

# **LIVE-STREAMING VIDEOTUOTANNOSSA**

LAHDEN AMMATTIKORKEAKOULU  
Mediatekniikan koulutusohjelma  
Teknisen visualisoinnin suuntautumisvaihtoehto  
Opinnäytetyö  
14.5.2007  
Antti Lehikoinen



**LEHIKOINEN, ANTTI: Live-streaming videotuotannossa**

Teknisen visualisoinnin opinnäytetyö, 49 sivua

Kevät 2007

**TIIVISTELMÄ**

---

Tässä opinnäytetyössä tutkitaan tekniikoita, joilla on mahdollista toteuttaa reaaliaikainen videolähetys tietoverkon kautta tapahtuvana datasiirtona. Työssä perehdytään streaming-tekniikoiden laite- ja ohjelmistovaatimukseen lähettäjän, palvelimen ja asiakkaan näkökulmista. Lisäksi tutkitaan streamingissa käytettäviä tiedonsiirtoprotokollia ja vertaillaan niiden käyttöä lähetyksen laadun ja toimivuuden kannalta.

Työssä käsitellään myös yleisimpiä streaming-formaatteja ja tutkitaan näiden välisiä eroja. Työhön valitut streaming formaatit ovat Windows Media, RealMedia, QuickTime ja Flash. Työssä käsitellään formaattien keskeisimpiä ominaisuuksia ja niiden käyttämiä teknisiä ratkaisuja. Vertailtavia formaatteja käsitellään pääasiassa lähetyksen katsojan näkökulmasta.

Työn viimeisessä osiossa raportoidaan suorana streaming-lähetyksenä toteutettu videotuotanto, joka käsitti Lahden Tiede- ja Yrityspaneisto Neopolissa 8.1.2007 – 19.1.2007 järjestetyn kaksiviikkoisen tapahtuman. Tuotanto oli laaja kokonaisuus, joka sisälsi suorana lähetyksenä toteutettuja haastatteluja ja monikamera-tuotantoja sekä aiemmin kuvattujen valmiiden ohjelmapakettien lähettämistä. Raportissa käsitellään tuotannon kulkua sekä perehdytään siinä käytettyihin kuvaus- ja tuotantolaitteisiin. Tuotannon lopputulosta verrataan alkuperäisiin tavoitteisiin viestinnällisestä ja teknisestä näkökulmasta ja lisäksi tarkastellaan streaming-lähetyksen toimivuutta pienillä budjeteilla toteutetuissa suorissa lähetyksissä.

Avainsanat: streaming, videolähetys, Anycast, Windows Media, RealMedia, Quicktime, Flash

**LEHIKOINEN, ANTTI: Live-Streaming in Video Production**

Bachelor's thesis in visualization engineering, 49 pages

Spring 2007

**ABSTRACT**

---

This thesis deals with data streaming technologies used in the creation of live video broadcasts over the Internet. The main objective was to study technologies and protocols used in reliable real-time data transfers and compare them to coexisting transfer methods used in file servers. Both server- and client-side hardware and software requirements were investigated.

The thesis also includes a comparison of the technological benefits and drawbacks of four most widely used streaming formats. The formats are Windows Media, RealMedia, QuickTime and Flash. The comparison was mostly done from the client's point of view.

A real-life implementation of streaming technologies took place in Lahti Science And Business Park during a two-week event in January 2007. The event was broadcasted live to the Internet and it included live interviews and multi-camera production. The success of broadcast showed the potential of streaming technologies in relatively small live productions. Real-time data transfer technologies proved to be an effective method of creating cost-effective live broadcasts for small to medium-sized audiences. With the rapid growth of available network bandwidth and increasing support from hardware manufacturers, streaming broadcasts are becoming a substantial media channel, which allows smaller producers to reach larger audiences without expensive TV-broadcasting equipment.

Keywords: streaming, video broadcast, Anycast, Windows Media, RealMedia, Quicktime, Flash

# SISÄLLYS

<b>1 JOHDANTO</b> .....	<b>1</b>
<b>2 STREAMING</b> .....	<b>2</b>
2.1 Määritelmä.....	2
2.1.1 Streaming-tekniikka.....	2
2.1.2 Live-streaming.....	5
2.1.3 On-demand streaming.....	5
<b>3 STREAMING-TEKNIIKAT</b> .....	<b>7</b>
3.1 Lähettäjän ja palvelimen tekniikat.....	7
3.1.1 Ohjelmistot.....	7
3.1.2 Laitteistot.....	8
3.1.3 Kaistanleveysvaatimukset.....	11
3.2 Asiakkaan tekniikat.....	12
3.2.1 Ohjelmat.....	12
3.2.2 Laitteistot.....	13
3.2.3 Yhteysvaatimukset.....	14
3.3 Protokollat ja tietoliikenne.....	15
3.3.1 TCP ja UDP.....	15
3.3.2 HTTP-protokolla.....	18
3.3.3 Streaming-protokollat.....	19
3.3.4 Quality of Service ja RVSP.....	22
3.4 Streaming-formaatit.....	24
3.4.1 Windows Media ja ASF.....	24
3.4.2 RealMedia.....	26
3.4.3 QuickTime.....	27
3.4.4 Flash.....	28
3.4.5 Streaming-formaattien ominaisuuksien vertailu.....	30
<b>4 RAPORTTI ANYCAST-TUOTANNOSTA</b> .....	<b>33</b>
4.1 Tuotannon lähtökohdat ja tarkoitus.....	33
4.2 Tuotannossa käytetty laitteisto.....	33
4.3 Tuotannon kulku ja lopputulos.....	37
<b>5 YHTEENVETO</b> .....	<b>42</b>
<b>LÄHTEET</b> .....	<b>43</b>
<b>LIITTEET</b> .....	<b>49</b>

## SANASTO

**DV:** Digitaalinen video, joka on pakattu DV-pakkauksella.

**Enkoodaus:** Datan, yleensä äänen tai videon muuntaminen toiseen muotoon jotakin pakkausmenetelmää hyödyntäen.

**Koodekki:** (engl. *Codec*, **compressor/decompressor**) Pakkauksenhallinta, jolla pakataan ja puretaan dataa.

**kbit/s:** Ilmaisee bittivirran suuruuden, 100 kbit/s = 100 kilobittiä sekunnissa.

**Live:** Reaaliaikainen siirto, suora lähetys.

**Media:** Useita eri merkityksiä, tässä opinnäytetyössä termillä viitataan *videon ja äänen* yhdistelmään tai *media-alaan* yleisesti.

**Metatiedosto:** Tiedosto, joka sisältää kuvauksen toisesta tiedostosta.

**Multicast:** Lähetystekniikka, jossa lähetys toimitetaan katsojille näitä lähinnä olevien reitittimien kautta streamig-palvelinta rasittamatta.

**Streaming:** Datan siirtoa sellaisessa muodossa, että sen sisältöä voidaan käyttää jo latausvaiheessa, vaikka tiedostoa ei olisi vielä kokonaan ladattu.

**Protokolla:** Yhteyksikäytäntö, jota noudattaen kaksi eri laitetta tai ohjelmaa voivat vaihtaa tietoa keskenään. Protokolla määrittelee siirrettävien tietojen esitystavan sekä mekanismin, jolla tiedot siirretään.

**Puskurointi:** Menetelmä, jossa tietoa tallennetaan väliaikaisesti muistiin ennen sen toistamista.

**TCP/IP:** Protokollaperhe, jota hyödynnetään valtaosassa Internet-tiedonsiirtoa. Perustuu verkon laitteiden yksilöllisiin IP-osoitteisiin ja niihin kohdistuviin lähettyksiin.

**Unicast:** Lähetystekniikka, jossa katsojat ottavat suoran yhteyden palvelimeen.

# 1 JOHDANTO

Kuluttajille suunnattujen laajakaistayhteyksien nopeutumisen myötä Internetin kautta tarjottavien multimediapalveluiden mahdollisuudet ovat kasvaneet merkittävästi. Nopeammasta tiedonsiirrosta ovat hyötäneet etenkin videopalvelut, joiden laatu on monella tapaa sidonnainen tiedonsiirron nopeuteen. Nykyiset hitaimmatkin laajakaistayhteydet riittävät sujuvaan videon toistoon kohtalaisella laadulla. Yhteyksien nopeutuminen ja sitä seurannut multimediapalveluiden kysyntä on avannut uusia mahdollisuuksia ja tuonut uudenlaisia toimijoita ja palveluntarjoajia medialle. Etenkin pienemmille tuottajille Internet on avannut uudenlaisen kustannustehokkaan jakelukanavan, jonka kautta voidaan saavuttaa suuria yleisömääriä. Myös valtakunnalliset uutis- ja mediatalot ovat alkaneet hyödyntämään streaming-palveluita ja tarjoavat ohjelmiaan myös Internetissä joko suorina lähetyksinä tai arkistoituna materiaalina.

Internet-pohjaiset multimediapalvelut tarvitsevat tehokkaita tiedonsiirto-menelmiä. Tyypillisesti web-sivujen siirtoon käytetyt tekniikat soveltuvat huonosti aikakriittisen multimedian, kuten esimerkiksi suoran videolähetyksen siirtoon, joten sitä varten on kehitetty suoratoisto- eli streaming-tekniikoita. Näiden sisältämien toimintojen avulla voidaan parantaa lähetyksen toimintavarmuutta, vähentää verkon kuormitusta sekä toteuttaa aitoja suorita lähetyksiä.

Tämän opinnäytetyön tarkoituksena on tutkia streaming-tiedonsiirrossa käytettyjä tekniikoita sekä perehtyä tarvittaviin laitteistoihin ja ohjelmistoihin lähettäjän, palvelimen ja asiakkaan näkökulmista. Näiden lisäksi tutkitaan streaming-protokollia ja niiden käyttöä suorissa lähetyksissä. Työssä käsitellään myös neljän eri valmistajan kehittämiä streaming-formaatteja ja vertaillaan näiden ominaisuuksia keskenään. Työn viimeisessä osiossa raportoidaan tammikuussa 2007 suorana streaming-lähetyksenä toteutettu videotuotanto.

## 2 STREAMING

### 2.1 Määritelmä

#### 2.1.1 Streaming-tekniikka

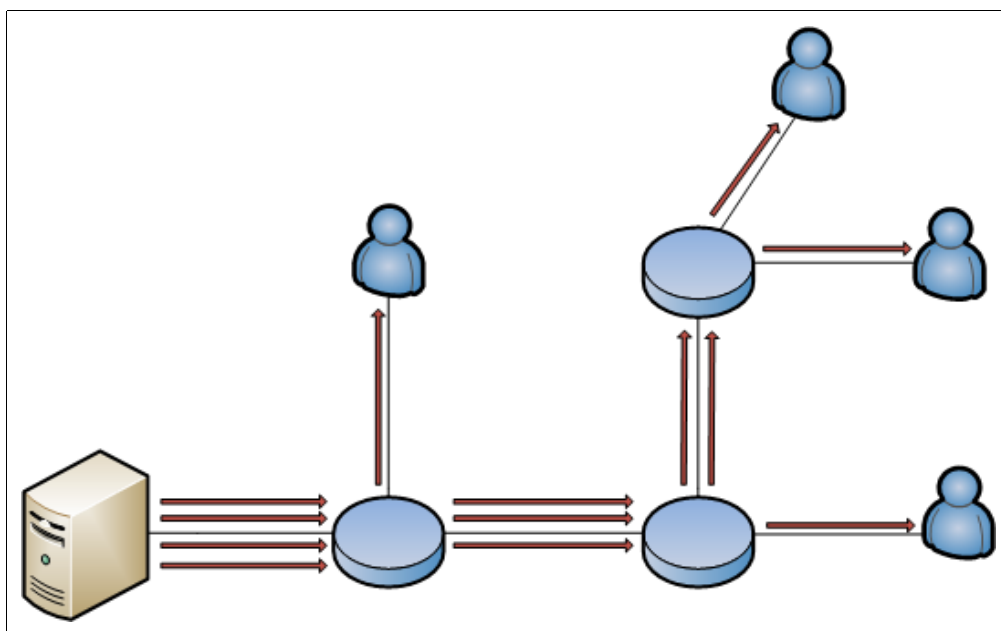
Streaming-tekniikalla tarkoitetaan tiedostojen siirtoa tietoverkossa sellaisessa muodossa, että tiedoston lataaja voi aloittaa sen sisällön käytön jo ennen kuin tiedosto on kokonaan ladattu tämän tietokoneelle. Tekniikkaa voidaan käyttää periaatteessa minkä tahansa tiedostotyypin siirtoon, joka tukee käyttöä jo latausvaiheessa, mutta tyypillisimpiä käyttökohteita ovat erilaiset multimediatiedostot, kuten kuva-, ääni- ja videotiedostot. Suurin rajanveto perinteisen kuvan- tai äänenvälityksen ja streamingin välillä on se, että streaming toteutetaan tietoverkon yli asiakkaalle suunnattuna tiedonsiirtona. Vastaavasti perinteinen televisio-ohjelma tai radiolähetys eivät ole varsinaista streamingia niiden siirtomenetelmistä johtuen, vaikka lähetystä voidaankin seurata samaan aikaan sen siirtymisen kanssa. Tyypillisiä perinteisiä medioita, joita ei voida luokitella streaming-tyyppiseksi, ovat esimerkiksi cd- ja dvd-levyt sekä kirjat. (Adobe 2001, 3-7.)

Videolähetysten kannalta streaming-tekniikalla toteutettu tietoliikenne sisältää useita etuja perinteisiin tiedostopalvelimissa käytettyihin siirto-tekniikoihin verrattuna. Streamingia varten suunnitellut protokollat ovat verkon kuormituksen kannalta tiedostopalvelimen tekniikoita tehokkaampia; ideaalitapauksessa kukin datavirta saa käyttöönsä juuri sen verran kaistanleveyttä, kuin se tarvitsee, kun taas tiedostopalvelin säätelee kaistanleveyden jakoa liikenteen mukaisesti niin, että lähetykselle tärkeää minimivaatimusta ei välttämättä saavuteta. Etua saadaan myös siitä, että streaming-tekniikka tukevaan tiedostomuotoon tallennettua materiaalia on mahdollista indeksoida ja sitä voidaan toistaa halutusta kohdasta, mikäli käytetään streaming-protokollia. Etenkin pidempiä lähetyksiä seurattaessa hyötyä saavutetaan myös siitä, että tiedostoja ei tarvitse tallentaa väliaikaisesti vastaanottajan päätelaitteen tai -ohjelman välimuistiin. Suurin hyöty tästä saadaan etenkin mobiilipuolella, jossa laitteiden muistimäärät ovat rajallisia. Streaming-tekniikan haittapuoliksi voidaan katsoa erilliset, formaatista riippuen kalliit



palvelinohjelmistot sekä se, että monet palomuurit oletusarvoisesti estävät streaming-protokollien käyttämien tietoliikenneporttien kautta tulevat tietoliikenteen. Myöskään tavallista web-liikennettä nopeuttavat välityspalvelimet eivät hyödytä streaming-lähetystä, ellei kyseessä ole erikseen streamingiin dedikoitu välityspalvelin (Häkkinen 2000, 11-12).

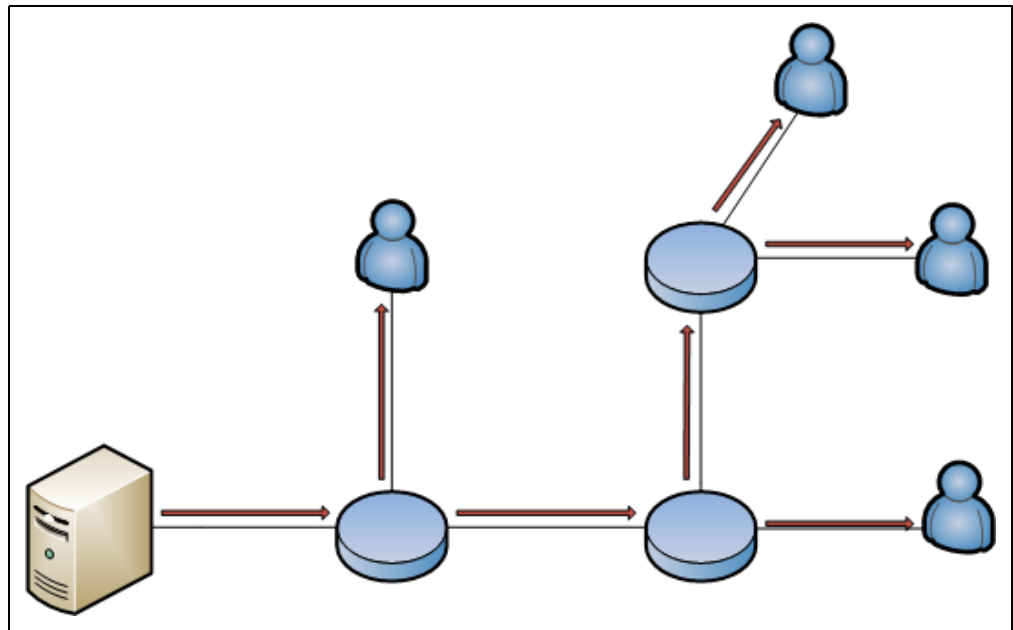
Multimedian siirtämiseen streaming-tiedonsiirron avulla liittyy olennaisena osana myös multicast-lähetystekniikka, jonka avulla voidaan saavuttaa merkittäviä säästöjä tiedonsiirron määrissä. Normaalisti datavirta lähetetään palvelimelta jokaiselle käyttäjälle erikseen, jolloin samaa identtistä dataa lähetetään rinnakkain useana virtana. Tällaista tekniikkaa kutsutaan unicast-lähetämiseksi (Kuva 1).



