

**Universitetet i Oslo
Institutt for informatikk**

**Måling av
talekvalitet over IP**

Thomas Tomter

Hovedfagsoppgave

15.mai 2003



FORORD	4
SAMMENDRAG	5
KAPITTEL 1 "INNLEDNING"	6
1.1 ET UTDRAG AV TELEFONIENS HISTORIKK	6
1.2 LUFTFARTSKOMMUNIKASJON I ET HISTORISK PERSPEKTIV	7
1.3 LUFTFARTSKONTROLL	9
1.4 ÅRSAKEN TIL AT IP-TELEFONI ER INTERESSANT FOR REELL ATC-BRUK	11
1.4.1 <i>Leid linje</i>	11
1.4.2 <i>Private linjer</i>	13
1.4.3 <i>Desentralisert struktur</i>	14
1.5 MOTIVASJON FOR PROSJEKTET	14
1.5.1 <i>Problemstillinger</i>	16
1.6 TESTMETODE	16
1.7 RAPPORTENS STRUKTUR.....	18
KAPITTEL 2 "IP-TELEFONI ARKITEKTURER"	20
2.1 HVA ER IP-TELEFONI?	20
2.2 ARKITEKTURER.....	22
2.2.1 <i>PC-til-PC</i>	22
2.2.2 <i>PC-til-telefon</i>	22
2.2.3 <i>Telefon-til-telefon</i>	23
2.3 STILLHETSUNDERTRYKKELSE OG "COMFORT NOISE"	24
2.4 PROSESSERING	25
KAPITTEL 3 "TALEKVALITET"	28
3.1 DEFINISJONER AV TALEKVALITET	28
3.2 TALEKVALITET ER IGJEN ET TEMA.....	29
3.3 FAKTORER SOM PÅVIRKER TALEKVALITETEN	29
3.3.1 <i>Pakketap</i>	29
3.3.2 <i>Pakkeforsinkelser</i>	31
3.3.3 <i>Jitter</i>	36
3.3.4 <i>Ekko</i>	37
3.4 KLARHET I TALEN	39
KAPITTEL 4 "TALEKODING"	41
4.1 DIGITALISERING AV ANALOG TALE	41
4.1.1 <i>Filtrering</i>	41
4.1.2 <i>Sampling</i>	42
4.1.3 <i>Kvantisering</i>	43
4.2 CODEC-ER	45
4.2.1 <i>Bitrate</i>	46
4.2.2 <i>Forsinkelse</i>	46
4.2.3 <i>Kompleksitet</i>	47
4.2.4 <i>Kvalitet</i>	47
4.2.5 <i>IP-telefoni codec-enes attributter</i>	48
4.3 FORSINKELSESBUDEJETT	50
KAPITTEL 5 "BAKGRUNN FOR FORSØKSOPPSETT"	53
5.1 ATC TALETEKNIKK OG -KVALITETSBEDØMMELSE	53
5.2 SIKKERHETSASPEKTER I ATC OG FAKTORER SOM PÅVIRKER TALEKVALITETEN	55
5.3 VIKTIGE FAKTORER FORSØKSOPPSETT BØR TA HENSYN TIL.....	56
5.3.1 <i>Type tale</i>	57
5.3.1.1 <i>Valgt type tale</i>	57
5.3.2 <i>Talemateriale</i>	58
5.3.3 <i>Språk</i>	58
5.3.3.1 <i>Prosjektets språkvalg</i>	59
5.3.4 <i>Aktuelle typer tester</i>	59

5.3.5 Oppsummering av diskusjonen	60
5.4 VALG AV EKSPERIMENTOPPSETT	61
5.4.1 Opptaks- og avspillingsrekkefølge	64
5.4.2 Valg av skalaer.....	66
5.5 MÅLET MED EKSPERIMENTENE.....	67
KAPITTEL 6 "IMPLEMENTASJON OG TESTOPPSETT"	69
6.1 SERVERENS OPPBYGNING OG FUNKSJONALITETER	69
6.1.1 Tråd 1	71
6.1.2 Tråd 2	72
6.1.3 Uttesting av serveren	73
6.1.3.1 Klient	74
6.2 MULIGGJØRE FORSENDELSE AV TALE IGJENNOM DGRAMSERVER	75
6.2.1 Divert sockets.....	75
6.2.2 Endringer i og bruk av dgramServer.....	76
6.3 TESTOPPSETT	77
6.3.1 Oppkobling.....	79
6.4 OPPTAK AV TALEN	82
6.4.1 Valg av parametre.....	82
6.4.2 Forholdene under opptakene.....	84
6.4.3 Prosessering av opptakene.....	85
6.4.4 Organisering av opptakene.....	85
KAPITTEL 7 "TESTRESULTATER"	88
7.1 PAKKETAP.....	89
7.1.1 Talekvalitet.....	89
7.1.2 Lytteinnsats	90
7.1.3 Resultatenes svar på om kvaliteten er tilstrekkelig for ATC.....	92
7.2 PAKKEFORSINKELSE	93
7.2.1 Talekvalitet.....	93
7.2.2 Lytteinnsats	94
7.2.3 Resultatenes svar på om kvaliteten er tilstrekkelig for ATC.....	95
KAPITTEL 8 "DISKUSJON".....	97
8.1 KONVERSASJONSDYNAMIKK	97
8.2 OVERRASKELSER I TESTRESULTATENE.....	98
8.3 SVAKHETER	102
8.4 VIDERE ARBEID.....	103
KAPITTEL 9 "KONKLUSJON"	105
APPENDIKS A "SIGNALERINGSPROTOKOLLER"	106
A.1 H.323	109
A.1.1 Arkitektur	109
A.1.2 Protokoller i H.323-paraplyen.....	111
A.1.2.1 H.225.0 RAS.....	112
A.1.2.2 H.225.0 Call Signaling	113
A.1.2.3 H.245 Control Signaling.....	113
A.1.2.4 RTP	114
A.1.2.5 H.450.X Tilleggstjenester.....	115
A.2 SIP	117
A.2.1 Overblikk.....	117
A.2.2 Tjenere	118
A.2.3 Meldinger.....	120
A.2.4 SDP	122
A.3 SAMMENHOLDING AV SIP OG H.323	123
A.3.1 Funksjonalitet	123
A.3.2 Skalerbarhet.....	124
A.3.3 Enkelhet	125
A.3.4 Utvidelsesmuligheter	125
A.3.5 Modularitet	126

