må pakkes inn i en container, som er rammeverket som benyttes og man koder filen i for korrekt fremstilling igjen ved avspilling. Ved avspilling av en lokal fil der alle data er tilgjengelig for avspilleren, kan det benyttes kodek som både komprimerer lite slik at bildekvaliteten blir god, og en container som baserer seg på at man har tilgjengelig en hel fil. Store data kan overføres internt i datamaskinen hurtig for avspilling. Det ville heller ikke være nødvendig å dele opp filen i segmenter slik som ved avspilling over nett, hvor dataen tar eksponentielt mye lengre tid å hente.

Både kodek og container er viktige deler av streaming prosessen for at man skal kunne oppnå et godt sluttresultat. I dag benyttes i overvekt noen utvalgte kodeker i streaming. De siste årene har AVC/H.264 vært den ledende med å bli benyttet av over 90% av de spurte i Bitmovin sin rapport. (2021) Selv om denne fortsatt er rangert i 2021 som den mest benyttede, er HEVC/H.265 på fremmarsj sett i forhold til hvor mange som benytter den. AVC/H.264 har en nedgang i respondenter som benytter denne, og færre har planer for å benytte den videre inn i 2022. I 2022 anslår Bitmovin sin rapport at det er HEVC, AV1 og VVC som vil ha mest økning.

I overordnede trekk vil en kodek og komprimeringen av videoinnhold arbeide med hva de kaller en gruppe med bilder (GoP) som illustrert i figur 12. Etersom video kun er et visst antall bilder per sekund, kodes disse gjennom en felles kodek og lagres i en container samlet. Derfor referer vi til et bilde av en videofil som en ramme, og for å spare plass og overføringskapasitet passer man også på å i minst mulig grad lagre duplikat informasjon om rammene, men heller komprimerer informasjonen.



Figur 12 Illustrasjon av hvordan en GoP kan se ut

I en GoP er den viktigste rammen en Intracoded frame, eller I-ramme. Dette er basisrammen som inneholder fullstendig bildeinformasjon. Denne er den minst komprimerbare rammen som holder på mest informasjon av rammene i en GoP, og er derfor den mest fyldige rammen også i figur 12. Videre finnes Predicted frame, eller P-rammen. Den inneholder kun endringene i en ramme fra den forrige rammen, og er mer komprimerbare. Og her imellom kan man også skape bidirectional frame, eller B-rammer. Dette er rammer som ser både til rammen fremover, og bakover for å fjerne eventuelle duplikat data. Dette gjør det til den aller mest komprimerbare rammen av de tre typene. (Hong, 2018, s.190) Dette er utgangspunktet for hvordan komprimeringen skjer i kodek, men med ulikt utgangspunkt for å beregne og kode innholdet. Videre skal jeg gå gjennom de viktigste kodekene i tilknytning til streaming.

Advanced Video Coding

Den aller mest brukte kodeken og videokompresjonsteknologien som benyttes i dag er Advanced Video Coding(AVC) eller H.264 som den er best kjent som. Kodeken ble utviklet som en internasjonal standard, og er inkludert i flere container formater som MPEG-4, QuickTime, Flash og MPEG transport Stream. Standarden har blitt tatt i bruk og implementer av flere leverandører og er kanskje det nærmeste vi har en universal kodek. Apple codec bruker denne, det samme gjør MainConcept codec og x264 codec. Store selskaper som Adobe og Microsoft benytter lisenser fra MainConcept, mens Apple bruker den i sine kjente produkter som Apple Compressor og QuickTime Pro. Det er stor konfigurerbarhet i standarden, og det gir liten uniformitet i kvaliteten som leveres med kodeken. Men kodeken kan spilles av i tilnærmet alle nye nettlesere og mobile enheter i dag og kodeken har vært og er fortsatt i bruk av store nettsteder som Youtube og vimeo. Det er nettopp dens allsidige avspillingsstøtte kombinert med god videokvalitet som har gjort den til den mest benyttede kodeken. (Ozer, 2011)

Det finnes patenter og royalties tilknyttet kodeken selv om den er en åpen standard. Under utviklingen har ulike selskaper delt av sine patenter for å få på plass standarden. Og det finnes derfor en royalty pool som følger opp og skal fordele royalties. I utgangspunktet er det ingen royalties på gratis distribusjon av video over internett, men ved benyttelse av store selskaper i sin distribusjon av innhold eller leveranse av andre tjenester som involverer kodeken skaper det inntekt for selskapene som bidro i utviklingen. I 2022 er første året kodeken er forventet å starte å miste oppslutning selv om den fortsatt er desidert størst. Men det er forventet at i løpet av dette tiåret så vil kodeken gå fra en oppslutning på nærmere 80% av markedet til kun 25%.

(Ozer, 2022) Det er derfor viktig å se på hvilke alternativer som har dukket opp og skaper denne forventningen for de kommende årene.

