

Avbrekk videostreamingsystem



NOTAT/NOTE

Norwegian Computing Center / Applied Research and Development

IMEDIA/04/02

Bror Gundersen

Oslo
April 2002

Tittel/Title:
Avbrekk videostreamingsystem

Dato/Date: April
År/Year: 2002
Notat nr/
Note no: IMEDIA/04/02

Forfatter/Authors:
Bror Gundersen

Sammendrag/Abstract:

På oppdrag fra Mercor AS utreder NR i dette notatet en rekke problemområder knyttet til leveranse av video over IP-nettverk, med utgangspunkt i en produksjonslinje fra produksjon til leveranse. Dette er diskutert videre i notatet med en praktisk vinkling ut i fra tilgjengelige kommersielle produkter som kan oppfylle Mercors ønske om kort tid til marked for sin videoklient, som benyttes i en applikasjon for tilrettelagte avspenningsøvelser for PC-brukere. Notatet er todelt hvor første del av notatet omhandler videokoding, innpakkingsformater for kodet video, videotransmisjonsprotokoller og verktøy for koding av video. Andre del i notatet diskuterer mer en mer teknologisesifikk løsning av Mercors videostreamingsbehov. En alternativ løsning til hovedløsningen er også kort diskutert, og sammenlignet med hovedløsningen. Deretter er det gitt et kostnadsestimat for forskjeller i softwarekostnader for løsningene.

Emneord/Keywords: Videostreaming, QuickTime, WindowsMedia, DRM

Tilgjengelig

Satsningsfelt

Antall sider/No of pages: 21

Innholdsfortegnelse

1. INNLEDNING	1
2. BESKRIVELSE AV PRODUKSJONSSCENARIO	1
2.1 Definisjoner	1
2.2 Produksjonssituasjon og leveranse	1
2.3 Systembeskrivelse	1
3. KODING	3
3.1 Kompresjon	3
3.1.1 Innledning	3
3.1.2 Redundans - tapsfri koding	3
3.1.3 Perseptuell koding - med tap	3
3.1.4 Bevegelseskompensasjon	3
3.1.5 Objektbasert koding	3
3.2 Kodere og dekodere	3
3.2.1 Innledning	3
3.2.2 MPEG 2	4
3.2.3 MPEG 4	4
3.2.4 Sorenson Codec 3	5
3.2.6 Microsoft Windows Media	5
3.3 Impakingsformat - wrapping	5
3.3.1 Innledning	5
3.3.2 MPEG 2 PES - Pacetized Elementary Stream og MPEG 2 Programstrøm	6
3.3.3 MPEG 4 filformat - MP4	6
3.3.4 QuickTime 5 - .mov	6
3.3.5 Microsoft Advanced Streaming Format - .asf	6
3.4 Verktøy for koding	6
3.4.1 QuickTime Pro	6
3.4.2 Java Media Framework	7
3.4.3 Microsoft Media Encoder - WME	7
3.4.4 Media Cleaner Pro – MCP	7
4. VIDEOLEVERANSE I IP NETT	7
4.1 Protokoller for transport og håndtering av video	7
4.1.1 Innledning	7
4.1.2 UDP – User Datagram Protocol	7
4.1.3. TCP – Transmission Control Protocol	8
4.1.4 RTP – Realtime Transport Protocol	8
4.1.5 RTSP – RealTime Streaming Protocol	8
4.1.6 RSVP – Resource Reservation Protocol	8
4.1.7 SDP – Session Description Protocol	9
4.1.8 HTTP – HyperText Transfer Protocol	9
4.1.9 MMS – Microsoft Media Streaming Protocol	9
5 DESIGNFORSLAG	9
5.1 Innledning	9
5.2 Komponentoversikt	10
5.2.1 Videokoding	10
5.2.2 Streamingtjener	10
5.2.3 Databaseløsning	10
5.2.4 Distribusjon	10
5.3 Systembeskrivelse	10
5.3.1 Innledning	10
5.3.2 Scenario 1 – Internettbasert leveranse	11
5.3.3 Scenario 2 – Leveranse på lokalnett	12
5.4 Integrasjon av serverløsninger	14
5.5 Dimensjonering av nettforbindelsen, optimalisering og spesielle hensyn	14
5.5.1 Krav til båndbredde	14
5.6 Autentisering mot Microsoft Streaming Server	15
5.7 Applikasjonsutvikling med Windows Media Player	16
5.8 Klientkobling mot tjenerdatabase	16
6. ALTERNATIV STREAMINGLØSNING – QUICKTIME6 MPEG4	16

7. KOSTNADSOVERSIKT	17
8. OPPSUMMERING OG KONKLUSJON	17
9. REFERANSER	18

1. Innledning

Dette notatet omhandler koding og nettbasert distribusjon av video, samt applikasjonsutvikling av videoavspillere. Dette er motivert ut i fra Mercors utvikling av et nettbasert video distribusjonssystem med tilhørende avspillere. Notatet vil fokusere på integrasjon av systemets komponenter og hver av disse komponentene ut i fra tekniske og kostnadmessige vurderinger. For å få den nødvendige helhetlige overbygningen for et slikt system vil notatet ta utgangspunkt i et produksjonsscenario.

Teknologimessig foregår det utstrakt forskning og utvikling særlig innenfor videokoding og annotering av film. Som følge av dette vil det være nødvendig å vurdere teknologier som ikke er ferdig implementert, men som etter prognoser vil være klar innen tidsrammene som er satt for videre utvikling av Mercors produkter. Det der spesielt innenfor videokoding at nye standarder vil forbedre kvaliteten.

2. Beskrivelse av produksjonsscenario

2.1 Definisjoner

Fra kringkasting av bilde i TV-produksjonssammenheng operer man gjerne med en rolleoppdeling av produksjon i kategoriene:

- Innholdsleverandør
- Kanaloperatør
- Distributør
- Informasjonskonsument

I det aktuelle produksjonsscenarioet vil informasjonskonsumenten være brukeren av videoklienten. Brukeren mottar video ut i fra en delt aktiv og passiv interaksjon med klientsoftwaren. Interaksjonsnivået foregår gjennom brukergrensesnitt i klientsoftwaren som er abstrahert vekk fra det teknisk underliggende arkitekturen som er gitt av kanaloperatøren. Kanaloperatøren er i dette tilfellet administrator av videotjener som leverer video, samt leverandør av netjtjenestene. Netjtjenestene kan være delt mellom flere ledd i kanalarkitekturen. Innholdsleverandøren er produsent av videoer som senere publiseres av kanaloperatøren.

2.2 Produksjonssituasjon og leveranse

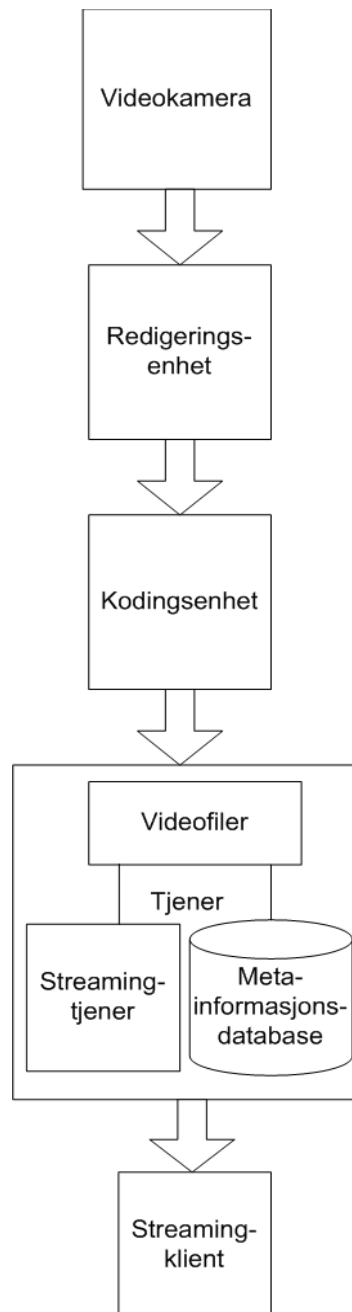
Innholdsleverandøren behøver en rekke verktøy for å produsere video med lyd for senere leveranse. Man kan ta utgangspunkt i video som råmateriale. Denne videoen overføres så digitalt til en computer, hvor den så konverteres til et filformat som kan leses, og brukes under redigering av råvideoen. Videoen kan så redigeres i en passende applikasjon. Resultatet er ukomprimert video som kan bearbeides for videre nettbasert publisering. Neste produksjonsledd er kompresjon av videoen, slik at filstørrelsen gjøres mins mulig for nettransport. Den ferdig komprimerte videoen kan så publiseres av en kanaloperatør. Videoene hentes nå ned på forespørsel fra informasjonskonsumenten, ved hjelp av interaksjon med et gitt brukergrensesnitt. Kanaloperatøren svarer så på denne forespørselen, og leverer video til informasjonskonsumenten ved hjelp av distributørens netjtjenester.

2.3 Systembeskrivelse

Systemet som det vil bli referert til senere i notatet vil bestå av følgende komponenter:

- Videokamera
- Redigeringsenhet
- Kodingsenhet
- Streamingtjener
- Video metainformasjonsdatabase
- Videoklient

Komponentene deles opp funksjonsmessig på denne måten, men i praksis kan for eksempel samme computer utføre flere oppgaver. Redigeringsenheten vil i mange tilfeller være den samme som kodingsenheten. Videostrømtjeneren vil også kunne inneholde videodatabasen. Videokamera og redigeringsenheten omfattes av kommersielle leverandører og vil ikke inngå i NRs videre systemutvikling for Mercor, og vil ikke bli omtalt nærmere bort sett fra der en videre produksjonen kan være avhengig av de nevnte leddene.



