



***Analyse av
feil og feilkorreksjonsmetoder
i video over IP systemer***

av

Fred Asplund

**Masteroppgave i
informasjons- og kommunikasjonsteknologi**

**Høgskolen i Agder
Fakultet for teknologi**

**Grimstad
mai 2006**

I Sammendrag

Ved overføring av sanntids video over et IP nettverk vil man bl.a. oppleve tap eller forsinkelse av datapakker i varierende antall, alt fra sporadiske enkeltpakker til store blokker av pakker. Man trenger derfor en metode for å gjenopprette data som er gått tapt eller er forsinket så mye at de må regnes tapt. Flere aktører har benyttet link (satellitt/radiolinje/kabel), SONET/SDH eller ATM baserte systemer for sanntids overføring av video. Dette er systemer hvor man har mulighet for god kontroll på kvaliteten i overføringskanalen. Et av formålene med denne oppgaven var å komme fram til et valg når det gjaldt de tre foreslåtte metodene for feilkorreksjon. Så lenge det er mest sannsynlig at videokontribusjon vil foregå på linjer med QoS garanti, vil Pro-MPEG FEC være et det anbefalte valg.

II Forord

Foranledningen for denne thesis oppgave er et samarbeid mellom T-VIPS AS, Høgskolen i Agder og meg som masterstudent. Oppgaven lyder som følgende:

Redegjør for hvilke feil som kan oppstå i et IP nett og hvilken innvirkning de forskjellige feiltypene vil ha på videosignaler. I dag finnes følgende tre foreslåtte metoder for feilkorreksjon for video over IP nett:

- *DVB IPI (Reed-Solomon FEC)*
- *Pro-MPEG FEC*
- *Raptor.*

Sammenlikn disse metodene ved å beskrive karakteristika, fordeler og ulemper. Det skal deretter innhentes empiriske data for feil i operative nettverk og settes opp karakteristika for disse. Avslutningsvis skal det sees på hvilken av de tre feilkorreksjonsmetodene som er mest egnet i forhold til de feilmønstrene som fremkommer i de empiriske dataene. Dersom tiden tillater det, skal det sees på eventuelle forbedringer som kan gjøres på den foreslåtte korreksjonsmetoden.

T-VIPS AS er et nyetablert firma, hovedsakelig av tidligere ansatte ved TANDBERG Television ASA. Det må rettes en særlig takk til T-VIPS som tok på seg oppgaven som veiledere, heriblant Janne Morstøl og Helge Stephansen, og stilte til disposisjon utstyr, laboratorieplass og mat i kantinen. Det ble også gitt tilgang til testing via internett som gav en mengde interessante data. Det ble også oppdaget at god informering er viktig under testing, da noen uinformerte ansatte nesten fikk ”hjerrestans” da de oppdaget en ”ukjent” utgående datastrøm med høy båndbredde.

Flere medstudenter har også medvirket i tester og stilt til disposisjon sine private bredbåndslinjer. Her må det rettes en takk til Frode Åsen og Morten Trydal, også for bruk av resultater fra et gruppeprosjekt i Formelle Metoder.

III Innhold

I	Sammendrag	2
II	Forord.....	3
III	Innhold.....	4
IV	Figurer.....	6
V	Tabeller	7
1	Innledning.....	8
1.1	Problem beskrivelse.....	8
1.2	Bakgrunn og problemområde.....	8
1.3	Problemstillinger	11
1.4	Avgrensninger.....	11
1.5	Hypoteser	12
2	Oversikt over terminologier og definisjoner	13
2.1	Terminologi	13
2.1.1	Digital TV	13
2.1.2	Sanntidsoverføring av video.....	13
2.1.3	Feilbeskyttelse og feilkorreksjonsmetoder.....	14
2.2	Forkortelser.....	15
3	Litteratur	17
3.1	Digital TV	17
3.1.1	MPEG-2 Video.....	17
3.2	Sanntidsoverføring av digital video	19
3.2.1	MPEG-2 TS.....	20
3.2.2	Sanntids video via RTP/UDP/IP	20
3.3	Nettverk	21
3.3.1	IP	21
3.3.2	UDP.....	21
3.3.3	RTP.....	21
3.3.4	IP i IP tunnel.....	22
3.3.5	Kilderuting	22
3.4	Feil i nettverk	23
3.4.1	Nettverksutstyr	23
3.4.2	Nettverks oppførsel og feil.....	24
3.5	Feilkorreksjonsmetoder.....	26
3.5.1	Bakgrunn og litt historie rundt Error Correction Coding.....	26
3.5.2	DVB IPI feilkorreksjon	27
3.5.3	Pro-MPEG feilkorreksjon	31
3.5.4	Raptor feilkorreksjon.....	35
3.6	Innhenting av empiriske data	38
3.6.1	Reelle målinger	38
3.6.2	Ferdigregistrert statistikk.....	38
4	Empiriske data.....	39
4.1	Metodevalg for innhenting av empiriske data	39
4.1.1	Valg av programvare.....	39
4.1.2	Prøvetest med IP tunnel.....	40
4.1.3	Prøvetest med kilderuting.....	40
4.1.4	Prøvetest via HIA's VPN Server.....	40
4.1.5	Prøvetest fra T-VIPS til HIA.....	40
4.2	Målinger i nettverket til HIA i Grimstad	41
4.2.1	Formål	41
4.2.2	Måleoppsett	41

4.3	Målinger i nettverk - T-VIPS til HIA.....	44
4.3.1	Formål	44
4.3.2	Måleoppsett	44
4.3.3	Målinger 28.04 – 02.05	46
4.3.4	Målinger 05.05 – 08.05	48
4.3.5	Kommentarer.....	49
5	Diskusjon.....	50
5.1	Empiriske data og relevans av målingene.....	50
5.1.1	Standard internett tilkobling.....	50
5.1.2	Høyhastighets internett tilkobling.....	50
5.2	Empiriske data og FEC påvirkning	51
5.2.1	DVB IPI (Reed-Solomon).....	51
5.2.2	Pro-MPEG FEC.....	52
5.2.3	DF Raptor FEC.....	52
6	Konklusjon.....	53
Appendix A	Referanser	54
Appendix B	Linker	55
Appendix C	Protokoll strukturer	56
C.1.1	IP struktur.....	56
C.1.2	UDP.....	56
C.1.3	RTP.....	56
C.1.4	IP i IP tunnel.....	57
C.1.5	Struktur for kilderuting (LSRR og SSRR).....	57
Appendix D	Iperf setup.....	59
Appendix E	Loggfiler	60

IV Figurer

Figur 1: Konseptuelt blokkdiagram av en IPB koder[200].	18
Figur 2 Studio-To-Studio for live broadcast.	19
Figur 3: FEC scheme overview.	27
Figur 4: RS(204,188) encoding.	27
Figur 5: Basic Forney interleaver.	28
Figur 6: IP packet format with RTP and FEC (IPv4).	29
Figur 7: Overføringstest ved HiA.	41
Figur 8: Test fra T-VIPS lab til HiA lab.	44

V Tabeller

Tabell 1: Maximum constraint parameters	17
Tabell 2: Interleaving modes	28
Tabell 3 Timeslot formula for FEC-L generation	33
Tabell 4 Timeslot formula for FEC-D generation.....	33

1 Innledning

1.1 *Problem beskrivelse*

Oppdragsgiver har behov for en metode som kan håndtere de feilmønstre som er typiske for IP nettverk som benyttes til sanntids distribusjon og kontribusjon av høykvalitets video. Ved overføring av sanntids video over et IP nettverk vil man bl.a. oppleve tap eller forsinkelse av datapakker i varierende antall, alt fra sporadiske enkeltpakker til store blokker av pakker. Man trenger derfor en metode for å gjenopprette data som er gått tapt eller er forsinket så mye at de må regnes tapt data. I tillegg bør ikke metoden innføre for mye ekstra forsinkelse i systemet.

1.2 *Bakgrunn og problemområde*

Problembeskrivelsen har sin bakgrunn i at man i større grad enn tidligere vurderer sanntids overføring av høykvalitets video over internett systemer. Dette gir både praktiske og kostnadmessige fordeler hos sendersiden og mottagersiden, samtidig som det også skaper nye problemstillinger som må løses. For profesjonelle aktører har videokvalitet, stabilitet og pålitelighet høy prioritet. Muligheten for å påvirke dette kan bl.a. gjøres ved å benytte metoder for feilbeskyttelse av data (ECC), bruk av reserve systemer, tilpassning av overføringen mellom hver link m.m.. Disse aktørene har ofte benyttet OSI lag-1 og lag-2 baserte systemer som har gitt disse mulighetene.

Internett systemer har i lengre tid vært benyttet som overføringssystem for ikke-sanntids baserte applikasjoner. Veksten innenfor dette området har de siste årene vært sterk og ser ut til å fortsette videre fremover. Utbyggingen av internett har gitt bedre tilgang til høy båndbredde, tilgjengelighet og et stort utvalg av utstyr. Dette har igjen medført lavere transportkostnader og utstyrs kostnader, noe som har aktualisert bruken av internett til sanntidstjenester.

T-VIPS AS ser på forskjellige metoder for feilkorreksjon og benytter i dag en anbefaling fra Pro-MPEG Forum for overføring av videoformatene MPEG-2 og SDI. I begge tilfeller benytter man seg av RTP for transport over internett. Kundegruppen er hovedsakelig profesjonelle aktører som har et høyt krav til kvalitet. Dette setter klare grenser til systemet for hva som er akseptabelt av data tap, forsinkelse og ustabilitet. Ved valg av metode for å opprettholde original kvalitet ser man også på andre sider enn bare den teoretiske siden. For å

kunne gjennomføre løsningen må en også bl.a. ta hensyn til kostnader og plassforbruk ved praktisk gjennomføring.

Profesjonelle aktører vil stå ovenfor to typer overføringskanaler når man vil gjøre sanntids overføring av video. Dette er noe man må ta hensyn til når man velger overføringsprotokoll. Ved overføring på OSI lag-1 og lag-2 vil man i stor grad ha tilgang til å regulere kvaliteten mellom hvert enkelt ledd. Flere aktører har benyttet link (satellitt/radiolinje/kabel), SONET/SDH eller ATM baserte systemer for sanntids overføring av video. Dette er systemer hvor man har mulighet for god kontroll på kvaliteten i overføringskanalen. Ved bruk av link kan man for eksempel endre modulasjonsform, øke antennestørrelsen eller forbedre kabel kvaliteten for å redusere feilhyppighet. Man har mulighet for å gjøre feilretting på bit nivå som igjen gir mulighet for lavere forsinkelse i systemet. Ved bruk av ATM, som er et forbindelsesorientert system i motsetning til IP, kjenner man til kvalitets parametere og man har mulighet for å velge en primær kvalitetsparameter

Ved overføring på transport eller nettverkslaget (OSI lag-3 og lag-4) har man stort sett bare mulighet for kvalitetsregulering i endepunktene. Avhengig av hvilken kontroll man har over hver enkel link, kan man velge feilbeskyttelsesmetoder som garanterer kvaliteten på mottagersiden. For uavhengige internett systemer (for eksempel ved transport gjennom flere tjenesteleverandører) må det settes høyere krav til overføringsprotokollene for å garantere kvaliteten enn for lukkede systemer hvor man har kontroll med hver enkel link i systemet og som dermed kan gi de kvalitetsgarantier som distributørene krever.

IP nettverk betegnes som "best effort" nettverk, det gjør det beste det kan for å overføre dataene. Imidlertid ved å benytte seg av enkelte reguleringsmekanismer som IntServ og DiffServ kan tjenesteleverandørene tilby en form for kvalitetstjeneste. Dette er imidlertid ikke godt nok og for å tilfredsstille de krav som stilles til kringkastingskvalitets video må det i systemet integreres feilkorreksjonsmetoder som vil utligne den kvalitetsreduksjon som nettverket innfører.

Overgangen fra et system til et annet medfører ofte at man må generere tilleggs protokoller for å ivareta tidligere kvalitet. Et resultat kan være at distributørene må lempe på kravene som stilles for å kunne ta i bruk et nytt system som totalt sett er mer fordelaktig. Avveininger som

her må tas hensyn til er for eksempel videokvalitet, aksesskontroll, distribusjonsområde, distribusjonstid og kostnader.

Vi har et problemområdet hvor vi skal gjennomføre datakommunikasjon i en overføringskanal med stokastiske feil og innføre metoder for feilkorreksjon basert på sannsynligheten for spesielle feilmønstre. Datakommunikasjonen følger OSI modellen og det er mulig å gjøre analyse på de forskjellige lagene og få statistikk på feil hvilke konsekvenser feil får.

1.3 Problemstillinger

Ut ifra den problembeskrivelsen vi har, må vi se på følgende forhold:

- Virkningen av feil
- Overføringskanalen
- Feilkorreksjonsmetoder

1: Hva slag sammenheng er det mellom feil i overføringskanalen og feil i video?

Komprimert video og ukomprimert video vil reagere ulikt på pakketap og således gi forskjellig kvalitets forringelse. Hvilke typer feil, som opptrer i IP systemer, er vesentlige?

2: Hvordan definere en overføringskanal som er relevant for undersøkelsen og som vi dermed kan innhente empiriske data fra? Ved korte distanser er de ukjente parametrene relativt få og mulig å klassifisere. Ved lengre distanser vil vi ha mindre mulighet for å kartlegge feilkildene. Vil vi ved å analysere mulige feilkilder i nettverket kunne forutsi feilsannsynlighet og feilmønstre. Hva med forskjellige typer nettverk som kan bli benyttet og hvordan nettverket er bygd opp?

