

HØGSKOLEN I ØSTFOLD

Avdeling for Informasjonsteknologi
Remmen
1757 Halden
Telefon: 69 21 50 00
URL: www.hiof.no

Høgskolen i Østfold

MIDTVEISRAPPORT

Prosjektkategori: Midtveisrapport for Hovedprosjektoppgave	<input checked="" type="checkbox"/> Fritt tilgjengelig
Omfang i studiepoeng: 20	<input type="checkbox"/> Fritt tilgjengelig etter:
Fagområde: Informasjonsteknologi	<input type="checkbox"/> Tilgjengelig etter avtale med samarbeidspartner

Rapporttittel: Utvikling av audiovisuell konsertformidling for Det Norske Blåseensemble - live og on-demand streaming av musikkopplevelser over Internett	Dato: 06.02.2008 Antall sider: Antall vedlegg:
Forfattere: Inge Thorin Eidsæther og Johan Sydseter	Veileder: Audun Vaaler
Avdeling / linje: Informatikk / Bachelor i Digital Medieproduksjon	Prosjektnummer: H08D21

Utført i samarbeid med: Det Norske Blåseensemble og Omni Produksjon	Kontaktperson hos samarb. partner: DNBE: Mattias Lundquist Omni: Trond-Atle Johansen
---	---

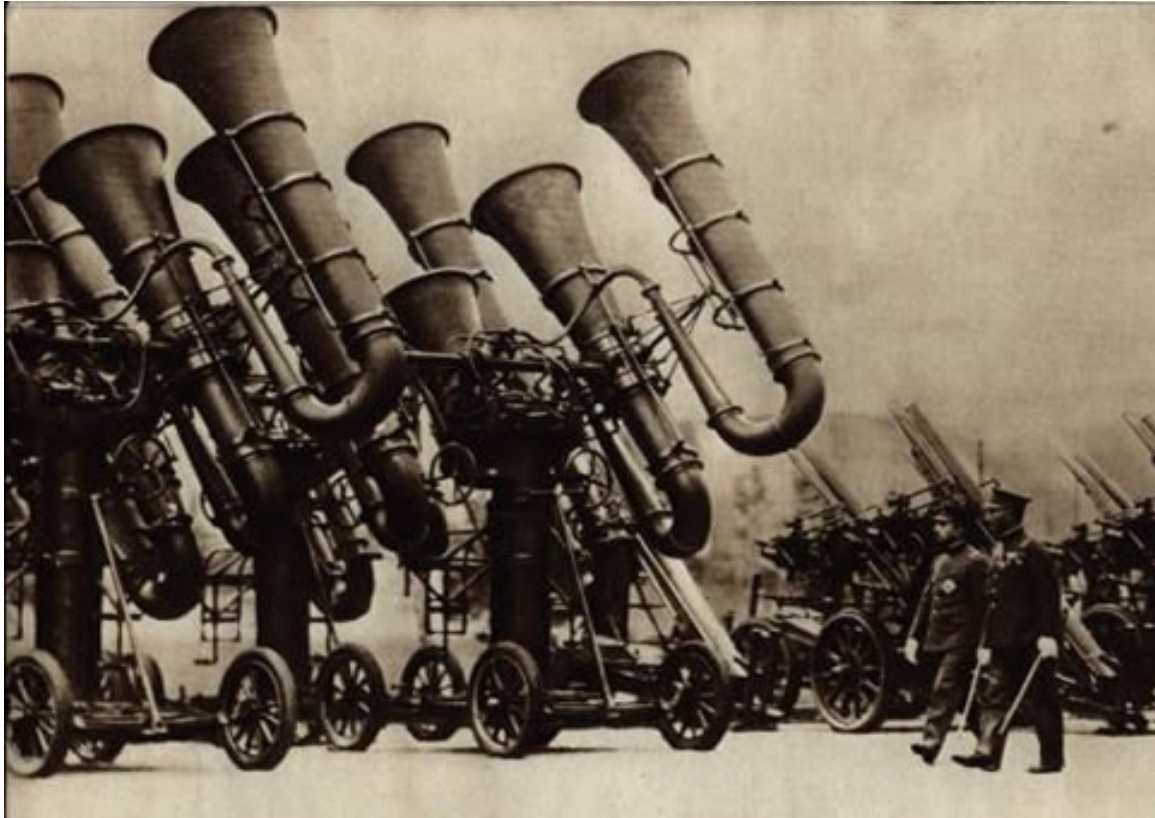
Ekstrakt: Det Norske Blåseensemble ønsker å formidle sin musikk til et større og bredere publikum gjennom nyskapende og visuelt appelerende flerkameraproduksjon av sine konserter sendt over Internet med moderne bredbåndsteknologi til ønskede fremvisningslokaler ved hjelp av høykvalitets live- og on-demand streaming av lyd og bilde, samt lagre konsertopptakene digitalt.

3 emneord:	<input type="text" value="Flerkameraproduksjon"/>
	<input type="text" value="Streaming"/>
	<input type="text" value="Konsertformidling"/>

Hovedprosjekt H08D21

Utvikling av audiovisuell konsertformidling for Det Norske Blåseensemble

- live og on-demand streaming av musikkopplevelser over Internett



Midtveisprosjektrapport

Gruppe H08D21

Inge Thorin Eidsæther

Johan Sydseter

Innholdsfortegnelse

- 1 Introduksjon**
 - 1.1 Prosjektbeskrivelse**
 - 1.2 Kravspesifikasjon**

- 2 Organisering**
 - 2.1 Oppdragsgiver**
 - 2.2 Medlemmer**
 - 2.3 Ansvarsfordeling**
 - 2.4 Møter**
 - 2.5 Nettside**

- 3 Prosjektmandat**
 - 3.1 Prosjektnavn**
 - 3.2 Prosjekteier**
 - 3.3 Bakgrunn**
 - 3.4 Formål**
 - 3.5 Mål**
 - 3.5.1 Effektmål**
 - 3.5.2 Prosessmål**
 - 3.5.3 Resultatmål**
 - 3.6 Omfang**
 - 3.7 Rammebetingelser**
 - 3.7.1 Forutsetninger**
 - 3.7.2 Begrensninger**
 - 3.8 Økonomi og lønnsomhet**
 - 3.8.1 Budsjett**
 - 3.9 Kritiske faktorer**
 - 3.9.1 Risikoanalyse**
 - 3.9.2 Interesseanalyse**
 - 3.10 Rapportering**

- 4 Faser**
 - 4.1 Startfase**
 - 4.2 Planleggingsfase**
 - 4.3 Forskningsfase**
 - 4.4 Utviklingssfase**
 - 4.5 Slutfase**
 - 4.6 Rapporteringsfase**

- 5 Oppgaver**
 - 5.1 Dokumentasjon**
 - 5.1.1 Rapporter**
 - 5.1.2 Referater**
 - 5.1.3 Notater**
 - 5.2 Milepæler**
 - 5.3 Aktivitetskart**

- 6 Datoer**
 - 6.1 Frister for innlevering**
 - 6.2 Eventer**

- 7 Systemløsning**
 - 7.1 Systemløsning**
 - 7.2 Diagram over systemløsning**

- 8 Serverløsninger**
 - 8.1 Innledning**
 - 8.2 Flaskehalsen og server problematikk**
 - 8.3 Media Streaming servere og brannmurer**
 - 8.4 Datakompresjon**
 - 8.5 Serverytelse**
 - 8.6 Operativsystemer, server programvare og grensesnitt**
 - 8.6.1 Flash Media Streaming Server**
 - 8.6.2 Darwin Streaming Server**
 - 8.6.3 QuickTime Streaming Server**
 - 8.6.4 Icecast Streaming Server**
 - 8.7 Erfaringer fra uttesting**
 - 8.8 Konklusjon**

- 9 Klientløsninger for streaming (kommer)**

- 10 Produksjonssystem for lyd og bilde/video (kommer)**

- 11 Oppsummering (Ikke fullstendig)**

- 12 Brukermanualer (Ikke fullstendig)**
 - 12.1 Wirecast**
 - 12.2 Capture Magic SD**
 - 12.3 VLC**
 - 12.4 Darwin Streaming Server**
 - 12.5 Icecast**
 - 12.6 Plugins for avspilling av media i nettlesere**

- 13 Referanser (Ikke fullstendig)**

- 14 Ordliste (kommer)**

- 15 Forkortelser (kommer)**

1 Introduksjon

1.1 Prosjektbeskrivelse

Det Norske Blåseensemble (DNBE) i Halden v/ Mattias Lundquist ønsker å nå frem til et større og bredere publikum med sin musikk og sine konserter, og har kontaktet Høgskolen i Østfold med et forslag til hovedprosjektoppgave for studenter ved avgangstrinnet. DNBE har fra før et pågående samarbeid med Omni Produksjon, som ved tidligere anledninger har gjennomført den tekniske delen av konsertformidling for DNBE.

Omni Produksjon v/ Trond-Atle Johansen er deltakende i prosjektet som rådgivere og tekniske konsulenter, men også som tjenesteleverandør ved større arrangementer som krever tekniske løsninger som går utover de tenkte rammene for hovedprosjektet.

I møte med representanter for oppdragsgiver og produksjonsselskap har gruppemedlemmene blitt enige om å komme med forslag til utforming av løsning (basert på den originale oppgaveteksten) *som vi etter våre kunnskaper og kompetanse ser oss i stand til å gjennomføre til oppdragsgivers fulle tilfredsstillelse*. Dette innebærer å foreta et *utvalg* av de nevnte mulighetene i oppgaveteksten og implementere disse.

Vårt forslag går primært ut på å skissere en portabel modell for flerkameraproduksjon og påfølgende live kringkasting av MPEG4 video og AAC audio over Internet ved hjelp av RTSP streaming fra konsertsted til fremvisningsted, og via transkoding til Theora video og Vorbis audio (eventuelt nedskalering av MPEG4 video og kvalitetsreduksjon av AAC audio) til webbrukere over HTTP, samt å lagre dette for on-demand nedlastning i form av podcasts, sekundært å utarbeide forslag til presentasjon av konsertene på en visuelt appellerende måte for DNBEs publikum.

1.2 Kravspesifikasjon

Vi valgte oppgaven levert av Det Norske Blåseensemblet i Halden som ønsket seg en løsning for sanntidsformidling av lyd og bilde fra sine konserter distribuert over Internett ved hjelp av flerkameraproduksjon og streaming-teknologi. Dessuten skulle data fra separate kilder kunne lagres for arkivering og senere gjøres tilgjengelig for on-demand streaming. Man så også for seg videreutvikling av formidlingsløsningen til et markedsføringsverktøy, hvor det blant annet var interessant med mobile applikasjoner og toveis-kommunikasjon med tilhørerne.