A.3.6 Oppsummering.....	128
APPENDIKS B "MÅLING AV TALEKVALITET"	130
B.1 ITU-T RECOMMENDATION P.800	130
B.1.1 "Conversational-opinion"-tester.....	131
B.1.1.1 Konversasjonstester i laboratorium	131
B.1.1.2 Skalaer.....	132
B.1.1.3 Eksempel på testprosedyre og konversasjonsoppgaver	133
B.1.1.4 Datainnsamling	134
B.1.1.5 Behandling av testresultater.....	134
B.1.2 "Listening-opinion"-tester.....	134
B.1.2.1 "Absolute Category Rating" (ACR).....	135
B.1.2.1.1 Eksperimentdesign.....	136
B.1.2.1.2 Skalaer	136
B.1.2.1.3 Analyse av testresultater.....	136
B.1.2.2 "Degradation Category Rating" (DCR).....	136
B.1.2.3 "Comparison Category Rating" (CCR).....	137
B.1.3 <i>Kritikk mot P.800</i>	137
B.2. OBJEKTIVE TESTER.....	139
APPENDIKS C "FORKORTELSER".....	141
APPENDIKS D "KILDEKODE"	143
D.1 DGRAMSERVER	143
D.2 DGRAMCLIENT	150
APPENDIKS E "PERSONLIG E-POST"	153
E.1 FRA NORSK LUFTFARTSMUSEUM.....	153
E.2 FRA FLYGELEDER	155
APPENDIKS F "TEKNISK SPESIFIKASJON"	156
F.1 "INTERNET PHONECARD"	156
F.2 "INTERNET PHONEJACK"	157
F.3 FORGRENER	158
APPENDIKS G "MANUS OG EVALUERINGSSKJEMA"	159
G.1 MANUS.....	159
G.2 EVALUERINGSSKJEMA	161
G.2.1 Norsk.....	161
G.2.2 Engelsk.....	163
APPENDIKS H "KONFIGURASJON"	165
H.1 GATEKEEPER.....	165
H.2 IP-TABLES.....	166
H.3 OPENPHONE.....	166
APPENDIKS I "EXCEL-ARK"	167
REFERANSER	170

Forord

Denne rapporten markerer avslutningen av hovedfagsstudiet mitt ved Institutt for Informatikk (IFI), Universitetet i Oslo, og har blitt til i samarbeid med Sintef Oslo, Elektronikk og Kybernetikk (EKY). Dette prosjektet var opprinnelig et studentprosjekt i et større prosjekt som Sintef i samarbeid med et eksternt firma søkte støtte til i Forskningsrådet (NFR). Dette prosjektet ble, imidlertid, ikke tildelt støtte fra NFR, men studentprosjektet levde likevel videre i form av min hovedfagsoppgave.

Hovedfagsoppgaven har innebært mange uforutsette problemer som først og fremst var av hardware-spesifikk karakter og som bidro til tidsoverskridelser. Årsaken til dette var en Linux-baert maskin som låste seg når all nødvendig software kjørte og at det viste seg svært vanskelig å finne en produsent av IP-telefonkort som var egnet for formålet i dette prosjektet. I strid med hva produsenten oppga, fungerte heller ikke medfølgende software slik det var ønskelig i dette prosjektet. Løsningen ble da å ty til tilgjengelig åpen kildekode, hvilket bød på problemer ved installasjon da dokumentasjonen for slike applikasjoner opplevdes å være svært mangelfull og all hjelp var basert på mailinglister. Det kunne ta lang tid å få svar fra slike lister, om noen da var villig til å svare. Svarene kunne også være til dels misvisende og motstridene. Eneste måten å finne ut av tingene på ble ved å prøve og feile – slikt tar tid. Videre var det også litt problematisk å få til den fysiske audiooppkobling som ble benyttet under de digitale lydopptakene i prosjektet og å finne produsenter av en svært viktig audioplugg. Uten denne ville ikke audiooppkoblingen ha vært komplett.

Selv om prosjektet har bydd på mange problemer og derigjennom mange nedturer, så har gleden vært desto større når problemene enedelig har løst seg og progresjonen har skutt fart fremover. Svært gledelig er all den støtten og hjelpen jeg har opplevet i tiden dette prosjektet har pågått. Spesielt vil jeg takke min veileder, Geir Horn, for å tålmodig ha hjulpet meg uansett hvor hektisk hans egen arbeidssituasjon har vært. Min interne veileder ved IFI, professor Olav Lysne, skal spesielt takkes for hjelpen med å anskaffe IP-telefoni kortene. Videre må Geir S. Hellum (EKY), Olav Storstrøm (EKY), Espen M. Rutger (tidligere EKY-ansatt) og min gode venn Morten Lauritsen takkes for all Linux-hjelp. Jeg må også takke Bjørn Bakka (EKY) for at han satte meg på idéen med å bruke digitale lydopptak, og Mona Kolstad (EKY) for at hun lånte meg en laptop til lydopptakene. Alle andre ved Sintef EKY, spesielt tidligere avdeling 7250, som jeg har hatt kontakt med, må takkes for å ha tatt meg imot med åpne armer. Avinor og alle flygelederne som deltok i prosjektet skal ha en stor takk. Mine foreldre skal ha takk for all støtte og for å ha hjulpet meg korrekturlesing. Sist og ikke minst, må min kjære samboer, Camilla Thalmann, ha en stor takk for å ha holdt ut med meg når det har stormet som verst. Hun er en solstråle som har gjort en tidsvis grå hverdag lys og trivelig – uten hennes varme støtte ville ikke det ha vært mulig å fullføre prosjektet. Hun skal også ha stor takk for deltagelsen i lydopptakene.

Oslo, 15. mai 2003.
Thomas Tomter

Sammendrag

Denne rapporten handler om subjektiv talekvalitetstesting av IP-telefonitale brukt i luftfartskommunikasjon (her: typisk mellom pilot og flygeleder). Testingen har vært webbasert og har foregått ved at flygeledere har lyttet til digitale lydopptak beskjemmet med kontrollerte feil som de lastet ned fra prosjektets hjemmeside. Disse lydopptakene var basert på replikkutveksling, hvor replikkene var hentet fra opplæringsmateriale i flygeledertale og illustrerte kommunikasjon mellom fly og tårn.