Figur. Produksjonsledd i nettbasert videoproduksjon

3. Koding

3.1 Kompresjon

3.1.1 Innledning

For å kunne transportere video mest mulig effektivt har man behov for å senke datamengden som video og lyd representerer. Video representeres ved hjelp av pikselinformasjon for hvert bilde i en sekvens. F.eks. for Mini-DV formatet med PAL standard representeres video med 25 bilder per sekund. Dette lagres så digitalt på et magnetbånd. Når videoen spilles av fra kamera eller på annen måte, vises hvert bilde. Dataene på magnetbåndet representerer 25000Kbit/s med data, selv om dette er komprimert med en rate på 5:1. Eksempelvis kan en enkel ISDN telefonlinje overføre 64Kbit/s. Forskjellen mellom hva en ubearbeidet video krever av båndbredde sammenlignet med hva man kan forvente av en ISDN forbindelse er dramatisk. For å kunne transportere video over begrensede medier for transport bruker man flere strategier. Det ene er å senke størrelsen på bildet. Videre strategier for kompresjon er beskrevet i pkt. 3.2 til 3.4

3.1.2 Redundans - tapsfri koding

Å bruke redundant informasjon i bildet er en annen måte å senke datamengden, det vil si å representere deler av bildet som er likt 2 eller flere ganger med en verdi. Enkelte verdier gjentar seg oftere i billedinformasjonen. Dette gjør sannsynligheten for gjentakelse av en verdi ikke-uniformt tilfeldig. Denne fordeling av verdier kan utnyttes ved at man koder de verdier som gjentar seg ofte med en kort kode, mens man gir en lengre kode i motsatte tilfelle. Denne fremgangsmåten er ofte referert til som Huffman koding, kalt opp etter matematikeren David A. Huffman. Denne og tilsvarende teknikker som brukes, gir tapsfri koding. Med det menes at det originale kodede bildet kan gjengis med nøyaktig lik originalen. Det finnes også andre teknikker for tapsfri koding som Run Length Coding og aritmetisk koding.

3.1.3 Perseptuell koding - med tap

Det menneskelige øye oppfatter bilder i bilde og video ut i fra en del fysiologiske og nevrologiske kriterier. Dette gjelder blant annet kontrast i farge. Ved å benytte kunnskap om dette kan man unngå å kode deler av bilder eller video, som uansett ikke oppfattes. Ved en optimering mot disse kriteriene kan man også vekte disse forskjellig etter synsmodeller, det vil si modeller for hvordan man oppfatter bilde og video. Når man kombinerer dette med redundanskoding kan man oppnå god kompresjonsgrad. Eksempelvis er en komprimert videostrøm fra en DVD 1,2 Mbit/s. Dette tilsvarer en kompresjonsgrad på 1,2:25 i forhold til mini-DV formatet. Ved ytterligere reduksjon i billedkvalitet kan man oppnå høyere grad av kompresjon. Menneskeøyet oppfatter forandring i lys lettere enn fargeforandring. Denne kunnskapen kan man benytte ved å legge mer vekt på lys enn farge i bildekodingen.

3.1.4 Bevegelseskompensasjon

Når man filmer objekter som beveger seg i bildet, vil det også gjøre at mange av elementene i bildet beveger seg i samme retning. Hvis man benytter felles retningsvektorer for mange av pikslene i bildet er sannsynligheten for at man kan senke dataraten og båndbredden relativt stor, samtidig som man bevarer informasjonen bedre for bevegelige motiver i videobildet.

3.1.5 Objektbasert koding

Video har ofte mer eller mindre fremtredende deler i motivet. Deler som beveger seg, eller forgrunnsfigurer for eksempel. Ved å skille bevegelige objekter eller andre forgrunnsobjekter i videoen fra bakgrunnen kan man også oppnå kompresjonsgevinst, siden bakgrunnen for eksempel ikke har de samme kravene til gjengivelse som hovedmotiv.

3.2 Kodere og dekodere

3.2.1 Innledning

Kodere og dekodere, også kalt codec, for video finnes i mange varianter som er tilpasset forskjellige virkeområder. De som omtales videre her, vil være de som er mest relevante i en videre

applikasjonsutvikling. Eldre kodere som har fått nye etterfølgere vil ikke bli omtalt. Flere av videokoderne som blir omtalt videre er standarder som definerer hvordan video skal spilles av med den respektive videodekoderen. Det vil si at standarden ikke definerer spesifikk hvordan koderen er implementert. Av denne grunn vil det være kvalitetsforskjell mellom forskjellige videokodere som kan levere video med samme avspillingsstandard. Real 8 codecen blir ikke vurdert her som følge av kostnadsvurdering av Real-teknologien.

3.2.2 MPEG 2

Moving Pictures Expert Group er et konsortium som arbeider med utvikling av videokodingsstandarder. De har gjennom flere år arbeidet med videokodingsstandarder, også kjent under versjonsnumrene 1,2 og 4. MPEG2 videokoding benytter samtlige av de foran nevnte teknikkene bort sett fra objektbasert koding. Dette innføres først i MPEG 4 standarden. Koderen lager video som egner seg for leveranse over nett ved hjelp av leveransemetoder som beskrives i kapittelet om videoleveranse i IP-nett. MPEG2 standarden definerer en standard for avspilling av video. MPEG2 definerer ikke hvordan koderen skal være implementert. Det medfører, som nevnt i innledningen, at det finnes et uttall av forskjellige MPEG2-kodere, med varierende kvalitet. Noen er optimalisert for å kode raskt for å lage video i sanntid, mens andre er laget for å gjengi best mulig video med lav eller høy båndbredde. Et av bruksområdene for MPEG2 er DVD-film. DVD-standarden gir koderen relativt høy tilgjengelig båndbredde, og følgelig er kodere for dette formålet implementert på en måte som utnytter dette. MPEG2 kodere som koder video for internettleveranse har mindre tilgjengelig båndbredde, og er derfor implementert på for å utnytte dette på best mulig måte. Hva angår størrelse kan man for videoleveranse av MPEG video si at MPEG2 krever nær 7-10 ganger høyere båndbreddebruk enn MPEG4 over Internett. Av denne grunn er det naturlig å velge MPEG4 hvis man ikke behøver MPEG2 kompatibilitet for leveranse på f.eks. digital-TV eller av andre grunner. MPEG2 kodere leverer video med ett lag båndbredde. Med dette menes at videoen kun kan spilles av i en båndbredde. Andre videokodere kan kombinere flere lag av video for at klient og tjenerapplikasjon kan tilpasse bruk av båndbredde etter reell tilgjengelig båndbredde.

3.2.3 MPEG 4

3.2.3.1 Innledning

MPEG4 standarden er et rammeverk av mediakodere og dekodere for presentasjon av forskjellige typer mediadata, på forskjellige enheter med vidt forskjellige egenskaper. Eksempelvis finnes det standarder for 2D/3D grafikk, koding av bilder, videoleveranse på mobile enheter, video over varierende tilknytning til Internett og animasjon av snakkende hoder. På lydsiden er det definert standarder for overføring av lyd for 2Kbit/s og oppover. Lydkodingen skifter koder under veis som man får mer data tilgjengelig for å kode lyden. I tillegg til å definere kodingen definerer MPEG4 standarder for hvordan forskjellige mediaobjekter kan kombineres i en mediastrøm ved hjelp av multipleksingsteknikker. MPEG4 definerer også et .mp4 filformat som bygger på Apple QuickTimes .mov format. MPEG4 standarden har definert en sesjonsprotokoll som holder rede på enkelte brukersesjoner. Siden hver bruker kan opprette sesjoner, åpner dette også for brukerorientert interaktivitet.

I og med at MPEG4 er et rammeverk for hvordan man kan sende og avspille media behøver man retningslinjer for koding og avspilling av de forskjellige mediatypene. Dette oppnås gjennom profiler. Eksempelvis definerer Visual Simple Scalable Profile koding av video i flere lag. Denne lagdelingen i videoen gjør at mottakere av video med forskjellig båndbredde på nettilkoblingen kan se en mer eller mindre detaljkodet versjon av videoen. Tilsvarende har man forskjellige audioprofiler. For eksempel Synthesis Profile gir retningslinjer for hvordan man sender parametere til en definert wavetable synthesizer slik at den spiller av musikk i dekodningen, eller omformer tekst til syntensert tale.

I de videre underpunktene vil forskjellige implementasjoner av videocodecer bli omtalt. Hvor vidt kodet video fra en koder fungerer under en annen dekode er ikke verifisert.

3.5.3.2 QuickTime 6 MPEG 4 Codec

Apple Computer Inc. er i ferd med å lansere sin MPEG4 baserte arkitektur for videoleveranse. QuickTime 6 implementerer flere av MPEG4 profilene. Koderne som er dokumentert er CELP talekoder for lav båndbredde og AAC audiokoder. I tillegg oppgir de at en ISO MPEG4 videokoder er implementert. I tillegg til MPEG4 funksjonaliteten QuickTime 6 tilbyr vil alle formater som er støttet i QuickTime 5 også være støttet. Quick Time 5 har ca 200 dekodere som kan prosessere forskjellige mediatyper som vektorgrafikk, lyd, bilde og video. QuickTime 6 video vil ha et standard MPEG4 filformat, som bl.a. er tilpasset streaming over IP-nett. Koderen vil gi god integrasjon med Darwin

Streaming Server og QuickTime Streaming Server. Sorenson har tidligere levert de sentrale videocodecene til QuickTime. Et whitepaper fra Sorenson angir at de har en Video Simple Profile MPEG4 codec i betatesting. Denne codecen vil etter all sannsynlighet dukke opp i QuickTime 6.