Ut ifra samtaler med oppdragsgiver, samt en vurdering av de forutsetninger vi hadde som gruppe med kun 2 medlemmer, har gruppen kommet frem til et kravspesifikasjon med et subset eller utvalg av de ovennevnte elementer som vist nedenfor:

1.2.1 Funksjonelle krav

- løsningen skal være portabel og kunne benyttes på lokasjoner der Internettlinjene er tilstrekkelige for mediastrømmer av akseptabel kvalitet
- løsningen skal gjøre bruk av minst 3 SD-DV videosignaler
- løsningen skal mikse videosignaler med stereo lyd og eventuelle preproduserte grafiske og/eller audiovisuelle elementer
- løsningen skal lagre individuelle videosignaler og mikse av audio for arkivering og senere redigering
- løsningen skal produsere 2 audio/video-strømmer i henholdsvis høy og redusert kvalitet
- løsningen skal ha en garantert oppetid minimum tilsvarende den tiden det tar å avvikle en konsert
- løsningen skal levere uavbrutt audio/video til RTSP og HTTP-klienter på Internett
- løsningen skal levere strømmer av valgt kvalitet til streaming-servere på LAN med marginalt pakketap
- løsningen skal ha et så enkelt, oversiktlig og sentralisert brukergrensesnitt som mulig

1.2.2 Tekniske krav

- løsningen skal i hovedsak implementeres på Apple Macintosh datamaskiner med operativsystemet OS X
- deler av løsningen skal, om nødvendig, kunne implementeres på en Mac eller PC med GNU/Linux som OS
- heldigital løsning ønskes dersom mulig, ingen analoge formater
- adekvat maskin- og programvare for miksing av lyd og video
- adekvat maskin- og programvare for sanntidslagring av audio og video til harddisk
- adekvat maskin- og programvare for distribusjon av live on on-demand mediastrømmer på Internett
- adekvat maskin- og programvare til samtidig lagring av 3 separate videokilder til harddisk

- adekvat maskin- og programvare til miksing av 3 separate videokilder og stereo audio
- adekvat maskin- og programvare til streaming av høykvalitets video og audio over RTSP og HTTP
- tilstrekkelig prosessorkraft og systemressurser til å gjennomføre en produksjon
- Internettilkobling med adekvat kapasitet og minimum uthastighet
- systemstabilitet over tid

2 Organisering

2.1 Oppdragsgiver

Oppdragsgiver for prosjektet er Det Norske Blåseensemble (DNBE) ved Mattias Lundquist. Veileder er Audun Vaaler ved Høgskolen i Østfold.

2.2 Medlemmer

- **Inge Thorin Eidsæther**
inge.t.eidsather@hiof.no, mob. tlf.: +47 40 45 56 71
- **Johan Sydseter**
johan.sydseter@hiof.no, mob. tlf.: +47 45 45 36 30

2.3 Ansvarsfordeling

Majoriteten av arbeidsoppgavene utføres av begge gruppemedlemmene (med unntak av rullering av referatskriving og bloggoppdatering), da vi er kun 2 personer på gruppa og derfor ikke har tid til utelukkende å konsentrere oss om faste arbeidsoppgaver gjennom prosjektets levetid. Arbeidsoppgavene så langt ser i grove trekk slik ut:

Inge: Forprosjektrapport, VLC transkoding, kravspec og systembeskrivelse

Johan: Utstyrsevaluering, midtveisrapport, test av Icecast og QTSS

Begge: Research, kunnskapstilegnelse, systemoppsett, testing og feilsøking

Vi justerer disse punktene kontinuerlig etter som vi finner ut hvem som er mest komfortabel med de enkelte arbeidsoppgavene, eller har best anledning til å utføre dem når de blir aktuelle. For å ha et referansepunkt, har vi et aktivitetskart som hjelper oss å holde rede for arbeidsoppgavene. (se under Oppgaver, 5.3)

2.4 Møter

Prosjektgruppa har ukentlige møter med veileder hver tirsdag kl. 10:15. Fra oppdragsgivers side er det ønskelig å delta med en representant på minimum hvert tredje prosjektmøte.

Gruppemedlemmene står ansvarlig for agenda, booking av møterom, innkalling på mail til deltakerne minst en dag før, og fører møtereferat som sendes på mail til de innkalte snarest etter møteavslutning, samt publiseres til prosjektnettsiden.

2.5 Nettside

Prosjektnettsiden er etablert på høgskolens prosjektserver, med adressen <http://frigg.hiof.no/h08d021>

På prosjektsiden ligger det prosjektbeskrivelse, gruppeintroduksjon, blog, møtereferater, notater, aktuelle linker, i tillegg til samtlige prosjektrapporter.

3 Prosjektmandat

3.1 Prosjektnavn

Utvikling av audiovisuell konsertformidling for Det Norske Blåseorkester -live og on-demand streaming av musikkopplevelser over Internett

3.2 Prosjekteier

Det Norske Blåseensemble

3.3 Bakgrunn

DNBEs ønsker om å nå ut til flest mulig ved hjelp av bredbåndsteknologi, og å komme bort fra det forutsigbare og forventede i en live konsertsammenheng.

3.4 Formål

- Presentere DNBEs musikk til et større og bredere publikum.
- Gjøre konsertopplevelsen mer visuelt appellerende og mindre forutsigbar for publikum.

3.5 Mål

3.5.1 Effektmål

- Skaffe Det Norske Blåseorkester et større og bredere publikum.

- Presentere DNBEs konserter på en mer visuelt appelerende og mindre forutsigbar måte.

3.5.2 Prosessmål

- Tilegne oss nødvendige kunnskaper om prosjektledelse og -samarbeid.
- Få mer erfaring med prosjektarbeid.
- Tilegne oss og forstå aktuelle begreper mht. behandling av lyd, bilde og video, samt live og on-demand streaming av data over Internett.
- Sette oss inn i rollene og funksjonene til en OB-buss (Outside Broadcasting).
- Opparbeide kompetanse på audio og videoproduksjon, encoding, streaming, kompresjon, regi, scenografi og konsertformidling.
- Realisere egenutvikling som følge av økt kompetanse på områder nevnt i punktet over.
- Oppnå en så god karakter som mulig i faget.

3.5.3 Resultatmål

Hovedresultatmål

- Utvikle en modell for flerkameraproduksjon og audiovisuell konsertformidling over Internett ved bruk av live og on-demand streaming.

Delmål

- Komme frem til en teknisk løsning for kringkasting av live konserter via streaming over Internett.
- Utarbeide et forslag til kreativ sceneløsning og presentasjon av lyd og bilde til dette formål.
- Gjennomføre kringkasting av en live konsert ved bruk av valgt løsning.
- Finne ut om (evt. når og hvor?) satsing på streaming er økonomisk forsvarlig og/eller teknisk gjennomførbar i allmen/lokal kringkastingssammenheng.

3.6 Omfang

Prosjektet er ikke en del av et større prosjekt, men kan betraktes som en parallell til det pågående samarbeidet mellom DNBE og Omni Produksjon, men i mindre skala. Prosjektet er allikevel omfattende, og innebærer flere punkter som er nærmere beskrevet i avsnitt 4: Utviklingsfaser.

3.7 Rammebetingelser

3.7.1 Forutsetninger

Utstyr

- Nødvendig utstyr for gjennomføring av prosjektet er tilgjengelig fra eller blir anskaffet og bekostet av Det Norske Blåseorkester. Se for øvrig budsjett.
- Oppdragsgiver har valgt Mac OS X som plattform

Tid

- Min. 5 timer/dag * 5 dager/uke * 17 uker = 425 timer per person

Ressurser

- Egne bærbare Mac G4 Powerbook maskiner
- Studentlisenser for programvare
- Internettlinje med kapasitet til overføring av 1 MPEG4 over RTSP

Uakseptable tiltak

- Privat bekostning av utgifter i forbindelse med prosjektets gjennomføring

3.7.2 Begrensninger

Begrensninger i prosjektet utgjøres blant annet av:

- Oppdragsgivers budsjett og investeringsvilje
- Tilgang og adminrettigheter til utstyr på utsiden av lokalnettet (ISP, etc.)
- Tilgjengelig tid for research, testing og implementering
- Tekniske muligheter i anvendt utstyr, maskin- og programvare

3.8 Økonomi og lønnsomhet

På bakgrunn av innledningsmessige problemer med begrensninger i anvendt maskin- og programvare som vi i etterkant har overkommet, fant vi det nødvendig for gjennomføring av prosjektet å anbefale bruk av dedikert ekstern videomikser.

Begrunnelsen var blant annet mengden data som skulle prosesseres, sannsynlig overbelastning av operativsystem og signalbuss på grunn av dette, enklere og mer oversiktlig operasjon, samt færre feilkilder.

Vi har brukt mye tid på å finne frem til alternativer for videomikser og monitorløsninger, da det er mange aktører på markedet, de fleste av dem med mange og forskjellige modeller med ofte svært omfattende spesifikasjoner.

Vi har gått bort i fra å anbefale denne løsningen, og i stedet føre opp dette som et alternativ, da audio/video miksing og streaming i programvare er blitt testet med vellykket resultat.

Produksjon av tekst overlay ('super') for lower thirds, etc, gjøres også i programvare, på samme måte som encoding og streaming av lyd og video.

Hva gjelder lønnsomhet for oppdragsgiver, er det å håpe at vårt løsningsforslag bidrar til økt eksponering og dermed økt popularitet blant publikum, som igjen forhåpentligvis resulterer i flere konsertoppdrag.

3.8.1 Budsjett

Antall	Beskrivelse	Pris	Annet
3	SD-DV kamera		DNBE/skolen har
3	Signalkabler til kamera		DNBE/skolen har
4	PR-radio	kr: 600	
4	Headset	kr:	
1	composite-firewire adapter		Alternativ løsning, HIØ har
3	Mac G4/G5 med OS X		1: Tekstoverlay og streaming av MPEG4 over RTSP 2: Reflektor av MPEG4 via QTSS 3: Transcoding og streaming av h.264 eller OGG/Theora over HTTP
1	WireCast lisens	kr: 2245	
1	Capture Magic SD	kr: 759	
sum:		kr: 3604	

3.9 Kritiske faktorer

3.9.1 Risikoanalyse

Planleggingsstadiet			
Årsak	Risiko	Konsekvens	Tiltak
Sykdom	Middels	Forsinkelser	Mer arbeid på øvrige involverte, økt tidspress
Splid i gruppen	Lav	Forsinkelser	Diskusjon og

For omfattende oppgave	Middels	Uferdig eller utilfredsstillende oppgaveløsning	Nedskalering av oppgaveteksten i forhold til hva som realistisk kan oppnås
Dårlig planlegging	Lav	Uferdig eller utilfredsstillende oppgaveløsning	Jevn jobbing og oppfølging av tidligere arbeid og resultater
Tidspress	Høy	For lite tid til testing, beslutninger tatt på for dårlig grunnlag	Følge faste rutiner for prosjektjobbing, ikke forfølge urealistiske mål
Kunnskapsmangel og inkompetanse	Lav	Uferdig eller utilfredsstillende oppgaveløsning	Gjøre research, sette seg inn i fagområder og terminologi, konsultere fagfolk
Produksjonsstadiet			
Feil på kamera	Lav	Færre bildeutsnitt å velge fra	Backupkamera
Snubling i kabler	Middels	Tap av bilde og/eller lyd, skade på utstyr som velter	Taping av kabler
Strømbrydd	Lav	Produksjonsstans	UPS
Feil på videomikser	Meget lav	Produksjonsstans	Ingen (evt. mulig backupløsning i software)
Feil på lydmikser eller mikrofoner	Meget lav	Produksjonsstans	Backupmikrofoner
Datamaskin henger	Lav	Ingen streaming, ingen lokal lagring	Restart av datamaskin
Programvare henger	Middels	Ingen streaming	Restart av programvare, eventuelt også datamaskin
LAN-problemer	Lav	Ingen streaming	

Internettilgang ustabil eller borte	Lav	Ingen streaming	
--	-----	-----------------	--

3.9.2 Interesseanalyse

TV Østfold: Formidling av streaming data til kabelTV-abonnenter
 Kommuner: Oppsett av mikrobølgelink i fbm. streaming via kabelTV
 Høgskolen: Potensiell videre drift av (deler av) systemet
 ISPer: Ved evt. co-location av streaming servere

3.10 Rapportering

Rapportering skjer i form av forprosjektrapport, midtveisrapport, sluttrapport (for levering) og en individuell rapport per milepæl (for arkivering).

4 Faser

4.1 Startfase

- Innhenting av mer detaljert informasjon om oppgaven
- Avklaring av oppdragsgivers ønsker og krav til produktets funksjonalitet.

4.2 Planleggingsfase

- Innhenting av informasjon, blant annet datoer og frister for innleveringer.
- Planlegging: hva MÅ og/eller BØR gjøres (og IKKE gjøres).
- Skrivning av forprosjektrapport som leveres til veileder og oppdragsgiver.