**KUVA 1: Unicast-lähetystekniikka (Lehikoinen 2007)**

Multicastin avulla jokaiselle käyttäjälle ei tarvitse lähettää omaa datavirtaa, vaan sama data voidaan kuljettaa yhtenä virtana verkon reitittimien välillä ja jakaa eteenpäin vasta käyttäjää lähinnä olevan reitittimen kohdalla (Kuva 2). Multicastia käytettäessä myös streaming-palvelimeen kohdistuva rasitus pienenee olennaisesti, koska palvelimen ei tarvitse käsitellä jokaiselle katsojalle siirrettävää lähetystä erikseen, vaan se ohjaa käyttäjien yhteydet näitä lähinnä oleville multicast-reitittimille. Multicast toimii virtuaalisessa Mbone- eli multicast backbone

-verkossa, joka käyttää datan siirtoon tavallisia Internet-liikennettä kuljettavia reitittimiä, eli Mbone on rakennettu Internet-verkon päälle. Reitittimet kykenevät jakamaan lähetystä mistä tahansa verkon solmukohdasta, mutta reitittimien välillä liikenne siirretään yhtenä unicast-virtana. Tällöin myös sellaiset reitittimet, jotka eivät tue multicastia, voivat lähettää multicast-paketteja edelleen, vaikka eivät osaisi käsitellä niiden sisältöä. Varsinaisen Mbonen muodostavat multicast-tekniikkaa tukevat palvelimet ja reitittimet. (Apple 2003, 17.)



**KUVA 2: Multicast-lähetystekniikka (Lehikoinen 2007)**

Videolähetyksissä multicastin käyttö vaatii joko yhteyden Mbone-verkoon tai vaihtoehtoisesti lähetys tulee toteuttaa sellaisessa verkossa, jossa kaikki reitittimet ovat asetettu tukemaan multicastia (Apple 2003, 17.). Monista eduistaan huolimatta multicast ei ole yleistynyt kovinkaan nopeasti, ja sen käyttöönottoa on vieroksuttu etenkin mahdollisten tietoturvariskien vuoksi. Vaikka reitittimet sisältäisivätkin tuen multicastille, ominaisuus on useimmiten oletusarvoisesti kytketty pois päältä. Lähitulevaisuudessa multicastin käyttö kuitenkin laajentunee viimeistään kaupallisten IP-TV -palveluiden yleistymisen ja runkoverkon laitteiston uusiutumisen myötä (FiCom 2006).

### **2.1.2 Live-streaming**

Live-streaming on tiedonsiirtoa sellaisesta lähteestä, jota luodaan reaaliaikaisesti. Tyypillisiä reaaliaikaisia lähteitä ovat video- ja äänipalvelut, kuten Internetin yli toteutettu suora videolähetys tai radiokanavalle lisäarvoa tuottava nettiradio. Tällaisissa palveluissa palvelin lähettää asiakkaille materiaalia sitä mukaa, kun sitä tuodaan palvelimelle. Suoran lähetyksen toteutus vaatii aina erillisen ohjelmiston, jolla lähetettävä materiaali muunnetaan haluttuun formaattiin ja lähetetään palvelimelle. Myös palvelimella tulee olla vastaavat ohjelmat, jotka ottavat vastaan streaming-lähetystä ja jakavat sen edelleen katsojille. Suorissa lähetyksissä kaikille katsojille lähetetään samaa dataa, eli katsojilla ei ole mahdollisuutta vaikuttaa siihen, mitä kohtaa lähetyksestä toistetaan. Suorista lähetyksistä voidaan vaihtoehtoisesti käyttää myös nimitystä webcasting. (Apple 2003, 14.)

Vaikka usein puhutaankin suorasta lähetyksestä, käytännössä kuitenkin lähetyksessä on useiden sekuntien mittainen viive. Tämä johtuu lähetyksen puskuroinnista asiakkaan, palvelimen, tai mahdollisesti molempien ohjelmissa. Puskurointia joudutaan käyttämään verkkotekniikoiden epävarmuudesta johtuen, koska datapaketit eivät välttämättä saavu vastaanottajalle alkuperäisessä järjestyksessä, tai paketit saattavat jopa hävitä kokonaan. Puskuroitua lähetystä esitettäessä kokonaan puuttuvat paketit voidaan pyytää palvelimelta uudestaan, jos toimenpide saadaan tehtyä puskurointiajan sisällä ja tarve vaatii pakettien uudelleenlähettämistä, mutta useimmiten yksittäisten pakettien puuttuminen ei aiheuta havaittavaa häiriötä lähetyksessä. (Adobe 2001, 7.)

### **2.1.3 On-demand streaming**

Streaming-tiedonsiirtoa käytetään laajalti myös valmiiksi palvelimella olevan materiaalin esittämiseen. Tällöin puhutaan on-demand -streamingista, jossa palvelin aloittaa streaming-siirron asiakkaalle tämän lähettäessä pyynnön halutusta palvelusta. Tyypiesimerkkinä on-demand -streamingista voidaan pitää suosittua YouTube -videopalvelua (YouTube, 2007). Kyseisessä palvelussa videot sijaitsevat kiinteästi

palvelimella, mutta ne ovat tallennettuna streaming-tekniikaa tukevaan tiedostoformaattiin, jolloin asiakas voi aloittaa videon katsomisen välittömästi latauksen alettua. On-demand -streaming voidaan toteuttaa joko oikeilla streaming-protokollilla tai vaihtoehtoisesti perinteisillä tiedostopalvelimissa käytetyillä siirtotekniikoilla. Jälkimmäisellä tavalla toteutettuja palveluita kutsutaan usein vale-streamingiksi, koska käytännössä kyseessä on vain tavallinen tiedostonlataus palvelimelta, ei niinkään aito streaming. Käyttäjälle on-demand palvelussa ei ole välttämättä suurtakaan merkitystä sillä, onko kyseessä aito- vai vale -streaming. Joitakin eroja saattaa esiintyä videon hallintaan liittyvissä kontrolloissa, kuten pikakelaustoiminnoissa. Sen sijaan verkon kuormituksen kannalta asialla voi olla etenkin laajassa mittakaavassa merkittäviä vaikutuksia. (Adobe 2003, 7-8.)

## 3 STREAMING-TEKNIIKAT

### 3.1 Lähettäjän ja palvelimen tekniikat

#### 3.1.1 Ohjelmistot

Streaming-lähetyksiä toteutettaessa lähetyksen tuotantopaikassa tarvitaan ohjelmisto, joka pakkaa videomateriaalin streaming-formaattiin ja lähettää sen edelleen palvelimelle. Tällaisia ohjelmia kutsutaan enkoodaus- tai lähetysohjelmiksi. Viimeistään enkoodausvaiheeseen tullessa video- ja audiosignaalit muutetaan digitaaliseksi, mikäli niitä on aiemmassa tuotantovaiheessa käsitelty analogisesti. Enkoodausohjelmat sisältävät toimintoja, joilla voidaan määritellä lähetettävän materiaalin asetuksia, kuten äänen ja kuvan bittivirta, kuvan resoluutio, lähetyksen salaaminen ja käytettävä pakkausmenetelmä. Useat enkoodausohjelmat mahdollistavat myös kolmannen osapuolen tekemien plug-in -laajennusten käytön, joiden avulla voidaan ohjelmallisesti käyttää kuvaa muokkaavia digitaalisia suotimia tai esimerkiksi lisätä kuvan päälle vesileiman kaltainen kuva. Useimmat enkoodausohjelmat ovat streaming-formaatin valmistajan tarjoamia maksuttomia ohjelmia, jotka sisältävät tärkeimmät toiminnot lähetyksen toteuttamiseen. Ohjelmista on saatavilla myös laajempia maksullisia versioita, joiden ominaisuudet vastaavat ammattitason tuotannon tarpeisiin. (Microsoft, 2005.)

Streaming-palvelinohjelmiston tehtävä suorassa lähetyksessä on yksinkertaisesti toimittaa palvelimelle lähetettävä videovirta edelleen lähetyksen katsojille. Palvelinohjelmistot mahdollistavat kuitenkin monimutkaisiakin konfiguraatioita, joissa voidaan yhtäaikaisesti lähettää useaa erillistä suoraa lähetystä tai samanaikaisesti palvella myös on-demand -videoiden katsojia. Palvelinohjelmiin voidaan määritellä enkoodaus- ja lähetysohjelmia varten salasanasuojatut käyttäjätilit, joiden avulla eri lähettäjät erotellaan toisistaan ja palvelin suojataan asiattomalta liikenteeltä. Tietoliikenteen rajoituksista, kuten kireistä palomuuriasetuksista johtuen useimmat palvelinpuolen -ohjelmistot mahdollistavat monien eri tietoliikenneprotokollien käytön samanaikaisesti. Tällöin voidaan kiertää tilanne, jossa asiakas on sellaisen palomuurin takana, joka hyväksyy vain web-sivujen lataamisen eli tietoliikenneportti 80:n kautta saapuvan liikenteen. Mikäli streaming-videon siirto suorite-

taan portin 80 kautta HTTP-protokollalla, on samalta palvelimelta hyvä sulkea mahdollinen web-serveri konfliktien välttämiseksi (Apple 2003, 40). Palvelimiin ja verkkoon kohdistuvassa rasituksessa esiintyy merkittäviä eroja riippuen käytetystä protokollasta.

Suurissa streaming-toteutuksissa saattaa olla käytössä useita eri palvelimia. Tällöin myös ohjelmien tulee olla skaalautuvia, eli niiden pitää kyetä jakamaan kuormaa usean palvelimen kesken. Palvelinohjelmistosta riippuen yhtä palvelinta voidaan käyttää pääpalvelimena, joka vastaanottaa yhteydet ja ohjaa ne vähiten kuormitetulle palvelimelle. Vastaavasti streaming-palvelimen tueksi on mahdollista rakentaa verkosto streaming-välityspalvelimia, jotka toistavat palvelimelta tulevan lähetyksen. Välityspalvelimien avulla verkon kuormitusta voidaan vähentää olennaisesti, koska jokainen asiakas voidaan ohjata tätä lähimpänä olevalle välityspalvelimelle. (RealNetworks, 2005.)

### **3.1.2 Laitteistot**

Videolähetyksen enkoodaamiseen vaadittavan laitteiston laskentatehon vaatimukset määräytyvät pääosin sen mukaan, mitä formaattia ja pakkausmenetelmää käytetään. Myös yhtäaikaisesti eri laaduilla lähetettävien streamien määrä lisää laskentatehon tarvetta, koska jokainen lähetyks joudutaan enkoodaamaan erikseen. Multicastin käytöllä voidaan saada merkittävää säästöä laskentatehon tarpeeseen, mutta käytännössä multicastin käyttöastetta ei voida kovinkaan tarkkaan ennakoida, koska se edellyttää tukea myös verkon reitittimiltä ja katsojien käyttämiltä ohjelmilta. Enkoodausohjelmistojen tueksi on kehitetty myös laitteistoja, joilla materiaalin pakkaaminen onnistuu reaaliajassa, ilman että lähetettävän laitteen suoritus kuormitetaan videon pakkaamisella. Suurin osa videon pakkaamiseen tarkoitetuista laitteista on suoraan tietokoneeseen asennettavia lisäkortteja, mutta myös kokonaan erillisiä laitteita on saatavilla. Lisäksi on olemassa sulautettuja järjestelmiä, joissa enkoodaamisen tekevään laitteeseen on yhdistetty muita videotuotannossa tarvittavia laitteita, kuten esimerkiksi kuvamikseri tai jokin tallennuslaite. (Sony, 2005; NewTek, 2007.)

Streaming-palvelinohjelmistojen laitteistovaatimukset riippuvat lähes täysin lähetyksen ja niiden katsojien määrästä. Mikäli halutaan palvella

useita tuhansia katsojia, streaming-palvelimina tulisi käyttää joko moni-prosessorijärjestelmiä tai vaihtoehtoisesti useita rinnakkaisia palvelinkoneita laskentatehon riittävyyden takaamiseksi. Laskentatehon tarpeeseen vaikuttaa olennaisesti myös se, käytetäänkö unicast- vai multicast-lähetystä, joista ensimmäinen on palvelimen kannalta huomattavasti jälkimmäistä raskaampaa, koska palvelin joutuu itse käsittelemään erikseen jokaisen lähetettävän datavirran. Pienissä, joitakin kymmeniä tai korkeintaan satoja katsojia käsittävissä tuotannoissa palvelimena voi toimia myös enkoodauskone. Tällöin puhutaan enkoodausohjelman pull-tekniikasta, jolloin joko välityspalvelin tai katsoja vetää lähetysten suoraan ulos enkoodausohjelmasta. Sen sijaan erilliselle streaming-palvelimelle lähetettäessä tekniikkaa kutsutaan push-tekniikaksi, jolloin enkooderi ikään kuin työntää lähetysten palvelimelle. (Microsoft, 2002a.)

RealNetworksin toimesta Lab Acquisition Corporation KeyLabs:n toteuttamasta tutkimuksesta ilmenee, että myös heikompi tasoinen palvelinlaitteisto riittää hyvälaatuiseen streaming-lähetykseen sellaisissa tapauksissa, joissa katsojia on muutamia satoja (Taulukko 1). Tutkimuksessa käytettiin RealMedia:n Helix Universal Server -palvelinohjelmaa, jolla lähetettiin RealMedia 9 -videota. Tutkimuksessa käsiteltiin myös eri käyttöjärjestelmien vaikutusta suurimpaan mahdolliseen toimivaan katsojamäärään. Palvelinkoneena järjestelmässä käytettiin Intel Pentium III Xeon 700Mhz -tuplaprosessorijärjestelmää, jossa oli 2 gigatavua muistia. (Smith, 2007.)

**TAULUKKO 1: Käyttöjärjestelmän vaikutus katsojamäärään (KeyLabs, 2002)**

<b>Suurin mahdollinen katsojamäärä, RealMedia 9</b>				
<b>Käyttöjärjestelmä</b>	<b>Sovellus</b>	<b>20 Kb/s</b>	<b>225 Kb/s</b>	<b>500 Kb/s</b>
<b>Microsoft Windows 2000 Advanced Server</b>	<b>Helix Universal Server</b>	<b>7000</b>	<b>900</b>	<b>580</b>
<b>Red Hat Linux 7.3</b>	<b>Helix Universal Server</b>	<b>11000</b>	<b>1360</b>	<b>930</b>

Taulukosta voidaan todeta, että jo vanhentunutta tekniikkaa edustavalla palvelinlaitteistolla on mahdollista toteuttaa hyvälaatuista lähetystä

noin tuhannelle katsojalle. Myös palvelimen käyttöjärjestelmällä on merkittävä vaikutus siihen, kuinka suurta katsojamäärä kyetään palvelemaan ilman lähetyksen laadun heikkenemistä. Tutkimuksen perusteella voidaan todeta Linux-käyttöjärjestelmän olevan Microsoftin Windows 2000 -palvelinympäristöä tehokkaampi alusta RealMedian Helix-palvelinohjelmiston ajamiseen.