Det ble det utviklet et socket-program, *dgramServer*, som ble benyttet til å innstille en viss mengde pakketap i prosent eller pakkeforsinkelse i millisekunder som IP-telefonitalen ble utsatt for. På denne måten ble det *simulert* at talen ble utsatt for varierte nettverksforhold. Slik ble talekvaliteten på lydopptakene også variert.

Fra hjemmesiden lastet flygelederene også ned et evalueringsskjema som de fylte inn etter avspilling av lydopptakene og returnerte via e-post. Dette skjemaet inneholdt skalaer for bedømmelse av talekvalitet, hvor mye innsats som trengtes for å forstå talen og om talekvaliteten var tilstrekkelig for luftfartskommunikasjon.

De viktigste funnene i testingene var:

- Et flertall på over 50% av flygelederne bedømte at alle testparametrene for pakkeforsinkelse holdt en talekvalitet tilstrekkelig for ATC.
- Et flertall på over 50% av flygelederne bedømte at alle testparametrene for pakketap *ikke* holdt en talekvalitet tilstrekkelig for ATC.
- Pakkeforsinkelse har mindre negativ påvirkning på talekvaliteten enn pakketap.
- Pakketap opptil 5% har potensiale til å bli godkjent for bruk i ATC.

Kapittel 1

Innledning

1.1 Et utdrag av telefoniens historikk

I 1854 foreslo franskmannen Charles Bourseul at det var mulig å sende menneskelig tale ved hjelp av elektrisitet [NoT98]. I 1876 førte Alexander Graham Bell den første telefonsamtalen i historien [NoT98]. Han overførte lyden mellom to telefonapparater ved å fange opp lydvibrasjonene og omdanne disse til elektriske signaler, som igjen ble overført via en elektrisk leder til mottageren. I 1880 åpnet den første manuelle telefonsentralen i Kristiania [NoT98] og dermed hadde telefonien endelig nådd Norge. I 1913 ble den første privateide automatiske telefonsentralen tatt i bruk internt i kontorbygningen til Siemens Norske Aktieselskab, mens i 1917 startet automatiseringen av de offentlige telefonsentralene [NoT98]. Utskiftningene gikk sakte og var ikke fullført før på 1980-tallet.

Fra midten av 1960-tallet begynte Televerket¹ med innføringen av digitale nettelementer, men det var først på 1970-tallet at digitaliseringen skjøt ekstra fart ettersom "*Puls Code Modulation*" (PCM)-systemene ble implementert. I 1960-årene ble også lyd kvaliteten en del bedre som følge av den digitale overføringen og at switching ble innført [Schul99]. PCM-systemet gjorde at stemmen ble digitalisert i sentralen før den ble overført digitalt til endesentralen for anropet [Sør00]. Slik kunne man benytte regeneratorer istedenfor analoge forsterkere for å bevare signalene og dermed unngikk man å forringe lyd kvaliteten. Videre åpnet denne digitaliseringen for overføring av signaler via fiberoptikk. Programmerbare switcher kom på 1970-tallet, hvilket åpnet for tonesignaler og lokale tjenester slik som "samtale venter" [Schul99]. Den utbredte implementeringen av såkalt "*Out-of-band Common Channel Signaling Systems*" på 1980-tallet, markerte en overgang fra analog transmisjon og signalering til digital linjeswitchet transmisjon og pakkebasert signalering [Schul99].

¹ I dag: Telenor.

Talen eller annen mediastrøm fikk nå egne kanaler², mens signaleringen delte én pakkebasert kanal. "Signaling System Number 7" (SS7³) er et eksempel på slike systemer. De første telefonsentralene ble i Norge digitalisert i 1986 etter mye planlegging. Siste skrittet i digitaliseringen av telefonien i Norge gikk ut på å digitalisere linjen helt frem til abonnenten via ISDN⁴ [Sør00], ADSL⁵ eller VDSL⁶.

Utviklingen av telefonien i retning av en infrastruktur bestående av kun pakkebaserte nettverk, fikk en viktig debut med Internett-telefonien (dette defineres lengre ned på siden). Selv om Internetts allmenne masseutbredelse og kunnskapen om "Internet Protocol" (IP) for folk flest er relativt ny (i historisk perspektiv), er ikke det å sende tale over pakkebaserte nettverk en teknologi som er "nyklekket". Faktisk ble de første vitenskapelige artikkelene om hvordan å overføre tale over pakkebaserte nettverk publisert tidlig på 1970-tallet, og de første eksperimentene på overføring av tale over pakkebaserte nettverk utført i august 1974 [Schul99]. I disse eksperimentene ble det overført sanntids tale via pakkebaserte nett mellom University of Southern California's Information Science Institute og Massachusetts Institute of Technology's Lincoln Laboratories (MIT). De første "Request for Comment" (RFC)-dokumentene fra "Internet Engineering Taskforce" (IETF) om pakkebasert tale ble utgitt i 1977 [Schul99]. Utviklingen gikk tregt frem til 1991-1992 hvor det ble utført eksperimenter på pakkebasert audio på "Defense Advanced Research Projects Agency (DARPA) Research Testbed network" (DARTnet) og de første IETF-møtene ble sendt multicast over "Multicast Backbone"⁷ (Mbone) [Schul99].

Fra cirka 1994 [McK00] kunne man bruke Internett til å laste ned software på en multimedia-PC som gjorde det mulig å holde en samtale med en mottager med samme software via Internett, såkalt *Internett-telefoni*. I 1995 var Vocaltec⁸ blant de første til å levere applikasjoner som tillot dette [Schul99]. Fra rundt 1996 ble Internett-telefonien forbedret ved at det ble mulig å ringe til en fasttelefon [McK00]. Fra 1997 ble det mulig å ringe fra telefon til telefon med Internett eller IP-baserte nettverk som underliggende transmisjonsmetode [McK00], såkalt "IP-telefoni".

1.2 Luftfartskommunikasjon i et historisk perspektiv

Før det kan forklares på en god måte hva denne rapporten handler om, er det nødvendig med en liten avsporing:

Den første⁹ flyvningen skjedde 17. desember 1901 i North Carolina, USA. Flyet var en skrøpelig konstruksjon bestående av metall, tre og tekstil og kjempet hardt for å komme i luften. Ferden fraktet ett menneske 260 meter. Likevel, dette var et viktig steg for folk i alle land til å eksperimentere videre. Idag har luftfarten utviklet seg til å

² Definert som "A connection between initiating and terminating nodes of a circuit" i <http://www.telecomterms.com/>.