3.2.3.3 DivX 4

Project Mayo DivX 4 codecen hevder at de følger MPEG4 Video Simple Profile i sitt kodingsformat. Denne koderen er først og fremst rettet mot koding av videoer som skal spilles av fra en lokal disk. Samtidig foregår arbeide med en streamingkoder i Open DivX Streaming prosjektet. Det er sluppet noe kildekode i prosjektet, men det er usikkerhet rundt når dette kan være ferdig, og hvor godt denne koderen lar seg integrere med eksisterende streamingtjenere. Prosjektet retter seg mot bruk av MPEG4 bitstrøm og RTP som leveranseprotokoll. *Mer om RTP i pkt. 4.1.4.* Den ordinære DivX 4 koderen gir kompresjon i området 1:7 – 1:10 i forhold til DV-kodet video.

3.2.3.4 IBM MPEG 4 Java Codec

IBM har tidligere lansert en MPEG4 codec for JMF - Java Media Framework. Koderen fungerer under Windows plattform, og dekodekoden er antatt å fungere under alle plattformer som kan kjøre JMF. *mer om JMF i pkt. 3.4.2* codecen er midlertidig trukket tilbake på grunn av forhandlinger om MPEG4 lisensbetingelser. NR har testet koderen og oppnådd en kompresjonsgrad på 1:10 på en transkodet MPEG2 fil. Testen viste også at filen ikke lot seg streames med Darwin Streaming Server 4 PP2. Hvilken profil codecen følger er ikke opplyst av IBM, men testing peker mot at det er MPEG4 Video Simple Profile, siden det kun er ett lag med video. Det er også usikkert når IBM slipper codecen for eventuell distribusjon.

3.2.4 Sorenson Codec 3

Sorenson Codec 3 er den mest sentrale codecen for videokoding i QuickTime 5. Dette er en ettlags codec som er godt utprøvet i videostreaming. Den er godt integrert i QuickTime og kan streames med Darwin Streaming Server versjon 3 og versjon 4. Standardversjonen har fast bitrate videokoding, men det finnes en Professional Edition som kan gi variabel bitratekoding, VBR, og potensielt bedre kvalitet. Denne koderen følger med i Media Cleaner Pro 5. *Mer om Mediacleaner Pro 5 i pkt 3.4.4.*

3.2.6 Microsoft Windows Media

3.2.6.2 Windows Media 8

WM8 er den mest sentrale codecen i MMS-arkitekturen. Den koder video i en båndbredde, og er egnet for streaming med http, mms, eller msbd protokollene. WM8 kodet video kan også spilles av fra lokal fil. WM8 kan kode video med variabel bitrate med Windows Media 8 Encoding Utility, mer om denne i pkt. 3.7.3. WM8 codecen er ikke offentlig dokumentert, så selve algoritmen er ukjent. Denne codecen fungerer i Windows Media Player 7.0 og nyere. Det vil si at den ikke kan kjøre under Windows NT.

3.2.6.3 Microsoft ISO-MPEG 4 dekode

MS ISO-MPEG4 dekodekoden gjør det mulig å dekode MPEG4 filer som følger Video Simple Profile. Denne følger med i Windows Media Player 7.1. Test på NR har gjort, har vist dårlig visuell kvalitet av dekodet video ved avspilling av MPEG4 kodet film. Hvorvidt dette skyldes dårlig kodet video eller dekodekoden har vi ingen klare indikasjoner på.

3.3 Innpakningsformat - wrapping

3.3.1 Innledning

Når man har kodet videoen slik at den er klar for avspilling eller forsendelse, har man behov for filformater eller innpakking som kan bære kodet video sammen med metainformasjon. Metainformasjon kan i denne sammenhengen være titler, eier av videoen eller annen programinformasjon. Innpakningsformatene er ofte det man kjenner som filformater, og blir ofte forvekslet med kodingsformatet på videoen. For å presisere dette kan vi si at innpakkingen ofte har sterk binding til selve kodingsformatet, men at det ikke definerer hvordan selve videokompresjonen foregår.

3.3.2 MPEG 2 PES - Pacetized Elementary Stream og MPEG 2 Programstrøm

Programstrømmen er et rammeverk som inneholder PES for å kunne sende flere samtidige MPEG2-strømmer. Dette kan for eksempel være lyd og video, siden lydsporet ligger som en egen strøm ved siden av videostrømmen.

PES inneholder en streamidentifikator som skiller flere MPEG 2 mediastrømmer fra hverandre i en programstrøm. PES kan også inneholde et dekodings tidsstempel, lengdeinformasjon om PES pakkens lengde, flere typer flaggingsmekanismer og flere felt som kan brukes valgfritt. For sending av MPEG2 kan en transportstrøm brukes.. Filene med dette innholdet har ofte endelsen .mpg eller .mpeg. Dette kan i midler tid også godt være MPEG1 kodet video. MPEG2 filer kan spilles av med Windows Media Player, QuickTime 5 Player og Real Player 8, siden disse har dekodere for MPEG2 avspilling.

3.3.3 MPEG 4 filformat - MP4

MP4 formatet skiller seg fra MPEG2 formatet på mange måter, siden det er åpning for langt flere forskjellige mediadatatyper i formatet. Det bygger i hovedsak på Apples QuickTime format, hvor de blant annet har adoptert atombegrepet. Atomene representerer enten metainformasjon eller kodet mediadata. Metainformasjonen i et atom kan også inneholde pekere som leder til filer eller en URL utenfor selve filen. Formatet er også utvidbart og har støtte for utveksling av data i redigeringsammenheng. Det finnes flere typer av MP4 formatet som i hovedsak støtter en eller flere av MPEG4 profilene. Disse har forskjellige ISO/IEC betegnelser. Video Simple Profile filformatet har for eksempel betegnelsen ISO/IEC 14496-2.

3.3.4 QuickTime 5 - .mov

Som nevnt i pkt. 3.3.3 bygger .mp4 formatet på Apple QuickTime filformatet. Quicktime formatet består av såkalte atomer. Hver atom er en dataenhet som har informasjon om sin type og størrelse knyttet til seg. Atomene har en hierarkisk struktur som gjør at et atom kan inneholde et subatom. Siden hvert atom er knyttet til en datatype, kan dekodertype for det enkelte atomet avgjøres ved parsring av filen. Siden man kan velge mellom over 200 mediatyper, er typetilknytningen sentral for å kunne dekode dette. Microsoft støtter parsring av .mov i sine Windows operativsystemer med Apple QuickTime-spilleren. Dette er de pålagt ved en dom etter at de hindret avspilling av enkelte atomtyper ved å sende mediastrømmen til sin Windows Media Player. Windows Media Player kan ikke parsere .mov filer, og heller ikke spille av mange av QuickTime 5 mediatypene. QuickTime .mov formatet er forventet å erstattes av et standard .mp4 format, men .mov formatet vil fortsatt kunne parseres.

3.3.5 Microsoft Advanced Streaming Format - .asf

ASF formatet benyttes sammen med streaming på Microsofts mediastreamingsplattform. ASF kan spilles av lokalt eller streames med Windows Media Player. Det finnes versjon 1, 2 og 3 av formatet. Microsoft benytter versjon 1.0 av formatet i sin streamingarkitektur. ASF filformatet er offentlig dokumentert og patentert av Microsoft. ASF åpner for å kombinere flere mediatyper i en fil, for eksempel lyd og video i en film. Formatet kan spilles av med Windows Media Player streamet over http, mms og msbd protokollene, samt spilles av fra lokal disk. ASF har en hierarkisk arkitektur for organisering av metainformasjon og kodet materiale som gjør det mulig å parsere og sende materialet videre til dekodere og bevare tidsinformasjon som er nødvendig under avspilling, samt gi nødvendig informasjon slik at avspilleren laster inn riktig dekode for de respektive kodingsformatene. ASF formatet åpner også for koding av flere samtidige videokvaliteter. Tester NR har utført har imidlertid vist at avspilling av multibåndbreddekodete videoer ikke fungerer stabilt i en Internettomgivelse, i henhold til produktinformasjonen.

3.4 Verktøy for koding

3.4.1 QuickTime Pro

QuickTime Pro 5 tilbyr muligheter for både koding og transkoding av film fra forskjellige formater. QT Pro kan man også benytte til enkel klipping i kodet film, i tillegg til andre ekstrarfunksjoner som ikke finnes i gratis versjonen av QuickTimespilleren. I produksjonssammenheng kan den fungere som et

videokodingsverktøy som benyttes for grovklipping i video og for koding av video til for eksempel videre publisering på en streamingserver.

3.4.2 Java Media Framework

JMF er et rammeverk skrevet i Sun Java programmeringsspråket. Det tilbyr API for utvikling av mediaapplikasjoner. Ved installasjon av JMF følger det med en applikasjon, JM Studio, som kan benyttes til koding og transkoding av videoer for å lage innpakkede filmer som kan streames eller spilles av lokalt. JM Studio kan også streames filmer over RTP protokollen. JM Studio fungerer også for opptak av lyd eller video.

