

# Direkte programoverføring av mediafiler via trådløse nettverk

NRK som kontribusjonsaktør via Trådløse Trondheim

**Erik Berg**

Master i kommunikasjonsteknologi

Oppgaven levert: Juni 2008

Hovedveileder: Leif Arne Rønningen, ITEM

Biveileder(e): Thomas Jelle, Trådløse Trondheim  
Kjetil Flatøy, NRK



# Oppgavetekst

Oppgaven har fokus på å bidra og potensielt distribuere mediafiler over et trådløst nettverk og se på muligheter med tanke på trådløs overføring av tekst, bilde, lyd og video. En av utfordringene NRK har i forhold til radio- og TV-intervjuer er direkte overføring i sanntid uten merkbare forsinkelser. Oppgaven omfatter både et teorigrundlag og testing i samarbeid med Trådløse Trondheim og NRK.

Oppgaven gitt: 15. januar 2008

Hovedveileder: Leif Arne Rønningen, ITEM



# Sammendrag

Ulike regimer for komprimering og koding av media har hatt stor utvikling i den seneste tid, dette i samband med Trådløse Trondheim A/S sitt initiativ om å dekke bykjernen i Trondheim med trådløst nett har medført nye muligheter for ulike multimediatjenester. I denne oppgaven vil det bli sett på muligheter for NRK til å bidra med innhold (det vil si sende råmateriale fra en live event til produksjon) over Trådløse Trondheim sin infrastruktur. Rapporten er todelt og inneholder et teoristudie samt beskrivelse av praktisk testing. Teoridelen tar for seg ulike aspekter som komprimering og koding av lyd og video ved hjelp av henholdsvis Advanced Audio Coding (AAC) for lyd og Advanced Video Coding (H.264/AVC) for video. Andre kodeker som ITU G.711 og G.729 for lyd MPEG-2 for video ble også vurdert, men en visjon om et framtidrettet system gjorde valgene av AAC og AVC åpenbare.

Det har blitt forsket på ulike testmetoder som simulert trafikk ved hjelp av IxChariot<sup>®</sup> og manipulering av streamingserveren Darwin Streaming Server. Det ble tilslutt besluttet å bruke WireShark, en applikasjon for nettverksdiagnostisering, til å fange opp pakker fra et gitt grensesnitt. Pakkene fanget av WireShark dannet rådata som grunnlag for utregning av ulike metrikker som forsinkelse, bitrate, pakketap og variasjon i forsinkelse. Som uttalt i problemstillingen er forsinkelse det største problemet for direkte programoverføring. Det viste seg at forsinkelsen for innhold sendt via Trådløse Trondheim holder seg markant under kravet om 1s tur/retur forsinkelse. Problemet kom i form av pakketap, dette hadde ødeleggende effekter på gjennomføring av streaming av video. Testene fra streaming av lyd derimot viste gode resultater, og det blir anbefalt Trådløse Trondheim og NRK å inngå et samarbeid angående kontribusjon av lyd.



# Forord

Denne masteroppgaven er skrevet av student Erik Berg som en del av Master programmet for kommunikasjonsteknologi ved Norges Teknisk- Naturvitenskaplige Universitet i Trondheim. Arbeidet ble utført våren 2008 i samarbeid med Professor Leif Arne Rønningen ved Institutt for Telematikk (NTNU), Thomas Jelle fra Trådløse Trondheim A/S (TrT) og Ketil Flatøy (NRK).

Jeg vil rette en stor takk til Leif Arne Rønningen, Thomas Jelle og Kjetil Flatøy for verdifull hjelp og støtte under hele prosessen.

Valgt testmetodikk krevdde mediafiler av ulike formater. Disse ble transkodet og produsert med hjelp fra Erlend Heiberg ansatt under Professor Leif Arne Rønningen ved Caruso-labben NTNU. Takk for hjelpa Erlend!

Jeg vil også takke mine kontorkoleger Håkon Rørvik Aune og Svein-Tore Landsverk for trivelig samvær og ikke minst teknisk input og sosial støtte gjennom våren. De har bidratt til å øke motivasjon for oppgaven samt kommet med konstruktiv kritikk og gode innspill.

For eventuelle kommentarer og spørsmål vedrørende oppgaven henvises leseren til mail-korrespondanse via [bareerik@gmail.com](mailto:bareerik@gmail.com)

Trondheim, juni 2008

Erik Berg





# Innhold

<b>Sammendrag</b>	<b>i</b>
<b>Forord</b>	<b>iii</b>
<b>Innholdsfortegnelse</b>	<b>ix</b>
<b>Figurer</b>	<b>xiii</b>
<b>Tabeller</b>	<b>xvi</b>
<b>Ordliste</b>	<b>xix</b>
<b>1 Innledning</b>	<b>1</b>
1.1 Bakgrunn . . . . .	1
1.2 Problemstilling . . . . .	1
1.3 Avgrensninger . . . . .	2
1.4 Relaterte arbeider . . . . .	2
1.5 Leserveiledning . . . . .	2
<b>I Teori</b>	<b>3</b>
<b>2 Trådløse Trondheim (TrT)</b>	<b>5</b>
2.1 Teknologier . . . . .	6
2.1.1 Aksessnett . . . . .	6
2.1.2 Matenett . . . . .	6
2.2 Infrastruktur . . . . .	6
2.3 Wi-Fi . . . . .	7
<b>3 Norsk Rikskringasting (NRK)</b>	<b>9</b>
3.1 Krav til Strømmer . . . . .	9
3.2 Kontribusjonsteknologier . . . . .	10
3.2.1 AudioTx Communicator . . . . .	10

3.2.2	Stratalink . . . . .	10
3.3	Scenarier . . . . .	11
<b>4</b>	<b>Kontribusjon over IP-nett</b>	<b>13</b>
4.1	Kontribusjon av video over IP . . . . .	13
4.1.1	Introduksjon . . . . .	13
4.1.2	Protokoller for overføring . . . . .	13
4.1.3	Pakketap . . . . .	14
4.1.4	Variasjon av forsinkelse . . . . .	14
4.1.5	Båndbredde . . . . .	15
4.2	Kontribusjon av lyd over IP . . . . .	15
4.2.1	Introduksjon . . . . .	15
4.2.2	Protokoller for overføring . . . . .	15
4.2.3	Pakketap . . . . .	16
4.2.4	Variasjon av forsinkelse . . . . .	16
4.3	Protokoller for overføring . . . . .	16
4.3.1	RTP og RTCP . . . . .	16
4.3.2	RTSP . . . . .	17
<b>5</b>	<b>Videokodeker</b>	<b>19</b>
5.1	MPEG-2 . . . . .	19
5.2	H.264/AVC (MPEG-4 del 10) . . . . .	20
<b>6</b>	<b>Lydkodeker</b>	<b>23</b>
6.1	ITU G.711 . . . . .	23
6.2	ITU G.729 . . . . .	24
6.2.1	G.729 A . . . . .	24
6.2.2	G.729 B . . . . .	24
6.3	Advanced Audio Coding (MPEG-4 AAC) . . . . .	25
6.3.1	MPEG-4 AAC-LC . . . . .	25
6.3.2	MPEG-4 HE-AAC . . . . .	26
6.3.3	MPEG-4 AAC-LD . . . . .	26
6.3.4	MPEG-4 AAC Scalable . . . . .	26
6.3.5	MPEG-4 AAC Scalable Lossless Coding (SLS) . . . . .	26
<b>7</b>	<b>Streaming</b>	<b>29</b>
7.1	Introduksjon . . . . .	29
7.1.1	Streaming . . . . .	29
7.1.2	Darwin Streaming Server . . . . .	30
7.1.3	Hinting av media . . . . .	31

7.2	QuickTime™suite . . . . .	31
7.2.1	QuickTime™Player . . . . .	31
7.2.2	QuickTime™Pro . . . . .	31
7.2.3	QuickTime™Broadcaster . . . . .	31
<b>8</b>	<b>Testing</b>	<b>33</b>
8.1	Metrikker for testing . . . . .	33
8.1.1	Bitrate . . . . .	33
8.1.2	Forsinkelse . . . . .	34
8.1.3	Variasjon av forsinkelse . . . . .	34
8.1.4	Pakketap . . . . .	35
8.2	Programvare brukt til testing . . . . .	36
8.2.1	Darwin Streaming Server . . . . .	36
8.2.2	WireShark . . . . .	37
<b>II</b>	<b>Testimplementasjon</b>	<b>39</b>
<b>9</b>	<b>Fremgangsmåte</b>	<b>41</b>
9.1	Programvareinstalasjon . . . . .	42
9.2	Klargjøring av media . . . . .	43
9.3	Testing . . . . .	43
9.3.1	Server . . . . .	44
9.3.2	Klient . . . . .	44
9.4	Uthenting av data . . . . .	46
9.4.1	Server . . . . .	46
9.4.2	Klient . . . . .	46
<b>10</b>	<b>Spesifikasjoner</b>	<b>47</b>
10.1	Oppsett . . . . .	47
10.2	Testlokasjoner . . . . .	48
10.3	Programvare . . . . .	49
10.3.1	Darwin Streaming Server . . . . .	49
10.3.2	WireShark . . . . .	49
10.3.3	VLC-Player . . . . .	50
10.3.4	MP4box . . . . .	50
10.4	Maskinvare . . . . .	50
10.4.1	Server . . . . .	50
10.4.2	Klient . . . . .	51

<b>III</b>	<b>Evaluering</b>	<b>53</b>
<b>11</b>	<b>Resultater</b>	<b>55</b>
11.1	Laststudie . . . . .	56
11.2	Lyd . . . . .	57
11.2.1	Målinger . . . . .	57
11.2.2	Oppsummering . . . . .	60
11.3	Video . . . . .	61
11.3.1	Målinger . . . . .	61
11.3.2	Oppsummering . . . . .	63
<b>12</b>	<b>Diskusjon</b>	<b>65</b>
12.1	Bakgrunn for valg . . . . .	65
12.1.1	Programvare . . . . .	65
12.1.2	Testmetodikk . . . . .	65
12.1.3	Kodeker . . . . .	67
12.2	Resultater . . . . .	68
12.2.1	Laststudie 11.1 . . . . .	68
12.2.2	Lyd 11.2 og Video 11.3 . . . . .	69
12.3	Mulige løsninger . . . . .	70
12.3.1	FEC og Error Concealment . . . . .	70
12.3.2	QualityShaping . . . . .	70
12.3.3	Tilbudt båndbredde . . . . .	70
12.3.4	Garantier for tjenestekvalitet . . . . .	71
12.4	Mulig kontribusjonsløsning . . . . .	71
<b>13</b>	<b>Konklusjon</b>	<b>73</b>
13.1	Generelt . . . . .	73
13.2	Resultater mot krav fra NRK . . . . .	74
13.3	Anbefalinger . . . . .	75
13.4	Videre arbeid . . . . .	75
	<b>Bibliografi</b>	<b>81</b>
	<b>Vedlegg</b>	<b>A-1</b>
<b>A</b>	<b>Protokoll-stakker</b>	<b>A-1</b>
<b>B</b>	<b>Kildekode</b>	<b>B-1</b>

<b>C</b>	<b>Atheros AR5005G</b>	<b>C-1</b>
<b>D</b>	<b>Mediakarakteristiker</b>	<b>D-1</b>
<b>E</b>	<b>Grafer</b>	<b>E-1</b>
E.1	Kontoret . . . . .	E-2
E.1.1	Video . . . . .	E-2
E.1.2	Lyd . . . . .	E-8
E.2	Torget . . . . .	E-14
E.2.1	Video . . . . .	E-14
E.2.2	Lyd . . . . .	E-20
E.3	Solsiden . . . . .	E-26
E.3.1	Video . . . . .	E-26
E.3.2	Lyd . . . . .	E-30
<b>F</b>	<b>Statistikk</b>	<b>F-1</b>



# Figurer

3.1	Gjeldende praksis for trådløs kringkasting i real-time. . . . .	11
4.1	Eksempel på kontribusjon. . . . .	14
4.2	RTP-data i en IP-pakke. . . . .	17
5.1	Struktur for H.264/AVC videoenkoder. . . . .	21
7.1	Enkelt oppsett for streamingsserver. . . . .	30
8.1	Grafisk grensesnitt for Darwin Steaming Server. . . . .	37
8.2	Grafisk grensesnitt for Wire Shark. . . . .	38
9.1	Skille mellom klient og server for oppsett av programvare. . . . .	41
9.2	Eksempel på tillegging av media i MediaCoder. . . . .	43
9.3	Eksempel på setting av media-parametre i MediaCoder . . . . .	43
9.4	Eksempel på hinting med MP4box. . . . .	44
9.5	Capture-meny WireShark. . . . .	45
9.6	VLC-Player meny for streaming. . . . .	45
10.1	Testoppsett . . . . .	47
10.2	Dekningskart for TrT med angitte testelokasjoner. . . . .	48
11.1	Forsinkelse ved overlast. . . . .	56
11.2	Bitrater for lyd sendt fra Kontoret. . . . .	57
11.3	Bitrater for lyd sendt fra Solsiden. . . . .	58
11.4	Forsinkelse ved sending av 152kbit/s lydstrøm fra Solsiden. . . . .	59
11.5	Pakketap ved sending av 152kbit/s lydstrøm fra Solsiden. . . . .	59
11.6	Bitrater for sending av video fra Solsiden. . . . .	61
11.7	Forsinkelse ved sending av 1mbit/s videostrøm fra Solsiden. . . . .	62
11.8	Pakketap ved sending av 1mbit/s videostrøm fra Solsiden. . . . .	62
11.9	Visuell sammenligning av videokvalitet. . . . .	64
12.1	Mulig framtidig oppsett for kontribusjon via TrT. . . . .	72

A.1	Oversikt over protokollstakker. . . . .	A-2
E.1	Resulterende bitrater ved sending av video fra Kontoret. . . . .	E-2
E.2	Forsinkelse ved sending av 3mbit/s videostrøm fra Kontoret. . . . .	E-3
E.3	Variasjon av forsinkelse ved sending av 3mbit/s videostrøm fra Kontoret. . . . .	E-3
E.4	Pakketap ved sending av 3mbit/s videostrøm fra Kontoret. . . . .	E-4
E.5	Forsinkelse ved sending av 2mbit/s videostrøm fra Kontoret. . . . .	E-4
E.6	Variasjon av forsinkelse ved sending av 2mbit/s videostrøm fra Kontoret. . . . .	E-5
E.7	Pakketap ved sending av 2mbit/s videostrøm fra Kontoret. . . . .	E-5
E.8	Forsinkelse ved sending av 1mbit/s videostrøm fra Kontoret. . . . .	E-6
E.9	Variasjon av forsinkelse ved sending av 1mbit/s videostrøm fra Kontoret. . . . .	E-6
E.10	Pakketap ved sending av 1mbit/s videostrøm fra Kontoret. . . . .	E-7
E.11	Resulterende bitrater ved sending av lyd fra Kontoret. . . . .	E-8
E.12	Forsinkelse ved sending av 283kbit/s lydstrøm fra Kontoret. . . . .	E-9
E.13	Variasjon av forsinkelse ved sending av 283kbit/s lydstrøm fra Kontoret. . . . .	E-9
E.14	Pakketap ved sending av 283kbit/s lydstrøm fra Kontoret. . . . .	E-10
E.15	Forsinkelse ved sending av 152kbit/s lydstrøm fra Kontoret. . . . .	E-10
E.16	Variasjon av forsinkelse ved sending av 152kbit/s lydstrøm fra Kontoret. . . . .	E-11
E.17	Pakketap ved sending av 152kbit/s lydstrøm fra Kontoret. . . . .	E-11
E.18	Forsinkelse ved sending av 59kbit/s lydstrøm fra Kontoret. . . . .	E-12
E.19	Variasjon av forsinkelse ved sending av 59kbit/s lydstrøm fra Kontoret. . . . .	E-12
E.20	Pakketap ved sending av 59kbit/s lydstrøm fra Kontoret. . . . .	E-13
E.21	Resulterende bitrater ved sending av video fra Torget . . . . .	E-14
E.22	Forsinkelse ved sending av 3mbit/s videostrøm fra Torget. . . . .	E-15
E.23	Variasjon av forsinkelse ved sending av 3mbit/s videostrøm fra Torget. . . . .	E-15
E.24	Pakketap ved sending av 3mbit/s videostrøm fra Torget. . . . .	E-16
E.25	Forsinkelse ved sending av 2mbit/s videostrøm fra Torget. . . . .	E-16
E.26	Variasjon av forsinkelse ved sending av 2mbit/s videostrøm fra Torget. . . . .	E-17
E.27	Pakketap ved sending av 2mbit/s videostrøm fra Torget. . . . .	E-17
E.28	Forsinkelse ved sending av 1mbit/s videostrøm fra Torget. . . . .	E-18
E.29	Variasjon av forsinkelse ved sending av 1mbit/s videostrøm fra Torget. . . . .	E-18
E.30	Pakketap ved sending av 1mbit/s videostrøm fra Torget. . . . .	E-19
E.31	Resulterende bitrater ved sending av lyd fra Torget . . . . .	E-20
E.32	Forsinkelse ved sending av 283kbit/s lydstrøm fra Torget. . . . .	E-21
E.33	Variasjon av forsinkelse ved sending av 283kbit/s lydstrøm fra Torget. . . . .	E-21
E.34	Pakketap ved sending av 283kbit/s lydstrøm fra Torget. . . . .	E-22
E.35	Forsinkelse ved sending av lydstrøm fra Torget. . . . .	E-22
E.36	Variasjon av forsinkelse ved sending av lydstrøm fra Torget. . . . .	E-23
E.37	Pakketap ved sending av lydstrøm fra Torget. . . . .	E-23
E.38	Forsinkelse ved sending av 59kbit/s lydstrøm fra Torget. . . . .	E-24



E.39	Variasjon av forsinkelse ved sending av 59kbit/s lydstrøm fra Torget. . . .	E-24
E.40	Pakketap ved sending av 59kbit/s lydstrøm fra Torget. . . . .	E-25
E.41	Resulterende bitrater ved sending av video fra Solsiden . . . . .	E-26
E.42	Forsinkelse ved sending av 3mbit/s videostrøm fra Solsiden. . . . .	E-27
E.43	Variasjon av forsinkelse ved sending av 3mbit/s videostrøm fra Solsiden. .	E-27
E.44	Pakketap ved sending av 3mbit/s videostrøm fra Solsiden. . . . .	E-28
E.45	Forsinkelse ved sending av 1mbit/s videostrøm fra Solsiden. . . . .	E-28
E.46	Variasjon av forsinkelse ved sending av 1mbit/s videostrøm fra Solsiden. .	E-29
E.47	Pakketap ved sending av 1mbit/s videostrøm fra Solsiden. . . . .	E-29
E.48	Resulterende bitrater ved sending av lyd fra Solsiden . . . . .	E-30
E.49	Forsinkelse ved sending av 283kbit/s lydstrøm fra Solsiden. . . . .	E-31
E.50	Variasjon av forsinkelse ved sending av 283kbit/s lydstrøm fra Solsiden. .	E-31
E.51	Pakketap ved sending av 283kbit/s lydstrøm fra Solsiden. . . . .	E-32
E.52	Forsinkelse ved sending av 152kbit/s lydstrøm fra Solsiden. . . . .	E-32
E.53	Variasjon av forsinkelse ved sending av 152kbit/s lydstrøm fra Solsiden. .	E-33
E.54	Pakketap ved sending av 152kbit/s lydstrøm fra Solsiden. . . . .	E-33
E.55	Forsinkelse ved sending av 59kbit/s fra Solsiden. . . . .	E-34
E.56	Variasjon av forsinkelse ved sending av 59kbit/s lydstrøm fra Solsiden. . .	E-34
E.57	Pakketap ved sending av 59kbit/s lydstrøm fra Solsiden. . . . .	E-35



# Tabeller

2.1	Komponenter i Trådløse Trondheim [50]	7
2.2	Sammenligning av 802.11a og b/g [9]	7
3.1	Krav til lyd-strømmer NRK.	9
3.2	Krav til video-strømmer NRK.	10
4.1	Samplingsfrekvenser for RTP-klokker [45].	17
5.1	Bruks-scenarier i H.264	22
6.1	Mulige filendelser AAC	25
6.2	MPEG-4-profiler og deres tilhørende typer av audio-objekter [10]	26
6.3	AAC samplingsfrekvenser.	27
10.1	Oversikt over testlokasjoner.	49
10.2	Darwin Streaming Server Build-spesifikasjon	49
10.3	WireShark Build-spesifikasjon	49
10.4	Spesifikasjon for VLC-Player	50
10.5	MP4box-spesifikasjon.	50
10.6	Spesifikasjon for Fujitsu Siemens AMILO Pro V3515	50
10.7	Spesifikasjon for Dell OPTIPLEX GX280	51
11.1	Samling av statistikk for for sending av lyd.	60
11.2	Samling av statistikk for for sending av video.	63
12.1	Samling av statistikk for alle målinger	69
12.2	Protokoller som tilbyr tjenestekvalitet (QoS)	71
12.3	Oversikt over metoder for å styre trafikk i nettverk (QoS).	71
13.1	Oppfyllelse av NRK sine krav til lyd-strømmer.	74
13.2	Oppfyllelse av NRK sine krav til video-strømmer.	75
A.1	Nivåer i H.264	A-4

C.1	Spesifikasjon Atheros AR5005G [5]	C-2
D.1	Media-karakteristiker H.264 1mbit/s.	D-2
D.2	Media-karakteristiker H.264 2mbit/s.	D-3
D.3	Media-karakteristiker H.264 3mbit/s.	D-4
D.4	Media-karakteristiker AAC 283kbit/s.	D-5
D.5	Media-karakteristiker AAC 159kbit/s.	D-6
D.6	Media-karakteristiker AAC 59kbit/s.	D-7
F.1	Samling av statistikk for alle målinger	F-2

# Ordliste

<i>AAC</i>	Advanced Audio Coding
<i>AAC – LC</i>	Low Complexity AAC
<i>AAC – LD</i>	Low Delay AAC
<i>AAC – SLS</i>	AAC Scalable Lossless Coding
<i>AVC</i>	Advanced Video Coding
<i>CNG</i>	Comfort Noise Generation
<i>CS – ACELP</i>	Conjugate-Structure Algebraic-Code-Excited Linear Prediction
<i>CSV</i>	Comma-Separated Values
<i>DSS</i>	Darwin Streaming Server
<i>DTX</i>	Discontinuous Transmission
<i>EBU</i>	European Broadcasting Union
<i>ER</i>	Error Recilient
<i>HCR</i>	Huffman Codeword Reordering
<i>HD</i>	High Definition
<i>HE – AAC</i>	High Efficiency AAC
<i>IPdv</i>	Instantaneous Packet Delay Variation
<i>ITU – T</i>	Telecommunication Standardization Sector
<i>LAN</i>	Local-Area Network
<i>LWAPP</i>	Lightweight Access Point Protocol
<i>MIME</i>	Multipurpose Internet Mail Extensions

<i>MMS</i>	Multimedia Messaging Service
<i>MP4</i>	MPEG-4 file format
<i>MPEG</i>	Moving Picture Experts Group
<i>NRK</i>	Norsk Rikskringkasting
<i>NTSC</i>	National Television System(s) Committee
<i>PAL</i>	Phase-alternating line
<i>PCM</i>	Pulse-Code Modulation
<i>PSS</i>	Packet-switched Streaming Service
<i>QAM</i>	Quadrature Amplitude Modulation
<i>QoS</i>	Quality of Service
<i>QPSK</i>	Quadrature Phase Shift Keying
<i>QTSS</i>	QuickTime Streaming Server
<i>RED</i>	Random Early Detection
<i>RFC</i>	Request For Comment
<i>RP – TNC</i>	Reverse-Polarity Threaded Neill-Concelman connector
<i>RSVP</i>	Resource Reservation Protocol
<i>RSVP – TE</i>	RSVP-Traffic Engineering
<i>RTP</i>	Real-time Transport Protocol
<i>RTSP</i>	Real Time Streaming Protocol
<i>RVLC</i>	Reversible Variable Length Code
<i>SBR</i>	Spectral Band Replication
<i>SD</i>	Standard Definition
<i>SDP</i>	Session Description Protocol
<i>SRTP</i>	Secure Real-time Transport Protocol
<i>TNC</i>	Threaded Neill-Concelman connector

<i>TOS</i>	Type of Service
<i>TP</i>	Twisted Pair
<i>TrT</i>	Trådløse Trondheim
<i>VAD</i>	Voice Activity Detection
<i>VCB11</i>	Virtual Codebooks
<i>VCIP</i>	Video Contribution over IP
<i>VHS</i>	Video Home System
<i>WCS</i>	Wireless Communications Service
<i>WDS</i>	Wireless Distribution System
<i>WLAN</i>	Wireless Local-Area Network
<i>WRED</i>	Weighted Random Early Detection





# 1

## Innledning

### 1.1 Bakgrunn

Komprimering av multimedieinnhold har hatt stort fokus i den siste tid. Dette har resultert i nye standarder som dramatisk reduserer behovet for båndbredde ved overføring og streaming av multimedieinnhold. I sammenheng med at flere samarbeidspartnere besluttet å dekke Trondheim med trådløse nett via sitt initiativ Trådløse Trondheim, har det blitt åpnet for mange nye muligheter innen både kringkasting og servering av tjenester. En potensiell tjenesteleverandør i så måte er NRK. Denne rapporten vil utforske muligheten for at nettopp NRK kan gjøre seg nytte av Trådløse Trondheim som transportnettverk for å bidra med innhold fra ulike lokasjoner innenfor det gitte dekningsområdet. Hvis testene lykkes kan en konsekvens være at også andre kringkastingsaktører ser dette som en positiv trend og vil benytte seg av kontribusjon over IP.

Spesielt for denne rapporten vil det bli forsket på et utvalg kodeker/komprimeringsregimer som kan egne seg til kontribusjon. Det vil så gjøres et selektivt utvalg blant disse slik at testingen konsentreres om så få kodeker som mulig og deretter observere ulike metrikker for å se hvordan disse oppfører seg i Trådløse Trondheim sin infrastruktur.

### 1.2 Problemstilling

Oppgaven har fokus på å bidra og potensielt distribuere mediafiler over et trådløst nettverk og se på muligheter med tanke på trådløs overføring av tekst, bilde, lyd og video.

En av utfordringene NRK har i forhold til radio- og TV-intervjuer er direkte overføring i sanntid uten merkbare forsinkelser. Oppgaven omfatter både et teoretisk studie og testing i samarbeid med Trådløse Trondheim og NRK.

### 1.3 Avgrensninger

Problemstillingen inkluderer en studie av overføring av tekst, bilde, lyd og video. Kandidaten vurderer overføring av tekst og bilde som trivielt da dette kan sammenlignes med sending av epost. Disse tema vil derfor ikke behandles i denne rapporten. For å forske på tema som er mer framtidsrettet og per nå kanskje ikke er gjennomførbare har det blitt valgt å fokusere på testing med tanke på overføring av lyd og video.

### 1.4 Relaterte arbeider

En masteroppgave skrevet av Petter Stray våren 2007 [48] har tester angående Voice over IP blitt utført. Stray sin oppgave kan sammenlignes med denne oppgavens del for lydtesting. Ulikhetene mellom oppgaven ligger i at denne oppgaven fokuserer på en-til-en kommunikasjon mellom to noder, mens Stray sin oppgave beskriver skalerbarhet ved mange en-til-en samtaler, med andre ord Trådløse Trondheim sin kapasitet.

### 1.5 Leserveiledning

Denne rapporten er delt inn i 4 deler: Innledning, teori, testimplementasjon og evaluering. Innledningen beskriver oppgaven og dens begrensninger. Teoridelen legger grunnlaget for den praktiske delen av oppgaven. Her blir brukte teknologier og kodeker beskrevet. Delen for testimplementasjon gir en innføring i hvordan testene har blitt satt opp og utført, og den gir en oversikt over hvilke programvare- og maskinvare-versjoner som har blitt brukt. Denne delen kan brukes for eventuelle etterfølgere å kunne replikere de utførte testene. Siste del er evaluering. I denne delen angis resultater fra testingen, deretter blir nødvendige tema fra rapporten og det praktiske arbeidet diskutert før det blir dratt konklusjoner av prosessen. Vedleggene til denne rapporten fungerer som en kilde til dypere innsikt i hva som har blitt gjort og inkluderer mer detaljrike innslag som av hensiktsmessige årsaker har blitt utelatt i selve rapporten. Dette for å unngå unødig repetisjon av momenter. Vedlagt DVD inneholder filer og innhold som enten ikke lar seg representere i analog bokform eller ville tatt opp for mye plass. Disse inkluderer en uttømmende liste av *grafer*, kopier av alle *kilder*, produksjonsfiler for dette dokument i form av  $\text{\LaTeX}$ -filer, en eksportert *MySQL-dump* med verdier fra alle målinger, diverse *måleresultat* ekstrahert fra de ulike testene og *PHP-skript* som har produsert grafer og statistikk.

Del I  
**Teori**



# 2

## Trådløse Trondheim (TrT)

30. november 2005 inngikk Trondheim Kommune, Sør-Trøndelag Fylkeskommune og Næringsforeningen i Trondheim en samarbeidsavtale med NTNU hvor partene forpliktet seg til å arbeide sammen for å lage et trådløst utendørs bredbåndsnett i Trondheim. Den opprinnelige samarbeidsgruppen har i den siste tid biltt utvidet til også å omfatte SpareBank 1 Midt-Norge, Trondheim Energi og Adresseavisen [35]. Det er nå opprettet et aksjeselskap for å innfri de initielle målsetningene:

- Å skape et utviklingslaboratorium i verdensklasse for forskning og utvikling innen trådløse teknologier, produkter og tjenester.
- Bidra til et fremtidsrettet norsk næringsliv med spisskompetanse innen trådløse teknologier, -produkter og -tjenester.
- Gjøre Trondheim og NTNU mer attraktivt for studenter og teknologibasert næringsliv.

I oppstartsfasen ble målgruppen for tjenesten begrenset til FoU-miljøet og storbrukere. Storbrukere er definert som studenter, ansatte ved NTNU, samt ansatte i Trondheim kommune og fylkeskommune. Begrensningen ble innført for i første omgang å teste ut om systemet skalerer, bygge en bæredyktig driftsstruktur og for å eliminere eventuelle oppstartsfeil/-mangler [26]. Per i dag har både tjenesteleverandører og privatpersoner fått tilgang. Dette skjedde allerede i september 2007. Forretningsmodellen tilsier at hver private bruker kan tilegne seg 12 timers aksess for NOK 10,-.

Per i dag dekker TrT områdene i Midt-byen samt strekningen mellom Midt-byen og NTNU Gløshaugen. Totalt dekkes et  $1.5km^2$  stort område. Dekningsområdet utvider seg kontinuerlig. For en oppdatert oversikt henvises det til Trådløse Trondheim sine nettsider [35].

Det følgende vil gi en kort oversikt over teknologier og virkemåte for TrT.

## 2.1 Teknologier

Trådløse Trondheim benytter seg av en to-delt teknologistruktur. Denne baserer seg på modellen med aksess- og matenett der kunden kobler seg opp mot et aksessnett og matenettet forsyner aksessnettet med kapasitet.

### 2.1.1 Aksessnett

De fleste potensielle kunder har allerede støtte for 802.11b/g, som opererer på 2.4 GHz-båndet, innebygd i sine terminaler. Derfor har TrT valgt å bygge ut aksesspunkter som støtter disse teknologiene. Kundens investeringskostnader er små og dermed terskel for å ta i bruk tjenesten i så måte liten eller ikke-eksisterende [26]. I tillegg støtter flere av aksesspunktene kommunikasjon over 802.11a. Denne teknologien benytter seg av 5 GHz-båndet og er enda ikke like utbredt som variantene b/g. Dette innebærer mindre støy og er derfor en stor fordel for å sikre tjenestekvalitet(QoS) [26].

### 2.1.2 Matenett

For å forsyne aksessnettet med kapasitet har fiber-optiske kabler blitt lagt til mange bygninger. Også univertitetets eksisterende fibernettverk er tatt i bruk. Siden det å gi hvert aksesspunkt egen fiber-tilgang kan være kostbart har Wireless Distribution System (WDS) eller MESH-topologi blitt innført. WDS/MESH tillater en sammenkobling av trådløse aksesspunkter slik at de kan utnytte samme tilgangspunkt til matenettet. Hvert aksesspunkt har derfor en 5 GHz radio for mating og en 2.4 GHz radio for brukeraksess [26].

## 2.2 Infrastruktur

Trådløse Trondheim er bygd opp av komponenter hovedsaklig fra Cisco Systems, Inc. De ulike komponentene er gjengitt i tabell 2.1. Nettverket inneholder "lettvekts" aksesspunkter av serien Aironet<sup>®</sup>1010, 1030, 1242, 1131, 1252 for lokal dekning, WLAN-kontroller i form av Cisco Location Appliance og et trådløst kontrollsystem (WCS) som forenkler nettverksadministrasjon [50]. For kontroll av aksesspunktene blir to Cisco 4404 WLAN kontrollere brukt. Hver av disse kan støtte minst 100 aksesspunkter [17].

Wireless LAN Controller 4404	Wireless Control System (WCS)
Cisco 1010,1242,1131,1252 AP	Cisco 1030 AP (med eksterne antenner)
5GHz sektorantenner (20° and 60°)	2.4GHz sektorantenner (70°, 120° and 150°)
Trådløst lokaliseringsverktøy	Antennekabler RP-TNC/N-male and N-female
Klimabokser	Hengelåser til klimaboksene

Tabell 2.1: Komponenter i Trådløse Trondheim [50]

Trafikken mellom aksesspunktene og kontrollsystemet blir tunnelert ved hjelp av LWAPP<sup>1</sup>. Denne protokollen gjør at aksesspunkter automatisk kan finne beste tilgjengelige trådløse kontroller uten manuell konfigurering [16].

Cisco Wireless LAN Controllers blir brukt til å kontrollere trådløse LAN-funksjoner i hele nettverket. Blant annet tar den seg av implementeringen av sentralstyrte sikkerhetsrutiner, forhindrer inntrengningsforsøk, administrerer radio-frekvenser, QoS og mobilitet. Systemet opererer i samarbeid med tidligere nevnte WCS og Aironet<sup>®</sup>-aksesspunkter for å understøtte kritiske trådløse applikasjoner [18].

## 2.3 Wi-Fi

De tre nettverksteknologiene 802.11 a og b/g som brukes i Trådløse Trondheim faller alle under Wi-Fi-kategorien. Selv om alle tilhører samme kategori er det forskjeller i bruk av frekvensbånd, maksimale bit-rater og rekkevidder[9]. En oversiktlig sammenligning gis i tabell 2.2. For en mer dyptgående gjennomgang av de respektive standarder henvises det til IEEE 802 sin hjemmeside [9].

Protokoll	Utgitt	Frekvens	Typisk throughput	Maks datarate	Innendørs rekkevidde	Utendørs rekkevidde
802.11a	1999	5GHz	23Mbit/s	54Mbit/s	ca. 35m	ca. 120m
802.11b	1999	2.4GHz	4.3Mbit/s	11Mbit/s	ca. 38m	ca. 140m
802.11g	2003	2.4GHz	19Mbit/s	54Mbit/s	ca. 38m	ca. 140m

Tabell 2.2: Sammenligning av 802.11a og b/g [9]

<sup>1</sup>Light-Weight Access Point





# 3

## Norsk Rikskringasting (NRK)

*”NRK A/S er et statlig eid aksjeselskap som har til formål å tilby allmennkringkasting for hele Norges befolkning i radio og fjernsyn samt på øvrige medieplattformer”* ([37] side 1). Å beskrive hele NRK sin virksomhet vil føre for langt for denne rapporten. I det følgende vil NRK sin nåværende praksis for kontribusjon bli beskrevet. Hvis Trådløse Trondheim skal kunne brukes som transportmedium for en eventuell kontribusjonsløsning vil det også være interessant å se på hvilke krav NRK stiller til lyd- og videostrømmer som de senere kan kringkaste. I kapittel 12 vil disse kravene stilles opp mot målte resultater og en anbefaling for et eventuelt videre samarbeid gis.

### 3.1 Krav til Strømmer

Lyd			
Type overføring	Kanaler	Forsinkelse	
		En-veis	To-veis
En-veis	2	500ms	1000ms
To-veis m/narrowband retur	2	500ms	1000ms
To-veis m/bredbånd retur	2	500ms	1000ms

Tabell 3.1: Krav til lyd-strømmer NRK.

Video					
Type overføring	Oppløsning i pixler		Lydkanaler	Forsinkelse	
	tur	retur		En-veis	To-veis
En-veis	720x576	-	2	500ms	1000ms
To-veis m/narrowband retur	720x576	-	2	500ms	1000ms
To-veis m/bredbånd retur	720x576	720x576	2	500ms	1000ms

Tabell 3.2: Krav til video-strømmer NRK.

## 3.2 Kontribusjonsteknologier

### 3.2.1 AudioTx Communicator

AudioTx Communicator er en kombinasjon av produkter. Produsenten kaller den *'et ideèlt utstyr for å kringkaste lyd'*. Den er en ISDN lydkodek, men kan også sende og motta real-time lyd over tradisjonelle nettverk som ADSL og kabel modemer. Blant kodekene som støttes er MPEG Lag 2 & 3 og G.722. Til forskjell fra andre ISDN-kodeker som vanligvis er implementert i programvare, er AudioTx Communicator kun programvare-basert. Dette gjør kodken billig i forhold til eksisterende løsninger<sup>1</sup> og den er lettere flyttbar [11].

### 3.2.2 Stratalink

For å bidra til video bruker NRK en STRATA 2GHz Transmitter. Denne er i følge hjemmesiden til produsenten[33] *"Bygd med tanke på moderne mobil kringkasting"*. Den kan brukes til utendørs kringkasting og andre mobile applikasjoner. Løsningen kan kringkaste i DVB-T og konfigureres gjennom Strata TX konfigurator. Innhold som kontribueres kan transporteres i analogt eller digitalt format via simplex, halv duplex eller full duplex konfigurasjoner. Kodingsalternativene løsningen tilbyr inkluderer MPEG (4:2:0, 4:2:2), QPSK<sup>2</sup>, 16QAM<sup>3</sup> og 64QAM og tillater NTSC<sup>4</sup> eller PAL<sup>5</sup> modulering med 2 lydkanaler. Data blir sendt ved hjelp av RF-teknologi på Tx- eller Rx-frekvensene 1.9 til 2.5 GHz eller 2.3 til 2.7 GHz [33].