4.3 Forskningsfase

- Research av teknologi, konsepter og terminologi fra flere fagområder
- Konsultasjon med fagfolk
- Kunnskapstilegnelse og kompetansebygging
- Evaluering av aktuelt utstyr, maskin- og programvare
- Undersøkelser av tilgjengeligheten til aktuelt utstyr
- Innhenting av priser og anbud på valgt utstyr

4.4 Utviklingsfase

- Utarbeidelse av kravspesifikasjoner
- Design og utvikling av prototype(r) for valgt løsningsforslag.
- Anskaffelse av ønsket utstyr etter oppdragsgivers avgjørelse

4.5 Testfase

- Oppsett og tilkobling av lyd- og videoutstyr, samt maskinvare
- Installasjon og konfigurasjon av programvare
- Testing av design og funksjonalitet av utstyret i fht. kravspesifikasjoner.
- Testing av, og feilsøking på samtlige involverte systemkomponenter.

4.6 Implementeringsfase

- Ferdigstille prototyper.
- Sette opp, bemanne og overvåke operasjonen av systemet.

4.7 Slutfase

- Utarbeidelse av poster og presentasjon
- Gjennomføring av presentasjon
- Skrivning av sluttrapport.

5 Oppgaver

5.1 Dokumenter

- Forprosjektrapport
- Midtveisrapport
- Sluttrapport
- En rapport per milepæl (for arkivering i prosjekperm)
- Prosjektnotater
- Kravspesifikasjon

- Diagrammer
- Møtereferater

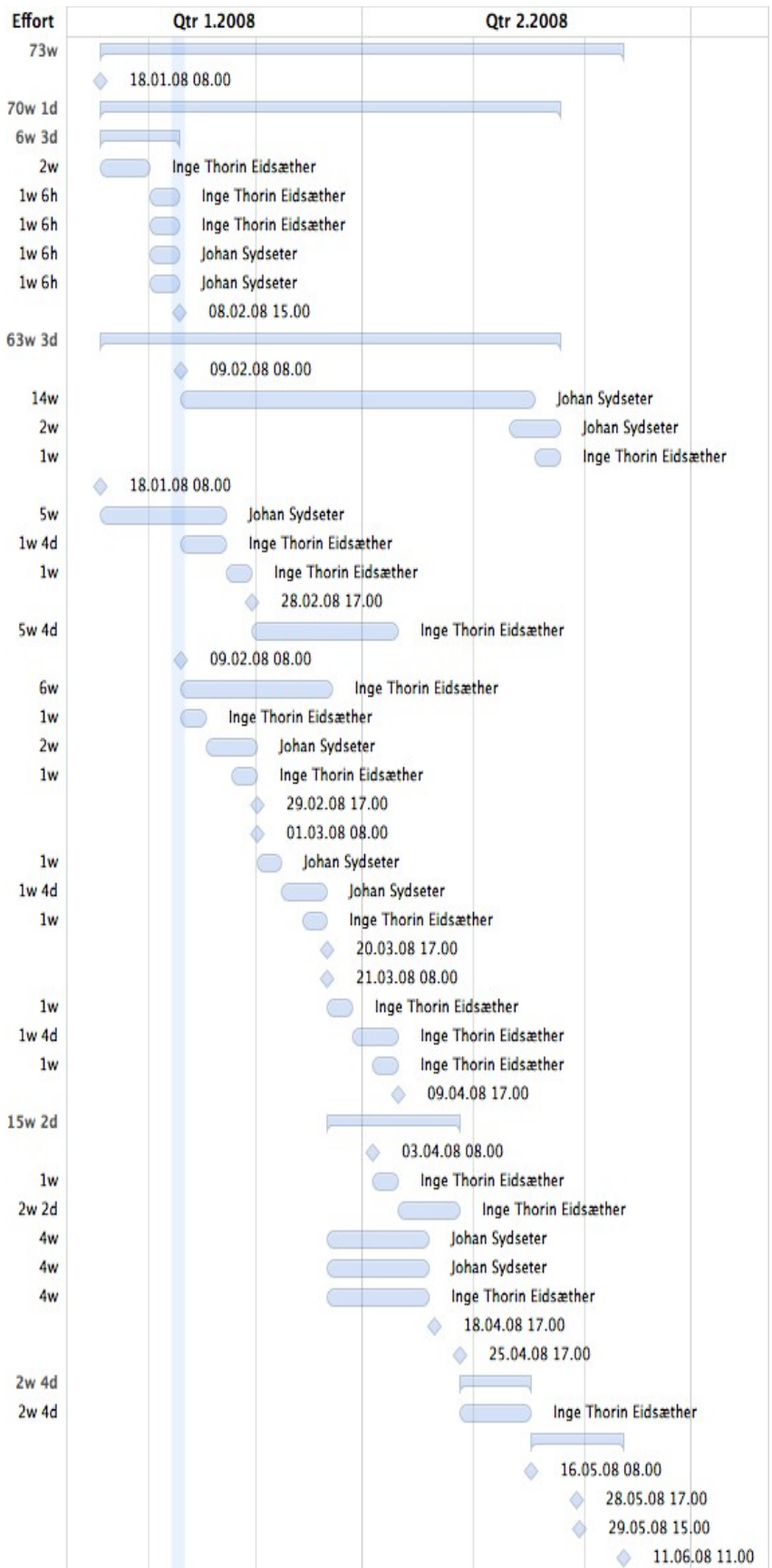
5.2 Milepælsplan

M01:	18.01	Prosjektstart
M02:	08.02:	Forprosjektrapport
M03:	09.02:	Start for dokumentasjonsfilm arbeid
M04:	28.02	Start for utarbeidelse av prosjektets kravspesifikasjoner
M05:	28.02	Slutføring av prosjektets kravspesifikasjoner
M06:	29.02	Klient/Serverløsning for streaming over RTSP.
M07:	20.03	Klient/Serverløsning for streaming over HTTP
M8:	09.04	Produksjonssystem med lyd og bilde/video
M9:	03.04	Start testing/feilsøking av det fullstendige systemet
M10:	18.04	Midtveisrapport
M11:	25.04	Ferdig testing/feilsøking av det fullstendige systemet
M12:	28.04-15.05	Konsert med Det Norske Blåseensemble
M13:	16.05	Pressemelding
M14:	29.05	Sluttrapport
M15:	11.06	Presentasjon

5.3 Aktivitetskart

Task

- 1) Utvikling av audiovisuell konsertformidling for Det...
- 1.1) M01 Prosjekt start
- 1.2) Forskningsfase
 - 1.2.1) Planleggingsfasen
 - 1.2.1.1) Prosjektmandat
 - 1.2.1.2) Milepølsplan
 - 1.2.1.3) Risiko analyse og interesse analyse
 - 1.2.1.4) Aktivitetskart
 - 1.2.1.5) Grovt Budsjett
 - 1.2.1.6) M02 Forprosjekt rapport
 - 1.2.2) Utviklingsfase
 - 1.2.2.1) M03 Start for dokumentasjonsfilm...
 - 1.2.2.2) filming
 - 1.2.2.3) motion graphics/composit...
 - 1.2.2.4) klipp
 - 1.2.2.5) M04 Start for utarbeidelse av prosjektet...
 - 1.2.2.6) Innsamling av forslag til innkjøp samt...
 - 1.2.2.7) korrigeringer i forhold til...
 - 1.2.2.8) budsjettering av det endelige budsjettet
 - 1.2.2.9) M05 Slutføring av prosjektets...
 - 1.2.2.10) Innkjøp
 - 1.2.2.11) M06 Start for utviklingen av en Prototyp...
 - 1.2.2.12) anskaffelse av utstyr til prototypene
 - 1.2.2.13) oppsett av infrastruktur i forminsket...
 - 1.2.2.14) utprøving, testing og tilpasning av RTSP...
 - 1.2.2.15) test av prototypen i et realistisk senario
 - 1.2.2.16) M07 Prototype av klient/serversystem...
 - 1.2.2.17) M08 Start for utviklingen av en Prototyp...
 - 1.2.2.18) oppsett av infrastruktur i forminsket...
 - 1.2.2.19) utprøving, testing og tilpasning av HTT...
 - 1.2.2.20) test av prototypen i et realistisk senario
 - 1.2.2.21) M09 Prototype av klient/serversystem...
 - 1.2.2.22) M10 Start for utviklingen av en Prototyp...
 - 1.2.2.23) oppsett av infrastruktur i forminsket...
 - 1.2.2.24) utprøving, testing og tilpasning av lyd...
 - 1.2.2.25) test av prototypen i et realistisk senario
 - 1.2.2.26) M11 Prototype av produksjonssystem f...
 - 1.2.2.27) Testfase
 - 1.2.2.27.1) M12 Start testing og feilsøking av...
 - 1.2.2.27.2) rigging av infrastruktur
 - 1.2.2.27.3) utprøving og feilsøking
 - 1.2.2.27.4) Teknisk beskrivelse av systemet
 - 1.2.2.27.5) Utforming av programvare...
 - 1.2.2.27.6) Utforming av brukerveiledningen
 - 1.2.2.27.7) M13 Midtveisrapport
 - 1.2.2.27.8) M14 Ferdig testing og feilsøking
- 1.3) Implementeringsfase
 - 1.3.1) M15 Konsert med det norske Blåse Ensemblet
- 1.4) Slutfase
 - 1.4.1) M16 Pressemelding
 - 1.4.2) M17 Ferdig dokumentasjonsfilm
 - 1.4.3) M18 Sluttrapport
 - 1.4.4) M19 Presentasjon av prosjektet



6 Datoer

6.1 Frister for innlevering

- mandag 7 februar: deadline for levering av forslag til oppgaveløsning
- fredag 8 februar: levering av forprosjektrapport
- fredag 18 april: levering av midtveisrapport
- fredag 16 mai: lage og levere pressemelding til Østfoldportalen
- torsdag 29 mai kl 14:00: levering av sluttrapporten

6.2 Eventer

- tirsdag 10.06: Presentasjon av prosjektet
- onsdag 11.06: Presentasjon av prosjektet
(foreløpig liste)

7 Systembeskrivelse

7.1 Eksisterende løsning

DNBE har frem til i dag gjort bruk av innleide tjenester fra Omni Produksjon for å spille inn lyd og bilde ved avvikling av sine konserter. I produksjonen har man benyttet opp til 6 profesjonelle kameraer og avansert lydutstyr tilkoblet en såkalt OB-buss (Outside Broadcasting) med maskinvare og utstyr til lagring av resultatet på tape og/eller levering av live feeds til bredbåndsleverandør (og evt. kabelTV-leverandør) for videre distribusjon.

7.2 Foreslått løsning

- 3 stk. SD-DV kamera kobles opp med 20m DistanceDV Firewire kabler til 3 stk. separate Firewire 400 hub/splitter.
- Videosignaler fra Firewire hub/splitter sendes over 2 separate kabler til 2 separate datamaskiner for henholdsvis lagring og live streaming av video.
- Kamera #1 og #2 kobles via hub/splitter til intern Firewire 400 buss, kamera #3 kobles via hub/splitter til PCI-E Firewire kort (på maskin for streaming).