3.4.3 Microsoft Media Encoder - WME

WME 7.1 er et verktøy for å kode filer i de forskjellige Microsoftformatene, blant annet for streaming. WME fungerer både for koding av filer og livestrømmer med video. Man kan med andre ord også benytte den til live sending av video til en Windows Media Server, og videredistribuerer videoen ut til klienter eller proxyservere. Den har et grafisk brukergrensesnitt for håndtering av filer og profiler for koding av video i forskjellig kvalitet og codec. Windows Media 8 Encoding Utility, WM8EU er en kommandolinje basert videokoder, som kan gi noe bedre kvalitet enn hva man kan få med WME 7.1. Denne støtter blant annet variabel bitrate koding av videoer, som potensielt kan forbedre gjengivelsen ved dynamisk allokering av bit etter behov i koderen. WM8EU kan kjøres med flere deler som kan forbedre kvaliteten i bildet. Siden denne er kommandolinjebasert er det noe enklere å lage script som kan brukes til koding av flere filmer uten brukerinteraksjon, såkalt batch encoding.

3.4.4 Media Cleaner Pro – MCP

MCP har videokodere for de fleste aktuelle kodingsformat som brukes i videostreamingsammenheng. Verktøyet har god mulighet for å lage jobber hvor flere filer kodes med forskjellige kodere, eller samme koder men forskjellige profil. Den har blant annet Windows Media 8 koder med VBR og Sorenson Video 3 med VBR. MCP er kun beregnet for koding av filer, og ikke livestrømmer, selv om den kan importere video fra et kamera via Firewire. Applikasjonen har mange avanserte muligheter for kombinasjon av flere mediatyper og annen funksjonalitet som strekker seg ut over Mercors behov for videokoding. MCP koster 6570,- inkl. mva. pr. 6. mars 2002.

4. Videoleveranse i IP nett

4.1 Protokoller for transport og håndtering av video

4.1.1 Innledning

Transport av kontinuerlig video over nett med hovedsakelig asynkron oppførsel gir mange utfordringer. Ordinær filoverføring eller nedlasting av websider fordrer ikke at dataene ankommer brukeren med tidssynkronisering. Under lastning av en webside forventer ikke brukeren nødvendigvis at alt innholdet kommer samtidig. Ved lav båndbredde er brukeropplevelsen preget av at teksten ankommer før bilder og andre større elementer. For video gjelder helt andre kriterier. Man behøver en jevn flyt av videodata slik at det ikke oppstår frys i bildet. Samtidig må videobildet ha god nok visuell kvalitet avhengig av hva som skal presenteres. For å oppnå de skisserte kriteriene er det nødvendig å optimalisere båndbreddebruk og videokoding. Gjennom tilpassede leveranseprotokoller får man en mer optimal bruk av båndbredde. Protokollene som omtales videre her er leveransemetoder som brukes på IP-protokoll baserte nett, for eksempel Internett eller lokalnett.

4.1.2 UDP – User Datagram Protocol

UDP er en datagram orientert transportprotokoll for datapakker. Når en UDP forbindelse er åpnet, sikrer ikke protokollen at alle pakkene . Denne protokollen egner seg til transport over nett der man er rimelig sikker på at datapakken ikke blir hindret på noen måte, og at det ikke er strengt tatt nødvendig å motta 100% av alle pakker som blir levert. Eks: Korrigerende algoritmer kan rette eventuelle feil som skyldes tap av noen få videopakker uten at det går ut over den visuelle videokvaliteten. Brukeren opplever videoen som sammenhengende, med ingen eller få feil, selv om noen videorammer mangler. UDP egner

til kringkasting hvor forbindelsen er orientert mellom en tjener og mange brukere, såkalt multicast. Samtidig er det mulig å opprette en til en forbindelse, såkalt unicast. UDP og TCP er hovedleveranseformene over IP-nett. *Mer om TCP i pkt. 4.1.3.*

UDP protokollen har noen svakheter for sanntidsstreaming av video som adresseres videre i pkt. 4.1.4. RTP. Et sentralt punkt er hva som skjer med tap av mange UDP-pakker, og dermed brudd i videostrømmen.

En praktisk årsak til å ikke sende video over UDP er at pakkene ofte stanses i brannvegger på grunn av sikkerhetsrisiko ved passivt mottak av UDP pakker.

4.1.3. TCP – Transmission Control Protocol

TCP protokollen er en forbindelsesorientert protokoll for transmisjon av data. Med dette menes at det vil foregå trafikk i begge retninger, blant annet for å sikre at alle sendte pakker kommer frem til mottakeren. Ideelt sett burde dette love godt for transport av video, men siden retransmisjon av videoen kan ta tid, kan dette medføre at de tapte pakkene ikke kommer frem i tide for avspilling. I mange tilfeller vil det være mer gunstig å akseptere tapte pakker, for å ikke stanse videostrømmen. TCP er egnet for forbindelsesorientert transport av video, såkalt unicast. TCP benytter mer båndbredde enn UDP og skalerer dårligere, særlig i store systemer. Problemer med TCP pakkeleveranse kan forplante seg i nettforbindelsen fra tjeneren, og skape køer av datapakker som ikke kommer videre, eller bruker for lang tid på leveranse. Dette er et typisk problem på Internettbasert videoleveranse TCP pakker slipper oftere gjennom brannvegger, da TCP forbindelser generelt vurderes som mer sikre enn UDP. Mer rigide brannveggkonfigurasjoner har allikevel stengt for TCP forbindelse på mange av portene. Typisk slipper man gjennom all TCP basert HTTP for www på port 80. *Mer om dette i pkt. 4.1.8 HTTP.*

4.1.4 RTP – Realtime Transport Protocol

RTP er en transportprotokoll beregnet for sanntids IP-basert leveranse av mediadata. Forbindelsen kan være multicast eller unicast. Det er vanlig å pakke RTP pakker inn i UDP pakker, men RTP kan også pakkes inn i andre transportpakker. Med UDP innpakking er RTP en ikke-forbindelses orientert transport. RTP tilfører denne transporten tre nye egenskaper, tidsstempel, sekvensnummer og Payload Type Identifiser. Tidsstempelet brukes til å synkronisere pakkene slik at pakkene kan pakkes ut og legges i en riktig rekkefølge slik at tidsinformasjonen i for eksempel en videostrømm bevares.

Sekvensnummeret i RTP pakken benyttes til å organisere UDP pakkene etter at disse er mottatt., siden pakkene kan mottas ikke-sekvensielt. Sekvensnummeret kan også benyttes til å holde rede på eventuelt pakketap. To RTP pakker kan ha samme tidsstempel, hvilket også er et argument for å ha sekvensnummer på pakkene. Payload Type Identifiser gir mottakerapplikasjonen informasjon om videopakkenes kodingsformat, og annen informasjon om selve videoinnholdet i RTP pakkens payload.

4.1.4.1 RTCP - Realtime Control Protocol

RTCP protokollen benyttes sammen med RTP for å kunne sende rapporter til tjener og klient for overvåking av servicekvalitet, QoS. RTCP kan også benyttes for sanntidssynkronisering av flere mediastrømmer, samt være med på å styre mediastrømmene for å kunne skalere systemer bedre ved høy belastning.

4.1.5 RTSP – RealTime Streaming Protocol

RTSP benyttes hovedsakelig for å styre avspilling av lyd eller video over nett. Det vil si funksjonalitet som i en konvensjonell videospiller, med blant annet play og pause funksjonalitet. Dette er som regel implementert på applikasjonsnivå, i kombinasjon med underliggende transportprotokoller f.eks. RTP. Dette benyttes i Real Player og QuickTime Player.

4.1.6 RSVP – Resource Reservation Protocol

RSVP kan benyttes for å reservere båndbredde, og sikre tilgjengelig båndbredde for mediastreaming langs strømmens transportrute. Siden denne reservasjonsprotokollen er avhengig av infrastrukturen langs transportveien, fordrer det at man undersøker dette før man kan implementere dette som en kvalitetssikring av videoleveransen. Lokalnetimplementasjon kan være aktuelt, siden det er lettere å få tilgang til eller få informasjon om tilgjengelig infrastruktur. Internettleveranse av video vil nyte fordeler av en RSVP, men som antydnet er dette avhengig av infrastruktur i pakkeruten. RSVP kan brukes både i

multicast og unicast.

4.1.7 SDP – Session Description Protocol

SDP er en ASCII basert protokoll for å kunne annonsere sesjoner for blant annet videobrukere. SDP filene legges tilgjengelig for brukeren slik at denne kan lastes ned, og innholdet tolkes av en klient med SDP implementert. Innholdet i filen kan blant annet være mediatype, transportprotokoll for media, sesjonstype, URL til mediafilen og port for mediatransporten. Det er en rekke andre parametere som også spesifiseres, disse er listet i RFC 2327.

4.1.8 HTTP – HyperText Transfer Protocol

HTTP er en TCP basert protokoll. *Mer om TCP i pkt. 4.1.3.* Siden HTTP er forbindelsesorientert, skalerer denne dårligere enn UDP ved et høyt antall klienter. Retransmisjon kan også få mediastrømmer til å stanse opp, eller brytes midlertidig. Ved å benytte asynkrone buffere kan man avverge små brudd i pakkestrømmen, men på serversiden vil pakkekøer og stopp i nettverksforbindelsen kunne gi høy båndbreddebruk. Dette definerer altså to kilder til ineffektivitet ved bruk av HTTP protokollen for streaming. Den ene er toveiskommunikasjonen, den andre skyldes potensielt mange retransmisjonsforespørsler ved problemer i nettet, særlig nære tjeneren. HTTP benyttes allikevel i stor grad ved både .mp3 streaming og videostreaming. Et av hovedargumentene for å benytte denne protokollen er at majoriteten av brannvegger som skiller Internett og lokalnett har åpning for HTTP-forbindelse på port 80 på grunn av www trafikk på denne porten. HTTP benyttes til unicast. Transporten foregår enten ved tunnelering av andre transportprotokoller gjennom HTTP, eller ved vanlig innpakking av videopakker i HTTP/TCP.