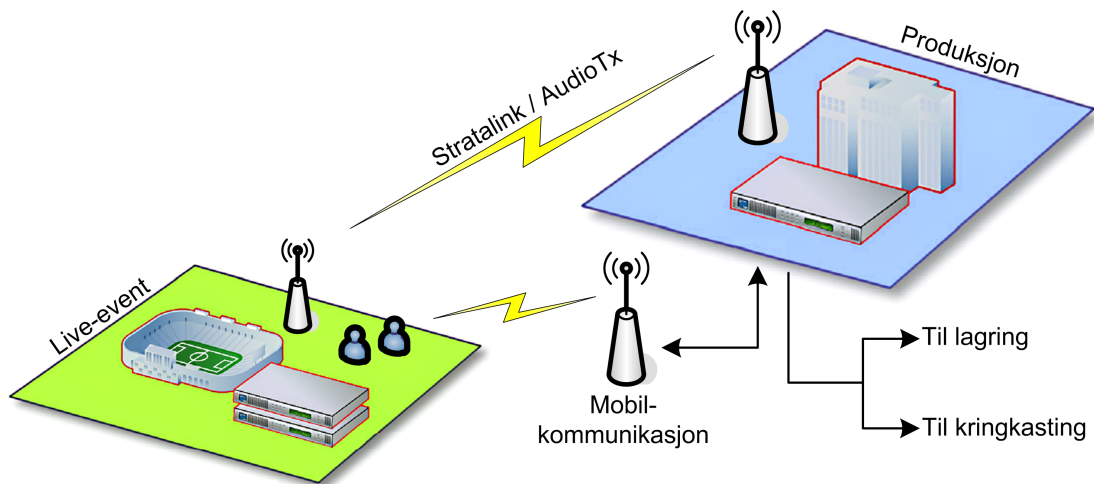
<sup>1</sup>Med eksisterende løsninger menes de som finnes i dag. Framtidige analyser vil kunne avdekke om løsningen fra denne rapporten vil gjøre det bedre.

<sup>2</sup>Quadrature Phase Shift Keying: Digital modulering som beforder data ved å modulere fasen til bæreølgen(referansesignalet).

<sup>3</sup>Quadrature Amplitude Modulation: Et moduleringskjema som beforder data ved å modulere amplituden av to bæreølger

<sup>4</sup>Analogt TV-system

<sup>5</sup>Fargekodesystem brukt i TV-kringkastingssystemer



Figur 3.1: Gjeldende praksis for trådløs kringkasting i real-time.

### 3.3 Scenarier

Etter observasjoner gjort hos NRK og samtaler med Ketil Flatøy vil det i det følgende bli gjengitt et eksempel på NRK sin kontribusjonsløsning. Figur 3.1 gir en oversikt over dagens scenarie.

Ved dekning av en live-event er det alltid personell fra NRK til stede. Dette inkluderer reportere og teknisk bemanning. For å kringaste et live intervju blir intervjuobjektet enten filmet inkludert lyd eller samtalen blir tatt opp. Baserer innholdet seg utelukkende på lyd transporteres dette via AudioTx til en produksjonsenhet. Som nevnt tidligere kan dette foregå på de fleste infrastrukturer som trådløse nettverk, linjetilkobling eller over kabeltilkoblet Internett. Baserer innholdet seg på video blir RF<sup>6</sup>-teknologien til Stratalink benyttet. Dette fordrer at avsender har "line-of-sight" til mottaker. For at et eventuelt studio ved produksjonsenheten skal kunne ha interaksjon med intervju-sesjonen trengs en form for returkanal. I dag er dette løst via mobilkommunikasjon tilbake til reporteren. Når innholdet har ankommet produksjonsenheten blir dette posessert og enten sendt videre til kringkasting eller lagring alt avhengig av settingen.

---

<sup>6</sup>RadioFrequens



# 4

## Kontribusjon over IP-nett

Kontribusjon omhandler transport av materiale i sanntid eller ikke sanntid. Som oftest er det direkte programoverføring mellom ulike lokasjoner, regioner eller nasjoner. Typisk gjelder dette transport mellom studioer, for å distribuere innhold mellom flere TV-kanaler eller for å sende råmateriale til en produksjonsenhet. Kontribusjonslinker trenger ikke være permanente linker over et fast nettverk, men blir ofte brukt for on-demand overføring av programinnhold. I tradisjonelle systemer har MPEG-2 blitt brukt for å kode/dekode materialet [36]. Dette kapittelet vil ta for seg kontribusjon av video og lyd over IP-nett, deretter vil de nødvendige protokollene beskrives.

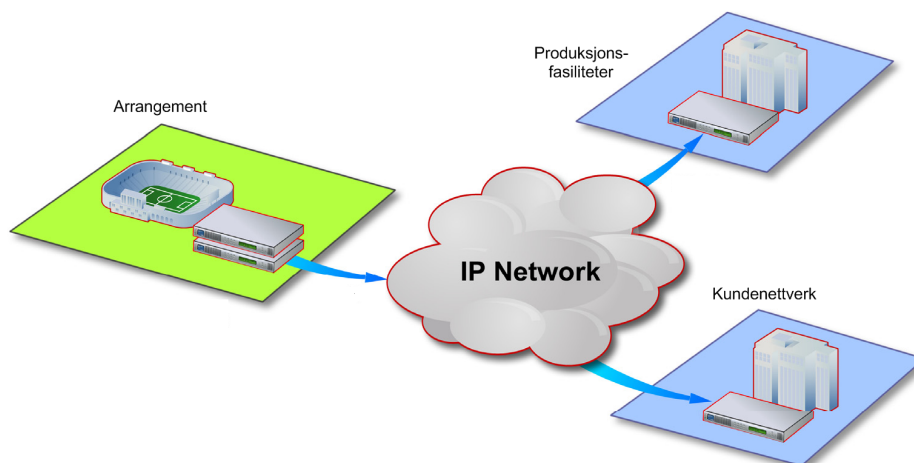
### 4.1 Kontribusjon av video over IP

#### 4.1.1 Introduksjon

Video-overføring over IP har i økende grad blitt populært, derimot er kontribusjon av video over IP (VCIP) for kringkasting er et relativt nytt forskningsområde.

#### 4.1.2 Protokoller for overføring

De fleste oppsett for Streaming Servere benytter seg av transportprotokollen Real-time Transport Protocol (RTP) og Real-time Streaming Protocol (RTSP). Det er nyttig å merke seg at RTSP ikke nødvendigvis er bundet til RTP [47]. Darwin Streaming Server



Figur 4.1: Eksempel på kontribusjon.

brukes i denne oppgaven for testing av H.264 over Trådløse Trondheim. Denne implementasjonen benytter seg også av RTSP for streaming.

### 4.1.3 Pakketap

IP-nettverk er designet for å håndtere data-trafikk og behandler pakker etter best-effort prinsippet. De garanterer hverken levering eller korrekt sekvensiering av pakkene. Når nettverket opplever metning kastes pakker og må sendes på nytt. Alle pakker blir behandlet likt. En tids-sensitiv videopakke kan dermed legges i kø bak en lang rekke forsinkelsestolerante pakker. Resultatet kan da bli pakketap eller store forsinkelser for videopakkene. Ved live overføring av video vil en pakke som er for sent ute bli betraktet som tapt da det ikke gir mening å spille av en sekvens som tilhørte en tidligere hendelse. En vanlig løsning på problemet er å overprovisjonere nettverk slik at metning ikke forekommer. Denne strategien fungerer i de fleste tilfeller, men kan likefremt ikke garantere levering av videopakker. Dette fordi en uavhengig bruker plutselig kan injisere store mengder data i nettet og okkupere kritiske linker.

### 4.1.4 Variasjon av forsinkelse

For å oppnå jevn avspilling av video-materiale kreves lav variasjon av forsinkelse (se seksjon 8.1.3). I den siste tiden har det blitt gjort store framskritt i komprimering av video slik at mer video-informasjon kan sendes på lavere bitrater. Dette alene kan ikke garantere lav variasjon av forsinkelse. Andre problemer som overbelastede router-porter

eller buffere med funksjonsfeil ett eller flere steder langs ende til ende stien.

#### 4.1.5 Båndbredde

For kontribusjon av video er det ønskelig å overføre strømmer med høyest mulig bit-rate for å maksimere kvalitet ved ankomst hos produksjonssenteret. Å redusere komprimering før kontribusjon medfører mulighet for mer komprimering ved distribusjon, men medfører dermed et større krav til bitrate. Bitraten og dermed krav til båndbredde avhenger i høyeste grad av hvilken kodek som blir benyttet. Videokodeker blir behandlet under kapittel 5. Det resulterende båndbreddekravet avhenger derfor av parametre som ønsket kvalitet, oppløsning og gitte minstekrav til forsinkelse.

## 4.2 Kontribusjon av lyd over IP

### 4.2.1 Introduksjon

Kringkastere benytter seg i økende grad av IP-nett for kontribusjon av blant annet lyd av høy kvalitet. Hovedgrunnen til dette er at flere har sluttet å tilby ISDN-tjeneste som historisk sett har blitt brukt til dette formålet [34].

Det kan identifiseres tre former for lyd over IP [34] :

- *En-veis overføring*: Lydstrømmen blir sendt direkte til produksjonsenheten uten behov for returlyd.
- *To-veis overføring med narrowband retur*: Lydstrømmen som senere skal kringkastes blir sendt direkte til produksjonsenheten, men det er behov for returlyd. Returlyden skal ikke kringkastes og kan derfor sendes med lave krav til bitrate. Et problem under to-veis overføring er forsinkelse. Metoden brukes gjerne under opptak av direkte intervjuer, derfor er lav forsinkelse viktig for å opprettholde en jevn dialog.
- *To-veis bredbånd overføring*: Lydstrømmen sendes med høy bitrate i begge retninger. Denne metoden er formålstjenelig ved direktesendte diskusjoner mellom flere studio, forsinkelse er i dette tilfellet også et problem.

### 4.2.2 Protokoller for overføring

For kontribusjon av lyd over IP har UDP-transport blitt foretrukket. Generelt gir denne varianten best resultat. UDP brukes i omgivelser hvor real-time levering er viktigere enn pålitelighet. *"De fleste firmaer tilbyr RTP på toppen av UDP"*[41].

Den Europeiske unionen for kringkasting (EBU) har sammen med en rekke utstyrproducenter blitt enige om et felles rammeverk for kontribusjon av lyd over IP. Rammeverket definerer RTP som en felles protokoll, denne anbefalingen finnes under [34].

### 4.2.3 Pakketap

Kontribusjonssystemer for lyd over trådløse nettverk er spesielt sårbare for pakketap. Dette fordi en tapt lydpakke kan i mange tilfeller medføre tap av lyd. Applikasjoner kan bruke sekvensnummer på meldingene, som i RTP, for å detektere pakketap. For å kunne tilby en akseptbar kvalitet bør metoder for feiloppretting/feilkontroll implementeres [42].

Problemet blir ekstra synlig ved lave bitrater hvor etterfølgende pakker gjerne avhenger av tidligere. Metoder for å skjule feil i forsendinger har blitt presentert i [43] og [55], men disse introduserer ekstra forsinkelse for strømmen. Selv om pakketap kan være ødeleggende for opplevd kvalitet er det vist at tap av pakker opp til 5% kan aksepteres i en samtale [25]. 5% er en relativt stor andel tapte pakker for en rå strøm uten mekanismer for korrigering og gjenoppretting, avhengig av regimer for feilkorrigering kan denne prosentandelen godt overstige denne grensen.

### 4.2.4 Variasjon av forsinkelse

Som nevnt i 8.1.3 er variasjon i forsinkelse også kalt jitter. For å redusere jitter har lydkodeker innebygde jitterbuffere. Hvis pakkene ankommer usynkronisert, vil dette bufferet tillate synkronisering av pakkene slik at innholdet kan spilles av i riktig rekkefølge. Dette tilfører da en ekstra forsinkelse til den samlede ende til ende forsinkelsen.

## 4.3 Protokoller for overføring

### 4.3.1 RTP og RTCP

Real-time Transport Protocol (RTP), definert i RFC3550 [46], transporterer lyd og video i sanntid og bygger på UDP for transport. RTP tilbyr ende-til-ende funksjoner for transport på nettverkslaget. Den passer for applikasjoner som overfører i sanntid over både multicast og unicast. RTP omfatter ikke ressursreservasjon og garanterer ikke QoS. Data-transporten er utvidet med Real Time Control Protocol (RTCP) for å tilby skalerbar overvåking av overføring i store multikastnettverk og for å tilby en lettvektsfunksjonalitet<sup>1</sup> for kontroll og identifikasjon. RTP og RTCP er designet med tanke på å være uavhengig av underliggende transport- og nettverkslag.

Siden mediapakkene er sendt over IP er det en mulighet for asynkron ankomst hos mottaker. Hvis to pakker sendes etter hverandre kan det tenkes at pakke nummer to ankommer før pakke nummer en. For at mottakeren skal kunne spille av strømmen på en forståelig måte, må det finnes en mulighet for å sortere pakkene. Dette gjøres i RTP ved hjelp av RTP-timestamps og RTP-sekvensnummer. Mottakeren lagrer pakkene i et

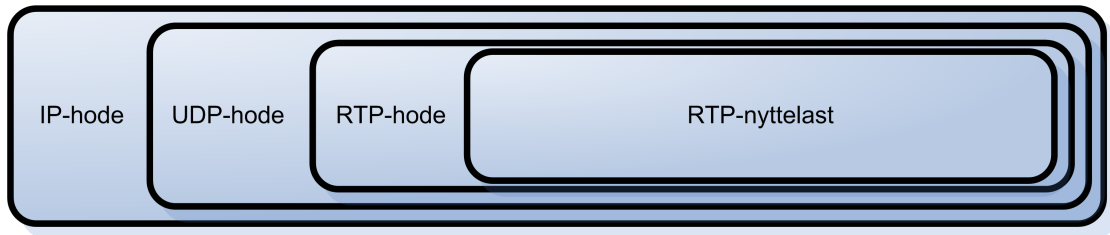
---

<sup>1</sup>Med lettvekts menes enkel å bruke og skaper liten overhead.



buffer og sorterer før avspilling. Sekvensnummer blir også brukt for å detektere pakketap. Den kanskje viktigste bruken av RTCP er at den kan sammenligne RTP-timestamps mot en referanseklokke og dermed kan synkronisere pakkene før avspilling hos mottaker[29].

For en utdypende oversikt over RTP og RTCP henvises det til RFC3550 [46]. Beskrivelsen av RTCP i denne RFC er tilnærmet identisk til RFC1889 (omhandler RTCP spesifikt), men inneholder noen punkter som utdaterer RFC1889.



Figur 4.2: RTP-data i en IP-pakke.

Utrekning av variasjon av forsinkelse er beskrevet i seksjon 8.1.3. Her benyttes RTP-timestamps-enheter. Denne reflekterer samplingsfrekvensen av den første oktetten i RTP-pakken. Samplingsfrekvensen må settes utifra en klokke som øker monotonisk og lineært slik at synkronisering kan utføres og variasjon av forsinkelse kan utlignes [46]. RTP-klokken som brukes til å lage RTP-timestamps er uavhengig av antall kanaler og koding. Normalt er den lik antall *samlingsperioder/sec* [45]. Samplingsfrekvensen som benyttes bør settes til en av verdiene fra tabell 4.1.

8000 Hz
11025 Hz
16000 Hz
22050 Hz
24000 Hz
32000 Hz
44100 Hz
48000 Hz

Tabell 4.1: Samplingsfrekvenser for RTP-klokker [45].

Det er viktig å merke seg at man ikke kan variere raten til RTP-timestamps innad i en sesjon.

### 4.3.2 RTSP

Real Time Streaming Protocol (RTSP) opererer på applikasjons-nivået i ISO-modellen. Protokollen tilbyr et kraftig rammeverk for kontroll over levering av lyd og video i sann-

tid. Datakildene kan være fra et direkte (live) opptak eller klipp lagret på en server. RTSP tar seg av oppsetting og kontroll av én eller flere tids-synkroniserte sammenhengende lyd og/eller video strømmer. Med andre ord er RTSP fjernkontrollen for multimediaservere. Protokollen kan bruke RTP[46] for overføring, men er ikke bundet av en spesiell transport protokoll. I tillegg kan RTSP bruke kanaler som UDP, Multicast UDP og TCP [47].

Strømmene som kontrolleres av en RTSP-tilkobling blir definert av en presentasjonsbeskrivelse. RFC for RTSP definerer ikke et spesielt beskrivelsesregime, men lar det være opp til den respektive utvikler å velge. For Darwin Streaming Servere brukes SDP [8] (Session Description Protocol) til dette formålet.

Bruken av RTSP ligner på mange måter på HTTP/1.1, det er noen forskjeller angitt i [8] som er verdt å merke seg:

- RTSP introduserer mange nye metoder og har en annen protokoll-identifikator (rtsp:// vs http://)
- RTSP må i de fleste tilfeller opprettholde tilstanden til sine noder.
- Data overføres av en annen protokoll.
- Request-URI<sup>2</sup> inneholder alltid absolutt URI.

Protokollen støtter følgende operasjoner:

- Henting av media fra en mediaserver.
- Invitasjon av en mediaserver til en konferanse.
- Legge til media til en eksisterende presentasjon.

Det er også viktig å merke seg at RTSP-anmodninger kan håndteres av proxy-servere, tunneler og cacher som i HTTP/1.1<sup>3</sup> [47].

---

<sup>2</sup>Uniform Resource Identifier

<sup>3</sup>Hypertext Transfer Protocol

# 5

## Videokodeker

### 5.1 MPEG-2

MPEG ble startet i 1988 som en undergruppe av ISO/IEC. Denne gruppen har som mål å definere standarder for digital komprimering av audio-visuelle signaler. MPEG-1 ble publisert i 1993 som ISO/IEC 11172. Denne standarden støtter videokoding opp til 1.5 Mbit/s og lydkoding opp til 192 kbit/s noe som gir en kvalitet på linje med VHS<sup>1</sup>. I løpet av 1990-tallet kom det behov for kodeker som støtter høyere bitrater som kunne brukes til blant annet kringkasting. Resultatet av dette behovet ble MPEG-2 som en videreutvikling av forløperen MPEG-1. MPEG-2 standarden kan kode vanlige TV-sendinger med en bitrate fra 3 til 15 Mbit/s og høy-oppløselige sendinger fra 15 til 30 Mbit/s. MPEG-2 forbedrer lyd kvaliteten i forhold til MPEG-1 i at den tilbyr koding for flerkanals surround-lyd. Det er verdt å merke seg at MPEG-2 også kan dekode MPEG-1-strømmer [51].

Systemspesifikasjonen til MPEG-2 beskriver hvordan MPEG-komprimerte strømmer av lyd- og videodata kan multiplekseres til å danne en enkelt datastrøm. I denne spesifikasjonen defineres et program som en kringkastningskanal. Programet består av en eller flere elementære strømmer. Den enkleste formen for program er en radiokanal som kringkastes ved hjelp av en elementær lydstrøm. Et litt mer komplisert program er en TV-sending. Tradisjonelt inneholder denne en strøm for kodet video, en for kodet lyd og en strøm for undertitler. I mange tilfeller vil det være praktisk å inkludere flere strøm-

---

<sup>1</sup>Video Home System

mer. For eksempel kan det være mulig å velge mellom strømmer som gir ulik oppløsning på video eller ulike språk for lyd og undertitler [44].

Kodingsstandarden for MPEG-2 har blitt laget for effektivt å kunne støtte sammenflettet og progressiv videokoding og produsere SD-video på omtrent 4 Mbit/s. Standarden bruker en blokkbasert hybrid kodingsalgoritme som kompenserer for feil i estimert bevegelese og DCT-transform. Bevegelseskompensasjon utnytter redundans i tidslinja, mens DCT-transformen utnytter redundans i den repsektive bilderammen. MPEG-2 benytter en asymmetrisk kompleksitet for koding og dekodning. Dette medfører dekoderen blir mye enklere og krever mindre prosesseringsressurser, mens høy kvalitet og effektivitet opprettholdes gjennom en mer kompleks koder [27].

## 5.2 H.264/AVC (MPEG-4 del 10)

H.264/AVC er den nyeste internasjonale videokodingsstandarden beskrevet i [58]. Videokoding for telekommunikasjon har utviklet seg gjennom standardene ITU-T H.261, H.262 (MPEG-2) og H.263 for videokoding. Tjenestene har utviklet seg fra bare å omfatte ISDN og T1/E1 til nå å kunne tilby trafikk over PSTN, mobile/trådløse nettverk og LAN/Internett. Målet for utviklingen har vært å maksimere kode-effektiviteten samt å kunne takle de ulike typene nett som finnes. [59]

Den nye standarden er laget for å brukes på minst ett av de følgende bruksområdene:

- Kringkasting over kabel, satellitt, kabelmodem, DSL, osv.
- Interaktiv- eller seriell lagring på optisk og magnetisk utstyr DVD, osv.
- Samtaletjenester over ISDN, Ethernet, LAN, DSL, trådløse og mobile nettverk, modemer, osv.
- Video-on-demand eller multimedia streaming tjenester over ISDN, kabelmodem, DSL, LAN, trådløse nettverk, osv.
- Multimedia messaging services (MMS) over ISDN, DSL, ethernet, LAN, trådløse og mobile nettverk, osv.

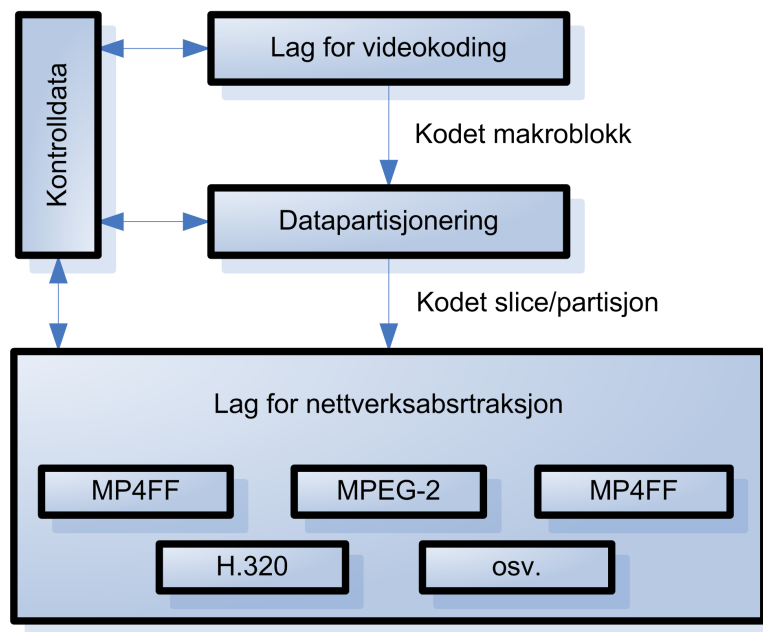
H.264/AVC og de nylig utviklede Fidelity Range Extensions (FRExt [31]) forventes å ha et stort bruksområde innen video-overføring og lagring av høy-oppløst (HD) materiale.

Denne teknologien blir i økende grad akseptert i programvarestandarder og spesifikasjoner. Kodeken med sine utvidelser har i den siste tiden blitt inkludert i løsninger som Digital Video Broadcasting (DVB), HD-DVD og Blu-Ray [53].

Generelt kan video kodes i to ulike varianter. Den ene benytter seg av faste kvantiseringskalaer noe som gir tilnærmet konstant videokvalitet på bekostning av variabel bit-rate. Den andre varianten tilpasser kvantiseringskalaer for å holde en konstant bitrate.

Dette medfører en variabel videokvalitet [53]. I forhold til DVB vil det være hensiktsmessig å benytte varianten med konstant video-kvalitet og variabel bitrate. Dette medfører en større belastning på infrastruktur fordi tilbudt båndbredde i noen tilfeller ikke er tilstrekkelig. I andre tilfeller der strømmens bitrate er relativt lav vil dermed kravene til infrastrukturen bli lavere. Fordelen med konstant videokvalitet er at sluttbrukeren kan forvente lik kvalitet på sendingen.

Nye applikasjoner vil tilstrebe bruk over eksisterende og framtidige nettverk. Dette medfører problemer med å håndtere mangfoldet av både applikasjoner og nettverk. H.264/AVC løser dette problemet ved å introdusere VLC og NAL. VLC er laget for effektivt å kunne representere videoinnhold og NAL formaterer VLC representasjonen og tilbyr headerinformasjon tilegnet det store mangfoldet av transportlag og/eller lagringsmedia (se figur 5.1 ).



Figur 5.1: Struktur for H.264/AVC videoenkoder.

Sammenlignet med MPEG-2 bruker H.264 fra 1/3 til 1/2 mindre bitrate for å produsere tilnærmet lik opplevd kvalitet. Gevinsten i bitrate skyldes innførselen av en rekke verktøy for koding. Dette medfører en mer kompleks kodek [27].

Som beskrevet i tabell 5.1 og [27] kan samme format brukes til alt fra kringasting av TV til mobil streaming. Dette gjøres ved hjelp av ulike definerte profiler. De viktigste profilene er gjengitt i den følgende punktlisten:

- *Baseline Profile (BP)*: Brukes primært for visning på mobile terminaler med lave ytelses-spesifikasjoner. Profilen benyttes ofte til video-konferanser.

Bruks-scenarier	Oppløsning	Frame Rate	Data-rate eksempel
Mobilt innhold	176x144	10-15 fps	50-60 Kbps
Internett/Standard Oppløsning (SD)	640x480	24 fps	1-2 Mbps
HD	1280x720	24fps	5-6 Mbps
Full HD	1920x1080	24fps	7-8 Mbps

Tabell 5.1: Bruks-scenarier i H.264

- *Main Profile (MP)*: Opprinnelig tenkt som hovedprofil for kringkasting og lagring. Profilen ble mindre viktig da High Profile ble utviklet til dette formålet.
- *Extended Profile (XP)*: Tiltent rollen som profil for video-streaming. Profilen har høy kompresjonsevne og tilbyr ekstra funksjonalitet for å håndtere tap av data og bytting av server-stream.
- *High Profile (HiP)*: Primærprofil for kringkasting og lagring av data, spesielt for HD-TV. Profilen er implementert i HD-DVD og Blu-ray.

For en dyperegående gjennomgang av anbefalingen H.264 henvises det til [21] og [22].

# 6

## Lydkodeker

### 6.1 ITU G.711

ITU<sup>1</sup> sin G.711-anbefaling er den mest brukte stemme-kodek i PSTN/ISDN. Kodeken bruker ikke-lineær kvantisering som medfører en bit-rate reduksjon på 40% sammenlignet med lineær kvantisering. Ved å bruke 8-bits PCM og en samplingsrate på 8kHz produserer kodeken en strøm på 64 kbit/s [52]. PCM<sup>2</sup> gir en digital representasjon av et analogt signal. Signalstyrken til de 8000 samplene blir fordelt innenfor 256 forhåndsdefinerte nivåer. Siden et 8-bits uttrykk kan representere 256 verdier, og samplingsraten er 8kHz kan derfor det digitale lydsignalet overføres med en datarate på 64kbit/s [56].

Det er to versjoner av G.711:

- A-law.
- $\mu$ -law.

$\mu$ -law er brukt i T1-standarden i Nord-Amerika og Japan, mens A-law er brukt i E1-standarden i resten av verden. Forskjellen ligger i hvordan det analoge signalet samples. I begge metodene blir signalet samplet på en logaritmisk måte. A-law gir en mer dynamisk rekkevidde sammenlignet med  $\mu$ -law. Resultatet blir da en klarere lyd siden ujevnheter i samplingen blir jevnet ut.

---

<sup>1</sup>International Telecommunication Union

<sup>2</sup>Pulse Code Modulation: Digital representasjon av et analogt signal der signalet blir representert av en rekke tallverdier som representerer signalets størrelsesorden.

Ved å bruke G.711 for sending av digitale lydsignaler vil man oppnå god lyd kvalitet og lav forsinkelse. Dette kan begrunnes ved at det ikke brukes komprimering og at det er samme kodek som brukes i PSTN/ISDN-linjer. Ulempen med ikke å bruke komprimering er et økt behov for båndbredde i forhold til andre kodeker, men etterhvert som tilbudt båndbredde øker vil dette bli et ubetydelig spørsmål [32].

## 6.2 ITU G.729

Kodeken G.729 er en 8-bits CS-ACELP<sup>3</sup>-talekomprimerings algoritme standardisert av ITU-T. G.729 tilbyr høy kvalitet og robusthet, men er til gjengjeld relativt kompleks. Den krever 10ms input-rammer og genererer rammer med 80 bits lengde. Prosesseringen av signalene i 10ms rammer og en 5ms look-ahead introduserer en algoritmisk forsinkelse på til sammen 15ms.

Siden G.279 er basert på CELP-modellen inneholder hver 80 bits ramme koeffisienter for lineær forutsigelse, indekser for kodeboken og forbedringsparametre som blir brukt av dekoderen for å reprodusere tale. Input og output fra algoritmen er 16 bits PCM-sampler som er konvertert til eller fra en 8kbit/s komprimert datastrøm [54].

G.729 er utvidet med to spesifikasjoner G.729A og G.729B. Disse vil bli behandlet i det følgende.

### 6.2.1 G.729 A

Denne spesifikasjonen beskriver en koder med redusert kompleksitet og med flere forenklinger. Disse forenklingene involverer kodeboken for koding og dekoding, mekanismer for søk og etterfiltrering. De nevnte modifikasjonene kan resultere i marginalt lavere talekvalitet, men fordelene ligger i en enklere implementasjon og dermed lavere prosesseringskostnader.

### 6.2.2 G.729 B

Denne spesifikasjonen beskriver en koder som bruker usammenhengende overføring (DTX), deteksjon av stemmeaktivitet (VAD) og generering av komfortabel støy (CNG) for å redusere båndbreddekrav ved å motvirke sending av rammer i talefrie sekvenser.

G.729 er laget og optimert for bruk sammen med anbefalingen V.70. V.70 dikterer bruken av G.729 A, men hvis nødvendig kan hele G.729-anbefalingen brukes for å bedre talekvaliteten [54].

---

<sup>3</sup>Conjugate Structure - Algebraic Code-Excited Linear Prediction



## 6.3 Advanced Audio Coding (MPEG-4 AAC)

*”På bakgrunn av sin enestående ytelse og kvalitet er AAC implementert i kjernen til MPEG-4-, 3GPP- og 3GPP2-spesifikasjonene. AAC tilbyr en langt mer effektiv komprimering enn eldre formater (for eksempel MP3), men leverer kvalitet på linje med ukomprimert CD-lyd” (Oversatt fra [13]).*

MPEG-4 AAC tilbyr en attraktiv sammenheng mellom lyd-kvalitet og bitrate. Kodeken tilbyr lydrekonstruksjon for bitrater helt ned til 32kbit/s per kanal. I trådløse systemer kan AAC brukes til koding av stereolyd på omtrent 32kbit/s. Inngangstrømmen blir samplet på 44.1kHz hvilket resulterer i en rammelengde på 23.22ms som igjen betyr 1024 inngangssampler. Informasjon fra den komprimerte bitstrømmen blir formatert til en pakkestrøm som blir transportert som en lydramme.[38]

MIME-typer som bør brukes for dette fil formatet er audio/aac eller audio/x-aac. Se tabell 6.1 for mulige filendelser.

---

.m4a  
 .m4b  
 .m4p  
 .m4v  
 .aac  
 .3gp  
 .mp4

---

Tabell 6.1: Mulige filendelser AAC

Kodingssystemet er utviklet av blant annet Fraunhofer IIS og tilbyr en rekke forskjellige varianter tilpasset mange bruksområder. Det følgende vil gi en oversikt over verktøyene Fraunhofer IIS har utviklet og som er i bruk i dag. (Kilden til denne informasjonen er Fraunhofer IIS og kommer med påstander om kvalitet som kan virke subjektive.)

### 6.3.1 MPEG-4 AAC-LC

Low Complexity AAC (AAC-LC) er den vanligste varianten. På en bitrate på bare 64 kbit/s pr. kanal tilbyr den en kvalitet som tilfredsstillende kravene fra European Broadcasting Union. Med en samplingsrate fra 8kHz til over 96 kHz og en bitrate opp til 256 kbit/s pr. kanal, er MPEG AAC en av de mest fleksible videokodekene på markedet. I tillegg støtter standarden lydkonfigureringer for mono, stereo og multi-kanal audio-strømmer.

- AAC-LC er standardisert i MPEG-2 og MPEG-4.

### 6.3.2 MPEG-4 HE-AAC

High Efficiency Advanced Audio Coding (HE-AAC), også kjent som AACPlus, er en kombinasjon av AAC-LC med Spectral Band Replication (SBR) bandwidth expansion tool. Kombinasjonen gir god stereokvalitet allerede på bitrater fra 32 kbit/s og egner seg til multi-kanal koding på et spekter av samplingsrater.

- AAC-HE er standardisert i MPEG-2 og MPEG-4.

### 6.3.3 MPEG-4 AAC-LD

Low Delay Advanced Audio Coding (AAC-LD) er standarden MPEG-4 bruker for lyd som skal ha lav forsinkelse og høy kvalitet. Algoritmisk forsinkelse begrenser seg til 20ms og i tillegg tilbyr AAC-LD gode forholdstall for komprimering og god lyd kvalitet for alle typer lyd.

### 6.3.4 MPEG-4 AAC Scalable

MPEG-4 AAC Scalable tilbyr utveksling og dekoding av mangelags bitstrømmer. Bitraten kan tilpasse seg dynamisk til varierende krav som for eksempel båndbreddekapasitet. Den store fordelen med dette er at den kan brukes på kanaler som kontinuerlig har varierende egenskaper eller på kanaler som har ukjent kapasitet mens kodingen foregår.

### 6.3.5 MPEG-4 AAC Scalable Lossless Coding (SLS)

MPEG-4 SLS tilbyr en skalerbar tapsløs utvidelse til AAC. Bruksområder kan være arkivering av rådata eller høykvalitets sending med konstant bitrate.

MPEG-4 profil	Audio Objekt-typer
AAC	AAC LC
HE-AAC	AAC LC, SBR
HE-AAC v2	AAC, SBR, PS
LD-AAC	AAC LD

Tabell 6.2: MPEG-4-profiler og deres tilhørende typer av audio-objekter [10]

AAC støtter 12 typer samplingsfrekvenser fra 8 til 96kHz som vist i tabell 6.3 og opp til 48 lydkanaler [20].

For å kunne møte omgivelser der det er store sannsynligheter for tap av pakker har et nytt forskingsområde tredd frem. Dette området kalles Error Resilient(ER) [38]. De er spesielt laget slik at kodesystemene i seg selv er mer robuste i forhold til feil. Spesifikt for

---

96 000 Hz
88 200 Hz
64 000 Hz
48 000 Hz
44 100 Hz
32 000 Hz
24 000 Hz
22 050 Hz
16 000 Hz
12 000 Hz
11 025 Hz
8 000 Hz

---

Tabell 6.3: AAC samplingsfrekvenser.

AAC har det blitt laget 3 metoder som senere har blitt standardisert i lyd-dimensjonen til MPEG-4:

- Huffman Codeword Reordering (HCR) for å unngå feil i utbredelses for spektrale data [28].
- Virtual Codebooks (VCB11) for å oppdage større feil i spektrale data [38].
- Reversible Variable Length Code (RVLC) for å redusere utbredelsesfeil for skalfaktor data. [38].



# 7

## Streaming

### 7.1 Introduksjon

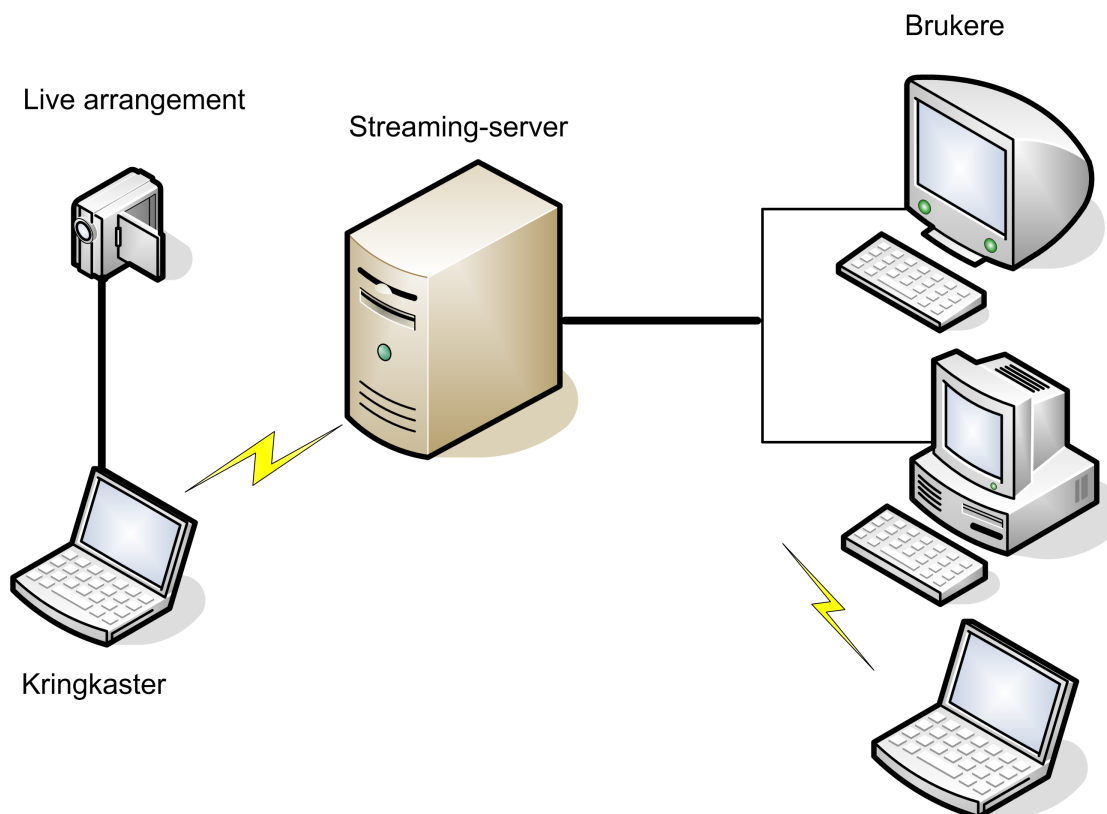
I utgangspunktet var det tenkt at IxChariot<sup>®</sup> skulle brukes til all testing. Det viste seg at dette testprogrammet ikke hadde støtte for H.264/AVC eller AAC. Derfor ble andre muligheter tatt i betraktning. I stedet for å simulere trafikken ble det besluttet å streame H.264-kodet video og AAC-LC-kodet lyd live via Trådløse Trondheim. Løsningen som ble valgt inkluderer en Darwin Streaming Server (DSS) fra QuickTime. DSS er Open Source og kan lastes ned gratis fra Apple sin hjemmeside. DSS representerer et subset av tjenesten tilbudt av betalings varianten QuickTime Streaming Server (QTSS). I det følgende vil streaming generelt- og DSS spesielt beskrives. Konfigurasjon og annet oppsett blir diskutert i kapittel for test-oppsett. Dette kappitelet baserer seg i stor grad på Apple sin Administrators's Guide for QTSS og DSS [12]

#### 7.1.1 Streaming

Streaming er et begrep som brukes på å transportere media fra en server over et nettverk og til en bruker. I motsetning til media-nedlasting vil media-filen ikke fysisk flyttes fra server til bruker, men blir spilt av hos brukeren ettersom den ankommer. Typer overføringer kan variere fra live til on-demand avspilling og det er også mulig å ta imot en forhåndsdefinert liste av media. Dette på lik linje med å velge en vanlig TV-kanal.