³ Det kan leses mer om SS7 i kapittel 8.3 i [Rik98].

⁴ Integrated Services Digital Network

⁵ Asymmetric Digital Subscriber Line (DSL).

⁶ very high bit-rate DSL. VDSL kan leses mer om her:

<http://computer.howstuffworks.com/vdsl.htm>

⁷ Et forskningsnett som brukes til digital kringkasting av "live" lyd og bilde via Internett [Tan96].

⁸ www.vocaltec.com.

⁹ I følge sidene til ICAO: <http://www.icao.org/cgi/goto.pl?icao/en/pub/memo.htm>.

være enormt viktig samfunnsmessig og økonomisk, når det gjelder å transportere både folk og gods over lengre avstander. Ikke minst har luftfarten vært viktig for å styrke handelen og relasjoner på tvers av landegrenser og kontinenter. Allerede på 1940-tallet hadde sivil luftfart fått en viss utbredelse. I 1944 kom derfor delegater fra 52 land sammen og ble enige om en traktat som opprettet ”International Civil Aviation Organization” (ICAO¹⁰) [Shil91]. ICAOs fremste oppgave skulle være å jobbe med å forbedre og trygge luftfartssikkerheten. Kampen for å trygge sikkerheten i sivil luftfart kunne kun vinnes ved innføring av enhetlige regler for luft- og luftfartskontroll, og en enhetlig uttrykksmåte¹¹ mellom luft og bakke.



Figur 1.1: Fra en større flymesse i Paris 1909¹².

I luftfartens veldig tidlige fase fløy piloter uten radio og måtte navigere etter veier og jernbane. Mens flyene fortsatt befant seg på bakken, ble pilotene nødt til å ty til håndsignaler. Det ble raskt oppdaget at det var påkrevet en måte å kommunisere med bakken på. Når man først fant ut av hvordan de skulle løse problemet, var det spesielt to typer telekommunikasjon som luftfarten gjorde bruk av, *radiokommunikasjon* og *radionavigasjon* [Shil91]. Radiokommunikasjon mellom bakke og fly ble utviklet under siste del av 1. verdenskrig, og radioutstyr ble brukt da Alcock og Brown fløy over Atlanterhavet for i 1919 [Pett02]. Inntil 1928 måtte flygere stole på astronavigasjon over vann. Over land brukte man i tillegg jernbanelinjer, bynavn malt på hustak og lyskastere om natten til å navigere etter. I 1928 ble det gjort bruk av såkalte retningsgivende ”radio-beacon”, spesielt beregnet for fly (hvor retningen blir vist på instrumentene om bord i flyet), under flyging fra Oakland til Hawaii. Rundt 1930 hadde alle store flyselskaper radiopeilere på bakken. Disse kunne på forespørsel gi flygeren retning til senderen. I 1935 utviklet PanAm et retningsgivende radiofyr med en rekkevidde på 2.400 miles, og da Howard Hughes i 1938 satte verdensrekord i flyging verden rundt, hadde han organisert en kjede med skip og bakkestasjoner utstyrt med radiofyr. Flyet hadde hele tiden kontakt med minst en stasjon. I Norge var

¹⁰ www.icao.org.

¹¹ På engelsk: ”phraseology”.

¹² Hentet fra: <http://www.icao.org/icao/en/ro/urnat/history01.htm>.

det først etter landets første alvorlige flyulykke i 1936 at flyradionavigasjonssystemet ble kraftig bygget ut [Pett02].

Idag er luftfarten totalt avhengig av radiokommunikasjon, selv før avgang er både piloter og crew helt avhengig av slik kommunikasjon for å koordinere bakkebevegelser og take-off [Shil91]. I luften er dette eneste måten å oppnå kontakt med bakken på for navigering og kommunikasjon. Når det gjelder sikkerheten for luftfartskommunikasjon trekker [Shil91] paralleller med kringkasting. For, om det er forstyrrelser i en kringkasting vil dette "kun" medføre tap av forståelighet eller gjør det nødvendig å retransmittere. Den samme graden av forstyrrelser vil i luftfart kunne utsette et fly for fare ved at navigeringen kan bli avbrutt eller bli utsatt for feil, eller forstyrre en kritisk instruksjon til piloten. Fordi flyene beveger seg så fort og frakter så mange mennesker, er det naturlig nok lite rom for feilmarginer. [Shil91] skriver at ICAO i 1983 nedsatte en gruppe for å forske på nye måter å løse begrensningene i dekningsområdene med radiokommunikasjonen. Denne gruppen konkluderte med at fremtidens løsning for å ivareta såkalte "communication, navigation and surveillance" (CNS)-funksjoner var via satellittkommunikasjon. I 1992 godkjente ICAO introduksjonen av satellittbasert CNS¹³ og planen er å implementere dette i løpet av det første tiåret av 21. århundret¹⁴.

1.3 Luftfartskontroll

Det kommunikasjonssystemet som brukes i luftfartskontroll (ATC)¹⁵ i dag, kalles "*Voice Communications and Control System*" (VCCS). VCCS leverer talekommunikasjonstjenester mellom flygeledere innenfor samme kontrollcenter (*interkom*), mellom flygeledere i forskjellige kontrollcenter, og mellom kontrollcenter og eksterne landbaserte nettverk (inkluderende landbaserte nettverk for mobile brukere, såkalt *bakke-til-bakke*¹⁶ kommunikasjon). Den kanskje viktigste oppgaven til VCCS er likevel å levere kommunikasjonstjenester mellom flygeledere og flypersonnel, såkalt *bakke-til-luft*¹⁷ kommunikasjon.

De viktigste karakteristikene som avgjør hvilke typer ATC-applikasjoner et VCCS-system er egnet for, er blant annet [NFR01]:

➤ **Skalerbarhet**

Et VCCS-system må være kost-effektive for både små installasjoner som betjener noen få flygeledere i et kontrolltårn og for større installasjoner som betjener flere hundre flygeledere i et travelt kontrollcenter.