- Kamera #1 og #2 kobles via hub/splitter til intern Firewire 400 buss, kamera #3 kobles via hub/splitter til CardBus Firewire adapter (på maskin for lagring).
- Ønsket antall mikrofoner tar opp audio og sender analoge signaler til miksepult over XLR-kabler for nedmiksing til stereo.
- Miksepult sender analog stereo audio miks til audio inn på alle kameraene, eventuelt inn på lydkort over XLR-til-Jack (3.5mm) -kabler på datamaskiner for BÅDE opptak og streaming. I det sist nevnte tilfelle må det påregnes nødvendig korrigerings av synkronisering mellom video og audio, mens i det først nevnte tilfelle introduseres mer kabling.
- Høykvalitets lydkort (PCI-E) og adaptere (Firewire/USB2) benyttes for audio innspilling dersom signal/Støyforhold ikke er tilfredsstillende.
- Capture Magic SD lagrer separate videosignaler til harddisk, Audacity tar opp stereo audio fra audio inn på lydkort dersom lyden ikke sendes til kamera, begge på en Apple MacBook Pro.
- Wirecast3 mikser audio/video, overlay og evt. preproduserte elementer (grafikk, lydklipp, videosnutter - evt fra Desktop Presenter installert på en annen maskin på samme lokalnett) på en Apple G5 PowerMac.
- Wirecast3 koder audio/video/grafikk miks til 2 strømmen av henholdsvis høy og redusert/nedskalert kvalitet (begge MPEG4/AAC) på samme maskin
- Wirecast3 lagrer en audio/video/grafikk miks i MPEG4/AAC format lokalt på harddisk på samme maskin.
- Wirecast3 sender MPEG4/AAC strøm i høy kvalitet over RTSP med QuickTime Announce til Darwin Streaming Server (DSS) eller QuickTime Streaming Server (QTSS) fra samme maskin.
- Wirecast3 sender MPEG4/AAC strøm i redusert kvalitet over Unicast ENTEN direkte til VLC på separat datamaskin (for transkoding til Theora video og Vorbis audio og påfølgende streaming til Icecast2 server) ELLER uendret til DSS/QTSS
- DSS/QTSS sender MPEG4/AAC høy kvalitetsstrøm over Multicast til videoklienter (QT, VLC, etc. - samt til evt. QTSS relay server(e)) på Internett over Unicast fra en Apple G5 PowerMac.
- VLC transkoder video/audio fra MPEG4/AAC til Theora/Vorbis (dersom det lar seg gjennomføre i praksis) som eneste (ressurskrevende!) applikasjon på separat Apple G5 PowerMac eller PC med GNU/Linux.

(ffmpeg2theora kan benyttes som alternativ til VLC for transkoding fra MPEG4/AAC til Theora/Vorbis, dersom utvikler får kompilert inn støtte for AAC i Mac OS X versjonen av applikasjonen)

- Icecast2 server sender Theora/Vorbis over HTTP til brukere på web med Java Runtime Environment (JRE) og nettleser med nødvendig støtte for JavaScript installert, evt. også til Icecast2 relay servere på Internett, alt fra samme maskin som kjører QTSS.
- Nettlesere med støtte for JavaScript påslått laster automatisk ned iTheora eller Cortado Java applet og spiller av Theora/Vorbis audio og video .
- DSS/QTSS sender MPEG4/AAC i redusert kvalitet til QuickTime Plugin embedded i webside, dersom transkoding med VLC mislykkes

(brukere på Windows-plattformen blir redigert til nedlastingsside for QuickTime plugin, dersom MPEG4/AAC strømmer forespørres)

- Nettside med informasjon om prosjektet og DNS-adresser til QTSS og Icecast servere lages og lastes opp til (evt. eksisterende) webserver.

7.2.1 Maskinvare på innspillingslokasjon:

Internettilkobling tilpasset ønsket antall brukere, ut ifra beregningen:

Video, høy kvalitet:	1000kbps
Audio, høy kvalitet:	128kbps
Til sammen:	1128kbps

Video, lav kvalitet:	300kbps
Audio, lav kvalitet:	64kbps
Til sammen:	364kbps

- 3 stk. DV-kamera eller HDV-kamera i DV-modus
- 3 stk. 20m DistanceDV kabler (med signalforsterkere) for bedre kameraposisjonering ved live konsertopptak
- 3 stk. Firewire 400 hub for split av videosignal til hhv. opptak og streaming
- 1 stk. 2-ports Firewire 400 Cardbus adapter for å muliggjøre samtidig opptak av 3 videokilder
- 1 stk. 2-ports Firewire 400 PCI-E kort for å muliggjøre miksing av 3 videokilder
- 1 stk. Apple MacBook Pro, for:
 - lagring av video fra separate kamera med Capture Magic SD til harddisk for arkivering og senere redigering

- lagring av lyd fra miksepult til harddisk med Audacity for arkivering og senere redigering
- 1 stk. Apple G5 PowerMac, for:
 - miksing av lyd, video og preproduserte elementer for live streaming med Wirecast3
 - lagring av video/lyd/grafikk miks til harddisk (for on-demand streaming) med Wirecast3
- 1 stk. Apple G5 PowerMac eller PC med Ubuntu GNU/Linux, for:
 - transkoding av MPEG4/AAC til Theora/Vorbis med VLC el. ffmpeg2theora
- X stk. mikrofoner for opptak av romlyd og individuelle instrumenter
- 1 stk. audio miksepult med minimum 12 kanaler for miksing av audio til stereo input for lagring og streaming

Eventuelt også:

- X stk. kommunikasjonsradio (PR-radio, 27MHz) med headset for beskjeder fra operatør av videomikser til kameraoperatører

Løsninger for lukket samband eksisterer, men er vesentlig dyrere

Som alternativ til WireCast som bildemikser kan videomiksing foretas med ekstern maskinvarebasert enhet, som f.eks: Datavideo SE-500. I et slikt tilfelle må video tas fra S-Video out eller Composite out fra kamera og føres via coax-kabler (for Composite) eller skjøtete S-Video kabler (koblet sammen med signalforsterkere) for å oppnå tilstrekkelig kabellengde, samt adekvat video capture kort/adapter installeres på datamaskiner for både opptak og streaming. Wirecast vil da kun brukes for å mikse inn overlay og preproduserte elementer.

7.2.2 Programvare på inspillingslokasjon

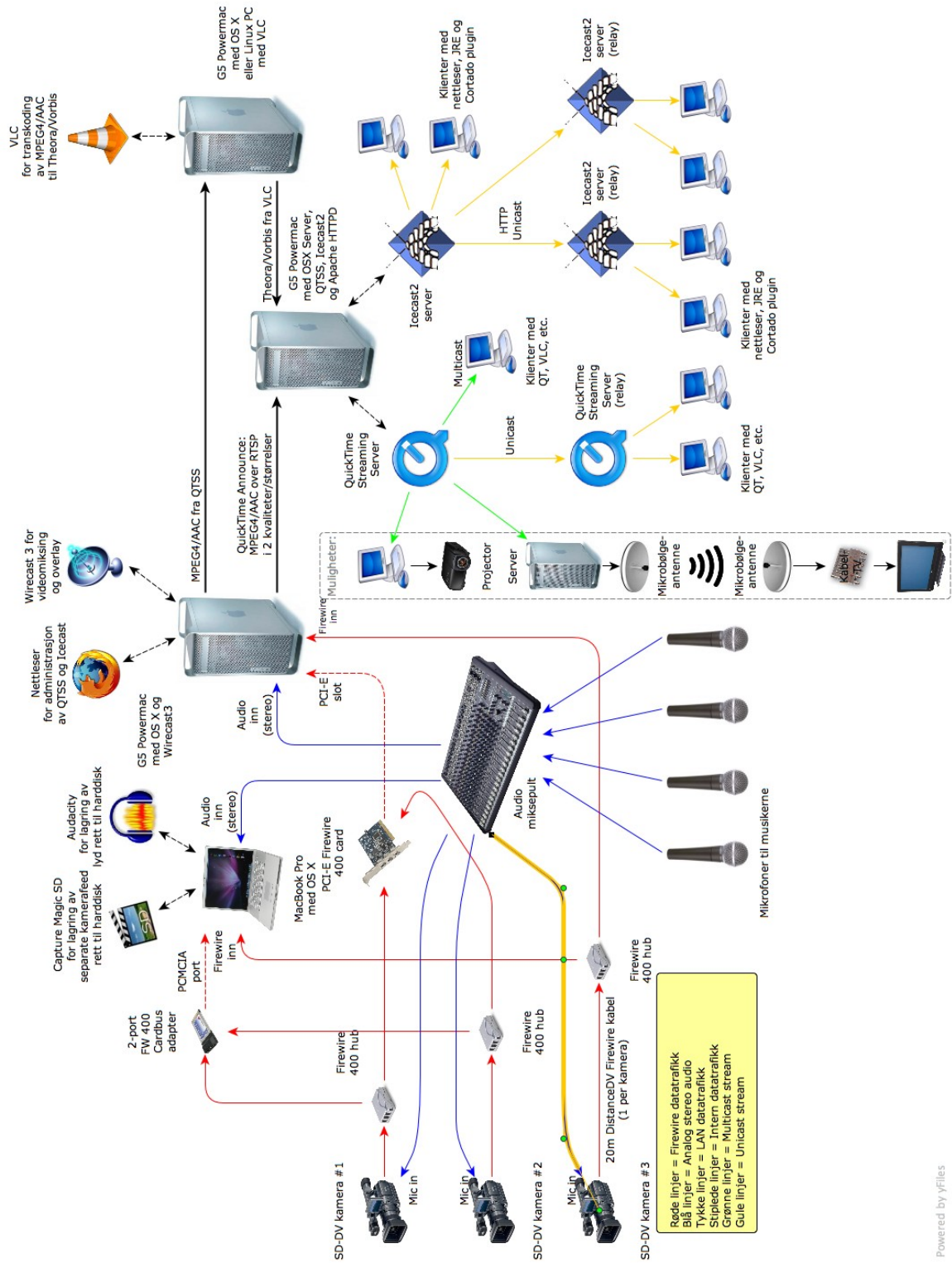
- Capture Magic SD
 - for lagring av video fra separate kamera til harddisk for arkivering og senere redigering
- Audacity
 - for lagring av lyd fra miksepult til harddisk for arkivering og senere redigering
- Wirecast3
 - miksing av lyd/video fra konsertene med evt. preprodusert materiale (grafikk, stillbilder, videoklipp, lydfiler, etc.)
 - QuickTime Announce (RTSP push) til QuickTime Streaming Server for live streaming
 - Unicast stream til VLC for transcoding til Theora/Vorbis

- Web browser/nettleser (Firefox, Opera, Camino, Safari, etc.), for:
 - administrasjon av QuickTime Streaming Server
 - administrasjon av Icecast2 server
- OS X Server (Leopard), for kjøring av QuickTime Streaming Server
- QuickTime Streaming Server
 - for distribusjon av fullkvalitets video/lyd i MPEG4/AAC format over multicast
 - distribusjon av redusert kvalitets video/lyd i MPEG4/AAC format over unicast
- VLC (alternativt: ffmpeg2theora)
 - for transkoding av MPEG4/AAC til Theora/Vorbis
- Icecast2 Server (under Fink på OS X)
 - for distribusjon av lavere kvalitets video/lyd i Theora/Vorbis -format over HTTP
- Apache HTTPD (følger med OS X)
 - for informasjon om prosjektet og DNS-adresser til QTSS og Icecast2 servere og relays

7.2.3 Maskin- og programvare på avspillingslokasjon

- Datamaskin med Windows, Mac OSX eller GNU/Linux
- JRE (Java Runtime Environment)
- Web browser/nettleser med støtte for Javascript påslått
- iTheora eller Cortado plugin (installerer automatisk)
 - for avspilling av Theora/Vorbis i nettleseren
- *QuickTime plugin for korrekt avspilling av MPEG4/AAC i MSIE