### 7.1.2 Darwin Streaming Server

DSS opererer omtrent på lik linje med en normal WebServer. Brukeren forespør gitte media, som svar overføres lyd og/eller video. Forespørslene blir behandlet av RTSP og overførselen blir håndtert av Real-Time Transport Protocol (RTP). Se 4.3.1 for mer informasjon om RTP. DSS kan lage strømmer av videoer og lyd lagret hos streaming serveren, men kan også viderefremde live-strømmer den har tilgang til.



Figur 7.1: Enkelt oppsett for streamingserver.

Figur 7.1 viser et enkelt oppsett for å streame lyd og bilde. I tilfellet med on-demand overføring er mikrofon og videokamera fra en eventuell live-event unødvendig. Streamingserveren brukt i denne oppgaven er nevnte Darwin Streaming Server, denne virker som en viderefremningsentral ved live kringkasting, mens den for on-demand overføring opptreer som et felles lagringsområde.

For å kunne oppnå live kringkasting er det behov for utstyr for å fange opp bilde og lyd samt programvare for å viderefremde. Eksempel på programvare til dette formålet beskrives kort under kapittel 7.2 QuickTime<sup>TM</sup>suite.

### 7.1.3 Hinting av media

En fil som støtter streaming inneholder informasjon om dataenhetene som skal streames. Denne informasjonen kalles hint-spor og legges ved andre media-spor i filen. Et hint-spor inneholder instruksjoner som streamingsserveren bruker til å lage pakker. Eksempel på slik informasjon kan være header-informasjon eller referansesegmenter for mediasporene. Formålet med et hint-spor er at serveren på bakgrunn av denne skal kunne lage pakker som egner seg for en spesifikk type nettverks-transport [14].

## 7.2 QuickTime<sup>TM</sup>suite

QuickTime har en lang tradisjon for å lage program- og maskinvare for produksjon og distribusjon av lyd og video. Deres produkter er i hovedsak utviklet for MacOS, men mange av utgivelsene har blitt tilpasset andre operativsystemer. Det følgende vil gi en kort oversikt over hvilke løsninger QuickTime<sup>®</sup>bidrar med til kontribusjon, produksjon og kringkasting av lyd og video. I tillegg kommer nevnte Darwin Streaming Server og QuickTime<sup>®</sup>Streaming Server.

### 7.2.1 QuickTime<sup>TM</sup>Player

Gratis medieavspiller som kan håndtere avspilling, interaksjon eller visning av video, lyd, virtual reality eller QuickTime-kompatibel grafikk.

### 7.2.2 QuickTime<sup>TM</sup>Pro

QuickTime Pro er en utvidelse av QuickTime Player. Denne utgaven er betalingsbelagt, men til gjengjeld får man et kraftigt verktøy for produksjon av innhold. Programmet støtter koding/dekoding av alt fra enkle formater til de mer komplekse, samt mulighet for å samle innhold av ulike formater til en mediefil.

### 7.2.3 QuickTime<sup>TM</sup>Broadcaster

I følge QuickTime<sup>TM</sup>Broadcaster sin beskrivelse er dette et system som tillater de fleste, med og uten teknisk bakgrunn, å produsere en live-event. Alle som er koblet til et felles nettverk kan bidra virtuelt. Løsningen baserer seg på Mac OS og er gratis nedlastbar. I oppgavens sammenheng er det viktig at MPEG-4 støttes for live streaming via Internett. I tillegg støtter QuickTime<sup>TM</sup>Broadcaster de fleste kodeker tilbudt av QuickTime<sup>TM</sup>slik at brukere kan tilpasse sine sesjoner slik de ønsker.





# 8

## Testing

### 8.1 Metrikker for testing

#### 8.1.1 Bitrate

For nettverk er bitrate et mål på hvor mye data som blir overført i et gitt tidsintervall, men for medieinnhold også beskrive kvaliteten av en lyd- eller videostrøm. En videostrøm komprimert til 3 mbit/s vil ha bedre dynamikk og avspilling vil opptre skarpere enn hos strøm komprimert til 1 mbit/s. Grunnen til dette er at strømmen komprimert til 3 mbit/s inneholder mer informasjon til å representere innholdet enn strømmen med lavere bitrate. På samme måte som kvaliteten til et bilde måles i oppløsning, måles kvaliteten til en lyd-/videostrøm i bitrate [49].

Resulterende bitrate til en gitt strøm over et nettverk vil avhenge av flere faktorer:

- Bitraten til lyd-/videofilen som streames.
- Tilgjengelig båndbredde.
- Forsinkelse.
- Variasjon av forsinkelse.
- Pakketap.

Forsinkelse, variasjon av forsinkelse og pakketap blir beskrevet i det følgende.

### 8.1.2 Forsinkelse

I følge RFC2679 [1] er "Type-P-One-way-Delay" definert som tiden ( $dT$ ) det tar fra kilden har sendt første bit av en Type-P pakke til destinasjonen har mottatt siste bit av pakken. For å regne ut forsinkelsen foreslår RFC3550 [46] bruken av funksjon 8.1. Hvis  $S_i$  er RTP-timestamp for pakke  $i$  og  $R_i$  er ankomsten i RTP-timestamp-enheter for pakke  $i$ , kan forsinkelsen  $D$  for to pakker  $i$  og  $j$  uttrykkes i henhold til funksjon 8.1.

#### Funksjon 8.1.

$$D_{i,j} = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$$

Å måle en-veis forsinkelse i et system er nyttig av flere årsaker:

- Enkelte programvarer oppfører seg dårlig hvis ende til ende forsinkelsen blir stor og/eller overstiger gitte terskler.
- Desto større forsinkelsen er desto vanskeligere er det for protokoller på applikasjonslaget å støtte store båndbredder.
- Minimumsverdien for forsinkelse er et bra mål på hvordan linken oppfører seg forutsatt lav trafikk.
- Minimumsverdien for forsinkelse er en indikator på forsinkelse kun introdusert av forplantningen i nettet og transport-tid.
- Måleresultatene kan indikere overbelastning på en gitt sti.

Grunnen til å måle en-veis forsinkelse i motsetning til tur/retur-forsinkelse er blant annet at ende til ende stier kan variere i dagens pakke-switchede nettverk. Selv om to stier er symmertiske kan de ha radikalt ulike ytelseskarakterisiker basert på algortimer implementert i ulike routere og switcher langs ende til ende stien [1].

### 8.1.3 Variasjon av forsinkelse

Variasjon av forsinkelse blir noen ganger referert til som jitter. Dette er et begrep som kan skape forvirring fordi det blir brukt på ulik måte av ulike grupperinger. Jitter har vanligvis to betydninger [7]:

1. Variasjon av et synkront signal i forhold til en referanseklokke, også kalt "wander".
2. Variasjon i en metrikk (gjennomsnittlig forsinkelse eller minimal forsinkelse). Denne betydningen blir ofte, men ikke alltid, brukt til å benevne variasjon av forsinkelse.

I henhold til anbefalingen til RFC3393 [7] vil begrepet jitter unngås i denne rapporten og bennevningen variasjon av forsinkelse foretrekkes.

Definisjonen for variasjon av forsinkelse hos IP-pakker (IPdv) kan angis for pakker innad i en pakkestrøm. IPdv er definert for to valgte pakker i en strøm som transporteres fra målingspunkt til målingspunkt. IPdv er forskjellen mellom en-veis forsinkelsen for de valgte pakkene [7].

Grunner til at IPdv oppstår er uforutsette hendelser i nettet. To pakker som transporteres over et pakke-switchet nettverk behøver ikke nødvendigvis følge samme ende-til-ende sti. Dette medfører ulike forutsetninger for transporttiden. Blant hendelser som kan medføre IPdv er routing-forsinkelse, ulike bufferalgoritmer i routere, variasjon i trafikk gjennom ulike knutepunkter og eventuelt feilfunksjonerende noder på ende-til-ende stien. For å minimere problemet er det viktig å innføre tilstrekkelig QoS-kontroll over IPdv-buffre<sup>1</sup> hos mottaker, båndbreddetildeling og prioriteringsmekanismer hos internett-routere [42].

I følge RFC3550 [46], kan variasjon av forsinkelse regnes ut ved hjelp av funksjonene 8.1 og 8.2. For å regne ut IPdv må først forsinkelsen til to pakker  $i$  og  $j$  regnes ut ved funksjon 8.1.

I følge RFC3550 [46] bør interankomst variasjon i forsinkelse regnes ut kontinuerlig ettersom hver pakke er mottatt fra kilden. Ved å bruke forsinkelsen  $D_i$  og  $D_{i-1}$  regnet ut i funksjon 8.1 kan IPdv ( $J_i$ ) regnes ut ved hjelp av funksjon 8.2.

### Funksjon 8.2.

$$J_i = J_{i-1} + \frac{|D_{i-1,i}| - J_{i-1}}{16}$$

For å se på varians i et datasett kan man regne ut forventningsverdien  $E(i)$  til settet for så å se på utslagene fra denne for hver enkelt tuppel. I dette tilfellet kan funksjon 8.3 brukes [60]. Dette betyr at verdiene blir skiftet slik at de oscilerer rundt 0 i stedet for rundt forventningsverdiensom i det oprinnelige datasettet.

### Funksjon 8.3.

$$J_i = D_i - E(i)$$

#### 8.1.4 Pakketap

RFC2680 [2] definerer metrikken en-veis pakketap. I følge definisjonen er Type-P-One-way-Packet-Loss fra kilde til destinasjon lik 0 hvis kilden har sendt det første bittet av pakken og destinasjonen har mottatt denne. Type-P-One-way-Packet-Loss er lik 1 hvis

---

<sup>1</sup>IPdv-buffer; ofte kalt jitterbuffer.

pakken ikke blir mottatt. Denne definisjonen gjelder pakke for pakke. Det er mer interessant å se på samlet pakke tap for en strøm over en gitt periode. I denne sammenhengen gjelder definisjonen framsatt av RFC1242 [4]. Denne sier at pakketap er prosentandelen av pakker som ikke har blitt overført av det totale antall pakker som, under under konstante omstendigheter, skulle blitt overført.

Ved implementasjon av testing for pakketap er det viktig å ta følgende punkter i betraktning:

- Hvor ligger skillet mellom tapt pakke og pakker med stor forsinkelse? For eksempel, i mange real-time tilfeller vil en pakke med stor forsinkelse regnes som tapt.
- Vil mottatt pakke som er korrumpert bli regnet som tapt.
- Hvis en pakke blir duplisert under transport og ellers er intakt slik at flere kopier av pakken mottas, vil pakken regnes som mottatt?
- Hvis pakken blir fragmentert og sammensetting ikke finner sted, vil pakken regnes som tapt?

Pakketap kan i hovedsak skyldes tre feilkilder:

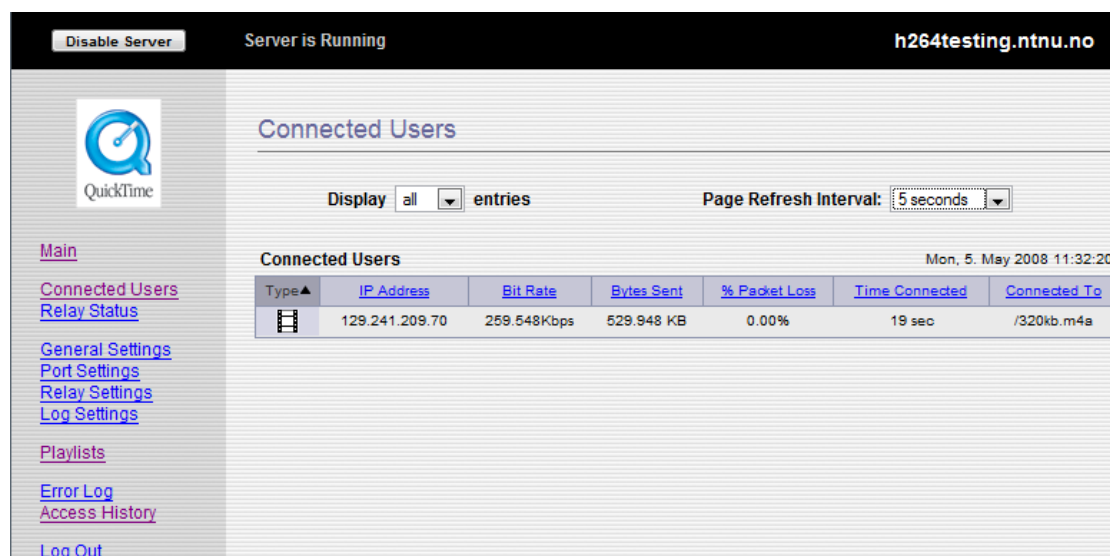
- Synkronisering mellom klokken til kilde og mottaker.
- Terskel for pakketap relatert til klokkesynkronisering.
- Begrensende faktorer i nettverket og/eller programvare for mottak av pakker.

## 8.2 Programvare brukt til testing

### 8.2.1 Darwin Streaming Server

Darwin Streaming Server (DSS) 4 støtter levering av media over Internett i real-time. Brukere kan se/høre på kringkasting av live eller lagret media. Brukere kan spille av strømmene så fort disse når fram og behøver derfor ikke vente til hele strømmen er lastet ned. Funksjonaliteten til DSS tillater direkte strømming av hintede MPEG-4 strømmer uten å måtte konvertere til QuickTime<sup>®</sup>sitt .mov-format. En DSS server tillater brukeren å sette opp en egen radio-/TV-kanal for å distribuere film og lyd til både sine venner og andre interesserte [12].

I denne oppgavens sammenheng blir administrasjonsserveren for DSS utnyttet til testingsformål. Kildekoden til denne serveren har blitt modifisert i henhold til seksjon 9.1. Dette tillater uthenting av data som ellers bare ville blitt vist i det grafiske grensesnittet som vist i figur 8.1.



Figur 8.1: Grafisk grensesnitt for Darwin Steaming Server.

### 8.2.2 WireShark

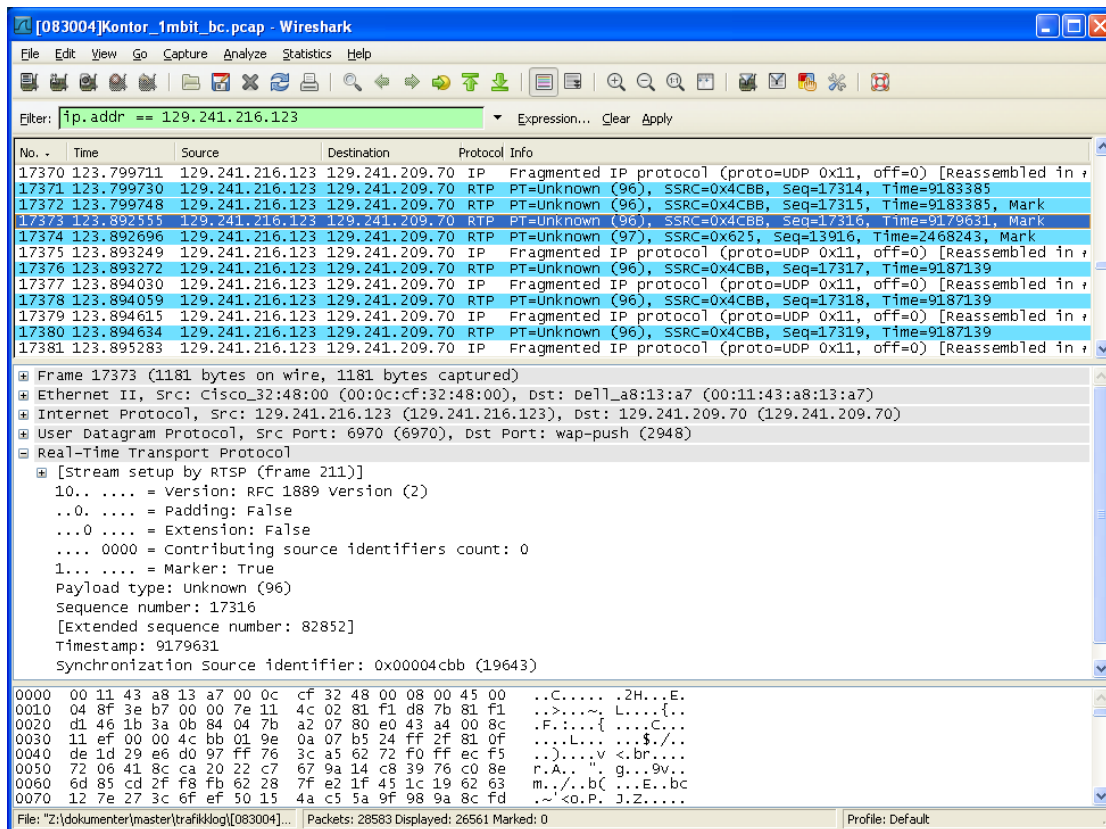
WireShark er Open-Source programvare for å analysere pakker sendt over nettverk. En slik analysator prøver å fange opp pakker fra et gitt interface for så å vise en så detaljert informasjon om disse pakkene som mulig. Mulige bruksområder for WireShark inkluderer [30]:

- Feilsøking i nettverk
- Søk etter feil i sikkerhetsimplementasjoner.
- Feilsøking i implementerte protokoller.
- Lære hvordan nettverksprotokoller opererer.
- For denne oppgaven brukes WireShark til å fange opp RTP-pakker og fra dem regne ut forsinkelse og variasjon av forsinkelse. Eksempel på resultat fra pakkefangst finnes på vedlagt DVD under mappen måleresultater, og grafisk grensesnitt er illustrert i figur 8.2.

Det er viktig å merke seg at WireShark kun måler hva som skjer på et interface. Det er ikke mulig å manipulere trafikk.

Oppdaterte manualer for WireShark finnes under:

<http://www.wireshark.org/docs/>



Figur 8.2: Grafisk grensesnitt for Wire Shark.

**Del II**

**Testimplementasjon**

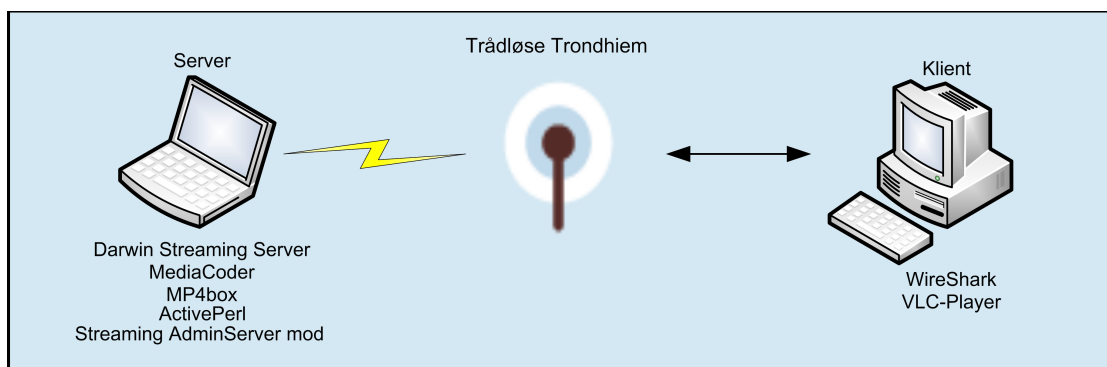




# 9

## Fremgangsmåte

For å vise hvordan testene har blitt utført og i framtiden kunne gjenskape testene og testresultatene gjengis fremgangsmåten i dette kapitlet. Beskrivelsen gir en detaljert steg for steg oversikt over hvordan de ulike programvarene har blitt installert og hvordan testene har blitt utført. I kapittel 10 blir programvaren og maskinvaren spesifisert slik at eventuelle feilkilder kan identifiseres og/eller utelukkes. I det følgende vil tall uthevet i fet skrift og med parentes angi de respektive stegene i prosessen, eksempel **4.**).



Figur 9.1: Skille mellom klient og server for oppsett av programvare.

## 9.1 Programvareinstallasjon

Før testing utføres må nødvendig programvare lastes ned og settes opp. Det er ikke nødvendig å sette opp alle varianter av programvare på alle noder. Derfor blir det i det følgende skilt mellom klient-side og server-side jamnfør figur 9.1.

1.) Last ned gjeldende versjoner av:

- Active Perl  
<http://aspn.activestate.com/ASPN/Downloads/ActivePerl/>
- Darwin Streamin Server  
<http://developer.apple.com/opensource/server/streaming/>
- MP4box  
<http://www.videohelp.com/tools/mp4box>
- Media Coder  
<http://mediacoder.sourceforge.net/download.htm>
- WireShark  
<http://www.wireshark.org/download.html>
- VLC-Player  
<http://www.videolan.org/vlc/>

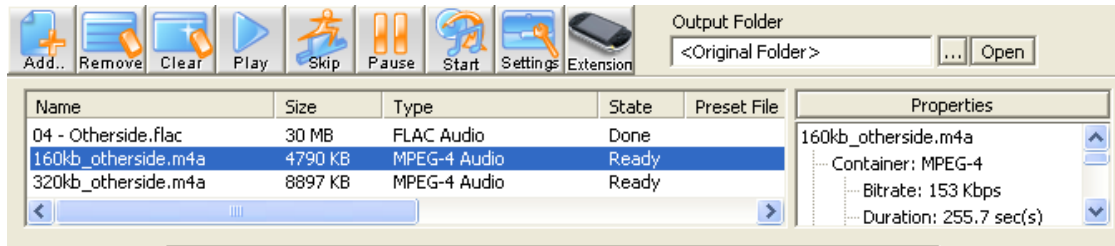
2.) Installer programvaren i henhold til figur 9.1. ActivePerl må installeres før Darwin Streaming Server. Streaming AdminServer-mod vil bli forklart i nærmere detalj under seksjon 9.3.

3.) Valider installasjoner.

4.) Modifiser Streaming Admin Server ved å lime kildekoden fra vedlegg B inn fra og med linje 268 i perl-filen:

```
[rot]/Darwin Streaming Server/AdminHtml/tag_vals.pl
```

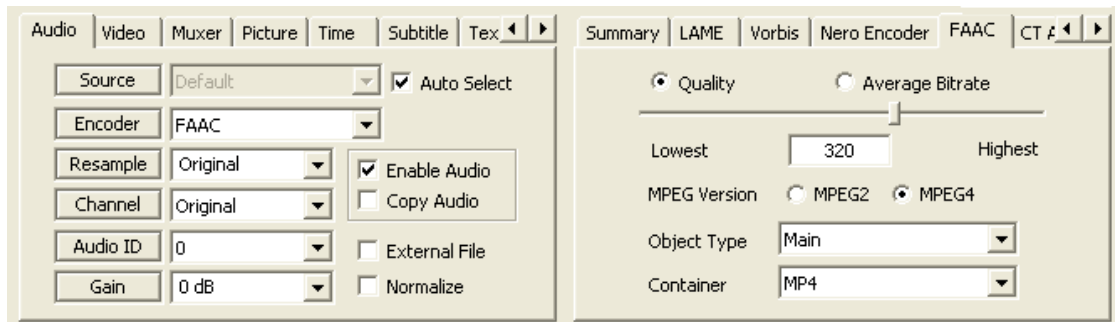
Programvaren kan settes opp og brukes relativt problemfritt. For oppsett av Darwin Streaming Server henvises det til brukermanualen [12]. Det største problemet er kommunikasjon mellom nodene. Derfor er det viktig å sette opp en eventuell brannvegg slik at Darwin Streamin Server, WireShark og VLC-player kan aksessere Internett når de har behov for dette. Det kan i tillegg være hensiktsmessig å tillate streaming på port 80 for å unngå blokkering i andre nettverk. Dette kan gjøres i Streaming Admin Server.



Figur 9.2: Eksempel på tillegging av media i MediaCoder.

## 9.2 Klargjøring av media

- 5.) Åpne MediaCoder og legg til ønsket lyd-/videoklipp i listen (se figur 9.2).
- 6.) Transkod klippene til ønskede format med ønskede parametre ved å gjøre innstillinger i MediaCoder (se figur 9.3).



Figur 9.3: Eksempel på setting av media-parametre i MediaCoder

På bakgrunn av seksjon 7.1.3 må alle mediafiler som skal streames inneholde et hintspor. For å hinte media-klipp er det viktig å merke seg at det er nødvendig å navigere fram til mappen hvor MP4box.exe er lokalisert.

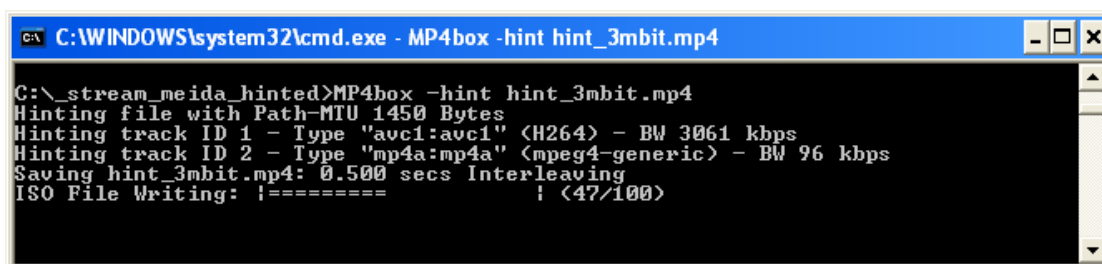
- 7.) Hint medie-klipp ved hjelp av MP4box fra kommando-linja(se figur 9.4):

```
MP4box -hint [filbane]
```

- 8.) Legg media-klippene i mappen designert som serverrot.

## 9.3 Testing

Denne seksjonen angir praktisk framgangsmåte for testingen. For å repetere testene punktene 9 - 16 til antall gjennomkjøringer er tilfredsstillende.



```
C:\WINDOWS\system32\cmd.exe - MP4box -hint hint_3mbit.mp4
C:\_stream_meida_hinted>MP4box -hint hint_3mbit.mp4
Hinting file with Path-MTU 1450 Bytes
Hinting track ID 1 - Type "avc1:avc1" <H264> - BW 3061 kbps
Hinting track ID 2 - Type "mp4a:mp4a" <mpeg4-generic> - BW 96 kbps
Saving hint_3mbit.mp4: 0.500 secs Interleaving
ISO File Writing: !=====
                   ! <47/100>
```

Figur 9.4: Eksempel på hinting med MP4box.

### 9.3.1 Server

- 9.) Start opp Darwin Streaming Server fra kommandolinjen.

```
[rot]/DarwinStreamingServer.exe -d
```

I visse tilfeller vil denne gi en feilmelding som sier at et annet program allerede lytter på settet av porter som DSS trenger. Hvis dette er tilfellet kan det lønne seg å stenge disse programmene. I de fleste tilfeller betyr denne feilmeldingen at DSS allerede kjører og man kan fortsette til neste punkt. Eventuelt kan man via administrasjonsserveren vurdere om serveren kjører eller ikke.

- 10.) Start opp administrasjonsserveren for DSS.

```
[rot]/perl streamingadminserver.pl
```

- 11.) Åpne grensesnitt for serveradministrasjon i en nettleser:

```
http://127.0.0.1:1220/
```

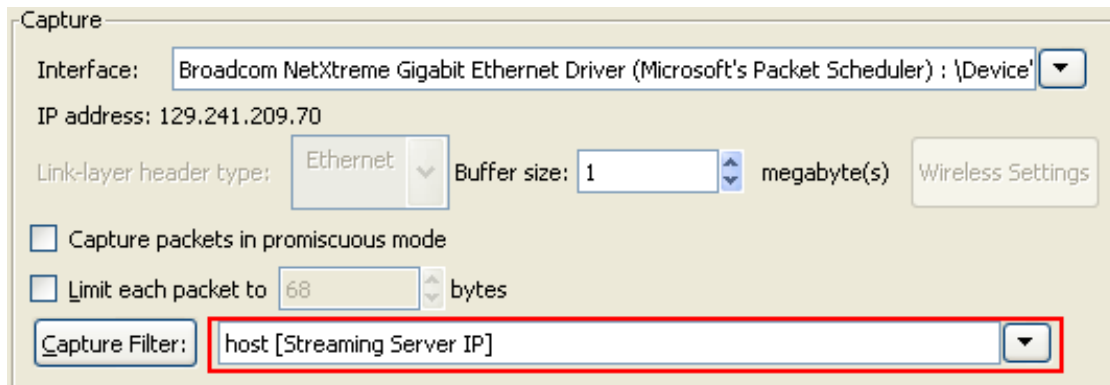
- 12.) Åpne linken *connected users* og velg ønsket oppdateringsintervall. Når klienten anmoder en kobling mot serveren vil denne siden oppdatere seg med gjeldende data i henhold til valgt intervall og verdiene lagres i en loggfil:

```
[rot]/Darwin Streaming Server/AdminHtml/test_H264.log
```

### 9.3.2 Klient

- 13.) Start opp WireShark og lytt på aktivt grensesnitt for å fange opp pakker jamnfør figur 9.5. Filtre bort pakker som ikke kommer fra Streaming Serveren (Capture Filter). *Det er viktig å presisere at WireShark kjøres fra klienten.*

```
Menyvalg: Capture -> Options
Velg riktig grensesnitt.
Capture Filter settes til: host [Streaming Server IP]
```



Figur 9.5: Capture-meny WireShark.

14.) Start opp VLC-Player og start streaming jamnfør figur 9.6.

Menyvalg: File → Open Network Stream  
 Sett feltet for http:// eller rtsp:// til:  
 [Server-IP] : [port] / [Navn]



Figur 9.6: VLC-Player meny for streaming.

Når strømmen er startet vil Streaming Admin Server produsere en loggfil for tilkoblingen og WireShark vil fange opp alle pakker mottatt på spesifisert grensesnitt.

## 9.4 Uthenting av data

### 9.4.1 Server

15.) Hent loggfilen laget av Streaming Admin Server:

```
[rot]/Darwin Streaming Server/AdminHtml/test_H264.log
```

### 9.4.2 Klient

16.) Hent statistikk fra strømmen:

```
Marker en vilkårlig RTP-pakke i ønske strøm.  
Menyvalg: Statistics -> RTP -> Stream Analysis...  
La WireShark analysere strømmen.  
Menyvalg: Save as CSV...
```

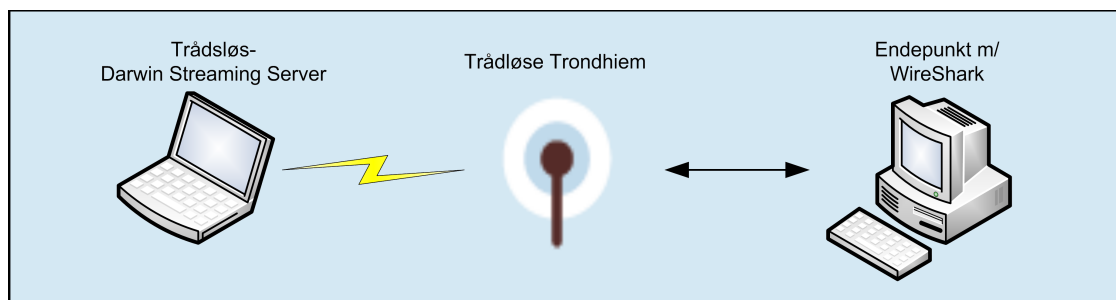
Loggfilene som genereres inneholder komma-separerte verdier (CSV) og kan brukes til å lage grafer og statistikk. I denne oppgaven har det blitt valgt å lage et PHP-script som leser CSV verdiene, legger de inn i en SQL-database og til slutt lager grafer. Kildekoden til disse PHP-scriptene er vedlagt denne oppgaven i vedlegg B. Fordelen med disse er at grafer og annen statistikk kan genereres ved hjelp av interaksjon med en database.

# 10

## Spesifikasjoner

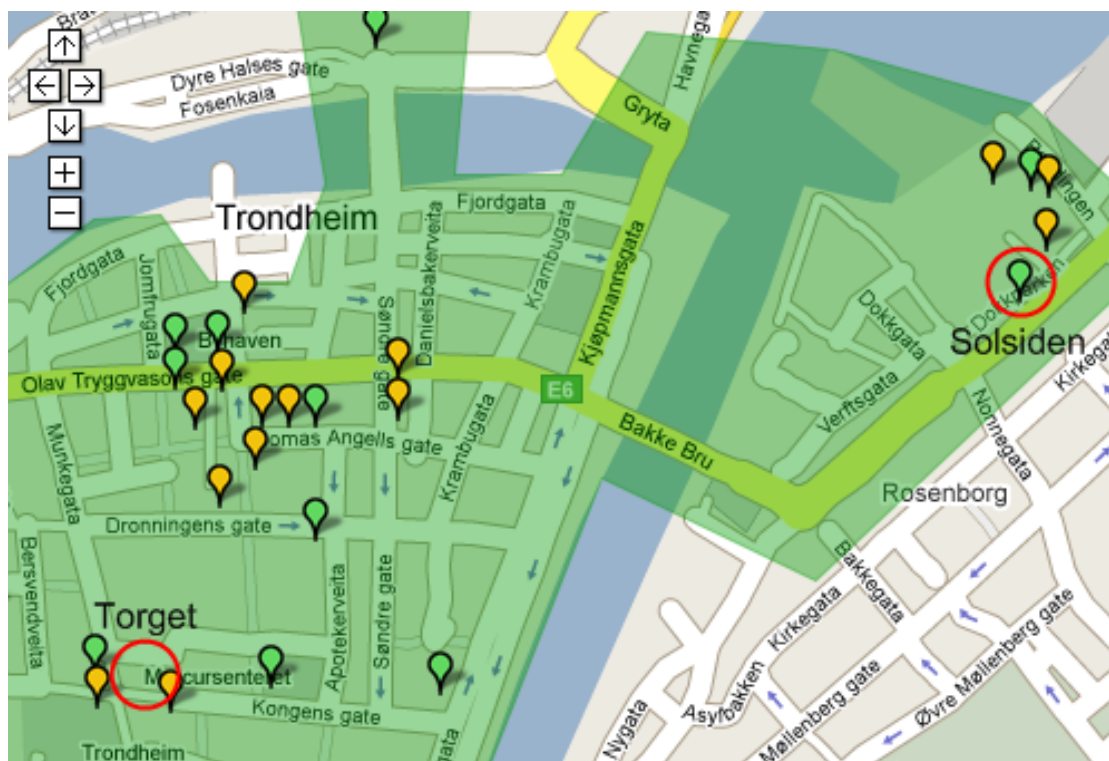
### 10.1 Oppsett

Figur 10.1 viser overordnet testimplementasjon. For å teste kvaliteten til ulike formater av media ble det valgt å sette opp en streamingserver på en laptop med trådløs tilkobling. Denne kan da flyttes rundt i nettet til Trådløse Trondheim og på den måten simulere et reportasje-team som er ute på oppdrag. Den resulterende strømmen hentes ved hjelp av en stasjonær PC som via Trådløse Trondheim kobler seg til media liggende på den trådløse serveren. Når endepunktet har koblet seg til strømmen, overvåkes den av analyseverktøyet WireShark og ved hjelp av administrasjonsservern til Darwin Streaming Server. Se kapittel 9 for en mer detaljert oversikt over gjennomføringen.



Figur 10.1: Testoppsett

## 10.2 Testlokasjoner



Figur 10.2: Dekningskart for TrT med angitte testlokasjoner.

For å kunne utføre tester i Trådløse Trondheim ble det valgt lokasjoner på bakgrunn av anbefalinger fra veiledere og med støtte fra dekningskart<sup>1</sup> angitt i figur 10.2. Røde sirkler i denne figuren angir spesifikke testlokasjoner. Testing fra *Torget* ble gjort uten-dørs i en avstand av omtrent 15 meter fra nærmeste aksess punkt. Denne lokasjonen ble valgt grunnet god tilgjengelighet til flere aksesspunkter. Testing fra *Solsiden* ble utført i lokalene til Bare Blåbær i en avstand på omtrent 5 meter fra nærmeste aksesspunkt. Denne lokasjonen ble valgt fordi NRK har en ukentlig radiosending i nabolokalet. Ideelt sett burde nabolokalet til Bær og Bar blitt benyttet, men det viste seg at TrT ikke tilbyr innendørs dekning i dette lokalet. Derfor ble det valgt å stille seg opp nærmere nabolokalet. En mulig løsning for TrT vil være å tilby innendørs dekning i disse lokalene ved å sette opp et aksesspunkt.

