TV-tittande 2018 – hur svårt kan de va?

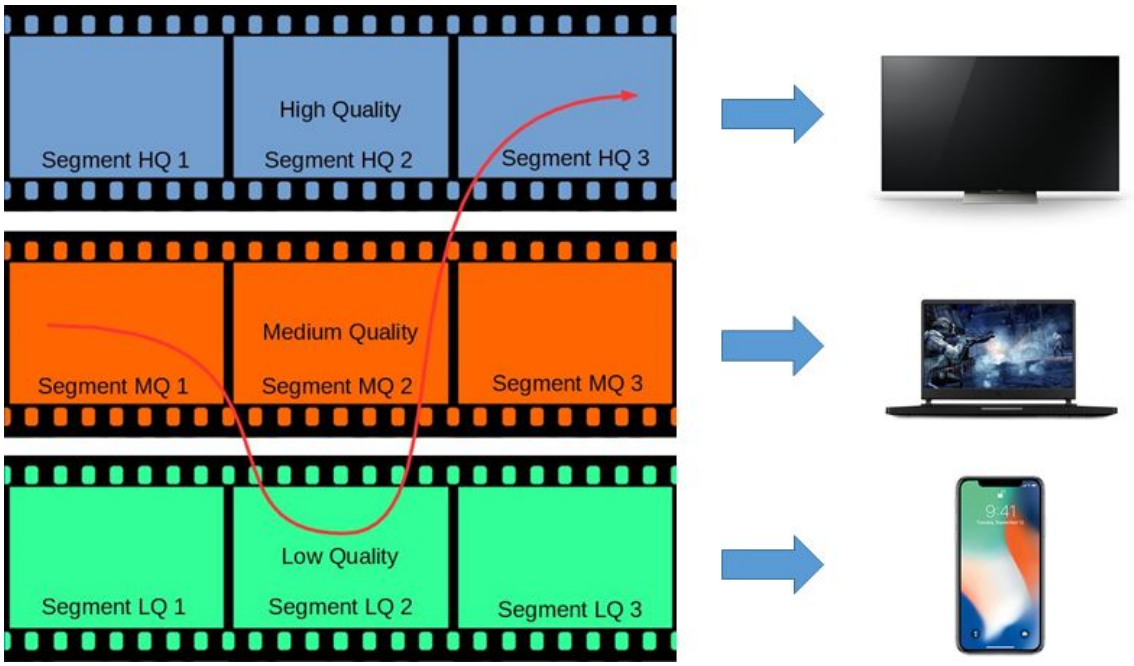
Detta är del två i Attentecs artikelserie om det moderna TV-tittandet. Den handlar om hur man optimerar signalen så att vi med minimal fördröjning kan börja titta och få rätt kvalitet på alla de olika enheter vi idag använder för att konsumera TV. Ytterligare två delar följer inom kort.

I det första avsnittet konstaterades att vi inte längre bara konsumerar film hemma framför TV:n. Vi sitter på olika ställen, med olika anslutning och olika plattformar med olika storlek på skärmarna - och tittar på olika saker. Gemensamt är att vi alla vill ha hög kvalitet - framför allt inget hackande eller långa laddtider!

Vi beskrev i förra avsnittet hur vi med komprimering kan minska storleken på våra videoströmmar och anpassa dem efter anslutning och skärmstorlek, så att vi får en jämn kvalitetsnivå oavsett om du tittar på din mobil på tåget eller på din projektor hemma.

Men hur anpassar vi video efter alla dessa olika förutsättningar? Det enkla svaret vid seklets begynnelse var att erbjuda en handfull alternativ som tittaren själv fick välja mellan efter anslutning och/eller skärmstorlek. Nackdelen var att det var odynamiskt. Om man ville byta kvalitet fick man börja om och ladda ned filen igen med en högre bitrate. I bästa fall kom spelaren ihåg var i filmen du var och kunde åtminstone hoppa dit.

Idag behöver man inte tänka på det - det sker automagiskt i din spelare. Tekniken som gör det här möjligt heter Adaptive BitRate streaming (ABR). ABR har funnits sedan omkring 2002 men slog igenom på allvar 2009 när Apple och Adobe båda släppte sina nya versioner av ABR-streaming som gör det möjligt att optimera uppspelning för “alla” plattformar, skärmar och anslutningar med samma streaming-arkitektur.