VP9

VP9 er en open-source kodek fra Google som stammer fra oppkjøpet av en mindre aktør som var kjent for sin videokompresjonsteknologi. VP9 er etterfølgeren til VP8 fra Google som har blitt sammenlignet og tester relativt likt med AVC kodeken. VP9 har i ytelse blitt sammenliknet med HEVC og AVC med ulikt resultat, men blir ytelsesmessig sammenliknet med HEVC som får tilnærmet likt resultat i kvalitet og filstørrelse. Google har donert VP8 og VP9 til Alliance for Open Media, og vil derfor ikke lenge komme med ny utvikling av kodeken selv. Men den får leve videre og er fritt tilgjengelig for bruk i både proprietær og open-source software. VP9 har støtte for avspilling i de fleste nyeste nettleserversjoner. AV1, som jeg kommer tilbake til, er bekreftet å være i stor grad basert på hva som skulle blitt VP10. (Ozer, 2016²)

Den største nedsiden er at Apple nektet allerede forgjengeren VP8 støtte for avspilling på iOS enheter og Safari, mens Microsoft ikke vil støtte avspilling i sin Internet Explorer. Tilgjengeligheten er viktig, og når standarden dukket opp på et senere tidspunkt enn vel etablerte AVC, så hadde kodeken liten sjanse å vokse. Ingen i streaming vil kunne adaptere en standard uten bred og pålitelig avspillingsstøtte for meningsfullt mange brukere, ettersom det er brukerne som er den drivende kraften i dette markedet. (Ozer, 2013)

High Efficiency Video Coding

High Efficiency Video Coding (HEVC) eller H.265 er en etterfølger av AVC/H.264 som også er utviklet gjennom samarbeid som en standard i Joint Collaborative Team in Video Coding. HEVC ble utviklet med mål om å i det minste doble kompresjons effektiviteten til AVC (Hong, 2018, s. 188 og Ozer, 2013) og skal støtte oppløsning helt opp til 8K (Ozer, 2013). HEVC skal også kunne redusere filstørrelsen på en full-HD innhold med hele 30 prosent, og det er derfor ikke vanskelig å tenke seg hvorfor dette er interessant blant streamingtilbyderne. Ettersom det er noe mer avansert kodek er det anslått at HEVC kan kreve så mye som ti ganger så mye datakapasitet ved koding. Og at det medfører opp til tre ganger så mye kompleksitet ved dekodning av innholdet.

Selv om HEVC er overlegen til AVC på flere områder, har og vil adaptasjonen ta noe tid. Antallet enheter som allerede støtter og har mulighet for avspilling av HEVC omgir oss

allerede. Forsinkelsen ligger heller i de signifikante investeringene som er gjort i AVC utstyr blant selskapene som tilbyr streaming. Kostnaden av oppgradering kan heller ikke rettfærdiggjøres ytterligere i forhold til kostnaden ved å bevare eksisterende chipsett som fortsatt bruker mindre tid på å kode og dekode AVC, sammenliknet med investering i nyere chipsett for HEVC. (Ozer, 2013)

I likhet med AVC er mange av teknologiene som ligger til grunn for HEVC patentert, og patent eierne vil ønske seg kompensasjon for bruk av deres teknologi. Det er derfor en del usikkerhet tilknyttet royalties i forbindelse med HEVC. Royalties på HEVC gjør Google sin VP9 til en større konkurrent, selv om den ikke har samme status som en felles standard. HEVC er allikevel i dag en av de største aktørene i HDR markedet. Men den har kun støtte i en liten andel av nettlesere, i all hovedsak i Apple produkter. VP9 har til sammenlikning støtte i tilnærmet alle nettlesere, og AV1 har støtte i de aller fleste nettlesere med unntak av Apple produkter. Mangelen på nettlese support hindrer allikevel ikke HEVC standarden som en premium produkt for distributørene hvor apper kan benyttes i leveransen av innholdet og tilbyr HEVC support. (Ozer, 2022)

AV1

AV1 er den første kodeken som kommer fra Alliance for Open Media. Den er skapt for å ta opp kampen med HEVC og har som mest distinkte karakter og fordel at den er uten royalties. Her er alle teknologiene som er tatt i bruk sjekket grundig i forhold til patenter og eierskap, og sikret at teknologien skal kunne være åpent tilgjengelig gjennom å få med alle større patentiere i alliansen som står bak kodeken. Del av grunnen for å ta opp kampen med HEVC er nettopp HEVC sin uklare royalty situasjon, og at man ikke skal havne i den igjen, men heller ha et åpent og tilgjengelig produkt. Flere av de som også har sine patenter i HEVC er nå også del av alliansen bak og støtter også AV1. Kodeken støtter 8K og har mulighet for HDR innhold for dypere farger akkurat som HEVC. (Ozer, 2018) Målet med AV1 har fra start vært å kunne konkurrere på de fleste større streamingområdene som er tilgjengelig, slik som i nettlese, på mobil, OTT, smart-Tver og set-top bokser. I tillegg skal kodeken prestere bedre og gi opptil 40% bedre kvalitet for 4K video sammenliknet med VP9 og HEVC. (Ozer, 2016¹)

Versatile Video Codec

Versatile Video Codec (VVC) eller H.266 er etterfølgeren til HEVC/.H.265 og AVC/.H.264, men er skapt for å støtte utover HEVC. Dette er også gjenspeilet i dens navngivning hvor versatile viser til allsidigheten og støtten formatet skal ha fra lav bit rate og oppløsning, til høy

bit rate og oppløsning. VVC skal igjen også bidra til lavere filstørrelse på videostrømmen gjennom mer effektiv komprimering av bitrate, men krever sin datakraft og hardvare support før det kan bli bredt adaptert. Det er snakk om opp mot 30% forbedring i ytelse og opp til 50% bitrate reduksjon sammenliknet med forgjengeren HEVC. VVC er royalty basert ettersom den i likhet med sine forgjengere baserer seg på flere selskapers patenterte teknologier. (Ozer, 2019²)