7.3 Diagram over Systemløsning



Powered by yFiles

8 Serverløsninger

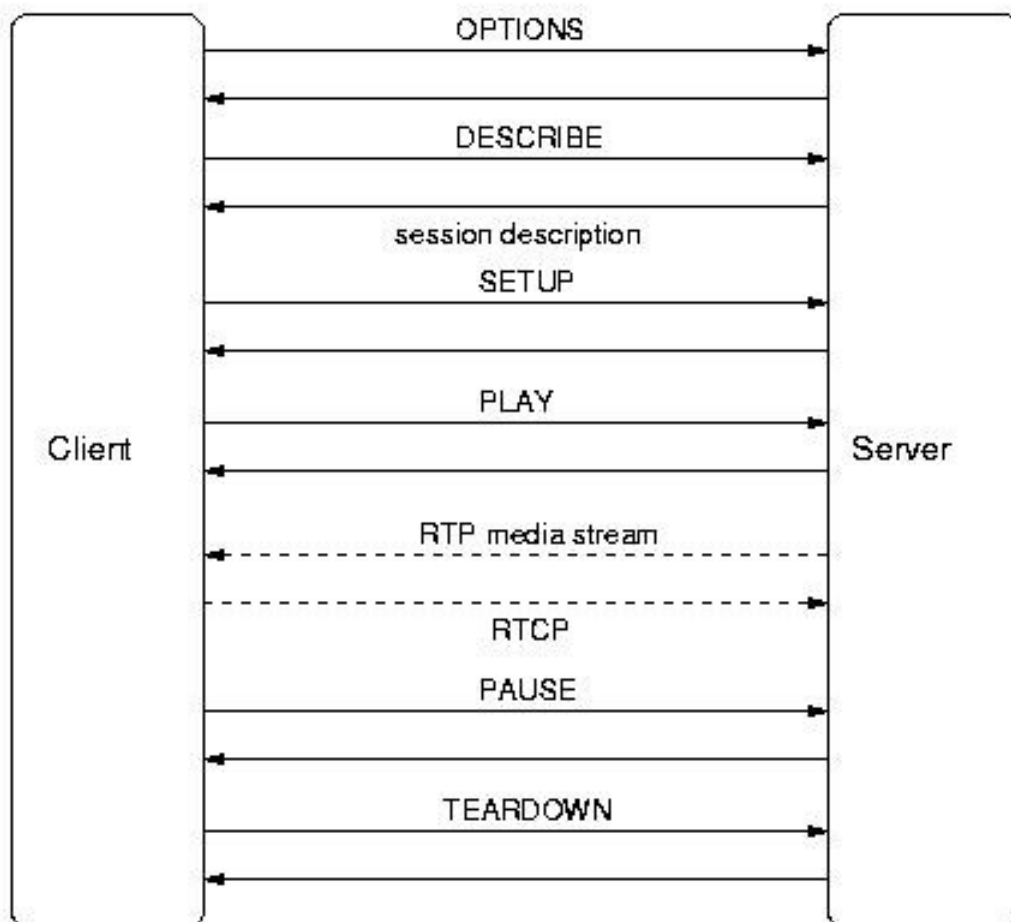
8.1 Innledning

Vi har sett på ulike server-løsninger for sanntids-streaming og har notert oss hvilke begrensninger og muligheter som finnes på de ulike plattformene. Streaming av H.264 over «Flash Media Streaming Server» (RTMP)*[Ref: 1], Streaming av H.264 over «Darwin Streaming Server» og «QuickTime Streaming Server» (RTSP)*[Ref: 2] og Streaming av Ogg/Theora ved hjelp av «Icecast» server (HTTP).*[Ref: 3] Vi har også sett på ulike protokoller for streaming og transkodning fra H.264 til «Ogg/Theora og Mpeg4 til Ogg/Theora. Innledningsvis blir noen utfordringer for streaming av media presentert. Så vil vi se på hvilke server-løsninger som finnes og hvilke erfaringer vi har gjort oss på bakgrunn av tester og research samt komme med en kort konklusjon.

8.2 Flaskehalsen og Server problematikk

Streaming servere tar bruk av IETF RTSP/RTP (Real Time Streaming Protocol/Real Time Protocol) eller RTMP (Real Time Messaging Protocol) for streaming. RTSP og RTMP er sanntid streaming-protokoller for signalisering mellom serveren og klienten. *[fig.1] RTSP og RTMP sendes innkapslet i TCP(Transmission Control Protocol) eller UDP. RTP tar så over etter at server og klient har blitt enige om å sende data. Den videre overføringen vil da enten gå over UDP (User Datagram Protocol) *[Ref: 4] eller TCP avhengig av hvor lang tid overføringen tar. Det er fortsatt ikke trivielt og kringkaste media over nett. Det finnes 4 ulike forhold man må ta hensyn til:

1. Brannmurer: De aller fleste hjemmenettverk bruker NAT eller PAT rutere.*[Ref: 5] Brannmurene i slike nettverk er konfigurert for å blokkere trafikk over UDP og RTP hvis det ikke er laget regler for hvilke porter denne type trafikk skal sendes over.
2. Datakompresjon: Siden antall bps i en DV fil er langt høyere enn hva som finnes tilgjengelig av båndbredde i nettet, må video komprimeres for å tilpasses den generelle båndbredden som finnes tilgjengelig hos brukeren.
3. System ytelse: Video og lyd legger større beslag på den tid brukeren er tilkoblet serveren og er mer resurskrevende enn tekst og bilder.
4. Operativsystem,server-programvare og Grensesnitt: Selv om dette ikke nødvendigvis er utslagsgivende, er det viktig å ta hensyn til. Ulike operativ systemer og grensesnitt krever ulik kompetanse. Tekstgrensesnitt er ofte mer krevende å forstå enn grafiske og har ofte en høyere begynnelse terskel. Mens flere operativsystemer ofte kan være plug-and-play, er andre mer modulære og krever at brukeren har inngående kjennskap til nettverk og systemadministrasjon.



*fig1: Signalisering over RTSP.

8.3 Media Streaming Servere og Brannmurer

Mange brannmurer er konfigurerte for å begrense trafikk over TCP og UDP. Det finnes 3 måter å sende media-strømmer på gjennom brannmurer. Ofte vil en kombinasjon av disse tre gi et vellykket resultat. *[Ref: 6

1. Streaming via port 80. Ved å sende strømmen over samme port som en vanlig web-server over RTSP, RTMP og RTP pakket inn i TCP, port 80 pakker, vil brannmuren slippe strømmen igjennom på lik linje som annen Internettrafikk siden mesteparten av Internettrafikken sendes over port 80, men å sende strømmer ved hjelp av TCP er arbeidskrevende. TCP har metoder for dataflyt- og datatrafikk kontroll og forskjellige integritetstester som sender data på nytt hvis datapakkene ikke når frem. Dette kan senke båndbredden i nettet og skape større forsinkelser i strømmen som igjen krever at man senker kvaliteten på det som sendes.

2. Åpne porter i brannmuren for hver klient som måtte ønske tilgang og på den måten oppnå høyere hastigheter og bedre ytelse. Problemet med denne løsningen er at brukerne selv må gå inn å forandre innstillingene i brannmuren. Ofte vil brukerens tekniske kompetanse være utilstrekkelig for å foreta slike justeringer. Man er nødt for å ta høyde for dette på klientsiden. Å håpe på tillatelse fra brukeren og brannmuren, vil være ren gjetting. Brukeren kan befinne seg bak 1 til 3 brannmurer uten å være klar over det. For streaming trenger man å åpne følgende porter:

TCP Port 80: hvis man ønsker å streame over RTSP/HTTP TCP
Port 554: over RTSP UDP Portene
6970 - 9999: for UDP streaming. TCP
Port 7070: brukes for RTSP av Real Server, QuickTime/Darwin og
WirecastTCP
Port 8000: brukes av Icecast.

3. Sette opp en proxy server i det lokale nettverkets DMZ på innsiden av den eksterne brannmuren og lage regler for kontakt mellom proxy-serveren og streaming serveren i brannmuren. Denne modellen har noe av den samme problematikken som punkt 2, men er ikke fullt så sårbar for angrep utenfra og brukes gjerne når avstandene mellom klient og server er for store som f.eks hvis man ønsker å sende strømmer Norge-Tyskland.*[Ref: 7]

Flash Media Streaming Server tar bruk av RTMP protokollen for streaming av media.*[Ref: 8] Ved å åpne port 1935 i brannmurene inn til web serveren som skal serve hjemmebrukerne eller ved å åpne port 80 ut fra maskinen som bruker Flash Media Streaming Server vil hjemmebrukere få tilgang til media strømmen via web serveren. Dette forutsetter at man ikke har en web server på samme maskin siden en web server bruker port 80 for å serve websider. QuickTime Streaming Server bruker port 554 som for RTSP streaming, men kan også sende data over port 80 for å traversere brannmurer.*[Ref: 9] Det samme gjelder for Darwin Streaming Server. Icecast tar bruk av 8000 over http. lavere porter enn 8000 anbefales ikke.*[Ref: 10] *Det er viktig å være klar over at selv om man sender RTSP over port 80 kan fortsatt video strømmene blokkeres av ISP'er og organisasjoner som filtrer ut applikasjoner for video og «per-to-per-trafikk», selv ved bruk over port 80.

8.4 Datakompresjon

Ulike streaming servere er tilpasset ulike containere for lyd og bilde og setter krav til hva slags kodere og protokoller som kan brukes til streaming. Vi gjorde ulike tester av forskjellige protokoller for streaming. Wirecast 3 fra Vara software*[Ref: 11] ble benyttet til dette. Testene ble gjort på en Mac Pro 2 x 2 Ghz Intel Xeon med 4 GB 667 Mhz DDR2 FB-DIMM. Vi så på streaming over RTP og UDP og sammenlignet Mpeg4, H.264 og Ogg/Theora. Under testingen, støtet vi på begrensninger i «throughput». Protokoller for streaming har en øvre grense for størrelsen på datapakkene som blir pakket. En metode ofte brukt i denne sammenheng og som også blir benyttet i Wirecast, er å sende RTP pakker over

UDP. Det finnes ingen absolutte grenser for hvor store RTP pakkene kan være, men det finnes noen retningsregler. *[Ref: 12] I prinsippet bør de holdes innenfor "ethernet frame payload" på 1500, for å unngå fragmentering og minske sjansene for pakketap. *[Ref: 13] *[Ref: 14] Wirecast, har en øvre grense for data som kan bli pakket på 1500 bit per RTP pakke. Vi fant at 1500kbps var max throughput gjennom 1 node uten pakketap. Ved høyere data rater, ble video signalet ustabile. Etter 4Mbps falt bilde ut ved flere anledninger. Testen ble gjort ved å peke kamera mot en slett overflate under samme type belysning. CPU lå på mellom 15-25 % under testen. Ved å kjøre ulike tester, fant vi ut at en data rate på 1500 Kbps var max av hva man kan forvente å kunne sende forsvarlig, uten pakketap, på en Mac Pro med de spesifikasjonene nevnt ovenfor. Dette utelukker bruk av kompresjon med større data rater som DV, Mpeg2 og andre containere som har fastsatte høye data rater. Derfor måtte vi sette et øvre tak på 1500 Kbps under testfasen. H.264 kom ikke overraskende best ut hva gjelder kvalitet, men krevde mye større prosessor kraft enn de to andre. Dette var allikevel overkommelig på test maskinen, tester vi hadde gjort på Powerbook G4, hadde tidligere ikke vært tilfredsstilene. H.264 introdusere også mye større forsinkelser på grunn av tyngre algoritmer for koding av video. Et annet problem var hvordan H.264 blir transportert over RTP. H.264 er veldig løst implementert hva gjelder tidskoder. Man kan velge på sende video strømmer mellom 5-30 fps. RTP har støtte for PTS (presentation timestamp) og ikke støtte for DTS (decoding timestamp) som kreves for transkoding og skrivning til fil. *[Ref: 15] Dette betyr at DTS må rekonstrueres. I Mpeg4 finnes DTS implementert noe H.264 ikke har støtte for. Det betyr at h.264 ikke egner seg til sanntids-streaming når man er avhengig av å transkode eller lagre strømmer. Dette utelukker i så fall Icecast som kun har støtte for Ogg/Theora og krever en klient som kan transkode H.264. QuickTime Streaming Server, Darwin Streaming Server og Flash Media Streaming Server derimot har alle støtte for h.264 og Mpeg4, men ikke Ogg/Theora.