I tilfellet testing ved de nevnte lokasjoner skulle medføre utilstrekkelige resultater ble også en tredje lokasjon valgt. I denne oppgaven refres den til som *Kontoret*. Denne

<sup>1</sup>Dekningsområdet utvider seg kontinuerlig, seneste versjon kan finnes på Trådløse Trondheim sine hjemmesider: <http://www.tradlose trondheim.no/>



lokasjonen er et lukket rom med et isolert aksesspunkt. Dette punktet ble i hovedsak brukt til å prøve ut ulike testmetodikker og testingen ble utført i en avstand på 1.5 meter fra aksesspunktet.

Lokasjon	$\Delta$ AP	Forventet X-trafikk	Omgivelser
<i>Kontoret</i>	ca. 1.5m	Minimal	Innendørs
<i>Torget</i>	ca. 15m	Høy	Utendørs
<i>Solsiden</i>	ca. 5m	Middels	Innendørs

Tabell 10.1: Oversikt over testlokasjoner.

## 10.3 Programvare

### 10.3.1 Darwin Streaming Server

Darwin Streaming Server	
Serverversjon	DSS/5.5.5
Server API-versjon	4.0
Build	489.16
Plattform	Win32
Status	beta

Tabell 10.2: Darwin Streaming Server Build-spesifikasjon

### 10.3.2 WireShark

WireShark	
Versjon	1.0.0
	Build 2600
Kjører på	Windows XP Service Pack 2
WinPCap versjon	4.0.2
Basert på	libpcap versjon 0.9.5 uten AirPcap.

Tabell 10.3: WireShark Build-spesifikasjon

### 10.3.3 VLC-Player

VLC-Player	
Versjon	0.8.6e Janus (wxWidgets interface)
Compiler	gcc versjon 3.4.5

Tabell 10.4: Spesifikasjon for VLC-Player

### 10.3.4 MP4box

MP4Box	
GPAC version	0.4.4
Compiled by	Kurtnoise - Jun 3 2007
GPAC Copyright	©Jean Le Feuvre 2000-2005 ©ENST 2005-200X

Tabell 10.5: MP4box-spesifikasjon.

## 10.4 Maskinvare

### 10.4.1 Server

Fujitsu Siemens AMILO Pro V3515	
Prossessor	Intel® Celeron®
Klokkehastighet	1.60GHz
Minne(RAM)	2.00GB
Windows Vista™ Basic	
Versjon	2007
Systemtype	x86
	Service Pack 1
Nettverksadapter	AR5005GS, spec: C.1

Tabell 10.6: Spesifikasjon for Fujitsu Siemens AMILO Pro V3515

**10.4.2 Klient**

Dell OPTIPLEX GX280	
Prossessor	Intel® Pentium® 4
Klokkehastighet	3.00GHz
Minne(RAM)	1.49GB
Microsoft Windows XP Professional	
Versjon	2002
Systemtype	x86
	Service Pack 2
Nettverksadapter	Broadcom NetXtreme 57xx

Tabell 10.7: Spesifikasjon for Dell OPTIPLEX GX280



Del III

**Evaluering**



# 11

## Resultater

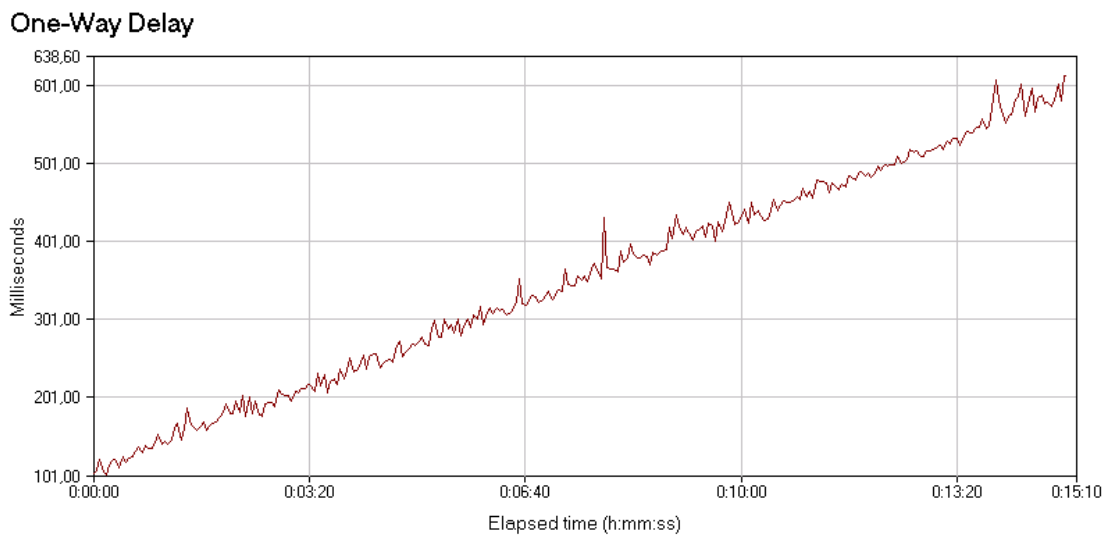
Dette kapitlet angir resultater fra tester utført i nettet til Trådløse Trondheim våren 2008. For å vise sammenheng i utvikling til ulike metrikker har det her blitt valgt å representere et lite utvalg av resultatsettene. I testene vil det bli fokusert på metrikkene fra teorien seksjon 8.1, og observerte sammenhenger mellom disse vil bli beskrevet.

Nyttige henvisninger med tanke på nomenklatur og spesifikasjoner:

- Framgangsmåte og testoppsett finnes under del II.
- Beskrivelse av testlokasjoner finnes under seksjon 10.1.
- Komplette grafiske presentasjoner av målesett finnes under vedlegg E.
- Mediakaraktetika finnes under vedlegg D.
- Tabell i vedlegg F.1 viser samlet statistikk for alle målinger.

## 11.1 Laststudie

Ved hjelp av IxChariot<sup>®</sup> ble ulike nettverkslaster prøvd ut. Denne typen test ble gjort i oppstartfasen for å bestemme hvor store laster nettverket og et eventuelt kontirbusjons-system kunne tåle. Det ble konkludert med at grensen for ulike laster kunne settes i underkant av 5mbit/s. Figur 11.1 viser resulterende en-veis forsinkelse ved et påtrykk av en 5mbit/s MPEG-2- strøm. Av denne grafen kan man lese at forsinkelsen øker relativt lineært. Etter studie av hva som skjer i systemet ved overlast ble det konkludert med at problemet ligger i lesing fra og skrivning til buffere. Hvis det ble skrevet mer til et buffer enn bufferet klarte sende videre ut, medførte dette lignende resultat vist i figuren. Forsinkelsen blir additiv og trafikken hoper seg opp i bufferet. For påtrykk av høyere bitrater blir resultatet det samme, forskjellen ligger i stigningstallet til forsinkelses kurven. Desto høyere påtrykk, desto raskere stiger kurven for en-veis forsinkelse. For lavere påtrykk enn 5mbit/s oppfører kurven seg normalt og oscilerer som forventet rundt sitt gjennomsnitt.



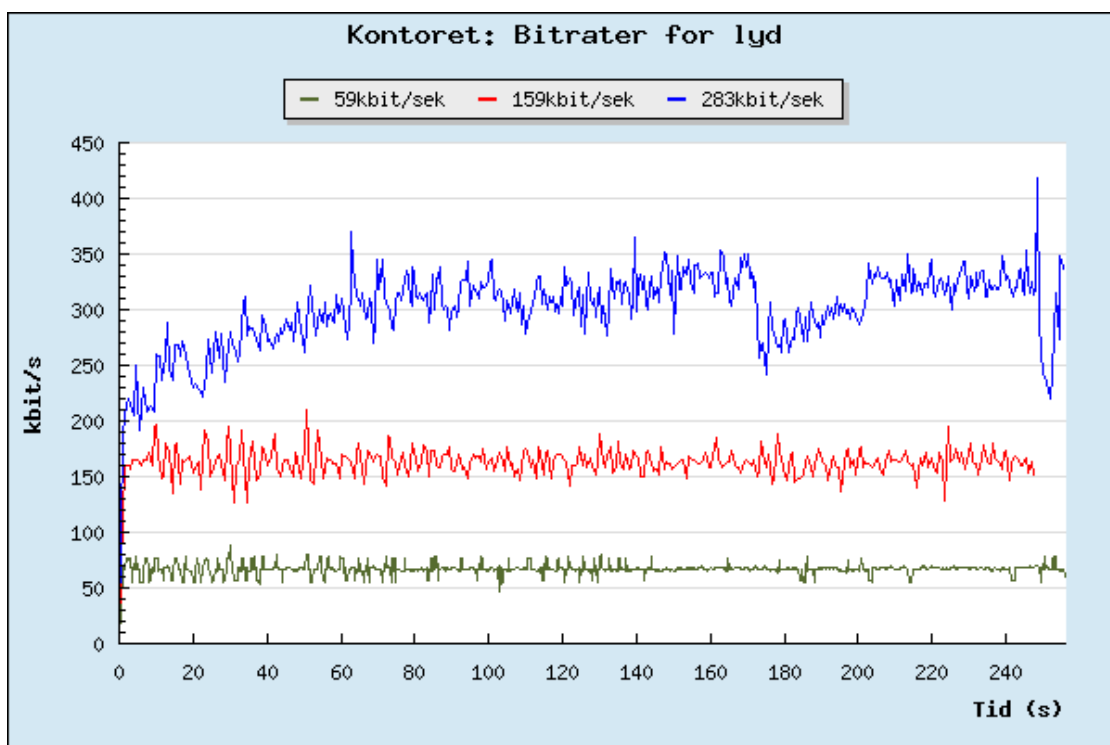
Figur 11.1: Forsinkelse ved overlast.



## 11.2 Lyd

### 11.2.1 Målinger

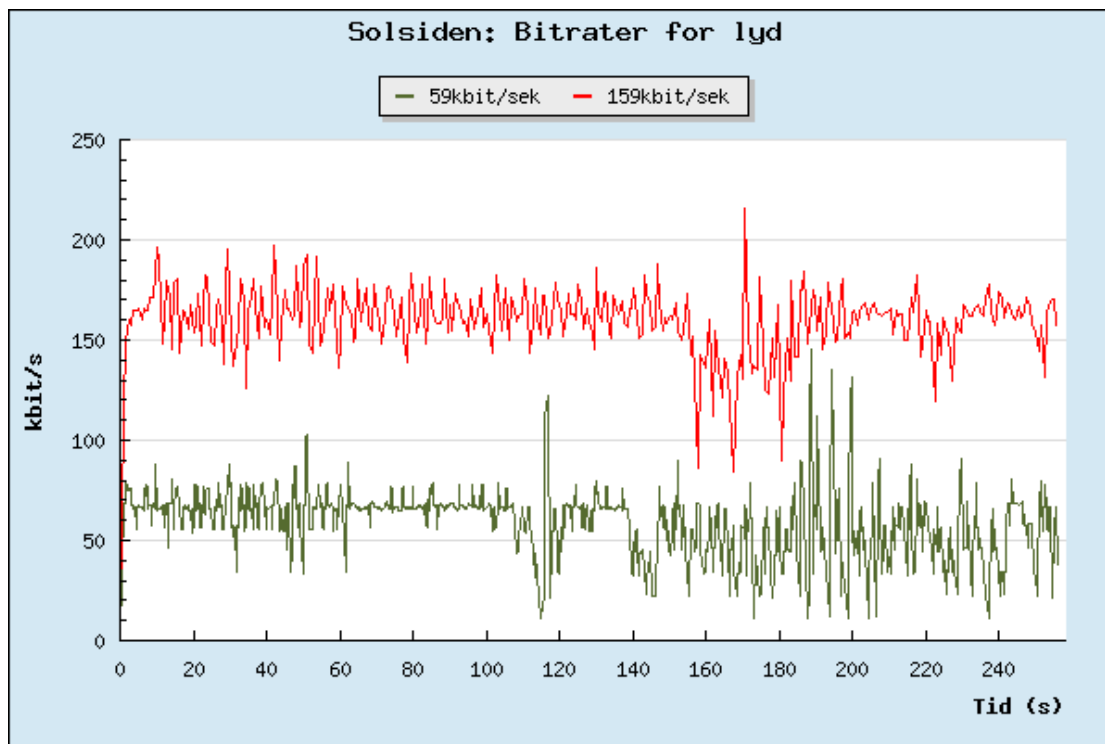
Første test involverte lydstrømmer med gjennomsnittlige bitrater fra 59 til 283 kbit/s sendt fra Kontoret. Denne bitraten angir hvor mye informasjon som kodes for senere å kunne spille av lyd. Det er forventet at resulterende bitrate inklusive protokoll-hoder og sendingsinformasjon vil ligge noe høyere. Som vist i figur 11.2 stemmer dette til en viss grad. Grunnen til at resulterende bitrate noen ganger faller under stømmens bitrate er at denne er satt til en gjennomsnittsverdi. Det vil si at lydstrømmenes originale bitrater ikke er konstante, de oscilerer rundt sine respektive gjennomsnitt.



Figur 11.2: Bitrater for lyd sendt fra Kontoret.

Medieoverføring i trådløse nettverk medfører problemer som følge av trådløse nettverk sine ubregnelige egenskaper. Dette kommer også fram i figur 11.2. Sending på testens laveste bitrate medfører ikke store problemer. Resulterende bitrate holder seg relativt konstant. Figuren viser at høyere bitrater medfører en større oscilering rundt forventet bitrate og at resulterende bitrate derfor blir mer uberegnelig. Ved utføring av denne testen må testlokasjonen tas i betraktning. *Kontoret* har minimal krysstrafikk og streaming serveren befinner seg i ubiddelbar nærhet til aksesspunktet. Målinger av bit-

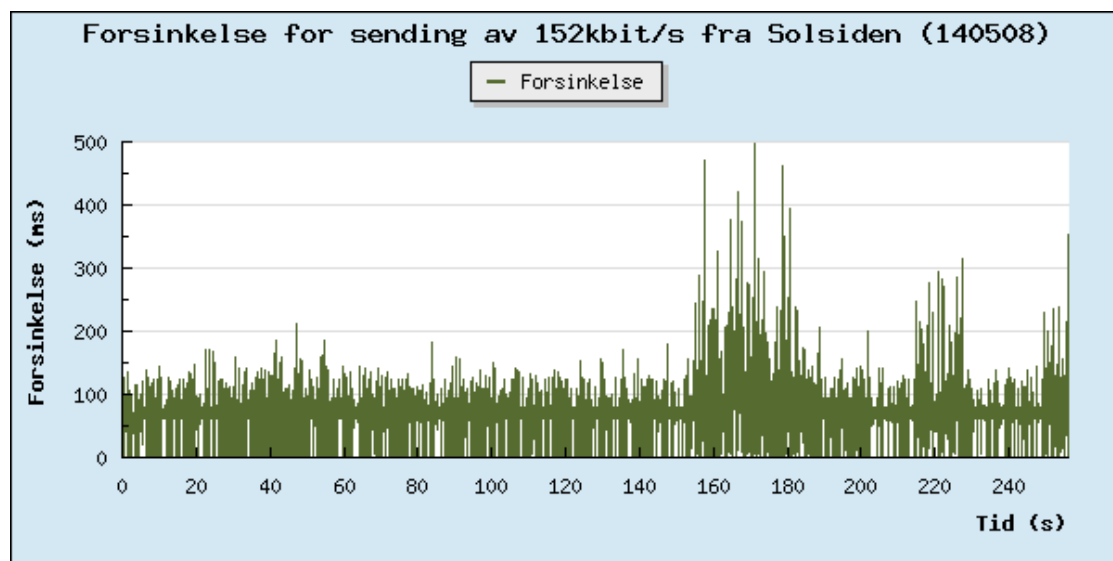
rater fra andre lokasjoner viser større ujevnheter. Figur 11.3 viser resulterende bitrater ved sending fra *Solsiden*. Selv om krysstrafikk her er antatt lavere enn under målinger gjort fra *Torget* og *Kontoret* gir disse resultatene testens største svingning i bitrate.



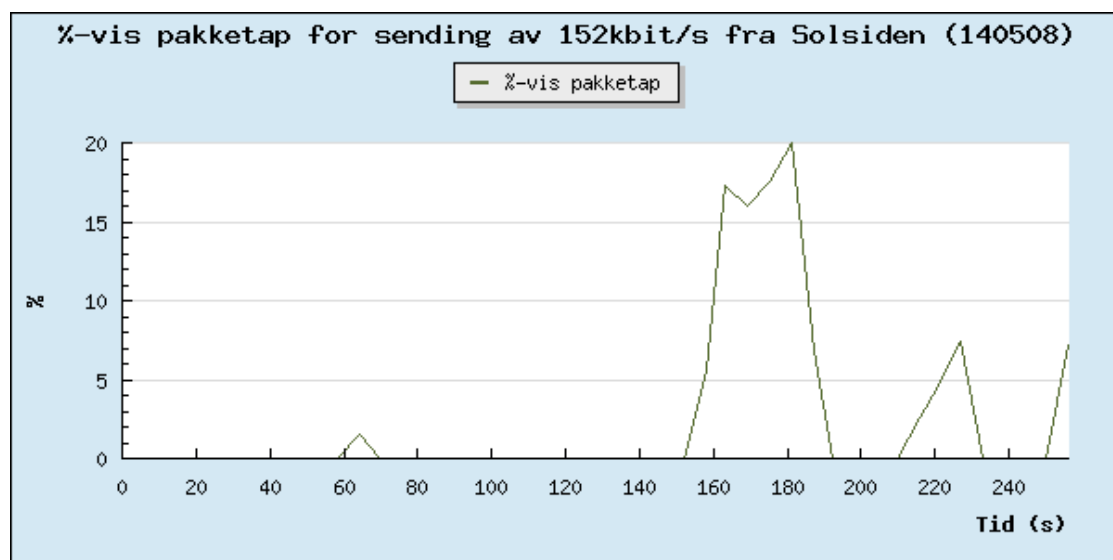
Figur 11.3: Bitrater for lyd sendt fra Solsiden.

Under testing av 283kbit/s strømmen fra Solsiden ble det notert stor aktivitet i lokalene til Bare Blåbær. Mange personer gikk fram og tilbake foran mellom aksesspunktet og streaming serveren. Dette kan forklare noe av de dårlige resultatene.

Ser man resulterende bitrate for lydstrømmen på 159kbit/s i sammenheng med forsinkelse (figur 11.4) og pakketap (figur 11.5) kan man gjøre noen interessante observasjoner. Bitraten synker markant ved tid 160s. Rett i forkant av denne hendelsen har det oppstått et betydelig pakketap. Og i forbindelse med forsinkelse ser man at denne får en markant topp i den beskrevne tidsregion. Etter observasjon av grafene for henholdsvis forsinkelse og pakketap kan det identifiseres et markant mønster. I samme tidsregion som betydelig pakketap oppstår øker forsinkelsen der etter. For alle typer streaming av media er IPdv en viktig metrikk. Grafisk presentasjon av denne utelates her grunnet likheten til grafen for forsinkelse, figuren er gjengitt i vedleggets figur E.53.



Figur 11.4: Forsinkelse ved sending av 152kbit/s lydstrøm fra Solsiden.



Figur 11.5: Pakketap ved sending av 152kbit/s lydstrøm fra Solsiden.

### 11.2.2 Oppsummering

Test for sending av lyd fra de ulike lokasjonene ga varierte resultater. Som forventet ga testene gjort fra *Kontoret* de beste verdiene for alle metrikkene, mens testene gjort fra *Solsiden* kom dårligst ut (se tabell 11.1).

Dato	Måling	Bitrate <sup>a</sup>	Pakketap <sup>a</sup>	Maks. IPdv	Forsinkelse <sup>a</sup>	Maks. forsinkelse
060508	59kb K <sup>1</sup>	66.91kb/s	0.52%	265.35ms	182.16ms	447.51ms
140508	59kb S <sup>2</sup>	69.29kb/s	12.43%	2187.66ms	212.66ms	2400.32ms
070508	59kb T <sup>3</sup>	66.30kb/s	1.30%	491.48ms	184.63ms	676.11ms
060508	152kb K <sup>1</sup>	162.73kb/s	0.29%	368.81ms	64.66ms	433.47ms
140508	152kb S <sup>2</sup>	159.88kb/s	2.37%	431.46ms	66.19ms	497.65ms
070508	152kb T <sup>3</sup>	148.95kb/s	7.37%	334.55ms	71.06ms	405.61ms
060508	283kb K <sup>1</sup>	304.66kb/s	0.22%	406.25ms	25.72ms	431.97ms
140508	283kb S <sup>2</sup>	301.92kb/s	10.57%	745.67ms	29.16ms	774.83ms
070508	283kb T <sup>3</sup>	287.98kb/s	5.25%	465.12ms	27.14ms	492.26ms

Tabell 11.1: Samling av statistikk for for sending av lyd.

<sup>1</sup> Kontoret. <sup>2</sup> Solsiden. <sup>3</sup> Torget.

<sup>a</sup> Verdiene angir gjennomsnitt av resultatsettet for denne metrikken.

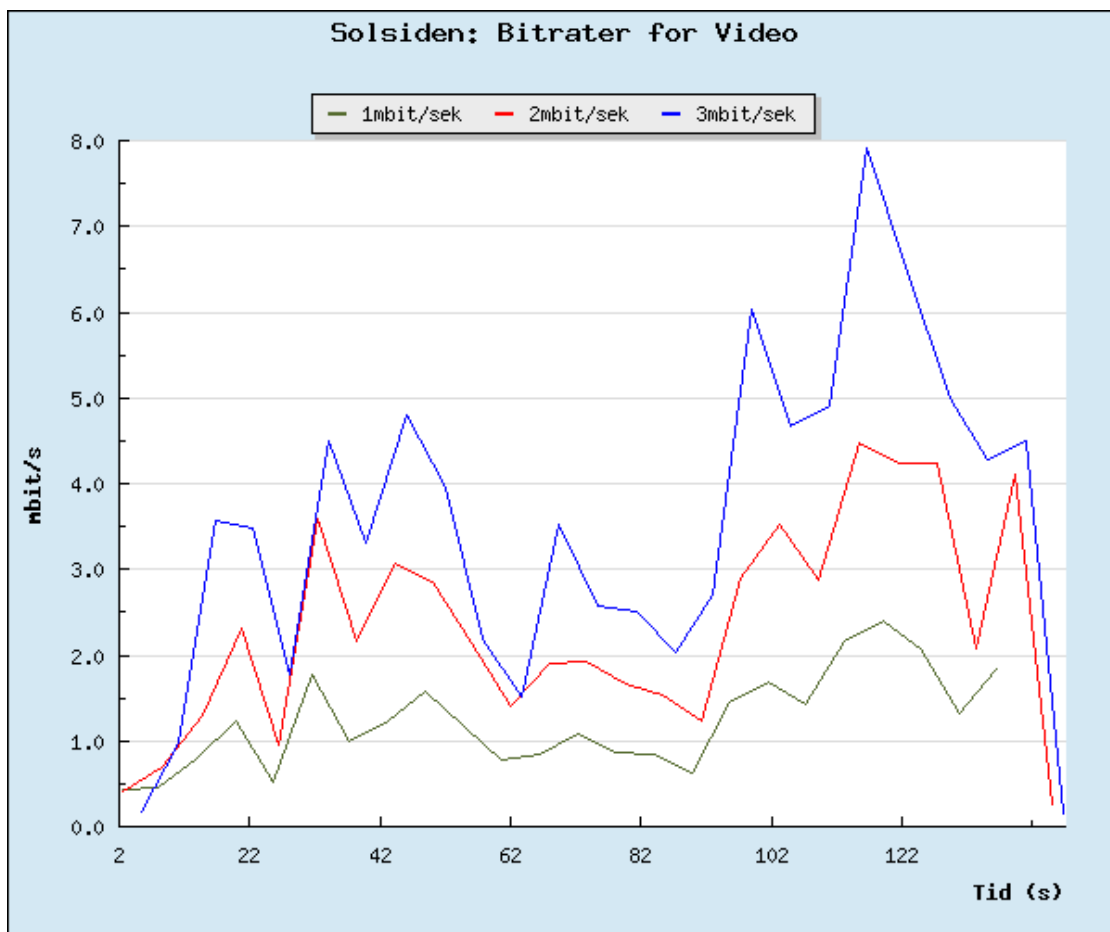
Under testing ble lyd kvaliteten observert ved fysisk å høre på strømmen som ble sendt. Det ble notert hendelser hvor kvaliteten falt merkbart og/eller avbrudd i lyd-avspillingen oppsto. For lydstrømmen gjengitt i det foregående (152kbit/s sendt fra *Solsiden*) oppsto ingen merkbare hendelser før spilletid 153s. I denne tidsregionen viser figur 11.5 et andel pakketap opp mot 20%. Dette medførte avbrudd i lydavspillingen. Tabell 11.1 viser et gjennomsnittlig pakketap for denne strømmen på 2.37%. Dette er en akseptabel verdi og kvaliteten på avspillingen burde ikke påvirkes av dette. Problemet med å basere seg på gjennomsnittsverdier er at unormale topper i verdirekken ikke blir reflektert. I dette tilfellet har strømmen normalt en lav andel tapte pakker, men mister en stor andel rund spilletid 170s. Dette gjør det umulig for dekodere å tolke de resterende pakkene og avbrudd/hakk i avspilling oppstår.

Med tanke på kontribusjon av lyd har NRK satt 1s som krav til maksimum round-trip forsinkelse. I forhold til de ulike formene for lyd-kontribusjon kan man som hovedregel si at en-veis forsinkelsen bør holde seg under 500ms. Tabell 11.1 angir en oversikt over målinger gjort fra alle testlokasjonene, her kan det leses ut at de fleste testen oppfyller NRK sine krav. Gjennomsnittlig forsinkelse på pakkenivå ligger fra 27.14ms - 212.66ms, mens maksimal forsinkelse i de fleste tilfeller ligger under angitte krav på 500ms.

## 11.3 Video

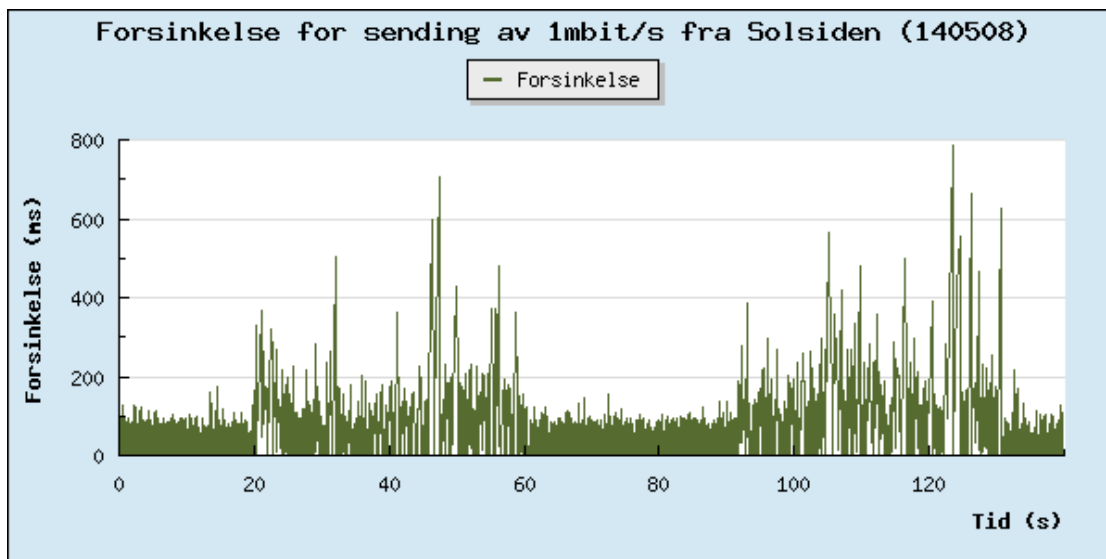
### 11.3.1 Målinger

Testing for sending av video medfører høyere bitrater enn tilsvarende for lyd. Dette fordi mer informasjon må sendes for å kunne spille av ønsket innhold. Resultatene fra samtlige målelokasjoner er relativt like i den sammenheng at jevn avspilling uansett bitrate ikke kunne gjennomføres. Figur 11.6 viser en oversikt over resulterende bitrater ved sending av video fra *Solsiden*.

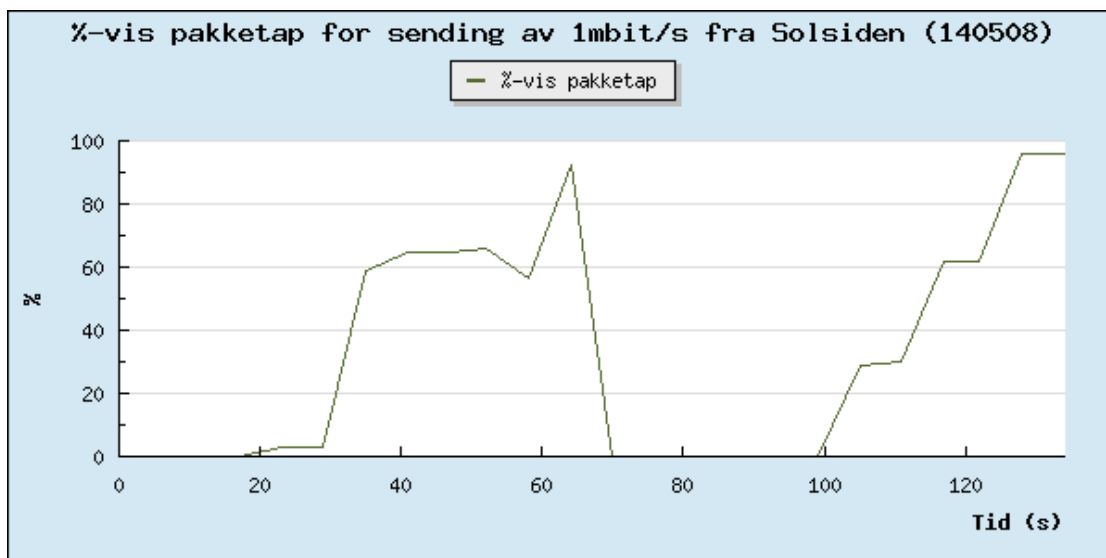


Figur 11.6: Bitrater for sending av video fra Solsiden.

En viktig observasjon er at resulterende bitratene til de respektive strømmene varierer i større grad enn hos lydstrømmene. Hovedgrunnen til dette er at det viste seg vanskelig å transkode kilden slik at bitraten ble konstant. Dette er heller ikke hensiktsmessig. Som nevnt i teorien er det viktig at en strøm som skal kringkastes holder relativt konstant kvalitet sekvensen gjennom. Dette kan kun oppnås ved å implementere variabel bitrate.



Figur 11.7: Forsinkelse ved sending av 1mbit/s videostrøm fra Solsiden.



Figur 11.8: Pakketap ved sending av 1mbit/s videostrøm fra Solsiden.

Sammenlignes figuren for bitrate (figur 11.6) med figurene for henholdsvis forsinkelse (figur 11.7) og pakketap (figur 11.8), ser man en sammenheng lignende den for sending av lyd. Når pakketap oppstår blir forsinkelsen større. Sammenhengen med resulterende fall i bitrate kan ikke åpentlyst sees fra figuren i lys av tidligere diskusjon om variabel bitrate.

Ved oberervasjon av resulterende bitrater i figur 11.6 ser man at disse som forventet utvikler seg på en lignende måte, men på ulike bitrater. Det er likevel interessant å se på avvik fra dette mønsteret. Ved spilletid 125s kan man for strømmen 1mbit/s i de respektive figurene observere et markant pakketap, markant fall i resulterende bitrate og økt forsinkelse.

### 11.3.2 Oppsummering

Dato	Måling	Bitrate <sup>a</sup>	Pakketap <sup>a</sup>	Maks. IPdv	Forsinkelse <sup>a</sup>	Maks. forsinkelse
300408	1mb K <sup>1</sup>	1226.62kb/s	0.30%	284.65ms	10.24ms	294.89ms
140508	1mb S <sup>2</sup>	1234.84kb/s	32.61%	765.68ms	20.67ms	786.35ms
070508	1mb T <sup>3</sup>	1269.78kb/s	11.34%	725.73ms	11.30ms	737.03ms
300408	2mb K <sup>1</sup>	2069.07kb/s	2.71%	491.76ms	5.42ms	497.18ms
070508	2mb T <sup>3</sup>	1547.93kb/s	17.24%	1425.71ms	6.41ms	1432.13ms
300408	3mb K <sup>1</sup>	3543.80kb/s	5.31%	526.40ms	3.98ms	530.38ms
140508	3mb S <sup>2</sup>	3501.82kb/s	70.47%	3393.45ms	399.12ms	3792.57ms
070508	3mb T <sup>3</sup>	3106.90kb/s	43.57%	1603.47ms	7.02ms	1610.49ms

Tabell 11.2: Samling av statistikk for for sending av video.

<sup>1</sup> Kontoret. <sup>2</sup> Solsiden. <sup>3</sup> Torget.

<sup>a</sup> Verdiene angir gjennomsnitt av resultatsettet for denne metrikken.

Generelt er streaming av video med dagens nettverksinfrastruktur så problematisk at det ikke lar seg gjennomføre. Valget av mediefiler og deres koding kan spille en viktig rolle for å bøte på dette. De valgte kodingskonfigurasjonene i denne testen gjør at feilkorrigerende og gjennvinning ved pakketap blir vanskelig. Resulterende andel pakketap gjør det umulig for mediaspillere å vise en jevn strøm med tilnærmet jevn kvalitet. Det vil i diskusjonen bli vurdert mulige løsninger for dette problemet. I de fleste tilfeller resulterer dette i høyere kodet bitrate.

De utførte testene viser et klart trekk at høyere bitrater i økende grad medfører en større ødeleggende effekt for den resulterende strømmen. Tabell 11.2 viser en oversikt fra alle tester gjort for streaming av video fra ulike lokasjoner i TrT. Tester gjort fra *Kontoret* viser ikke uventet de beste verdiene for alle metrikkene. Avspilling av 1mbit/s

strømmen var i hovedsak problemfri, men artefakter kunne spores som vist i figur 11.9. Effekten både pakketap og forsinkelse hadde på visningen vises mest tydelig ved observasjon av figur 11.9(a) sammenlignet med figur 11.9(b), mens figur 11.9(c) sammenlignet med figur 11.9(d) viser klart hvordan artefaktene har sin innvirkning på rending av tekst

Som fremvist i tabell 11.2 ble andel pakketap ved sending av video fra *Solsiden* målt til 70.47%. Denne andelen ble så høy at enhver form for visning ikke kunne gjennomføres. Gjennom hele testsettet ser man en tydelig tendens i at høyere bitrater medfører høyere andel pakketap og en lengre tid for maksimum IPdv.



(a) Streaming fra *Kontoret*.



(b) Snapshot fra original.



(c) Streaming fra *Kontoret*.



(d) Snapshot fra original.

Figur 11.9: Visuell sammenligning av videokvalitet.



# 12

## Diskusjon

### 12.1 Bakgrunn for valg

Følgende seksjon vil diskutere de ulike valgene som har blitt gjort i gjennomføringen av denne oppgaven.

#### 12.1.1 Programvare

Valg av programvare har blitt gjort på bakgrunn av fleksibilitet, tilgjengelighet og informasjonstettheten hos de respektive produsentene. Det ble besluttet å velge programvarer som var godt dokumentert og dermed var mulig å raskt sette seg inn i og få hjelp med ulike problemer via diskusjonsfora på nett. Med unntak av den begrensede bruken av IxChariot<sup>®</sup> er all programvare av typen Open Source. Dette medfører at kildekodene kan lastes ned og i ytterste konsekvens endres. For å implementere spesifikke detaljer i en eventuell løsning er dette fleksibilitetskriteriet viktig. Å velge blant tilnærmet like programvarer er ikke alltid enkelt. Bakgrunnen for de prioriteringene og utvelgelsene som ble gjort var dels begrunnet i omtaler funnet på nett og råd fra faglærer og kollegaer som kunne vise til erfaring innefor testing, men hovedsakling begrunnet i funksjonalitetstester gjort med tanke på utførelse av denne oppgaven.

#### 12.1.2 Testmetodikk

For effektivt å kunne teste de ønskede metrikkene ble ulike metoder tatt i betraktning. I alt tre metoder ble vurdert før en beslutning ble tatt. Listen under angir rekkefølgen

disse metodene ble utprøvd i:

1. Simulering av trafikk via **IxChariot**<sup>®</sup>.
2. Manipulering av kildekoden til **Darwin Streaming Server**.
3. Fangning av sendte pakker via **WireShark**.

Ideelt sett skulle man ønske at den første løsningen som ble prøvd ut fungerte. Dette var som forventet ikke tilfellet. I det følgende vil de ulike metodene, beskrives deres beslag på tid og bakgrunnen for hvorfor de ulike valgene ble gjort.

### **IxChariot**<sup>®</sup>

IxChariot<sup>®</sup> er en velkjent programvare for testing [6]. Denne metoden har blitt brukt i tidligere masteroppgaver, men i disse tilfellene for å teste egenskapene til VoIP og annen taletrafikk [48]. I utgangspunktet er dette en metode som er å foretrekke siden skaperne av IxChariot<sup>®</sup> er velrenomerte og har god erfaring med ulike test-applikasjoner. Ulempen med denne metoden er at IxChariot<sup>®</sup> er betalingsbelagt og det medførte store kostnader å oppdatere programvaren slik at denne kunne brukes. Det viste seg altså at i denne oppgavens sammenheng ville det ikke føre fram å benytte IxChariot<sup>®</sup>. Det var ønskelig å bruke H.264 og AAC som kodek. Disse er ikke støttet gjennom IxChariot<sup>®</sup>. Derfor ble det i starten av oppgaven fokusert på å tilpasse IxChariot<sup>®</sup> ved hjelp av produksjon av ulike skript slik at karakteristiske trekk ved H.264 og AAC kunne gjenskapes. Denne prosessen viste seg tidkrevende og på bakgrunn av begrenset tidligere erfaring med H.264 og AAC og ikke minst produksjon av tilpassingsskript gjennom IxChariot<sup>®</sup> ble det etter hvert besluttet å se bort fra denne løsningen. Hovedkonklusjonen var at H.264 og AAC ikke tilstrekkelig kunne representeres gjennom det arbeidet som ble utført.