Utviklingen til streaming

Netflix har vært en ledende aktør innen å tilby streaming av film og tv-innhold, og har under de siste årene vokst til en all-time-high mens korona pandemien har lagt begrensninger på hvor og hva personer verden over har hatt mulighet for å dra og gjøre. I 2020 hadde Netflix hele 530 millioner abonnenter, noe som gjør de til den desidert største aktøren blant de største streamingtjenestene. (Kats, 2022) Netflix har også banet vei og stått fremst i utviklingen av streaming, og det kan derfor være greit å kjenne utviklingen til selskapet da den på mange måter representerer også i grove trekk den generelle utviklingen til strømmetjenestene som største aktør.

Netflix startet som et videoutleieselskap som hadde over hundre tusen titler i sitt sortiment og opptil femti distribusjonssentre med lokasjoner over hele USA hvor de sendte ut DVD-er til sine kunder. Det ble sendt ut rundt tolv millioner DVD-er hver uke. (Vorel, 2021) I 2007 startet Netflix å tilby streaming til sine kunder gjennom Internet Explorer sin nye Windows Media Player plugin ActiveX. Siden den gang har både leveranse metoden og formatene vært gjennom mange endringer. (Ronca, 2013)

Tidlig strømming

Den første strømmen hadde profilnavnet CE1 som var en multiplexed stream der bilde og lyd ble mikset sammen i en og samme fil, som ble sendt og avspilt gjennom ActiveX i Internet Explorer. Containeren som ble benyttet for video og lydinnholdet var ASF-fil som vil si filtypen wmv som er organisert i en header objekt og data objekter. ASF er organisert i seksjoner og er laget for å kunne sendes over internett, og at det er designet slik at det også skal kunne spille av selv om noen pakker i filen gikk tapt eller ikke var lastet ned. ASF lastet ned et header-objekt med oversikt over datainnholdet og så minst et dataobjekt for å kunne spille av. (Microsoft, 2020) Kanskje ikke spesielt ulikt hvordan man tenker når man i dag sender segmenter av video i en jevn videostrøm til sluttbrukeren. Men på dette stadiet er medieinnholdet fortsatt en stor fil man sender i deler til avspilleren. Og det antas at hele filen skal sendes til mottakeren selv om man avbryter avspillingen på klientsiden. Dette var lite effektivt og er del av utfordringene som senere har blitt løst med å ha en mer aktiv avspiller. Og der avspiller heller ikke ber om å bli tilsendt segmenter etter at økten i realiteten er avsluttet av brukeren eller om økten ble brutt.

Neste steg for Netflix var å utvide til å både støtte Mac og PC gjennom Silverlight Player. Denne Silverlight profilen gav de første ordentlige mulighetene for adaptiv strømming. Den

benyttet separate ASF filer for lyd og bilde, og tok i bruk muligheten for en index fil per stream i header objektet, der det blir beskrevet hvert segment i ASF-filen. Dette tillot Netflix å begynne med variabel bitrate for å håndtere buffer på video, mens lyd som er mye mindre data bare blir valgt en bitrate av basert på båndbredde ved oppstart for å simplifisere noe. (Ronca, 2013) Dette blir den første muliggjøringen av å sende data med ulik bitrate til avspilleren, og hva som nærmer seg det vi kjenner som adaptiv bitrate i dag.

I omtrent samme periode hadde Apple sin egen konkurrerende løsning til Microsoft sin, der det åpnet for muligheten å strøme innhold direkte til sine enheter gjennom QuickTime. De bygde en robust tjeneste og server til klient struktur, men tjenesten brukte ikke standardporter på dataoverføringen gjennom Real Time Streaming Protokollen. Dette resulterte i at brannmurer ofte blokkerte innholdet. Dette problemet kombinert med gjennomsnittlig lave internetthastigheter hos forbrukerne, gjorde at produktet aldri nådde de store massene. (Wilbert, 2022) Det samme problemet kan ikke sies å gjelde Silverlight og Flash.

Plugins

Netflix sin CE2 profil benytter så Adobe Flash plugin og leverer i likhet med Silverlight separat lyd og bilde gjennom ASF. Noe av det nye med dette var også hvordan nettsiden benyttet Flash til et grafisk grensesnitt med plakater og avspillingsmulighetene i Flash. Flash ble mye brukt på nett tidligere for å lage nettopp pene grafiske brukergrensesnitt, og avspilling av videoinnhold. På et tidspunkt var Flash hovedteknologien for å levere video over internett. (Wilbert, 2022) CE1, CE2 og Silverlight har til felles at de har brukt ASF og Microsoft sin VC-1 videokodek, men selv Microsoft har gått bort fra VC-1 kodeken. Kodeken var en gang ansett som et alternativ til AVC/H.264, men som ble valgt bort til fordel for AVC når de etter hvert beveget seg videre til fragmentert MP4 i MSS. Og når man så hvor stor markedsandel AVC ble sittende på. (Streaming Media, u.å.) I dag er alle disse profilene avvirket av Netflix, og ikke lengre i bruk. Det samme gjelder også avspilleren ettersom Flash ikke lengre er støttet i nettlesere, og HTML5 har overtatt og gir bred støtte i de aller fleste nettlesere for grafiske nettsider.