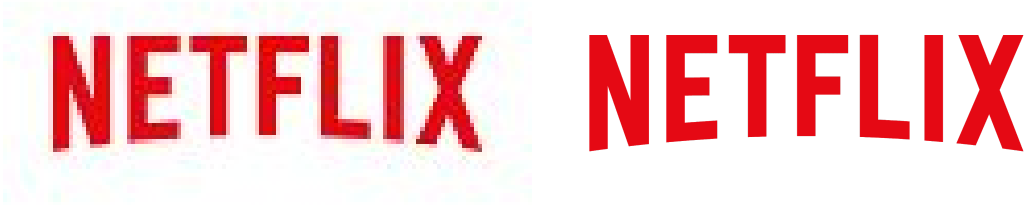
Vi har fortfarande kvar alla våra varianter av filmer med olika komprimering och olika format men vi har delat upp dem i tidsmässigt lika stora segment (typiskt 2-10s långa). Inte en enda stor lång fil alltså. Varje segment är exakt lika långt tidsmässigt för bra respektive dålig kvalitet men de har naturligtvis helt olika bitrates (kanske även encoding) och tar olika mycket lagringsutrymme.

En manifest-fil beskriver ett klipps alla segment och kvalitetsnivåer och din mediaspelare använder sedan den för att ladda ned klippet, segment för segment från vilken kvalitet som nu verkar mest lämpligt *on-the-fly* med hänsyn till din skärmstorlek och anslutning utan din inblandning. Manifest-filer är inte speciellt märkvärdiga, de kan läsas av en människa och är i princip bara en lista över kvalitetsnivåer och deras associerade segment.

Men hur gör man med live-sändningar då, när det inte är lika självklart vad som är början och vad som är slutet? Det fungerar faktiskt på i princip samma sätt. Det som sker är att nya manifestfiler skapas kontinuerligt i takt med att nya segment skapas. Ett manifest innehåller kanske de tre första segmenten. Därefter får man tanka ner nästa manifestfil som beskriver de tre nästa segmenten (med ett eller ett par segments överlapp). Detta introducerar dock latens (fördröjning) eftersom spelaren behöver buffra upp ett par tre segment innan uppspelning påbörjas. iOS-spelaren buffrar typiskt tre segment á 2-10 sekunder, vilket alltså innebär 6-30 sekunders eftersläpning (oaktat avkodning, avkryptering och anslutning). Det går naturligtvis att hålla sig till mycket korta segment, men en segmentlängd på under två sekunder är svårt att uppnå pga den underliggande komprimeringstekniken samt att det blir väldigt mycket “prat” över HTTP när alla dessa små segment ska laddas hem.

Apple HLS och MPEG-DASH är de vanligaste implementationerna av detta koncept, Microsoft MSS och Adobe HDS är andra exempel. De bygger alla på samma princip, men skiljer sig något åt när det gäller codec-stöd, funktioner för Digital Rights Management, reklamstöd etc.

Dålig kvalitet i början
Du kanske har noterat att det ofta är dålig videokvalitet en kort stund när du börjar titta på en film eller ett videoklipp?



Det är inte av en slump. Anledningen är att vi som tittare är tämligen otåliga - vi blir fasligt provocerade av den där spinnande bollen vi såg i första avsnittet av denna artikelserie. Enligt en studie genomförd av Akamai/Gomez så förväntar sig nästan hälften av alla webbanvändare att en webbsida ska ladda klart på max två sekunder och vi tenderar att lämna en sida som tar mer än tre sekunder att ladda. Vi ogillar den spinnande bollen så mycket att vi (statistiskt sett) föredrar snabb laddning framför initial kvalitet!

ABR möjliggör just detta. Din spelare börjar med låga ambitioner (det första segmentet, med lägst kvalitet och minst bandbredd) för att sedan växla upp (eller ned) efter förutsättningarna. Nästa gång du slår på Netflix, YouTube eller annan web-TV, testa att räkna tiden från att det börjar spela tills bilden "knäpper till" och blir bra - det är längden på de underliggande videosegmenten!

Men ABR kommer dock inte utan nackdelar; det blir en enorm mängd filer att hålla reda på, som kräver mycket i form av encoding, utrymme och underhåll. Netflix uppger t ex att de kodar om varje film i 60 olika format! Detta kräver sin arkitektur, men är också ett problem som går att bryta ned till flera mindre jobb som kan göras samtidigt - förutsatt att man har datorkraften. I nästa avsnitt kommer vi att titta närmare på plattformen som gör detta möjligt; molnet.

Denna artikelserie är skriven av Attentecs medarbetare Martin Andersson, Stefan Sax, Mikael Silvén och Anders Weiland. För mer information om Attentec och streaming media, kontakta Stefan Sax, tel 070-6191250, eller Anders Weiland, tel 070-6191230. Läs mer på vår hemsida.