Som tidligere nevnt, har Icecast kun støtte for Ogg/Theora. Theora video er et «open source» format for video kompresjon. *[Ref: 16] Fordelen med formatet er fri tilgjengelighet av kode, og at det er fritt frem å videreutvikle formatet for utvikling av streaming-løsninger, men implementering mangler støtte for Vorbis, i Mac OS X som kreves for støtte av lyd. Det finnes et QuickTime komponent som kan lastes ned og integreres i OS X, *[Ref: 17] men det ligger foreløpig på utvikler stadiet og mangler god nok integrasjon av Vorbis for kompresjon av lyd. VLC kan allikevel ta i mot en video strøm og transkode strømmen til Ogg/Theora samt sende den videre, men Ogg/Theora er ikke godt nok integrert her heller. *[Ref: 18] Det finnes en løsning som heter FFMpeg2theora, men den er foreløpig på utvikler stadiet. *[Ref: 19]

8.5 Serverytelse

Tester gjort på Icecast servere, viser at man har rom for mer enn 56000 brukere samtidig, men båndbredden og prosessorkraft legger en demper på hvor mange lyttere/seere man kan ha samtidig. *[Ref: 20] streaming servere er tilpasset å kunne betjene et stort antall brukere. web servere er ikke tilpasset media-streaming, men i følge en studie gjort av Dr. Louis Slothouber, er dette sjeldent

et problem siden max båndbredde vil nåes lenge før antall samtidige brukere går over grensen for hva en web server kan håndtere. På Windows og Mac Os X finnes det et tak for hvor mange TCP forbindelser som kan gjøres i sekundet implementert i hardware og software. På Unix og Windows NT er dette antallet ubegrenset. Problemet med en ubegrenset løsning er at antall TCP forbindelser og jobber vil hope seg opp samtidig som færre og færre jobber blir utført. Muligens er dette også tilfelle på mange Linux løsninger. TCP forbindelsen vil da kunne være sårbar for kø dannelser og i verste fall gå ned. Skulle man i slike tilfeller ha web server og streaming server kjørende på samme maskin, vil web serveren og streaming serveren konkurrere om forbindelsene. Dette er ofte ikke ønskelig hvis man har mange brukere siden overføringen av Lyd og video strømmer kan stoppe. Et hoderom, hvor man har ledig 30 % av resursene, er mer forsvarlig for å opprettholde en frisk server løsning. Man må også se på fremtidig vekst og ta hensyn til hva som kan bli aktuelt for å hindre en situasjon hvor man allerede fra dag 1 opplever at det ikke finnes ressurser nok. En løsning som kun tar hensyn til den etterspørselen som finnes i dag vil aldri holde mål. Veksten og kravene til multimedia økes proporsjonalt med folks krav til kvalitet og båndbredde. Hva som holder mål i dag vil ikke holde mål i morgen. En skalerbar løsning vil være fornuftig. Det er ønskelig å fordele web-trafikk over flere web servere med en dedikert streaming server som sørger for å betjene brukere i et «round-robin-nettverk». Det kan både være resursbesparende og billigere og sette bort media filer til en egen dedikert server for streaming av multimedia innhold. Det vil hindre mange av de problemene som finnes med skalering og oppgradering av servere og kunne åpne opp for en besparelse i både resurser og penger. *[Ref: 21]

Siden det er maskin kraft og båndbredde som utgjør flaskehalsen i en multimedia løsning er det viktig å velge en maskin som har muligheter til å servere nok brukere. Samtidig er det ønskelig å ha nok båndbredde. Nye standarder for nettverkstrafikk gjør det mulig å sette opp Gb forbindelser. 100BASE, 10GBASE *[Ref: 22] er de største nåværende standardene, men IEEE er også kommet i gang for å utrede en mulighet for å implementere en 100 Giga bit linje kalt 100GbE. *[Ref: 23] Man må påberegne 2000 brukere per Gb for en 364Kb/s videostrøm.

8.6 Operativsystemer, Server programvare og Grensesnitt

Valg av operativsystem og grensesnitt vil være utslagsgivende for den kompetanse som kreves og hvilke servere man ønsker å satse på. Det finnes utallige servere som er plattform uavhengig og fungerer på flere enn en plattform, men det er viktig å være klar over de begrensninger som finnes. Ofte kan det være et operativsystem som ikke støttes eller implementeres dårlig med tekst grensesnitt og modulære installasjoner. Installasjon og administrasjon fra et tekstgrensesnitt kan være vanskelige å gjennomføre mindre man har erfaring eller kompetanse på slike systemer. På den andre siden kan det være problematisk for en mindre bedrift og betale lisenser for dyre «plug-and-play» løsninger som allikevel krever en hvis kompetanse og innsikt i nettverkdrift og hardware. Jeg skal fort gå gjennom 3 løsninger vi har sett på. Det finnes mange flere, men på grunn av begrenset tid og ressurser, er det en umulighet å gi en

full oversikt. Det er heller ikke ønskelig ut i fra oppgavens kontekst og målsetninger.

8.6.1 Flash Media Streaming Server

Flash Media Streaming Server eies av Adobe og er en proprietær løsning for streaming over nett og har fra flash player 9 støtte for H.264 og Mpeg4. Den tar bruk av Real Time Messaging Protocol (RTMP) utviklet av Adobe Systems for Streaming av audio, video og data over Internet mellom en flash player og en server. Den ligger på toppen av Transmisjon Control Protocol (TCP). Flash Media Streaming Server finnes for Microsoft Windows Server 2003 og Linux Red Hat til US \$995. Den har støtte for sanntid- og on-demand streaming i H.264 og mpeg4-video og har innebygde sikkerhetssystemer for beskyttelse av åndsverk og distribusjon av multimedia på mobil. Serverens sterkeste side er bruk av RTMP protokollen som kan brukes på 3 forskjellige måter.

1. «Enkel» RTMP fungerer som egen protokoll på applikasjonslaget på port 1935 med TCP som transportprotokoll.
2. RTMPT som er en innkapslet utgave hvor RTMP pakkene blir pakket inn i en HTTP strøm før den blir sendt videre til TCP. Dette gjør det mulig å serve multimedia til hjemmebrukere uten at brannmur stopper strømmene.
3. RTMPS innkapslet på samme måte som RTMPT, men i kryptert form over HTTPS. Noe som vanskeliggjør muligheten for «man in the middle attacks».*[Ref: 24]

Flash Media Streaming Server er en enkel måte å kringkaste media på og kan håndteres av profesjonelle så vel som amatører. RTMP er tregere enn UDP baserte protokoller og vil bruke lengre tid på streaming enn UDP baserte protokoller med lengre ventetider etc, men Flash Media Streaming Server er stødig og pålitelige. Hvis vi kan stole på tallene til Adobe, har flash også størst utbredelse av alle systemer som finnes for visning av multimedia i nettlesere.*[Ref: 25] Det er også vært og nevne at dette ikke er den eneste løsningen for streaming av flash-video og -lyd. Wowza media server *[Ref: 26] og RED5 *[Ref: 27] er også vært og ta en titt på. Sist nevnte er en «open source» gratis server-løsning som med et stort utviklingspotensiale.

8.6.2 Darwin Streaming Server

Darwin Streaming Server var den første streaming serveren, med åpen kildekode, som har tatt utgangspunktet i RTSP protokollen. Den er basert på QuickTime Streaming Server og har derfor støtte for QuickTime streaming i ulike formater som Mpeg4 part 2, H.264/Mpeg4 AVC Mp3 og 3GP. Den har et enkelt webgrensesnitt som gjør den lett å administrere og installere. På tester kommer den godt ut hva gjelder stabilitet og ytelse i forhold til andre servere på markedet. Darwin Streaming Server har ikke teknisk support, men finnes for Linux, Windows og Mac *[Ref: 28] *[Ref: 29] *[Ref: 30] *fig2

Streaming Media Server Features

	Apple Computer Darwin Streaming Server 5.5.1	Adobe Systems Macromedia Flash Media Server 2	Microsoft Windows Media Server 9	RealNetworks Helix Mobile Server 11.0.0.1596
Server platform				
Windows 2000	Y	Y	N	Y
Windows 2003	Y	Y	Y	Y
Mac	Y	N	N	N
Linux	Y	Y	N	Y
Solaris	Y	N	N	Y
Client platform				
Windows	Y	Y	Y	Y
Mac	Y	Y	Y	Y
Linux	N	Flash 7	N	Y
Solaris	N	Flash 7	N	Y
Other	N	Flash 6 on PocketPC, HP-UX; Flash 4 on OS/2, Irix	N	RealPlayer 8 on Irix, HP-UX; RealPlayer 5 on OS/2
Encoder				
Testing codec	H.264	On2 VP6	Windows Media Video 9	RV 10
Other video codecs	H.263, H.261, MPEG-4, Sorenson 3 Video, DV, Cinepak, Motion JPEG	Sorenson Spark	Windows Media Video 7, Windows Media Video 8, Windows Media Video 9 Image 2	RealVideo 9, RealVideo 8
Audio codecs	AAC, Apple Lossless, A-Law, AMR Narrowband, IMA, MACE, Qdesign Music 2, Qualcomm PureVoice, u-Law, PCM, 5.1 channels	MPEG Layer III (MP3)	Windows Media Audio 9.1, Windows Media Audio 9 Voice, Windows Media Audio 9.1 Professional, PCM	RealAudio, Stereo RealAudio, StereoSurround RealAudio 10, Multichannel RealAudio 10 5.1
Access control				
NT authentication	N	Y, via Flash programming	Y	Y
User name/password	Y	Y, via Flash programming	N	Y
IP	N	Y, via Flash programming	Y	Y
Price				
Encoding/authoring	\$30 (QT Pro)	\$699	Free	\$199
Server, 100 streams	Free	\$4,500	Free with OS	\$5,000
Server, 400+ streams	Free	\$11,340 (450 streams)	Free with OS	\$13,000 (unlimited)
Player	Free	Free	Free	Free

Y = Yes, N = No

*fig2 (referanse nummer: 31) Sammenligning av ulike serverløsninger.

8.6.3 QuickTime Streaming

Har mye til felles med Darwin, men kommer som en del av Mac OS X Server som er en samling av ulike server løsninger for Mac som podcast, wiki, ical, web, mail etc. Prisen er 499 \$ og finnes bare på Mac. Ved siden av å ha et enkelt grafisk grensesnitt har den også et web grensesnitt identisk med Darwin Streaming Server. Den støtter også de samme formatene og har support hos Apple. QuickTime Streaming Server kan være en god løsning hvis man ønsker å kun bruke Apple maskiner. *[Ref: 32]

8.6.4 Icecast Streaming Server

Icecast er en fri tilgjengelig Streaming Server bygd på prinsipper for åpen kildekode. *[Ref: 33] Den støtter Ogg/Theora, Vorbis, Mp3, ACC og NSV og finnes for Unix, Linux og Windows. *[Ref: 34] *[Ref: 35] Den tar i bruk HTTP og kan derfor fanges opp på innsiden av et strengt kontrollert nettverk. Konfigureres i er tekst basert grensesnitt og må aksesseres gjennom DOS eller Secure Shell for skrivetilgang. Det kan være utfordrende å installere og administrere for brukere med minimal dataerfaring. Ogg/Theora er heller ikke veldig utbredt, men mangelen kan bøtes på ved å implementere moduler som ltheora *[Ref: 36] og Cortado *[Ref: 37] for avspilling av Ogg/Theora. Den største fordelene med Ogg/Theora er at formatet er fritt tilgjengelig med åpen kildekode og kan gi tilsvarende lik Mpeg4 kvalitet. Det finnes flere source-klienter for Icecast som kan ta et direkte video/audio signal fra et kamera og sende det videre til serveren, men flerkameraproduksjoner kan være vanskelig å få til med mindre man har tilgang på en dyktig programmerer.