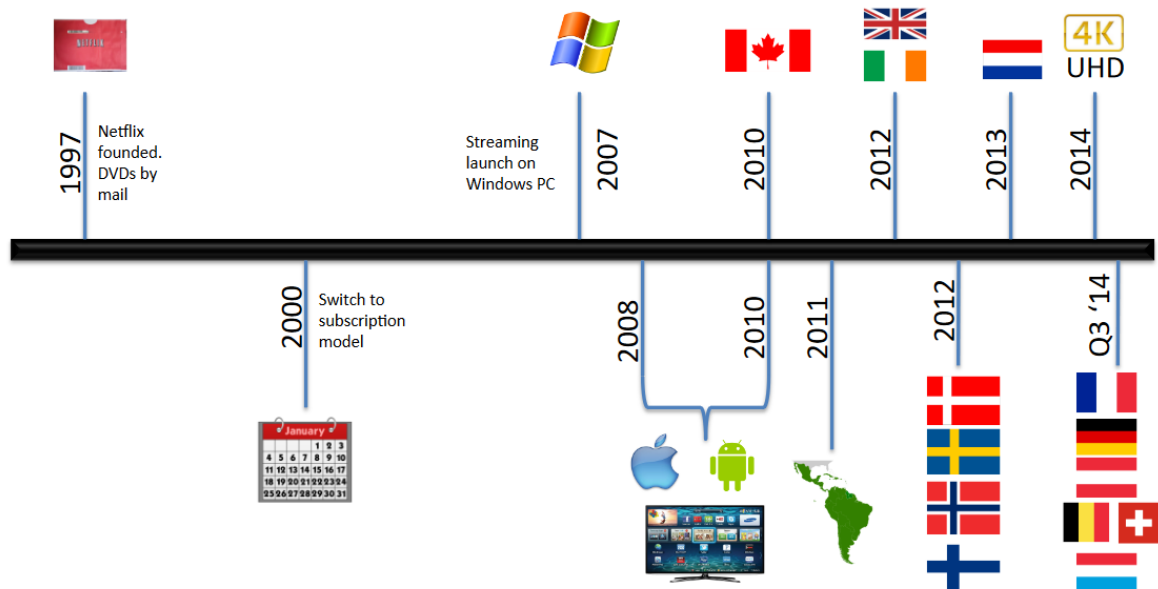
Netflix gikk ut allerede i 2013 og annonserte at de også ville gå bort fra bruk av Silverlight som skulle kun skulle få støtte av Microsoft frem til 2021. (Albanesius, 2013) I dag benyttes avspillere bygget rett inn i nettsidene med HTML5, og også avspillere på andre apparater som støtter standarden. Prosessen for å gå over til HTML5 løsningen skjøt virkelig fart for flere leverandører i 2015 etter at Google gjorde kjent sine planer om gradvis deaktivering av

plugins som Silverlight og Flash i sin nettleser Chrome. Også TV2 Sumo flyttes derfor over til HTML5 videoavspiller og MPEG-DASH format. Men litt av grunnen for at det ikke har kunnet skje tidligere for TV2 sier de at skyldes strenge krav til kryptering og ivaretagelse av strenge rettighets retningslinjer. Det har også vært usikkert om teknologien var moden nok også for bruk på live-innhold. (Brombach, 2015)

Felles format

Etter å ha gått fra mange ulike formater og profiler som hver seg kun støtter noen få enkeltavspillere, lanserer Netflix Profilen CE3-DASH der de har en bred støtte for avspilling på mange ulike enheter gjennom DASH. Allerede i sent 2009 gjorde katalogen Netflix hadde tilgjengelig i de ulike profilene for avspilling på ulike enheter omtrent 25-50TB per profil. (Ronca, 2013) Det var derfor naturlig å søke mot en felles profil med bredere støtte for avspilling. Den nye profilen benytter AVC/H.264 kodek og innhold i FMP4 format som støttes av de fleste avspillerplattformer. Omtrent samtidig med den første utviklingen, inngår Netflix i 2010 et samarbeid med Apple med utvikling og tilbud av Netflix iPad og iPhone enheter. Her må det tilpasses og benyttes Apples egen HLS for å kunne spille av. Netflix sin iOS profil blir i utgangspunktet gjort tilpasninger av tidligere profiler til å kunne levere til HLS, da den første lanseringen ble gjennomført på i underkant av et par måneder. (Ronca, 2013) Men profilen blir forbedret og går fra å bruke multiplex video og lyd, til at det deles opp og kan adresseres individuelt fra manifestet og det benyttes AVC og M2TS format.

CE4-DASH er Netflix første profil med støtte for streaming i 4K/UHD format, og utnytter de nyere kodekene HEVC/H.265 og VP9 som kan levere UHD til de enheter som støtter dekode av kodeken og har 4K/UHD støtte. Ellers bruker profilen i likhet med forgjengeren FMP4 container for medieinnholdet.