3: Hvordan innhente relevante empiriske data og hvor stor er gyldigheten av disse? Finnes det metoder/protokoller for statistisk innsamling av slike data til dette formålet?

4: Hvilke krav må vi tilfredsstille når det gjelder feilkorrigerende og høykvalitets video? Hvilke krav gjelder for kringkastingskvalitet video? Hvilke feil klarer de nevnte metoder å korrigere? Er det nødvendig å forbedre algoritmene for nevnte metoder for feilkorreksjon

1.4 Avgrensninger

Medieformater som behandles og hva som ikke skal sees på.

Det finnes et stort utvalg av forskjellige digitale videoformater. I denne oppgaven vil vi kun se på et komprimert videoformat, MPEG-2 Video. Innenfor MPEG-2 Video er det også flere grader av nivåer og profiler (kombinasjoner av ML/MP, ML/HP, HL/MP og HL/HP som angir hhv fargekvalitet og oppløsning) og kompresjonsnivåer. Det vil ikke bli behandlet spesielt, annet enn som betydning for feilfølsomhet ved tapte data.

Overføringsmetoder som benyttes og hvilke som utelukkes.

Som overføringsmetode vil vi kun se på sanntids overføring av høykvalitets sanntids video og ikke IPTV til DTH.

Valg av overføringskanal og hvilke som ikke behandles

For kontribusjon vil vi ta for oss et IP system som består av IPv4 ethernet basert utstyr (rutere og svitsjer) og som håndterer RTP/UDP/IP (ikke fragmentering av MTU < 1500)

Feiltyper som behandles og hva som ikke behandles.

Selv om jitter og forsinkelse er en viktig parametere, vil vi kun ha fokus på feil som er pakketap relaterte (som pakketap, pakker utenfor rekkefølge).

Feilrettingsmetoder som velges og hvem som ikke behandles.

Vi vil se på 3 feilrettingsmetoder, DVB IPI, Pro-MPEG FEC og DF Raptor FEC

Avgrensning av empirisk data innsamling (ex data vil ikke omfatte trådløse nett)

Primært er det ønskelig med data fra et live nett. Dette er noe vanskelig da man trenger nokså detaljert statistikk. Testing på ordinære nett og på HIA sitt interne nett er det mest aktuelle.

Trådløse nett vil ikke bli omfattet her.

1.5 Hypoteser

De feil som vil være av avgjørende betydning er blokker av pakketap som i størrelse vil være umulig å gjenopprette. Jo lengre en sti er jo større er sannsynligheten for at dette skjer.

Innhenting av data skal kunne gjøres ved å ha et punkt hvor en både sender og mottar. IP protokollen åpner for både IP-i-IP innkapsling og kilderuting.

Det er trolig at alle tre FEC metodene vil virke tilfredsstillende i et lukket og kontrollerbart system, mens de alle vil feile i et åpent ukontrollerbart system. Buffer størrelsen i ruterne og feilprogrammering av rutere er de punktene trolig vil forårsake de største problemene.

2 Oversikt over terminologier og definisjoner

2.1 Terminologi

2.1.1 Digital TV

MPEG-2 TS

En pakkebasert datastrøm for overføring av datatjenester. Benyttes av både DVB og ATSC. Protokollen er spesifisert i ISO/IEC 13818-1. Hver pakke kan være på 188 eller 204 bytes. En pakke på 204 bytes inneholder 16 bytes som brukes til feil deteksjon og korreksjon. Dataratene kan variere fra 50 kbit/s to 100 Mbit/s

MPEG-2 Video

En standard for komprimering av video som er spesifisert i ISO/IEC 13818-2 og som begynte å bli tatt i bruk rundt 1990. Man benytter hovedsakelig to kvalitetskategorier, 4:2:0 for vanlig distribusjon og 4:2:2 profesjonell distribusjon. Innenfor disse to gruppene kan man velge mellom forskjellige horisontale og vertikale oppløsningsvarianter, bl.a. 704h x 576v, 352h x 288v.

Kringkastingskvalitet Video

En video definisjon som i de fleste tilfeller betegner den høyeste kvaliteten på video og som benyttes for kontribusjon (studio til studio). Den er imidlertid ikke fast definert, så den benyttes også som produktreklame, uten felles forankring.

2.1.2 Sanntidsoverføring av video

MPEG-2 Video via ATM

Sanntidsoverføring via ATM gjøres ved å dele opp MPEG-2 pakkene for å tilpasse dem ATM pakker som er ca 1/4 del av en MPEG-2 pakke. Man kobler opp en virtuell forbindelse mellom endepunktene som kobles ned når sanntidsoverføringen er ferdig.

2.1.3 Feilbeskyttelse og feilkorreksjonsmetoder

DVB IPI FEC

En protokoll (IPI2001-016) for feilkorreksjon som er utviklet av DVB. Metoden er basert på byte interleaving av MPEG-2 204 bytes pakker hvor en utnytter Reed-Solomon's feilbeskyttelses egenskaper, som er en del av MPEG-2 spesifikasjonen [4].

Pro-MPEG FEC

En protokoll for å regenerere tapte RIP pakker. Et bestemt antall pakker settes opp i en matrise og så beregnes det et sett med nye pakker basert på XOR operasjoner på vertikale og/eller horisontale rader i matrisen. Protokollens hovedfokus er MPEG-2 TS pakkeoverføring via IP nettverk for kontribusjon og ikke DTH.

DF Raptor FEC

A protokoll for feilkorreksjon utviklet av Digital Fountain [8] som bygger på en distribusjons metode av hvert symbol slik at når man har mottatt et bestemt antall symboler, skal man være i stand til å gjenskape den originale datastrømmen.

2.2 Forkortelser

DVB	Digital Video Broadcasting
IPI	(DVB) IP Infrastructure
FEC	Forward Error Correction
MPEG	Moving Picture Expert Group
MTS	MPEG-2 Transport Stream
IP	Internet Protocol
RTP	Real-time Transport Protocol
UDP	User Datagram Protocol
TS	Transport Stream
ECC	Error Correction Coding
MTU	Maximum Transmission Unit

3 Litteratur

3.1 Digital TV

3.1.1 MPEG-2 Video

MPEG-2 Video tilbyr et rikt utvalg av formater, komprimering

Level	Profile					
	Simple I,P 4:2:0	Main I,P,B 4:2:0	4:2:2 I,P,B	SNR I,P,B 4:2:0	Spatial I,P,B 4:2:0	High I,B,P 4:2:0 or 4:2:2
High		1920x1152 60fps 80Mb/s				1920x1152 60fps 100Mb/s
High 1440		1440x1152 60fps 60Mb/s			1440x1152 60fps 60Mb/s	1440x1152 60fps 60Mb/s
Main	720x576 *30fps 15Mb/s	720x576 30fps 15Mb/s	720x608 30fps 15Mb/s	720x576 30fps 15Mb/s		720x576 30fps 20Mb/s
Low		352x288 30fps 4Mb/s		352x288 30fps 4Mb/s		

**Tabell 1: Maximum constraint parameters
for MPEG-2 levels and profiles [200]**

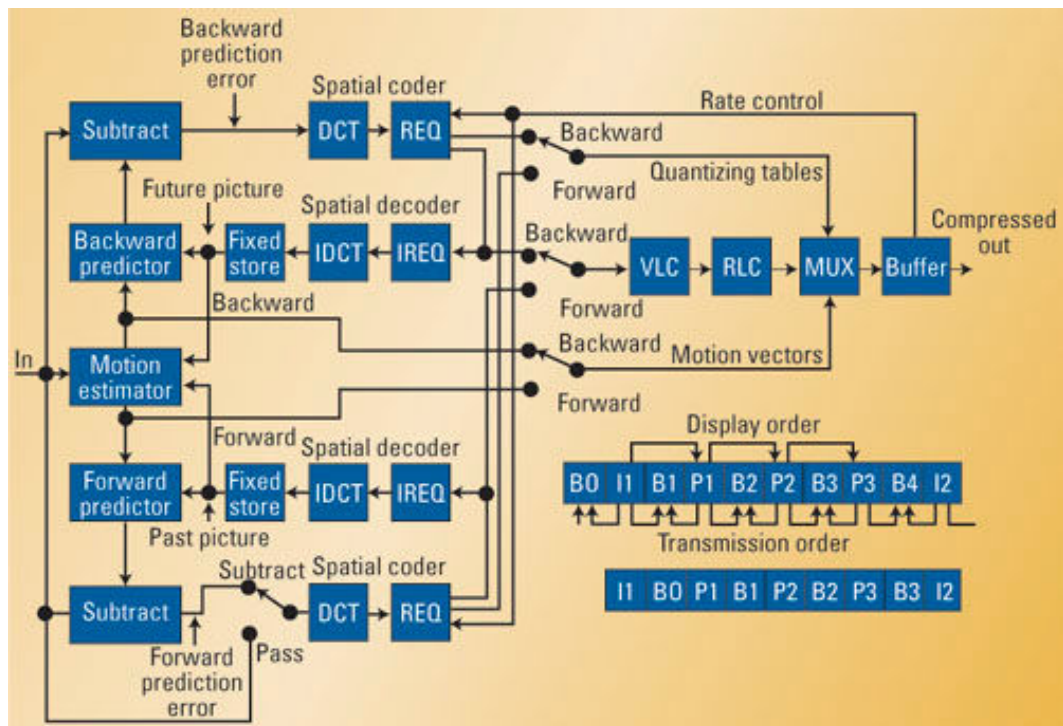
Prinsippet med bilde oppsett er likt for disse og består av følgende varianter:

- Video strømmen inneholder I- og B-bilder
- Video strømmen inneholder I-, B- og P-bilder

I-bilder er basert på en svak komprimering av et uavhengig bilde i full oppløsning, uten fragmenter fra andre bilder. Dette bildet er kritisk for oppbygningen av en GOP sekvens som består av I-, P- og B-bilder. Feil i dette bildet vil være svært synlig.

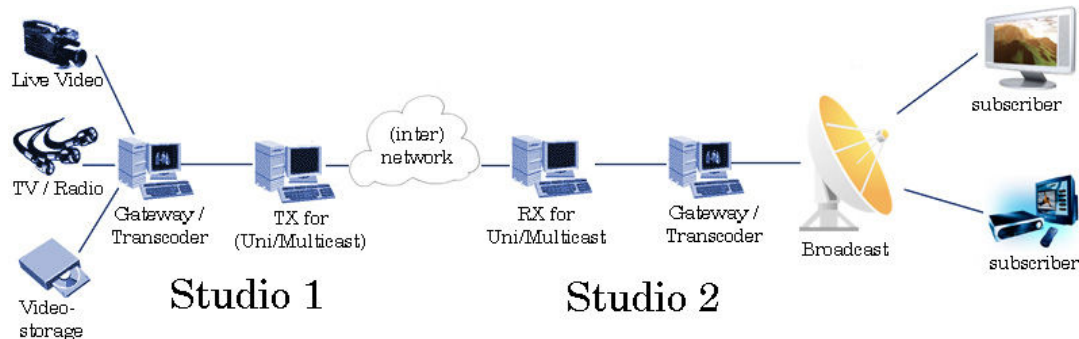
P-bilder er basert på forutsigelse av et tidligere I-bilde. Det har hardere komprimering enn I-bildene. De er noe mindre kritiske for oppbygningen av en GOP sekvens

B-bilde er basert på forutsigelse av et tidligere og fremtidig I- eller P-bilder. De er noe minst kritiske for oppbygningen av en GOP sekvens. Se figur.1 [200].



Figur 1: Konseptuelt blokkdiagram av en IPB koder[200].

3.2 Sanntidsoverføring av digital video



Figur 2 Studio-To-Studio for live broadcast

Sanntidsoverføring av video og bruken av FEC er mye behandlet når det gjelder lag 2 transmisjon (ATM, SONET/SDH og satellittsystemer). På lag-3 transmisjon er det mindre litteratur. Grunnen til at IP nettverk ikke er benyttet skyldes først og fremst mangelen på QoS men også tilgangen på bredband. Det er i den senere tid at bredband er blitt mer tilgjengelig og dermed aktualisert temaet [20]. Det er imidlertid studier som viser at det er delte meninger[1] om effekten av FEC. I enkelte tilfeller er det mer aksept for tapt data enn forsinkelse.

På lag 1 i OSI modellen vil det være mulig å regenerere tapte bits mens det på lag 3, nettverkslaget, vil være nødvendig å regenerere hele pakker eller blokker av pakker. For å gjøre denne regenereringen benyttes det FEC. Dette gjøres ved at det legges ekstra data til den originale datastrømmen på sendersiden og som på mottagersiden benyttes til å regenerere eventuelle tapte datapakker. Dette skaper en ekstra overhead og/eller forsinkelse. Begge deler er i utgangspunktet uønsket og man søker å holde dette på lavest nivå. I tillegg må man ta hensyn til materialkostnader som eventuelle FEC algoritmer vil innføre. Ved sanntidsoverføring av video er det mest vanlig å benytte UDP. Ved bruk av UDP unngår man retransmisjon av tapte data og man unngår trafikkregulering slik TCP blir utsatt for.

3.2.1 MPEG-2 TS

MPEG-2 Transport Stream (MTS) har normalt vært overført på lag 1 og 2 teknologi, som for eksempel satellitt overføring, kabel-tv og ATM, som har egne protokoller feilkorreksjon. Her kan nevnes systemer som benytter Inner FEC og Outer FEC for feilretting. ATM har sin egen protokoll for sikker overføring (QoS). For å overføre MTS over ATM må man fragmentere MTS pakker, som er 188 eller 204 bytes, ned til 48 bytes pakker.