### **Darwin Streaming Server**

Det ble forsket på å modifisere kildekoden til DSS slik at den kunne lagre informasjon om alle pakker som ble generert. Derfor ble kildekoden lastet ned fra [15]. Å implementere alle ønskede funksjoner viste seg vanskelig og ble til slutt en for stor utfordring i forhold til programmeringskompetanse og korrekthet i de ekstraherte data. Som med forsøket gjort med IxChariot<sup>®</sup> ble også denne metoden vurdert til å bli for krevende med tanke på tid og kjennskap til programmering. Fordelen med en slik metode er at den impliserer en real-time sending av de fysiske data som på implisitt utviser alle karakteristika for de gitte kodingsregimene. Dette uten behov for å gjøre tilpassinger med tanke på nyttelast og andre spesifikke egenskaper ved kodekene.

## WireShark

Gjennom prosessen med modifisering av kildekoden til DSS ble det vurdert dithen at streaming av fysiske data var den ønskelige metoden for å vurdere ytelsen til kodekene i den distribuerte settingen. Problemet lå i å ekstrahere ønskede målinger fra innhold sendt mellom de respektive nodene. I denne sammenheng ble programvaren WireShark anbefalt av testingeniør Erik Sandnes ved Thales Norway A/S. De utøver utstrakt bruk av denne Open Souce Programvaren i sitt arbeide. Programvaren ble testet ut og det viste seg at denne tar ut konkrete verdier for de fleste ønskede metrikker, verdiene kan videre lagres i en CSV<sup>1</sup>-fil som lar seg importere av de fleste databehandlingsverktøy.

Det ble tidligere nevnt at de fleste verdier kunne ekstraheres ved hjelp av WireShark. En viktig bemerkning her er at variasjon av forsinkelse ikke regnes ut på riktig måte. For å regne ut variasjon i forsinkelse hos RTP-pakker benyttes RTP-timestamps. Disse er et resultat av samplingsfrekvensen til innholdet. WireShark støtter ikke samplingsfrekvensene brukt i de benyttede kodekene. Derfor blir verdien regnet ut av WireShark for variasjon av forsinkelse basert på feil smaplingsfrekvens og resulterende verdi dermed feil. Dette har ingen innvirkning på utregningen av henholdsvis bitrate og forsinkelse da disse baserer seg på tider angitt i IP-headerfeltene. På bakgrunn av denne feilaktige utregningen av IPdv ble utrgening av denne gjort manuelt på pakke-til-pakke basis ut fra den respektive forsinkelsen og i fohold til formel 8.3 gitt under teoriseksjonen metrikker for testing.

## DSS Admin Streaming Server

WireShark henter ut metrikker ved å behandle strømmen pakke for pakke. Dette vil gi liten mening når man skal teste for pakketap. I denne sammenhengen ble det bestemt å bruke DSS sin egen Admin Server. Når DSS er satt opp vil amdminstrasjonsserveren være tilgjengelig i en nettleser (jamnfør fremgangsmåte seksjon 9.3.1). Denne viser ulike metrikker helt ned til 5 sekunders intervaller. Det ble bestemt at denne skulle modifiseres slik at verdiene kunne lagres i en CVS-fil og representeres på lik linje med de andre resultatene.

### 12.1.3 Kodeker

Ved en gjennomgang av litteratur for ulike kodeker og studie av trådløse nettverk ble det lagt vekt på å ta i bruk fleksible kodeker. Dette er en fordel siden et eventuelt system for kontribusjon kun trenger støtte et lite antall kodeker og alle lagrede mediafiler har samme format. De valgte kodekene H.264/AVC for video og AAC for lyd blir i litteraturen beskrevet som ”cutting-edge”. Med dette menes at disse er relativt ferske og implementerer de nyeste teknologiene for komprimering og koding. Dette medfører at

---

<sup>1</sup>Comma-Separated Values

de i nærmeste framtid vil undergå en rekke endringer og utvidelser. For en oversikt over planlagte utvidelser henvises det til [23],[24] for H.264/AVC og for AAC.

I forhold til overføring i trådløse nettverk er det en viktig egenskap at kodekene støtter gjennoppretting ved feil i overføring og har minimale båndbreddekrav i forhold til ønsket kvalitet.

En avgjørende faktor for valget av H.264/AVC for video er nedfelt i ITU sitt overordnede mål for kodeken: *Oppnå en reduksjon i bitrate med en faktor på 2 sammenlignet med andre konkurrerende standarder*. Tester har vist at kodeken tildels innfrir sin målsetting i at den tilbyr tilsvarende oppløsning ved en reduksjon i bitrate på  $\frac{1}{3}$  til  $\frac{1}{2}$  av andre gjeldende standarder (jamnfør teoridel 5.2).

Den største fordelen ved å bytte ut MPEG-2 fra tidligere løsning er også bitrate-avhengig. MPEG-2 bruker omtrent 15-30Mbit/s for koding av HD-video. Til sammenligning oppnår H.264 samme oppløsning med omtrent 8 Mbit/s [39]. I denne kvaliteten kan media brukes til kringkasting. I forhold til overføring i TrT vil denne bitraten foreløpig være en for stor påkjenning, men eksempelet viser et stort potensiale for fremtidige kontribusjons-/kringkastingssystemer.

På bakgrunn av at AAC er tatt i bruk av en rekke kommersielle aktører, falt valget på denne ganske naturlig. Kodeken ble i 2003 omtalt som den mest kraftfulle lydkodeken tilgjengelig [19] og holder fortsatt den påstanden. Andre kodeker brukt i tradisjonelle kontribusjonssystemer ble vurdert å ha for liten grad av fleksibilitet og produserer bitrater som egner seg best til samtalsystemer.

En ulempe ved bruk av både H.264/AVC og MPEG-4 AAC er at utstrakt bruk av komprimering er beregningsintensivt både ved koding og dekodning. I et tilfelle der innspilling skjer ved hjelp av bærbart utstyr vil koding medføre større krav til batterilevetid og ytelses-spesifikasjoner.

## 12.2 Resultater

### 12.2.1 Laststudie 11.1

Laststudiet ble utført for å sette et omtrentlig område for tilbudt bitrate. Dette området ble funnet til å ha en maksverdi på 5mbit/s. Ved minimal krysstrafikk kan TrT i utgangspunktet tilby høyere bitrater. Noe overraskende var det ikke bare bitraten som skapte problemer for systemet, men i tillegg prosesserings-hastigheten til lese- og skrive buffere i ende-til-ende-stien. I utgangspunktet var det tenkt at video-filene skulle ligge på bitrater fra 5-7mbit/s. Dette ville medført at systemet kunne kontribuert innhold i HD til FullHD format. I lys av dette laststudiet ble det besluttet å kutte bitratene til å ligge på et nivå fra 1-3mbit/s. Dette medfører at streaming kun kan foregå i normalt SD-format. I forhold til NRK sin kravsspesifikasjon vil dette være tilstrekkelig, men i framtidsrettede systemer vil dette ikke holde mål. Dog for denne oppgavens hensikt ble

det vurdert at SD kunne virke som et utgangspunkt for så å kunne viderføre studiene til formater av høyere kvalitet gjennom utbedringer av infrastruktur og lingnende beskrevet under seksjonene 12.3 og 13.4.

### 12.2.2 Lyd 11.2 og Video 11.3

En interessant observasjon som omhandler både lyd og video er at gjennomsnittlig forsinkelse synker når bitraten økes. Denne observasjonen vises i tabell 12.1 som er et utdrag fra alle målingene av den aggregerte metrikken gjennomsnittlig forsinkelse gjort fra måleloaksjonen *Kontoret*. Ved testloaksjonene *Solsiden* og *Torget* er det enkeltmålinger som skiller seg ut fra denne tendensen, men i disse isolerte tilfellene blir det konkludert med at det foreligger eksterne forstyrrelser som hindrer trafikken. Det ble sett på to påstander for at dette fenomenet oppstår. Den ene er at en høyere bitrate medfører kortere ventetid mellom to etterfølgende pakker. Det vil si at de etterfølgende pakkene relativt sett kan sendes raskere ut og dermed forsake kortere gjennomsnittsforsinkelse. Den andre påstanden er at ved høyere bitrater vil andelen kollisjoner med pakker fra krysstrafikk minke. Dette vil også føre til en lavere gjennomsnittsforsinkelse for strømmen som helhet.

Dato	Måling	Bitrate <sup>a</sup>	Forsinkelse <sup>a</sup>
06/05/08	59kbit Kontoret	66.91 kbit/s	182.16 ms
06/05/08	152kbit Kontoret	162.73 kbit/s	64.66 ms
06/05/08	283kbit Kontoret	304.66 kbit/s	25.72 ms
30/04/08	1mbit Kontoret	1226.62 kbit/s	10.24 ms
30/04/08	2mbit Kontoret	2069.07 kbit/s	5.42 ms
30/04/08	3mbit Kontoret	3543.80 kbit/s	3.98 ms

Tabell 12.1: Samling av statistikk for alle målinger

<sup>a</sup> Verdiene angir gjennomsnitt av resultatsettet for denne metrikken.

En svakhet ved validiteten av observasjonene gjort for de ulike metrikken er at de baserer seg på et relativt tynt resultatsett. Det har blitt gjort et selektivt utvalg av bitrater og kodeker for å effektivt kunne etterligne en gitt kontribusjonssituasjon. Hadde testene blitt gjenntatt to eller tre ganger ville man i større grad kunne vist til felles utviklingstrekk for de ulike metrikkene. Grunn til at dette ikke ble utført er tidsrammen for fullføring av rapporten. I etterkant kan det diskuteres at det ble brukt for lang tid på å finne fram til en fornuftig testmetodikk. Kandidaten har kanskje hatt for stor grad av tålmodighet og vilje til å tilpasse de respektive løsningene på ønskelig måte før de etter tur ble forkastet. Fordelen med dette har vært en lærerik prosess, mens ulempen i form av tidsbruk har gått på bekostning av testingens omfang.

## 12.3 Mulige løsninger

Denne seksjonen vil skissere mulige løsninger for å utbedre kvalitet for media-streaming generelt og via Trådløse Trondheim spesielt. Løsningene varierer mellom manipulering av media og utbedring av infrastruktur. Innholdet er i stor grad hentet fra litteraturen. Grunnet begrensninger i denne oppgavens omfang vil løsningene kun kort beskrives og leseren hevides derfor til ulike kilder for dyperegående forklaringer.

### 12.3.1 FEC og Error Concealment

FEC og Error Concealment kan brukes til å forebygge uakseptable avbrudd i et begrenset nettverk. Error Concealment dekker over effektene av pakketap ved hjelp av interpolasjon. FEC injiserer redundante pakker i strømmen slik at eventuelle tapte pakker kan rekonstrueres uten behov for retransmisjon. FEC50 legger til 50% mens FEC100 legger til 100%. [41]. Ulempen med disse løsningene er at resulterende bitrate for transport blir høyere. Dette har gjennom rapporten vist seg problematisk. Det henvises til [41] og [57] for en dyperegående introduksjon til disse tema.

### 12.3.2 QualityShaping

*”QualityShaping er et konsept som introduserer en grasiøs degradering av kvalitet når det oppstår overlast i nettverket eller en systemkomponent.”*(Oversatt [40] side 2.). I følge artikkelen baserer metoden seg på å dele innholdet inn i objekter og subobjekter. Når overlast oppstår vil det bli utført en kontrollert kasting av utvalgte sub-objekter(pakker), sceneoppløsningen og kodeparametre blir dertil skalert. Konseptet garanterer en gitt maksimal verdi for ende-til-ende forsinkelse og dermed også en maksimumsverdi for ende-til-ende variasjon av forsinkelse. For videre fordypning i konseptet henvises det til [40].

### 12.3.3 Tilbudt båndbredde

For alle multimediasystemer vil det være fordelaktig å ha så mye tilbudt båndbredde som mulig. Isolert sett vil ikke en økning av tilbudt båndbredde ha stor innvirkning på de ulike metrikkene på grunnlag av trådløse nett sin uberegnelige natur. Felles for de tidligere nevnte løsningene i denne seksjonen er at de fordrer en høyere bitrate enn den originale strømmen. Konklusjonen fra dette resonementet er at økning i tilbudt båndbredde alene ikke vil føre fram, den må settes i samband med andre løsninger for feil-detektering og gjennoppretting eller regimer for tjenestekvalitets-garantier bør innføres.

### 12.3.4 Garantier for tjenestekvalitet

Det finnes ingen systemer pålitlige nok til presist å kunne forutse om det til et gitt tidspunkt vil være overlast i et trådløst nettverk. Det beste man kan gjøre er å gi et grovt estimat. I forhold til direkte programoverføring av TV-sendinger vil det være et behov for tjenestegarantier slik at sendingen kan gjennomføres uten at det oppstår brudd og at forsinkelsen holdes under gitte nivåer. En garanti for tjenestekvalitet kan utstedes for ulike metrikker som forsinkelse, pakketap, IPdv, synkronisert avlevering og gjennomstrømning (throughput). Tabell 12.2 [61] viser en oversikt over protokoller som tilbyr garantier for tjenestekvalitet. Følgende informasjon tar utgangspunkt i Wikipedia sin oversikt for QoS [61].

DiffServ	FrameRealy	IntServ
RSVP	RSVP-TE	ATM
IEEE 802.1p	IEEE 802.11e	IEEE 802.11p
MPLS	TOS-felt i IP-header	

Tabell 12.2: Protokoller som tilbyr tjenestekvalitet (Qos)

I tillegg finnes det metoder for å tilby tjenestekvalitet ved fysisk styring av trafikk i nettverket. Tabell 12.3 [61] lister opp ulike varianter innen trafikkforming, tidsplanlegging og unngåing av metning i nettverket.

Forming	Planlegging	Metning
Token Bucket	Weighted fair Queuing	RED, WRED
Leaky Bucket	Classbased weighted fair Queuing	Trafikkpolitikk
TCP rate controll	Deficit weighted round robin	Explicit congestion notification

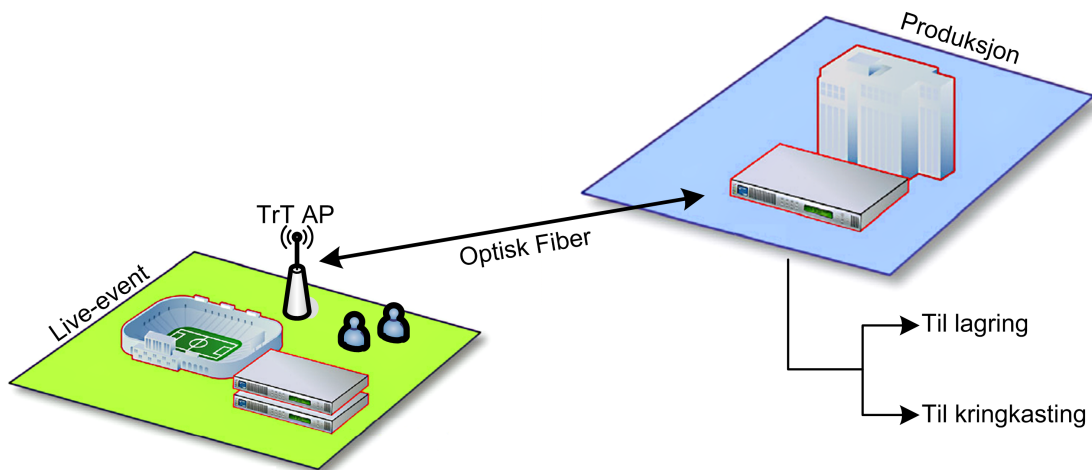
Tabell 12.3: Oversikt over metoder for å styre trafikk i nettverk (QoS).

I tillegg til innføringen Wikipedia gir, henvises det også til [3] for en gjennomgang av temaet QoS i Trådsløse nettverk.

## 12.4 Mulig kontribusjonsløsning

Noen av problemene Kjetil Flatøy ved NRK har uttalt er at Stratalink og til dels AudioTx er *dyre* løsninger og at det for Stratalink kreves et relativt stort støtteapparat for å befordre kontribusjon. I figur 12.1 angis en løsning der en reporter eventuelt støttet av en kameramann kan utføre hele prosessen ved hjelp av opptaksutstyr og en bærbar datamaskin med trådløst nettverkskort. Den mulige løsningen vil ekskludere behovet for retur via mobiltelefon. Tur/retur-interaksjon skjer kun ved hjelp av kommunikasjon via

TrT. Dette er ytterpunktet. I en overgangsfase kan man se for seg en løsning der retur fortsatt skjer via mobiltelefon, men dette vil ikke være hensiktsmessig siden infrastrukturen for tur/retur kommunikasjon via TrT allerede er på plass. Det som mangler er en implementering som fasiliterer dette.



Figur 12.1: Mulig framtidig oppsett for kontribusjon via TrT.



# 13

## Konklusjon

### 13.1 Generelt

Streaming later til å være framtidens system for kontribusjon og kringkasting. I den sammenheng er det viktig å se på ulike kodingssystemer og hvordan innhold kodet av disse oppfører seg sendt via ulike nettveverk og under tilhørende forutsatte og uforutsatte hendelser. Oppgaven så spesielt på kontribusjon av innhold fra ulike lokasjoner innenfor Trådløse Trondheim sitt dekningsområde. Den største utfordringen gjennom den nåværende infrastrukturen har vært fenomenet økt pakketap ved økte bitrater. I så måte har kontribusjon av video vært en utfordring. I forhold til lyd krever video minst 10 ganger så store bitrater. Resulterende pakketap gjorde streaming av video umulig. Det oppsto store mengder artefakter i visning og ved ulike tidsregioner medførte store pakketap også avbrudd i visning. Seksjonen Mulige Løsninger 12.3 skisserer en rekke framgangsmåter for hvordan slik type streaming kan løses. Problemet i så måte er at de fleste av disse løsningene fordrer høyere bitrater, noe som til slutt blir en selvforsterkende effekt. Den løsningen som er mest i øynefallende er å gi garantier for tjenestekvalitet. Denne kan prioritere trafikk i nettverket og dermed unngå forhøyde bitrater.

Gjennom teoristudiet (del I) har et utvalg kodeker både fra eksisterende systemer og nye blitt studert. Visjonen om et framtidensrettet system var avgjørende for at AAC og H.264/AVC ble valgt for henholdsvis lyd og video. Disse standarder er implementert i kjernen til markedsledende produkter og viser attraktive sammenhenger mellom kvalitet og bitrater. I forhold til bruk i trådløse nettverk kan begge de valgte kodekene skaleres

til ulike bitrater med tanke på hendelser i nettverk, og ønskede garantier for sending kan justeres slik at mengden av nettverk og -hendelser kan i møtekommes. Et viktig moment å understreke er at overdimensjonering av nettverksinfrastruktur alene ikke kan bøte på problemer ved levering av media. I forhold til trådløse nettverk sin uberegnelige natur er det også viktig at systemer som brukes kan støtte metodikk for feildetektering og feilgjennoppretting. Isolert sett vil ikke én av løsningene (skissert under seksjon 12.3) i møte komme problemene på en tilfredstillende måte. Dermed er fellesnevneren for alle metoder at de bør utføres i kombinasjon med overprovisjonering av infrastrukturen.

## 13.2 Resultater mot krav fra NRK

For å kunne evaluere et eventuelt kontribusjonssystem med NRK som kringkastingsaktør har det blitt satt visse krav til strømmere. I følge oppgavens problemstilling er metrikken forsinkelse den mest kritiske. For video er det også hensiktsmessig å se på hvilke bitrater et system kan tilby. Tabell 13.1 for lyd og 13.2 for video viser resultatoversikt i forhold til kravene satt fra NRK. For hvert innslag i tabellene gis det en vurdering for om kravet er oppfylt eller ikke. Det er vanskelig å bestemme et konsist ja eller nei i tilfellene for video. Dette fordi akseptert kvalitet på en sending varierer med hensyn på viktigheten i innholdet. For eksempel vil en rykende fersk nyhet, hvis hovedformål er å informere offentligheten, ha kun det kravet at informasjonen kommer gjennom. Mens det i andre tilfeller settes høyere krav til hva som kan vises i en direktesending.

Lyd					
Oppfylt	Type overføring	Kanaler	Forsinkelse		
			En-veis	To-veis	
ja	En-veis	2	500ms	1000ms	
ja	To-veis m/nb <sup>1</sup> retur	2	500ms	1000ms	
ja	To-veis m/bb <sup>2</sup> retur	2	500ms	1000ms	

Tabell 13.1: Oppfyllelse av NRK sine krav til lyd-strømmer.

<sup>1</sup> Narrowband. <sup>2</sup> Bredbånd.

Resultater i forhold til kravene fra tabell 13.1 og 13.2 tilsier at Trådløse Trondheim kan brukes til infrastruktur for kontribusjon av både lyd og video. NRK har bestemt sine krav hovedsaklig med tanke på forsinkelse. Målte verdier for forsinkelse ligger godt under kravene som er gitt. Grunnen til at lyd anbefales og ikke video er resulterende pakketap. Det viste seg at påtrykk av større bitrater medførte større andeler pakketap og for video ble både bitratene så store og pakketapene dertil stort at overføring ikke lot seg gjøre. Dog er det verdt å merke seg at ved sending av video på den laveste

Video						
Oppfylt	Type overføring	Oppløsning i pixler		Lyd-kanaler	Forsinkelse	
		tur	retur		En-veis	To-veis
nei	En-veis	720x576	-	2	500ms	1000ms
nei	To-veis m/nb <sup>1</sup> retur	720x576	-	2	500ms	1000ms
nei	To-veis m/bb <sup>2</sup> retur	720x576	720x576	2	500ms	1000ms

Tabell 13.2: Oppfyllelse av NRK sine krav til video-strømmer.

<sup>1</sup> Narrowband.    <sup>2</sup> Bredbånd.

bitraten (1mbit/s) fra lokasjonen *Kontoret* lot overføringen seg gjennomføre. Denne observasjonen kan medvirke til at fremtidige løsninger har større suksess hvis det for eksempel kan gis garantier for tjenestekvalitet og/eller andre foreslåtte løsninger fra diskusjonen gjennomføres.

### 13.3 Anbefalinger

I forhold til resultater fra denne oppgaven anbefales det NRK og TrT å se videre på et samarbeid angående kontribusjon av lyd og dermed radiosendinger. Resultatene fra kontribusjon av video kunne framvise dårlige metrikker og anbefales dermed ikke. I lys av at ny infrastruktur for TrT, i form av WiMAX, testes ut i nær framtid vil det samtidig anbefales at dette eller lignende studier gjenbesøkes slik at målinger og utprøving kan gjøres under nye forutsetninger.

### 13.4 Videre arbeid

Denne oppgaven har sett på mulighetene for å sende lyd og/eller video med Trådløse Trondheim som transportnett. Konfigurasjonen for testingen har tilstrebet å etterligne et tradisjonelt kontribusjonssystem med NRK som en mulig kringkastingsaktør. Det har ikke vært fokus på å sammenligne spesifikk ytelse i TrT for ulike kodingssystemer. Et eventuelt tema for videre forskning kan dreie seg om å sammenligne ulike nivåer/profiler hos de to kodekene som allerede er utprøvd, eller gjøre en studie som sammenligner ytelsen til ulike kodeker. I forhold til TrT sitt framtidige samarbeid med NRK kan det være interessant å sette opp et reelt kontribusjonssystem der innhold blir innspilt og sendt i real-time. Denne problemstillingen dreier seg per dato om kontribusjon av lyd, men kan også, gitt nevnte utbedringer, omhandle video. En eventuell implementasjon av et slikt system bør tilstrebe enkelhet i den grad at det direkte kan brukes av NRK.



## Bibliografi

- [1] G. Almes, S. Kalidindi, and M. Zekauskas. A one-way delay metric for ippm.  
<http://www.ietf.org/rfc/rfc2679.txt>, 1999.
- [2] G. Almes, S. Kalidindi, and M. Zekauskas. A one-way packet loss metric for ippm.  
<http://www.ietf.org/rfc/rfc2680.txt>, 1999.
- [3] Vittorio Biló, Adriano Di Pasquale, Fabio Fioravanti, Michele Flammini, Luca Forlizzi, Francesco Lo Presti, Giovanna Melideo, Enrico Nardelli, Alfredo Navarra, and Guido Proietti. Quality of service in wireless networks.  
<http://ieeexplore.ieee.org/iel5/8608/27277/01213411.pdf?tp=&arnumber=1213411&isnumber=27277>, 2003.
- [4] S. Bradner. Benchmarking terminology for network interconnection devices.  
<http://www.ietf.org/rfc/rfc1242.txt>, July 1991.
- [5] Atheros Communications. Ar5005gs.  
<http://www.atheros.com/pt/AR5005GS.htm>, 2008.
- [6] Ixia Communications. Ixchariot.  
<http://www.ixiacom.com/products/display?skey=ixchariot>, Mai 2008.
- [7] Demichelis and Chimento. Rfc3390.  
<http://tools.ietf.org/html/rfc3393>, 2002.
- [8] M. Handley and V. Jacobson. Sdp: Session description protocol.  
<http://www.ietf.org/rfc/rfc2327.txt>, 1998.
- [9] IEEE. Ieee 802.11 wireless local area network (wlan) working group.  
<http://ieee802.org/11>, 2006.
- [10] Fraunhofer Institute for Integrated Circuits IIS. Mpeg-4 audio encoders and decoders on various platforms.  
[http://www.iis.fraunhofer.de/fhg/Images/Fraunhofer\\_IISMpeg4AudioSoftware\\_1\\_6\\_tcm278-67522.pdf](http://www.iis.fraunhofer.de/fhg/Images/Fraunhofer_IISMpeg4AudioSoftware_1_6_tcm278-67522.pdf), 2007.

- [11] Airlinx Communications Inc;. Audio tx communicator.  
<http://www.airlinx.com/products.cfm/product/6-154-484.htm>, Mai 2008.
- [12] Apple Computer Inc;. Qtss dss administrators guide.  
[http://developer.apple.com/opensource/server/streaming/qtss\\_admin\\_guide.pdf](http://developer.apple.com/opensource/server/streaming/qtss_admin_guide.pdf), 2002.
- [13] Apple Computer Inc;. Aac audio.  
<http://www.apple.com/quicktime/technologies/aac/>, 2007.
- [14] Apple Computer Inc;. Quicktime file format specification.  
<http://developer.apple.com/documentation/QuickTime/QTFF/qtff.pdf>, 2007.
- [15] Apple Computer Inc;. Darwin releases.  
<http://www.opensource.apple.com/darwinsource/>, Februar 2008.
- [16] Cisco Systems Inc;. Understanding the lightweight access point protocol point protocol (lwapp).  
[http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps6306/prod\\_white\\_paper0900aecd802c18ee.pdf](http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps6306/prod_white_paper0900aecd802c18ee.pdf), 2005.
- [17] Cisco Systems Inc;. Cisco wireless lan controllers.  
[http://www.cisco.com/en/US/prod/collateral/wireless/ps6302/ps8322/ps6307/product\\_data\\_sheet0900aecd802570b0.pdf](http://www.cisco.com/en/US/prod/collateral/wireless/ps6302/ps8322/ps6307/product_data_sheet0900aecd802570b0.pdf), 2007.
- [18] Cisco Systems Inc;. Cisco 4400 series wireless lan controllers - introduction.  
<http://www.cisco.com/en/US/products/ps6366/index.html>, 2008.
- [19] MPEG-4 IndustyForum. Aac.  
<http://www.m4if.org/public/documents/vault/m4-out-30036.zip>, April 2003.
- [20] ISO/IEC. Information technology generic coding of moving picture and associated audio information part 7: Advanced audio coding (aac), 2006.
- [21] ITU-T. H.264 (05/03).  
<http://www.itu.int/rec/T-REC-H.264/recommendation.asp?lang=en&parent=T-REC-H.264-200305-S>, 2003.
- [22] ITU-T. Recommendation h.264 (2005) amendment 2 (04/07).  
<http://www.itu.int/rec/T-REC-H.264/recommendation.asp?lang=en&parent=T-REC-H.264-200704-T!Amd2>, 2007.

- [23] ITU-T. H.264 : Advanced video coding for generic audiovisual services.  
url<http://www.itu.int/rec/T-REC-H.264/en>, Mai 2008.
- [24] ITU-T. Itu-t newslog.  
<http://www.itu.int/ITU-T/newslog/Video+Coding+Work+Progressed+In+April+Meetings.aspx>, Mai 2008.
- [25] Nugehally Jayant and Susan Christensen. Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure.  
<http://ieeexplore.ieee.org/iel5/8159/23935/01094975.pdf>, 1981.
- [26] Thomas Jelle. Trådløse trondheim. *Fakultet for informasjonsteknologi matematikk og elektroteknikk*, 2006.
- [27] Hari Kalva. Issues in h.264/mpeg-2 video transcoding.  
<http://ieeexplore.ieee.org/iel5/9037/28685/01286946.pdf>, 2004.
- [28] Jari Korhonen and Ye Wang. Schemes for error resilient streaming of perceptually coded audio.  
<http://ieeexplore.ieee.org/iel5/8535/26996/01200077.pdf?arnumber=1200077>, 2003.
- [29] Vassilios Koukoulidis and Mehul Shah. The ip multimedia domain: service architecture for the delivery of voice, data, and next generation multimedia applications.  
[http://www.item.ntnu.no/fag/ttm5/papers\\_2007/IP-MultiMedia-Domain.pdf](http://www.item.ntnu.no/fag/ttm5/papers_2007/IP-MultiMedia-Domain.pdf), 2006.
- [30] Ulf Lamping, Richard Sharpe, and Ed Warnicke. *Wireshark User's Guide*. NS Computer Software and Services P/L, 25283 for wireshark 1.0.0 edition, 2008.
- [31] Marpe, Wiegand, and Gordon. H.264/mpeg4-avc fidelity range extensions: Tools, profiles, performance, and application areas.  
<http://ieeexplore.ieee.org/iel5/10242/32660/01529820.pdf?arnumber=1529820>, 2005.
- [32] Enzo Michelangeli. Itu g.711.  
<http://www.voip-info.org/wiki-ITU+G.711>, 2005.
- [33] MRCBroadcast. Strata 2ghz transmitter.  
url[http://www.mrcbroadcast.com/broadcast/portable/strata/strata\\_2\\_ghz\\_transmitter](http://www.mrcbroadcast.com/broadcast/portable/strata/strata_2_ghz_transmitter), Mai 2008.
- [34] NACIP. Audio contribution over ip.  
[http://www.ebu.ch/CMSimages/en/tec\\_doc\\_t3326-2008\\_tcm6-54427.pdf](http://www.ebu.ch/CMSimages/en/tec_doc_t3326-2008_tcm6-54427.pdf), 2007.

- [35] Trådløse Trondheim (nettside);. Trådløse trondheim.  
<http://www.tradlosetrondheim.no/>, 2008.
- [36] France Nextream. Sd and hd video contribution over ip/mps networks.  
[http://www.thomsongrassvalley.com/docs/Application\\_Notes/transmission/vibe/VCN-6002\\_AN\\_SD\\_and\\_HD\\_Video\\_Contribution.pdf](http://www.thomsongrassvalley.com/docs/Application_Notes/transmission/vibe/VCN-6002_AN_SD_and_HD_Video_Contribution.pdf), 2004.
- [37] Kultur og Utdanningsdepartementet;. Stortingsmelding nr.6.  
<http://www.regjeringen.no/nb/dep/kkd/dok/regpubl/stmeld/2007-2008/Stmeld-nr-6-2007-2008-/9.html?id=489034>, 2008.
- [38] Othman, , Ng, and L. Hanzo. Turbo-detected unequal protection mpeg-4 audio transceiver using convolutional codes, trellis coded modulation and space-time trellis coding.  
<http://ieeexplore.ieee.org/iel5/10360/32960/01543590.pdf?tp=&arnumber=1543590&isnumber=32960>, 2005.
- [39] PixelTools. H.264 advanced video coding: A whirlwind tour.  
[http://www.pixeltools.com/h264\\_paper.html](http://www.pixeltools.com/h264_paper.html), 2003.
- [40] Leif Arne Rønningen. A protocol stack for futuristic multimedia.  
<http://www.item.ntnu.no/fag/ttm4140/host06/ICSPCS07-1.pdf>, 2007.
- [41] Pierre Robidoux. Ip audio codecs for radio broadcasting.  
<http://www.cbc.radio-canada.ca/technologyreview/pdf/issue5-codec.pdf>, 2008.
- [42] Sanneck. Packet loss recovery and control for voice transmission over the...
- [43] H. Sanneck, A. Stenger, K. Younes, and B. Girod. A new technique for audio packet loss concealment, 1996.
- [44] P.A. Sarginson. Mpeg-2 : Overview of the systems layer. *Research and Development Report*, 1996.
- [45] H. Schulzrinne and S. Casner. Rtp profile for audio and video conferences with minimal control.  
<http://www.ietf.org/rfc/rfc3551.txt>, 2003.
- [46] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications.  
<http://www.ietf.org/rfc/rfc3550.txt>, 2003.
- [47] H. Schulzrinne, A. Rao, and R. Lanphier. Real time streaming protocol (rtsp).  
<http://www.ietf.org/rfc/rfc2326.txt>, 1998.



- [48] Petter Stray. Field test of mobile terminals in a wireless city. Master's thesis, NTNU, 2007.
- [49] TechTerms.com. Bitrate.  
<http://www.techterms.com/definition/bitrate>, Mai 2008.
- [50] Trådløse Trondheim. Wireless trondheim - a solution overview. -, 2007.
- [51] P.N. Tudor. Mpeg-2 video compression. *Electronics and Communication Engineering Journal*, 1995.
- [52] Trond Ulseth and Finn Stafnsnes. Voip speech quality - better than pstn? *Teletonikk 1.2006*, 2006.
- [53] Geert Van der Auwera, Prasanth T. David, and Martin Reisslein. Traffic characteristics of h.264/avc variable bit rate video.  
[http://www.fulton.asu.edu/~mre/COMMAG\\_06\\_00562\\_revised.pdf](http://www.fulton.asu.edu/~mre/COMMAG_06_00562_revised.pdf), 2006.
- [54] Ltd. VOCAL Technologies. G.729 speech coder.  
[http://www.vocal.com/data\\_sheets/g729.html](http://www.vocal.com/data_sheets/g729.html), 2007.
- [55] B. Wah and D. Lin. Lsp-based multiple-description coding for real-time low bit-rate voice transmissions, 2002.
- [56] Theodore Wallingford. *Switching to VoIP*. O'Reilly Media, 2005.
- [57] Charles Wang, Dean Sklar, and Diana Johnson. Forward error-correction coding.  
<http://www.aero.org/publications/crosslink/winter2002/04.html>, 07 05.
- [58] Thomas Wiegand and Gary Sullivan. Draft itu-t recommendation and final draft international standard of jvs(itu-t rec. h.264 iso/iec 14496-10 avc).  
<http://www.ncesd.org/vc/docs/JVT-G050.pdf>, 2003.
- [59] Thomas Wiegand, Gary Sullivan, Gisle Bjøntegaard, and Ajay Luthra. Overview of the h.264/avc video coding standard.  
[http://ip.hhi.de/imagecom\\_G1/assets/pdfs/csvt\\_overview\\_0305.pdf](http://ip.hhi.de/imagecom_G1/assets/pdfs/csvt_overview_0305.pdf), 2003.
- [60] Wikipedia. Packet delay variation.  
[http://en.wikipedia.org/wiki/Packet\\_delay\\_variation](http://en.wikipedia.org/wiki/Packet_delay_variation), Mai 2008.
- [61] Wikipedia. Quality of service.  
[http://en.wikipedia.org/wiki/Quality\\_of\\_service](http://en.wikipedia.org/wiki/Quality_of_service), 9. Juni 2008.

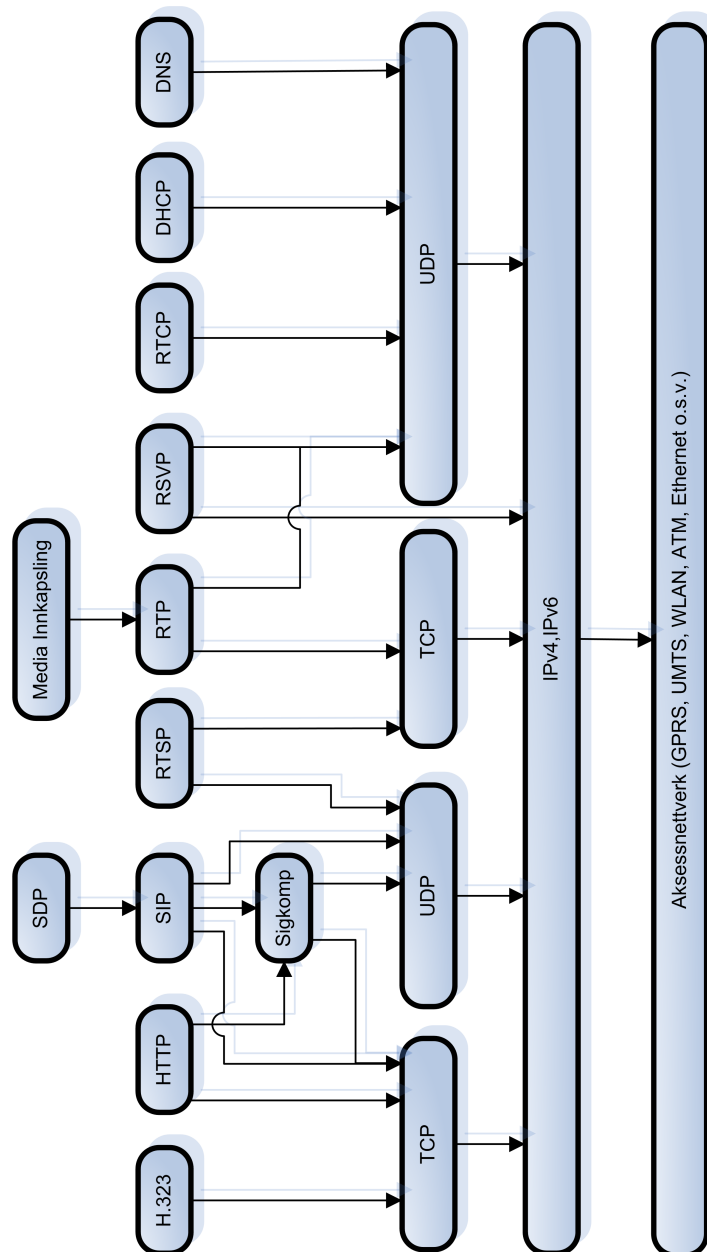


# Vedlegg





# Protokoll-stakker



Figur A.1: Oversikt over protokollstakker.