Figur 13 Netflix tidslinje (Aaron, 2014)

I 2020 tok Netflix i bruk den nyere AV1 kodeken i sin stream for å levere innhold til en del enheter, og listen over kompatible enheter som støtter AV1 dekodning utvides til stadighet. AV1 krever som kjent moderne kompatible prosessorer for å kunne dekode videostrømmen som er av bedre kvalitet til samme bitrate sammenliknet med andre formater. (Peters, 2021) Med denne kodeken åpnes det også for mer avanserte bilde og lydprofiler med blant annet HDR og utvidede fargespekter blir komprimert inn i kodeken og tilbys til de apparater som støtter dette for å spille av en ekstraordinært god kvalitet på strømmen.

Norden fikk tilgjengelig Netflix i 2012, og siden den gang har den amerikanske strømmegiganten fått fotfeste. Ti år senere kjenner de aller fleste til eller har et forhold til Netflix. Men vi har også flere andre internasjonale aktører vi har blitt kjent med, og nasjonale aktører som TV2 Play som tok i bruk MPEG-DASH, og NRK TV som benytter både HLS og MPEG-DASH.

Hvor leder utviklingen?

Det går mot en bedre og mer effektiv komprimering og mer avanserte kodeker som muliggjør å sende bedre kvalitet i videostrøm enn noen gang tidligere. Vi får bedre hastigheter på nettet, og det muliggjør og tillater avspillerne å laste ned høyere kvalitet data i forbindelse med strømming. Dette kombinert med mer effektive kodeker gjør at de høyeste datakvalitetene ofte kan spilles av hos de fleste, men det er fortsatt viktig å ha støtte for de kundene som sliter med for lav hastighet eller presset kapasitet på nettet hos seg. Dette har muliggjort også å

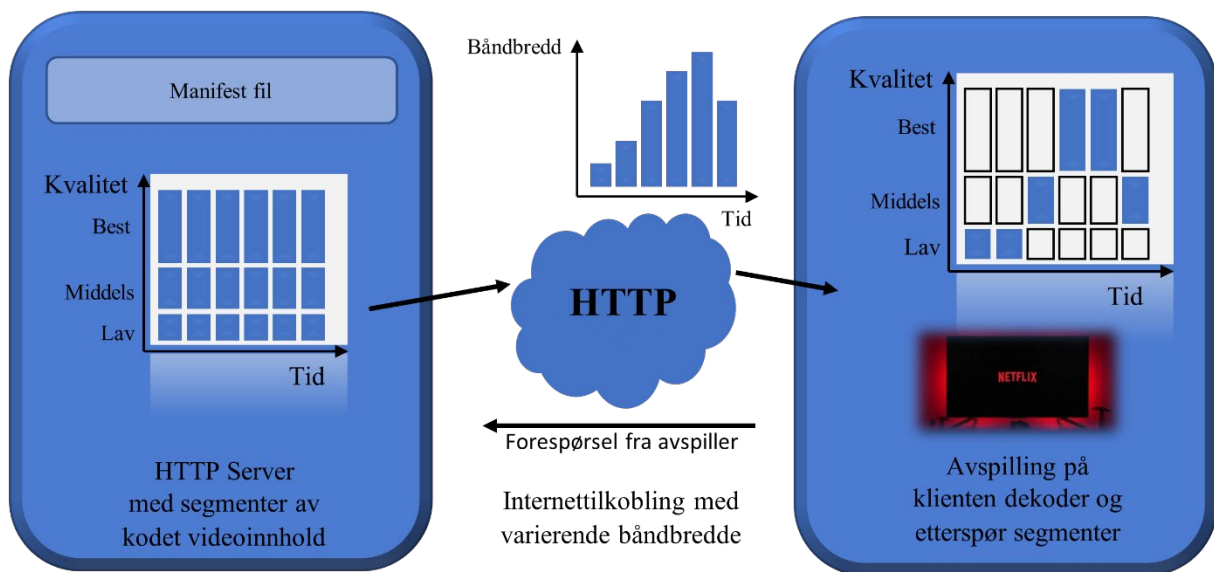
sende høyere kvalitet av innhold i piksler og bitrate, men også innhold av høyere kvalitet som for eksempel HDR-innhold som har mer avansert fargeinnhold som det som smått har dukket opp støtte for både i strømmeformatene, kodekene og på avspillingsenhetene.

Høy satsning og adaptasjon av strømmeteknologier i mange markeder rettferdiggjør også en satsning og investering i både infrastruktur og utvikling for mange selskaper. Teknologiene som i dag vokser mest, er nettopp de teknologiene i streaming som simplifisert leveransen av multimedieinnhold gjennom å benytte HTTP og standard webservere i sin leveranse av formatene. Dette både kostnads effektiviserer og gjør det mulig å utvide strømmetjenestene også gjennom eksisterende infrastruktur og med bred støtte av enheter. I stor grad har utviklingen de senere årene sluttet seg rundt å skape en standardisert leveransetype for streaming som allikevel er fleksibel nok til å støtte alles behov og egeninteresser.

Store selskaper med egne markedsinteresser har tidligere ledet utviklingen av streamingteknologiene. Microsoft med sine formater, Apple med sine, Adobe med sitt og Google med sin kodek. Det har lyktes for perioder, men det har også vært et hinder i utviklingen til teknologien at man må forholde seg til så mange ulike og proprietære teknologier der ingen klarer å skaffe seg full dekning og støtte ute i markedet. De største selskapene kan virke å ha gått noe bort fra å satse på å kunne kontrollere og sitte alene på rettighetene til teknologiene, men har heller samlet seg om å utvikle felles formater og standarder som kan brukes felles for å få støtte til et format ut til ideelt sett alle enheter.