➤ **Pålitelighet og tilgjengelighet**

Det stilles strenge krav til VCCS med hensyn på pålitelighet og tilgjengelighet. For eksempel vil et par sekunder avbrutt kontakt mellom et fly og kontrolltårn kunne gi katastrofale utfall. Følgene av dette er at det stilles like strenge krav til mer desentraliserte deler av systemet som for sentral deler av VCCS. Det stilles også strenge krav til den fysiske påliteligheten til utstyret, karakterisert som "*Mean Time Between Failure*" (MTBF). Den totale funksjonelle tilgjengeligheten

¹³ Fra ICAO: <http://www.icao.int/icao/en/atb/fep/c257-foreword.htm>.

¹⁴ Fra ICAO: <http://www.icao.int/icao/en/pub/MEMO.htm> og <http://www.icao.int/icao/en/nr/pio9614.htm> (dokument nr PIO 14/96).

¹⁵ På engelsk: "Air Traffic Control" (ATC). Heretter vil hovedsakelig det engelske begrepet bli benyttet.

¹⁶ På engelsk: "Ground-to-ground".

¹⁷ På engelsk: "Ground-to-air".

til systemet er avhengig av kvaliteten på software applikasjonene med hensyn på repeterbarhet og forutsigbarhet, av kvaliteten på tjenestene levert av VCCS's interne infrastruktur og av vedlikeholdsevnen til systemet. Det viktigste kravet til tilgjengelighet retter seg allikevel først og fremst mot funksjoner med høy prioritet, slik som å foreta og besvare anrop over radio, via telefon eller interkom.

➤ **Varsling og vedlikeholdsevne**

På bakgrunn av de strenge kravene til VCCS-systemer er det spesielt viktig at feil blir detektert, varslet og utbedret så fort som mulig. Jevnlige og regelmessige metoder for automatisk systemsjekk bør være iverksatt for å detektere feil hurtigst mulig. Slik oppnår man å minimere såkalt "Mean Time to Repair" og samtidig sikre størst mulig oppetid. Alle feil som kan ha betydning for tilgjengeligheten skal rapporteres til flygelederen så kjapt, pålitelig og entydig som mulig, for å unngå at en spesiell tjeneste ikke er tilgjengelig ved behov. Det generelle kravet til systemet når det gjelder robusthet og feiltoleranse, er at ikke skal tillates å eksistere udetekterte feil.

➤ **Oppstarts- og responstid**

Systemet skal typisk være driftsklar innen 30-60 sekunder etter at strømmen er slått på. Den viktigste responstiden er tiden fra "snakkeknappen"¹⁸ er trykket til samsvarende elektriske signalet er å se på systemets grensesnitt, såkalt "*Push-to-talk (PTT) time*". Responstiden på PTT skal ikke være større enn 25-50 ms. Tiden det tar å opprette et interkom-kall skal ikke overstige 100 ms. Lyden skal komme igjennom senest 100 ms etter at mottageren har besvart kallet. Responstiden på et innkommende radiosignal er tilnærmet lik tiden for PTT, altså cirka 25-50 ms. For utgående telefonsamtaler, skal samtalen være oppsatt innen 200 ms. Tilsvarende skal innkommende telefonsamtaler være opprettet etter 200 ms.

For VCCS-nettverk er det også et krav om det må være mulig å opprette en forbindelse mellom to noder over minst to uavhengige veier for å unngå såkalte "*single point of failure*". Videre er det strenge krav til lave forsinkelse, et eksempel på dette er responstiden for PTT på 25-50 ms som skal overholdes for 98% av tilfellene [NFR01].

Med bakgrunn i ATC-systemenes høy kostnad har det blitt laget flere *simulatorer* basert IP-teknologi. Blant annet har man trent opp flygeledere ved *Eurocontrol*¹⁹ og København Lufthavn Kastrup med slike simulatorer, hvorav det mest kjente systemet er *AudioLAN*²⁰. AudioLAN ble startet opp i 1996 av "Eurocontrol Research and Development Centre" og er et kommunikasjonssystem for tale via radio- og telefon basert på software og IP. Dette systemet har blitt kommersialisert og blitt tatt i bruk i flere installasjoner for simulering og trening av flygeledere.

Blant fordelene ved å bruke IP-basert teknologi i ATC-systemer, er at tale- og datatrafikk kan integreres. På denne måten kan man blant annet få overført digitale radarbilder mellom forskjellige lokasjoner samtidig som beskjeder kan utveksles mellom bakke og luft. Videre gir dette muligheter for å basere seg på datautstyr, hvilket er mye billigere enn spesiallaget ATC-utstyr. IP-teknologi vil også tillate mer distribuerte systemer, hvilke er mindre sårbar for feil ett sted i systemet og er mer robuste, da belastningen er fordelt rundt i systemet.

¹⁸ På engelsk: "push-to-talk button". Heretter vil det engelske uttrykke brukes.

¹⁹ www.eurocontrol.fr

²⁰ <http://www.eurocontrol.fr/ino/>

1.4 Årsaken til at IP-telefoni er interessant for reell ATC-bruk

Grunnen til at det er aktuelt å introdusere IP-telefoni for ATC har bakgrunn i et ønske blant luftfartsmyndigheter om å etablere en *sentralisert* struktur (se figur 1.2) for organiseringen av luftfartskontroll. Dette innebærer at man fra en sentralisert hovedkommunikasjonssentral kan styre flytrafikken på mindre og perifere flyplasser. På denne måten kan den enkelte flyplass kontrollere sitt luftrom under trafikkerte deler av døgnet og deretter overlate styringen til en hovedkommunikasjonssentral i perioder med liten trafikk, som for eksempel nattetid. Slik kan luftfartskontrollen drives mer kostnadseffektivt. En slik sentralisert struktur er spesielt aktuelt i land med en veldig spredt befolkning over store geografiske områder, som for eksempel Norge og Canada, hvor det finnes mange mindre flyplasser rundt om i alle landsdeler. I en slik struktur, finnes det to måter å frakte ATC-talen på, via en *leid* linje eller *privat* linje.

1.4.1 Leid linje

Leid linje innebærer sannsynligvis at det vanlige telenettet (benevnt *offentlig transportnettverk* i figur 1.2) benyttes og at det blir stilt til rådighet en *gjennomsnittelig* ytelse i nettverket for frakt av talen. I switcher²¹ underveis til destinasjonen vil det, imidlertid, kunne bli tilbudt en dedikert kapasitet.