## Nivåer i H.264

Nivå	Maks macro-blokker pr. sek	Maks ramme-størrelse	Maksimum video bit-rate			Oppløsning @ rammerate	
			BP, MP XP	High Profile	High 10 Predictive Profile		
2	11880	396	2 Mb/s	2.5 Mb/s	6 Mb/s	8 Mb/s	320x240@36.0 (7) 352x288@30.0 (6)
2.1	19800	792	4 Mb/s	5 Mb/s	12 Mb/s	16 Mb/s	352x480@30.0 (7) 352x576@25.0 (6)
2.2	20250	1620	4 Mb/s	5 Mb/s	12 Mb/s	16 Mb/s	352x480@30.7(10) 352x576@25.6 (7) 720x480@15.0 (6) 720x576@12.5 (5)
3	40500	1620	10 Mb/s	12.5 Mb/s	30 Mb/s	40 Mb/s	352x480@61.4 (12) 352x576@51.1 (10) 720x480@30.0 (6) 720x576@25.0 (5)
3.1	108000	3600	14 Mb/s	17.5 Mb/s	42 Mb/s	56 Mb/s	720x480@80.0 (13) 720x576@66.7 (11) 1280x720@30.0 (5)
3.2	216000	5120	20 Mb/s	25 Mb/s	60 Mb/s	80 Mb/s	1280x720@60.0 (5) 1280x1024@42.2 (4)
4	245760	8192	20 Mb/s	25 Mb/s	60 Mb/s	80 Mb/s	1280x720@68.3 (9) 1920x1088@30.1 (4) 2048x1024@30.0 (4)

Tabell A.1: Nivåer i H.264



B

Kildekode



## Kilde: index.php

```

<?PHP
/*
Skriptets hovedmetode. Denne bruker to filer: read_index.php og graf.php.
read_index.php:
    Tar inn en loggfil og legger informasjonen inn i en tilhørende MySQL-database.
graf.php:
    Lager grafer fra tilhørende databaser
    Tar to argumenter:
        $name - navnet til databasen
        $db - hvilken type graf skal lages
*/

include ( "read_file.php" );

echo " < center >" ;
/*
Åpner mappen med logfilene og leser hvilke filer som ligger der.
*/
if ( $handle = opendir ( 'logs' ) ) {
    $check = false ;
    while ( false !== ( $file = readdir ( $handle ) ) ) {
        $db = "" ;
        $name = "" ;
        $name = explode ( "." , $file );
        $db = explode ( "_" , $file );

        if ( ( $file == "." ) or ( $file == ".." ) ) {
        }
        else {

            //Inkluderer en fil som kobler seg til databasen.
            include ( "db_conn.inc" );
            $query = "showtableslike" . $name [ 0 ] . "" ;
            $result = @mysql_query ( $query ) OR die ( 'Får ikke kontakt med serveren!' );

            //Lukker databasetilkoblingen
            include ( "db_disconn.inc" );

            //Hvis databasen ikke finnes lages den.
            if ( @ mysql_fetch_assoc ( $result ) == "" ) {
                if ( $db [ 0 ] == "ws" ) {
                    makeDbWs ( $name [ 0 ] );
                }
                else if ( $db [ 0 ] == "ds" ) {
                    makeDbDss ( $name [ 0 ] );
                }
            }
        }
    }
}

```

```

$check = true ;

//Skriver grafene til browseren.
if ( $db [ 0 ]== "ws" ){
echo " <hr>Wiresharkdelay-" . $file . " <br/><br/>" ;
echo '<br/><br/>';
echo " <hr>" ;
}
else if ( $db [ 0 ]== "ds" ){
echo " <hr>DSSbandwidth-" . $file . " <br/><br/>" ;
echo '<imgsrc =" graf.php?db =' . $name [ 0 ]. ' &y = bw"border = 0align = center ><br/><br/>'
;
echo " <hr>" ;
echo " <hr> DSSpacketloss -" . $file . " <br/><br/>" ;
echo ' <imgsrc =" graf.php?db =' . $name [ 0 ]. ' &y = pl"border = 0align = center ><br/><br/>'
;
echo " <hr>" ;
}
}
}
}

closedir ( $handle );
if (! $check ){
echo "Mappen/logs/ertom." ;
}
}
else echo "Mappenlogskanikkeaksesseres." ;
echo " </center >" ;
? >

```

## Kilde: readfile.php

```

<?PHP
/*
Funksjonen inkluderes i hovedmetoden index.php
Tar inn URL til en loggfil og skriver informasjonen inn i en MySQL-database.
To funksjoner trengs for to ulike loggfil-formater.
*/

function makeDbWs ( $filename ){
    //Åpner filen
    $lines = file ( "logs/" . $filename . ".csv" );
    //Åpner tilkobling mot databasen
    include ( "db_conn.inc" );
    //Lager Databasen
    $create_query = "
CREATE TABLE ' . $filename . "' (
    'packet' bigint(20) NOT NULL default '0',
    'sequence' text NOT NULL,
    'delta_ms' text NOT NULL,
    'jitter_ms' text NOT NULL,
    'ip_bw' text NOT NULL,
    'marker' text NOT NULL,
    'status' text NOT NULL,
    'time' text NOT NULL,
    'length' text NOT NULL,
    PRIMARY KEY ('packet')
) TYPE=MyISAM;
" ;
    @mysql_query ( $create_query ) OR die ( 'Kan ikke lage tabellen: ' . $filename );
    //Populerer databasen
    foreach ( $lines as $line ) {
        $sql_input = explode ( ",", $line );
        //Benevning på verdiene i $sql_input: Packet,Sequence,Delta (ms),Jitter (ms),IP BW (kbps),Marker,Status,Date,Length
        if ( $sql_input [ 0 ] != 0 ) {
            $query = "INSERT into " . $filename . " values(
                " . $sql_input [ 0 ] . ",
                " . $sql_input [ 1 ] . ",
                " . $sql_input [ 2 ] . ",
                " . $sql_input [ 3 ] . ",
                " . $sql_input [ 4 ] . ",
                " . $sql_input [ 5 ] . ",
                " . $sql_input [ 6 ] . ",
                " . $sql_input [ 7 ] . ",
                " . $sql_input [ 8 ] . ");" ;
            //Legger verdiene inn i databasen.
            @mysql_query ( $query ) OR die ( 'Kan ikke legge inn pakke # ' . $sql_input [ 0 ] );
        }
    }
}

```

```

}
//Lukker tilkoblinga mot databasen
include ( "db_disconn.inc" );
}
function makeDbDss ( $filename ){
//Åpner filen
$lines = file ( "logs/" . $filename . ".log" );
//Åpner tilkobling mot databasen
include ( "db_conn.inc" );
//Lager Databasen
$create_query = "
CREATE TABLE ' " . $filename . "' (
'id' BIGINT NOT NULL AUTO_INCREMENT ,
'qtssCliSesPresentationURL' TEXT NOT NULL ,
'time' BIGINT NOT NULL ,
'qtssCliSesPacketLossPercent' TEXT NOT NULL ,
'qtssCliSesRTPBytesSent' TEXT NOT NULL ,
'qtssCliSesCurrentBitRate' TEXT NOT NULL ,
'qtssCliRTSPSessRemoteAddrStr' TEXT NOT NULL ,
PRIMARY KEY ( 'id' ));" ;
@mysql_query ( $create_query ) OR die ( 'Kan ikke lage tabellen: ' . $filename );
//Populerer databasen
foreach ( $lines as $line ) {
$sql_input = explode ( "," , $line );
//if($sql_input[0] != 0) {
$query = "INSERT into " . $filename . " (
'qtssCliSesPresentationURL',
'time',
'qtssCliSesPacketLossPercent',
'qtssCliSesRTPBytesSent',
'qtssCliSesCurrentBitRate',
'qtssCliRTSPSessRemoteAddrStr')
values(
" . $sql_input [ 1 ] . ",
" . $sql_input [ 2 ] . ",
" . $sql_input [ 3 ] . ",
" . $sql_input [ 4 ] . ",
" . $sql_input [ 5 ] . ",
" . $sql_input [ 6 ] . ");" ;
@mysql_query ( $query ) OR die ( 'Kan ikke legge inn pakke #' . $query );
}
//Lukker tilkoblinga mot databasen
include ( "db_disconn.inc" );
}
?>

```

## Kilde: graf.php

```

<?PHP
/*
Dette scriptet leser fra en database og lager grafer.
Verdiene i $_REQUEST[] settes av hovedmetoden index.php
JPGraph: Gratisbibliotek som lastes ned eksternt.
*/
require_once ( 'jpgraph/src/jpgraph.php' );
require_once ( 'jpgraph/src/jpgraph_bar.php' );
require_once ( 'jpgraph/src/jpgraph_line.php' );

$req = explode ( "_" , $_REQUEST [ 'db' ] );

//Inkluderer en fil som kobler seg til databasen.
include ( "db_conn.inc" );
$query = "SELECT * FROM '" . $_REQUEST [ 'db' ] . "' ORDER BY 'time' ASC;" ;
$result = @mysql_query ( $query ) OR die ( '(Query)' );

/*
Lesing av database avhengig av hvilken type graf som skal lages. Parametre sendes fra hovedmetoden i index.php.
*/

if ( $req [ 0 ] == "ws" ){
    $query1 = "SELECT time FROM '" . $_REQUEST [ 'db' ] . "' ORDER BY 'time' ASC LIMIT 0 , 1;" ;
    $result1 = @mysql_query ( $query1 ) OR die ( '(Query 1)' );
    $sqldate = @mysql_fetch_assoc ( $result1 );
    $refdate = explode ( ":" , $sqldate [ 'time' ] );

    while ( $row = @mysql_fetch_assoc ( $result )){
        $date = explode ( ":" , $row [ 'time' ] );
        //Finner referansetid for første pakke sendt.
        if ( $refdate [ 1 ] == $date [ 1 ]){
            $values [ 'x' ][] = $date [ 2 ] - $refdate [ 2 ];
        }
        else {
            $values [ 'x' ][] = (( $date [ 1 ] - $refdate [ 1 ] ) * 60 ) + $date [ 2 ]
] - $refdate [ 2 ];
        }
        $values [ 'y' ][] = $row [ 'delta_ms' ];
    }
}

if ( $req [ 0 ] == "ds" ){
    $query1 = "SELECT time FROM '" . $_REQUEST [ 'db' ] . "' LIMIT 0 , 1;" ;
    $result1 = @mysql_query ( $query1 ) OR die ( '(Query 1)' );
    $reftime = @mysql_fetch_assoc ( $result1 );
    while ( $row = @mysql_fetch_assoc ( $result )){
        if ( $_REQUEST [ 'y' ] == "bw" ){
            $values [ 'y' ][] = $row [ 'qtssCliSesCurrentBitRate' ] / 1000000 ;
        }
    }
}

```

```

else if ( $_REQUEST [ 'y' ]== "pl" ){
    $values [ 'y' ][] = $row [ 'qtssCliSesPacketLossPercent' ];
}
$values [ 'x' ][] = round ( ( ( int ) $row [ 'time' ] - $reftime [ 'time' ] ) /
1000 , 0 );
}
}

//Lukker tilkoblingen mot databasen
include ( "db_disconn.inc" );
//Lager graf med biblioteket JGGraph
$graph = new graph ( 600 , 400 , "png" , 0 , true );
$graph -> img -> SetMargin ( 60 , 30 , 70 , 60 );
$graph -> SetScale ( 'intlin' , 0 , 0 , min ( $values [ 'x' ] ) , max ( $values [ 'x' ] ) );
$line1 = new LinePlot ( $values [ 'y' ] , $values [ 'x' ] );
$line1 -> SetColor ( 'darkolivegreen' );
$line1 -> SetWeight ( 1 );
$line1 -> SetCenter ( );
$line1 -> SetLegend ( getLegend ( ) );
$graph -> Add ( $line1 );

$graph -> legend -> SetLayout ( LEGEND_HOR );
$graph -> legend -> Pos ( 0.5 , 0.09 , 'center' );

$graph -> title -> Set ( getHeader ( ) );
$graph -> title -> SetFont ( FF_FONT2 , FS_BOLD , 50 );

$graph -> yaxis -> SetLabelAngle ( 0 );
$graph -> yaxis -> title -> Set ( 'Forsinkelse (ms)' );
$graph -> yaxis -> title -> SetFont ( FF_FONT1 , FS_BOLD );
$graph -> yaxis -> title_margin = 40 ;

$graph -> xaxis -> SetLabelAngle ( 0 );
$graph -> xaxis -> title -> Set ( 'Tid (s)' );
$graph -> xaxis -> title -> SetFont ( FF_FONT1 , FS_BOLD );
$graph -> xaxis -> title_margin = 15 ;

$graph -> Stroke ( );

//Funksjon for å bestemme tittel på grafen.
function getHeader (){
    $array = explode ( "_" , $_REQUEST [ "db" ] );
    $return = "" ;
    if ( $array [ 0 ]== "ws" ){
        $return = "Forsinkelse" ;
    }
    else if ( $array [ 0 ]== "ds" ){
        if ( $_REQUEST [ "y" ]== "bw" ){
            $return = "Bitrate" ;
        }
        else if ( $_REQUEST [ "y" ]== "pl" ){

```



```

    $return = "%-vis pakketap" ;
}
}
$return .= " for sendnig av " . $array [ 1 ]. "/s fra " . $array [ 2 ]. " (" . $array [ 3 ].
)" " ;
//Returnerer tittel
return $return ;
}
//Funksjon for å bestemme benevning på grafen.
function getLegend (){
$array = explode ( "_" , $_REQUEST [ "db" ] );
$return = "" ;
if ( $array [ 0 ]== "ws" ){
$return = "Forsinkelse" ;
}
else if ( $array [ 0 ]== "ds" ){
if ( $_REQUEST [ "y" ]== "bw" ){
$return = "Bitrate" ;
}
else if ( $_REQUEST [ "y" ]== "pl" ){
$return = "%-vis pakketap" ;
}
}
}
//Returnerer benevning
return $return ;
}

```

?>

## Kilde mod.pl

```
#Modifikasjon av administrasjons-skript for Darwin Streaming Server

#Navn på loggfilen som skal lages
my $file = 'test_H264.log' ;

#Åpne/Lage loggfil
open FILE , '>>$file' or die 'unable to open $file $!' ;
#Populerer loggfil
foreach ( @valuesArr ) {
    print FILE '$_,' ;
}
print FILE '$connType\n' ;
#Lukker loggfil
close FILE , '>>$file' ;
```

C

Atheros AR5005G

AR5005GS Solution Specifications	
Frequency Band	2.300 to 2.500 GHz
Network Standard	802.11b, 802.11g
Modulation Technology	OFDM with BPSK, QPSK, 16 QAM, 64 QAM; DBPSK, DQPSK, CCK
FEC Coding Rates	1/2, 1/3, 1/4
Hardware Encryption	AES, TKIP, WEP
Quality of Service	802.11e draft
Media Access Technique	CSMA/CA
Host Interface	Mini PCI, PC Card, PCI
Communication Interface	High Speed UART
Peripheral Interface	GPIOs, LEDs
Memory Interface	EEPROM
	Supported Data Rates
IEEE 802.11b	1 to 11 Mbps
IEEE 802.11g	1 to 54 Mbps
Atheros Super G	mode Up to 108 Mbps
Chip Specifications	AR2414
Operating Voltage	1.8V +/-5% 3.3V +/-10%
Package Dimensions	13mm x 13mm
Packaging	224 Plastic Ball Grid Array

Tabell C.1: Spesifikasjon Atheros AR5005G [5]

D

Mediakarakteristiker

## Video

Generelt	H.264 Video
Navn	<b>1mb.mp4</b>
Format	MPEG-4
Format/Info	ISO 14496-1 Base Media
Format/Familie	MPEG-4
Filstørrelse	18.9 MiB
Lengde	2mn 23s
Bitrate	1103 Kbps
Strømstørrelse	989 KiB
Video	
Kodek	H.264
Kodek/Info	H.264 (3GPP)
Lengde	2mn 22s
Bitrate	968 Kbps
Nominal Bitrate	995 Kbps
Bredde	1280 pixels
Høyde	544 pixels
Display Aspect ratio	2.35
Rammerate	23.976 fps
Strømstørrelse	16.5 MiB
Audio	
Kodek	AAC LC-SBR
Kodek/Info	AAC Low Complexity with Spectral Band Replication
Lengde	2mn 22s
Bitrate	85 Kbps
Bitrate -modus	VBR
Kanal(er)	2 channels
Kanalplassering	L R
Samplingsrate	48 KHz
Oppløsning	16 bits
Strømstørrelse	1.44 MiB

Tabell D.1: Media-karakteristiker H.264 1mbit/s.

<b>Generelt</b>	<b>H.264 Video</b>
Navn	<b>2mb.mp4</b>
Format	MPEG-4
Format/Info	ISO 14496-1 Base Media
Format/Familie	MPEG-4
Filstørrelse	37.6 MiB
Lengde	2mn 23s
Bitrate	2198 Kbps
Strømstørrelse	1.71 MiB
<b>Video</b>	
Kodek	H.264
Kodek/Info	H.264 (3GPP)
Lengde	2mn 22s
Bitrate	2025 Kbps
Nominal Bitrate	2068 Kbps
Bredde	1280 pixels
Høyde	544 pixels
Display Aspect ratio	2.35
Rammerate	23.976 fps
Strømstørrelse	34.4 MiB
<b>Audio</b>	
Kodek	AAC LC-SBR
Kodek/Info	AAC Low Complexity with Spectral Band Replication
Lengde	2mn 22s
Bitrate	85 Kbps
Bitrate -modus	VBR
Kanal(er)	2 channels
Kanalplassering	L R
Samplingsrate	48 KHz
Oppløsning	16 bits
Strømstørrelse	1.44 MiB

Tabell D.2: Media-karakteristiker H.264 2mbit/s.

<b>Generelt</b>	<b>H.264 Video</b>
Navn	<b>3mb.mp4</b>
Format	MPEG-4
Format/Info	ISO 14496-1 Base Media
Format/Familie	MPEG-4
Filstørrelse	51.6 MiB
Lengde	2mn 24s
Bitrate	3005 Kbps
Strømstørrelse	2.24 MiB
<b>Video</b>	
Kodek	H.264
Kodek/Info	H.264 (3GPP)
Lengde	2mn 24s
Bitrate	2667 Kbps
Nominal Bitrate	2735 Kbps
Bredde	1280 pixels
Høyde	544 pixels
Display Aspect ratio	2.35
Rammerate	25.000 fps
Strømstørrelse	45.8 MiB
<b>Audio</b>	
Kodek	AAC LC
Kodek/Info	AAC Low Complexity
Lengde	2mn 24s
Bitrate	211 Kbps
Bitrate -modus	VBR
Kanal(er)	2 channels
Kanalplassering	L R
Samplingsrate	48 KHz
Oppløsning	16 bits
Strømstørrelse	3.62 MiB

Tabell D.3: Media-karakteristiker H.264 3mbit/s.



## Audio

<b>Generelt</b>	<b>AAC Audio</b>
Navn	<b>320kb.m4a</b>
Format	iTunes
Format/Info	Apple AAC Audio with iTunes info
Format/Familie	MPEG-4
Filstørrelse	9.21 MiB
Lengde	4mn 15s
Bitrate	302 Kbps
Strømstørrelse	578 KiB
<b>Audio</b>	
Kodek	AAC LC
Kodek/Info	AAC Low Complexity
Lengde	4mn 15s
Bitrate	284 Kbps
Nominal Bitrate	288 Kbps
Bitrate -modus	VBR
Kanal(er)	2 channels
Kanalplassering	L R
Samplingsrate	44.1 KHz
Oppløsning	16 bits
Strømstørrelse	8.65 MiB

Tabell D.4: Media-karakteristiker AAC 283kbit/s.

<b>Generelt</b>	<b>AAC Audio</b>
Navn	<b>152kb.m4a</b>
Format	iTunes
Format/Info	Apple AAC Audio with iTunes info
Format/Familie	MPEG-4
Filstørrelse	4.99 MiB
Lengde	4mn 15s
Bitrate	164 Kbps
Strømstørrelse	368 KiB
<b>Audio</b>	
Kodek	AAC LC
Kodek/Info	AAC Low Complexity
Lengde	4mn 15s
Bitrate	152 Kbps
Bitrate -modus	VBR
Kanal(er)	2 channels
Kanalplassering	L R
Samplingsrate	44.1 KHz
Oppløsning	16 bits
Strømstørrelse	4.63 MiB

Tabell D.5: Media-karakteristiker AAC 159kbit/s.

<b>Generelt</b>	<b>AAC Audio</b>
Navn	<b>59kb.m4a</b>
Format	iTunes
Format/Info	Apple AAC Audio with iTunes info
Format/Familie	MPEG-4
Filstørrelse	2.09 MiB
Lengde	4mn 15s
Bitrate	69 Kbps
Strømstørrelse	300 KiB
<b>Audio</b>	
Kodek	AAC LC
Kodek/Info	AAC Low Complexity
Lengde	4mn 15s
Bitrate	59 Kbps
Bitrate -modus	VBR
Kanal(er)	2 channels
Kanalplassering	L R
Samplingsrate	44.1 KHz
Oppløsning	16 bits
Strømstørrelse	1.80 MiB

Tabell D.6: Media-karakteristiker AAC 59kbit/s.

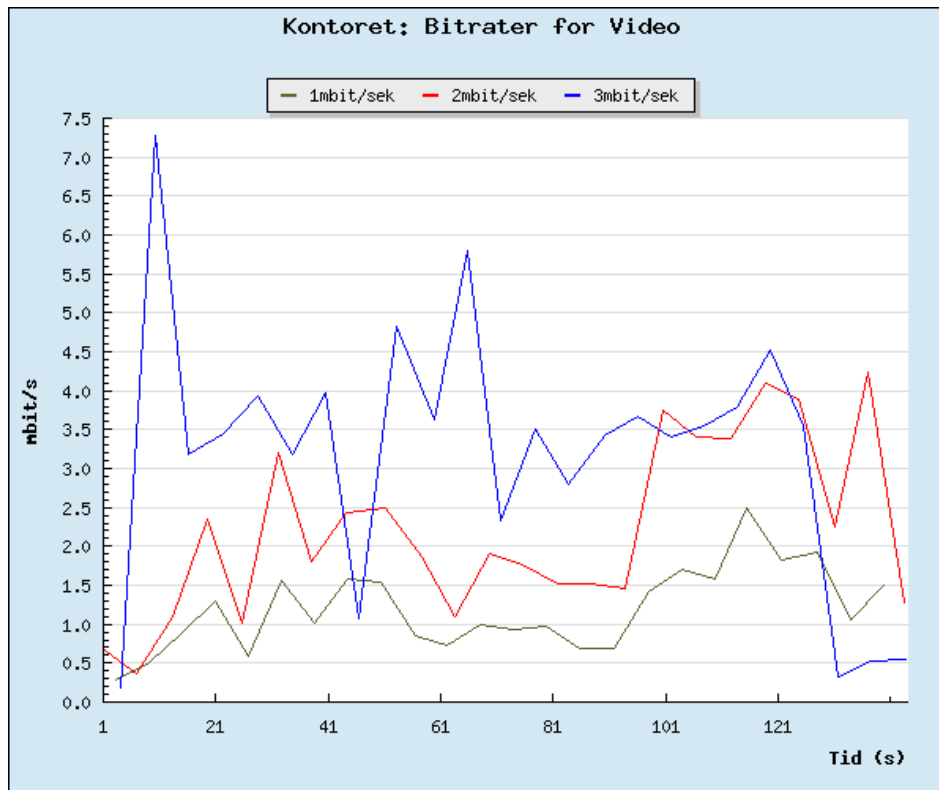


# E

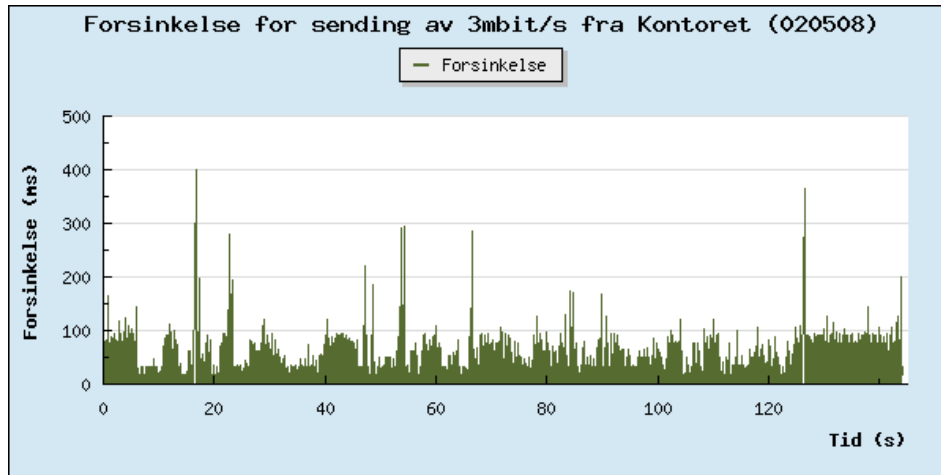
Grafer

## E.1 Kontoret

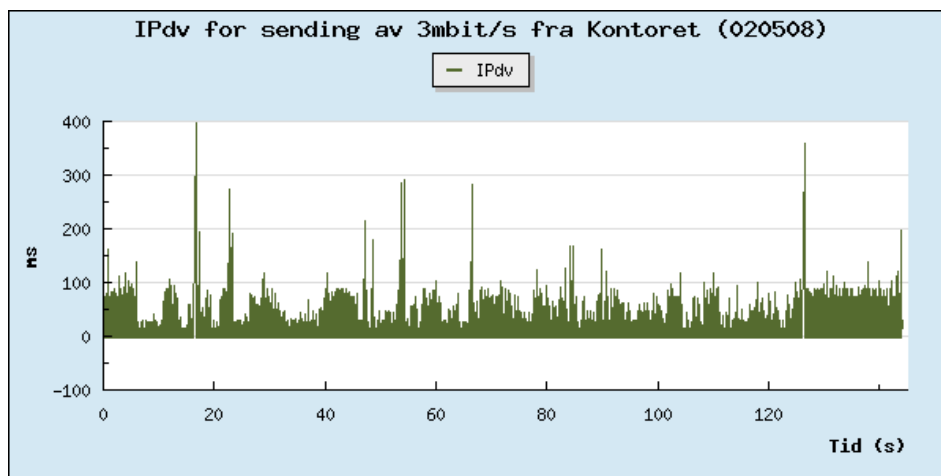
### E.1.1 Video



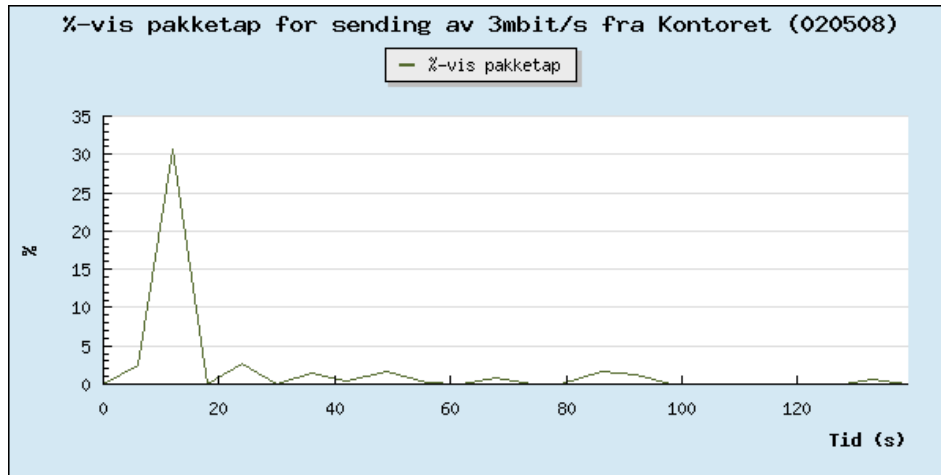
Figur E.1: Resulterende bitrater ved sending av video fra Kontoret.



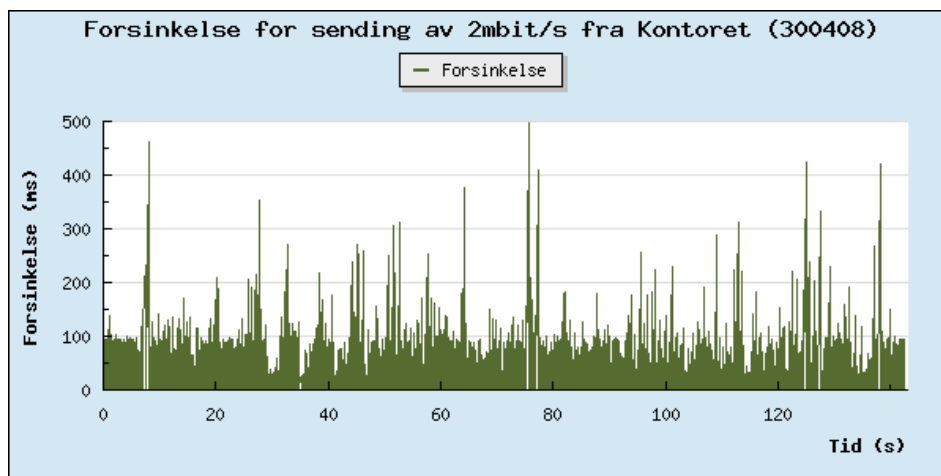
Figur E.2: Forsinkelse ved sending av 3mbit/s videostream fra Kontoret.



Figur E.3: Variasjon av forsinkelse ved sending av 3mbit/s videostream fra Kontoret.

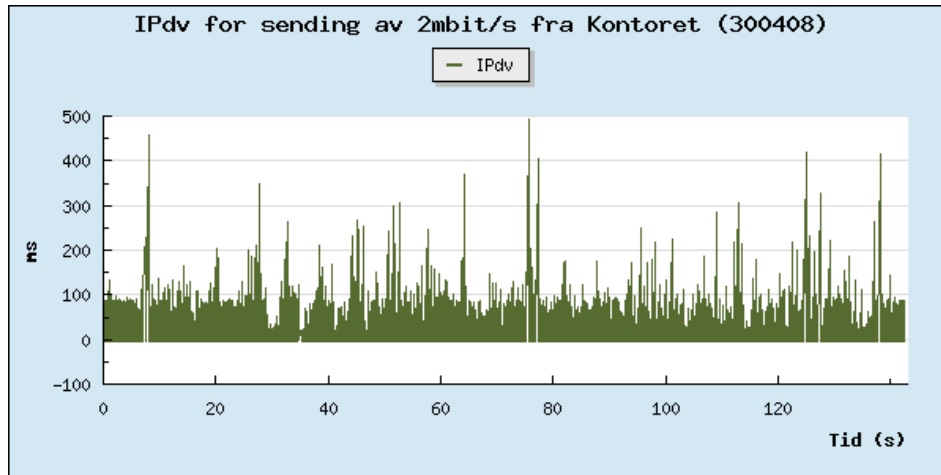


Figur E.4: Pakketap ved sending av 3mbit/s videostrøm fra Kontoret.

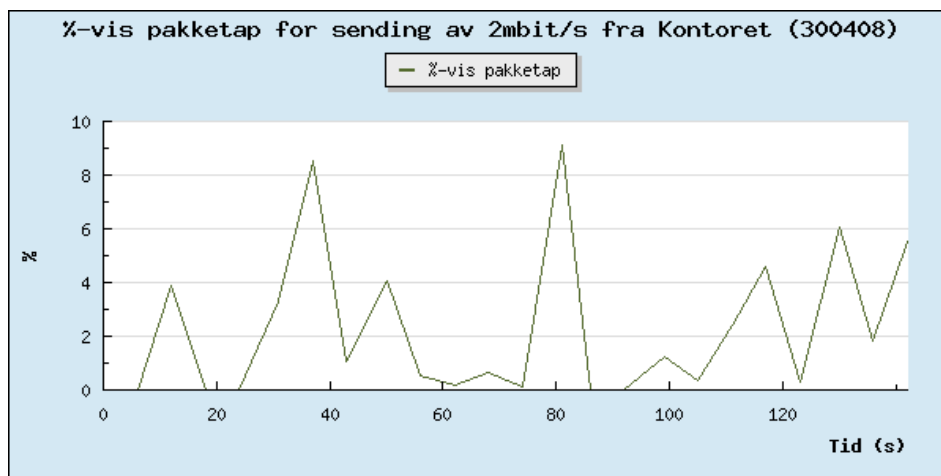


Figur E.5: Forsinkelse ved sending av 2mbit/s videostrøm fra Kontoret.

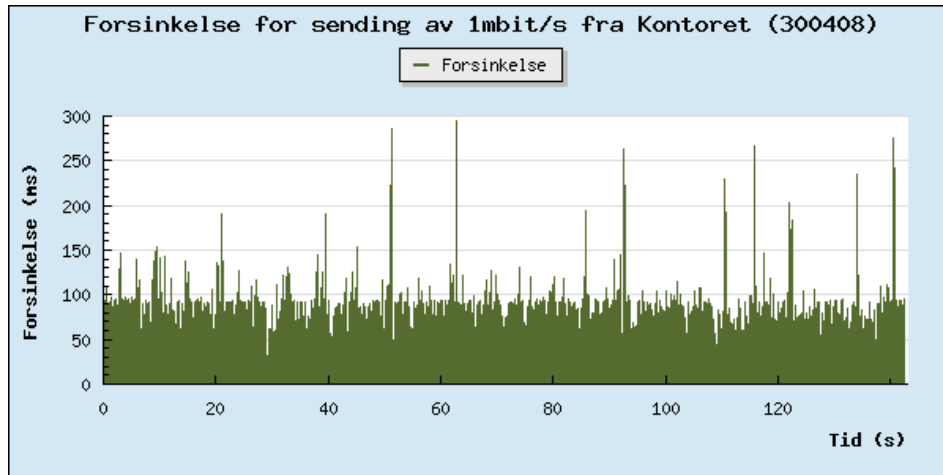




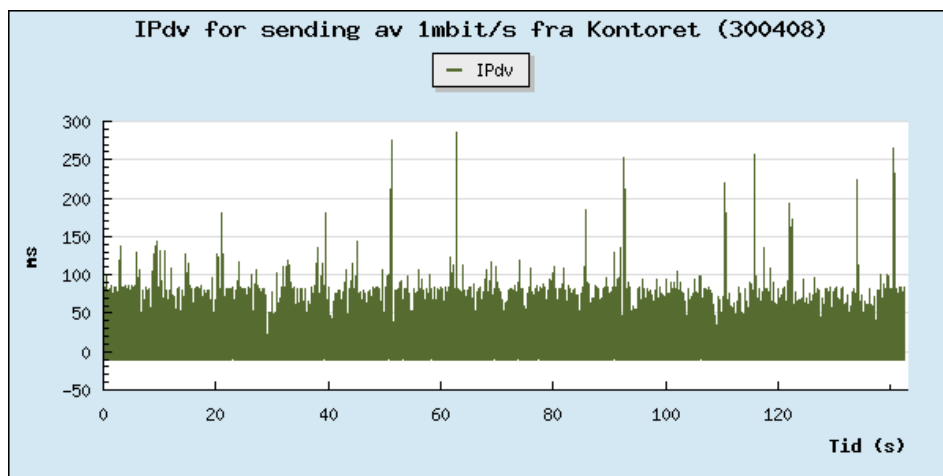
Figur E.6: Variasjon av forsinkelse ved sending av 2mbit/s videostrøm fra Kontoret.



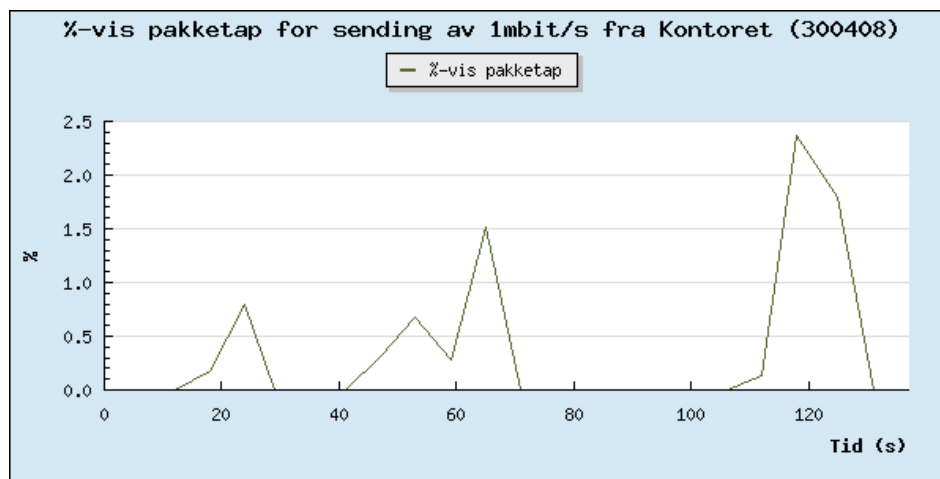
Figur E.7: Pakketap ved sending av 2mbit/s videostrøm fra Kontoret.



Figur E.8: Forsinkelse ved sending av 1mbit/s videostream fra Kontoret.