Oppsummering

Gjennom arbeidet med oppgaven har det blitt klart hvordan streaming løses fra OTT tilbydere i dag når de satser på å støtte alle enhetene som brukerne ønsker. De mest aktuelle måtene å strøme på i dag er forsøkt illustrert gjennom en generell beskrivelse i figur 14. Det er tydelig at strømmeleverandørene sitt fokus har gått til å skulle støtte flest mulig enheter i en felles protokoll for å simplifisere prosessen, og ha et sett filer og leveringsmetode til alle enheter.



Figur 14 Illustrasjon av dataflyten i streaming i dag

Slik jeg har kartlagt prosessen vil en videofil kodes med en kodek og container i mindre segmenter og et manifest utarbeides for å holde oversikt over segmentene. Disse segmentene tilgjengeliggjøres så på en ordinær HTTP-webserver for å etterspørres av klienten, men også de mest etterspurte segmentene blir lagt i CDN i likhet med cachen på en datamaskin, slik at de er raskere tilgjengelig for forbrukeren og det er mindre risikoavstand for dårlig QoS. Protokollen, kodeken og containeren som benyttes er ofte av den mest brukte kombinasjonen MPEG-DASH med AVC/H.264 kodek i en MPEG-TS container. Eller mye bruke kombinasjonen HLS med AVC/H.264 kodek i en fMP4 container, men kan også være av andre nyere kodeker eller av andre protokoller både nyere og eldre. Avspilleren er i dag ofte basert på adaptiv bitrate og henter inn et manifest der de kan etterspørre segmenter i ulik bitrate basert på båndbredden til klientenheten, som illustrert i figur 14. Klientenheten skal kunne bytte nett, ha varierende nett kvalitet, men strømmen skal forhåpentligvis allikevel forbli kontinuerlig uten avbrudd og sluttbrukeren skal få en best mulig QoE.

Det var originalt i denne oppgaven ønskelig å se på og sammenlikne de største OTT strømmetjenestene vi har i Norge i dag, og kartlegge hvilke formater og kodeker som er brukt både tidligere og brukes i dag for hver tjeneste. Dette viste seg å være vanskeligere å finne informasjon på enn forventet, og det man har funnet av informasjon tilsier at det meste i dag baserer seg på nettopp de to strømmeprotokollene MPEG-DASH og HLS. Selv om man brukte en god del tid og ressurser på research av dette, var det forbløffende lite og tilnærmet ingen utdypende informasjon åpent tilgjengelig om hvilke tjenester som benyttet de ulike spesifikasjonene. Ikke engang hos NRK som har sin særegne konkurransesituasjon i forhold til de andre strømmetjenestene, virker de å dele noen teknisk innsikt på hvordan deres formater, kodeker og tekniske løsninger for innholdet i NRK TV er satt opp og deles. Det lille informasjon man finner stammer fra når tjenesten går vekk fra Flash i sin videoavspilling i 2016, og gjør det klart i en bloggpost at det er HLS og MPEG-DASH som blir etterfølgeren i NRK. (Gundersen, 2016) Og at selskapet senest i 2022 søker etter utviklere med kunnskap om både HLS og MPEG-DASH i sine utlysninger. De kommersielle norske aktørene er heller ikke noe mer delende om hvordan de har lagt opp sine strømmetjenester og gir lite innsyn. Mest tilgjengelig informasjon finnes fra nettopp Netflix som har skaffet seg en så særegen posisjon i markedet, at det ikke er noe problem å blogge om deres utvikling og arbeid med både formater og kodeker. Oppgaven er derfor også delvis formet ut fra hvilken informasjon som de ulike selskapene er villige å dele.

Videre forskning

OTT og streaming er et stort emne som også arbeider delvis på kryss av andre tekniske disipliner som video og web teknologi. Det finnes derfor mange bruksområder som ikke videre er dekket av denne oppgaven som bruk av livestream i tv-produksjon. Der data helst skal komme så ukomprimert og med høy bitrate som mulig mellom lokasjoner.

Det hadde også vært interessant å dykke ytterligere ned i de utallige andre formålene streaming brukes til i dag, både live og andre «on-demand» tjenester enn OTT og underholdning. Blant annet i forbindelse med undervisning og konferanser, samt digitale møterom og andre arenaer der streaming har fått fotfeste etter å ha blitt tilpasset disse formålene.

Som en større og mer forskende oppgave, ville det også vært interessant å kontakte de ulike strømmeselskapene og etterspørre informasjon og data på hvilke kodek, protokoll og andre data de har om sine tjenester og hvordan de leverer. Og hvordan de har strategisk valgt disse protokollene og kodekene i forbindelse med sin egen forretningsstrategi. Men igjen virker det å være to ledende protokoller på markedet, og det er ikke sikkert at ulikhetene er så store mellom leverandørene. Det er også et stort usikkerhetsmoment om leverandørene faktisk skulle ville dele data om dette selv om det kunne gi en flott innsikt i streamingtjenestene. Men på sikt er det kanskje ikke utenkelig allikevel ut fra hvilken retning streaming virker å ta videre.