Nykyaikaisemmilla laitteilla päästään konetehon puolesta huomattavasti suurempiin katsojamääriin. RealNetworksin toteuttamassa todellisessa streaming-tuotannossa käytettiin laitteistoa, jossa oli kaksi kappaletta Intel Xeon 3,4 Ghz -prosessoreita ja 4 gigatavua muistia. Helix-palvelinta ajettiin Red Hat Linux Enterprise Server 4 -käyttöjärjestelmässä, ja sillä lähetettiin viittä yhtäaikaista suoraa lähetystä 225 kbit/s -laadulla. Lähetysten keskimääräinen katsojamäärä pysytteli noin 1300 katsojassa, jolloin gigabitin verkkoyhteyden käyttöaste oli noin 300 Mbit/s. Parhaimmillaan pysyvä kaistankäyttö oli noin 600 Mbit/s ja huipussaan noin 900 Mbit/s, jolloin yhtäaikaista katsojia oli noin 4000. Vaikka laitteiston ja ohjelmiston suorituskyky olisikin mahdollistanut vieläkin suuremman katsojamäärän, rajoittavaksi tekijäksi muodostui verkkoyhteyden nopeus. Tietoverkon muu dataliikenne, kuten TCP-kuittausviestit ja pakettien uudelleenlähettämiset muodostavat merkittävän osan verkkoliikenteestä. RealNetworksin tuotannon perusteella voidaan todeta, että mikäli gigabitin yhteydellä varustettu palvelin käyttää kaistaa jatkuvasti yli 300Mbit /s, ollaan verkkoyhteyden kapasiteetin puolesta varsin hyvissä lukemissa ja kaikki tämän ylittävä voidaan katsoa ylimääräiseksi. Hallitussa testiympäristössä vastaavalla laitteistolla oli tultu tilanteeseen, jossa gigabitin verkkoyhteys ei riittänyt palvelemaan kaikkia käyttäjiä, vaikka koneen laskenta-teho olisikin riittänyt vielä suuremman katsojamäärän palvelemiseen. (Smith, 2007.)

Esimerkitapauksesta voidaan todeta, että vaikka palvelimille voidaankin määritellä konetehon kannalta suurin mahdollinen katsojamäärä, usein suurin rajoittava tekijä on verkkoyhteyden nopeus. Myös käyttöjärjestelmän ja laitteiston nopeus käsitellä ja lähettää dataa vaikuttaa suurimpaan mahdolliseen katsojamäärään. Laitteiston fyysisen muistin määrän lisäksi suorituskykyyn saattaa vaikuttaa se, kuinka paljon käyttöjärjestelmä kykenee antamaan muistia yksittäiselle sovellukselle. Esimerkitapauksen Helix -palvelinohjelmisto voi käyttöjärjestel-



mästä riippuen saada käyttöönsä maksimissaan kaksi tai kolme gigatavua muistia. Käytännössä kuitenkin verkkoyhteyden kapasiteetin raja saavutetaan ennen muistin loppumista. Vastaavasti myös laitteiston suoritusnopeus voi teoriassa muodostua katsojamäärää rajoittavaksi tekijäksi, mutta mikäli käytössä on vähänkään nykyaikaisempia palvelimiin suunniteltuja prosessoreita, verkkoyhteyden raja tulee käytännössä aina aiemmin vastaan. (Smith, 2007.)

Verkkoyhteyksien rajoitteiden vuoksi suuremmissa tuotannoissa on järkevää käyttää useaa palvelinta hajautetusti, koska yksittäisen tehokkaan palvelimen laskentatehosta ei välttämättä kyetä hyödyntämään sen koko kapasiteettia. Tällöin myös mahdollisissa vikaantumisissa koko tuotanto ei keskeydy yhden palvelimen tai yhteyden vikaantumisen myötä vaan kuormaa voidaan jakaa muille palvelimille (Smith, 2007). Myös streaming-välityspalvelimien käytöllä voidaan mahdollistaa huomattavasti suuremmat katsojamäärät yksittäiseen palvelimeen verrattuna, koska palvelimet voivat sijaita fyysisesti eri verkoissa, jolloin ne eivät jaa samaa kaistaa pääpalvelimen kanssa.

### **3.1.3 Kaistanleveysvaatimukset**

Digitaalisesti tallennettu video sisältää runsaasti dataa ja se vaatii paljon tallennustilaa. Tästä johtuen myös tietoverkon kautta lähetettävää videota varten tarvitaan kohtalaisen paljon kaistanleveyttä. Käytännössä kaikki verkon yli lähetetty videomateriaali muunnetaan vähemmän kaistanleveyttä vaativaan muotoon jollakin kompressointi- eli pakkaustekniikalla (Adobe 2001, 30-33). Ilman pakkaustekniikoita jopa gigabitin verkkoyhteyden kapasiteetti täytyisi jo muutamalla kymmenellä katsojalla, koska DV-laatuinen video vaatii kaistanleveyttä 25 Mbit/s jokaista käyttäjää kohden. Tehokkailla pakkaustekniikoilla lähetyksen bittivirta saadaan pudotettua murto-osaan alkuperäisestä, mutta samalla materiaalin laatu kärsii. (MainConcept 2006-)

Videoiden vaatima kaistanleveys katsojamäärään nähden saadaan laskettua videon bittivirrasta. Laskutoimituksissa on hyödynnetty binäärikorjausta, jolla saadaan laskettua kaista- ja tilavaatimusten todellinen määrä tietokoneen binäärijärjestelmässä (TechTerms, 2007).

Puolitoista tuntia kestävä videotuotanto 384 kbit/s streaming-lähetyksenä tuhannelle katsojalle vaatii kaistanleveyttä seuraavasti:

$$384 \text{ kbit/s} \cdot 1000 = 384\,000 \text{ kbit/s} = 384 \text{ Mbit/s}$$

Tästä saadaan laskettua kokonaiskaista puolentoista tunnin ajalta:

$$384 \text{ Mbit/s} \cdot 5400 \text{ s} = 259,2 \text{ GB/h}$$

josta todellinen kaistan määrä binäärikorjattuna:

$$259,2 / 1,074 = 241,34 \text{ GiB/h}$$

Käytännössä jo esimerkkitapauksen kaltainen pienehkö tuotanto vaatii streaming-palvelimelta gigabitin yhteyden verkkoon, jotta kaikille katsojille riittäisi kaistaa. Vaihtoehtoisesti voitaisiin käyttää neljää erillistä palvelinta, joista jokaisella on vähintään sadan megabitin yhteys verkkoon. Mikäli video haluttaisiin lisäksi tallentaa on-demand -palvelua varten, sille tarvittava tallennustila voidaan niin ikään laskea videon bittivirrasta:

$$\text{tilantarve (MiB)} = \text{pituus (s)} \cdot \text{bittivirta (bit)} / 8, 388\,608$$

Edellä mainittu 384 kbit/s -laadulla tallennettu puolitoista tuntia kestävä video vaatisi tilaa seuraavasti:

$$(5400 \text{ s} \cdot 384 \text{ kbit/s}) / 8, 388\,608 = 247,2 \text{ MiB}$$

## 3.2 Asiakkaan tekniikat

### 3.2.1 Ohjelmat

Streaming-lähetyksen seuranta vaatii asiakkaalta lähetyksen formaattia tukevan vastaanotto-ohjelman. Useimmat ohjelmat ovat saatavilla erikseen asennettavina kokonaisina ohjelmina tai vaihtoehtoisesti Internet-selaimeen asennettavina laajennuksina. Kokonaisina ohjelmina asennettavat versiot tarjoavat käyttäjälle laajempia palvelukokonaisuuksia sekä Internet-selaimesta riippumatonta toiminnallisuutta. Lä-

hes kaikista asiakasohjelmista on saatavilla maksuttomia versioita, mutta nämä sisältävät lähes poikkeuksetta mainoksia eivätkä mahdollista kehittyneempien ominaisuuksien käyttämistä, kuten sisällön tuottamista kyseisen ohjelman formaatilla. Erillisiin ohjelmiin on usein myös integroitu jonkintasoinen Internet-selain, joilla käyttäjille toimitetaan tuoreimmat uutiset ja joiden avulla on mahdollista selata julkisia multimediasivustoja, kuten Internet-radioita tai videolähetyksiä.

Suoria lähetyksiä seurattaessa ohjelmat ottavat yhteyden streaming-palvelimeen esimerkiksi web-sivulla olevan linkin kautta ja lähettävät yhteydenavaamispyynnön palvelimelle. Streaming-formaatista riippuen yhteys voidaan avata myös metatiedoston avulla, joka sisältää lähetyksikohtaisia määrittäjiä tai ohjaukskomentoja, kuten mainosten tai muun sisällön toistamisen ennen yhteyden avaamista varsinaiseen lähetykseen. Metatiedostot ovat lähes poikkeuksetta tekstitiedostoja, jotka sisältävät tiedot XML-muodossa. (Microsoft 2003) Session eli istunnon avaamisen yhteydessä käytetään formaatista riippuen eri sisällönkuvausprotokollia, joiden avulla asiakkaan ohjelma saa tarvittavat tiedot siitä, miten liittyä lähetykseen. (Apple 2003, 47.)

### **3.2.2 Laitteistot**

Asiakkaalla tulee olla riittävän tehokas laitteisto, jotta streaming-lähetyksiä voitaisiin seurata sujuvasti. Pääsääntöisesti streaming-palveluiden videokuvan resoluutio on kohtalaisen pieni, joten sen toistamiseen ei tarvita erityisen tehokasta laitetta. Useimmissa streaming-palveluissa käytetty resoluutio on 320 x 240 pikseliä tai jopa pienempi. PC-laitteistoissa nykyisten käyttöjärjestelmien minimivaatimukset ovat täysin riittävät tavallisten streaming-lähetyksien seurantaan. Myös lähetyksen pakkaamiseen käytetyt video- ja audiokoodekit vaikuttavat laitteiden laskentatehon vaatimuksiin toisten pakkausmuotojen ollessa raskaampia purkaa.

Mobiilipuolella laskentatehosta saattaa muodostua kynnyskysymys, joka vaikuttaa lähetyksen käytettävän bittivirran ja pakkausmuodon valintaan. Monet puhelin- ja kämmentietokonevalmistajat ovatkin tuoneet markkinoille malleja, joissa on laitteistopohjainen tuki yleisimmille videonpakkausmenetelmille. Tällöin videon purkamisprosessin pitäisi

onnistua suoraan laitteistolla, jolloin toiston pitäisi olla sujuvaa. (Manifest Technology, 2006.) Lähitulevaisuudessa myös PC-puolella laitevaatimukset tulevat nousemaan bittivirtojen kasvaessa ja etenkin siinä vaiheessa, kun aletaan lähettää HD-tasoista materiaalia. Teräväpiirto-lähetysten sisältämän datanmäärän purkaminen vaatii huomattavasti suurempaa laskentatehoa suurentuneen kuvakoon ja kehittyneempien pakkausmenetelmien vuoksi. HD-streaming edellyttää kuitenkin nykyistä huomattavasti suurempia kaistaleveyksiä, eivätkä varsinkaan tavallisimmat kotitalouksille tarjotut laajakaistaliittymät ole vielä riittävällä tasolla HD-aikakautta varten. Tavallisella MPEG-2 -koodekillä pakatut HD-lähetykset vaativat yhteyden nopeudelta noin 16 Mbit/s, mutta tehokkaammalla MPEG-4:lla päästään jo 5 – 8 megabittiin sekunnissa. (Ficom 2006.)

### **3.2.3 Yhteysvaatimukset**

Asiakkaan yhteysvaatimukset ovat huomattavasti palvelintason vaatimuksia kevyemmät, koska asiakkaan ei tarvitse ottaa vastaan kuin yhtä datavirtaa. Tällöin asiakkaan yhteyden miniminopeuden määrittelee streaming-lähetysten bittinopeus. Käytännössä kuitenkin videon bittivirta ei saisi ylittää yli 75 prosenttia asiakkaan saatavilla olevasta kaistanleveydestä, koska verkkoyhteyden tekniikasta ja mahdollisista muista yhtäaikaisista palveluista johtuen teoreettista maksiminopeutta ei saavuteta käytännössä koskaan (Apple 2005, 20). Useissa palveluissa käytetyn 384 kbit/s -nopeudella toistettavan streaming-lähetysten katsomiseen riittää hyvin 512 kbit/s laajakaistayhteys, joka oli yleisin laajakaistanopeus Suomessa vuoden 2006 viimeisellä neljänneksellä. (Viestintävirasto, 2006). Käytettäessä oikeita streaming-protokollia voidaan käyttää toimintoja, joilla asiakkaan yhteysnopeus voidaan tunnistaa, ja tälle voidaan lähettää tarvittaessa sopivammalla bittinopeudella enkoodattua materiaalia. Tällöin streaming-lähetystä voidaan lähettää hyvinkin korkealla laadulla, mutta se on siltikin katsottavissa myös hitaammilla yhteyksillä. Tekniikkaa kutsutaan adaptiiviseksi eli mukautuvaksi enkoodaamiseksi (Häkkinen 2000, 19). Mukautuvan enkoodauksen heikkoutena ainoastaan lähetysten bittivirtaa voidaan muuttaa, mutta sen resoluutiota ei voida vaihtaa lennosta. Käytännössä rajoitus aiheuttaa sen, että minimi- ja maksimilaadun välillä ei voi olla kovinkaan suurta eroa. Tämä aiheutuu siitä, että mukautuvan enkoodaamisen hyödyt jäisivät pieniksi bittivirran vaihtelun ääripäissä,

koska pientä kuvaa on epäkäytännöllistä lähettää suurella bittivirralla ja vastaavasti pieni bittivirta tekee suuresta kuvasta huonolaatuisen. Vaihtoehtona mukautuvalle enkoodaukselle samaa lähetystä voidaan lähettää rinnakkaisina streameina, jolloin katsoja itse valitsee seurattavan laadun. Menetelmä kuitenkin edellyttää katsojalta riittävää tietämystä yhteysnopeudestaan.

### 3.3 Protokollat ja tietoliikenne

#### 3.3.1 TCP ja UDP

Kaikessa IP-verkon yli tapahtuvassa tiedonsiirrossa käytetään jotakin ydinprotokollaa. Streaming-tietoliikenteessä, kuten myös yleisimmissä web-palveluissa käytetään TCP/IP-protokollaperheeseen sisältyviä TCP- tai UDP-protokollia. Protokollat määrittelevät lähetettävän datan muodon ja rakenteen. TCP:ta käyttävät sovellukset käsittelevät tietoja tietovuona, jossa itse tieto kulkee datasegmentteinä, jotka koostuvat TCP- eli Transmission Control Protocol -otsikosta ja tietokentästä. UDP:n eli User Datagram Protocol:n kohdalla data kulkee sovelluksien välillä viesteinä ja itse tieto muodostuu paketeista, jotka sisältävät TCP:n tapaan otsikkotiedot ja tietokentän. Vaikka TCP ja UDP sisältävätkin useita vastaavuuksia, niissä on paljon merkittäviä eroja yhteyskäytäntöjen ja verkon kuormituksen kannalta, jotka osaltaan määrittelevät sen, kumpiko tekniikoista on parempi eri käyttötarkoituksissa. (Casad & Willsey 1999, 101-107.)

Lähtökohtaisesti TCP-tekniikka on tiedonsiirron kannalta varmempaa, mutta palvelimen kannalta huomattavasti UDP-muotoa raskaampaa, koska TCP sisältää laajat toiminnot tiedon perille menemisen varmistamiseksi, sekä useita erilaisia ohjaustoimintoja (Taulukko 2). Osa tarkastuksista saattaa olla sovelluksen tarpeista riippuen täysin turhia, jolloin ne aiheuttavat vain lisäkuormaa yhteydelle. Sen sijaan UDP-tekniikkaan ei sisälly juuri minkäänlaista virheenkorjausta ja sen käsittely onkin huomattavan paljon TCP:ta nopeampaa (Taulukko 3). UDP-tekniikka on yleisempi multimediaa sisältävissä streaming-toteutuksissa sen keveyden vuoksi, vaikka se onkin toiminnaltaan TCP:ta epävarmempi. (Casad & Willsey 1999, 86.) Tyypillisimpiä UDP:n käyttökoh-

teita ovat suorat videolähetykset, voice-over-IP -puhelut, videokonferenssit ja -palvelut, sekä online-pelit, joissa vähäinen datapakettien häviäminen ei aiheuta näkyvää tai kuuluvaa haittaa. Videolähetysten kannalta olennaisimmat asiat tiedonsiirrossa ovat datapakettien tahdistus, käytettävissä oleva kaista, sekä lähetysviive ja sen vaihtelu. UDP tarjoaa TCP:ta paremmat ominaisuudet etenkin viipeen ja kaistavaatiusten suhteen. (Häkkinen 2000, 15-16.)