4.1.9 MMS – Microsoft Media Streaming Protocol

MMS er en proprietær protokoll utviklet av Microsoft uten offentlig tilgjengelig dokumentasjon bort sett fra beskrivelse av funksjonalitet. Det er oppgitt at Windows Media Player forhandler med Windows Media Server over mms:// for valg av best mulig leveranseprotokoll. Forhandlingen foregår etter følgende prioritet:

1. Modifisert UDP-basert protokoll, med tilsvarende funksjonalitet som RTP protokollen.
2. Antatt modifisert TCP protokoll.
3. Antatt modifisert HTTP protokoll.

Antakelsene er basert på produktdokumentasjon fra Microsoft og studie av tjener-klient funksjonalitet. Produktdokumentasjonen oppgir også at tjeneren kan negociere med klienten om best mulig videokvalitet i forhold til tilgjengelig båndbredde. Tester utført i driftssammenheng på NR har vist at dette ikke fungerer tilfredsstillende. Siden det er divergerende opplysninger i produktdokumentasjonen og faktisk funksjonalitet, vil vi ikke beskrive protokollen nærmere før det finnes dokumentasjon av den tekniske implementasjonen. MMS protokollen gir tilstrekkelig funksjonalitet for videostreaming, men er en lukket standard som ikke gir noen interoperabilitet med andre streamingteknologier.

5 Designforslag

5.1 Innledning

Under vurdering av flere konkurrerende teknologileverandører har spesielt leverandører av komplette streamingløsninger pekt seg ut. Dette gjelder Apples QuickTime arkitektur og Microsofts Windows Media. Begge leverandører kan tilby løsninger som peker i retning av Mercors behov for streamingteknologi. Reals streamingteknologi har også vært vurdert, men på grunn av lisens per tilkoblede bruker er Real ikke med i videre vurderinger. Mercors markedsmessige vurderinger og ønsker dreier mot en Microsoft Media, heretter kalt MSM, basert løsning. Videre beskrevet løsning bygger derfor rundt MSM. Siden dokumentasjon for en del av løsningene ikke er offentlig tilgjengelig vil deler av løsningen være tuftet på forutsetninger om at Microsoft har gjort korrekte design og implementasjonsvalg i sine proprietære løsninger. Der dokumentasjon er tilgjengelig vil det refereres til denne i størst mulig grad. Pkt. 6 inneholder noe informasjon om en QuickTime 6 basert løsning siden den har en del andre innfallsvinkler med hensyn til teknologi og prising.

5.2 Komponentoversikt

Windows Media teknologien har komponenter som kan brukes i leveranse av video over IP nett. Verktøyene strekker seg fra koding av video, distribusjon, streaming og lokal avspilling av video. PostgreSQL er en database som kan egne seg i Avbrekk-systemet, også for et stort antall samtidige brukere.

For å sikre Windows NT eller 2000 tjenere mot uautorisert bruk eller nettinnbrudd er det behov for flere komponenter enn det Microsoft leverer med sine tjenere.

5.2.1 Videokoding

Windows Media 8 den mest sentrale codecen i Windows Media systemet. *Codecen er nærmere beskrevet i pkt. 3.4.3.* For å oppnå optimal kodingskvalitet på WM8 kodet video, anbefales Windows Media 8 Encoding Utility. Denne er også egnet for å skripte koding av flere filer ved blant annet transkoding av Mercors MPEG1 filer, det vil si å kode om filen fra MPEG1 formatet til WM8 video i .asf streamingfilformatet.

5.2.2 Streamingtjener

Windows Media Server følger med i Windows 2000 Server og kan levere video til klienter over Microsoft Media Streaming protokoll, heretter kalt MMS. MMS er, som tidligere nevnt, en modifisert utgave av Hypertekst Transfer Protocol. Denne er ikke offentlig dokumentert av Microsoft, men SDP prosjektet har forsøkt å avdekke hva som skjuler seg under forkortelsen.

MMS protokollen bruker TCP og UDP over IP for transport av datapakker. I tillegg til ren transport av pakker foregår det også negotiering mellom Media Player klientapplikasjonen og MMS-tjeneren. Denne negotieringen bestemmer hvilken transportprotokoll som skal benyttes i sending av video fra tjener til klient. Prioriteringen av protokoller skjer etter prinsippet om best mulig utnyttelse av tilgjengelig båndbredde, og foregår i følgende rekkefølge:

1: UDP – 2: TCP – 3: HTTP

5.2.3 Databaseløsning

NR anbefaler å benytte PostgreSQL database for håndtering av metainformasjon knyttet til videoene. Vurderingen er gjort med vurdering av skalerbarhet for databasen opp mot 1000 brukere. Vi har tatt hensyn til både skalerbarhet i softwaren og hardwaremessig. PostgreSQL har også støtte for databasereplikering over TCP/IP. SQL99 er implementert, og autentisering mot databasen kan foregå over kryptert SSL eller SSH forbindelse. PostgreSQL kan kjøres på Windows 2000/NT og en rekke andre operativsystemer inkludert Linux og Solaris på de tilgjengelige prosessorplattformene.

PostgreSQL er en Open Source implementert database uten lisenskostnader.

5.2.4 Distribusjon

For å kunne automatisere distribusjon av video over Internett og lokalnett har man behov for filreplikeringsmekanismer. Windows 2000 Server har mulighet for replikering av filer over nett. For å opprette en sikker forbindelse med en fjerntliggende maskin kan man sette opp en sikker tunnel med Virtual Private Networking forbindelse, VPN. Databasen kan replikeres over en secure shell, SSH, tunnel. Siden det ikke er ssh applikasjoner installert med standard Windows 2000 server anbefales Open SSH som et alternativ for SSH-tunnelering. F-Secure leverer også SSH klientapplikasjon og SSH serverdemon for Windows 2000 Server. Et annet alternativ er å distribuere en replikert versjon av databasen i tillegg til videofilene på CD.

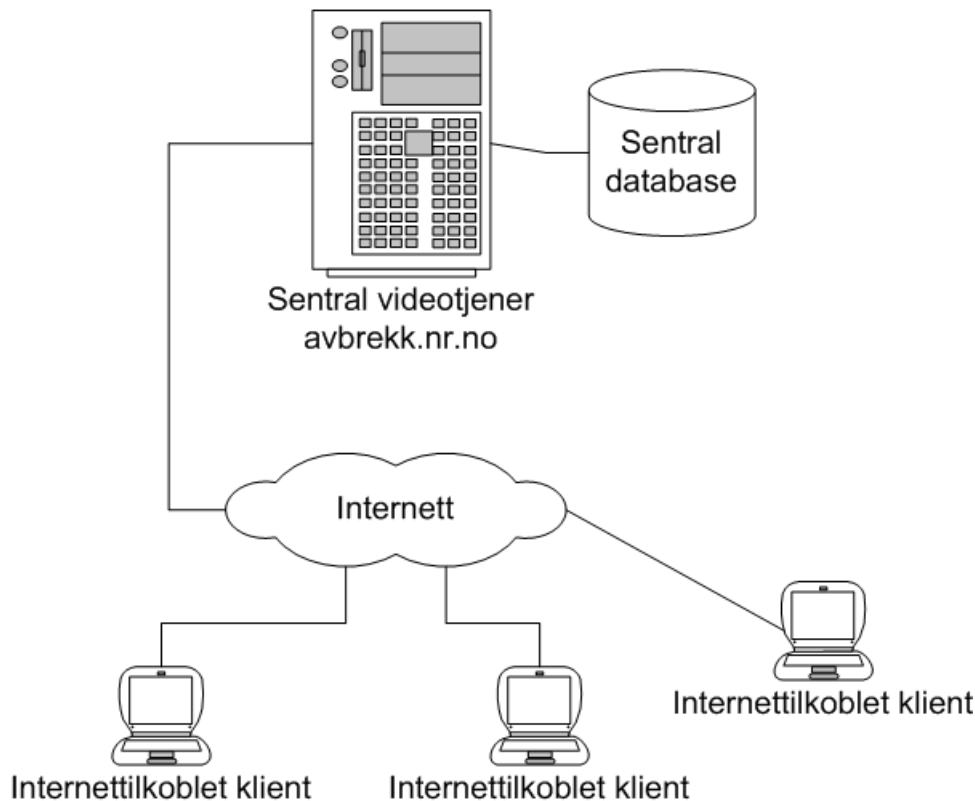
5.3 Systembeskrivelse

5.3.1 Innledning

Avbrekk videosystemet er tiltenkt å kunne levere video i lokale nettverk og over Internett. Systemmessig stiller det forskjellige krav til å levere video over LAN og over Internett. Av denne grunn vil systemet beskrives ved hjelp av to hovedscenarier. Leveransemetodene kan også variere innenfor disse scenariene. Det sentrale er hele tiden å kunne utnytte tilgjengelig båndbredde for best mulig kvalitet, samtidig som man ikke belaster det tilgjengelige nettverket unødig.

5.3.2 Scenario 1 – Internettbasert leveranse

Systemscenario 1 innebærer leveranse fra en sentral server med Internett som leveransevei for video. Dette scenarioet kan deles opp i 3 subscenarier som beskriver forskjellige måter å levere video på Internett.