8.7 Erfaringer fra uttesting

Vi begynte med uttesting av Wirecast og QuickTime Broadcaster den 13. Februar. Bakgrunnen for å velge Wirecast var at Det Norske Blåseensemblet ønsket å se på en software løsning for streaming og video switching for å slippe og forholde seg dyre hardware løsninger. (Se møte referat: 14.02.08) Vi fant ut under uttesting at Wirecast innebygde streaming server, hadde et betydelig pakketap, men Wirecast fungere også som source-klient for Darwin og QuickTime Streaming Server. DNBE hadde gitt uttrykk for at de ønsket en løsning for Machintosh. Så vi valgte derfor på grunnlag av dette, å bruke Darwin Streaming Server for streaming av strømmen over multicast. Multicast ble valgt for å ta bruk av Uninetts Ipv6 egenskaper. De fleste forskningsinstitusjoner har tilgang der og kan ta imot Multicast-strømmer. Dette krever liten båndbredde, kan være resursbesparende og øke muligheten for bedre kvalitet på strømmene. *[Ref: 38] Vi trengte allikevel en løsning for å traversere brannmurer, og tidligere erfaringer viste at enkelte rutere kan sperre for RTSP trafikk på tross av at trafikken blir sendt over port 80. Vi så derfor på en rekke flash løsninger, men fant at Adobe Flash Media Streaming Server var for dyr og at RED 5 fortsatt ikke hadde kommet langt nok i implementeringen av en sanntidvideoløsning for mac. RED 5 har de siste ukene utviklet seg kraftig og kan derfor fremover få en stor betydning for streaming av multimedia. *[Ref: 39] Vi har dessverre ikke tatt høyde for denne utviklingen og etterlater derfor stafettpinnen til andre som måtte ønske å gå Flash streaming nærmere i sømmene. Et av de større problemene vi stod ovenfor tidlig i prosessen, var bruk av H.264 komprimering over RTSP. Dette problemet kunne muligens vært løst om vi hadde gjort dette i Flash siden RTMP nå støtter streaming av H.264 strømmen. Flash vil kunne ta høyde for de problemene som finnes med brannmurer og manglende støtte for H.264 i nettlesere. *[Ref: 40] Det vil være tilfeller hvor RTSP trafikk ikke vil komme gjennom port 80 så vi så for oss en HTTP løsning, som det sikreste alternativet. Icecast var en slik løsning. Vi bestemte oss derfor for å gå for Icecast slik at vi var sikre på å kunne

traversere brannmurer. Icecast har kun støtte for Theora og NSV-video samt Vorbis, Mpeg3 og AAC på audio fronten. Vi var nødt for å se på en mulighet for Theora støtte på Machintosh plattformen og prøvde derfor ut ulike løsninger for transkodning av H.264 og Mpeg4 til Theodor. VLC og FFMpeg2Theora kan begge ta i mot Mpeg4 og H.264 og har støtte for å sende en strøm videre til Icecast, men det skulle vise seg å være problematisk og transkode RTSP-strømmer.

*[Ref: 41] Problemet med H.264-strømmer over RTSP er at det kun finnes (PST) og ikke (DST) som er nødvendig for lagring av strømmer til fil og transkodning. Vi prøvde også å installere et QuickTime-komponent fra Xiph.org for koding av Theora og Vorbis, men Vorbis er for dårlig implementert på Machintosh til at man kan dra nytte av det. En slik løsning vill kreve at man sender 2 strømmer fra Wirecast. En kodet i Theora og en i Mpeg4 til Darwin Streaming Server. Det fantes også en mulighet for å sende Theora-video og AAC-lyd, men det er usikkert om dette kunne ha fungert på klientsiden. Vi har fortsatt til gode å prøve ut dette. Vi ville i så fall ha gått utenom Darwin Streaming Server via VLC for streaming til Icecast.

Et annet problem som også dukket opp var at Darwin Streaming Server byttet om på kana-id når den fikk en SDP fil fra Wirecast. Wirecast og Darwin tolker SDP noe forskjellig. VLC mottok derfor audio som video og video som audio fra Darwin Streaming Server. Vi bestemte oss derfor for å gå rundt Darwin Streaming Server for å løse problemet og bare bruke Darwin Streaming Server til streaming av H.264 over Uninett, men selv om dette gikk bedre, hadde vi store problemer med audio synkronisering etter transkodingen uten at vi forstår helt hva som kan være grunnen til dette. Den sikreste løsningen vi har per av i dag er derfor Darwin Streaming Server kjørende på port 80 for traversering av brannmurer. Alternativt kan man bruke Mac OS X Server hvis man ønsker support fra Apple. Icecast vil kunne brukes til Video-on-demand, men sanntid streaming er per av i dag mindre realistisk å få til med de forutsetninger som er lagt i dette prosjektet, men muligheten er der hvis man kun trenger å streame fra en videokilde.

Wirecast kan bytte mellom 3 live video strømmer samtidig med å være source klient for en sanntid streaming server. Alle andre alternativer må løses ved å ta i bruk en hardware video switcher ellers kan selvfølgelig splitting og multiplekling av media strømmer med integrerte nettverksprotokoller for betjening av sanntids-streaming servere programmeres, men vi føler at dette overgår våre evner innen dette fagfeltet. Dessverre er Darwin Streaming Server og QuickTime Streaming Server de eneste plattformene som støttes. Vi skulle gjerne ha sett mer på flash streaming. Vi mener RED 5 kan være en ypperlig løsning for streaming. Det samme er også tilfellet for Flash Media Streaming Server, men vi konkludere med at valget av Wirecast som en software løsning for video switching, førte til at Flash Streaming og Icecast ikke var aktuelle. Muligheten er allikevel tilstede hvis man ønsker en hardware løsning for video switching noe som ikke var tilfellet i dette prosjektet. Av de løsningene vi har sett på, er det Mac Os X server og Flash Streaming Server som har de enkleste grensesnittene. De er også de som fungerer best hvis man ønsker å sende videostrømmer i sanntid. Alternativt kan man også bruke Wowza, Red 5 og Darwin Streaming Server til dette, men de kan ikke kommuniserer med Wirecast.

8.8 Konklusjon

Tanken med «Live» TV, slik det fungerer i tradisjonell kringkasting, er å kunne bryte de fysiske grenser for tid og rom slik at de mennesker som ikke har muligheten til å være til stede ved begivenheten, kan oppleve den som om de er til stede. På sitt beste, binder det sammen mennesker på tross av de fysiske barrierer som måtte finnes. Ønsket om å finne en løsning for streaming over Internett, henger sammen med de økonomiske bekostninger som kreves for å kunne drive en TV-kanal samt de juridiske konsekvenser hva omhandler kringkastingsloven. Denne rapporten konkluderer med at dette er per av i dag er fullt mulig å gjøre dette. Den løsningen man eventuelt måtte bestemme seg for, er avhengig av hva slags krav som settes til et slikt kringkastings system og de tekniske begrensninger som finnes i de server løsningene, utviklet for sanntids-streaming. Til vårt prosjekt, har vi kun funnet Darwin Streaming Server og QuickTime Streaming Server egnet til den bruk som DNBE har etterlyst.

9 Klientløsninger for streaming (kommer)

10 Produksjonssystem for lyd og bilde/video (kommer)

11 Oppsummering (kommer)

12 Brukermanualer

12.1 Wirecast

Om applikasjonen:

Wirecast3 fra Vara Software er en applikasjon som brukes for å ta i mot audio og video fra en eller flere SD-DV videokamera tilkoblet datamaskinens Firewire buss(er), samt audio fra line-in- og mikrofoninnganger på datamaskinens lydkort. Disse elementene blir valgt ut og mikset i ønsket rekkefølge og grad av overlapping etc. i det grafiske grensesnittet, eventuelt sammen med preproduserte elementer i form av grafikk, lydklipp, stillbilder og videosnutter hentet fra en harddisk, for så å bli streamet (sendt over datanettverk) til applikasjoner på andre datamaskiner, og/eller lagret som en fil til harddisk. Desktop Presenter, også fra Vara Software, er en frittstående applikasjon som kan lastes ned gratis og installeres på datamaskiner i det samme lokalnettverket og derfra levere video av skrivebordet til Wirecast for presentasjoner, etc.

Oppstart:

Klikk på Wirecast-ikonet i dock (parabolantenne med blå ringer), eller fra /Applications/Wirecast i Mac OS X Finder

Preferanser:

Gå til WireCast > Preferences i menyen, sett følgende:

- General > Aspect Ratio = 4:3 eller 16:9 avhengig av videosignal fra kamera
- Advanced > Capture Device Size = Native
- Advanced > Device Deinterlacing = Blend
- Kryss også av for 'Use High Quality Video' og 'Use GPU Compression'

Konfigurasjon:

- Slå på samtlige valg i Layout -menyen
- Gå til Window > Encoder Presets i menyen
- Velg ønsket forhåndsdefinert audio/video preset fra Encoder Presets -listen
- Trykk Save for å lagre, eller rediger parametre til ønskede verdier
- Lagre evt. valgt preset med endrede parametre under samme navn med Save eller velg Encoder Preset > Save Preset som valgt navn, deretter velg Save
- Gå til Broadcast > Broadcast Settings i menyen
- Klikk på plusstegnet for å legge til en jobb, dersom listen er tom
- Marker jobben og huk av i boksen til venstre for denne
- Velg preset fra Encoder Preset -listen
- Velg ønsket jobbtype fra Destination -listen
- Fyll inn tilhørende felt for adresser og portnumre
- Klikk evt. på tannhjulet og lagre SDP-fil til harddisk dersom Unicast eller Multicast
- Kopier eventuell SDP-fil til korrekt folder på datamaskin(er) som skal ta i mot strømmen(e)
- Lokasjonen til folder(e) avhenger av typen server eller applikasjon - les dokumentasjon for disse

Annet:

WireCast kan lagre hver enkelt videostrøm til disk (på samme måte som Capture Magic SD) i tillegg til selve videomiksen, men det må åpnes 3 ekstra vinduer, og hver jobb må startes for seg.

Feilsøking:

Problem: Tilkoblede og fungerende kamerakilder dukker ikke opp i Main Shot list under Preview Area og Live Area.

Løsning: Avslutt Wirecast3 og slett prosjektfiler (.wcst) som automatisk åpnes ved programstart. Restart og lagre disse på nytt

12.2 Capture Magic SD

Om applikasjonen:

Capture Magic er en applikasjon som kan ta imot videosignaler fra en eller flere SD-DV videokamera tilkoblet datamaskinens Firewire buss(er), og kan skrive hver enkelt videostrøm til harddisk for arkivering og senere redigering.

Oppstart:

Klikk på Capture Magic SD -ikonet i dock (film clapper), eller fra /Applications/Capture Magic SD

Preferanser:

- Gå til Capture Magic SD > Preferences i menyen
- Gå til Devices, sett Source Aspect Ratio til korrekt format
- Sett Buffering og Sampling slidere helt til venstre
- Gå til Recording, Velg folder for lagring og sett Recorded File Type til QuickTime
- Sett Record til Until Stopped, og Rate til Full Speed

Konfigurasjon:

- Pass på at samtlige tilkoblede kameraer er representert i listen over Firewire-enheter
- Dobbeltklikk hver enhet på listen for å få frem et preview vindu for hvert enkelt kamerasignal
- Klikk Record All -knappen for å skrive alle kamerastrømmer til harddisk (*writing* blinker)
- Klikk Stop All -knappen for å stoppe skriving av kamerastrømmer til harddisk
- Vent til Capture Magic har stoppet skrivingen (*writing* blinker ikke lenger - dette tar tid ved lang produksjon!)

Annet:

Capture Magic SD har verktøy som Waveform, RGB Parade, Vectorscope og Histogram til hjelp for justering av bildekvalitet for å påse at denne er innenfor akseptable verdier, også kamerasignalene sammenlignet med hverandre.