**TAULUKKO 2: TCP-segmentin tietokentät (Casad & Willsey 1999, 101.)**

<b>Lähdeportti</b>			<b>Kohdeportti</b>		
<b>Järjestysnumero</b>					
<b>Kuittausnumero</b>					
<b>Pituus</b>	<b>Varattu</b>	<b>Kontrollibitit</b>		<b>Ikkuna</b>	
<b>Tarkistussumma</b>			<b>Kiireellisen datan osoitin</b>		
<b>Optiot</b>				<b>Täyte</b>	
<b>Tietoalue (pituus vaihtelee)</b>					

**TAULUKKO 3: UDP-datagrammin tietokentät (Casad & Willsey 1999, 106.)**

<b>Lähdeportti</b>		<b>Kohdeportti</b>	
<b>Pituus</b>		<b>Tarkistussumma</b>	
<b>Tietoalue (pituus vaihtelee)</b>			

Teknisesti TCP:n ja UDP:n olennaisimman eron muodostaa se, että TCP on yhteydellinen ja UDP yhteydetön protokolla. Yhteydellistä protokollaa käytettäessä jokaisesta onnistuneesti vastaanotetusta datapaketista lähetetään kuittaus takaisin lähettäjälle. Mikäli lähetetystä datapaketista ei saada kuittausa tietyn ajan kuluessa, se lähetetään uudestaan. Datan valvontaa ja mahdollista uudelleenlähettämistä kutsutaan vuon ohjaukseksi, joka on toteutettu TCP:n kohdalla jo itsessään protokollassa. Yhteydellinen protokolla määrittelee myös yhteyden avaus- ja katkaisutoiminnot. Yhteydetön protokolla sen sijaan lähettää datan yksisuuntaisesti päämääräänsä ilman, että se varsinaisesti ilmoittaisi vastaanottajalle, että dataa on tulossa. Myöskään vastaanotettava kone ei ilmoita lähettäjälle datan siirron onnistumisesta. UDP:ssa

voidaan kuitenkin valinnaisesti käyttää rajoitettua virheenkorjausta, jolla voidaan tarkistaa, ettei datapaketti ole korruptoitunut matkan varrella. Huolimatta rajallisesta virheenkorjauksesta, myös UDP:n avulla voidaan tehdä täysin luotettavia tiedonsiirtoratkaisuja. Tällöin virheenkorjaus tapahtuu dataa lähettävässä tai vastaanottavassa sovelluksessa, jolloin vuon ohjausta suoritetaan vastaanotetun datan perusteella ohjelmallisesti. Sovelluksen luonteesta riippuen virheenkorjaus voidaan jättää kokonaan pois. (Casad & Willsey 1999, 87-94.)

TCP ja UDP eroavat toisistaan olennaisesti myös niiden aikasidonnaisuuden osalta. TCP sisältää toiminnot datanpakettien uudelleenjärjestämiseen määränpäässä. Suorien videolähetysten kohdalla uudelleenjärjestely voi periaatteessa aiheuttaa viivettä, mutta tämän kaltaisia ongelmia varten on yleensä luotu varmistukset ohjelmallisesti, jolloin vastaanottava ohjelma osaa tarvittaessa pudottaa osan paketeista pois, vai vaihtoehtoisesti vastaanottoa voidaan puskuroida. Etenkin videopalveluissa lähetystä usein puskuroidaan, eli sitä otetaan vastaan puskurimuistiin useita sekunteja, jolloin mahdolliset aikaväristymät voidaan korjata käyttäjän huomaamatta. Sen sijaan joissakin aikakriittisissä palveluissa, kuten videokonferensseissa, usean sekunnin mittainen puskurointi on käytännössä mahdotonta tilanteen reaaliaikaisuuden vuoksi. Vaikka datapaketteja pudotettaisiinkin pois, putoamisia tarvitaan useimmiten kohtalaisen suuri määrä, jotta lähetyksessä voisi olla havaittavaa katkonaisuutta (O'Reilly 2005, 7-8). Mikäli lähetyksessä käytetään UDP-muotoa, pakettien saapumisjärjestystä ei muuteta protokollatasolla, eikä hävinneitä paketteja myöskään pyydetä uudelleen. Useimmissa sovelluksissa myös UDP-pakettien oikea järjestys tarkastetaan niiden sisällön perusteella ohjelmallisesti. Tällöin tulematta jääneet paketit voidaan pyytää erillisellä pyynnöllä, mikäli tarve vaatii. TCP-protokollaa hyödyntävässä sovelluksessa vastaava pyyntö tapahtuisi automaattisesti, ilman että datan saapumista tarvitsisi seurata ohjelmallisesti. (Casad & Willsey 1999, 93-94.)

Videolähetyksissä siirrettävä data koostuu pääasiassa itse lähetyksen sisällöstä. Liikenteeseen sisältyy kuitenkin myös paljon ohjaukseen tai lähetyksen toimivuutta ja laatua seuraavia viestejä, jotka vaativat tarkempia varmistuksia. Useat streaming-palvelut toimivatkin siten, että itse sisältö siirretään kevyemmässä UDP-muodossa, kun taas ohjaukset ja kuittaukset siirretään virheenkorjauksen sisältävässä

TCP-muodossa. Tällöin voidaan varmistaa se, että tärkeät ohjauskäskyt saadaan varmasti perille, vaikka itse sisältö siirretäänkin verkkoyhteyttä vähemmän rasittavan UDP:n kautta.

### 3.3.2 HTTP-protokolla

HTTP-protokolla on Internetissä hyvin yleisesti käytetty tiedonsiirtoprotokolla. Se perustuu pyyntö-vastaus -periaatteeseen, jossa asiakas lähettää palvelimelle pyynnön, joka sisältää tiedonsiirron kannalta olennaisten teknisten versiotietojen viestin, joka määrittelee pyydetyn datan. Palvelin lähettää asiakkaalle vastauksen, joka sisältää mahdollisen onnistumis- tai virheilmoituksen, sekä pyydetyn datan. HTTP-protokollalla ei ole jatkuvaa tilaa, vaan jokainen palvelimelle lähetetty pyyntö käsitellään omana tapahtumana. (RFC 2616)

HTTP-protokolla on ensisijaisesti suunniteltu web-sivujen lataamiseen ja se käyttää tiedonsiirtoon TCP-protokollaa. Tällöin se ei sovellu hyvin multimediaesitysten esittämiseen TCP:n mahdollisten synkronointiongelmien vuoksi. Ongelma kertaantuu myös siitä syystä, että TCP-protokolla säättää latausnopeutta suhteessa muuhun palvelimen kuormitukseen, jolloin streaming-dataa ei voida priorisoida muun vähemmän kaistaa vaativan liikenteen edelle. Tästä syystä HTTP:ta käytettäessä saattaa esiintyä katkoksia huomattavasti enemmän kuin oikeita streaming-protokollia käytettäessä. Lisäksi HTTP:lla on vain rajoittuneet ominaisuudet tiedostojen sisällön satunnaiseen lukemiseen, jolloin esimerkiksi videopalveluissa pikakelaustoiminnot ovat yleensä erittäin rajoittuneita ja useimmiten pikakelausmahdollisuus on vain siinä osassa tiedostoa, joka on jo latautunut katseluohjelman välimuistiin. HTTP ei myöskään tue multicast -tekniikkaa. (Microsoft 2007b.)

HTTP-protokollan suurin rajoitus streaming-palveluiden toteuttamisessa on se perusvaatimus, että pyydetyn tiedoston tulee sijaita kokonaisuudessaan palvelimella. Tämä rajaa pois mahdollisuuden käyttää HTTP-protokollaa suorien lähetysten tuottamiseen, koska näissä lähdetiedostoa luodaan reaaliaikaisesti. HTTP-protokollaa voidaan kuitenkin hyödyntää apumenetelmänä live-lähetysten tekemisessä, mikäli palomuuuri tai jokin muu tietoturvatarkkijä rajaa pois oikeiden streaming-protokollien suoran käytön. Tällöin HTTP:n avulla luodaan tunneloitu yhteys palo-



muurien läpi, jolloin streaming-protokollat voivat lähettää suoraa lähetystä tunneloidun yhteyden avulla. Näin ollen streaming-lähetys näyttää palomuurien kannalta normaalilta portin 80 kautta tapahtuvalta web-liikenteeltä, vaikka se todellisuudessa onkin tunneloitua streaming-liikennettä (Adobe 2002). Tunneloinnista aiheutuu kuitenkin lisärasitusta palvelimelle ja muille tietoliikennelaitteille, joka saattaa ilmetä asiakkaan ohjelmassa lisääntyneiden häiriöiden muodossa. Rajoituksistaan huolimatta HTTP-protokollan avulla voidaan luoda kustannustehokkaita on-demand -tyyppisiä palveluita, koska HTTP-streaming ei vaadi palvelimelta erityistä ohjelmistotukea, vaan se voi toimia tavallisen tiedostopalvelimen kaltaisesti.

### **3.3.3 Streaming-protokollat**

Formaatista riippuen streaming-istuntojen aloittamisen yhteydessä käytetään erilaisia sisällönkuvausprotokollia. Useiden streaming-formaattien kanssa määrittelyt tehdään SDP- eli Session Description Protocol:n mukaisesti. Protokollien käytössä esiintyy kuitenkin merkittäviä formaattikohtaisia eroja, ja osaa protokollista käytetään ainoastaan multicast-lähetysten tai videokonferenssien määrittelyyn. (RFC2327, 2.)

Streaming-tiedonsiirtoa varten on kehitetty HTTP:ta paremmin soveltuvia tietoliikenneprotokollia. Näistä yleisimmin käytössä ovat RTSP-, RTP ja RTCP-protokollat, joita voidaan käyttää erikseen, mutta joita useimmissa tapauksissa käytetään yhtä aikaa, koska ne täydentävät hyvin toisiaan. Streaming-protokollien toimintaperiaate eroaa HTTP:sta selkeimmin siinä, että niiden kautta liikennöitäessä käyttäjä lähettää palvelimelle pysyvän pyynnön, jolloin palvelin alkaa suorittaa haluttua kommentoa, kuten suoran videolähetyksen siirtoa käyttäjälle. Riippumatta siitä, käytetäänkö yhteydellistä vai yhteydetöntä ydinprotokollaa, streaming-protokollilla on HTTP:sta poiketen jatkuvasti määritelty tila, jolloin palvelin tallentaa jokaista käyttäjää varten oman istuntotunnisteen. Tällöin pysyvää TCP-yhteyttä palvelimelle ei tarvita, eikä jokaista datapakettia jouduta pyytämään erikseen, kuten HTTP:n yli liikennöitäessä. Streaming-protokollien avulla tiedonsiirron kontrolliliikenne voidaan pitää minimaalisena eikä HTTP:lle tyypillisiä kuittausviestejä tarvitse lähettää. Streaming-protokoliin on lisätty ominaisuuksia, joilla on pyritty löytämään ratkaisuja tyypillisimmille pakettipohjaisen tietoliiken-

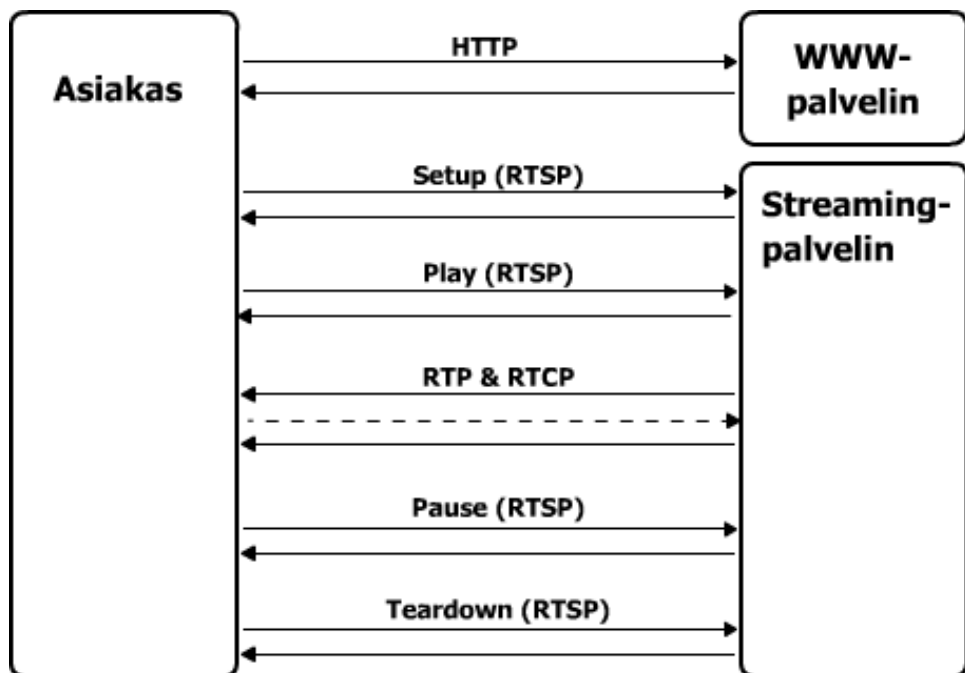
teen ongelmille, kuten lähetysviipeelle, epätasaisille lähetysväleille, pakettien häviämislle ja väärässä järjestyksessä tai useaan kertaan tuleville paketeille (Leon-Garcia & Widjaja 2000, 793).

RTSP eli Real-Time Streaming Protocol rakentuu HTTP-protokollan päälle. Siitä on pyritty luomaan HTTP:ta paremmin toimiva protokolla streaming-tietoliikenteelle osittain lisäämällä uusia ominaisuuksia ja osittain jättämällä pois HTTP:n heikkouksia streamingin kannalta. Protokollilla onkin useita yhteneväisyyksiä, mutta RTSP sisältää HTTP:sta poikkeavia menetelmiä tiedonsiirtoon, kuten out-of-band -määrittely, jonka avulla samalle käyttäjälle voidaan lähettää yhtäaikaaisesti useaa rinnakkaista mediavirtaa. Tällä tavalla on mahdollista lähettää ääni ja kuva erillisinä streameina, jolloin voidaan helpommin kontrolloida tilanteita, joissa saatavilla oleva kaistanleveys on rajallinen. RTSP ei kuitenkaan siirrä itse dataa, vaan se ainoastaan ohjaa datan siirtoa ja ylläpitää avattua yhteyttä. Suoria lähetyksiä seurattaessa protokollaa käytetään myös palvelimen ja asiakasohjelman synkronointiin yhteyttä avattaessa. Myös videon toistoon liittyvät komennot kuuluvat RTSP:hen, jolloin protokolla toimii ikään kuin etäkukosäätimenä. (RFC 2326, 4-12.)

RTP eli Real-Time Transport Protocol on varsinainen datan kuljetusprotokolla, joka on suunniteltu etenkin reaaliaikaisen datan siirtämiseen. Se tukee sekä unicast- että multicast-lähetystapoja ja sisältää toiminnot lähettävien datapakettien tyyppin tarkistukselle, järjestysnumeroinnille, aikaleimaamiselle sekä pakettien lähetyksen seurannalle. RTP ei kuitenkaan sisällä varsinaisia tarkistustoimenpiteitä, joilla esimerkiksi valvottaisiin pakettien saapumisjärjystä. Tarkistuksiin käytetään RTCP-protokollaa eli Real-Time Transport Control Protocol:aa, joka sisältää toiminnot myös lähetyksen synkronointiin ja datan priorisointiin. RTP:lla suoritettavassa datansiirrossa asiakkaan ja palvelimen välillä lähetetään tasaisin väliajoin RTCP-paketteja, joilla tarkkaillaan pakettien saapumista määränpäähensä ja joiden perusteella voidaan tehdä säätöjä lähetyksen asetuksiin. RTCP:n välittämien tietojen perusteella voidaan esimerkiksi ohjata mukautuvaa enkoodausta. RTCP-pakettien sisällöstä voidaan myös päätellä, montako osanottajaa istunnossa on mukana. Tiedon perusteella voidaan dynaamisesti säätää RTCP-pakettien lähetysintervallia, jolloin palvelimelle aiheutuva kontrollipakettien lähettämistä muodostuva kuormitus skaalautuu

käyttäjämäärän mukaisesti. RTCP:n suunnitteluperiaatteen mukaan kontrolliliikennettä saa olla maksimissaan 5% kokonaisliikenteestä. (RFC 1889, 3-4; Kaario 2002, 157-161.)

Tyypillisesti streaming-istunnon aloitus tapahtuu siten, että asiakas ottaa ensin yhteyden WWW-palvelimeen, joka ohjaa asiakkaalle linkin streaming-palvelimelle (Kuva 3). Tässä vaiheessa liikenne tapahtuu HTTP-protokollaa hyödyntäen. Palvelimelta saatu osoite avataan joko erillisessä ohjelmassa tai selaimen asennetussa laajennuksessa. Varsinaisen istunto alkaa RTSP:n setup-komennolla, jolla määritellään session asetukset ja tarvittavat muuttujat. Lähetettävän materiaalin siirto asiakkaalle aloitetaan play-komennolla, jonka jälkeen dataa siirretään RTP:lla ja siirtoa tarkkaillaan ja tarvittaessa säädetään RTCP:lla. Lähetysten seuraamisen lopettaminen tapahtuu RTSP:n kautta lähetettävällä pause-komennolla, jonka jälkeen yhteys puretaan teardown-komennolla. Yhteys voidaan katkaista myös palvelimen aloitteesta. Tällöin kyseessä on joko lähetysten loppuminen tai tilanne, jossa palvelin ei ole saanut asiakkaalta RTCP-kontrollipaketteja, jolloin tämän yhteyden katsotaan katkenneen. Teardown-komennon jälkeen asiakkaan streaming-istunto poistetaan palvelimelta.