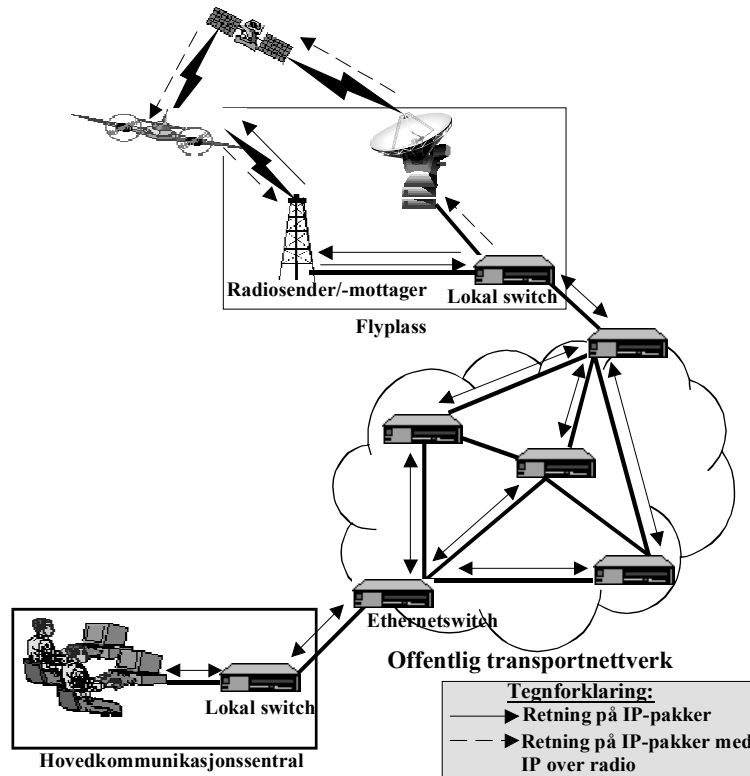
Det offentlige transportnettverket vil sannsynligvis være i gradvis utvikling, hvor mer og mer av det vil bli basert på Ethernet. Dette vil i så fall innebære at eksisterende switcher i det offentlige transportnettverk vil bli erstattet av Ethernetswitcher. Disse vil være IP-baserte og dermed *vil* talen i den sentraliserte strukturen bli sendt over IP-nettverk til den aktuelle flyplassen. På denne måten vil det indirekte være aktuelt å nyttegjøre IP-telefoni til å sende flygeledertale²².

Når IP-pakkene har nådd flyplassen, vil nødvendige konverteringer bli foretatt slik at talen kan bli transmittert via en lokal sender til flyene over en radiolink. På sikt vil det bli implementert en transmisjonsmetode som tillater frakt av IP-pakker over radio. Eurocontrol har et slikt prosjekt pågående, kalt "IPsky"²³. Typisk vil flyet her motta IP-pakker over satellitt, mens flyet vil sende IP-pakker via radio til bakke. Men, inntil dette blir en realitet, må en type konverteringsmekanisme være tilstede slik at talesignalene kan bli omformet til et format passende for radiolinken som benyttes for transmisjonen. Ved å installere et IP-nettverk ombord på flyet vil dette åpne for utnyttelse av tilgjengelig båndbredde ned til flyet til Internett-aksess for de reisende. Dette vil være mest aktuelt når flyet er i stabil høyde og når kapasiteten ikke brukes til å sende ATC-tale.

²¹ Begrepet "switcher" vil bli brukt i forbindelse med lag 2 (som for eksempel Ethernet, hvor det switches på MAC (Medium Access Control)-adresser), mens begrepet "ruter" vil brukes i forbindelse med lag 3 (som for eksempel på det offentlige Internett, hvor det routes på IP-adresser).

²² Ofte benevnt "Air Traffic Control" (ATC)-tale. Dette begrepet vil bli brukt i det følgende.

²³ <http://www.eurocontrol.fr/ino/>



Figur 1.2: I en sentralisert struktur vil flygeledere i en hovedkommunikasjonssentral kunne dirigere flytrafikken i luftrommet over en fjerntliggende flyplass i stille perioder. Slik kan man nedbemanne antall flygeledere på jobb på mindre trafikkerte flyplasser. ATC-kommunikasjonen blir fraktet over et IP-basert transportnettverk til flyplassen. IP-pakkene med talen blir sendt flere veier igjennom transportnettverket for å være sikker på at i alle fall én vei fører frem. Til å begynne med vil IP-telefonitalen bli konvertert til et passende format og transmittert via radiolink til flyet. Når det blir mulig å sende IP-pakker over radio, vil flyene motta IP-trafikk (tale og eventuelt Internett) via satellittlink. Trafikk fra flyet til bakken vil fortsatt bli sendt over radiolink.

Det eksisterer også et kostnadsmotiv for hvorfor IP-telefoni er interessant for bruk i ATC. Det er nemlig å kostbart å utvikle telefonsentraler skreddersydd for ATC, hvor det også brukes en del spesiallaget maskinvare for å sikre en robusthet god nok for ATC. Den bakenforliggende ideen er her å benytte standard datautstyr som er laget for et langt større marked, enn det spesiallaget ATC-utstyr er, og som dermed er mye billigere å anskaffe. Dessuten har datautstyr en tendens til å stadig falle i pris. Dette er videre utstyr som er COTS²⁴ og som det dermed er god tilgjengelighet på. Til en viss grad kan det også eksistere IP-nettverk lokalt på flyplassen som kan opprustes for mindre kostnader og brukes til å frakte ATC-talen frem til sender.

Frakt av tale over slike leide nett vil innebære en risiko for at pakker tapes i switcher. Hovedårsaken til dette, er at talen vil konkurrere om båndbredden med annen trafikk i nettet. Luftfartsmyndigheter kan ikke kontrollere transportnettverket mellom avsender- og mottagerlokasjonene. I disse lokasjonene, derimot, kan det sørges for prioritering og dedikert kapasitet for talen i lokale nettverk. Så langt er ikke dette mulig å få til i transportnettverket, siden prioriteringsmekanismer per i dag ikke er implementert i Ethernetswitcher. For å være sikker på at pakkene kommer frem, vil de måtte sendes

²⁴ Forkortelse for "Commercials Of The Shelf".

redundant over transportnettet, altså over flere veier, til destinasjonen. Hensikten med dette er at IP-pakkene skal komme frem på minst én av veiene.