Skjematisk oversikt over Internett-levert streamingvideo

5.3.2.1 Unicast direkte klientleveranse

Unicast innebærer at hver klient mottar en separat videostrøm fra tjeneren. Dette medfører at båndbredden som benyttes fra tjeneren vil tilsvare antall klienter ganger båndbredde ved UDP transport av videopakkene. Hvis sesjonen går over til å benytte andre protokoller som TCP eller HTTP vil dette gi bruk av mer båndbredde. TCP og HTTP protokollen er i motsetning til UDP forbindelsesorienterte protokoller som blant annet utfører retransmisjon av eventuelle tapte pakker. Microsoft har modifisert UDP protokollen slik at klienten ber om retransmisjon, kalt UDP Resend Request.

Unicast leveransen egner seg godt for enkeltklienter, for mange klienter som befinner seg på et LAN vil Internettforbindelsen til lokalnettet bli belastet i med samme grad som antall klientesjesjoner. I dette tilfellet vil det være aktuelt å vurdere et lokalt filspeil. Unicast leveransen kan hente videostrøm fra en lokal fil, eller en montert livesending.

5.3.2.2 Multicast direkte klientleveranse

Windows Media klienten har muligheter for å motta multicast transmisjon av video over Internett. Multicast kan best sammenlignes med kringkasting. Med multicast har klientene mulighet for å lytte til en flerbruker transport av en enkelt video. Ideelt gir dette et båndbreddeforbruk på en gangers videoens båndbredde. Multicast er avhengig av understøttende hardware. Ruterne må støtte såkalt MBONE leveranse av datagram. I praksis betyr det at man må analysere den tilgjengelige infrastrukturen fra tjener til streamingklient. På Internett gir dette en rekke usikkerhetsmomenter, siden man ikke alltid har tilgang til, eller kjenner ruterkonfigurasjonene over hele videopakkens rute. Multicast er en ikke-forbindelsesorientert UDP forbindelse. Av sikkerhetsgrunner blir ofte UDP trafikk stanset i brannvegger som skiller LAN fra Internett.. For å omgå dette problemet kan man tunnelere Windows Media multicast

gjennom brannveggen over andre protokoller. Dette blir videre beskrevet i pkt. 5.3.2.3.

5.3.2.3 Unicast videresending til proxy

Videresending av Microsoft Media Server strømmer til en proxy har to hovedvarianter. Den ene sørger for tunnelering av multicast trafikk gjennom brannmurer. Den andre varianten er én unicast til mange unicast videostrømmer. Tunnelering av multicast forutsetter bruk av MSBD protokollen eller HTTP. Montering av en unicaststrøm skjer ved hjelp av MMS protokollen.

5.3.2.4 Konfigurasjon av brannvegg

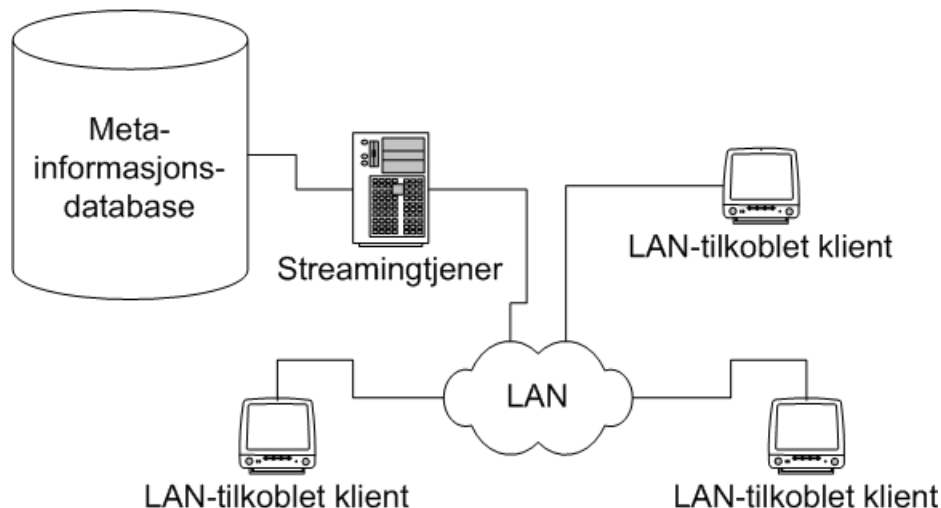
Hvis man ønsker å konfigurere en brannvegg for mottak av videotrafikk fra en Windows Media Server behøver man å åpne nødvendige porter i denne. Siden behovet varierer etter hva man skal motta skiller man på følgende måte sett fra klientsiden:

Trafikktype	Åpne porter inn	Åpne porter ut
Multicast	1-65000 I adresseområdet: 224.0.0.1 til 239.225.225.225	
Unicast UDP	1024-5000	1755
Unicast TCP	1755	1755
Unicast HTTP	80	80
MSBD	7007	7007
DCOM – Distributed COM	HTTP 80	80
For fjernadministrasjon av Streaming Server.	UDP og TCP 135	

5.3.3 Scenario 2 – Leveranse på lokalnett

5.3.3.1 Innledning

Leveranse på lokalnett, LAN, skiller seg i noen grad fra Internettleveranse ved at man i langt større grad her mulighet for detaljkunnskap om infrastrukturen videopakkene skal sendes over. Teknisk sett er begge typer nett IP-baserte, og har mange fellestrekk. Vi antar videre at lokalnettet ikke er avgrenset internt med brannvegger. Dette kan avvike, særlig i større LAN med behov for oppdeling av nettverksinfrastrukturen.



Skjematisk oversikt over lokalnettlevert streamingvideo

5.3.3.2. Unicast på lokalnett

Unicast fra Microsoft Media Server foregår over MMS-protokollen eller HTTP-protokollen. MMS tilbyr som tidligere nevnt et spekter av flere leveranseprotokoller, alt etter hva klienten er i stand til å akseptere. Unicast benytter båndbredde tilsvarende n ganger antall klienter. Dette medfører at båndbreddebruk øker lineært med antall brukere.

Eksempel på Unicast:

Videobåndbredde = 250Kbit/sekund

Antall brukere = 1000

Totalt forbrukt båndbredde = 25Kbit/s * 1000 = 250000Kbit/sekund = 250Mbit/sekund

Mer om dimensjonering i pkt. 5.4

Unicast kan levere forskjellig video til hver bruker, til forskjellig tidspunkt. Unicast kan levere video fra flere forskjellige videokilder.

1. On demand streaming fra lokal fil. Videofilene må ligge lokalt på proxy eller videoserver.
2. On demand streaming remote – Streaming av filer som ligger på en annen tjener. På et lokalnett vil en proxy typisk streame video fra en mer sentral tjener.
3. On demand live videostreaming – En live videostrøm monteres på en proxy eller server, for så å kunne hentes via en unicast videostrøm på et lokalnett.

5.3.3.3 Multicast på lokalnett

Med multicast via MMS får mange brukere anledning til samtidig å se samme videosending. Båndbreddekravet fra tjeneren er lik videoens båndbredde. Lokalnettets infrastruktur må støtte leveranse av multicast datagram, også kalt MBONE. Multicastsendingen kan bruke følgende videokilder under sending:

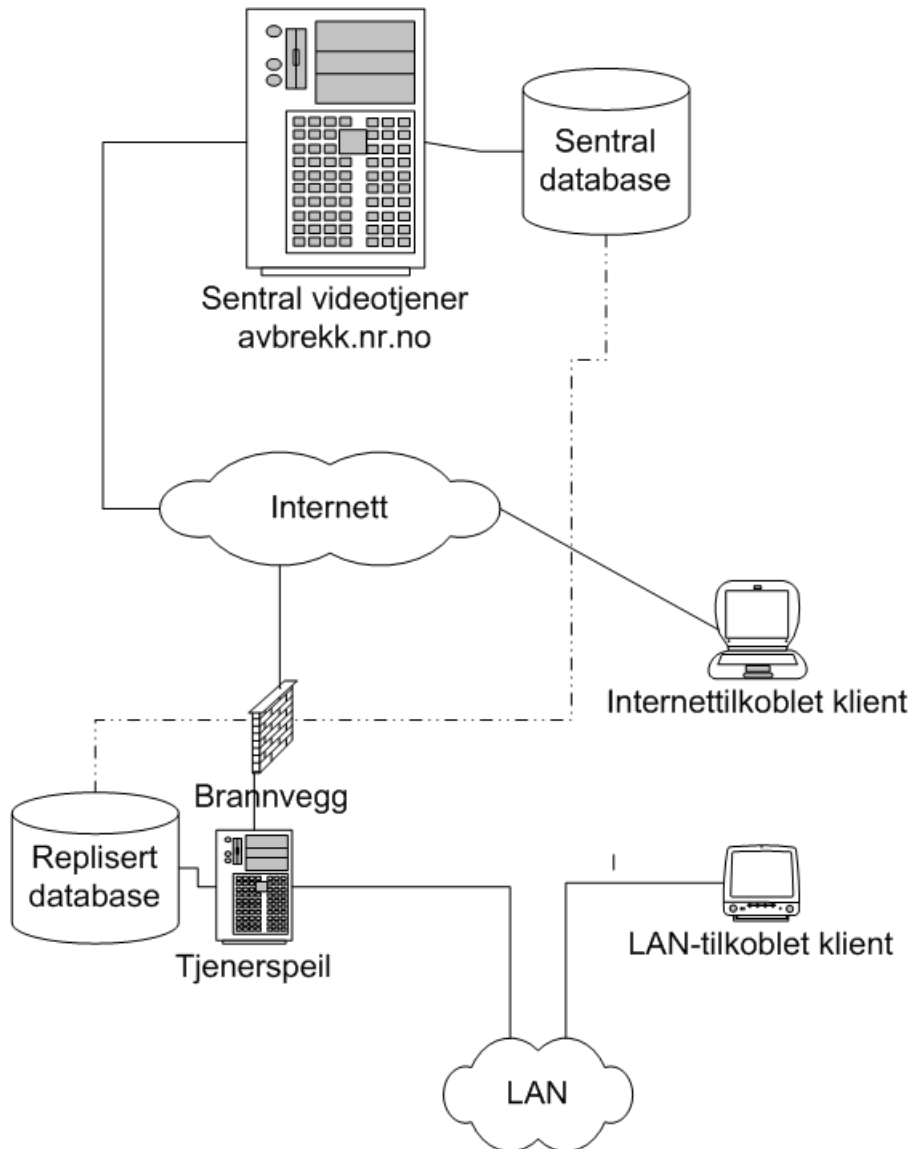
1. Lokal fil.
2. Fil på annen streamingserver eller streamingproxy, tunnelert gjennom MSBD-protokollen.
3. Live videostrøm fra en annen multicastkilde.
4. Live videostrøm fra en annen unicastkilde.