Referanseliste

- Albanesius, C. (2013) *Netflix Dropping Silverlight for HTML5*. Tilgjengelig fra: <https://uk.pcmag.com/internet-3/16103/netflix-dropping-silverlight-for-html5> (Hentet: 08.05.22).
- Apple (u.å.¹) *Creating a Multivariant Playlist*. Tilgjengelig fra: https://developer.apple.com/documentation/http_live_streaming/example_playlists_for_http_live_streaming/creating_a_multivariant_playlist (Hentet: 30.04.22).
- Apple (u.å.²) *HTTP Live Streaming*. Tilgjengelig fra: https://developer.apple.com/documentation/http_live_streaming (Hentet: 30.04.22).
- Apple (u.å.³) *HTTP Live Streaming (HLS) Authoring Specification for Apple Devices*. Tilgjengelig fra: https://developer.apple.com/documentation/http_live_streaming/http_live_streaming_hls_authoring_specification_for_apple_devices (Hentet: 01.05.22).
- Bitmovin (2021) *Bitmovin video developer report 2021*. Tilgjengelig fra: <https://go.bitmovin.com/video-developer-report> (Hentet: 14.04.22).
- Brombach, H. (2015) *TV 2 Sumo tar første skritt vekk fra Silverlight*. Tilgjengelig fra: <https://www.digi.no/artikler/tv-2-sumo-tar-forste-skrutt-vekk-fra-silverlight/278290> (Hentet: 14.05.22).
- Demir, H. (2020) *CMAF (Common Media Application Format) Explained. What is CMAF and How Does It Work?*. Tilgjengelig fra: <https://antmedia.io/cmaf-streaming/> (Hentet: 01.05.22).
- Fati, S. M., Sumari, P. (2018) *IPTV: Delivering TV Services over IP Networks*, i Fati, S. M., Azad, S. & Pathan, A. K (red.) *IPTV Delivery Networks*. West Sussex: Wiley.
- Fileinfo.com (u.å.) *.F4M File Extension*. Tilgjengelig fra: <https://fileinfo.com/extension/f4m> (Hentet: 01.05.22).
- Garrison, R. (2021) *Structure of a MPEG-DASH MPD*. Tilgjengelig fra: <https://ottverse.com/structure-of-an-mpeg-dash-mpd/> (Hentet: 24.04.22).
- Groot, J. D. (2018) *What is Digital Rights Management?*. Tilgjengelig fra: <https://digitalguardian.com/blog/what-digital-rights-management> (Hentet: 14.04.22).
- Grut, S. (2020) *Pandemien gjør at Hollywood dropper kinoer og satser på strømming*. Tilgjengelig fra: <https://nrkbeta.no/2020/11/30/pandemien-gjor-at-hollywood-dropper-kinoer-og-satser-pa-stromming/> (Hentet: 23.04.22).
- Gundersen, Ø. R. (2016) *Et farvel til Flash og veien dit*. Tilgjengelig fra: <https://nrkbeta.no/2016/01/25/flash-i-nrks-videoavspillere/> (Hentet: 12.02.22).
- Haivision (2020) *Secure Reliable Transport (SRT) Deployment Guide*. Tilgjengelig fra: <https://www.srtalliance.org/srt-deployment-guide/> (Hentet: 08.05.22).

Hong, S. (2018) *Delivering High-Definition IPTV Services over IP-Based Networks*, i Fati, S. M., Azad, S. & Pathan, A. K (red.) *IPTV Delivery Networks*. West Sussex: Wiley.

Kats, R. (2022) *Netflix statistics: How many subscribers does Netflix have? Worldwide, US member count and growth*. Tilgjengelig fra: <https://www.insiderintelligence.com/insights/netflix-subscribers/> (Hentet: 23.04.22).

Krings, E. (2021) *Streaming Protocols for Live Broadcasting: Everything You Need to Know*. Tilgjengelig fra: <https://www.dacast.com/blog/streaming-protocols/> (Hentet: 12.03.22).

Long, B. (2015) *The structure of an MPEG-DASH MPD*. Tilgjengelig fra: <https://www.brendanlong.com/the-structure-of-an-mpeg-dash-mpd.html> (Hentet: 24.04.22).

Masciopinto, M., Comesaña, P. & Pérez-González, F. (2018) *IPTV Streaming Classification*, i Fati, S. M., Azad, S. & Pathan, A. K (red.) *IPTV Delivery Networks*. West Sussex: Wiley.

Microsoft (2020) *Overview of the ASF Format*. Tilgjengelig fra: <https://docs.microsoft.com/en-us/windows/win32/wmformat/overview-of-the-asf-format> (Hentet: 08.05.22).

Microsoft (u.å.) *Kundestøtten utvikles for Silverlight*. Tilgjengelig fra: <https://support.microsoft.com/nb-no/windows/kundest%C3%B8tten-avvikles-for-silverlight-0a3be3c7-bead-e203-2dfd-74f0a64f1788> (Hentet: 01.05.22).

Mueller, C. (2015) *Microsoft Smooth Streaming*. Tilgjengelig fra: <https://bitmovin.com/microsoft-smooth-streaming-mss/> (Hentet: 01.05.22).

Ozer, J. (2011) *What Is H.264*. Tilgjengelig fra: <https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-H.264-74735.aspx> (Hentet: 18.04.22).