KUVA 3: Tyypillinen protokollien käyttö streaming-istunnossa (LUT, 2000)

RTP-protokollaa käytettäessä jokainen mediavirta lähetetään omana streamina. Kaikki mediavirrat yksilöidään SSRC- eli synkronointitunnisteella, jonka perusteella saman tunnisteeseen omaavat virrat voidaan synkronoida keskenään. Vaikka mediavirtojen ajallinen synkronointi voidaan toteuttaa SSRC:n avulla, niiden varsinainen yhteenkuuluvuus todetaan RTCP-pakettien sisällöstä. Videolähetysten kannalta erillisistä video- ja audiovirroista on hyötyä etenkin silloin, jos kaistanleveyttä on rajallisesti saatavilla. Tällöin lähetyksen kannalta olennaisin mediavirta voidaan priorisoida muuta tärkeämmäksi. Videolähetyksissä useimmiten äänen osuus muodostuu kuvaa tärkeämmäksi, ellei kyse ole tekstipohjaisesta esityksestä. Mikäli kaistanleveyttä on vähän, videolähetysten sujuva seuranta voidaan toteuttaa priorisoimalla äänen kuvan edelle, jolloin ääni toistuu sujuvasti, mutta kuva päivittyy normaalia hitaammin. Erillisten datavirtojen avulla on sovelluksesta riippuen mahdollista jättää joku tietty mediavirta pois ja vastaanottaa vain haluttua osaa lähetyksestä. (Leon-Garcia & Widjaja 2000, 795.)

RTP mahdollistaa myös usean eri lähteestä tulevan mediavirran yhdistämisen. Tällöin käytetään järjestelmiä, joita kutsutaan streaming-miksereiksi. Näiden avulla on mahdollista yhdistää esimerkiksi useita äänistreameja yhdeksi lähetykseksi. Mikseri korvaa alkuperäisten lähetysten synkronointitunnisteiden, mutta lisää RTP-paketteihin listan alkuperäisistä lähteistä. Lisäksi RTP-pakettien kanssa voidaan käyttää muuntimia, jotka muokkaavat pakettien sisältöä, mutta jättävät muut tiedot, kuten synkronointitunnisteiden ennalleen. Muuntimia ovat esimerkiksi laitteet, jotka kääntävät sisällön formaatista toiseen tai jotka muuttavat multicast-lähetysten unicast-lähetykseksi. (RFC 3550, 6.)

### **3.3.4 Quality of Service ja RVSP**

Koska streaming-tietoliikenne on reaaliaikaista, sille on äärimmäisen kriittistä se, että kaistanleveyttä on riittävästi saatavilla koko lähetyksen ajan. Sen sijaan tavallisen tiedostonlatauksen kannalta ei ole merkittävää, vaikka välillä data siirtyisi hitaamminkin. Dataliikenteelle määriteltävät erilaiset kiireellisyysasteet mahdollistavat liikenteen priorisoinnin, jolloin kiireellisimmin liikenne voidaan lähettää muuta liikennettä aiemmin. Dataliikenteen priorisointia varten on kehitetty QoS eli Quality of Service -järjestelmä, jonka avulla tärkeemmän prioriteetin omaavat paketit priorisoidaan muun liikenteen edelle. (Kaario 2002, 224-226.) Quality

of Service käsittää 8 eri palvelutasoa, joiden perusteella verkon reitittimet muuttavat pakettien lähettämistä niiden tärkeyden mukaan (Taulukko 4).

**TAULUKKO 4: QoS-liikennetyypit ja prioriteetit IEEE802.1D-standardin mukaisesti. (Wikipedia 2007)**

<b>Prioriteetti</b>	<b>Liikenteen tyyppi</b>
<b>1</b>	<b>Background</b>
<b>2</b>	<b>Spare</b>
<b>0</b>	<b>Best Effort (oletus)</b>
<b>3</b>	<b>Excellent effort</b>
<b>4</b>	<b>Controlled load</b>
<b>5</b>	<b>Video</b> (viive ja viipeen vaihtelu <= 100ms)
<b>6</b>	<b>Voice</b> (viive ja viipeen vaihtelu <= 10ms)
<b>7</b>	<b>Network Control</b>

Streaming-videolähetykset sijoittuvat tärkeysasteeltaan joko tasolle 4 tai 5, jolloin ne priorisoidaan huomattavasti normaalia web-sivujen latausliikennettä tärkeämmäksi. Varsinainen QoS-varaus ja riittävän kaistanleveyden varaaminen palvelimelta ja reitittimiltä tehdään RSVP eli ReSerVation Protocol:n avulla. RSVP ei itsessään siirrä tai reititä dataa, vaan se toimii käytetyn reititysprotokollan päällä ja huolehtii ai-noastaan halutun QoS-prioriteettitaso ylläpitämisestä. Mikäli dataa siirrettäessä jokin reitin varrella oleva verkkolaite ei tue RSVP:ta, se siirtää paketit best-effort -eli paras mahdollinen nopeus -periaatteella. RVSP mahdollistaa myös multicast-varausten tekemisen. (Leon-Garcia Widjaja 2000, 695-696; RFC 2205, 3-5.)

## 3.4 Streaming-formaatit

### 3.4.1 Windows Media ja ASF

Microsoftin kehittämä ASF eli Advanced Systems Format rakentuu Windows Media Video ja Windows Media Audio -pakkausmuotojen ympärille. ASF ei itsessään ole multimediaa varten suunniteltu tiedosto- tai pakkausmuoto, vaan se toimii tiedostokäteenä, jolla on kehittyneet multimediaominaisuudet, jotka mahdollistavat sen lähettämisen streaming-tekniikalla. Vaikka ASF onkin tiedostokääre, eli se voi sisältää periaatteessa millä tahansa koodekillla pakattua materiaalia, useimmat ASF-pohjaiset ratkaisut kuitenkin perustuvat Windows Media -pakkaukseen. Tämä johtuu osittain siitä, että ASF sisältää patentoituja tekniikoita, joiden käyttämisestä sovelluksissa muodostuisi suuria lisensointikuluja. Toinen syy, minkä vuoksi ASF nivoutuu usein yhteen Windows Median kanssa, on sen DRM- eli digitaalinen tekijänoikeuksien valvonnan mahdollistavat toiminnot, jonka avulla on mahdollista tehdä rajoituksia sisällön käytölle ja mahdolliselle laittomalle kopioinnille. DRM-toiminnot ovat käytettävissä ainoastaan Windows Media -formaatin kanssa. Microsoftin oma suositus Windows Median toistoon on valmistajan oma Windows Media Player. (Microsoft, 2004.)

Windows Median toistaminen voidaan aloittaa joko suoralla yhteydelle palvelimelle tai vaihtoehtoisesti ASX-muotoisen metatiedoston kautta. ASX sisältää listan tiedostoista, jotka näytetään katsojalle. Toiminnon avulla on mahdollista toteuttaa mainospalveluja, joissa käyttäjälle näytetään aina ensin tietyt mainosvideot ja vasta näiden katsomisen jälkeen avataan yhteys itse videoon. (Microsoft, 2007c.)

Windows Median streamingissa hyödynnetään valmistajan omaa Microsoft Server -palvelinympäristöä, jossa ajetaan Windows Media Services -palvelua. Palvelimelle on mahdollista lähettää yhtä aikaa useaa erisuuruista bittivirtaa samasta lähteestä, jolloin katsojan on mahdollista valita tälle sopiva nopeus. Ohjelmisto mahdollistaa myös multicast-lähtämisen. Tietoliikenneprotokollien osalta Windows mediassa hyödynnetään lähtökohtaisesti RTSP-protokollaa, mutta myös Microsoftin oma, joskin vanhentunut MMS-protokolla, sekä HTTP ovat tuettuja. Lisäksi Microsoft on kehittänyt HTTP 1.0 ja 1.1 versioihin WMS-laajen-

nukset, joiden avulla streamingista saadaan toimintavarmempaa myös HTTP:ta käytettäessä. Ideaalitapauksessa streaming toteutetaan RSTP:n yli UDP-paketteina, mutta mikäli tämä ei tietoturvasyistä ole mahdollista, ohjelmisto hyödyntää Protocol Rollover- tekniikkaa, jonka avulla selvitetään paras mahdollinen siirtotapa. Protocol Rollover käy läpi seuraavat vaihtoehdot:

1. RTSPU, (RTSP UDP-muodossa)
2. RTSPT, (RTSP TCP-muodossa)
3. MMSU (MMS UPD-muodossa)
4. MMST (MMS TCP-muodossa)
5. HTTP. (Microsoft, 2007a.)

Käytettävän protokollan valintaa voi tietoturvatekijöiden lisäksi rajoittaa eri ohjelmistoversiot. Windows Media Server 2003 tai sitä uudemmat versiot eivät tue vanhentunutta MMS-protokollaa. Vastaavasti vanhemman Windows 2000 Server -palvelimen mediapalvelut eivät tue RTSP-protokollan kautta lähettämistä. Palvelimen ja asiakkaan ohjelmaversioista aiheutuvassa ristiriitatilanteessa Protocol Rollover valitsee käytettäväksi protokollaksi parhaan mahdollisen, jolloin useimmissa versioristiriitatapauksissa päätytään HTTP:n käyttöön. (Microsoft, 2007a.)

Microsoft tarjoaa formaatin käyttäjille ilmaisen enkoodaus- ja lähetysohjelmiston. Ainoa kustannus ohjelmasta on sen sidonnaisuus käyttöjärjestelmän lisenssiin. Ilmaisuudestaan huolimatta Windows Media Encoder sisältää laajat ominaisuudet, joiden avulla lähetys voidaan määritellä halutun kaltaiseksi. Lisäksi ohjelmaan on luotu ohjelmointirajapinta, jonka avulla siihen on mahdollista luoda ulkoisia laajennuksia ja sitä mukaa laajentaa ohjelman toimintoja. (Microsoft 2002b.)

### 3.4.2 RealMedia

RealNetworksin kehittämän RealMedian ensimmäisen versio julkaistiin jo vuonna 1997, joten RealNetworks on kohtalaisen pitkäikäinen toimija ja streaming-palveluiden kehittäjänä. Alun perin vain valmistajan omassa RealPlayer-ohjelmassa toiminut RealMedia -formaatti on alusta lähtien suunniteltu streaming-lähetyksiä varten ja itsessään formaatti on vain tiedostokääre, johon video- ja äänivirrat pakataan. Uusimmat versiot käyttävät kuvan pakkaamiseen RealNetworksin omaa RV40-koodekkia ja myös äänen pakkaamisessa käytetään valmistajan omia koodikkeja. RealMedia-tiedostot on myös mahdollista suojata DRM-tekniikalla tekijänoikeuksien valvomiseksi (RealNetworks 2007a). Windows Median tapaan myös RealMediaa käytettäessä voidaan hyödyntää metatiedostoja, joilla käyttäjän katseluohjelma ohjataan haluttuun kohteeseen. (RealNetworks 2006, 83.)

RealMedian streaming-palvelinratkaisut perustuvat Helix-palvelinohjelmistoon. Helix sisältää laajat toiminnot streaming-palveluiden toteuttamiseen, eikä se rajoitu ainoastaan RealMedian toistoon, vaan sen avulla voidaan lähettää valmistajan mukaan yli viittäkymmentä eri formaattia, kuten Windows Media, QuickTime, MP3, MPEG-4 sekä lähinnä mobiilituotteisiin suunniteltua 3GPP-muotoa H.263- ja H.264 -pakkauksella. Myös Helix-palvelimella voidaan lähettää useaa eri bittivirtaa asiakkaan yhteysnopeuden mukaan. Ominaisuudesta käytetään nimitystä SureStream, ja sillä pyritään takaamaan sujuva lähetys katsojan yhteysnopeudesta riippumatta. SureStreamin avulla toteutettu nopeuden säätö ei rajoitu vain istunnon avaamishetkeen, vaan asiakkaan yhteyttä voidaan tarkkailla jatkuvasti, jolloin nopeutta voidaan säätää reaaliaikaisesti kuormituksen mukaan. Lisäksi Helix tukee myös multicast-lähetystekniikkaa. Helix käyttää lähetysten hallintaan RTSP-protokollaa, mutta datan siirtoon käytetään oletusarvoisesti Realin omaa RDT- eli Real Data Transport -protokollaa. Myös RTP-siirto on tuettu, koska RDT:n käyttö saattaa olla tietoturvasyistä rajoitetumpaa. Aivan kuten Windows Mediaa, myös RealMediaa voidaan lähettää HTTP:n kautta, mikäli tilanne niin vaatii. Streaming-istuntojen luomisessa ja määrittelyssä RealMedia hyödyntää SDP-protokollaa. Kaupallisten versioiden lisäksi RealNetworks on julkaisut Helix-palvelimesta myös avoimeen lähdekoodiin perustuvan Helix DNA -version. (RealNetworks 2005.)



RealMedia-muotoisen videon lähettäminen palvelimelle perustuu Real-Producer -enkoodausohjelmaan. Ilmainen RealProducer Basic sisältää perustoiminnot lähetyksen toteuttamiseksi, mutta käytännössä vähintään laajempi tuotanto vaatii maksullisen Plus-version. Lisäksi RealNetworks tarjoaa valmiita Niagara-enkoodauslaitteistoja, jotka ovat käytännössä streamingin kannalta sopivista komponenteista rakennettuja räkkipalvelinkoneita. RealProducerin uusimman version keskeisiä uudistuksia ovat lähetyksiä pienentävät tekniikat. TrueLive-tekniikka vähentää lähetyksen puskurointia ja optimoi RTSP-pakettien käsittelyä, jolloin lähetyksen viive saadaan pudotettua jopa kahteen sekuntiin. RealProducer sisältää myös toimintoja, joilla lähetyksen enkoodauksen laatua voidaan säätää suhteessa enkoodauslaitteen konetehtoon. Heikkotasoinen enkoodaus mahdollistaa pienemmän lähetyksiin kuvanlaadun kustannuksella, kun taas parempilaatuinen kuva vaatii enemmän enkoodausaikaa. Säätöjä voidaan myös tehdä täysin automaattisesti, jolloin ohjelma tarkkailee enkoodauslaitteen suorituskykyä ja säätää laatua sen mukaisesti. (RealNetworks, 2007b.)

### **3.4.3 QuickTime**

QuickTime on Applen kehittämä tiedostokääre, joka julkaistiin jo 1991. Ensimmäiset versiot eivät kuitenkaan sisältäneet vielä streaming-toimintoja, vaan ne tulivat vasta QuickTime 4 -versiossa, joka julkaistiin 1999. Tiedostomuoto ei rajoitu ainoastaan videon ja äänen pakkaamiseen, vaan se voi sisältää myös esimerkiksi tekstiä tai interaktiivisia panoraamakuvia. Pääasiallinen käyttökohde on kuitenkin multimedia, etenkin videotiedostot. Valtaosa QuickTime-formaattia hyödyntävistä videototeutuksista käyttää Applen omaa QuickTime Movie – pakkausmenetelmää, mutta myös MPEG-4 -formaattit ovat laajalti käytössä. QuickTime-muotoisen multimedian toistoon valmistaja suosittelee tämän omaa QuickTime Playeria. (Wikipedia 2007a.)

QuickTime Streaming Server -palvelut sisältyvät Mac OS X Server 10.4 -palvelinkäyttäjärjestelmään. Palvelin mahdollistaa lähettämisen joko QuickTime Movie-, MPEG-4- tai 3GPP-formaatissa. Palvelimen uusimmat versiot mahdollistavat myös H. 264-pakatun HD-tasoisien materiaalin lähettämisen. RealNetworksin tapaan myös Apple on julkaissut palvelinohjelmistostaan avoimeen lähdekoodiin perustuvan ver-

sion. Vaikka ilmainen Darwin Streaming Server on tekniikaltaan lähes täysin kaupallista versiota vastaava, se ei kuitenkaan sisällä QuickTime Streaming Serverin käyttöliittymäkomponentteja. (Apple 2007a.)