MTS benytter seg av Reed-Solomon (RS(204,188)) og en 4 bits kontinuitets indeks for feilkorreksjon og pakketaps deteksjon. Dette er en svakhet med tanke på at man på høyere lag opererer med pakketap og eventuelt tap av blokker av pakker. RS vil kun korrigere 8 eller 10 bytes i en MTS pakke og en 4 bits teller tar kun høyde for tap maks 15 MTS pakker.

3.2.2 Sanntids video via RTP/UDP/IP

David A. Rosenthal beskriver i en analyse [25] hvilke variable som er funnet å påvirke kvaliteten på sanntidsoverføring av video over IP nettverk. Det foretas målinger av Peak Signal-to-Noise Ratio (PSNR) og Picture Quality Rating (PQR). De variable som er tatt hensyn til er de som typisk kommer som følge av metningskontroll, nemlig pakketap, pakkeforsinkelse, jitter og båndbreddebegrensninger. Han konkluderer med at kvaliteten på video varierer med grad av påførte feil, tilgjengelig båndbredde og codec konfigurasjon.

3.3 Nettverk

Vi har valgt kun å fokusere på kabel/fiber basert nettverk. Tap på lag 1 vil da i realiteten være utelukket eller så små at man kan se bort fra det. For fiber regner man med en BER på $10E-9$

3.3.1 IP

RFC791 beskriver IP struktur vi vil benytte oss av (se Appendiks C for detaljert oppbygning):

De parametrene som blir benyttet litt i utvidet betydning her er:

- "IHL" som må utvides når vi benytter opsjoner
- "Protocol" som blir brukt som UDP (0x11) eller IP i IP innpakning (0x04)
- "Options" som blir benyttet under kilderuting.
- "Padding" som blir benyttet til å avslutte IP headeren slik at vi alltid opererer med n antall 32 bits ord

3.3.2 UDP

RFC768 beskriver strukturen på UDP. I denne oppgaven er det ingen særskilt bruk utover det vanlige. UDP brukes i kombinasjonen med RTP.

3.3.3 RTP

RFC1889 beskriver RTP strukturen.

RFC2250 som grunnlag når MPEG-2 sendes via RTP. De første kontroll bitene settes til følgende:

- V (Version) = 2
- P (Padding) = 0
- X (Extension) = 0
- CC (CSRC count) = 0
- M (Marker) = 0
- PT (Payload Type) = 33
- Sequence No = stepping med 1
- Timestamp = 90kHz oppløsning (RFC2250)

(d.v.s. at RTP strømmen er 12 bytes lang, starter med 0x8021 og har ingen CSRC)

3.3.4 IP i IP tunnel

RFC791 definerer en protokoll parameter som definerer innholdet av IP pakken. Denne parameteren blir lest av rutere som bruker dette i en avgjørelse. Ved å legge til en ekstra IP header med protokoll type 4, vil ruterens lese header 1 og se at den sende pakken videre til adressen definert i header nr 2. Header 1 vil bli fjernet før pakken videre sendes.

Dette er en metode som blir benyttet i forbindelse med mobil IP trafikk.

3.3.5 Kilderuting

RFC791 definerer to varianter av kilde ruting (løs og fast kilde ruting):

```
Options:
  Loose Source and Record Route (LSRR)
  +-----+-----+-----+-----+-----+//-----+
  |10000011| length | pointer|   route data   |
  +-----+-----+-----+-----+-----+//-----+
  Type=131

  Strict Source and Record Route (SSRR)
  +-----+-----+-----+-----+-----+//-----+
  |10001001| length | pointer|   route data   |
  +-----+-----+-----+-----+-----+//-----+
  Type=137
```

- "length" er den totale lengden med 8 bits oppløsning (oktett). Dvs. hvis vi har 2 IP adresser i "route data" så vil lengden ha verdi 4.
- "pointer" skal peke på neste hopp adresse etter "Destination Address". Pekeren har 8 bits oppløsning, d.v.s. at første IP adresse vil ha verdi 4, neste adresse vil verdi 8 o.s.v. Hvis pekeren har større verdi enn lengden, så skal det tolkes som om at "Destination Address" er neste IP adresse.
- "route data" er IP adresser til rutere/gatewayer som IP pakken må via før den når endepunktet. Ved løs kilde ruting (LSSR) så spesifiserer man de av adressene IP pakken må innom, altså ikke hele stien, før IP pakken kommer fram. Det er dermed ikke gitt at alle pakkene følger samme sti til endepunktet ved løs kilde ruting (LSSR) i motsetning til fast kilde ruting (SSRR) hvor man må spesifisere hele stien. Hvis "pointer" har lavere verdi enn "length", så vil IP pakken når den kommer til noden med "Destination Address" få kryssbyttet adressen pekeren peker på og "Destination Address". I tillegg vil pekeren bli endret til å peke på neste adresse

IP opsjoner avsluttes 0x00

3.4 Feil i nettverk

3.4.1 Nettverksutstyr

For å danne seg et bilde av hvilke typer feil som kan oppstå i IP nettverk, trenger man en oversikt over hvordan nettverket er bygd opp. Vi vil se på forbindelsene mellom de tre nederste lagene som er beskrevet i OSI modellen og utstyr som benyttes til kommunikasjon mellom lagene. For to av lagene, fysisk lag og linklaget, har man en relativt god kontroll på overføringskvaliteten. Det tredje laget, nettverkslaget, er mer komplekst og utsatt for trafikkregulering og omprogrammering[10] og er derfor mer uforutsigbart. Man kjenner ikke til hvilke lavere lags systemer som benyttes og er prisgitt det svakeste leddet.

Forbindelser på fysisk lag

De mest brukte forbindelsene mellom punkt til punkt forbindelser i IP nettverk er optisk fiber og ethernet kabler. For optisk link regner man med en BER tilsvarende $10E-9$ [101],

Lokale nettverk og intranett

Lokale nettverk er som regel bygd opp av et antall switcher og evt. rutere. Operatørene har en relativt god kontroll med systemet og kan lett gjøre trafikkanalyser og regulere nettverkstrukturen for best mulig effektivitet. Man kan tilrettelegge for spesiell trafikk og sørge for minimale feilkilder

Internett

Internettverk er bygd opp av et ukjent antall ruternettsverk som i viss grad styres uavhengig av hverandre. En operatør vil kunne styre sitt autonome system (AS) og ha en kontroll over trafikken som er delt opp i kjernetrafikk og grensetrafikk. Ut ifra dette vil det være mulig å tilby en viss kvalitet på trafikktenester (QoS) til andre autonome systemer (AS). Mellom ruterne vil det utveksles statusinfo (RIP). Enkelte endringer i denne informasjonen vil føre til rutertabeller beregnes på nytt og dette gjør at vi har et dynamisk bilde hvor QoS kan variere når det gjelder jitter og forsinkelse, sekvensfeil og pakketap.

Traffic-engineering er en kompleks oppgave hvor man konfigurerer sitt ruter system for å regulere trafikken både i kjernen og i grenseruterne. Dette kan medføre at man i enkelte

situasjoner er inkompatibel med andre nettverk i kortere eller lengre tid ved at man ikke er i stand til å håndtere all trafikk inn i systemet eller gjennom systemet.

3.4.2 Nettverks oppførsel og feil

Fysisk kommunikasjon

Kabeltilkobling mellom 2 enheter.

Ved direkte tilkobling benyttes det enten elektrisk eller optisk. For elektrisk tilkobling er lengde og skjerming av kabel avgjørende elementer. Over lange distanser vil man måtte regenerere dataene opp på lag 2 flere ganger for å unngå datatap. Alternativt vil man måtte øke dimensjonene på kabelen. Innstråling fra andre signalkilder vil kunne skape bitfeil.

Optiske kabler regnes som en relativt sikker tilkobling. Her er det degradering av selve kabelen som kan være feilkilde.

Bruk av HUB.

Signalsplitting hvor alle som er tilkoblet "ser" alle de andre hele tiden. Flere vil kunne sende samtidig og dermed skape datakollisjon

Datalink kommunikasjon

Bruk av ethernet switch.

Her vil vi ha punkt til punkt link. All trafikk blir behandlet på en separat databuss og effektiviteten vil være avhengig av hastigheten på denne prosesseringen. Tilkoblede enheter blir registrert i en tabell basert på MAC adresser og data mellomlagres midlertidig i et buffer. Multicast og data til adresser som ikke er registrert i oppslagstabellen blir sendt på alle porter (unntatt datakildeporten). Ved høy trafikk på alle porter vil en algoritme avgjøre hvilke pakker som forkastes.

Nettverks kommunikasjon

Bruk av single ruter.

Her vil det være punkt til punkt link. Tilkoblede enheter blir registrert i en tabell basert på IP adresser. Data til og fra disse portene blir mellomlagret i buffer. Mulighet for konfigurering av avansert filtrering av data trafikk kan skape feilsituasjoner. Muligheten til å endre MTU til lavere enn forutsatt for videotrafikk er en kritisk faktor her.

Bruk av ruternetverk.

I et ruternetverk vil det være mulighet for konfigurering av avansert filtrering av data trafikk for hver enkelt ruter i systemet. Dette kan skape inkompatibilitet med resten av rutersystemet.

Vi kan ha lange distanser som gir større forsinkelser og evt. mer jitter.

Pakker i feil rekkefølge (sekvensfeil) kan oppstå som følge av at ikke alle pakkene følger samme vei hele tiden og i noen tilfeller vil disse måtte oppfattes som pakketap.

3.5 Feilkorreksjonsmetoder

3.5.1 Bakgrunn og litt historie rundt Error Correction Coding.

FEC er mye omtalt både i faglitteratur for digital kommunikasjon [22] og matematikk [23] og er et stadig tilbakevendende tema siden kvalitetskravene for kommunikasjonskanaler endrer seg som følge av nye bruksmønstre (for eksempel sanntids overføring av data). I denne litteraturen finner man metoder for flere varianter av feilkorreksjon, deriblant de som benyttes i DVB IPI og Pro-MPEG FEC..

Feilkorreksjon kan deles inn i flere metodegrupper avhengig om de er basert på sannsynlighetsberegning eller har faste algoritmer..

- Near optimal erasure codes
- Near optimal fountain (rateless erasure) codes
- Optimal erasure codes

Det finnes en rekke metoder for feilkorreksjon og noen av de stammer fra tidlig på 1900 tallet.

En kort oversikt viser noen av metodene som finnes:

- Reed-Solomon
- Forney interleaver
- LDPC
- Hamming
- Turbo codes
- Fountain codes
- LT codes
- Raptor codes

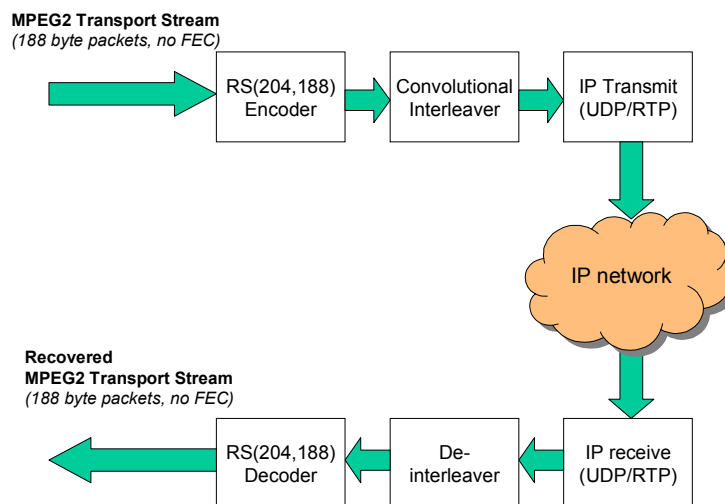
Både ARQ, som baserer seg på retransmisjon, og FEC som baserer seg på en enveis kanal. Ved sanntidsoverføring av video benytter man seg av UDP siden det ikke er ønskelig med retransmisjon. Man kan i stedet lage en kontrollerbar ekstrastrøm av FEC data som benyttes til feilkorrigerende

3.5.2 DVB IPI feilkorreksjon

Beskrivelse

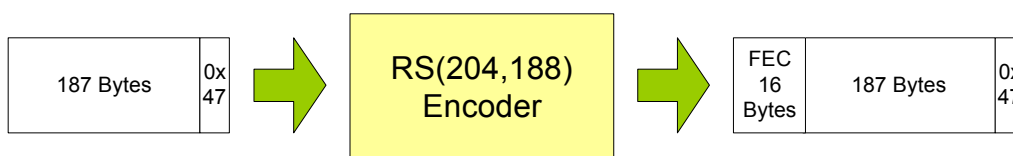
DVB IPI protokollen tar for seg levering av DVB MPEG-2 tjenester til DTH over IP nettverk. Som en del av dette beskriver DVB IPI-016 en FEC implementering som er basert på en revisjon av en Forney interleaver modell. For å kunne håndtere pakketap, sekvensfeil og dupliserte pakker må man benytte en protokoll med sekvensnummerering slik RTP har.

Figur 2 nedenfor, som er hentet fra DVB IPI dokumentet, viser en prinsippskisse for systemet.



Figur 3: FEC scheme overview

Modellen vil ta i bruk feilkorreksjonsegenskapene til MTS. For feilkorreksjon benytter MTS seg av Reed-Solomon (RS(204,188)). Hvis man kjenner feilposisjonene kan man med en iterativ prosess kunne rette opp til 16 bytes, ellers vil det være mulig å rette opptil 8 bytes. Siden vi med denne metoden vil kjenne til feilposisjonene vil vi kunne dra nytte av maksimal feilretting.