Unicastsendingen blir beskrevet i en .nsc-fil. Denne tilsvarende .sdp-filer som benyttes under RTP/RTSP multicast. .Nsc-filen benyttes for å angi flere parametere spilleren behøver for å vite når multicastsendingen starter og hvordan den er kodet, i tillegg til en rekke andre nødvendige parametere.

Mer om .sdp finnes i pkt. 4.1.7

5.4 Integrasjon av serverløsninger

Mercors behov peker i retning av behov for å integrere både lokalnett og Internettbaserte streamingløsning. Scenario 3 beskriver integrasjon mellom en sentral tjener, en proxy og klienter med Internettbasert og lokalnettbasert leveranse av videostrømmer. De Internettbaserte klientene mottar unicaststrømmer med video, mens de lokalnettbaserte klientene mottar både unicast og multicast-baserte strømmer. Illustrasjonen viser også at Microsoft Media Server har mulighet for å streame over flere protokoller og med flere metoder samtidig.



Integrert Internett og LAN-streaming

5.5 Dimensjonering av nettforbindelsen, optimalisering og spesielle hensyn

5.5.1 Krav til båndbredde

5.5.1.1 Innledning

Sentralt for dimensjonering av tilgjengelig båndbredde er at man har nok ressurser til å levere video i de

enkelte delene av nettforbindelsen mellom tjener og klient.

For å kunne vise video av med tilstrekkelig billedkvalitet bør man foreta studier av hvordan videoene oppfattes i brukersituasjon. Særlig Internettbasert leveranse av video gir store begrensninger i forhold til hva slags kvalitet som kan tilbys i sammenligning med en ordinær TV-sending. Kringkastingskvalitet som leveres på DVD formatet tilsvarer 1,2 Mbit/sekund med MPEG2 kodet video. Microsofts Video 8 baserte format gir noe bedre kvalitet på kodet video i forhold til MPEG2. Eksempelvis vil en 320 ganger 240 piksler stor video kunne gjengis med akseptabel kvalitet med 250Kbit/sekund. I dette eksempelet referer vi til Microsoft Video 8 Codec som benyttes til leveranse av video for bredbåndsbrukere. Med bredbånd menes ADSL eller kabelforbindelse med minimum 340Kbit/sekund ideell nedlastingshastighet. Her kan vi se at videoen er kodet med en langt lavere bitrate enn hva linjekapasiteten skulle tilsi. Dette gjøres for å sikre stabiliteten i leveransen siden ulike leverandører ikke til enhver tid kan levere den oppgitte båndbredde for alle sine brukere. Vi antyder her at Internettbasert leveranse av video har mange kritiske ledd, siden man ikke kan kontrollere all infrastruktur i leveranseveiene, og at mange av tjenesteleverandørene opererer med tjenestegarantier som baserer seg på leveranse etter beste evne.

5.5.1.2 Båndbredde på klientsiden

Hvis man tar utgangspunkt i en 250Kbit/sekund kodet video vil ideell situasjon kreve båndbredde tilsvarende videoens båndbredde. Et lokalnett opererer med båndbredde på 10 eller 100 Mbit hos de fleste av Mercors aktuelle kunder. Avhengig hvordan dette er koblet kan man minimum kunne teoretisk kunne levere $10000/250 = 40$ samtidige UDP-unicastsendinger i et 10Mbit LAN hvor alle datapakker benytter samme linje fra en lokal tjener med 10Mbit tilkobling. I tillegg kommer båndbredde for eventuell feilkorreksjon, og retransmisjon av tapte pakker. Fra Microsoft anbefales en 80% utnyttelse av båndbredden ved normal drift for å unngå feilsituasjoner. En multicastsending på et lokalnett vil benytte 250Kbit/s båndbredde som deles mellom alle klientene. Et sentralt punkt for breddebehov på klientsiden er at dette er statisk lik videoens båndbredde. Andre nettverksleverte tjenester som for eksempel nettlesere forsøker å hente ned de nødvendige data med all tilgjengelig båndbredde seksjonsvis. For video er nødvendig båndbredde derfor enklere å beregne siden det hele tiden er snakk om en konstant verdi som tilsvarer videoens båndbreddeforbruk. Med den skisserte videokvaliteten behøver også en Internettilkoblet klient en reell båndbredde tilsvarende videoens virkelige båndbredde. Ved høy bruk av båndbredde vil dette naturlig nok kunne påvirke den andre trafikken på den respektive nettforbindelsen. Andre leveransemetoder som lokale filtjenere eller webtjenerleveranse gir noe mer usikkerhet rundt båndbreddebehovet. Nedlasting av denne typen foregår etter prinsippet at man forsøker å utnytte all tilgjengelig båndbredde. Ved flere samtidige nedlastinger av video kan dette gi uønsket høy bruk av båndbredde.

5.5.1.3 Båndbredde på serversiden

Tjener og proxy er i praksis like, derfor omtaler vi de som tjener under dette punktet.

Båndbreddebehov for streamingserveren er proporsjonalt avhengig av hvor mange klientstrømmer som må håndteres. Hvor mye båndbredde hver klient benytter er igjen avhengig av hvilken protokoll de bruker. Microsoft Media Server kan konfigureres til å begrense båndbredde og antall klienter som kan koble seg til ved hjelp av parametere som settes i serverbrukergrensesnittet. UDP trafikk benytter forenklet sett videobåndbredden ut fra tjeneren mens TCP trafikk vil generere vesentlig mer returtrafikk til tjeneren.

5.5.1.4 Båndbreddereservasjon og QoS

RSVP og pakkeprioritering med 802.1p protokollen er implementert i Windows 2000 Server. RSVP er omtalt i pkt. 4.1.6. 802.1p er en standard for prioritering av pakker i nettverkinfrastruktur.

5.6 Autentisering mot Microsoft Streaming Server

I Avbrekksystemet er det nødvendig med autentisering mot videotjeneren og proxyer for å kunne styre tilgang til videoer. Dette er spesielt nødvendig fordi deler av innholdet vil være tilgjengelig via Internett. Autentiseringen mot Microsoft Media Server kan foregå etter en metode kalt Basic Authentication, BA. BA er ukryptert autentisering med brukernavn og passord. Hvilke videoer brukeren kan streamere følger Access Control List funksjonen i Windows 2000, hvilket vil si at brukeren må ha lesetilgang til den aktuelle videoen. ACL listene kan være gitt ut i fra et NT-domene. Det vil si at alle som er meldt inn i NT-domenet på det lokale nettet, også får tilgang til videoene ved autentisering mot tjeneren.

Brukerinnlogging på en Windows 2000 Server krever at man har Terminal Client lisens. *Mer om softwarepriser i pkt. 7.* Ukryptert innlogging over Internett vil medføre betydelig sikkerhetsrisiko, og dårlig sikring av videofiler. Det er mulig å autentisere sikkert mot databaseapplikasjonen. Man kan tenke seg å utvikle en mer sikker innlogging ved å benytte Windows Media Services SDK. I denne SDKen finnes det C++ biblioteker som har grensesnitt for å kommunisere direkte med streamingtjeneren. Et annet alternativ er å benytte Microsofts Digital Rights Management, DRM. Kostnadene forbundet med dette er ikke kjent, men det er oppgitt at denne teknologien må lisensieres fra Microsoft.

5.7 Applikasjonsutvikling med Windows Media Player

Applikasjonen med Windows Media Player kan utvikles på flere plattformer da det finnes verktøy for å kommunisere gjennom mediaspillerens COM grensesnitt. Man kan nå de nødvendige parametere på .NET plattformen, Visual Basic eller C++ utvikling under Microsoft Visual Studio.

5.8 Klientkobling mot tjenerdatabase

Klientkobling mot database er avhengig av endelig utforming av brukergrensesnitt for streamingklienten, så noen endelig beskrivelse vi ikke følge her. Vi skisserer noen aktuelle scenario for kobling mellom klient og tjenerdatabase.

Scenario 1: Klienten gjør spørring mot databasen over lokalnett eller Internett og får returnert titler over alle filer som finnes tilgjengelig på tjeneren. Disse titlene sammenlignes med en lokal oversikt over aktuelle URLer, og den lokale oversikten oppdateres i forhold til nye tilgjengelige titler og metainformasjon for disse. Streaming av aktuelle filer foregår i forhold til den lokale oversikten. Man kan tenke seg at oppdateringen mot tjenerdatabasen foregår med ett døgn intervaller.