QuickTime-formaatti käyttää oletuksena RTSP-protokollaa yhteyksien avaamiseen ja määrittelyyn. Data siirretään RTP-protokollalla. QuickTime tukee myös Applen omaa Reliable UDP-tekniikkaa, joka sisältää UDP-standardia kehittyneemmät virheenkorjausmenetelmät, jolloin lähetyksen toimintavarmuus paranee. Useimpien streaming-formaattien tapaan QuickTime mahdollistaa HTTP-streamingin tietoliikenneportti 80:n kautta, jolloin lähetyksen kannalta liian kireät palomuurien asetukset voidaan kiertää. RealMedian tapaan QuickTime käyttää standardin mukaista SDP-protokollaa istuntojen määrittelyihin. (Apple 2007b.)

Myös QuickTimen lähetyks- ja enkoodausohjelmisto sisältyy OS X Server -käyttöjärjestelmään. QuickTime Broadcaster mahdollistaa monimutkaisetkin lähetyksasetukset, kuten multicast-lähetykset ja lisäksi se kykenee hakemaan reaaliaikaista tilastotietoa palvelimelta esimerkiksi katsojamääristä ja suorittimen käyttöasteesta. Broadcaster:n ohella OS X Server sisältää myös QuickTime Streaming Server Publisherin, jonka avulla on mahdollista hallita median siirtoa streaming-palvelimelle sekä luoda soittolistoja, joiden avulla voidaan automatisoida Broadcaster:n lähetyksistä. (Apple 2007c.)

#### **3.4.4 Flash**

Nykyään Adoben omistaman Macromedian kehittämä Flash-teknologia on muodostunut yhdeksi suosituimmista multimedian esitystekniikoista Internetissä. Vaikka useimmat Flashia hyödyntävät web-sovellukset ovat vain pieniä selaimessa ajettavia sovelluksia, Flashilla voidaan toteuttaa myös raskaampia asiakas-palvelin -tyyppisiä sovelluksia, kuten videopalveluita. Datansiirtoa varten Flashin rinnalle on kehitetty Flash Media Server, jonka kautta on mahdollista tehdä myös suoria videolähetyksiä. Flashin versiosta 8 alkaen videoiden oletusmuotona käytetään FLV- eli Flash Video -formaattia. Videon pakkausta varten Flashin mukana tulee Sorenson Spark ja On2 VP6 -koodekit, joista jälkimmäinen on huomattavasti kehittyneempi ja tehokkaampi. On2 VP6 tuottaa samalla bittivirralla Sorensonia huomattavasti parempaa kuvaa, ja li-

säksi videoon on mahdollista enkoodata alpha-kanava, jonka avulla on mahdollista toteuttaa läpinäkyvällä taustalla varustettuja videoita, jolloin toisto-ohjelma on helpompi upottaa haluttuun ympäristöön. FLV-formaatissa ääni pakataan MP3-pakkausmenetelmällä. Muiden formaattien tapaan myös FLV-video voidaan suojata DRM-menetelmillä. (Streamingmedia.com, 2006.)

Muista formaateista poiketen Flash ei käytä yleisiä avoimia protokollia, vaan ainoastaan Flash Media Serverin käyttämää Real Time Messaging Protocol eli RTMP-protokollaa, jota käytetään käytännössä kaikkien tiedonsiirtoon palvelimen Flash Player -ohjelman välillä. Flash ei varsinaisesti sisällä erityisesti videolähetysten katsomista varten kehitettyä asiakasohjelmaa. Flashin avulla on kuitenkin helppoa toteuttaa joko selaimessa tai erillisessä Flash Playerissa toimiva web-sovellus, jonka kautta streaming-lähetystä on mahdollista seurata, koska Flash sisältää valmiita videon vastaanottamiseen suunniteltuja komponentteja. Palvelimelta siirrettävän datan muotoa ei ole rajattu ainoastaan video- tai audiomuotoon, vaan Flashilla on mahdollista toteuttaa laajoja verkkosovelluksia, joihin voidaan videokuvan lisäksi tuoda myös muuta dataa. Tämä mahdollistaa laajempien palvelukokonaisuuksien kehittämisen. (O'Reilly 2005, 4-5.)

Myös Flash sisältää Windows Median Protocol Rollover-tekniikan kaltaisen protokollatarkistuksen. RTMP-protokolla käyttää oletuksena tietoliikenneporttia 1935. Mikäli kyseinen portti ei ole käytettävissä, yhteyttä voidaan yrittää muiden streaming-formaattien tapaan yleisemmin käytettyjen porttien kautta. Mikäli RTMP ei ole käytettävissä, siirrytään käyttämään joko RTMPT- tai RTMPS-protokollaa. Näistä ensimmäinen on tunneloitu yhteys HTTP:n yli portin 80 kautta, ja jälkimmäinen on niin ikään tunneloitu, mutta lisäksi salattu yhteys portin 443 kautta HTTPS-protokollan avulla. (Adobe 2002) Videolähetysten kannalta epäedullisesti RTMP ei sisällä minkäänlaisia multicast-toimintoja, eikä se tue datan lähettämistä UDP-muodossa. Tästä johtuen Flash on verkon kuormituksen kannalta huomattavasti muita formaatteja raskaampi. (O'Reilly 2005, 7-8.)

Videon lähettäminen Flash Media Server -palvelimelle tapahtuu Flash Media Encoderin kautta. Adobe tarjoaa ohjelman ilmaiseksi, mutta sen käyttö on sidottu Flash Media Serverin lisenssiin. Ohjelma on toimintoiltaan kohtalaisen rajoittunut ja tarjoaa vain perustoiminnot lähetyksen tekemiseen. Flash kuitenkin mahdollistaa katsojan sovelluksen laajan kustomoinnin, jolloin mahdolliset lisäominaisuudet voidaan toteuttaa katsojan sovellukseen. (Adobe 2007.)

### 3.4.5 Streaming-formaattien ominaisuuksien vertailu

Eri streaming-formaateissa on joitakin keskeisiä eroja toisiinsa nähden. Erot painottuvat videokuvan laadun sijasta formaattien teknisiin ratkaisuihin, koska nykyiset kuvanpakkausmenetelmät tuottavat lähes yhtä hyvää laatua. Tiedostokääreinä toimivat formaatit saattavat käyttää keskenään täysin samoja kuvanpakkausmenetelmiä, jolloin niiden vertaaminen keskenään ei olisi mielekästä.

Katsojan kannalta suurimman eron formaattien välillä muodostaa lähetyksen seurannan mahdollistavat katseluohjelmat. Käytännössä kaikki formaatit vaativat oman erillisen toisto-ohjelman asennuksen, mutta osa ohjelmista kykenee toistamaan myös muita formaatteja (Taulukko 5). Pääsääntöisesti kuitenkin eri formaattien etenkin streamingiin kohdistuvat erikoistoiminnot toimivat kunnolla ainoastaan näiden omissa toisto-ohjelmissa. (Wikipedia 2007b.)

**TAULUKKO 5: Toisto-ohjelmien tukemat videoformaattit (Wikipedia 2007)**

Ohjelma	Windows Media	RealMedia	QuickTime	Flash
Windows Media Player 9	•	•	•	
RealPlayer	•	•	•	
QuickTime			•	
Flash Player				•

Taulukosta on nähtävissä, että Windows Media Player ja RealPlayer kykenevät toistamaan myös toistensa ja QuickTimen formaatteja. Tämä ei kuitenkaan ole poikkeuksellista, koska esimerkiksi avoimeen lähdekoodiin perustuva VLC Player kykenee toistamaan kaikkia taulukon videoformaatteja (VideoLan, 2007). Flash-videotiedostojen toistaminen ei onnistu suoraan yhdelläkään valmistajien omista ohjelmista, ei edes Adoben omalla Flash Playerilla. Toistaminen kuitenkin onnistuu kaikilla vertailun toisto-ohjelmilla, mikäli video tuodaan ohjelmaan valmiin Flash-esitykseen, eli SWF-tiedoston kautta, joiden avaamiseen myös Flash Player on suunniteltu.

Streaming-formaattien valmistajien omat toisto-ohjelmat sisältävät merkittäviä eroja myös niiden toimivuudessa eri käyttöjärjestelmissä. (Taulukko 6).

**TAULUKKO 6: Toisto-ohjelmien tukemat käyttöjärjestelmät (Wikipedia 2007)**

Ohjelma	Windows	Mac OS X	GNU/Linux	BSD Unix	Solaris
<b>Windows Media Player 9</b>	•	•			
<b>RealPlayer</b>	•	•	•	•	•
<b>QuickTime</b>	2000/XP/ 2003/Vista	•			
<b>Flash Player</b>	•	•	•	•	•

Taulukosta on luettavissa, että tuessa eri käyttöjärjestelmille RealPlayer ja Flash Player omaavat kaikkein laajimman tarjonnan. QuickTimen kaikki toiminnot ovat käytettävissä ainoastaan Applen omassa käyttöjärjestelmässä ja tuki Windowsille rajoittuu ainoastaan uusimpiin versioihin. Taulukosta on helposti tulkittavissa myös käyttöjärjestelmien valmistajien uskollisuus omia formaatteja kohtaan. (Wikipedia 2007b.)

Myös ohjelmien käyttämissä tiedonsiirtoprotokollissa on nähtävissä selkeitä keskinäisiä eroja (Taulukko 7).

**TAULUKKO 7: Streaming-formaattien tukemat tiedonsiirtoprotokollat (Wikipedia 2007)**

Ohjelma	HTTP	RTSP	MMS	RTMP
Windows Media Player 9	•	•	•	
RealPlayer	•	•	•	
QuickTime	•	•		
Flash Player	•			•

Taulukko osoittaa, että Windows Media ja RealMedia ovat myös tiedonsiirtoprotokollien suhteen kaikkein universaaleimpia. Suurimman eron muihin ottaa Flash, joka ei tue edes RTSP-protokollaa. Ainoa yhteneväisyys kaikkien formaattien kesken on se, että tarvittaessa tiedonsiirto voidaan tehdä myös streamingille epäedullisen HTTP-protokollan kautta. (Wikipedia 2007b.)

## **4 RAPORTTI ANYCAST-TUOTANNOSTA**

### **4.1 Tuotannon lähtökohdat ja tarkoitus**

Lahden Tiede- ja Yrityspuisto Neopolissa järjestettiin 8.-19.1.2007 Vesijärvi II -järvenhoitoprojektin tiede- ja kunnostuspäivät. Projekti on koko olemassaolonsa ajan panostanut vahvasti tiedotukseen ja koko kaksiviikkoinen tapahtuma oli päätetty lähettää suorana streaming-lähetyksenä projektin Internet-sivuille. Tapahtuman ensimmäisen viikon pääpaino oli paikallisten päättäjien ja vaikuttajien ympäristöaiheisissa haastatteluissa. Paikalla oli myös useita päijäthämäläisiä kansanedustajia. Viikko oli jaettu erilaisiin teemapäiviin, kuten kuntapäivä, eduskuntapäivä ja opiskelijapäivä, joiden teemojen mukaan jokaiselle päivälle räätälöitiin omanlainen ohjelmapaketti. Jälkimmäisellä viikolla pääpaino sijoittui loppuviikon tiede-, kunnostus- ja viestintäpäiville, jolloin järjestettiin järvien kunnostukseen liittyviä seminaareja ja keskustelutilaisuuksia. Tuotannon kannalta suurimmat haasteet sijoituivat toiselle viikolle, jolloin tilaisuudet lähetettiin suorana monikameratuotantoina. Lisäksi koko kaksiviikkoisen ajan seurattiin paikallisten kalastajien pyydystalkoita Neopolin Expo-hallissa.

Streaming-tuotannon tarkoituksena oli saada kustannustehokasta julkisuutta projektille. Vesijärvi II -projekti päättyi virallisesti vuoden 2006 lopussa ja sen jatko ja sitä mukaa Vesijärven asianmukainen kunnostus on vaakalaudalla. Tehokkaalla viestittämisellä pyrittiin tuomaan asia niin tavallisten kansalaisten kuin myös päättäjien tietoon.

### **4.2 Tuotannossa käytetty laitteisto**

Streaming-lähetysten toteutus rakentui laitteiston osalta Sony Anycast Station AWS-500G -laitteen ympärille (Kuva 4). Anycast on pieniin ja keskisuuriin videotuotantoihin suunniteltu helposti liikuteltava tuotantoyksikkö. Se sisältää kuva- ja äänimikserin, logo- ja tekstityssovellukset, luma- ja chromakey-toiminnot sekä laajat liitännäismahdollisuudet erilaisiin kameroihin tai muihin kuva- ja äänilähteisiin. Streaming-lähetyksiä varten Anycastilla on mahdollista kääntää video RealMedia-formaattiin

ja laite itse voi toimia joko lähettimenä tai palvelimena tuotannon suuruudesta riippuen. Lähettämisen lisäksi materiaali voidaan samanaikaisesti tallentaa RealMedia-muodossa video-on-demand -palvelua varten. Videomateriaali voidaan myös tallentaa DV-laatusena joko ulkoiselle kovalevylle, nauhurille tai jollekin muulle tallennusmedialle.



**KUVA 4: Anycast Station AWS-500G (dimagemaker.com)**

Tuotannossa käytettiin useita erilaisia kuvalähteitä. Kolme digitaalivideokameraa kytkettiin suoraan Anycastiin FireWire-liitäntöjen kautta. Lähetyksissä käytetyt koko kuva-alan peittävät esitykset tai kuvasarjat näytettiin kannettavalta tietokoneelta, joka oli kytkettynä Anycastiin VGA-liitännällä. Myös suurin osa uusintoina näytetyistä videoista toistettiin kannettavan tietokoneen kautta. Koska Anycastiin on mahdollista kytkeä vain kuusi erillistä kuvalähdettä, siihen kytkettiin s-video -liitännällä erillinen Datavideo SE-500 -videomikseri, johon oli kytkettynä kaksi kameraa, sekä DV-nauhuri ja DVD-soitin. Tällä tavalla ketjutamalla Anycastiin saatiin tuotua yhteensä yhdeksän erillistä kuvalähdettä normaalin kuuden sijaan. Usean eri kuvalähteen sijaitseminen yhdessä Anycastin oman videomikserin sisääntulossa teki laitteiston käytöstä joiltakin osin monimutkaisempaa, joten ulkoiseen mikseriin päätettiin kytkeä sellaiset kamerrat, joita tarvittiin tuotannossa muita vähemmän.