Figur 4: RS(204,188) encoding

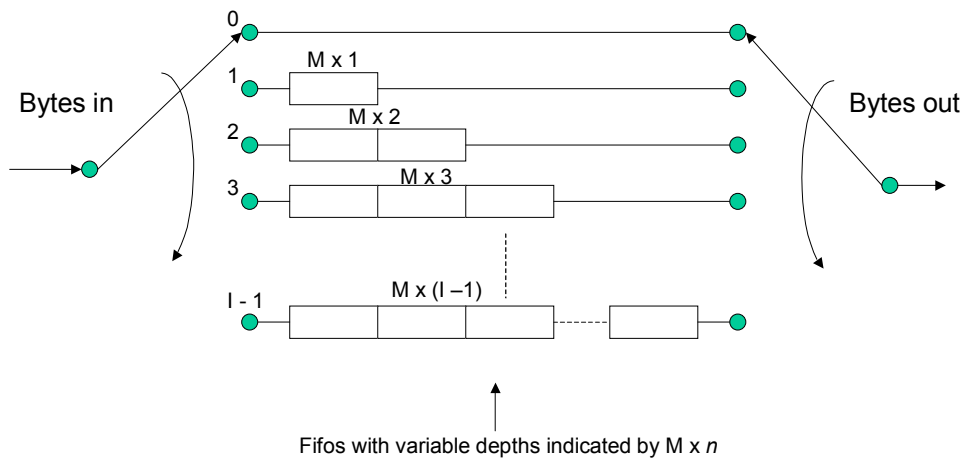
Tabellen fra DVB IPI-016 spesifikasjonen viser følgende:

Mode #	N	M	I	Max. burst length (bytes)		Buffer req. (bytes)	Approx delay at 4Mbit/s (seconds)
0	204	4	51	408		10200	0,02
1	204	2	102	816		20604	0,04
2	204	1	204	1632		41412	0,08
3	408	2	204	3264		82824	0,17
4	612	3	204	4896		124236	0,25

Tabell 2: Interleaving modes

N Number of bytes in the block to be distributed
M Each branch j shall be a FIFO shift register of depth $j \times M$ bytes
I Interleaver depth

Figur X til Y er hentet fra DVB IPI-016 spesifikasjonen.



Figur 5: Basic Forney interleaver

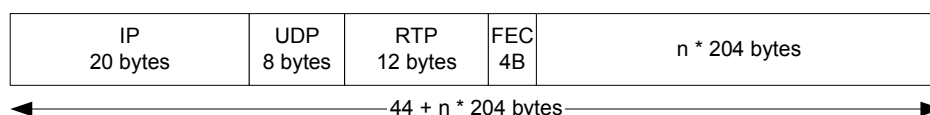
For Mode 0 vil en 204 bytes MTS pakke være spredt ut i tid utover 51 MTS pakker.

- Pakke(0) vil inneholde Byte(0,51,102,153)
- Pakke(1) vil inneholde Byte(1,52,103,154)
- o.s.v.
- Pakke(50) vil inneholde Byte(50, 101, 152, 203)

For Mode 4 vil en 204 bytes MTS pakke være spredt i tid utover 612 MTS pakker

- Pakke(0) vil inneholde Byte(0)
- Pakke(3) vil inneholde Byte(1)
- o.s.v.
- Pakke(611) vil inneholde Byte(203)

Det benyttes i de fleste tilfellene 7 MTS pakker pr IP pakke som vil gi en IP pakke like under 1500 bytes som er normal MTU i IPv4 nettverk.



Figur 6: IP packet format with RTP and FEC (IPv4)

DVB IPI2001-016 er først og fremst tenkt mot DTH markedet men tar også litt høyde for distribusjon og kontribusjon. Den var tenkt som en del av ETSI TS 102034 protokollen men er foreløpig ikke godkjent. Den er implementert i enkelte produkter for distribusjon av DVB MTS.

Egenskaper

Mode 0:

Håndterer blokkta på 4 MTS pakker med en avstand på 51 pakker. En IP pakke bør maks inneholde 4 MTS pakker ellers vil kun ett IP pakketap medføre feil i MTS strømmen.

Mode 1:

Håndterer blokkta på 8 MTS pakker med en avstand på 102 pakker

Mode 2:

Håndterer blokkta på 16 MTS pakker med en avstand på 208 pakker

Mode 3:

Håndterer blokkta på 32 MTS pakker med en avstand på 408 pakker

Mode 4:

Håndterer blokkta på 48 MTS pakker med en avstand på 612 pakker

Fordeler

Ingen lisensavgifter (royalties)

Ulemper

Har vist seg å være ineffektiv for retting av blokkfeil[1].

Man er nødt til å endre original MTS. Dekodere som ikke kan IPI dekoding, vil ikke kunne motta denne type data.

Bruker my regnekapasitet ved pakketap.

3.5.3 Pro-MPEG feilkorreksjon

Beskrivelse

Pro-MPEG FEC er en protokoll for feilkorreksjon ved overføring av MPEG-2 TS (MTS) over IP nettverk. Protokollen er et produkt av Pro-MPEG Forum som ble dannet for å skape en felles plattform med interoperabilitet mellom utstyr fra forskjellige produsenter, laget for kontribusjon og distribusjon av video. Forumet er et ikke-kommersielt forum hvor interessenter kan både delta og bidra.

Pro-MPEG FEC er en videreutvikling av protokollen RFC2733 "Generic FEC for RTP Payload Format". RFC2733 har en del begrensninger, bl.a. er området for en FEC pakke begrenset til maks å gjelde 24 sammenhengende RTP pakker. For å kunne ta høyde for større blokktao og en del andre elementer er Pro-MPEG Code of Practice 3 (CoP #3) blitt spesifisert.

I tillegg til å videreutvikle RFC2733 definerer CoP #3 et rammeverk for bruken av protokollene for MPEG-2, IP, UDP og RTP. Det er definert et eget sett av parametere og format regler for hva som skal benyttes i pakkehoder og pakkeoata.

Selve FEC beregningen gjøres ved å sette opp 1- eller 2-dimensjonale matriser av RTP pakker. Så dannes det en ny pakke ved å gjøre en XOR operasjon på en byte fra hver pakke på rader og/eller kolonner.

Eks. $FEC1B1 = P1B1 \text{ xor } P2B1 \text{ xor } P3B1 \text{ xor } P4B1$

Dersom man taper en pakke i for eksempel en kolonne, så kan man regenerere den tapte pakken ved å gjøre XOR operasjon på de resterende pakkene pluss tilhørende FEC pakke.

Ved bruk av den endimensjonale løsningen vil man være i stand til å korrigere inn et pakketap pr blokk. Ved bruk av den todimensjonale løsningen vil man i de gunstigste tilfellene være i stand til å korrigere inn L+D-1 pakketap pr blokk ved bruk av iterativ korreksjon. I de mest ugunstige tilfeller vil man bare kunne håndtere 3 tapte pakker. Dette som følge av at paritets beskyttelse ikke kan håndtere 2 feil innenfor samme blokk..

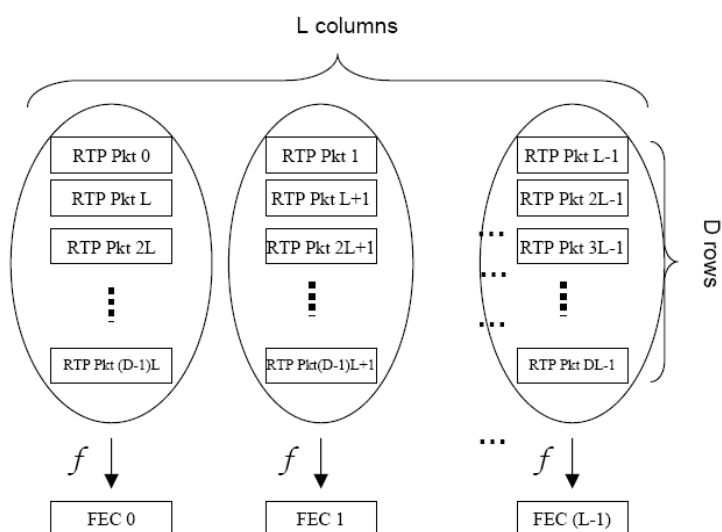


Fig.1 Encoding scheme ...

Som minimums krav må implementeringer støtte følgende 3 krav til valg av størrelse på kolonner og rader i matrisen:

- $L * D \leq 100$
- $4 \leq L \leq 20$
- $0 \leq D \leq 20$

Under en presentasjon på PBS 2005 Technology Conference presenterte Philippe Lemonnier fra Thomson[] følgende statistikk:

Typical performance

In seconds
In days !

Reference :
Video at 4Mb/s transported with 7 MPEG-2 TS packets per RTP/IP datagram

Legend:
 L : matrix row length
 D : matrix column depth
 I : Interleaving depth used in FEC packets sequencing
 PLR : Network Packet Loss Ratio
 MTBE : Mean Time Between Errors

Error distribution: random/uniform

L	D	I	PLR	overhead	latency (ms)	MTBE without FEC (sec)	MTBE with FEC (day)		
10	10	10	1,E-03	20%	525	2,74	32,87	} 2D FEC	
10	10	10	1,E-04	20%	525	27,40	42105,50		
10	10	10	1,E-05	20%	525	274,00	42336213,68		
5	5	5	1,E-03	40%	127	2,74	35,13		
5	5	5	1,E-04	40%	127	27,40	35431,83		
5	5	5	1,E-05	40%	127	274,00	33688413,06		
L	D	I	PLR	overhead	latency (ms)	MTBE before FEC (sec)	MTBE after FEC (day)		
10	10	10	1,E-05	10%	525	274,00	70,47	} Column only 1D FEC	
10	10	10	1,E-06	10%	525	2740,00	7047,13		
5	5	5	1,E-05	20%	127	274,00	158,56		
5	5	5	1,E-06	20%	127	2740,00	15856,19		
L	D	I	PLR	overhead	latency (ms)	MTBE before FEC (sec)	MTBE after FEC (day)		
10			1,E-05	10%	30	274,00	70,47		} Row only 1D FEC
10			1,E-06	10%	30	2740,00	7047,13		
5			1,E-05	20%	16	274,00	158,56		
5			1,E-06	20%	16	2740,00	15856,19		

Figure tar for seg en $10 * 10$ matrise og viser verdier for ekstra båndbredde pga av FEC, forsinkelse, tidsavstand mellom synlige feil som en funksjon av forskjellige pakketaps forhold. Den forutsetter også en tilfeldig uniform feilfordeling.

Det er anbefalt å benytte kolonne interleaving (se fig. xxxxx) for å øke robustheten ved blokkfeil.

For koder modulen er forsinkelsen ikke mere enn størrelsen på en IP pakke. Tabellene nedenfor viser en algoritme for å finne riktig tidsluke for innsetting av FEC pakken når kolonne interleaving benyttes.

(Formler hentet fra et gruppeprosjekt i Formal Methods)

Timeslot calculation Column FEC

$$FEC-L \rightarrow (SEQ-1) - ((SEQ-1) \text{MOD } L) * (L+1) \text{MOD } (L * D) == 0$$

Tabell 3 Timeslot formula for FEC-L generation

Timeslot calculation Row FEC

$$FEC-D \rightarrow ((SEQ - 1) \text{MOD } L) == 0$$

Tabell 4 Timeslot formula for FEC-D generation

Dekodermodulen vil ha et kritisk punkt som er hvor mye forsinkelse den skal akseptere, d.v.s. hvor lenge den skal vente på en FEC pakke. Jo lengre den kan vente jo høyere er sannsynligheten for at man oppnår feilfri avspilling. Hvis man forutsetter en matrise uten interleaving vil man minimum måtte innføre en ekstra forsinkelse på $(L * D) + L - 1$.

Egenskaper

CoP#3:

Innenfor en matrise håndteres spredte tap i på opptil 24 RTP pakker dersom ingen FEC pakker går tapt og blokketap på opptil 21 RTP pakker dersom ingen FEC pakker går tapt.

Jitter egenskaper på 60ms

Fordeler

Man trenger ikke endre MTS. FEC vil bli sendt som en egen datastrøm

Bedre enn DVB IPI2001-016 når det gjelder feilbeskyttelse.

Protokollen er basert på bruk av åpne standarder med unntak av MPEG-2 (som holdes av ITU-T).

Lett å implementere

Få XOR operasjoner

Ulemper

Man er ikke i stand til å korrigere enkelte tapsmønstre, selv ved få tapte pakker.

Det er relative mye overhead for god beskyttelse

Har ingen prioritering av pakker som inneholder kritiske data.

Dårligere jitter egenskaper (60ms) enn DVB IPI2001-016 (40ms)

Ved lave bitrater er forsinkelsen relativt høy.

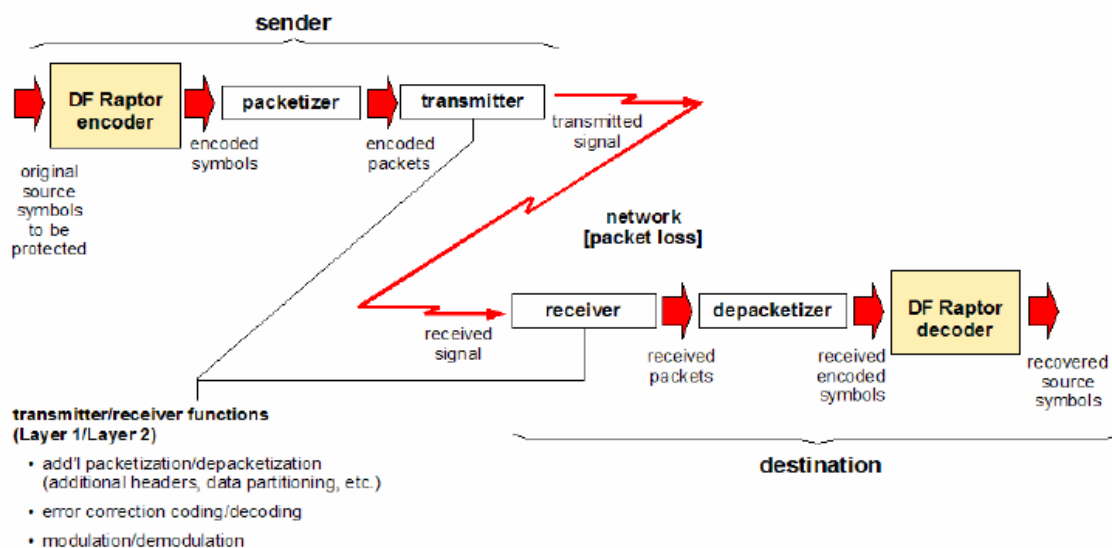
Den endimensjonale løsningen vil kun være i stand til å korrigere inn et pakketap pr blokk.