En ikke helt utenkelig situasjon er at IP-pakkene kan bli sendt via en omvei til destinasjonen fordi en sentral switch er nede. Denne eventuelle ekstra avstanden IP-pakkene kan bli transportert over, vil kunne påvirke talekvaliteten på IP-telefonitalen ved at IP-pakkene blir så forsinket at talen ikke kan gjenskapes på mottagersiden. Videre kan denne ”omkjøringen” medføre ekstra belastning av switcher som ikke er beregnet for frakt av ATC-tale. Følgene av dette igjen kan være at IP-pakker kan tapes i disse switchene.

I og med at IP-telefonitalen vil kunne bli utsatt for både *pakketap* og *pakkeforsinkelser*, som kan ha store innvirkning på talekvaliteten, blir det derfor svært viktig å teste i hvilken grad dette påvirker talekvaliteten. Ikke minst blir det viktig å sjekke om sikkerhetskravene til ATC-tale fortsatt kan oppfylles.

1.4.2 Private linjer

Som et alternativ til å leie kapasitet på det offentlige transportnettverket, kan luftfartsmyndigheter etablere egne lukkede transportnettverk, ved å for eksempel grave ned sine egne kabler. I motsetning til løsningen med leide linjer, vil det slik tillates prioriteringer og ressursreserveringer for IP-telefonitalen. For å være sikker på at talen ikke vil ”kollidere” med annen trafikk i nettet og at talekvaliteten blir utmerket, kan linjen helt enkelt dedikeres til tale.

En klar ulempe med private linjer er at når for mange slike kabler blir trukket, øker sannsynligheten for uforutsette kabelbrudd. Denne sannsynligheten øker også med *lengden* på hver enkelt kabel, med *mange* lange kabler blir denne sannsynligheten enda høyere. Følgene kan bli fysiske brudd på kabelen ved for eksempel gravearbeider. Slik kan luftfartssikkerheten bli satt i en alvorlig situasjon bare fordi en enkelt entreprenør ikke har gjort grundig nok undersøkelser eller innhentet nødvendig klarering før graving ble påbegynt.

Videre, en slik løsning blir svært kostbar når det skal trekkes kabler fra en hovedkommunikasjonssentral til ethvert endepunkt for talen over hele landet eller innenfor en region. Om man da skal sikre seg mot fysiske skader ved å grave ned redundante kabler, for eksempler tre til fire kabler i alle retninger utfra hovedkommunikasjonssentral, blir dette enda dyrere.

På denne måten står valget mellom leid- eller privatlinjer tilbake som en risikovurdering. En vurdering mellom om hva som truer luftfartssikkerheten mest, om deler av talen blir borte grunnet konkurrerende datatrafikk i nettet eller en gravemaskin fysisk kutter over en kommunikasjonskabel. Det finnes reelle eksempler fra Europa hvor slikt har skjedd og hvor all luftfartskommunikasjon ble brutt for en lengre tidsperiode.

1.4.3 Desentralisert struktur

I land med flere større flyplasser vil en desentralisert løsning være mer aktuelt. Her vil hver enkelt flyplass, i større utstrekning, ha ansvaret for kontrollere luftrommet sitt kontinuerlig, enn i land med flere mindre flyplasser. I en slik desentralisert struktur vil det likevel være en mulighet å benytte IP-telefoni, men det er ikke strengt tatt nødvendig. Det eksisterer tross alt et kommunikasjonssystem som fungerer helt utmerket. Det samme gjelder for såvidt også enkelt flyplasser i en sentralisert struktur. Det kan også være tenkelig at mindre flyplasser i omegn til en større kan styres fra den større i perioder med liten trafikk, noe liknende ideen bak den sentraliserte løsningen. Her kunne det også tenkes at talen mellom flyplassene kunne bli fraktet over IP-baserte nettverk.

Anvendelser av IP-telefoni i slike scenarioer vil kun være aktuelt om det eksisterende systemet er såpass gammelt, så utstabilt at det trenges å byttes ut eller at det skal nyetableres et kommunikasjonssystem ved for eksempel ved åpning av nye flyplasser. I slike situasjoner blir spørsmålet om etablering av IP-telefoni et rent kostnadsspørsmål. De samme kostnadsbegrunnelsene gitt i kapittel 1.4.1 vil også være gjeldende her.

I en desentralisert struktur vil det være mulig å etablere et isolert LAN utelukkende til bruk for frakt av taletrafikk. Slik kan det dermed sikres at annen trafikk i nettet ikke vil påvirke talekvaliteten. Også i denne strukturen vil det i utgangspunktet være aktuelt å benytte IP til å sende tale frem til sendere og deretter benytte radio til å transmittere talen til flyet, inntil det er mulig å sende IP over radio.

1.5 Motivasjon for prosjektet

IP-telefoni er i høyeste grad en anvendbar teknologi og det finnes mange implementerte løsninger. Bladet ”*Nettverk og kommunikasjon*”²⁵ presenterer en løsning med en spesiallaget telefon fra leverandøren *Doro*²⁶ for trådløs IP-telefoni over det trådløse nettverket 802.11b. ”*Network Magazine*”²⁷ presenterer i en lengre artikkel en bank i Wisconsin, USA, som har implementert IP-telefoni og som bruker det i sitt daglige virke. Avisen ”*Computerworld*”²⁸ skrev i desember 2001 om *Telenor Business Solutions*²⁹ som tilbyr bedriftskunder IP-telefoni. En representant for selskapet uttaler for øvrig at man anslår at IP-telefoni vil være dominerende i slutten av dette tiåret. Videre, i begynnelsen av januar 2003 uttalte den nye administrerende direktøren i TeleDanmark Norge til Finansavisen³⁰ at IP-telefoni er et satsningsområde. Det finnes mange kjente og ukjente leverandører og tjenestetilbydere innen IP-telefoni. Noen av de mest kjente leverandørene er Ericsson³¹, Siemens³², Cisco³³, 3Com³⁴, Nortel³⁵ og Vocaltec³⁶.

²⁵ Nummer 6, 2001.

²⁶ Kjent leverandør av blant annet trådbaserte og –løse telefoner. <http://www.doro.com/>.