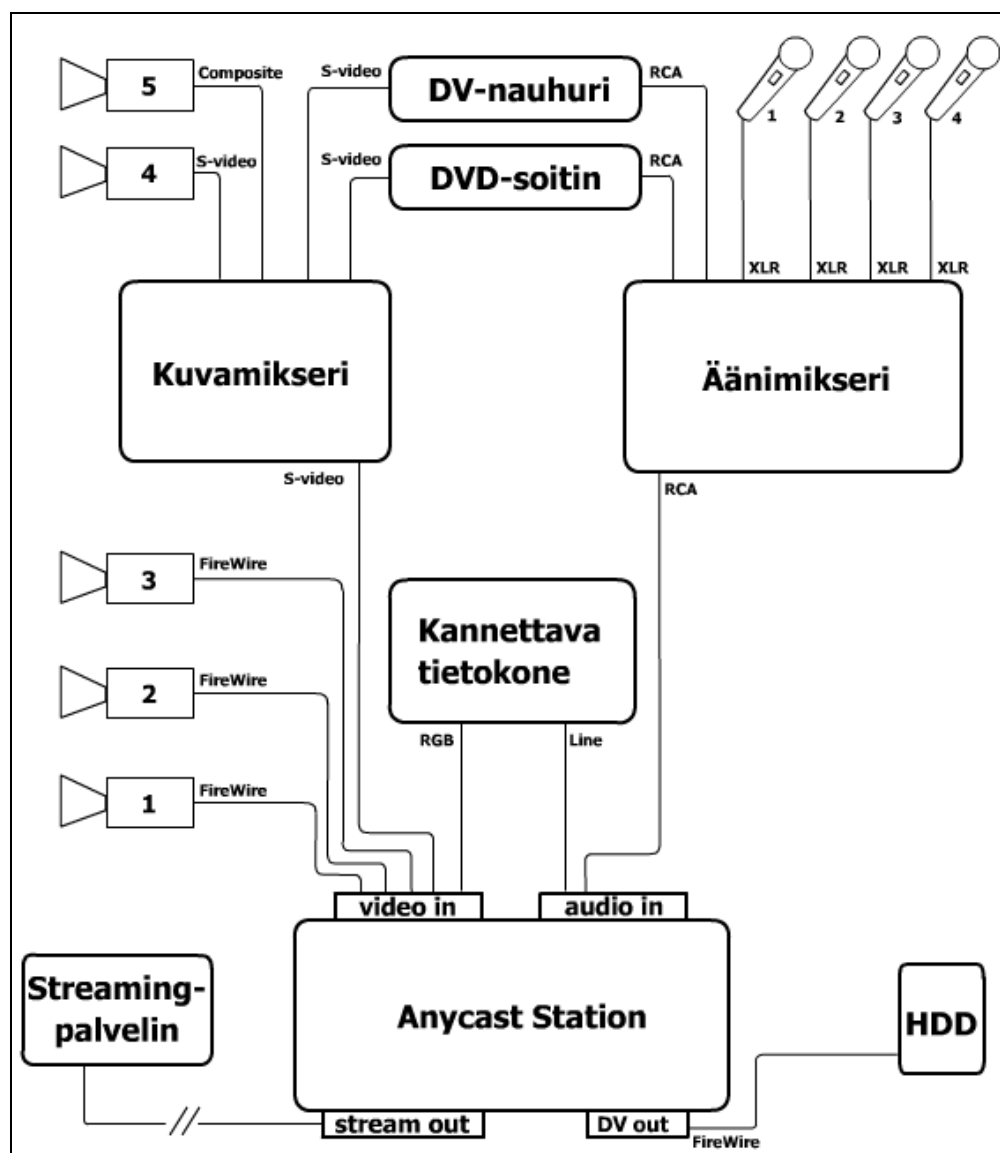
Tuotantoa varten ei ollut saatavilla identtisiä kameroita. Kaksi kame- roista oli Sony DSR-500 -tuotantokameroita ja niiden lisäksi käytettiin kolmea pienempää DV-kameraa, joista yksi oli pääsääntöisesti kiin- teässä paikassa oleva kylmä kamera ja kahdella kuvattiin insert-mate- riaalia päivän tapahtumista. Pienistä kameroista kaksi oli Sonyn ja yksi Panasonicin valmistama. Panasonicista kuitenkin luovuttiin tuotannon myöhemmässä vaiheessa, koska sillä ei sen vanhemman tekniikan an- siosta ollut mahdollista kuvata aitoa laajakuvaa. Toisen viikon keskus- telutilaisuuksissa myös pieniä kameroita käytettiin suorassa lähetyk- sessä. Tilaisuuksia taltioitaessa kaikki kamerat olivat jalustoilla, jotta kuvasta saataisiin mahdollisimman vakaata. Toinen isommista Sonyn tuotantokameroista asetettiin jalustalle, jossa on renkaat alla. Tämä mahdollisti yksinkertaisten kamera-ajojen tekemisen ja sitä mukaa ku- vallisesti elävämmän lopputuloksen muuten varsin rauhallisille tilai- suustaltioinneille. Normaalien DV-kameroiden lisäksi käytössä oli So- nyn BRC-300 robottikamera, jota ohjattiin RM-BR300 kauko-ohjaimella VISCA -liitännän kautta (Kuva 5).



**KUVA 5: RM-BR300 kauko-ohjain ja BRC-300 robottikamera (Videobotics & blogmaster.fr)**

Äänilaitteiston osalta tuotanto oli kohtalaisen yksinkertainen. Anycastiin kytkettiin erillinen äänimikseri, johon tuli äänet neljästä mikrofoniasta, sekä DV-nauhurista ja DVD-soittimesta. Seminaareissa ja keskusteluti-

laisuuksissa ääni lähetettiin mikserin kautta myös salin puolella oleviin kovaäänisiin. Muista äänilähteistä poiketen kannettavan tietokoneen äänilinjat kytkettiin suoraan Anycastiin, jolloin koneelta toistettavien videoiden äänentasausta pystyttiin helposti säätämään muista äänilähteistä riippumatta. Koska tuotanto sisälsi myös monikameralähetystä, siinä hyödynnettiin Datavideon ITC-100 intercom-laitteistoa, jonka kautta lähetysten ohjaajan oli mahdollista käskyttää kameramiehiä. Intercom-laitteisto sisälsi myös tally-merkkivalot, joiden avulla kameramiesten olisi ollut mahdollista tarkkailla, milloin minkäkin kameras kuvaa menee lähetykseen. Tally-valoja ei kuitenkaan pystytty käyttämään, koska niitä ei saatu kytkettyä Anycastiin.



KUVA 6: Tuotannon laitekaavio (Lehikoinen 2007)

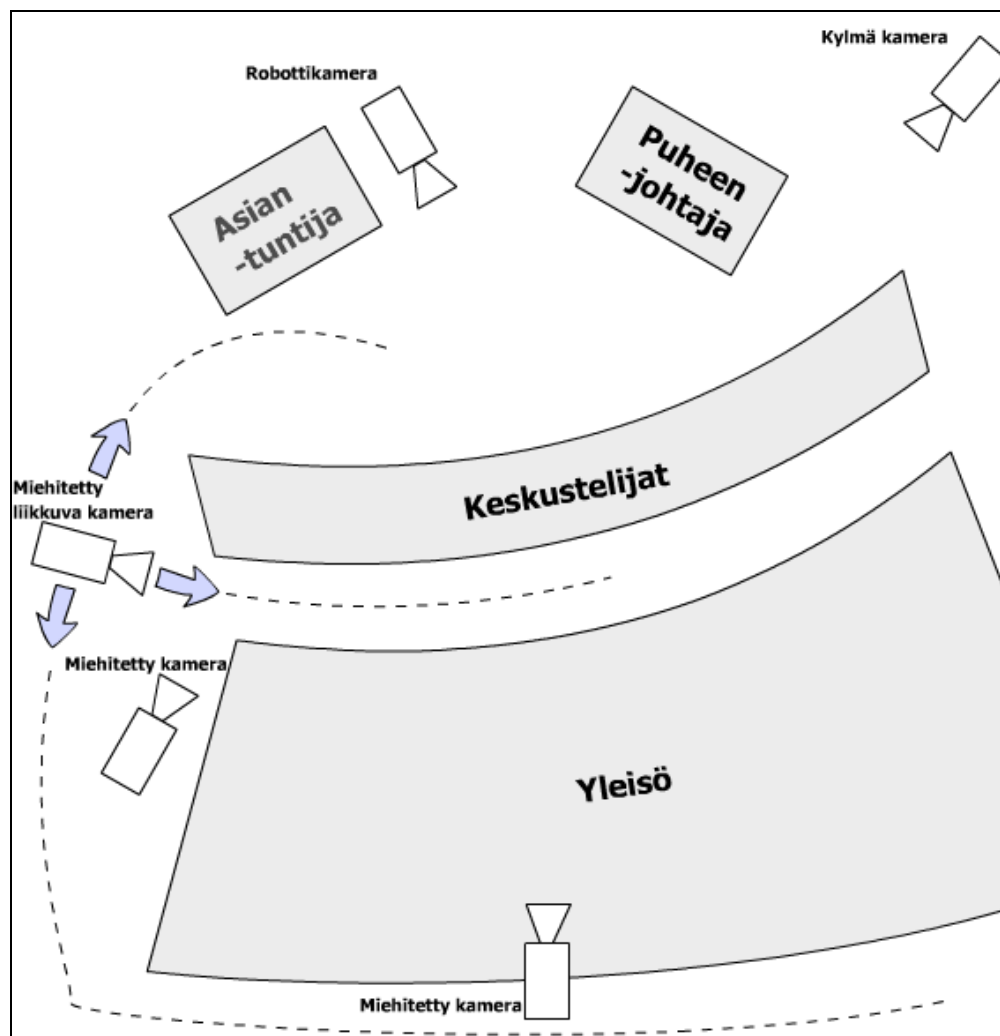
Streaming-lähetys toteutettiin Anycastin kautta 384Kb/s -bittivirralla ja 360 x 180 -laajakuvaresoluutiolla 15 kuvan sekuntinopeudella. Tuotantopaikalta streaming-lähetys siirtyi vain Anycastille dedikoidun laajakaistayhteyden kautta streaming-palveluntarjoajan palvelimelle ja sieltä eteenpäin lähetyksen katsojille. Lähetyksen puskuroinnista johtuen lähetyksessä oli havaittavissa noin 11-13 sekunnin viive, mutta käytännössä voidaan kuitenkin puhua suorasta lähetyksestä. Projektin verkkosivuilla olevassa RealPlayer-ikkunassa video venytettiin 640 x 360 pikselin kokoiseksi, jolloin kuvasta saatiin suurempi ja katsojan kannalta miellyttävämpi. Kuvan skaalaaminen teki siitä myös hieman pehmeämmän, mutta tästä ei ollut merkittävää haittaa, koska pienehköllä bittivirralla pakatussa kuvassa ei muutenkaan ollut erityisen tarkkoja yksityiskohtia.

### **4.3 Tuotannon kulku ja lopputulos**

Tuotannon ensimmäisellä viikolla tyypillinen päivä sisälsi kohtalaisen paljon arkistomateriaalin ja paikan päällä nauhalle kuvatun materiaalin toistoa, sekä muutamia suorana lähetyksenä toteutettuja haastatteluja. Jokaiselle päivälle oli myös koottu tiedepaketti projektiin liittyvistä aihealueista. Kaikki päivän aikana näytetty materiaali tallentui ulkoiselle kovalevylle, josta se saatiin nopeasti siirrettyä jatkokäsittelyyn tiivistettäväksi. Päivän päätteeksi lähetykseen jätettiin pyörimään päivän ohjelma tiivistettynä kokonaisuutena, josta oli karsittu päivän teeman kannalta epäolennaiset vaiheet pois. Ensimmäisen viikon päätteeksi koko viikon tiivistettyä ohjelmaa lähetettiin viikonlopun yli uusintana. Materiaalin tiivistysvaiheessa ohjelmien alkamisajoista ja kestoista tehtiin tarkka lista, jonka avulla niiden esitysajat ilmoitettiin projektin verkkosivuilla.

Toisen viikon alku toteutettiin tuotannon osalta ensimmäisen viikon kaltaisesti. Loppuviikko sen sijaan painottui suorana lähetyksenä toteutettaviin monikameralähetyksiin seminaari- ja keskustelutilaisuuksista, joita järjestettiin kolmena eri päivänä. Kuvauksissa käytettiin kaikkia viittä kameraa, joista kolmea pidettiin jatkuvasti miehitettyinä (Kuva 7). Seminaarit olivat pääsääntöisesti neljän tunnin mittaisia, jaettuna kahteen kahden tunnin tilaisuuteen. Toisen viikon päätyttyä sen ohjelmaa lähe-

tettiin vielä uusintana viikonlopun ajan. Varsinaisen tuotannon päätyttyä kaikesta sen aikana kuvatusta materiaalista koottiin kattava tiivistelmä, jota lähetettiin vielä viikon ajan. Lisäksi kuvatuista haastatteluista koottiin runsas määrä videoleikkeitä, jotka siirrettiin projektin verkkosivuille katsojien ladattavaksi myöhempää katselua varten.



**KUVA 7 : Kameroiden paikat tilaisuuksia taltioitaessa (Lehikoinen 2007)**

Tuotanto onnistui täyttämään sille asetetut viestinnälliset tavoitteet kohtalaisen hyvin. Kahden viikon aikana Vesijärvi II -projektin verkkosivuille oli kertynyt sivunlatauksia yli 13000 kappaletta ja uniikkeja kävijöitä lähes 2000. Projektilla saatiin myös näkyvyyttä useissa sanomalehdissä sekä useilla verkkosivuilla aina valtakunnallisia tiedotusvälineitä myöten (YLE 2007). Tämä on huomattava lisäys sivuston normaaliin liikenteeseen.

teeseen verrattuna. Streaming-palvelimen tilastoihin kertyi kahden ensimmäisen viikon aikana yhteensä 479 uniikkia IP-osoitetta. Lukema ei kuitenkaan kerro suoraan katsojamäärää, koska esimerkiksi isojen yritysten palomuurien takaa tulevat yhteydet näkyvät yhtenä osoitteena, vaikka katsojia olisikin useita. RTSP-yhteyksiä palvelimelle otettiin yhteensä lähes 3300 kappaletta. Palvelimen kaistankäyttö ei kuitenkaan käynyt maksimirajoilla missään vaiheessa tuotantoa. Yhteensä lähetystä katsottiin yli 740 tuntia josta keskimääräinen katseluaika oli noin 11 minuuttia. Lukemaa pienentävät huomattavasti vain muutamia sekunteja kestäneet yhteydet, jotka ilmeisesti päättyivät virhetilanteeseen. Tuotannon aikana dataa siirtyi yli 570 gigabittia. Siirretyn datan määrä olisi ollut todennäköisesti huomattavasti suurempi, ellei tuotannon edetessä lähetyksen bittivirtaa olisi jouduttu pudottamaan 256 kbit/s -nopeuteen. Nopeuden laskeminen johtui useilla katsojilla esiintyneistä ongelmista, kuten lähetyksen katkeamisista tai jatkuvasta kuvan nykimisestä. Syitä ongelmiin ei ehditty kovinkaan tarkkaan tutkia, mutta todennäköisesti useimpien katsojien ongelmat johtuivat hitaista laajakaistayhteyksistä, jotka riittivät ainoastaan 256 kb/s -nopeuden seurantaan. Tilastoihin ei sisälly tuotannon kolmatta viikkoa, jolloin lähetettiin aiempien viikkojen ohjelmaa uusintana.

Viestinnän toimivuutta häiritsi epätietoisuus aikatauluista. Koska osa haastatteluihin kutsutuista henkilöistä jättivät tulematta paikalle eivätkä ilmoittaneet asiasta, aikatauluja jouduttiin muuttamaan kesken tuotannon. Tästä johtuen oli lähes mahdotonta tiedottaa tarkasti tulevan ohjelman sisällöstä, ja projektin verkkosivuilla olikin vain suuntaa-antavia tietoja päivän ohjelmasta. Eräs tuotannon mielenkiintoisimmista viestintään liittyvistä havainnoista oli se, että katsojat eivät juurikaan antaneet painoarvoa lähetettävän kuvan laadulle, vaan tärkeintä oli se, että kuva ylipäättään näkyi. Tätä voidaan osaltaan selittää uudella tekniikalla, joka saattaa hämmästyttää katsojan. Välttämättä uusille tekniikoille ei edes osata asettaa erityisiä laatuvaatimuksia, koska itse tekniikka on vielä vierasta. Lähetyksen laadusta tulikin pääosin hyvää palautetta koko tuotannon ajan.

Teknisen toteutuksen osalta tuotanto onnistui tyydyttävästi. Ennen tuotantoa tehdyt kaaviot tuotantolaitteiston kytkennöistä osoittautuivat toimivaksi ratkaisuksi, eikä itse laitteiston kanssa ilmennyt ongelmia muutamia rikkiäisiä kaapeleita lukuun ottamatta. Sen sijaan ajoittaisia

hankaluuksia esiintyi erityisesti lähetyksen ohjaajan työmäärän suhteen, jonka tehtäviin kuului kameramiesten käskyttäminen, kuvamikkusaus, tekstiplanssien näyttäminen, sekä tarvittaessa äänenvoimakkuuksien säätäminen. Myös erillisen videomikserin käyttö Anycastin oman ohella yhtä aikaa aiheutti joitakin tilanteita, joissa lähetykseen meni hetkellisesti väärää kuvaa. Kokonaisuutena tuotannon tekniset ratkaisut kuitenkin osoittautuivat toimiviksi, mikä vahvasti osaltaan käsitystä streaming-tuotantojen toimivuudesta pienen budjetin suorissa lähetyksissä.

Vaikka tuotannossa käytettiin selvästi eritasoisia kameroita, niiden välistä eroja kompensoi kuvauspaikan kirkas ja suhteellisen tasapainoinen valaistus. Myös se, että kamerat olivat samanmerkkisiä, helpotti kuvan säätöjen, kuten valkotasapainon asettamista, koska kameroiden asetusvalikoissa säädöistä käytetään identtisiä termejä. Suurimmat kuvanlaadulliset erot näkyivät DSR-500 -tuotantokameroiden kuvaa verrattaessa pienempien kameroiden kuvaan; Suuremmilla kennoilla varustetut kamerat tekivät etenkin värien osalta täyteläisempää kuvaa. Selvästi havaittavia eroja oli nähtävissä myös eri kameroiden syväterävyydessä. Pienempien kameroiden pienet kennot yhdistettynä pieneen polttoväliin tuottivat useimmissa tapauksissa lähes kauttaaltaan terävän kuvan, kun taas isommilla tuotantokameroilla oli helpompaa saada käyttökelpoista kuvaa terävän etualan ja sumean taustan ansiosta.