3.5.4 Raptor feilkorreksjon

Beskrivelse

DF Raptor FEC er utviklet av Digital Fountain ved AS. De er en metoder for feilbeskyttelse av blokker med symboler. DF Raptor FEC vil generere et sett ekstra symboler som benyttes ved regenerering av tapte pakker. Den originale datastrømme sendes nå sammen med de ekstra beskyttelses pakkene.

Bilder fra DF_RaptorOverview.pdf

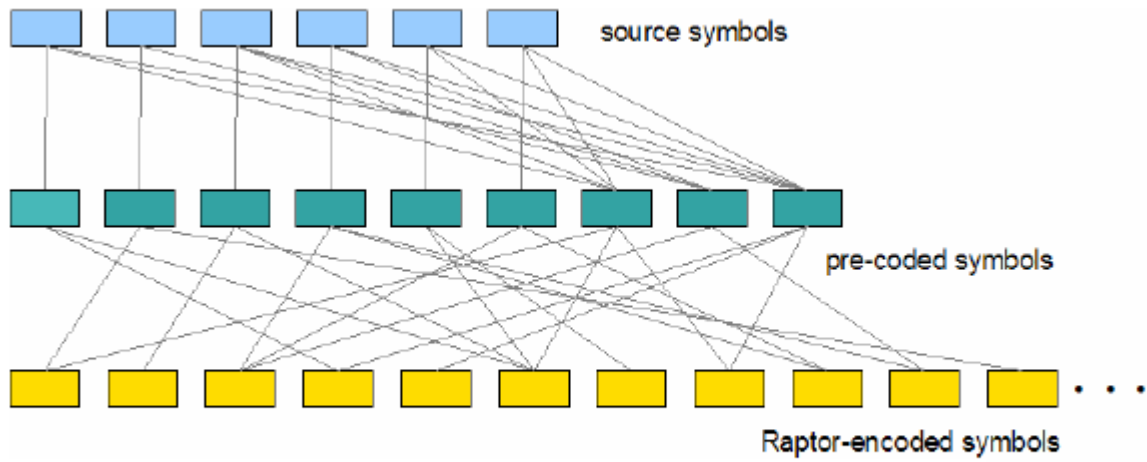


Digital Fountain's DF Raptor FEC technology reliably delivers data regardless of network conditions

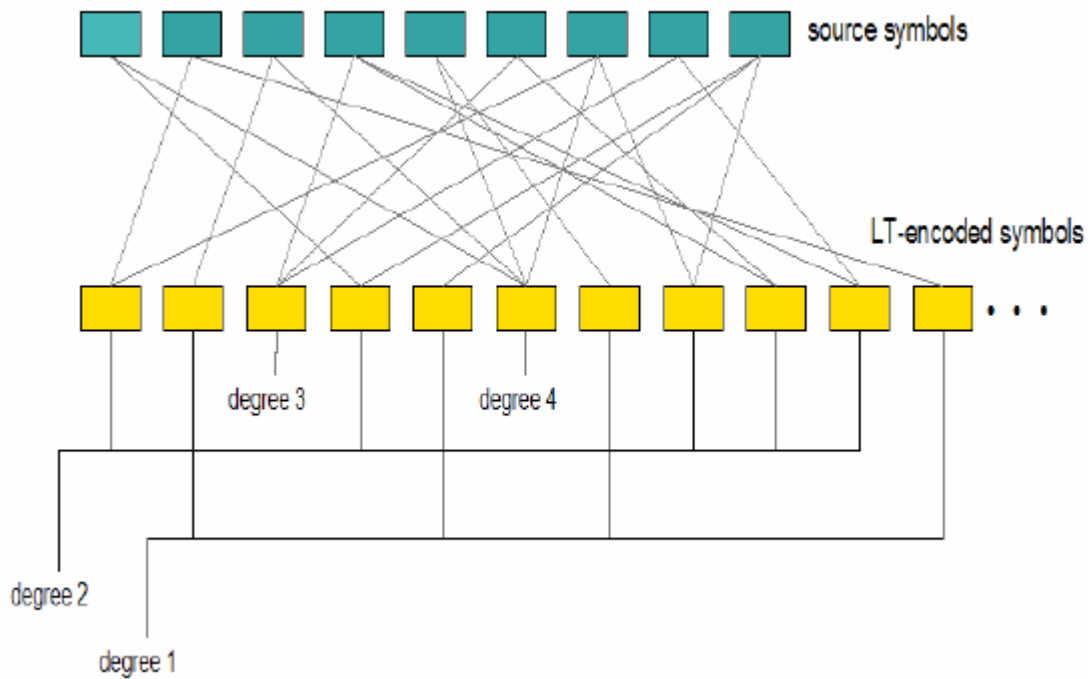
Raptor baserer seg på XOR av et varierende antall pakker. Man har flere nivåer av hvor mange pakker som skal inngå i XOR operasjonen.

For å generere et sett med symboler, velger man ut en blokkstørrelse med L symboler. Så allokere man plass M nye symboler hvor $M > L$. Antall ekstra symboler regnes ut fra en bestemt algoritme A. En algoritme B regner ut fra symbolene i L hva M skal inneholde

Et oversiktsdokument fra Digital Fountain viser følgende figurer:



DF Raptor encoding uses first a pre-code algorithm and then an LT fountain code to achieve optimum performance



Egenskaper

For kodersiden er det antatt ca 10 – 15 XOR operasjoner.

For dekoderside er det antatt ca 12 XOR operasjoner.

Referansemodellen er en 206MHz Intel StrongARM som skal håndtere en 27Mbits strøm.

Fordeler

Er dynamisk og kan teoretisk dekke alle typer blokkta

Ulemper

Krever flere regneoperasjoner enn andre metoder. Dette vil kreve mye av moduler som skal håndtere bitrater over 100Mbit/s

3.6 Innhenting av empiriske data

Vi vil her se på to muligheter for å innhente empiriske data. Den ene er å innhente ferdig registrert statistikk og den andre er å utføre reelle målinger.

3.6.1 Reelle målinger

IP protokollen gir grunnlag for flere alternativer når det gjelder å innhente empiriske data til denne oppgaven. Det er spesielt 2 opsjoner som er interessante.

Den ene er muligheten for å kunne tunnellere trafikk ved å legge til en IP header utenpå den originale headeren. Dette gir mulighet for å sende trafikk til en bestemt ruter vil tolke den ytre headeren og så sende trafikken videre. En annen variant er å sette opp en VPN forbindelse til fjernt sted og så sende trafikken via det stedet og tilbake.

Den andre muligheten er å kunne utføre kilderuting. Her finnes det to varianter, den ene er løs kilde ruting og den andre er fast kilde ruting. Begge trenger å utvide IP headeren med en rutertabell som sier hvilke rutere IP pakken må være innom før den når målet sitt. Også her er vi avhengige av at ruterne ikke er programmert til å blokkere slik trafikk. Begge metoder kan misbrukes til

Siste alternativ er å sette opp et system på to forskjellige steder for å gjøre målinger. Dette vil sannsynligvis bare gi tilgang til lavhastighets linjer.

3.6.2 Ferdigregistrert statistikk

Mye av artiklene som omhandler tapsstatistikk viser bare summeriske verdier som man ikke kan dra nytte av i denne sammenheng. Disse har da kun interesse som sammenligningsgrunnlag i sluttdelen.

4 Empiriske data

4.1 Metodevalg for innhenting av empiriske data

4.1.1 Valg av programvare

For å kunne gjøre en analyse av DVB IPI FEC og Pro-MPEG FEC er man avhengig av detaljert statistikk ved logging av UDP trafikk. Et oversiktsbilde på www.caida.com over applikasjoner til bruk i testformål gav en link til Iperf som er blitt videreutviklet til å generere ønsket statistikk. Andre applikasjoner som ble overfladisk evaluert så ut til å fokusere på summerisk statistikk enn på detaljer ved hver enkelt hendelse. Kildekoden til Iperf er tilgjengelig under BSD lisens og lot seg kompilere både under Linux og Windows. Denne pakken leveres nå som "ekstra" software til Fedora Core 4 og 5. Linux er interessant i dette tilfellet pga bedre egenskaper når det gjelder stabilitet og ytelse.

De viktigste parameterne vi valgte under målingen var måling av antall tapte pakker pr hendelse, sekvensnummer differanse mellom pakketap, relative forsinkelse, hastighet, måleperiode, statistikk over pakketap, dupliserte og ekvensfeil pakker

Selve måleperioden begynte med å prøve ut metoder for å lage "tur/retur" tunneler og med forsøk på kilderuting. For å gjøre dette begynte man med kilde- og eksempelkode til Winpcap som er noe av grunnlaget til Ethereal (et analyseverktøy for ethernet trafikk)

For å gjøre tester med IP tunnel og kilde ruting falt valget på Winpcap/Libpcap som er en del av grunnlaget for Ethereal. Her finnes det vedlagt en del kildekodeeksempler som kan videreutvikles til formålet. Dette i sammen med litt kildekode fra T-VIPS ga grunnlaget for å kunne generere egnede IP pakke sekvenser.

Programvarepakken Iperf fra NLANR/DAST har vist seg å være et godt utgangspunkt. Iperf-1.7.0 er en programvarepakke som er mulig å kjøre i Windows og Linux miljøer. Kildekoden (C/C++) er tilgjengelig lar seg kompilere med GCC under Linux og Cygwin. I tillegg finnes det en prosjektfil for Visual C++. Det finnes også et GUI brukergrensesnitt laget i Java (Jperf) som kan benyttes sammen med Iperf. En nyere utgave, Iperf-2.0.2, er en pakke som følger

med Fedora Core 4 og 5 i katalogen for ekstra materiale. Et søk på internett etter Iperf viser at programpakken benyttes en del i testmiljøet til IETF.

Umodifisert gir Iperf en statistikk over periodisk pakketap, bitrate og jitter. For å kunne bruke dette programmet videre trengte mer detaljer rundt pakketap. Iperf-1.0.7 ble modifisert til kunne gi statistikk om tid for hendelse, antall tapte pakker ved hver tapshendelse, datagram nummer før og etter for å kunne tolke avstand mellom feil, forsinkede pakker, dupliserte pakker. I tillegg periodisk logging av bitrate og jitter. Opprinnelig kunne Iperf kun TCP og UDP. For å kunn benytte TVG420 som en konstant kilde ble Iperf også modifisert til å kunne sende og motta RTP med MPEG-2 nullpakker.

4.1.2 Prøvetest med IP tunnel

Noen av de første problemene man møter på er sperrer forårsaket av brannmurer og lignende programvare for beskyttelse av PC og bruker. Noe manglende kunnskap gjorde at man etter kort tid gikk over til å utprøve dette under Linux. Selve testen viste at ruterne leste den indre IP headeren or sendte pakken umiddelbart i retur. Mislykket med andre ord.

4.1.3 Prøvetest med kilderuting

Under testing får man problem med rutere som avviser slik trafikk. Dette har sin bakgrunn i at kilderuting kan utgjøre et sikkerhetsmessig problem.

4.1.4 Prøvetest via HIA's VPN Server

For å gjøre forsøk via en VPN tunnel har HiA tilgjengelig Cisco VPN klient programvare og en operativ VPN server i Kristiansand. En Linux VPN klient vil logge inn i Grimstad.

En TVG420 fra T-VIPS vil sende en RTP strøm fra lokalnettet i Grimstad til VPN klienten.

Forsøket viste at VPN server satte en trafikk begrensning på 10Mbit. Iperf serveren fikk noen store pakketap til å begynne med. Siden det var en del annen aktivitet på PC-en samtidig, kan dette ha en sammenheng.