Feilsøking:

Problem: Kamera er tilkoblet, men dukker ikke opp i listen over tilkoblede Firewire-enheter

Løsning: Stopp andre applikasjoner som leser fra video input (sperrer tilgangen for Capture Magic SD)

12.3 VLC

Om applikasjonen:

VLC er en applikasjon for avspilling av audio/video fra filer på harddisk og/eller fra strømmer sendt over nettverket, men kan også brukes for å transkode strømmer fra et format til et annet, skalere, filtrere og på andre måter prosessere strømmene i etterkant, og skrive dem til harddisk eller streame dem videre til andre applikasjoner for videre prosessering.

Oppstart:

VLC startes normalt i Mac OS X fra ikon i dock (orange kjegle), eller fra /Applications/VLC i Finder.

Applikasjonen kan også startes fra kommandolinjen som /Application/VLC.app/Contents/MacOS/VLC, og kan ta en rekke parametre for å utføre forskjellige typer oppgaver (brukes i script, der valgene ikke finnes i det grafiske brukergrensesnittet)

Preferanser:

Kan settes i det grafiske brukergrensesnittet, men settes i dette tilfellet via parametre på kommandolinjen via et shellsript. Disse valgene overstyrer valg gjort i det grafiske brukergrensesnittet.

Konfigurasjon:

All konfigurasjon av VLC i denne oppgaven spesifiseres på kommandolinjen, i form av en rekke argumenter, opsjoner og en --sout -kjede (stream out) dersom den skal brukes for å streame til andre applikasjoner eller transkode media data. Fullstendig syntaks her er avhengig av hva applikasjonen skal brukes til, og de forhold og problemer man måtte støte på under arbeidet med å sette opp en oppgave, sett i forhold til de applikasjoner VLC skal ta i mot fra og sende til, og tilgjengelige ressurser (evt. mangel på), som dikterer bufferstørrelser, etc.

Annet:

VLC er en meget kapabel programvare som kan utføre en rekke forskjellige slags oppgaver innen avspilling, streaming og prosessering av media i en rekke forskjellige formater, og er derfor svært kompleks. VLC kan startes med argumentene --longhelp --advanced for å liste ut mulige kommandolinjeopsjoner og tilhørende verdier.

Feilsøking:

Ved oppstart av VLC med -vvv som argument vil applikasjonen gi utfyllende tilbakemelding og feilmeldinger

til terminalemulatorvinduet hvorfra den ble startet, slik at man kan diagnostisere eventuelle problemer.

12.4 Darwin Streaming Server

Om applikasjonen:

Darwin Streaming Server (DSS) er en media server for MPEG-4 video og AAC/MP3 audio fra Apple, som er gratis og fritt tilgjengelig. DSS er en usupportert utgave av QuickTime Streaming Server, som er en kommersiell streaming server med et grafisk brukergrensesnitt i tillegg til et webgrensesnitt og support fra Apple.

Oppstart:

```
$ sudo /usr/sbin/streamingadminserver.pl  
$ sudo /usr/sbin/QuickTimeStreamingServer
```

(i en terminalemulator)

Preferanser:

Alle preferanser settes i webgrensesnittet på adresseformatet `http://server_IP_adresse:1220/`

Konfigurasjon:

Ved installasjon av DSS velger man brukernavn og passord til en administrator, dette benyttes senere ved pålogging til webgrensesnittet. All konfigurasjon gjøres i webgrensesnittet som nevnt over for preferanser. Ved streaming av filer fra harddisk legges disse i en folder som er kontrollert av DSS, samt legges inn i en spilleliste (Playlist), hvorfra en klient kan forespørre dem ved navn på spilleliste eller ved navn på fil i spillelisten. Videre sending (og mangfoldiggjøring) av streams sendt til DSS gjøres ved å benytte en applikasjon som leser fra audio/video -kilden(e) og sender dem til DSS. Eksempler på slike applikasjoner er Apples egen QuickTime Broadcaster, eller VLC, som er vesentlig mer fleksibel.

Annet:

Prosjektet kommer til å kjøre QTSS snarere enn DSS, da oppdragsgiver har dette installert som standard på server. Forskjellene på DSS og QTSS er minimale, og konfigurasjon via webgrensesnittet så godt som identisk.

Feilsøking:

Problem: Playlist vil ikke starte

Løsning: Media fil er ikke hintet for streaming, eksporter filen på nytt, merket for streaming fra server

Løsning: Host datamaskin har fått ny IP, restart DSS prosesser (script for dette blir implementert i løsningen)

12.5 Icecast

Om applikasjonen:

Icecast er en streaming server for Theora video og Vorbis audio over HTTP protokollen

Oppstart:

```
/sw/bin/icecast2 -c /sw/etc/icecast2/icecast.xml &
```

Preferanser:

Settes i XML fil som nevnt i kommandolinjen vist ovenfor.

Konfigurasjon:

Settes i XML fil i likhet med Preferanser

Annet:

Streaming av filer fra harddisk til Icecast gjøres med en såkalt 'source client' applikasjon ved å beskrive fullt stinavn på filene i et XML konfigurasjonsscript for denne applikasjonen som denne leser ved kjøring/eksekvering. Videresending av streams gjøres i VLC ved å angi mount point i --sout -kjeden.

Feilsøking:

Ingen spesielle

12.6 Plugins for avspilling av media i nettlesere

Det siste leddet i prosesskjeden fra audio/video-kilder til avspilling av media for brukere på web er ikke helt utredet, da mye tid har gått med til problematikken rundt transkoding av mediastrømmene. Vi satser på en backupløsning som innebærer streaming av MPEG-4/AAC i 2 kvaliteter - en høykvalitets strøm for multicast-enablede nettverk, og en strøm med lavere kvalitet i samme format, tiltenkt brukere på web - dersom ikke problemene rundt transkoding til Theora/Vorbis lar seg løse.

I det første tilfelle kommer vi til å benytte en QuickTime plugin, i det siste tilfelle kommer vi til å benytte iTheora eller Cortado.

Vi har testet iTheora med streaming fra fil til Icecast med VLC, og fått dette til. Så vi vet at iTheora virker.

13 Referanser (Ikke fullstendig)

- 1: Adobe, «Flash Media Server»,
<http://www.adobe.com/products/flashmediaserver/> , sett: 16.04.08
- 2: Apple, «QuickTime Streaming Server»,
<http://www.apple.com/quicktime/streamingserver/> , sett: 16.04.08
- 3: icecast.org, «Icecast2», <http://www.icecast.org/> , sett: 16.04.08
- 4: Wikipedia, «UDP», http://en.wikipedia.org/wiki/User_Datagram_Protocol ,
sett: 16.04.08
- 5: Wikipedia, «NAT», http://en.wikipedia.org/wiki/Network_address_translation
, sett: 16.04.08
- 6: Soundscreen, «Streaming Server and Firewalls»,
<http://soundscreen.com/streaming/firewall.html>
, sett: 16.04.08
- 7: Wikipedia, «DMZ»,
http://en.wikipedia.org/wiki/Demilitarized_zone_%28computing%29 , sett:
16.04.08
- 8: Wikipedia, «RTMP»,
http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol , sett: 16.04.08
- 9: Apple, «QTSS Administration»,
http://images.apple.com/server/macosx/docs/QTSS_and_Broadcasting_Admin_v10.5.pdf_s31 , sett:
16.04.08
- 10: Gnuware, «What is Icecast», http://www.gnuware.com/icecast/chap_01.html
, sett: 16.04.08
- 11: <http://www.varasoftware.com/products/wirecast/> , sett: 16.04.08
- 12: imc.org, «RTP UDP audio max packet size per codec»,
<http://www.imc.org/ietf-rtpsec/mail-archive/msg00337.html> , sett: 16.04.08
- 13: Gory Fairhurst, «Ethernet Frame Calculations»,
<http://64.233.183.104/search?q=cache:jBPxoeEYKicJ:www.erg.abdn.ac.uk/users/gorry/course/lan-pages/enet-calc.html+http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/enet-calc.html&hl=en&ct=clnk&cd=1&client=safari> , sett: 16.04.08

- 14: Wikipedia, «Ethernet», <http://en.wikipedia.org/wiki/Ethernet> , sett: 16.04.08
- 15: The VideoLan Forums, «Can not save H.264 stream to H.264 file»,
<http://forum.videolan.org/viewtopic.php?f=14&t=20463&st=0&sk=t&sd=a&start=15> , sett: 16.04.08
- 16: theora.org, «Theora Video Compression», <http://www.theora.org/> , sett: 16.04.08
- 17: xiph.org, «QuickTime Components», <http://xiph.org/quicktime/download.html> , sett: 16.04.08
- 18: wiki.xiph.org, «Theora Software kodere»,
<http://wiki.xiph.org/index.php/TheoraSoftwarekodere> , sett: 16.04.08
- 19: v2v.cc, «FFMpeg2Theora», <http://v2v.cc/~j/ffmpeg2theora/download.html> , sett: 16.04.08
- 20: icecast.org, «Icecast Load Test Results»,
<http://www.icecast.org/loadtest1.php> , sett: 16.04.08
- 21: webdeveloper.com, «Does Multimedia Have a Dark Side?»,
http://www.webdeveloper.com/multimedia/multimedia_dark_side.html , sett: 16.04.08
- 22: Wikipedia, «Gigabit Ethernet», http://en.wikipedia.org/wiki/Gigabit_Ethernet , sett: 16.04.08
- 23: Wikipedia, «100 Gigabit Ethernet»,
http://en.wikipedia.org/wiki/100_Gigabit_Ethernet , sett: 16.04.08
- 24: Wikipedia, «Man in the middle attack»,
http://en.wikipedia.org/wiki/Man_in_the_middle_attack , sett: 16.04.08
- 25: adobe.com, «Flash Player Penetration»,
http://www.adobe.com/products/player_census/flashplayer/ , sett: 16.04.08
- 26: wowzamedia.com, «Wowza», <http://www.wowzamedia.com/> , sett: 16.04.08
- 27: osflash.org, «Red 5», <http://osflash.org/red5> , sett: 16.04.08
- 28: Wikipedia, «Darwin Streaming Server»,
http://en.wikipedia.org/wiki/Darwin_Streaming_Server , sett: 16.04.08
- 29: apple.com, «Open Source Streaming Server»,
<http://developer.apple.com/opensource/server/streaming/index.html> , sett: 16.04.08

- 30: Network Computing, «Streaming Media Servers (Screaming Streaming Media)»,
<http://www.NetworkComputing.com/showArticle.jhtml?articleID=177103372&pgno=1&queryText=streaming+server> , sett: 16.04.08
- 31: cmpnet.com, «Streaming Media Servers Features»,
<http://i.cmpnet.com/nc/1702/graphics/1702f3g.gif> , sett: 16.04.08
- 32: apple.com, «QTSS», <http://www.apple.com/quicktime/streamingserver/> , sett: 16.04.08
- 33: wikipedia, «Icecast», <http://en.wikipedia.org/wiki/Icecast> , sett: 16.04.08
- 34: Fink, «Package Icecast», <http://pdb.finkproject.org/pdb/package.php/icecast> , sett: 16.04.08
- 35: icecast.org, «Download Icecast2», <http://www.icecast.org/download.php> , sett: 16.04.08
- 36: free.fr, «ltheora», <http://menguy.aymeric.free.fr/theora/> , sett: 16.04.08
- 37: flumotion.net, «Cortado», <http://www.flumotion.net/cortado/> , sett: 16.04.08
- 38: uninett.no, «Uninett», <http://www.uninett.no/> , sett: 16.04.08
- 39: osflash.org, «Installing Red 5», <http://osflash.org/red5/docs?s=live%20video> , sett: 16.04.08
- 40: osflash.org, «Flash Video (FLV)», <http://osflash.org/flv?s=264%20support> , sett: 16.04.08
- 41: The VideoLan Forum, «Can not save H.264 stream to H.264 file»,
<http://forum.videolan.org/viewtopic.php?f=14&t=20463&start=0> , sett: 16.04.08

14 Ordliste (kommer)

15 Forkortelser (kommer)