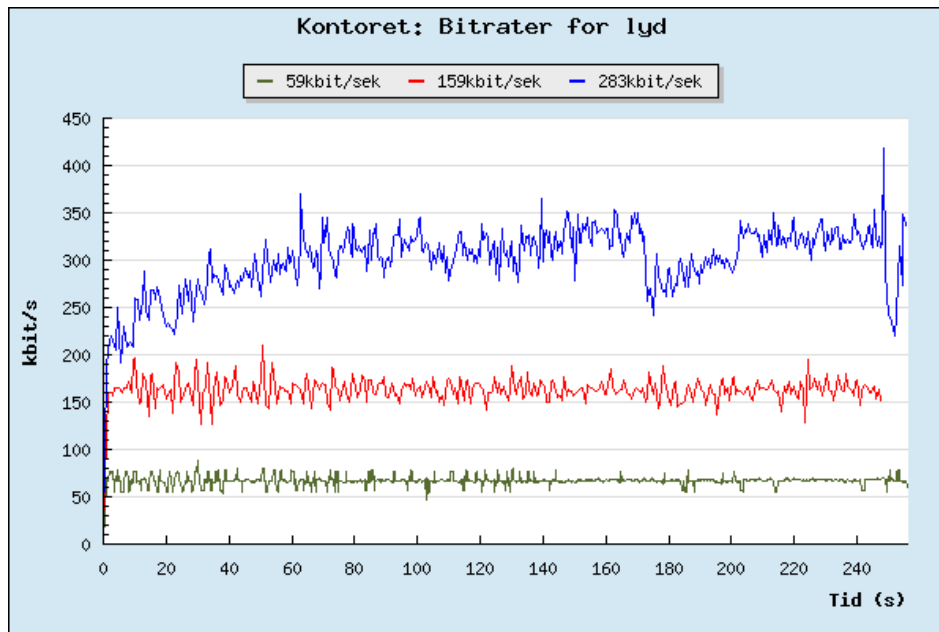


Figur E.9: Variasjon av forsinkelse ved sending av 1mbit/s videostream fra Kontoret.

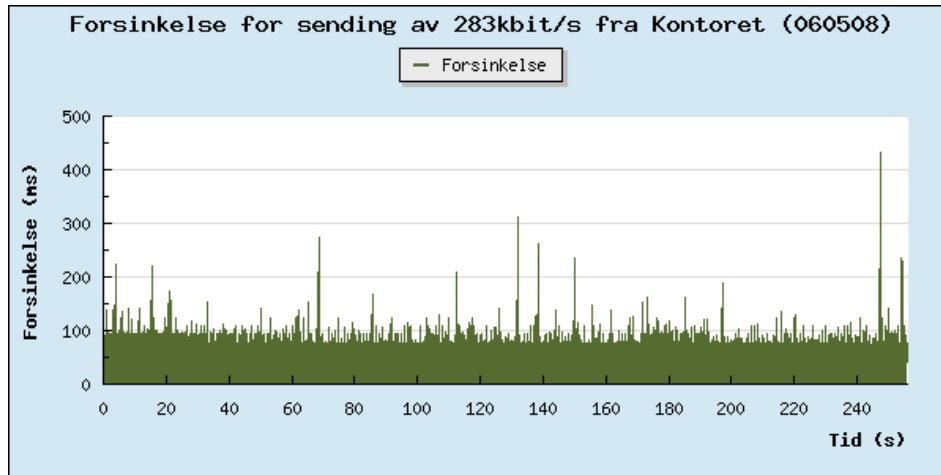


Figur E.10: Pakketap ved sending av 1mbit/s videostrøm fra Kontoret.

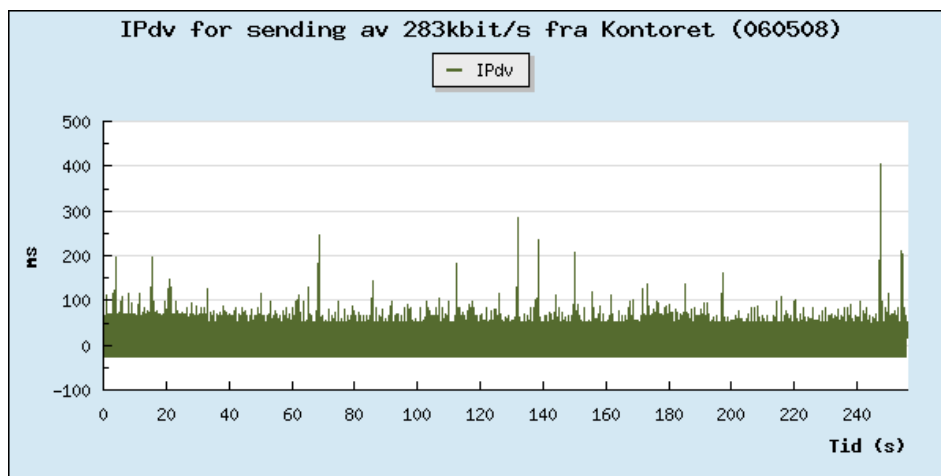
## E.1.2 Lyd



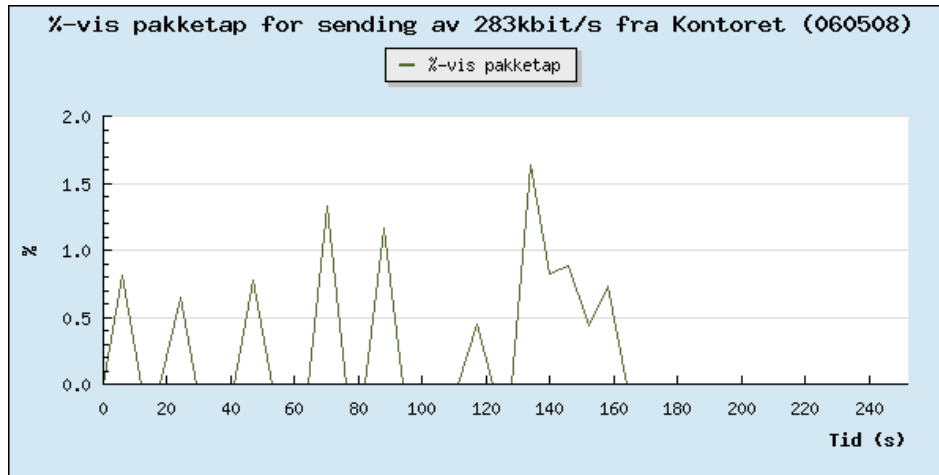
Figur E.11: Resulterende bitrater ved sending av lyd fra Kontoret.



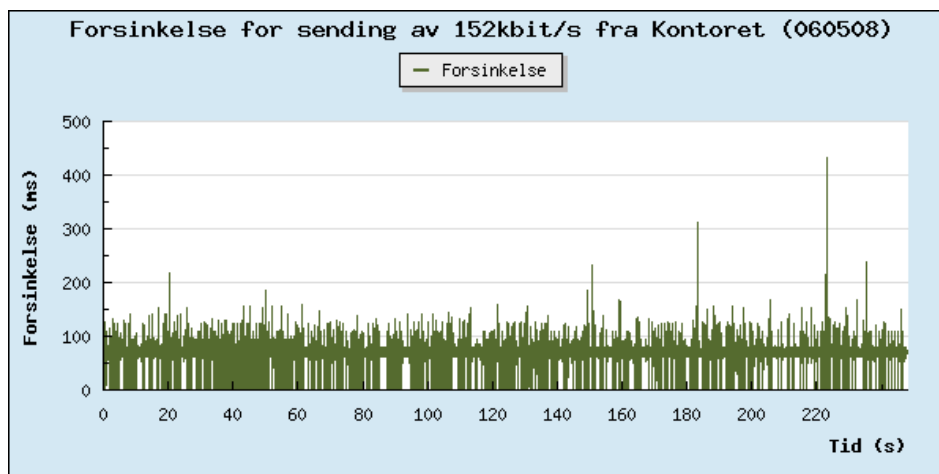
Figur E.12: Forsinkelse ved sending av 283kbit/s lydstrøm fra Kontoret.



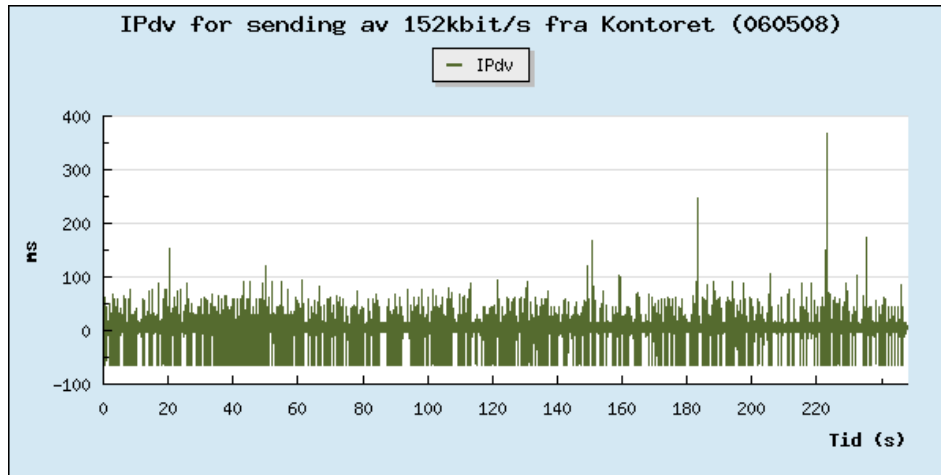
Figur E.13: Variasjon av forsinkelse ved sending av 283kbit/s lydstrøm fra Kontoret.



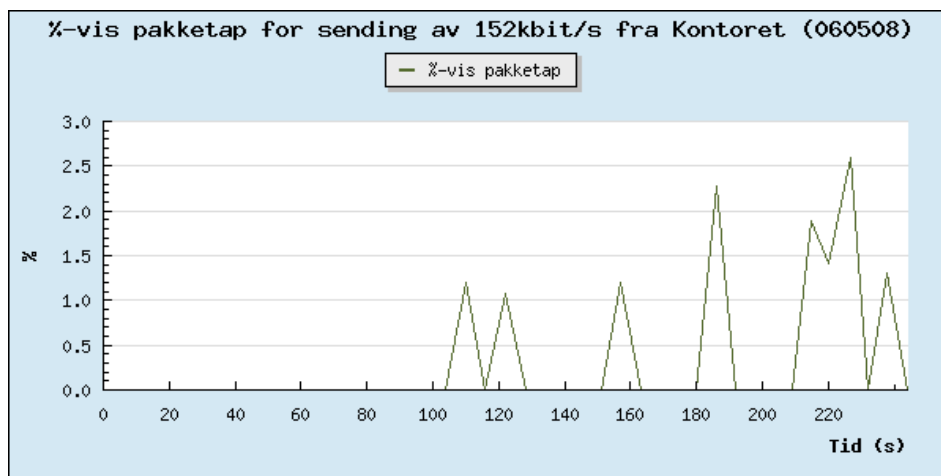
Figur E.14: Pakketap ved sending av 283kbit/s lydstrøm fra Kontoret.



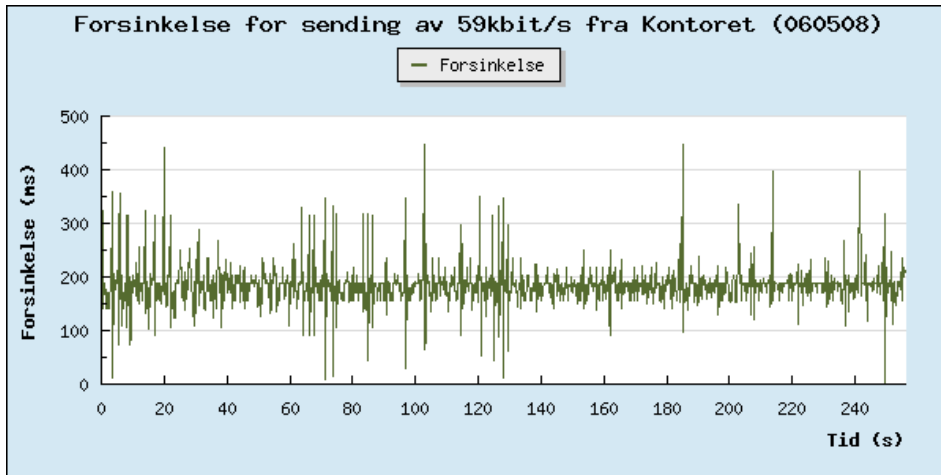
Figur E.15: Forsinkelse ved sending av 152kbit/s lydstrøm fra Kontoret.



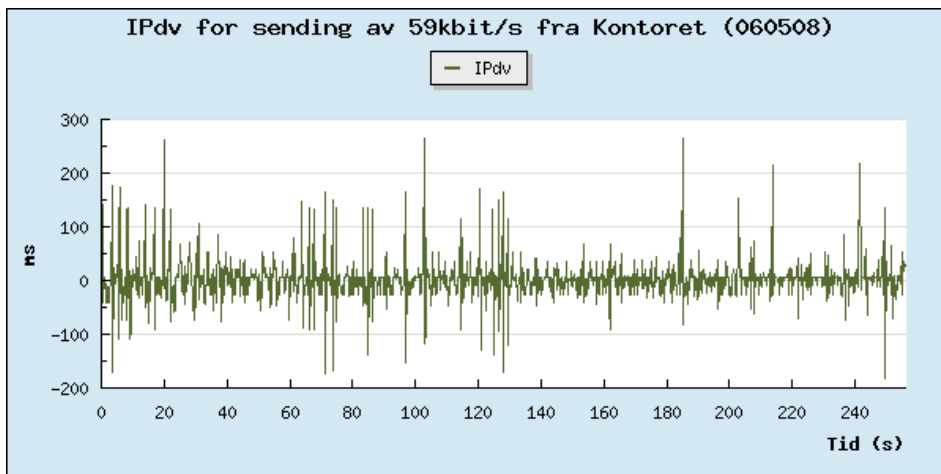
Figur E.16: Variasjon av forsinkelse ved sending av 152kbit/s lydstrøm fra Kontoret.



Figur E.17: Pakketap ved sending av 152kbit/s lydstrøm fra Kontoret.

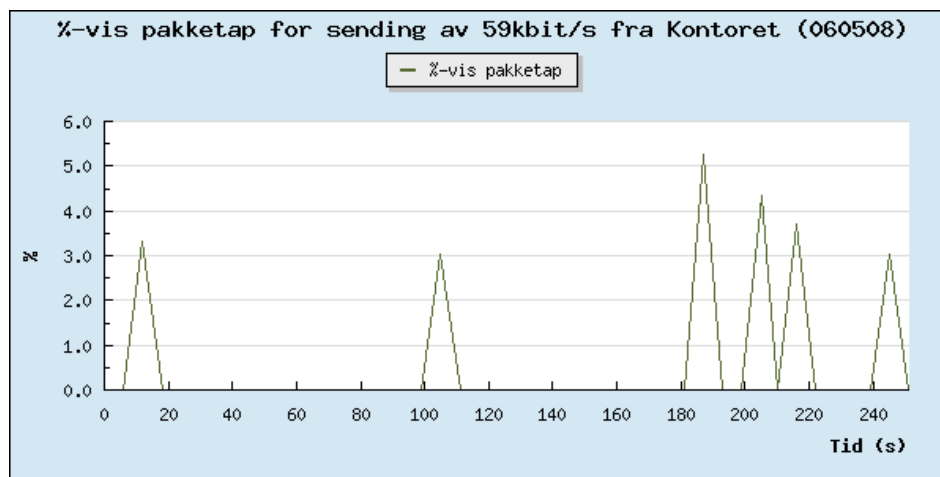


Figur E.18: Forsinkelse ved sending av 59kbit/s lydstrøm fra Kontoret.



Figur E.19: Variasjon av forsinkelse ved sending av 59kbit/s lydstrøm fra Kontoret.

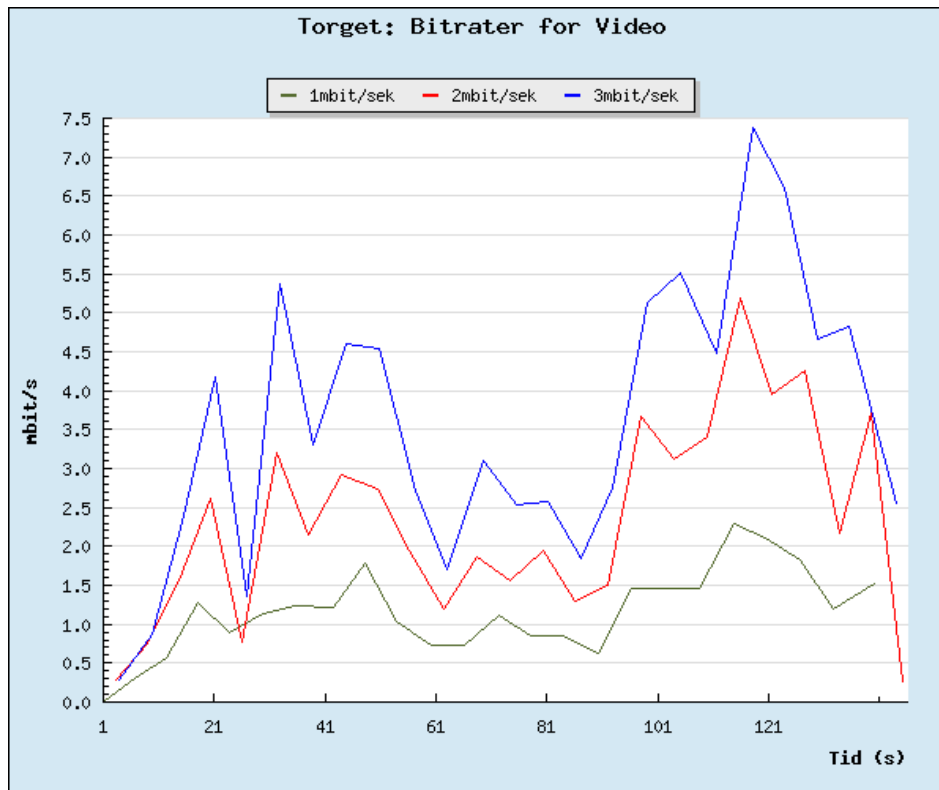




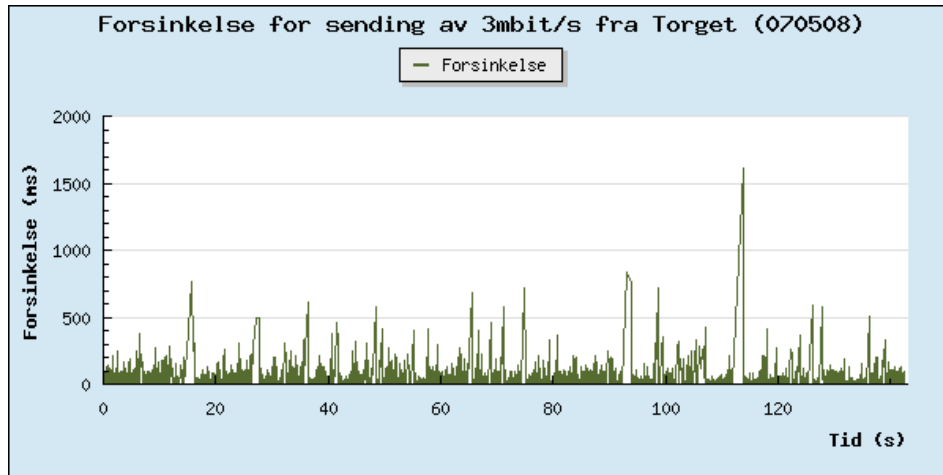
Figur E.20: Pakketap ved sending av 59kbit/s lydstrøm fra Kontoret.

## E.2 Torget

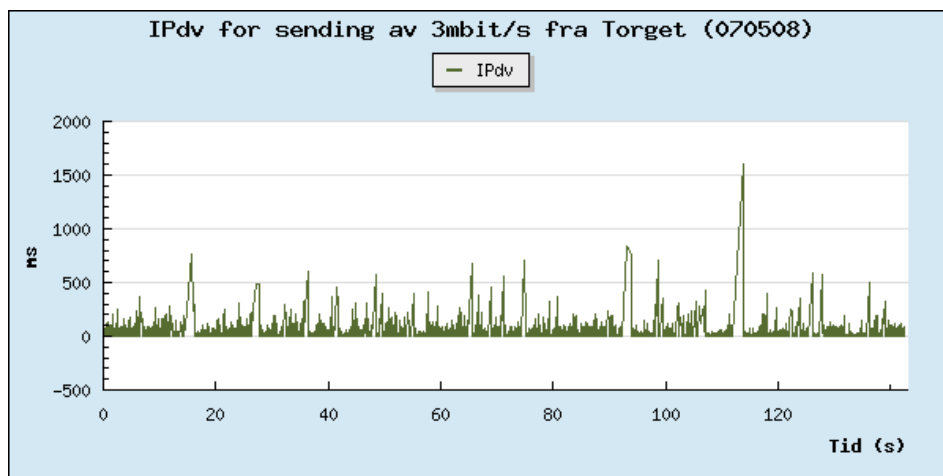
### E.2.1 Video



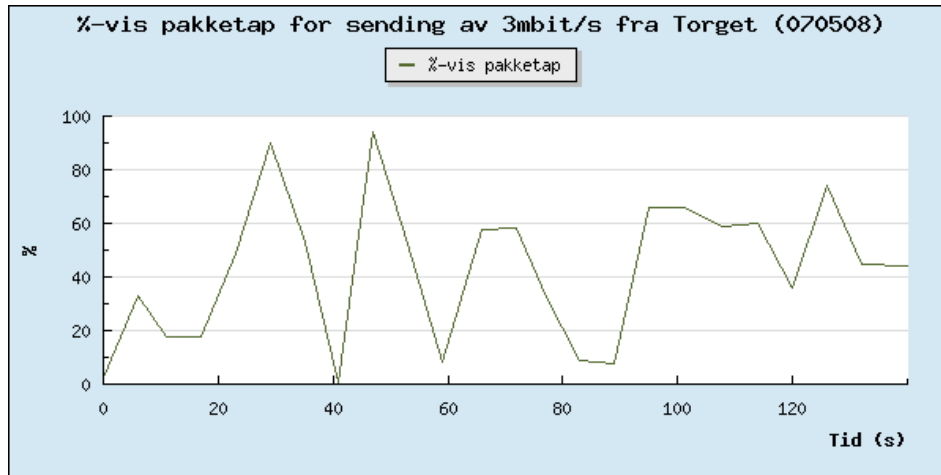
Figur E.21: Resulterende bitrater ved sending av video fra Torget



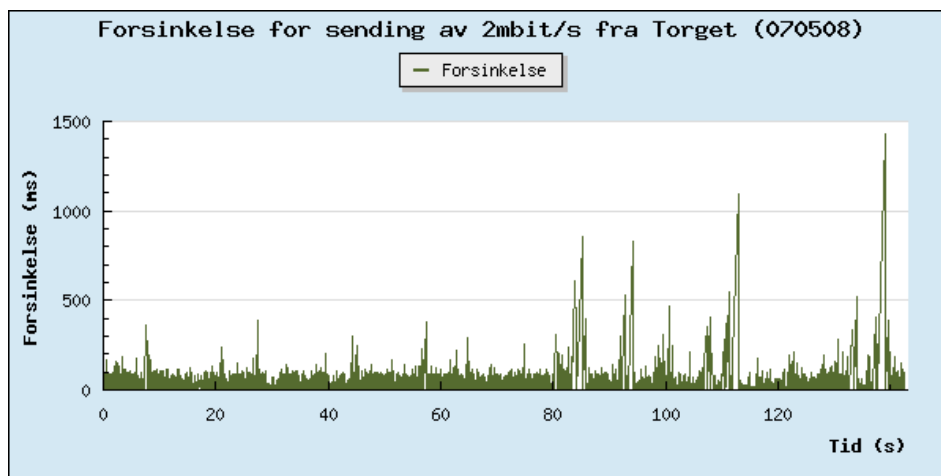
Figur E.22: Forsinkelse ved sending av 3mbit/s videostrøm fra Torget.



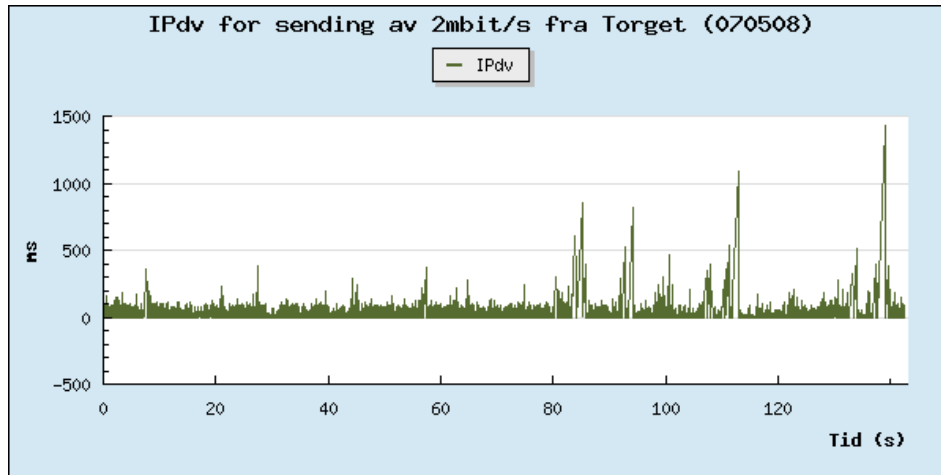
Figur E.23: Variasjon av forsinkelse ved sending av 3mbit/s videostrøm fra Torget.



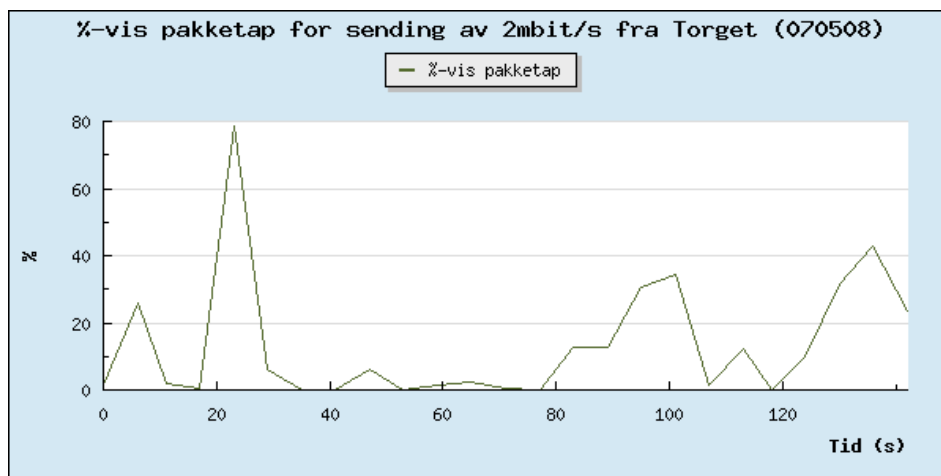
Figur E.24: Pakketap ved sending av 3mbit/s videostrøm fra Torget.



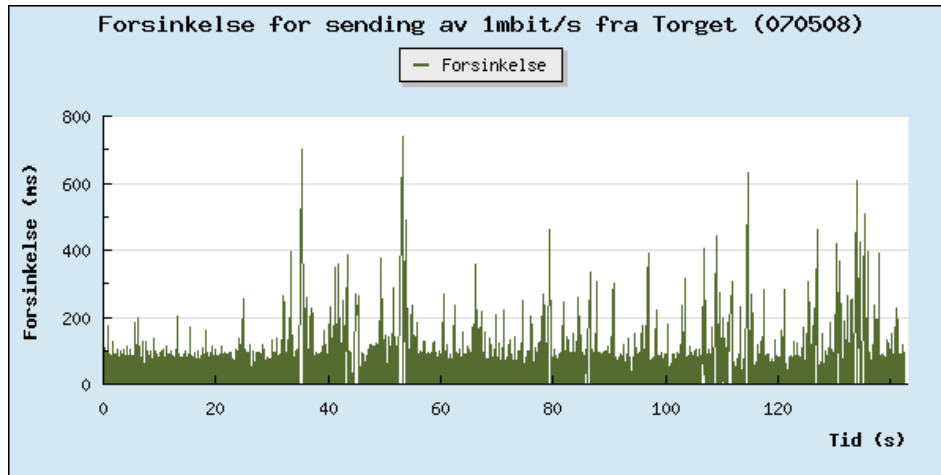
Figur E.25: Forsinkelse ved sending av 2mbit/s videostrøm fra Torget.



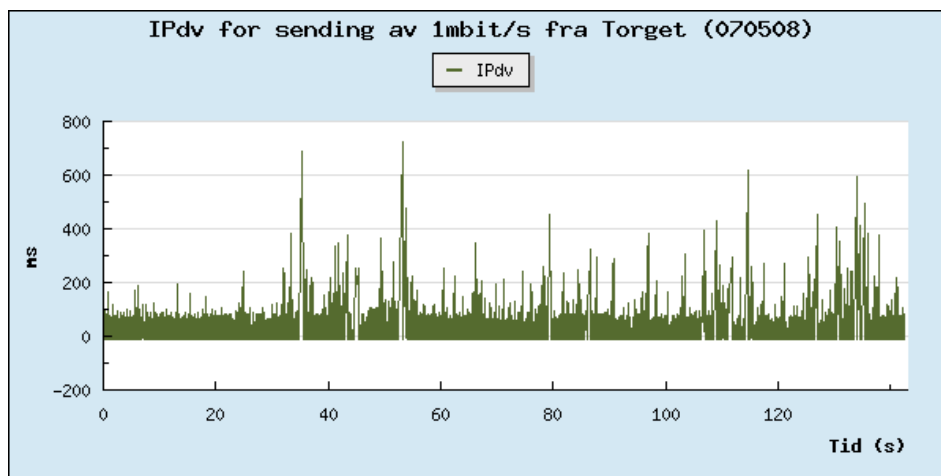
Figur E.26: Variasjon av forsinkelse ved sending av 2mbit/s videostrøm fra Torget.



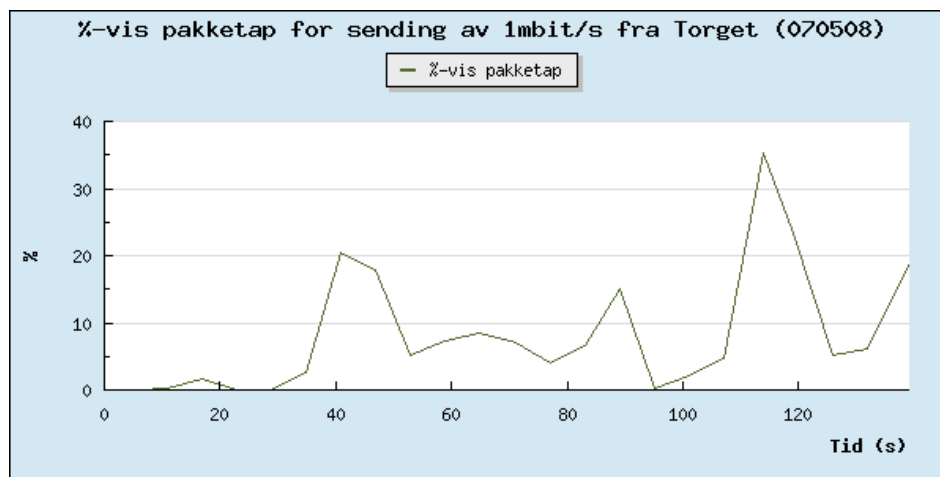
Figur E.27: Pakketap ved sending av 2mbit/s videostrøm fra Torget.



Figur E.28: Forsinkelse ved sending av 1mbit/s videostream fra Torget.

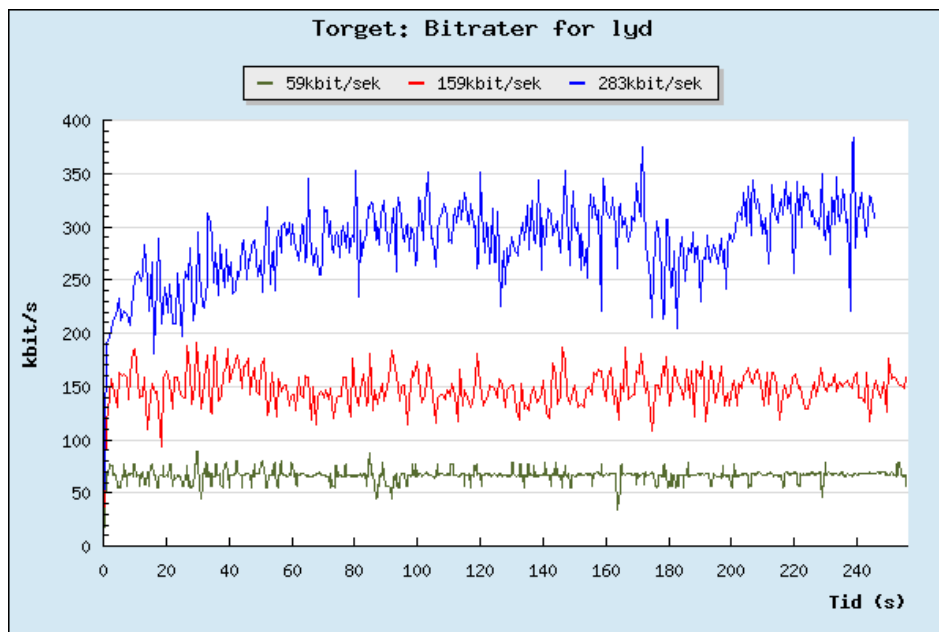


Figur E.29: Variasjon av forsinkelse ved sending av 1mbit/s videostream fra Torget.



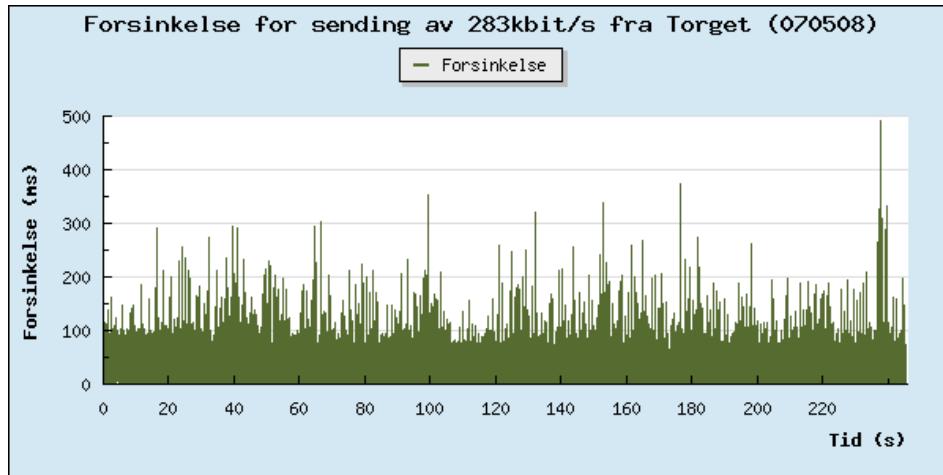
Figur E.30: Pakketap ved sending av 1mbit/s videostrøm fra Torget.

## E.2.2 Lyd

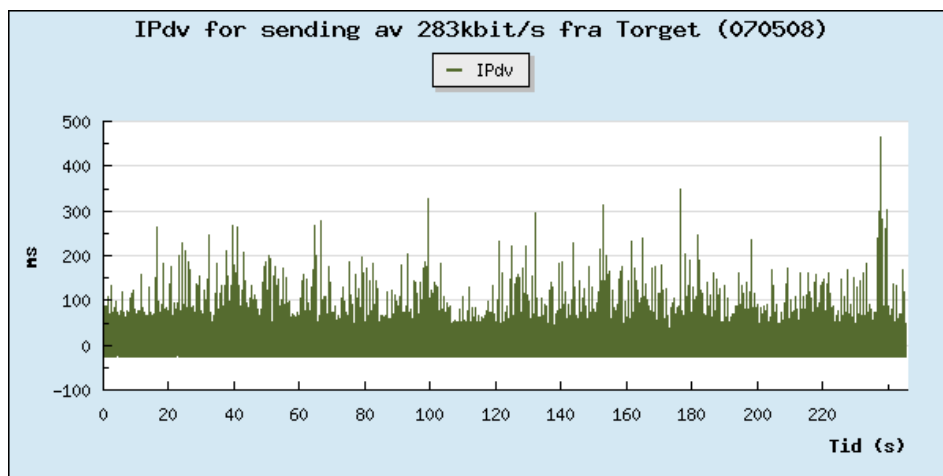


Figur E.31: Resulterende bitrater ved sending av lyd fra Torget

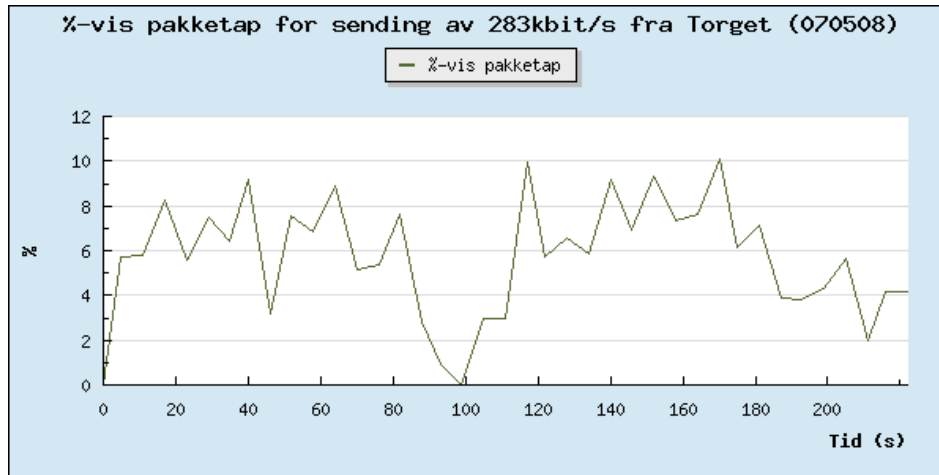




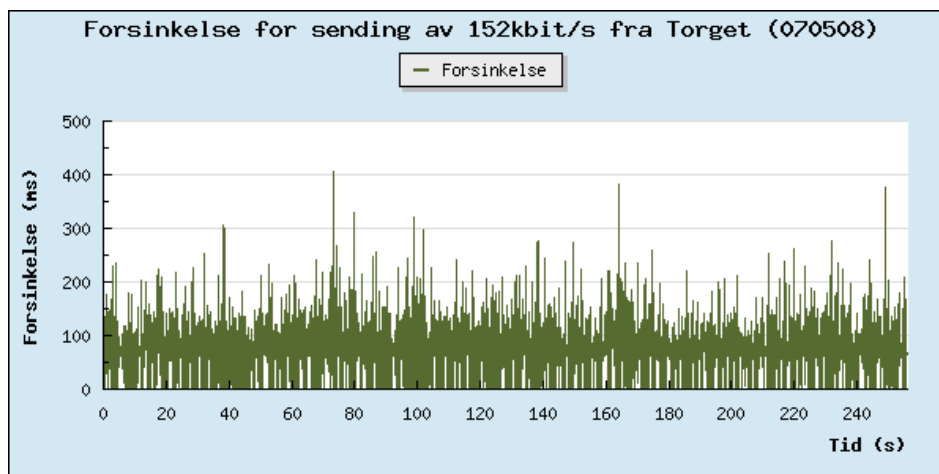
Figur E.32: Forsinkelse ved sending av 283kbit/s lydstrøm fra Torget.