4.1.5 Prøvetest fra T-VIPS til HIA

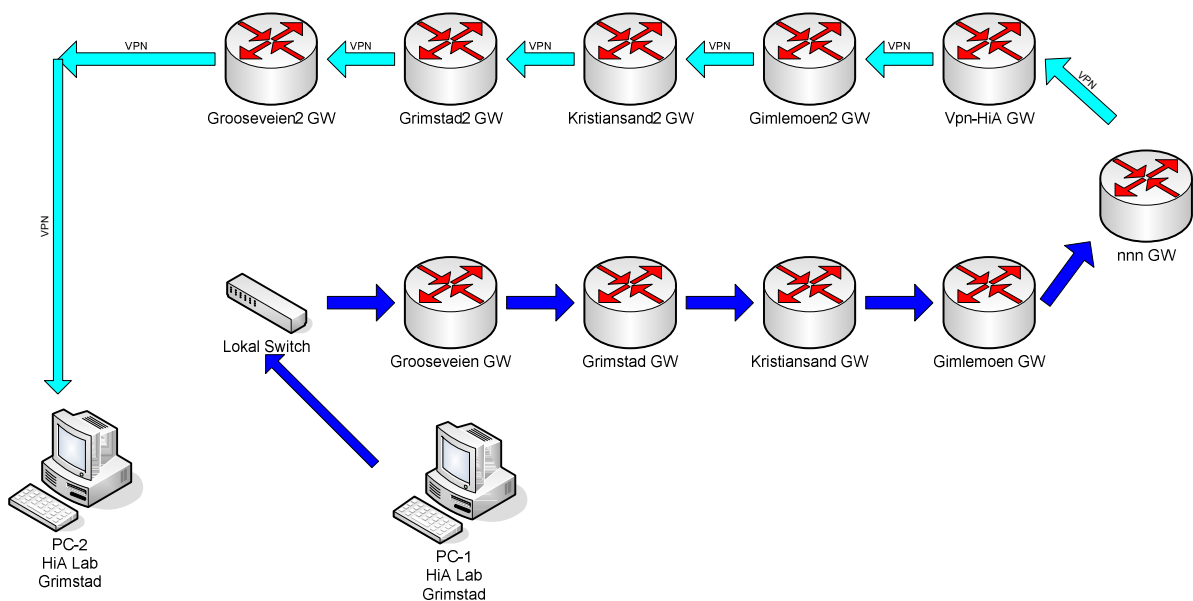
Ende til ende test gjøres med en PC som har en Iperf-klient på T-VIPS og en PC som Iperf server på HiA i Grimstad. Dette fungerer greit innenfor maksimal utgående båndbredde som er 2 Mbits/s. Det benyttes utgående linje fordi den antas å være minst aktiv.

4.2 Målinger i nettverket til HIA i Grimstad

4.2.1 Formål

Formålet med denne testen er å innhente tilnærmet realistisk statistikk over pakketap (enkelttap og blokketap) i IP nettverk når man overfører video med en relativt høy bitrate. For å få til dette må man sørge for datastrømmen går gjennom flere aktuelle og potensielle tapkilder.

4.2.2 Måleoppsett



Figur 7: Overføringstest ved HiA

Figuren viser et oppsett hvor man har en klient og server på lokalnettet i Grimstad. Serveren er innlogget på en VPN server i Kristiansand mens klienten er koblet til direkte lokalt. Meningen er å styre IP trafikken via en ruter i Kristiansand og tilbake til Grimstad for å øke antall hopp før IP strømmen når serveren som analysere den. Samtidig er det ønskelig med en god kvalitet på linjen. De første testmålingen viste at det periodisk ble mottatt pakker med sekvensfeil noe som tyder på at trafikken antagelig blir splittet underveis.

Kapasiteten fra laben er en 100Mbits linje. Mellom Grimstad og Kristiansand skal det være en 2Gbits linje. For å få en oversikt over ruternetverket ble programmet Traceroute kjørt på begge PC-ene. PC-2 er logget inn på Vpn-HiA-GW og ser PC-1 fra nnn-GW. Den listen som

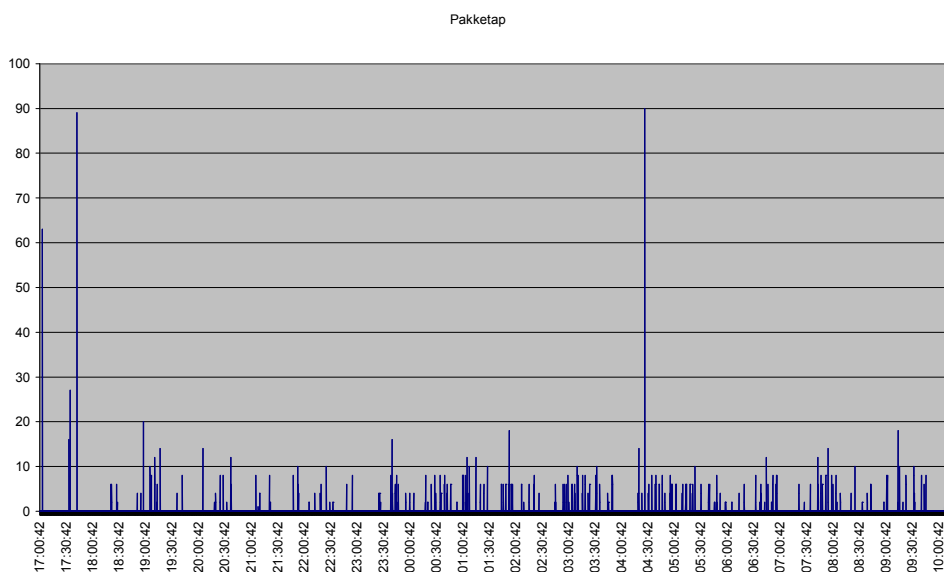
kom fram ved gjentatte forsøk viste 2 separate ruter stier. Det gjør at man sannsynligvis ville passere gjennom 2 sett av rutere. Det gir bedre testforhold med tanke på om RTP strømmen skulle påvirke seg selv.

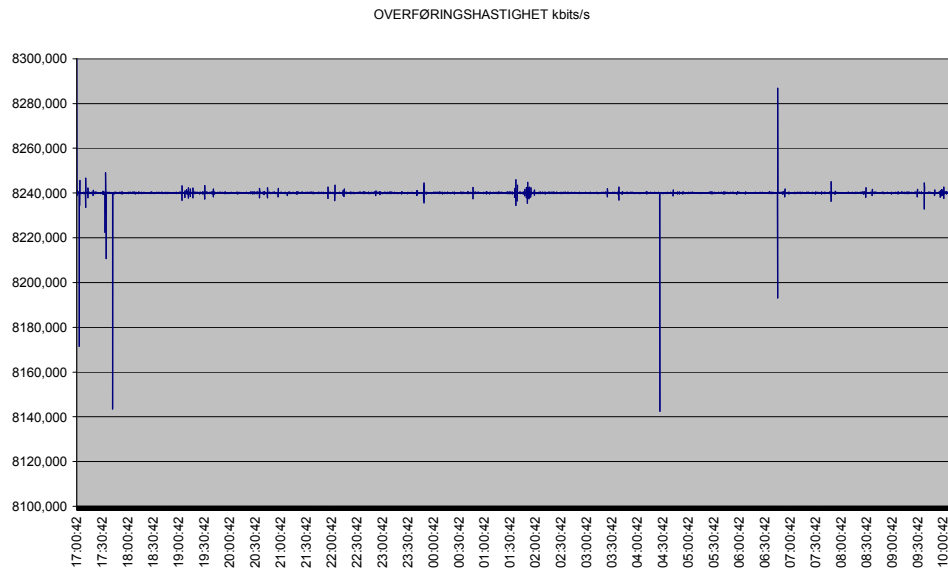
Følgende oppsett er benyttet i Test-1:

- Overføringshastigheten er satt til 8Mbit/s.
- En modifisert utgave av Iperf-1.7.0 benyttes for logging (kan nå håndtere RTP pakker)
- En TVG420 fra T-VIPS benyttes som sender av RTP/UDP/IP pakker.
- Klientprogrammet på T-VIPS sender fra en 3GHz Pentium 4 PC med Windows XP Professional. Her benyttes en versjon av Iperf som er kompilert i MS Visual Studio 2005.
- Serverprogrammet (logging) kjøres på en Linux PC, VIA ITX 1GHz, med Fedora Core4 installert. Denne er direkte tilkoblet internett med 100Mbits ethernet.

Resultatet for målingen er:

- 61950 sekunders testperiode (17,2 timer)
- 47068855 pakker er mottatt
- 1908 pakker er tapt (0,004 %)
- 0 pakker hadde sekvensfeil
- 0 dupliserte pakker
- 90 pakker som høyeste blokkatap





Følgende oppsett er benyttet Test-2:

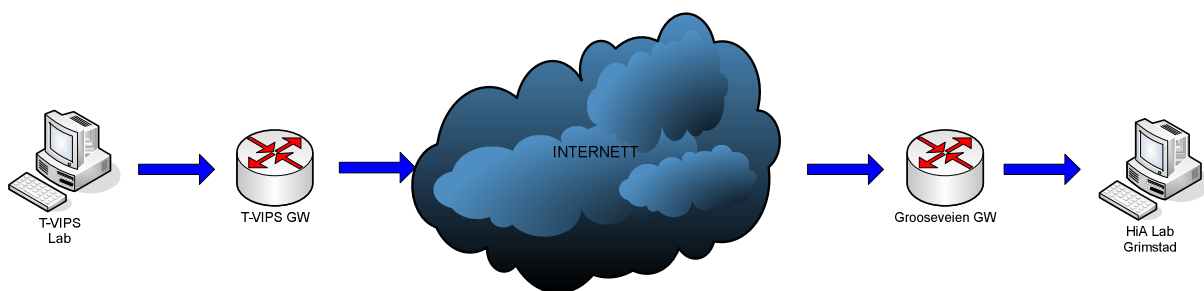
- Overføringshastigheten er satt til 1Mbit/s.
- En modifisert utgave av Iperf-1.7.0 benyttes for logging (server) og sending (klient). Iperf er modifisert til å gi mer detaljert periodisk logge informasjon. Bl.a. logges nå størrelse og antall på blokkta.
- Klientprogrammet på T-VIPS sender fra en 3GHz Pentium 4 PC med Windows XP Professional. Her benyttes en versjon av Iperf som er kompilert i MS Visual Studio 2005.
- Serverprogrammet (logging) kjøres på en Linux PC, VIA ITX 1GHz, med Fedora Core4 installert. Denne er direkte tilkoblet internett med 100Mbits ethernet.

4.3 Målinger i nettverk - T-VIPS til HIA

4.3.1 Formål

Formålet med denne testen er å innhente tilnærmet realistisk statistikk over pakketap (enkelttap og blokketap) i IP nettverk når man overfører video med en relativt høy overføringshastighet og via mange hopp. For å få til dette må man sørge for datastrømmen går gjennom flere aktuelle og potensielle tapskilder.

4.3.2 Måleoppsett



Figur 8: Test fra T-VIPS lab til HiA lab

T-VIPS har en standard 2Mbits SDSL linje tilkoblet internett via XXXXX. Den er ikke bestilt med noen spesielle QoS opsjoner, så man må definere den som en "best effort" linje. Dette gjør at man må forvente en relativt høy taps prosent. Det er foretatt 2 målinger (Test-1 og Test-2) over ca 3 dager fra T-VIPS til HiA. Begge er gjennomført fra fredag kveld til mandag/tirsdag.

Følgende oppsett er benyttet Test-1 (28.04 til 02.05):

- Overføringshastigheten er satt til 1Mbit/s.
- En modifisert utgave av Iperf-1.7.0 benyttes for logging (server) og sending (klient). Iperf er modifisert til å gi mer detaljert periodisk logge informasjon. Bl.a. logges nå størrelse og antall på blokketap.
- Klientprogrammet på T-VIPS sender fra en 3GHz Pentium 4 PC med Windows XP Professional. Her benyttes en versjon av Iperf som er kompilert i MS Visual Studio 2005.
- Serverprogrammet (logging) kjøres på en Linux PC, VIA ITX 1GHz, med Fedora Core4 installert. Denne er direkte tilkoblet internett med 100Mbits ethernet.

Resultatet for målingen er:

- 314500 sekunders testperiode (87 timer)
- 29601128 pakker er mottatt
- 23900 pakker er tapt (0,08 %)
- 135 pakker hadde sekvensfeil
- 0 dupliserte pakker

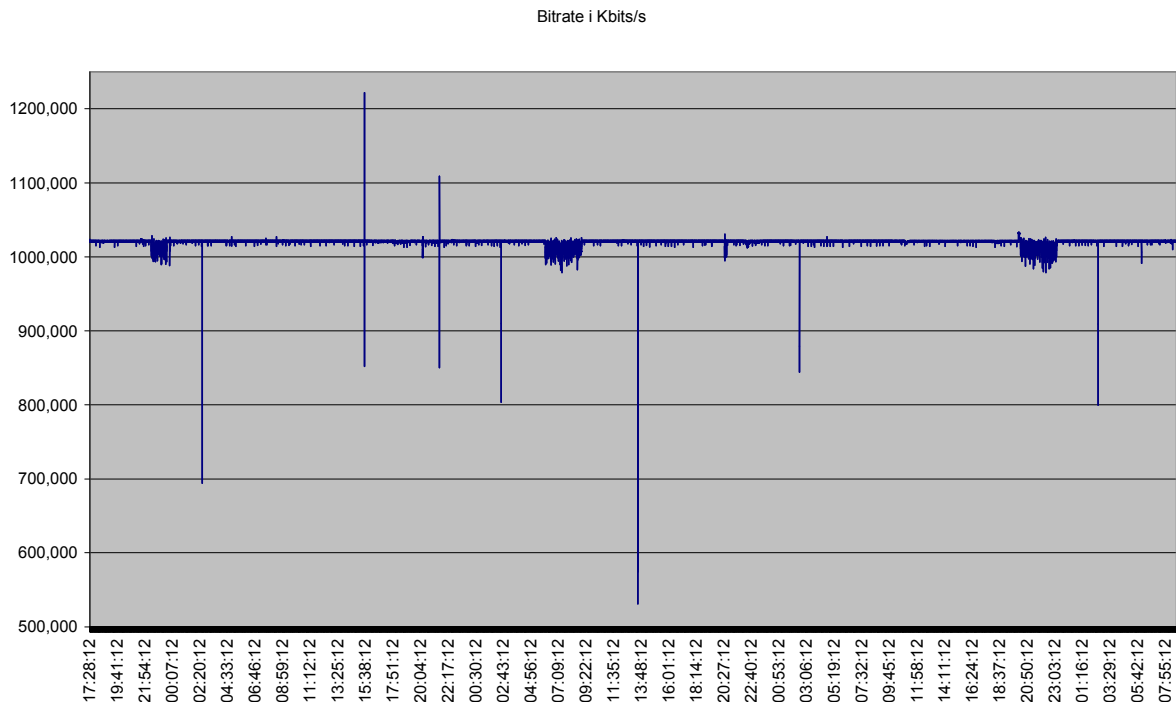
4.3.3 Målinger 28.04 – 02.05

Diagrammet viser pakketap over e periode på 3 dager, fra fredag til mandag. I flere tilfeller er det blokketap på mer enn 100 pakker. Det største enkelt tapet var på mellom 500 og 1000 (logg hadde ikke detaljer) pakker.