Scenario 2: Klienten gjør en spørring mot tjenerdatabasen i forhold til ønsket video, basert på metainformasjon og får returnert URLer til de aktuelle videoene. Klienten gjør et utvalg i de mottatte URLene på grunnlag av hvor ofte denne unike klienten har streamet de aktuelle videoene. Den minst sette videoen velges ut. Hvis to videoer har samme frekvens velges en av dem tilfeldig.

6. Alternativ streamingløsning – QuickTime6 MPEG4

QuickTime 6, heretter QT6, har tilsvarende arkitekturer som Microsoft Mediastreamingplattformen, heretter kalt MMSP. Siden plattformen skiller seg både i pris og teknologi ønsker vi å presentere noen av resultatene fra forskning i QuickTime-teknologien. Vi kommer inn på forskjellene mellom Microsoft Streaming og QuickTime/MPEG4. QuickTime codec'er og innpakkingsformater er allerede presentert i pkt. 3.2 og 3.3, så vi går ikke nærmere inn på spesifikke deler av formatet med hensyn til koding og innpakking. Verktøy for å lage QuickTime-filer er også presentert nærmere i pkt. 3.4.

Teknologiforskjellene mellom MMSP og QT6 ligger kort oppsummert i at MMSP benytter proprietære protokoller for leveranse av video, proprietære videokodere, og proprietære filformat for streamingvideo. QT6 benytter i motsetning åpne standarder for streaming med RTP/RTSP leveranseprotokoll, MPEG4 ISO filformat og MPEG4 ISO videocodec. Det gir interoperabilitet med andre ikke-QT6 produkter som for eksempel Java Media Framework og Real Player. Sistnevnte når denne får MPEG4 dekodere. Windows Media Player har ikke støtte for å streame video over RTSP/RTP, selv om den har mulighet for avspilling av ISO-MPEG4.

Darwin Streaming Server 4, DSS4, er Apples åpen kildekode streamingserver for streaming av MPEG4 over RTSP/RTP. Denne er identisk med QuickTime Streaming Server 4. DSS4 er tilgjengelig uten lisenskostnader. Den har en modularkitektur som gjør det mulig å laste inn autentiseringsmoduler og andre ferdige moduler. Man kan i tillegg skrive sine egne moduler som får tilgang til tjeneren med et modul grensesnitt. Darwin Streaming Server kan kjøres under Windows NT, Linux, Solaris, BSD, MacOSX m.fl.

QuickTime har programmeringsomgivelser for utvikling av tjenerapplikasjoner i C++ og Java på Windows plattform, og under OSX for Java-basert utviklingsplattform. QuickTime 6 er ikke sluppet som produkt fra Apple per i dag, men lansering er ventet i nær fremtid. QuickTime 5 har den samme funksjonaliteten som er beskrevet for QT6, med unntak av at det benyttes .mov filformat og Sorenson Codec 3 for videokoding.

7. Kostnadsoversikt

Priser er innhentet fra Inmeta og er beregnet uten merverdiavgift. Kostnadsoversikten gir softwarerelaterte kostnader, med utgangspunkt i et system med 50 brukere. Det er ikke tatt hensyn til andre utviklingskostnader. Det er noe usikkerhet omkring MPEG4 lisenskostnader for QuickTime 6 siden dette ikke er ferdig forhandlet mellom Apple og MPEG-LA. For Microsofttjeneren er det i den beskrevne løsningen forutsatt at man behøver autentisering mot tjeneren for å beskytte videoene, og knytte brukeren opp mot tjenerdatabasen. Denne funksjonaliteten er lisensbelagt med 1 brukerlisens per samtidige innlogginger per år. Hvis man velger å ikke benytte autentisering mot Microsofttjeneren, behøver man ikke denne lisensen. På et lokalnett kan dette være aktuelt. Hvis man velger denne løsningen gir det ikke mulighet for å knytte informasjon i databasen opp mot brukernavn. For Internettbasert forbindelse mot tjeneren vil brukernavn og passord utgjøre et minimum av beskyttelse for uautorisert tilgang, selv om det er en meget mangelfull løsning. Her anbefaler vi å vurdere andre løsninger, som for eksempel Microsofts DRM-løsning for beskyttelse av mediadata. DRM-løsningen er lisensbelagt, og er inkompatibel med Windows 95 og Windows NT ved benyttelse av WM8 codec. DRM er tilgjengelig for de nevnte plattformene med WM7 codec. QuickTime-løsningen medfører ingen ekstrakostnad for sikker autentisering over Internett.

<i>Produkt</i>	<i>QuickTime</i>	<i>Microsoft Media</i>
	Linux	Windows 2000 Server
Operativsystem	0,-	7085,-
50 klientpåloggingslisenser	0,-	284 NOK * 50 = 14200,- (Årlig kostnad)
Redistribusjon av klientsoftware	(QT 5 pris) 0,-	Avhengig av avtale med Microsoft
Database	PostgreSQL 0,-	PostgreSQL 0,-
Kostnad pr. klient 1. år	<u>0,-</u>	21285/50 = <u>425,-</u> Uten autentiseringslisens: 7085/50 = <u>141,-</u>

8. Oppsummering og konklusjon

For Mercors videre planlegging er det av avgjørende betydning å gjøre konkrete plattformvalg med hensyn på format og utviklingsplattform. Både Microsoft og Apple kan levere integrerte løsninger for leveranse av video for Mercors kundegruppe. Løsningene tilbyr på de fleste områder den samme funksjonaliteten for leveranse av video med en lokalnettbasert løsning. Plattformene skiller seg i midler tid på kostnad forbundet med etablering av streamingløsning, spesielt for Internettbasert streaming av

video. Apples løsning har lavere kostnad knyttet til etablering av streamingtjenerløsning, samt lavere kostnad knyttet til sikring av videodata gjennom autentisering mot tjeneren. QuickTime-løsningen har også bedre kompatibilitet for alle de aktuelle plattformene med hensyn til Microsofts DRM-løsning kontra Apples autentisering mot tjener. *Mer om DRM-kompatibilitet i pkt 7.* Det som taler til fordel for den Microsoft-baserte løsningen er Mercors allerede utbygde streamingklient som antatt kan modifiseres for å støtte streaming av video i tillegg til avspilling fra lokal fil slik som per i dag. For en lokalnettløsning er vår vurdering at en Windows Media basert løsning for streaming vil være tilstrekkelig. For en Internettbasert løsning vil den Microsoft-baserte løsningen gi langt høyere kostnad enn Apples QuickTime-løsning, og ikke kunne anbefales ut i fra forventet inntjening for Mercors salg av klientlisenser.

Teknisk sett er Microsofts og Apples løsninger ikke kompatible. QuickTimes videokodere har under testing å gi bedre visuell gjengivelse av kodet video med Sorenson Codec 3 enn den tilsvarende Windows Media 8 Codec. Windows Media Server har høyere krav til maskinvare enn Darwin Streaming Server på Linux, noe som vil medføre en ekstrakostnad hvis man veier antall mulige brukere på en tjener mot investeringskostnad for maskinvare. Microsoft har en spesiell løsning, mens Apples løsning gir mulighet for samvirke med andre løsninger som også støtter MPEG4 standarden.

9. Referanser

SDP – ASF og MMS bakover engineering: <http://sdp.ppona.com>

Microsoft - Windows Media Service Deployment Guide

Microsoft – Best Practices for Windows Media Technologies

Perceptual Models and Architectures for Video Coding Applications – PHD Thesis
Christian J. Van Den Branden Lambrecht - 1996

N3062 - December, 1999

MPEG-2 Video Elementary Stream Supplemental Information

Apple Computer Inc. – Broadcasting with QuickTime 5, 2001

Apple Computer Inc. – Inside QuickTime Movies

Apple Computer Inc. – Summary QuickTime for Java

Apple Computer Inc. – QuickTime File Format

Apple Computer Inc. – QuickTime Streaming Server Modules

Apple Computer Inc. – QuickTime Streaming

Apple Computer Inc. – QuickTime 5, the media streaming standard

Pressenotat om QuickTime 6 fra Apple Computer Inc.
<http://www.apple.com/pr/library/2002/feb/12qt6.html>

Overview of the MPEG-4 Standard

INTERNATIONAL ORGANISATION FOR STANDARDISATION

ORGANISATION INTERNATIONALE DE NORMALISATION

ISO/IEC JTC1/SC29/WG11

CODING OF MOVING PICTURES AND AUDIO

<http://mpeg.telecomitalia.com/standards/mpeg-4/mpeg-4.htm>

Synthetic and SNHC Audio in MPEG-4

Eric D. Scheirer

http://leonardo.telecomitalia.com/icjfiles/mpeg-4_si/10-SNHC_audio_paper/10-SNHC_audio_paper.htm

RFC 1889 – RTP Real Time Transport Protocol

RFC 2326 – RTSP Real Time Streaming Protocol

RFC 2327 – SDP Session Description Protocol

Redistribusjon av Microsoft Windows Media Services SDK:

http://msdn.microsoft.com/workshop/imedia/windowsmedia/sdk/lic_redist.asp

Microsoft Digital Rights Management:

<http://www.microsoft.com/windows/windowsmedia/drm.asp>

Windows 2000 Server Internetworking Guide

Windows 2000 Server TCP/IP Core Networking Guide

MPEG Video Compression Standard – ISBN 0-412-08771-5

Joan L. Mitchell

William B. Pennebaker

Chad E. Fogg

Didier J. LeGall