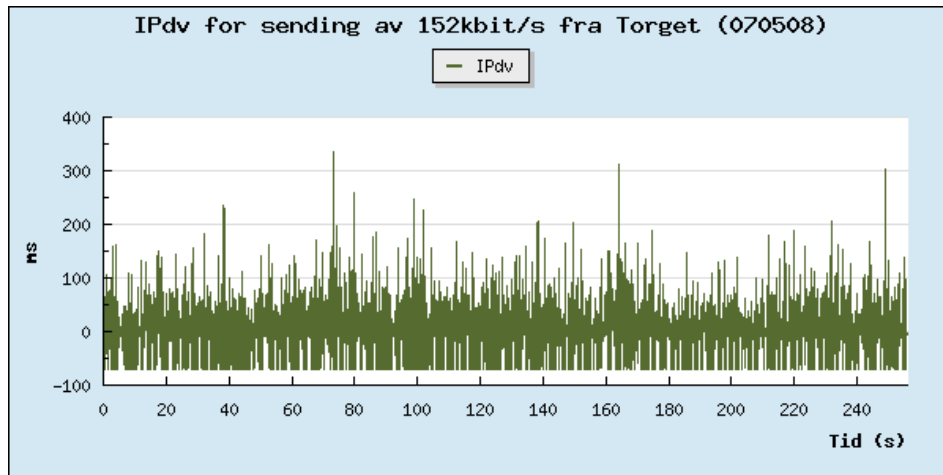
Figur E.33: Variasjon av forsinkelse ved sending av 283kbit/s lydstrøm fra Torget.



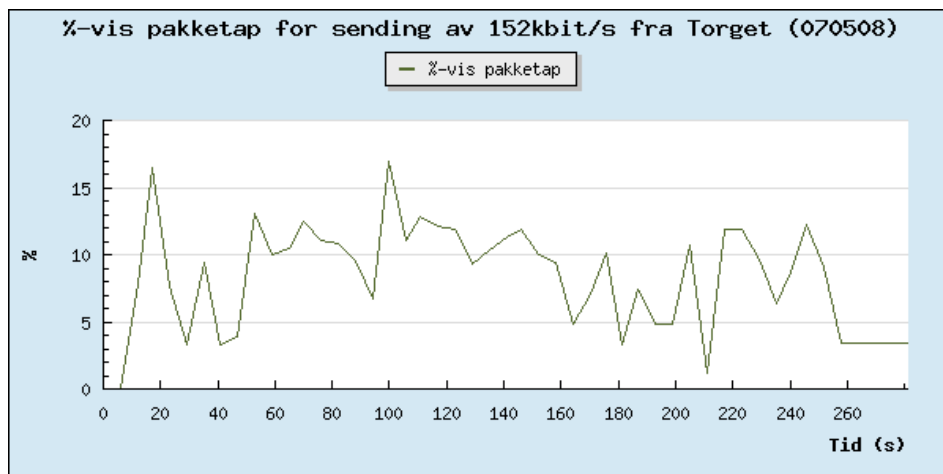
Figur E.34: Pakketap ved sending av 283kbit/s lydstrøm fra Torget.



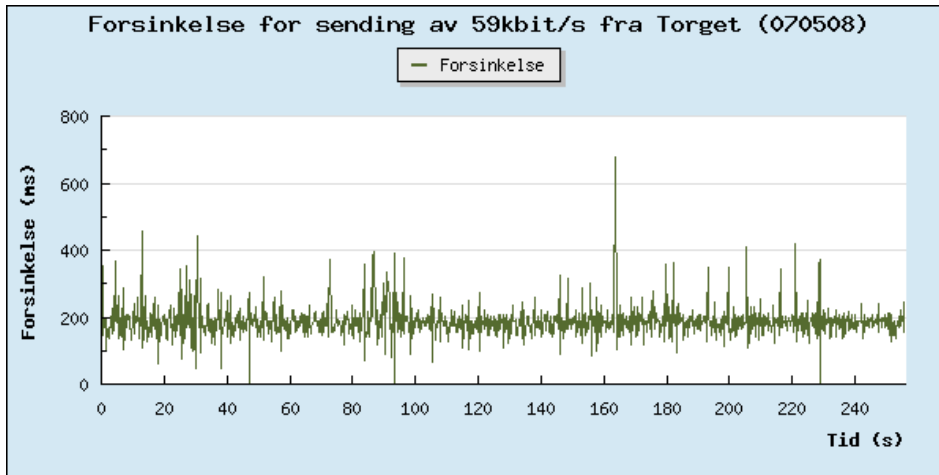
Figur E.35: Forsinkelse ved sending av lydstrøm fra Torget.



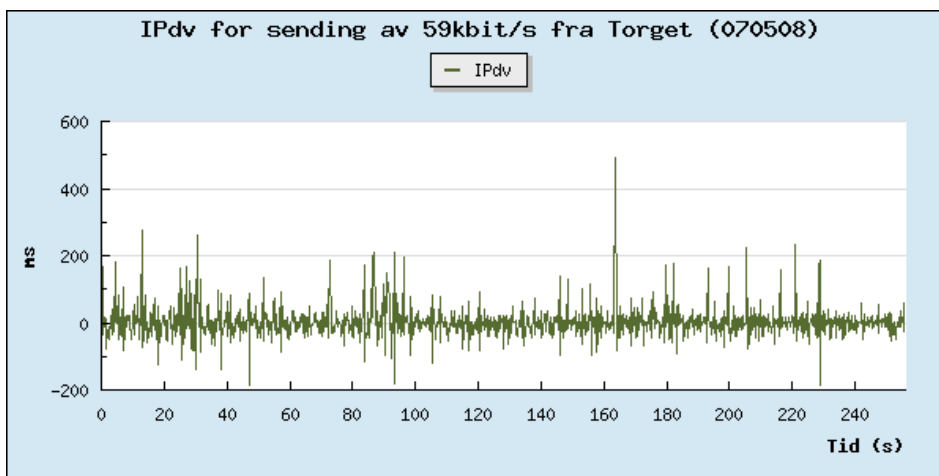
Figur E.36: Variasjon av forsinkelse ved sending av lydstrøm fra Torget.



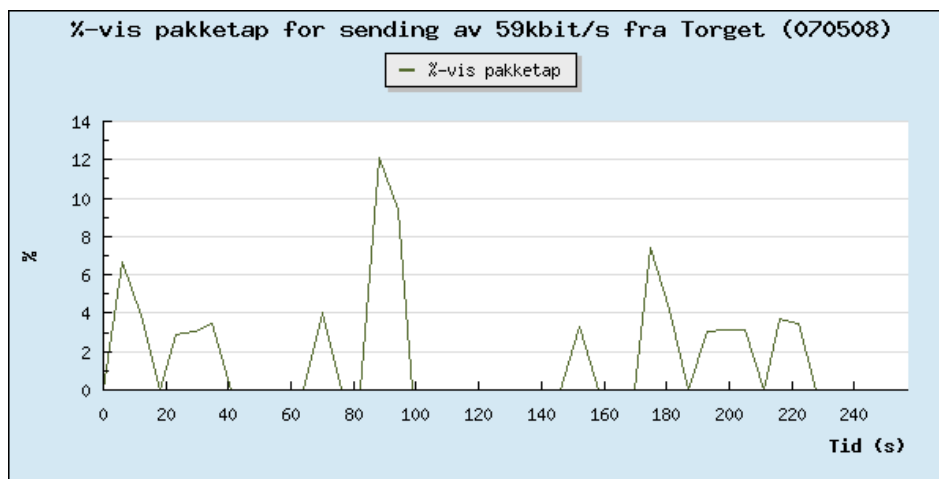
Figur E.37: Pakketap ved sending av lydstrøm fra Torget.



Figur E.38: Forsinkelse ved sending av 59kbit/s lydstrøm fra Torget.



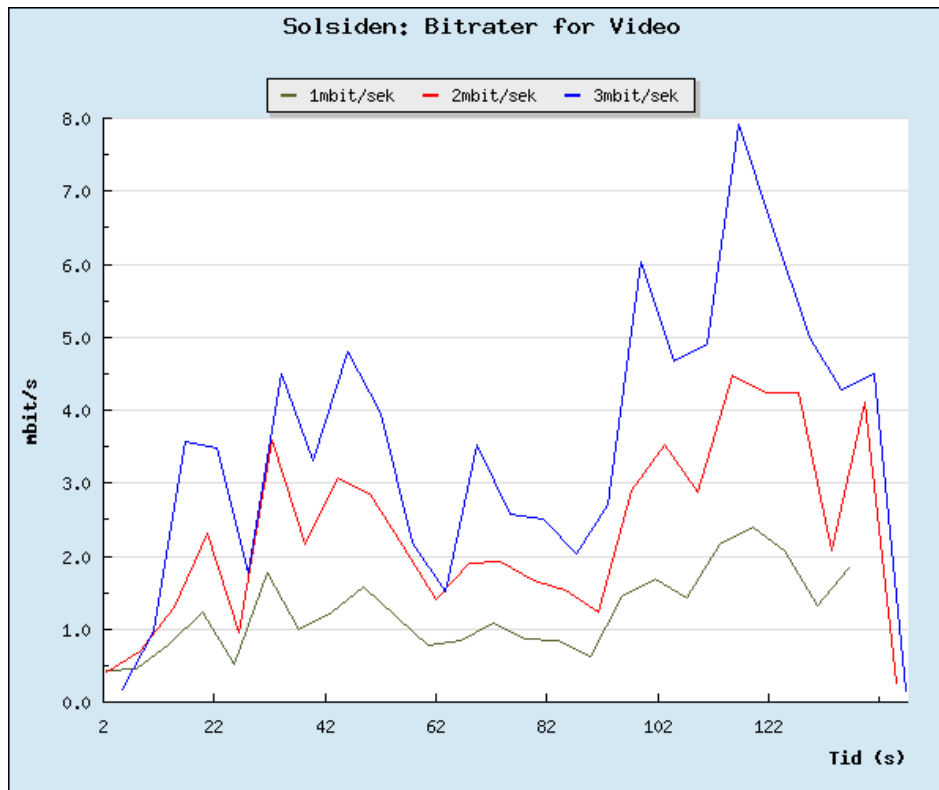
Figur E.39: Variasjon av forsinkelse ved sending av 59kbit/s lydstrøm fra Torget.



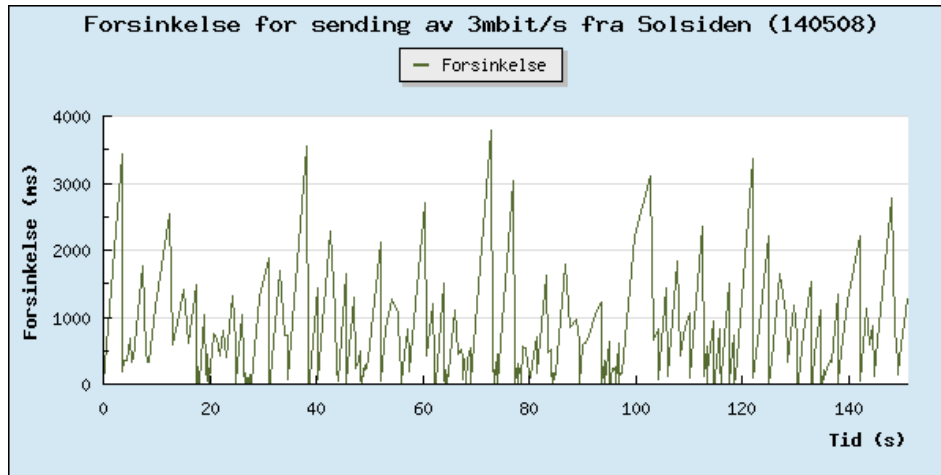
Figur E.40: Pakketap ved sending av 59kbit/s lydstrøm fra Torget.

## E.3 Solsiden

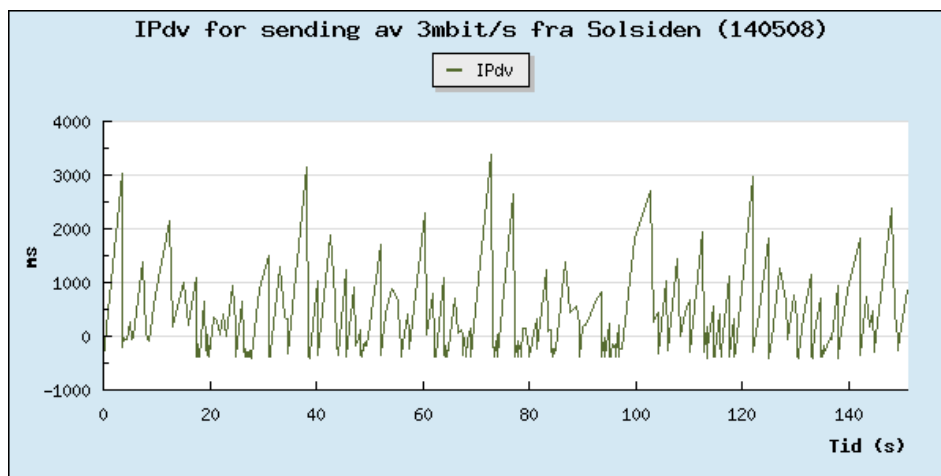
### E.3.1 Video



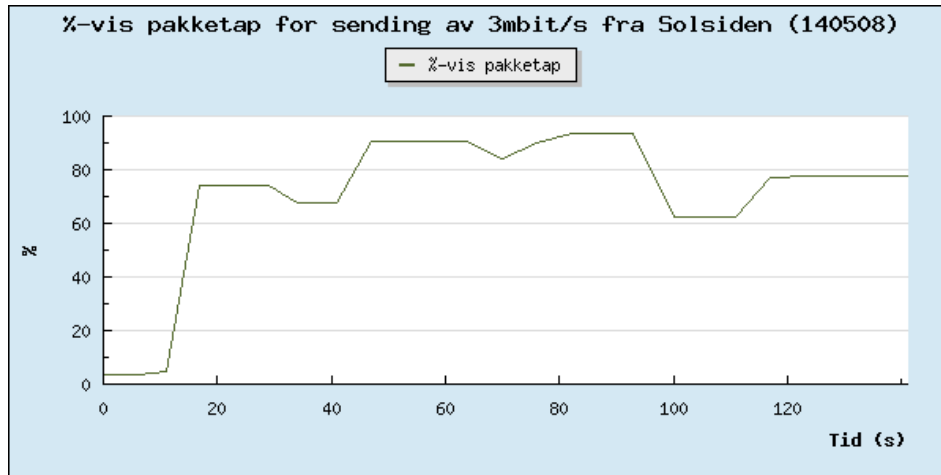
Figur E.41: Resulterende bitrater ved sending av video fra Solsiden



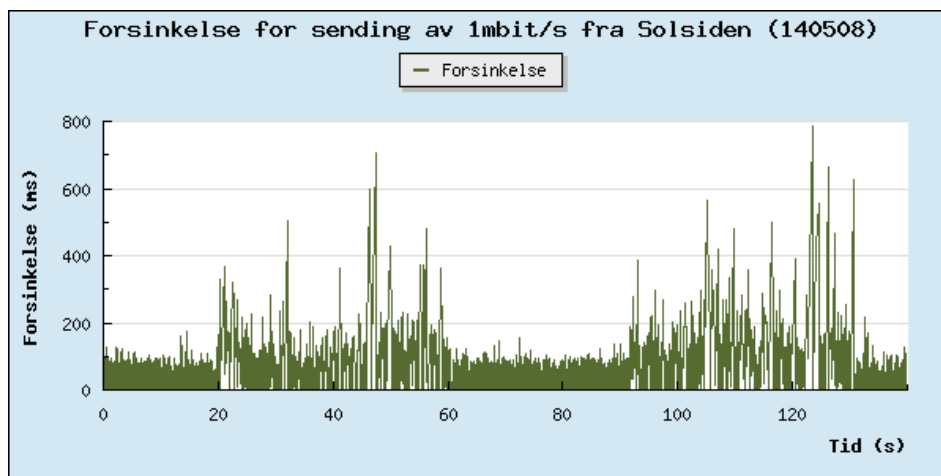
Figur E.42: Forsinkelse ved sending av 3mbit/s videostrøm fra Solsiden.



Figur E.43: Variasjon av forsinkelse ved sending av 3mbit/s videostrøm fra Solsiden.

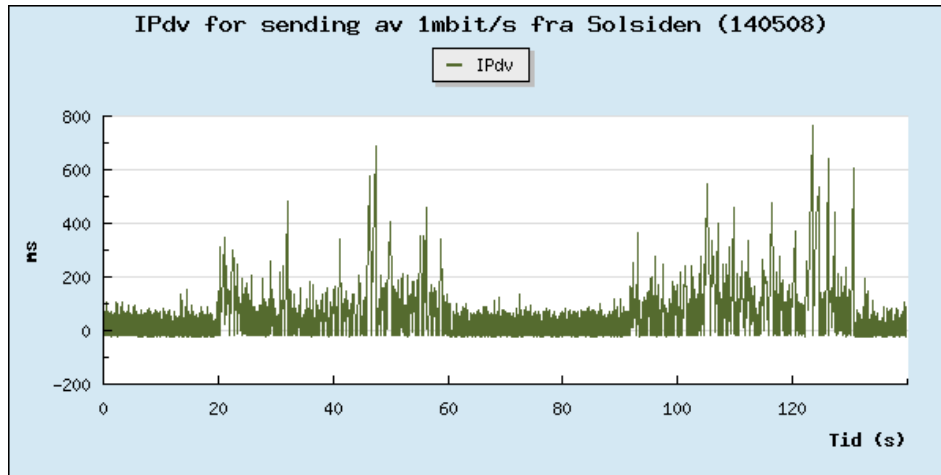


Figur E.44: Pakketap ved sending av 3mbit/s videostrøm fra Solsiden.

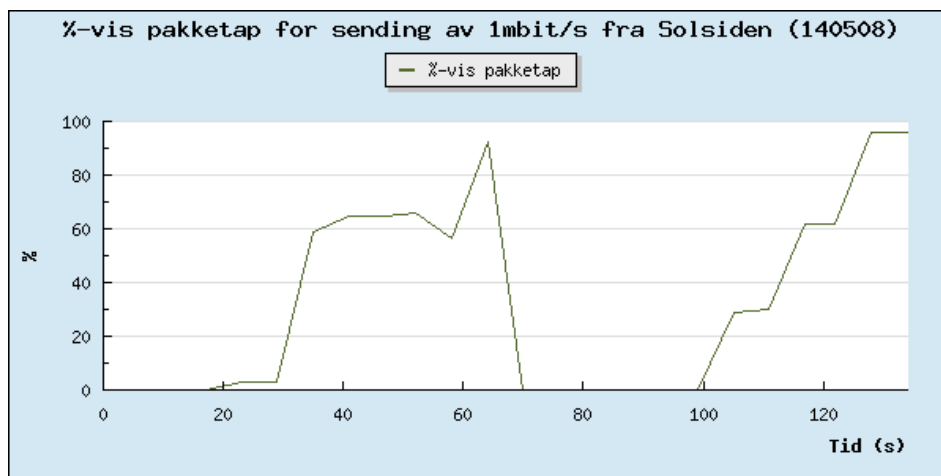


Figur E.45: Forsinkelse ved sending av 1mbit/s videostrøm fra Solsiden.



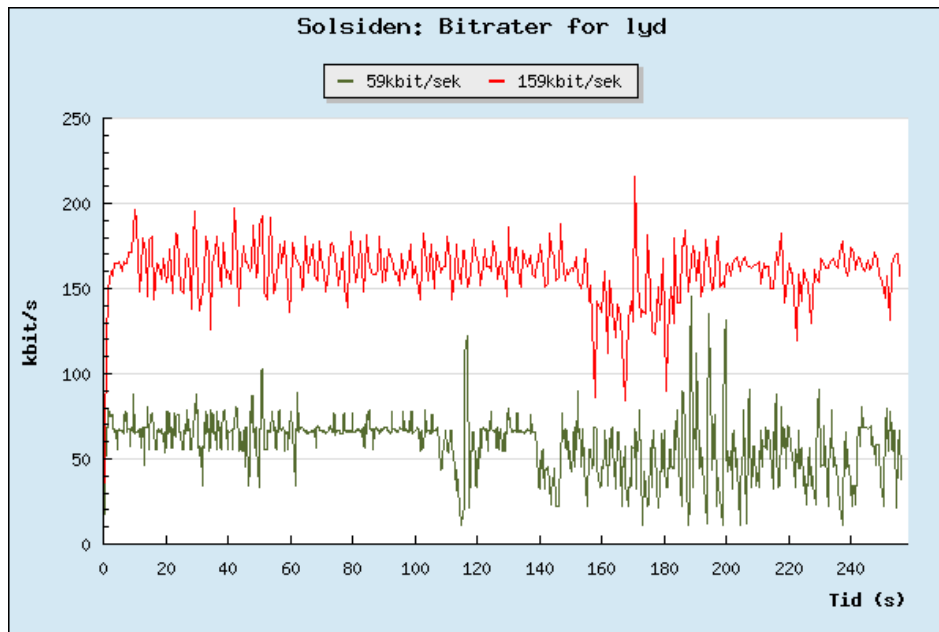


Figur E.46: Variasjon av forsinkelse ved sending av 1mbit/s videostrøm fra Solsiden.

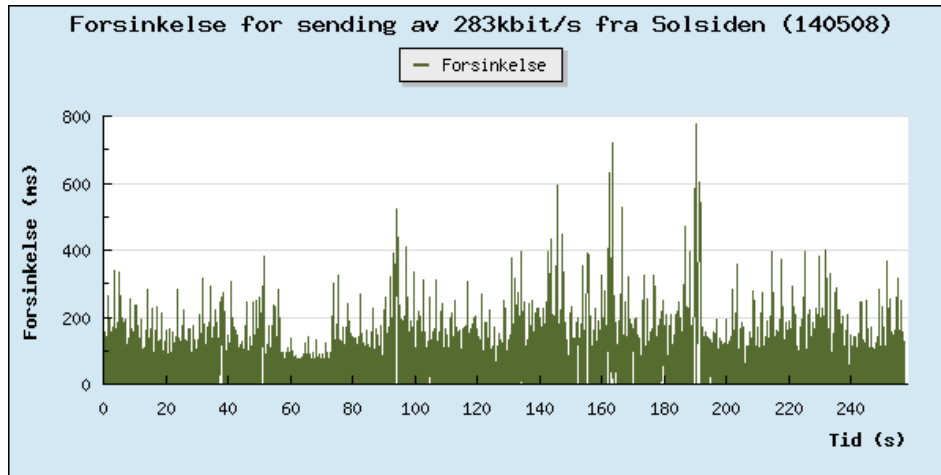


Figur E.47: Pakketap ved sending av 1mbit/s videostrøm fra Solsiden.

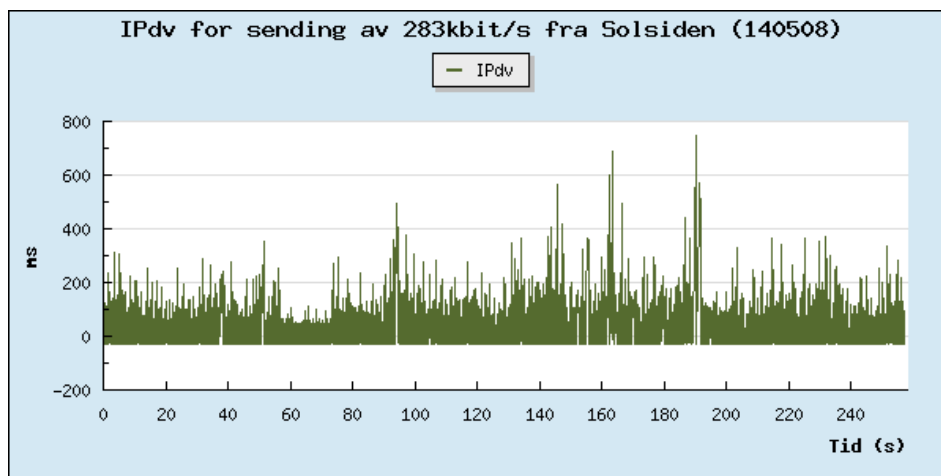
## E.3.2 Lyd



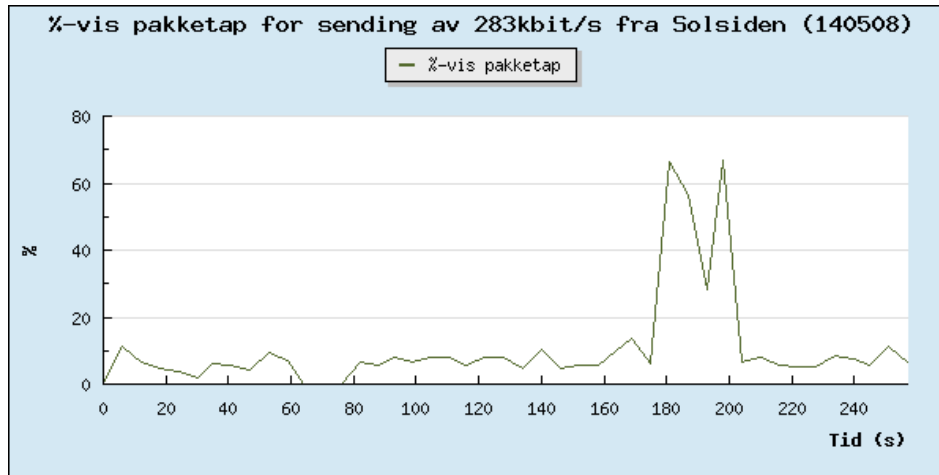
Figur E.48: Resulterende bitrater ved sending av lyd fra Solsiden



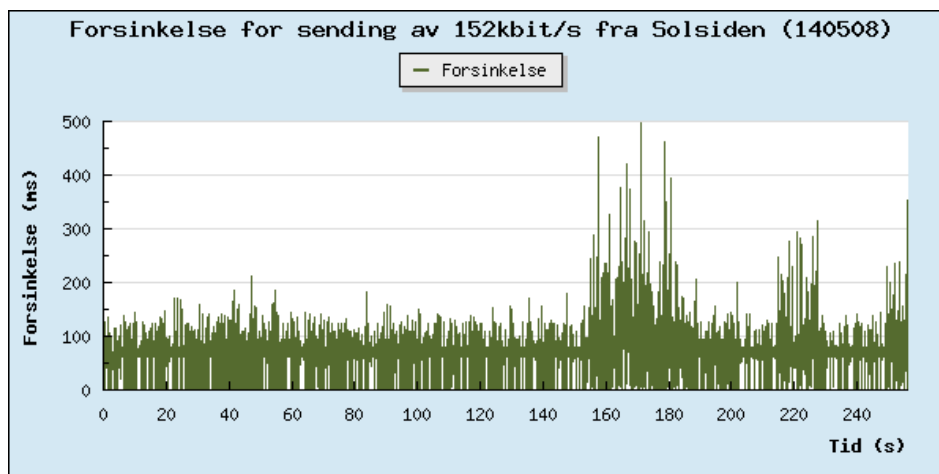
Figur E.49: Forsinkelse ved sending av 283kbit/s lydstrøm fra Solsiden.



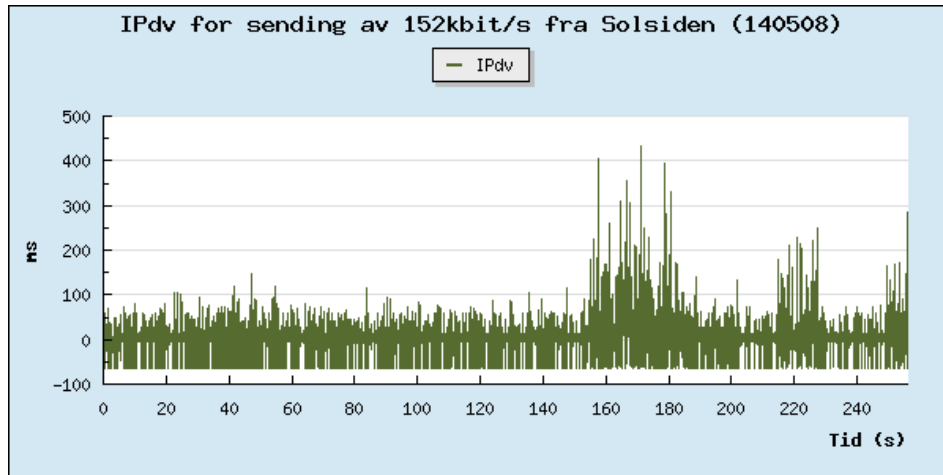
Figur E.50: Variasjon av forsinkelse ved sending av 283kbit/s lydstrøm fra Solsiden.



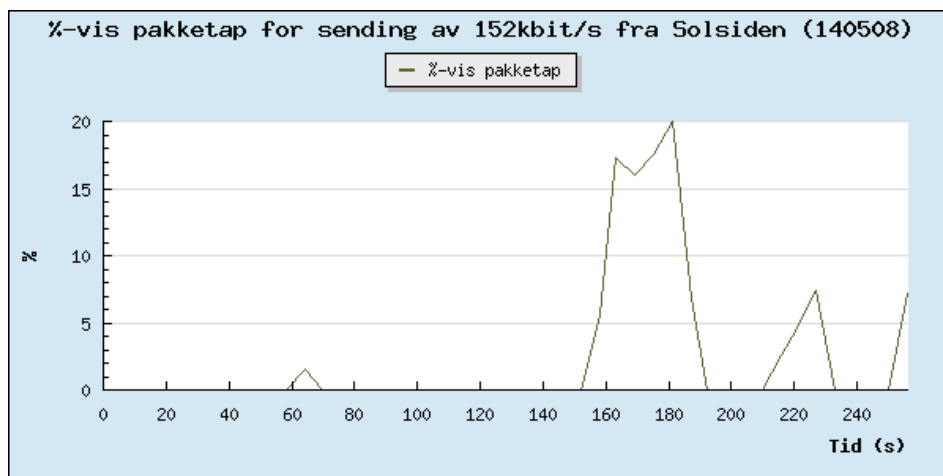
Figur E.51: Pakketap ved sending av 283kbit/s lydstrøm fra Solsiden.



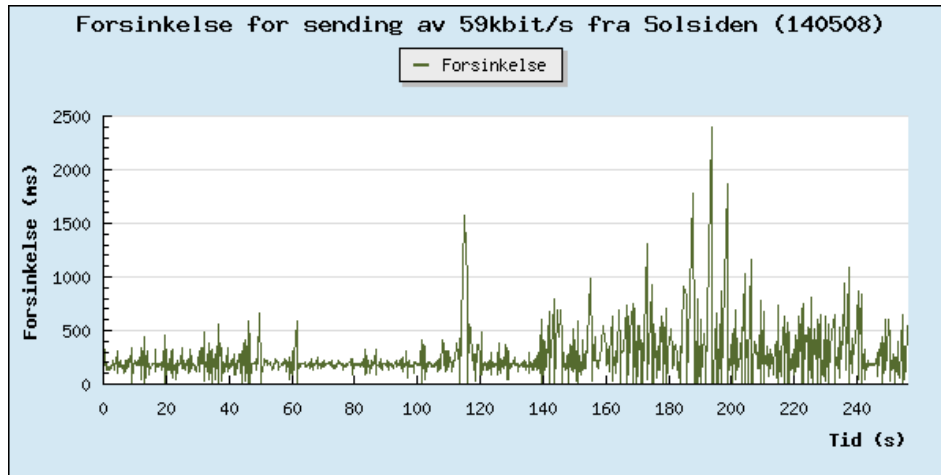
Figur E.52: Forsinkelse ved sending av 152kbit/s lydstrøm fra Solsiden.



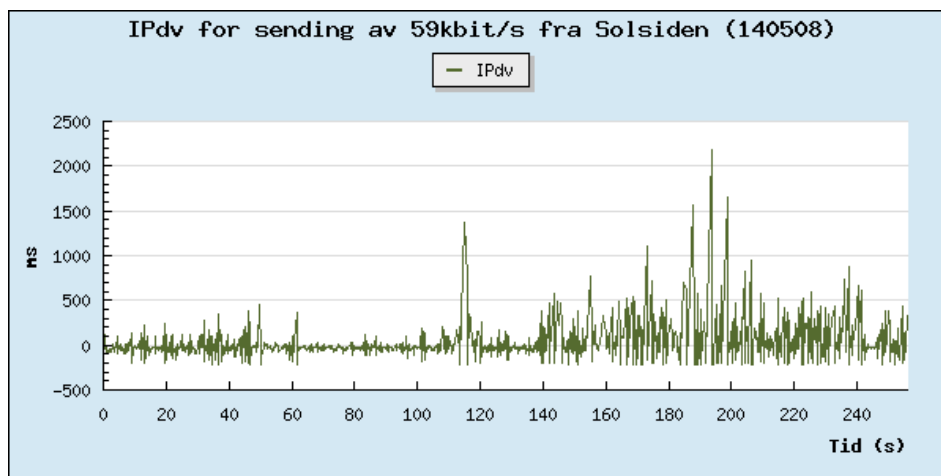
Figur E.53: Variasjon av forsinkelse ved sending av 152kbit/s lydstrøm fra Solsiden.



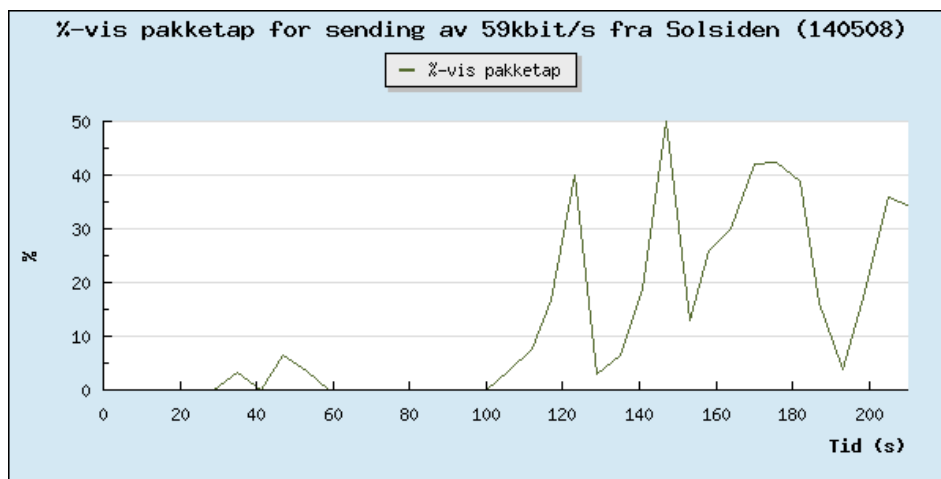
Figur E.54: Pakketap ved sending av 152kbit/s lydstrøm fra Solsiden.



Figur E.55: Forsinkelse ved sending av 59kbit/s fra Solsiden.



Figur E.56: Variasjon av forsinkelse ved sending av 59kbit/s lydstrøm fra Solsiden.



Figur E.57: Pakketap ved sending av 59kbit/s lydstrøm fra Solsiden.





F

Statistikk

Dato	Måling	Bitrate <sup>a</sup>	Pakketap <sup>a</sup>	Maks.		Maks. forsinkelse
				IPdv	Forsinkelse <sup>a</sup>	
06/05/08	59kbit Kontoret	66.91 kbit/s	0.52 %	265.35 ms	182.16 ms	447.51 ms
14/05/08	59kbit Solsiden	69.29 kbit/s	12.43 %	2187.66 ms	212.66 ms	2400.32 ms
07/05/08	59kbit Torget	66.30 kbit/s	1.30 %	491.48 ms	184.63 ms	676.11 ms
06/05/08	152kbit Kontoret	162.73 kbit/s	0.29 %	368.81 ms	64.66 ms	433.47 ms
14/05/08	152kbit Solsiden	159.88 kbit/s	2.37 %	431.46 ms	66.19 ms	497.65 ms
07/05/08	152kbit Torget	148.95 kbit/s	7.37 %	334.55 ms	71.06 ms	405.61 ms
06/05/08	283kbit Kontoret	304.66 kbit/s	0.22 %	406.25 ms	25.72 ms	431.97 ms
14/05/08	283kbit Solsiden	301.92 kbit/s	10.57 %	745.67 ms	29.16 ms	774.83 ms
07/05/08	283kbit Torget	287.98 kbit/s	5.25 %	465.12 ms	27.14 ms	492.26 ms
30/04/08	1mbit Kontoret	1226.62 kbit/s	0.30 %	284.65 ms	10.24 ms	294.89 ms
14/05/08	1mbit Solsiden	1234.84 kbit/s	32.61 %	765.68 ms	20.67 ms	786.35 ms
07/05/08	1mbit Torget	1269.78 kbit/s	11.34 %	725.73 ms	11.30 ms	737.03 ms
30/04/08	2mbit Kontoret	2069.07 kbit/s	2.71 %	491.76 ms	5.42 ms	497.18 ms
07/05/08	2mbit Torget	1547.93 kbit/s	17.24 %	1425.71 ms	6.41 ms	1432.13 ms
30/04/08	3mbit Kontoret	3543.80 kbit/s	5.31 %	526.40 ms	3.98 ms	530.38 ms
14/05/08	3mbit Solsiden	3501.82 kbit/s	70.47 %	3393.45 ms	399.12 ms	3792.57 ms
07/05/08	3mbit Torget	3106.90 kbit/s	43.57 %	1603.47 ms	7.02 ms	1610.49 ms

Tabell F.1: Samling av statistikk for alle målinger

<sup>a</sup> Verdiene angir gjennomsnitt av resultatsettet for denne metrikken.