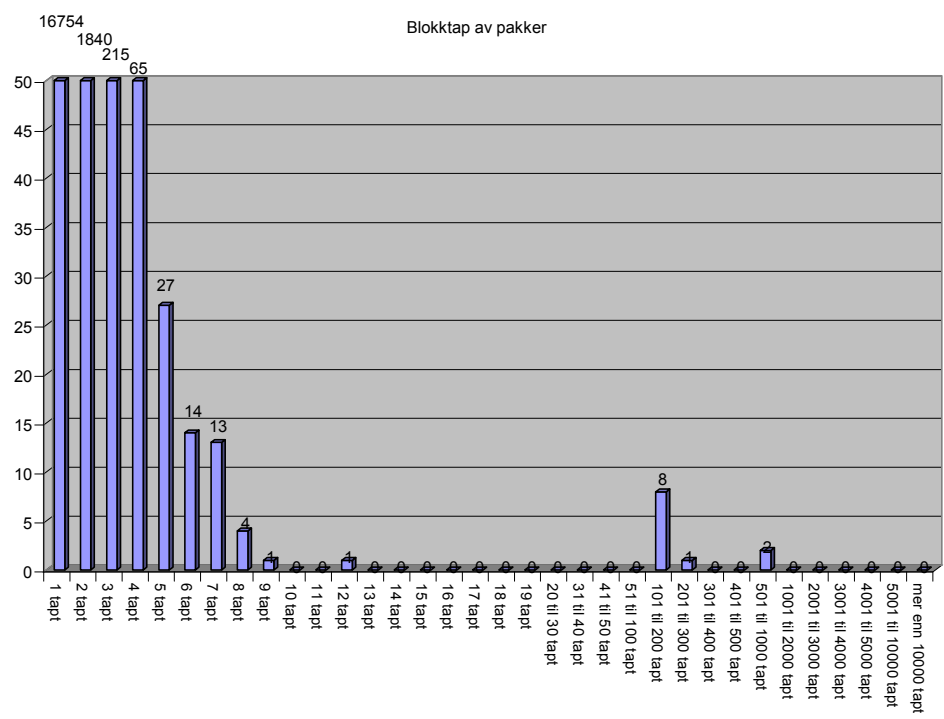
Resultatet for målingen er:

- 314500 sekunders testperiode (87 timer)
- 29601128 pakker er mottatt
- 23900 pakker er tapt (0,08 %)
- 135 pakker hadde sekvensfeil
- 0 dupliserte pakker

Grafen viser hvordan overføringshastigheten varierer. De punktene med store utslag er områder hvor det skjer større pakketap.



Bar grafen viser antall hendelser med hvert blokkta. Denne og den neste bar grafen viser samme tendens. Det er en uke mellom disse to testene



4.3.4 Målinger 05.05 – 08.05

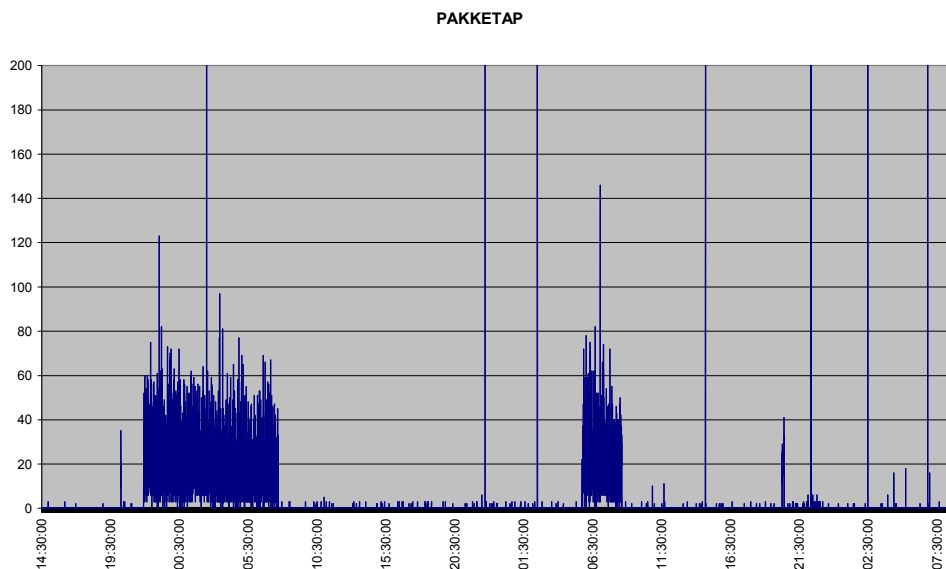
Diagrammet viser pakketap over e periode på 3 dager, fra fredag til mandag. I flere tilfeller er det blokketap på mer enn 100 pakker. Det største enkelt tapet var på ca 3554 pakker. Det vil tilsvare et avbrudd på ca 38 sekunder ved den hastigheten vi benyttet.

Følgende oppsett er benyttet T-VIPS-2 (05.05 til 08.05):

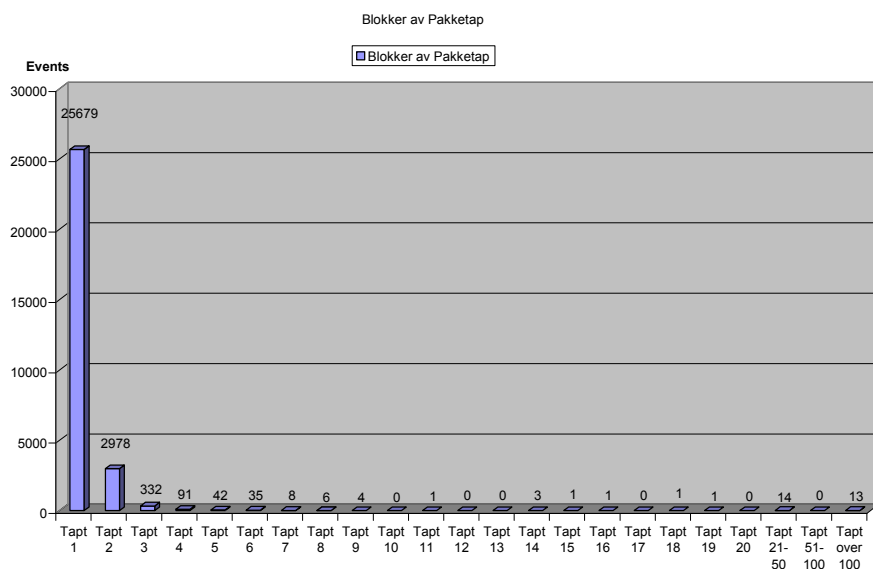
- Overføringshastigheten er satt til 1Mbit/s.
- Endring fra forrige test er en nyere utgave av Iperf-1.7.0. Det logges nå mer detaljer for hver enkelt feil som opptrer og differanser til forrige feil, i motsetning til tidligere hvor det ble logget antall blokketap.

Resultatet for målingen er:

- 237820 sekunders testperiode (66 timer)
- 23177343 pakker er mottatt
- 39472 pakker er tapt (0,17 %)
- 96 pakker hadde sekvensfeil
- 0 dupliserte pakker



Et oversikts bilde over antall hendelser med hvert blokktaap.



Grafen viser at det i de fleste tilfellene er tap på 1 pakke. Mange av disse kommer tett etter hverandre slik at de ville falle innenfor feilkorreksjonsramme.

4.3.5 Kommentarer

De store tapene kan være et resultat av service på linjen. Det har vært registrert flere slike tilfeller på T-VIPS. Linjen er en bedriftslinje med 2Mbit/s hver vei. Denne type linje vil sannsynligvis være uaktuell sanntidsoverføring av video i en profesjonell sammenheng. En grov oversikt viser at man med en 4x4 matrise i Pro-MPEG FEC ville feile minimum 88 ganger.

5 Diskusjon

5.1 *Empiriske data og relevans av målingene*

5.1.1 Standard internett tilkobling

Målingene mellom T-VIPS og HiA er foretatt med en standard internettlinje ut fra T-VIPS. Overføringshastigheten er lav i forhold til hva som trengs når det skal overføres høykvalitets video. Profesjonelle aktører vil også høyest sannsynlig bestille overføringslinjer med kvalitetsgaranti (QoS). Likevel er det enkelte aktører, som iflg. T-VIPS, vurderer denne typen internettlinjer som et alternativ. Det er derfor interessant å kunne sammenlikne aktuelle FEC algoritmer mot de hendelser som skjer i åpne nettverk. Målingene vil kunne gi en indikasjon på hvilke egenskaper som er nødvendig av FEC algoritmer.

Selv ved denne lave hastigheten viser målingene periodisk høye tap. Spesielt kritisk er de høye blokktapene som faller utenfor aktuelle FEC algoritmer.

5.1.2 Høyhastighets internett tilkobling

De testene vi fikk gjort på HiA ble gjort med 100Mbits linjer. Dette ga mulighet for mer reelle tester med tanke på overføringshastigheter som benyttes for profesjonell video. Feilhyppigheten sank betraktelig selv om det var enkelte tap som ikke ville ha vært mulig å korrigere. En annen type feil, sekvensfeil, ble mer tydelig her. Tidligere var det kun en sjelden pakke som hadde sekvensfeil. Med overføringshastighet på 20Mbits/s fikk vi flere sammenhengende pakker som hadde sekvensfeil. Dette har sannsynligvis noe med måten ruterne er koblet. Vurderingen om at det er satt opp et rutersystem hver vei mellom Kristiansand og Grimstad synes ikke å stemme helt.

5.2 Empiriske data og FEC påvirkning

5.2.1 DVB IPI (Reed-Solomon)

Når man sammenligner måleresultatene og hva DVB IPI FEC er i stand til å korrigere ser man at det vil oppstå mye feil som vil være synlig i video. Den beste metoden vil kunne håndtere blokkta på maks 48 MTS pakker og med 612 MTS pakkens avstand noe som vil tilsa maks 6 IP pakket i blokkta med en avstand på minst 88 IP pakker. Ved sporadiske feil er nedre grense 16 MTS pakker d.v.s. 2 IP pakker. Mode =0 tas ikke med i statistikken da den krever et lavere antall MTS pakker pr IP pakke enn de andre. Verdiene er basert på bar grafene ovenfor.

Minimum antall feil

	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	av totalt
HiA 1	X	>463	>205	>5	>5	47068855
Test 2	X					
T-VIPS 1	X	>2191	>351	>71	>30	29601128
T-VIPS 2	X	>3531	>553	>130	>53	23177343

I tillegg vil de antall ganger hvor summe av flere hendelser kommer innenfor en ramme.

5.2.2 Pro-MPEG FEC

Mange av målingene har vist pakketap som langt overstiger hva som er mulig å regenerere. Noen av tapene vil neppe oppstå i nettverk som allokerer for sanntidsoverføring av video. Vanlige internettlinjer baserer seg på retransmisjon som en hovedfunksjon for dataoverføring. Verdiene er basert på bar grafene ovenfor

Minimum antall feil

	4 x 4	10 x 10				av totalt
HiA 1	>5	>5				47068855
Test 2						
T-VIPS 1	>71	>12				29601128
T-VIPS 2	>88	>35				23177343

I tillegg vil de antall ganger hvor summe av flere hendelser kommer innenfor en ramme.

5.2.3 DF Raptor FEC

6 Konklusjon

Et av formålene med denne oppgaven var å komme fram til et valg når det gjaldt de tre foreslåtte metodene for feilkorreksjon. Noe av grunnlaget har vi i de målingene som er foretatt i to forskjellige nettverk. Nettverket mellom T-VIPS og laben på HiA i Grimstad viste store tap ved enkelte tidspunkt. Også mellom Grimstad og Kristiansand viste det seg enkelte ganger til dels høye pakketap. Internettlinjene som ble testet var ikke konfigurert for typisk sanntidsoverføring av høykvalitetsvideo, så dette må også tas med i betraktning. En profesjonell linje antas, iflg. T-VIPS, å ha en BER mellom $10E-6$ og $10E-9$. Valget kunne ha vært Raptor FEC med de egenskapene den metoden har, men et av ankepunktene her er hva slag prosessorkraft som trengs for å håndtere 500Mbit/s og 1Gbit/s. Det har ikke vært nevnt tester i dette området (T-VIPS' eget utstyr er beregnet å kunne dekke noe av dette området.) Det oppgis også at det er bundet en OEM lisens til metoden. Så lenge det er mest sannsynlig at videokontribusjon vil foregå på linjer med QoS garanti, vil Pro-MPEG FEC være et det anbefalte valg. Skal man over på ordinære internettlinjer bør Raptor FEC vurderes sammen med en aksept for større forsinkelse.