Tuotannossa käytetty robottikamera osoittautui todella käytännölliseksi, koska sen sai asetettua paikkaan, johon normaalisti kameraa ja sen operoijaa ei olisi mahdollista sijoittaa. Etenkin keskustelutilaisuuksissa yleisöä kuvattaessa robottikamera oli varsin huomaamaton, jolloin kuvauskohteena olevat ihmiset eivät osanneet jännittää tilannetta ja toimivat luonnollisesti. Robottikameroiden käyttökelpoisuus pienissä vähämiehittyissä tuotannoissa onkin huomioitu myös Anycast Stationin käyttöä ajatellen, koska laite sisältää tuen VISCA-liitännän kautta hallittavien kameroiden ohjaamiseen ja näiden asetusten säätämiseen. Vaikka kameroiden ohjaus olisikin voitu toteuttaa pelkästään Anycastilla, tuotannossa päädyttiin käyttämään erillistä kauko-ohjainta sen paremman ergonomian ja käytettävyyden vuoksi.

Vaikka kaikki materiaali saatiinkin tallennettua DV-laatusena koko tuotannon ajalta, siinä esiintyi häiritsevä ja pahimmillaan varsin merkittävä ja helposti havaittava ongelma. Osassa suorana lähetyksenä kuvatus- ta materiaalista kuva ja ääni eivät pysyneet synkronoituna, vaan kuva tuli huomattavasti ääntä jäljessä. Ongelmaa ei havaittu kesken tuotan- non, koska sitä oli hyvin vaikea todeta itse streaming-lähetyksestä sen hitaasta kuvataajuudesta johtuen, eikä ongelmaa myöskään havaittu materiaalia tiivistettäessä. Myöhempien kokeilujen perusteella voitiin harmittavasti todeta, että mikäli Anycastiin tuodaan ääntä analogisten liittimien kautta ja kuvaa digitaalisesti firewall-liitännän kautta, syntyy synkronointiongelmia. Aikaeroa voidaan kompensoida Anycastin kautta asettamalla viivettä äänilähteille, mutta tämä on ongelmallista, mikäli ääntä ja kuvaa tulee useasta eri lähteestä erilaisilla liitännöillä. Synkro- nointiongelmat eivät kuitenkaan ole ainoastaan Anycastin ongelma, vaan niitä esiintyy millä tahansa digitaalisella videomikserillä, koska kuvasignaalin pakkaaminen DV-muotoon kamerassa ja purkaminen mikserissä kestää laitteistosta riippuen joitakin satoja millisekunteja. Viive on selkeästi havaittavissa etenkin tilanteissa, joissa kuvataan esi- merkiksi puhujia, tai muita sellaisia tilanteita, joissa on helposti havait- tavia kuvaan sidonnaisia ääniä. Myös analogista signaalia vastaanotta- vat digitaalilaitteet kärsivät jonkinasteisesta viipeestä, mutta niissä ero ei ole yhtä huomattava, kuin DV-signaalia käsiteltäessä. Tuotannossa käytetyssä Datavideon SE-500 -videomikserissä havaittiin suurimmil- laan noin 160 millisekunnin ero kuvan ja äänen välillä, joka on kohtalai- sen helposti havaittavissa kuvamateriaalista riippuen. Tästä johtuen di- gitaaliset ratkaisut soveltuvat varsin huonosti tuotantoihin, joissa kuva välitetään myös paikan päällä oleville katsojille esimerkiksi valkokan- kaan ja videotykin välityksellä. Viipeestä johtuen valtaosassa ammatti- tason tuotannoista signaalia siirretäänkin tuotantopaikalla täysin analo- gisesti ja se muutetaan digitaaliseen muotoon vasta tallennusvaihees- sa.

## 5 YHTEENVETO

Streaming-tekniikoiden jatkuva kehitys ja niille suunnattu lisääntyvä tuki vaikutusvaltaisilta tahoilta ennakoivat streamingin suosiolle nopeaa kasvua. Vaikka streaming-tekniikat ovatkin kehittyneet toiminnoiltaan, niitä vaivaa etenkin ongelmat tietoturvaratkaisujen suhteen, jotka pahimmillaan estävät kehittyneimpien ominaisuuksien käytön kokonaan. Tästä huolimatta streaming-tekniikoita hyödyntävien multimediaspalveluiden määrä on kasvanut räjähdysmäisesti.

Neljän yleisimmän streaming-formaatin välillä ei ole havaittavissa merkittäviä laadullisia eroja. Sen sijaan teknisten ratkaisujen, etenkin tietoliikenneprotokollien käytön suhteen, formaatit sisältävät olennaisia eroja. Kaikille neljälle formaatille näyttäisi olevan tällä hetkellä omat käyttäjäkuntansa, ja vain aika näyttää, miten formaattien väliset voimasuhteet tulevat muuttumaan tekniikan ja palveluiden kehittyessä.

Case-osiossa raportoitu videotuotanto osoitti monia streaming-lähetyksen mahdollisuuksia. Käytännössä vastaavanlaista suorana lähetyksenä toteutettua videotuotantoa ei olisi ollut millään muulla tavalla mahdollista saada suuren yleisön katsottavaksi yhtä kustannustehokkaasti. Tuotanto osoitti myös joitakin uuden tekniikan varaan rakennetun palvelun heikkouksista; tuotannosta saadusta palautteesta voidaan päätellä, että asiakkaan ohjelmistot eivät ole kovinkaan helppokäyttöisiä ja ne vaativat kohtalaista tietämystä tietokoneen käytöstä. Mikäli streaming-lähetyksistä voitaisiin alkaa puhua todellisena uutena koko kansan mediana, niistä pitäisi saada niin helposti käytettävä, että yhdellä napin painalluksella päästäisiin tiukimpienkin palomuurien ohi. Tällöin voitaisiin puhua uudesta valtamediasta, joka uhkasi merkittävästi jo televisionkin suosiota, mikäli laadulliset ominaisuudet saataisiin vastaavalle tasolle.



## LÄHTEET

### Painetut lähteet:

Casad, J. & Willsey, B. 1999. TCP/IP Trainer. IT Press, Helsinki.

Kaario, K. 2002. TCP/IP -verkot. WS Bookwell, Porvoo

Leon-Garcia, A. & Widjaja I. 2000. Communication Networks, Fundamental Concepts and Key Architectures. McGraw-Hill Book Co, Singapore.

### Painamattomat lähteet:

Häkkinen, P. 2000. Videokuva ja videomateriaalin käyttö verkossa, Hypermedian perusteet. Digitaalisen Median Instituutti TTKK, Hypermedialaboratorio. Opetusmateriaali.

### Sähköiset lähteet:

A Streaming Media Primer [verkkodokumentti]. Adobe Systems Incorporated, 2001, [viitattu 15.2.2007]. Saatavissa:  
<http://www.adobe.com/products/aftereffects/pdfs/AdobeStr.pdf>

Adobe Flash Player Version Penetration [online]. Adobe Systems Incorporated, 2006 [viitattu 20.2.2007] Saatavissa:  
[http://www.adobe.com/products/player\\_census/flashplayer/version\\_penetration.html](http://www.adobe.com/products/player_census/flashplayer/version_penetration.html)

Anycast Station – Live Content Producer [online]. Sony Corporation, 2005 [viitattu 5.1.2007]. Saatavissa:  
<http://bssc.sel.sony.com/BroadcastandBusiness/minisites/anycast/anycastnew.shtml>

ASF Specification [online]. Microsoft Corporation, 2004. [viitattu 27.3.2007].  
Saatavissa:  
<http://www.microsoft.com/windows/windowsmedia/forpros/format/asfspec.aspx>

Broadcasting content live, Windows Media Encoder Help [ohjedokumentti].  
Microsoft, 2002a.

Choosing a Codec [online]. Streamingmedia.com, 2006 [viitattu 29.3.2007].  
Saatavissa:  
<http://www.streamingmedia.com/article.asp?id=9259>

Comparison of Media Players [online]. Wikipedia, 2007. [viitattu 30.3.2007].  
Saatavissa:  
[http://en.wikipedia.org/wiki/Comparison\\_of\\_media\\_players](http://en.wikipedia.org/wiki/Comparison_of_media_players)

Deploying Helix Proxy [verkkodokumentti]. RealNetworks Inc., 2005  
[viitattu 25.3.2007]. Saatavissa:  
[http://docs.real.com/docs/rn/datasheet/Helix\\_Proxy\\_S307.pdf](http://docs.real.com/docs/rn/datasheet/Helix_Proxy_S307.pdf)

FiCom tiedote 28.8.2006 [online]. FiCom, 2006 [viitattu 26.3.2007].  
Saatavissa:  
[http://www.ficom.fi/tietoa/tietoa\\_4\\_1.html?ld=1140700945.html](http://www.ficom.fi/tietoa/tietoa_4_1.html?ld=1140700945.html)

Flash Media Encoder [online]. Adobe Systems Incorporated, 2007  
[viitattu 27.3.2007]. Saatavissa:  
<http://www.adobe.com/products/flashmediaserver/flashmediaencoder/>

Helix Server Common Features [online]. RealNetworks Inc, 2007b  
[viitattu 27.3.2007]. Saatavissa:  
<http://www.realnetworks.com/products/mediaservers/features/common.html>

Helix Server Administration Guide[verkkodokumentti]. RealNetworks Inc.,  
2006 [viitattu 29.3.2007]. Saatavissa:  
<http://docs.real.com/docs/server11/wireline/HelixServerAdmin.pdf>

Helix DRM [online]. RealNetworks Inc., 2007a [viitattu 29.3.2007].  
Saatavissa:  
<http://www.realnetworks.com/products/drm/index.html>

HTTP Tunneling protocols [online]. Adobe Systems Incorporated, 2002, [viitattu 18.1.2007] Saatavissa:  
[http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn\\_16631](http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_16631)

Hypertext Transfer Protocol – HTTP/1.1 [verkkodokumentti]. Internet Engineering Task Force, 1999 [viitattu 10.12.2006]  
Saatavilla: <http://www.ietf.org/rfc/rfc2616.txt>

Manual DVCPPro 25/50 DV Codec [verkkodokumentti], Mainconcept AG., 2006 [viitattu 29.3.2007]. Saatavissa:  
[downloads.mainconcept.com/DVCPPro3.0.16\\_en.pdf](http://downloads.mainconcept.com/DVCPPro3.0.16_en.pdf)

NewTek Tricaster – Live Production System [online]. NewTek, Inc, 2007. [viitattu 23.3.2007]. Saatavissa:  
<http://www.newtek.com/tricaster/>

Ostajan opas - laajakaistayhteydet [online]. Viestintävirasto, 2006 [viitattu 19.3.2007]. Saatavissa:  
<http://www.ficora.fi/index/palvelut/viestintapalvelutkuluttajille/ostajanoppaat/ostajanopask.html>

Plug-ins, Windows Media Encoder Help [ohjedokumentti]. Microsoft, 2002b.

Programming Flash Communications Server [verkkodokumentti]. O'Reilly, 2005 [viitattu 25.1.2007]. Saatavissa:  
<http://www.oreilly.com/catalog/progflashcs/chapter/ch01.pdf>

Protocol Rollover [online]. Microsoft Corporation, 2007a, [viitattu 15.2.2007]. Saatavissa:  
<http://msdn2.microsoft.com/en-gb/library/aa390673.aspx>

QuickTime Streaming Server 5.0 Administration [verkkodokumentti]. Apple Computer Inc., 2003. [viitattu 15.12.2006]  
Saatavilla: [images.apple.com/server/pdfs/QT\\_Streaming\\_Server.pdf](http://images.apple.com/server/pdfs/QT_Streaming_Server.pdf)

QuickTime Streaming Server [online]. Apple Computer Inc., 2007a [viitattu 28.3.2007]. Saatavissa:  
<http://www.apple.com/quicktime/streamingserver/faq.html>

QuickTime Streaming Server – Tech Specs [online]. Apple Computer Inc., 2007b. [viitattu 28.3.2007]. Saatavissa:  
<http://www.apple.com/quicktime/streamingserver/specs.html>

QuickTime BroadCaster [online]. Apple Computer Inc., 2007c  
[viitattu 23.8.2007]. Saatavissa:  
<http://www.apple.com/quicktime/broadcaster/>

Streaming From a Web Server [online]. Microsoft Corporation, 2003.  
[viitattu 29.3.2007]. Saatavissa:  
<http://www.microsoft.com/windows/windowsmedia/howto/articles/webserver.aspx>

Streaming media [online]. Wikipedia, vapaa tietosanakirja, 2007a  
[viitattu 29.3.2007]. Saatavissa:  
<http://en.wikipedia.org/wiki/QuickTime>

Streaming Video on Mobile Phones [online]. Manifest Technology, 2006  
[viitattu 26.3.2007]. Saatavissa:  
[http://www.manifest-tech.com/ce\\_wireless/wireless\\_stream.htm](http://www.manifest-tech.com/ce_wireless/wireless_stream.htm)

TechTerms – Mebibyte [online]. TechTerms, 2007  
[viitattu 26.3.2007]. Saatavissa:  
<http://www.techterms.org/definition/mebibyte>

VideoLAN Player [online]. VideoLAN, 2007 [viitattu 29.3.2007]. Saatavissa:  
<http://www.videolan.org/vlc/>

Web Server vs. Streaming Server [online]. Microsoft Corporation, 2007b,  
[viitattu 27.3.2007]. Saatavissa:  
<http://www.microsoft.com/windows/windowsmedia/compare/webservvstreamserv.aspx>

Windows Media Metafile Reference [online ]. Microsoft Corporation, 2007c  
[viitattu 27.3.2007]. Saatavissa:  
<http://msdn2.microsoft.com/en-us/library/aa393399.aspx>

Windows Media Encoder FAQ [online]. Microsoft Corporation, 2005.  
[viitattu 29.3.2007]. Saatavissa:  
<http://www.microsoft.com/windows/windowsmedia/forpros/encoder/faq.aspx>

YLE 24 Online ympäristöuutiset [online]. Yleisradio, 2007  
[viitattu 8.1.2007]. Saatavissa:  
<http://www.yle.fi/uutiset/ymparisto/id50688.html>

YouTube-verkkosivusto [online]. YouTube Inc., 2007  
[viitattu 23.3.2007]. Saatavissa:  
<http://www.youtube.com>

### **Muut lähteet:**

Smith, D. Channel Director. RealNetworks Ltd. Sähköpostihaastattelu 7.-  
8.2.2007.

### **Kuvalähteet:**

Kuva 1: Unicast-lähetystekniikka. Antti Lehikoinen, 2007

Kuva 2: Multicast-lähetystekniikka. Antti Lehikoinen, 2007

Kuva 3: Tyypillinen protokollien käyttö streaming-istunnossa, Multimediajär-  
jestelmät – luentomoniste, Lappeenrannan teknillinen korkeakoulu, 2000.  
Muokkaus: Antti Lehikoinen

Kuva 4: Anycast Station AWS-500G. Saatavissa:  
<http://www.dimagemaker.com/ktml2/images/uploads/video/s1.jpg>

Kuva 5: RM-BR300 kauko-ohjain ja BRC-300 robottikamera. Saatavissa:  
Videobotics 2007, <http://www.videobotics.com/brc300.jpg>  
Blogmaster.fr 2007, <http://idata.blogmaster.fr/0/00/85/00/rm-br300.jpg>  
Muokkaus: Antti Lehikoinen

Kuva 6: Tuotannon laitekaavio, Antti Lehikoinen

Kuva 7: Kameroiden paikat tilaisuuksia taltioitaessa, Antti Lehikoinen

Taulukko 1: Käyttöjärjestelmän vaikutus katsojamäärään, Helix Universal Server Comparative Load Test Final Report, 2002, KeyLabs.

Muokkaus: Antti Lehikoinen

Taulukko 2: TCP-segmentin tietokentät, Casad, J. & Willsey, B. 1999.

TCP/IP Trainer. IT Press, Helsinki, s. 101. Muokkaus: Antti Lehikoinen

Taulukko 3: UDP-datagrammin tietokentät, Casad, J. & Willsey, B. 1999.

TCP/IP Trainer. IT Press, Helsinki. s. 106, Muokkaus: Antti Lehikoinen

Taulukko 4: QoS-liikennetyypit ja prioriteetit IEEE802.1D-standardin mukaisesti. Saatavissa: <http://en.wikipedia.org/wiki/Qos>, Muokkaus: Antti Lehikoinen

Taulukot 5, 6 ja 7: Toisto-ohjelmien tukemat videoformaatit, käyttöjärjestelmät ja tiedonsiirtoprotokollat. Comparison of media players, Wikipedia 2007.

Saatavissa: [http://en.wikipedia.org/wiki/Comparison\\_of\\_media\\_players](http://en.wikipedia.org/wiki/Comparison_of_media_players),

Muokkaus: Antti Lehikoinen

## LIITTEET

Liite-cd, joka sisältää:

- Opinnäytetyö pdf-muodossa
- Suomenkielinen tiivistelmä rtf-tiedostona
- Englanninkielinen abstrakti rtf-tiedostona
- Videomateriaalia case-tuotannosta kansiossa VIDEO