Ozer, J. (2016¹) *What Is AV1?*. Tilgjengelig fra: <https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-AV1-111497.aspx> (Hentet: 18.04.22).

Ozer, J. (2016²) *What Is VP9?*. Tilgjengelig fra: <https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-VP9-111334.aspx> (Hentet: 18.04.22).

Ozer, J. (2019¹) *What Is Streaming?*. Tilgjengelig fra: <https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-Streaming-74052.aspx> (Hentet: 13.04.22).

Ozer, J. (2019²) *What Is VVC?*. Tilgjengelig fra: <https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-VVC-135215.aspx> (Hentet: 18.04.22).

Ozer, J. (2020) *How to Produce CMAF Output and Testing Playback*. Tilgjengelig fra: <https://www.streamingmedia.com/Articles/Editorial/Featured-Articles/How-to-Produce-CMAF-Output-and-Testing-Playback-139742.aspx> (Hentet: 14.05.22).

Ozer, J. (2021) *Beginner's Guide to Video QoE and QoS*. Tilgjengelig fra: <https://ottverse.com/beginners-guide-to-video-qoe-and-qos/> (Hentet: 15.05.22).

Ozer, J. (2022) *The State of Video Codecs 2022*. Tilgjengelig fra: <https://www.streamingmedia.com/Articles/Editorial/Featured-Articles/The-State-of-Video-Codecs-2022-151189.aspx> (Hentet: 17.04.22).

Pennington, A. (2022) *The State of OTT 2022*. Tilgjengelig fra: <https://www.streamingmedia.com/Articles/Editorial/Featured-Articles/The-State-of-OTT-2022-152085.aspx> (Hentet: 14.04.22).

Peters, J. (2021) *You can see Netflix's new AV1 streaming tech on select TVs and the PS4 Pro*. Tilgjengelig fra: <https://www.theverge.com/2021/11/10/22775150/netflix-av1-codec-tv-streaming-ps4-pro> (Hentet: 08.05.22).

Retail magasinet (2019) *Forbrukertrender: Vi blir klokere og mer kresne*. Tilgjengelig fra: <https://www.retailmagasinet.no/euromonitor-forbrukertrender/forbrukertrender-vi-blir-klokere-og-mer-kresne/427856> (Hentet: 19.03.22).

Ronca, D. (2013) *A Brief History of Netflix Streaming*. Tilgjengelig fra: <https://conferences.infoday.com/documents/171/2013SMEast-C101.pdf> (Hentet: 23.04.22).

Siglin, T. (2011) *What Is a Content Delivery Network (CDN)?*. Tilgjengelig fra: [https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-a-Content-Delivery-Network-\(CDN\)-74458.aspx](https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-a-Content-Delivery-Network-(CDN)-74458.aspx) (Hentet: 20.03.22).

Språkrådet og Universitetet i Bergen (u.å.) *«strømming» i Bokmålsordboka*. <https://ordbokene.no/bm/114994/str%C3%B8mming> (Hentet: 09.04.2022).

Streaming Media (u.å.) *VC-1*. Tilgjengelig fra: <https://www.streamingmedia.com/Glossary/Terms/VC-1> (Hentet: 08.05.22).

Swami, U. (u.å.) *HDS*. Tilgjengelig fra: <https://headendinfo.com/hds-http-dynamic-streaming/> (Hentet: 01.05.22).

Tariq, H. (2021) *What Is OTT Advertising, And Why Is It A Trend?*. Tilgjengelig fra: <https://www.forbes.com/sites/forbescommunicationscouncil/2021/02/08/what-is-ott-advertising-and-why-is-it-a-trend/> (Hentet: 14.04.22).

Telenor (2020) *Content Delivery Network – Tjenestebeskrivelse og SLA*. Tilgjengelig fra: <https://www.telenorwholesale.no/wp-content/uploads/2020/06/Bilag-2c-Internet-Service-Portfolio-CDN-tjenestebeskrivelse-Gyldig-2020-07-01.pdf> (Hentet: 23.04.22).

Telia (2020) *Telia Solutions for media*. Tilgjengelig fra: <https://www.teliasolutions.com/uploads/Brochure/Telia-Solutions-for-Media-Brochure.pdf> (Hentet: 28.01.22).

Telia (2022) *Telia lanserer TV-rapport: Kundene vil ha i pose og sekk*. Tilgjengelig fra: <https://presse.telia.no/pressreleases/telia-lanserer-tv-rapport-kundene-vil-ha-i-pose-og-sekk-3165055> (Hentet: 09.04.22).

Vorel, J. (2021) *The Former Netflix DVD Library Is a Lost Treasure We'll Never See Again*. Tilgjengelig fra: <https://www.pastemagazine.com/movies/netflix/netflix-dvd-service-plan-subscribers-discontinued-closing/> (Hentet: 08.05.22).

Wilbert, M. (2022) *What is HLS Streaming and When Should You Use It? [2022 Update]*. Tilgjengelig fra: <https://www.dacast.com/blog/hls-streaming-protocol/> (Hentet: 23.04.22).

Aaron, A. (2014) *Video Encoding At Netflix*. Tilgjengelig fra: <https://mpeg.chiariglione.org/sites/default/files/files/standards/parts/docs/MPEGWorkshopNetflixEncoding.pdf> (Hentet: 08.05.22).