Appendix A Referanser

[10] Frode Sørensen, "Innføring i Nettverk", IDG Norge, 2004, p412

[10] Frode Sørensen, "Moderne IP-nett", IDG Norge, 2004, p412

[22] George C. Clark, Jr. and J. Bibb Cain, "Error-Correction Coding for Digital Communication", Plenum Press, 1981

[23] Todd K. Moon, "Error Correction Coding, Mathematical Methods and Algorithm", Wiley, 2005

[100] IEEE 802.3 2000, Ethernet Protocol ratified in 2002

[101] Alberto Leon-Garcia, Indra Widjaja, "Communication Networks", McGraw-Hill, 2nd edition, p166-190

[] EBU Abbreviations, http://www.ebu.ch/en/technical/trev/trev_abbreviations.pdf

[] N. Becerra Yoma, C. Busso and I. Soto, "Packet-loss modelling in IP networks with state-duration constraints", IEE Proc.-Commun., Vol. 152, No. 1, February 2005

Appendix B Linker

[] Pro-MPEG Forum,
<http://www.pro-mpeg.org/>

[] Pro-MPEG FEC,
<http://www.pro-mpeg.org/publications/pdf/Vid-on-IP-CoP3-r2.pdf>

[] DVB IPI,
http://www.dvb.org/technology/standards_specifications/internet_protocol/dvbipi/

[] DVB IPI Group,
http://www.dvb.org/groups_modules/technical_module/tmipi/index.xml?groupID=14

[] DF Raptor,
<http://www.digitalfountain.com/index.cfm>

[] DF Raptor FEC,
<http://www.digitalfountain.com/solutions/mobileMultimedia.cfm?uid=57>

[] DF Raptor FEC,

[] NIST NET,
<http://snad.ncsl.nist.gov/nistnet/>

[] The Distributed Applications Support Team (DAST),
<http://dast.nlanr.net/>

[] The National Laboratory for Applied Network Research (NLANR),
<http://www.nlanr.net>

[] Cooperative Association for Internet Data Analysis,
<http://www.caida.org/home/>

[]
<http://www.fh-friedberg.de/fachbereiche/e2/telekom-labor/zinke/mk/mpeg2beg/beginnzi.htm>

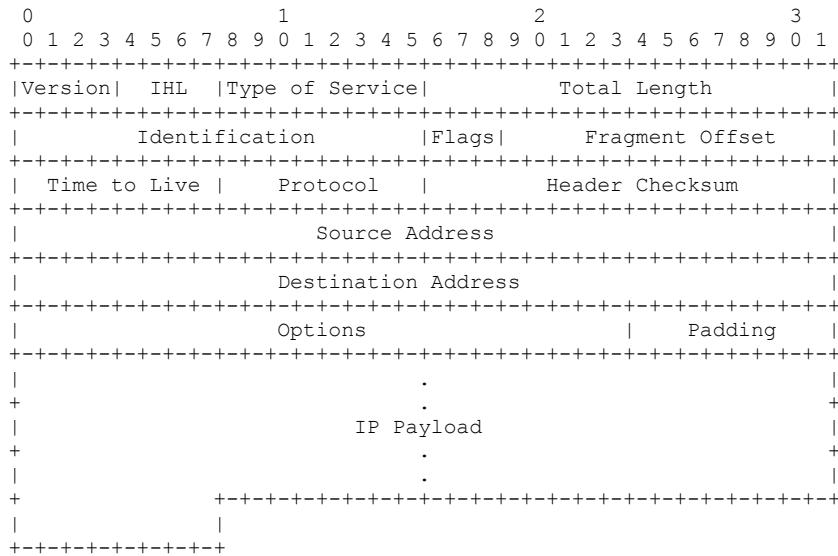
[] BBC
http://www.bbc.co.uk/rd/pubs/papers/paper_14/paper_14.shtml

[]
<http://ipnetwork.bgtmo.ip.att.net/pws/method.html>

Appendix C Protokoll strukturer

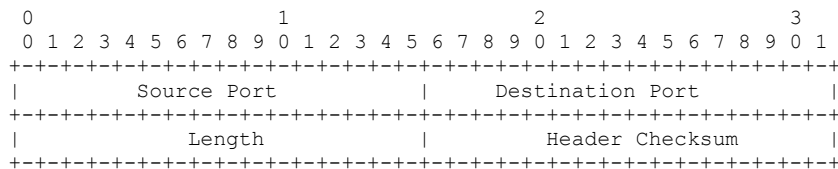
C.1.1 IP struktur

RFC791 beskriver følgende IP struktur:



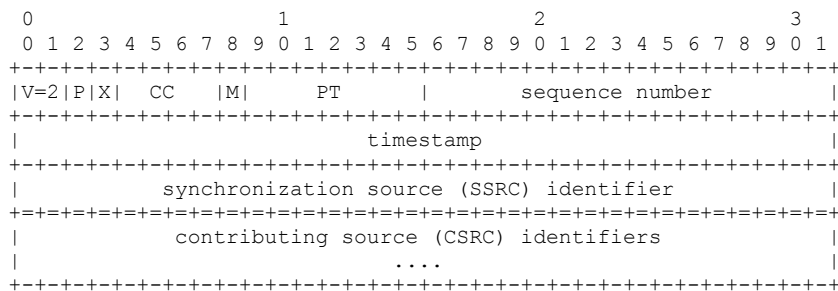
C.1.2 UDP

RFC768 beskriver strukturen på UDP på følgende måde:



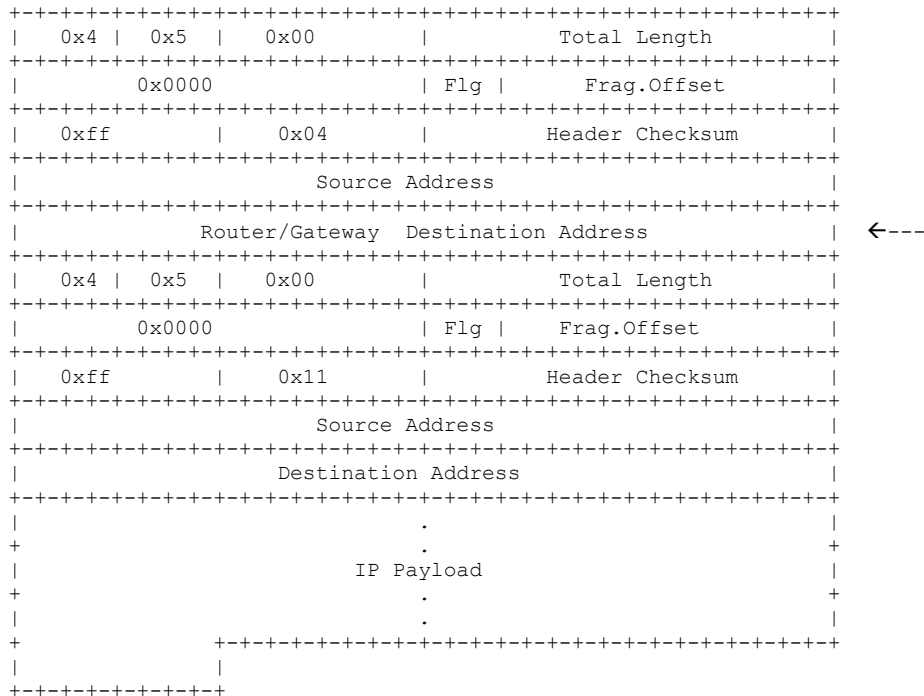
C.1.3 RTP

RFC1889 beskriver RTP strukturen på følgende måde:



C.1.4 IP i IP tunnel

RFC791 og RFC2003 beskriver følgende IP i IP pakke struktur:



C.1.5 Struktur for kilderuting (LSRR og SSRR)

RFC791 definerer to varianter af kilde ruting (løs og fast kilde ruting):

Options:

```

Loose Source and Record Route (LSRR)
+-----+-----+-----+-----+-----+-----+-----+-----+
|10000011| length | pointer| route data |
+-----+-----+-----+-----+-----+-----+-----+-----+
Type=131

```

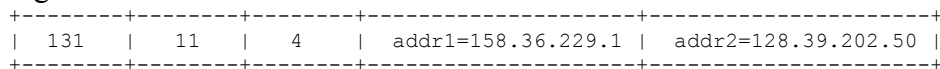
```

Strict Source and Record Route (SSRR)
+-----+-----+-----+-----+-----+-----+-----+-----+
|10001001| length | pointer| route data |
+-----+-----+-----+-----+-----+-----+-----+-----+
Type=137

```

En IP pakke som skal ha løs kilderuting

- fra 192.168.0.100
- til 128.39.202.50
- via 128.39.0 1
- og 158.36.229.1



Fra kilde node 1 (Dest.Addr. er lik første via punkt):

```
Ver=4  IHL=8      TOS      Total Length
      Id          Flags     Fragment Offset
      TTL        Protocol   Header checksum
      Source     Address = 192.168.0.100
      Destination Address = 128.39.0.1

Type = 131      Length = 11      Pointer = 4
                Address 1 = 158.36.229.1
                Address 2 = 128.39.202.50

Type = 0
```

IP Payload Data

Fra node 1 til node 2 (Dest.Addr. og R.Addr.1 er kryssbyttet):

```
Ver=4  IHL=8      TOS      Total Length
      Id          Flags     Fragment Offset
      TTL        Protocol   Header checksum
      Source     Address = 192.168.0.100
      Destination Address = 158.36.229.1

Type = 131      Length = 11      Pointer = 8
                Address 1 = 128.39.0.1
                Address 2 = 128.39.202.50

Type = 0
```

IP Payload Data

Fra node 2 til endepunkt Dest.Addr. og R.Addr.2 er kryssbyttet):

```
Ver=4  IHL=8      TOS      Total Length
      Id          Flags     Fragment Offset
      TTL        Protocol   Header checksum
      Source     Address = 192.168.0.100
      Destination Address = 128.39.202.50

Type = 131      Length = 11      Pointer = 12
                Address 1 = 128.39.0.1
                Address 2 = 158.36.229.1

Type = 0
```

IP Payload Data

Appendix D Iperf setup

Programvare med kildekode er vedlagt på CD

Client

Parameter opsett for Iperf Client:

```
./iperf -c 192.168.0.201 -u -P 1 -i 10 -p 55028 -w 10M -l 1328 -C -f k -b 200000k -t 240 -T 64
```

Dette vil starte utgående IP strøm på 20Mbits/s til server på adresse 192.168.0.201 port 55028

Loggeintervallet vil være på 10 sekunder.

Server

Parameter opsett for Iperf Server:

```
./iperf -s -u -P 1 -i 10 -p 55028 -l 1328 -f k
```

Dette vil starte en server som lytter på port 55028

Kildekode

Iperf-1.7.0 er den originale kildekoden

Iperf-1.7.7 er en modifiser versjon som skrive ut detaljert logg data

Iperf-1.7.9 er en forbedring av Iperf-1.7.7 med egen tråd socket receive

For kompilering på Linux:

Login som SuperUser (su) i et terminal vindu.

Make config

Make

Appendix E Loggfiler

Se vedlagt CD under katalogen ”Logg data”

Her er loggfiler fra HiA og T-VIPS

Under utskrift er de to header linjer, en for periodisk utskrift og en for feilhendelser.

Feilhendelser:

sck	log-id	time-s	time-us	event	drgm-id	prev-id	err-id	pcks	delay
-----	--------	--------	---------	-------	---------	---------	--------	------	-------

sck	socket id								
log-id	periodisk logg(1)		feilhendelse(2)						
time-s	PC tid i antall sekunder fra 01.01.1970								
time-us	mikrosekund teller								
event	hendelse teller								
drgm-id	nåværende datagram id								
prev-id	sist motatte datagram id								
err-id	feiltype pakke								
pcks	antall pakke i hendelsen								
delay									

Periodisk utskrift

sck	log-id	time	start	stop	udp-sleep	sleep-min	sleep-max	delay	d-min	d-max	jitter	j-min	j-max	rate-kb	rate-ms	r-min	r-max	byte-in	b-lost	pkt-in	p-dup	p-OoO	p-loss	p-lost	L01	L02	L03	L04	L05	L06	L07	L08	L09	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	B00	B01	B02	B03	B04	B05	B06	B07	B08	B09	B10	B11	B12	B13	B14
-----	--------	------	-------	------	-----------	-----------	-----------	-------	-------	-------	--------	-------	-------	---------	---------	-------	-------	---------	--------	--------	-------	-------	--------	--------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

sck	socket id																												
log-id	periodisk logg(1)		feilhendelse(2)																										
time	PC tid i antall sekunder fra 01.01.1970																												
start	sekund teller																												
stop	sekund teller																												
udp-sleep	ledig ventetid																												
sleep-min	ledig ventetid minimum																												
sleep-max	ledig ventetid maximum																												
delay	relativ pakkeforsinkelse mellom sender og mottager																												
d-min																													
d-max																													
jitter	jitter måling iht. RFC																												
j-min																													
j-max																													
rate-kb	overføringshastighet																												
rate-ms	tid i ms mellom hver innkommende pakke																												
r-min																													
r-max																													

byte-in	payload byte teller
b-lost	bytes tapt (ikke aktiv)
pkt-in	antall mottatte IP pakker
p-dup	antall dupliserte pakker
p-OoO	antall pakker med sekvensfeil
p-loss	delta pakketap
p-lost	antall tapte pakker
L01	Blokktap med 1 pakke
L02	Blokktap med 2 pakker
L03 L04 L05 L06 L07 L08 L09 L10	
L11 L12 L13 L14 L15 L16 L17 L18 L19	

B00	20 til 30 tapt
B01	31 til 40 tapt
B02	41 til 50 tapt
B03	51 til 100 tapt
B04	101 til 200 tapt
B05	201 til 300 tapt
B06	301 til 400 tapt
B07	401 til 500 tapt
B08	501 til 1000 tapt
B09	1001 til 2000 tapt
B10	2001 til 3000 tapt
B11	3001 til 4000 tapt
B12	4001 til 5000 tapt
B13	5001 til 10000 tapt
B14	mer enn 10000 tapt