²⁷ September 2001, side 74-79.

²⁸ Nr 89, fredag 14. desember 2001, side 8.

²⁹ <http://www.telenor.no/business/>

³⁰ Side 14, årgang 12, Nummer 4, mandag 6. januar 2003.

³¹ www.ericsson.com

³² www.siemens.com

³³ www.cisco.com

➤ **Dette prosjektets innhold**

Dette prosjektet handler om en ny og annerledes måte å frakte ATC-tale, nemlig i IP-pakker over IP-baserte nettverk. På denne måten kan IP-telefoni på sikt erstatte dagens system for talekommunikasjon mellom blant annet flytårn og fly (såkalt "bakke-til-luft"), og mellom tårn og diverse beredskaps- og serviceenheter (såkalt "bakke-til-bakke"). I tillegg kan teknologien benyttes i ATC-opplæringsøyemed. Denne rapporten omhandler subjektiv talekvalitetstesting av slik tale, med flygeledere som deltagere i testingen. Etter vår kjennskap har det ikke tidligere blitt utført testing av talekvalitet for ATC-tale overført i IP-pakker via et IP-basert nett *med* flygeledere, som subjektivt kan bedømme om talen holder en talekvalitet tilstrekkelig for ATC-bruk.

➤ **Vanlig anvendelse av IP-telefoni**

Anvendelsene av IP-telefoni teknologien i dette prosjektet er ny og atypisk. Motivasjonen for mer tradisjonelle IP-telefoni applikasjoner har stort sett vært, i bedriftsmiljø, å utnytte eksisterende infrastruktur (eventuelt med en rimelig opprustning) for å ringe billig mellom ulike avdelingskontorer, eller å unngå fjerntakst ved å ringe via IP-telefoni til et kontor i en gitt region for å ringe lokalt derfra til den aktuelle mottageren. Talekvaliteten på IP-telefoni må som minimum ligge på samme nivå som vanlig telefoni (PSTN³⁷) dersom denne teknologien skal tas i kommersiell og allmenn bruk. Får man ikke summetonen når man tar av røret eller at talekvaliteten er såpass dårlig at det er anstrengende å benytte IP-telefoni, vil det for de fleste være normalt å ty til det gamle telefonsystemet som man *vet* fungerer. Slik blir talekvaliteten for vanlige IP-telefoni anvendelser også et spørsmål om de eventuelle nødvendige oppgraderingene i infrastrukturen kan forsvares økonomisk.

➤ **IP-telefoni i ATC**

For ATC-anvendelser av IP-telefoni er den underliggende motivasjonen beskrevet i kapittel 1.4. Talekvalitetskravene til IP-telefoni brukt i ATC stilles på et helt annet grunnlag enn i de vanlige anvendelsene, nemlig utfra *sikkerhet*. For dårlig kvalitet kan i ytterste konsekvens forårsake tap av menneskeliv. Slik som [Shil91] sa er det uhyre viktig at meldinger forstås. Det finnes ikke rom for feilmarginer og det eksisterer kun ett krav: talen *skal* komme frem og forstås. I reelle anvendelser, vil derfor talepakkene sannsynligvis bli sendt flere veier for å være sikker på at pakkene som inneholder tale kommer frem på én av veiene. I dette prosjektet vil IP-nettverkernes påvirkning på IP-telefonitalen *simuleres*.

Det er med dette sikkerhetskravet i bakhodet at ønsket om å trekke inn flygeledere i talekvalitetstesting oppstod. Det er nemlig kun de som kan avgjøre om talekvaliteten på IP-telefoni er egnet for bruk i ATC-sammenheng. Bakgrunnen for dette er en antagelse om at de innehar en helt unik forståelse for hvilken talekvalitet ATC-kommunikasjon må holde og generelt har en bedre gehør enn vanlige personer. Det har tidligere blitt utført en rekke subjektive talekvalitetstester for IP-telefoni (hvor

³⁴ www.3com.com

³⁵ www.nortel.com

³⁶ www.vocaltec.com

³⁷ Benevnes som "Public Switched Telephony Network" (PSTN) eller "Plain Old Telephony System" (POTS) på engelsk.

[ETSI02]³⁸ er blant de nyeste), men etter vår kjennskap skal det ikke ha blitt utført slike tester med hensyn på om IP-telefoni teknologien er egnet for bruk i ATC.

Oppsummert går dette prosjektet ut på å teste talekvaliteten på IP-telefoni med flygeledere, som kan bedømme om talen holder en kvalitet tilstrekkelig for ATC. Den testmetoden som skal brukes i testingen vil i en viss grad ta utgangspunkt i anbefalingene for lytte- og konversasjonstestene beskrevet i ITU Recommendation P.800 ”*Methods for subjective determination of transmission Quality*” [ITU96] og ITU-dokumentet ”*Handbook on Telephony*” [ITU92]. Testmetoden må også ta hensyn til de kravene ATC stiller til hvordan talen skal arte seg. Blant annet kreves det at meldingene skal være korte, lett forståelige og i en dagligdags tone, hvor det ikke er tillatt med følelsesladde utbrudd eller ekstrem høflighet [Pat01]³⁹. Videre er konversasjonen halv-dupleks, det vil si at kun én prater ad gangen.

1.5.1 Problemstillinger

Med bakgrunn i motivasjonen presentert ovenfor, kan problemstillingene for dette prosjektet formuleres slik:

- Utarbeidelse av et program for å *simulere* ulike forhold i IP-nettverk ved at ulike mengder pakketap og pakkeforsinkelse kan innstilles.
- Hvilket eksperimentoppsett bør benyttes for testing av talekvaliteten under ulike simulerte nettverksforhold?
 - ↳ Herunder: Utarbeidelse av eksperimentdesign, fysisk oppkobling og konfigurering av testplattformen.
- Hvordan blir talekvaliteten påvirket av ulike simulerte nettverksforhold?
 - ↳ Herunder: Komme frem til grafer som viser talekvaliteten som funksjon av økende forringelse.
- Hva kan talen maksimalt tåle av pakketap og forsinkelser, og allikevel holde en talekvalitet tilstrekkelig for ATC-kommunikasjon?
 - ↳ Herunder: Hvilken talekvalitet må da IP-telefonien holde? Hvor mange prosent pakketap eller millisekunders forsinkelser tilsvarer dette?

1.6 Testmetode

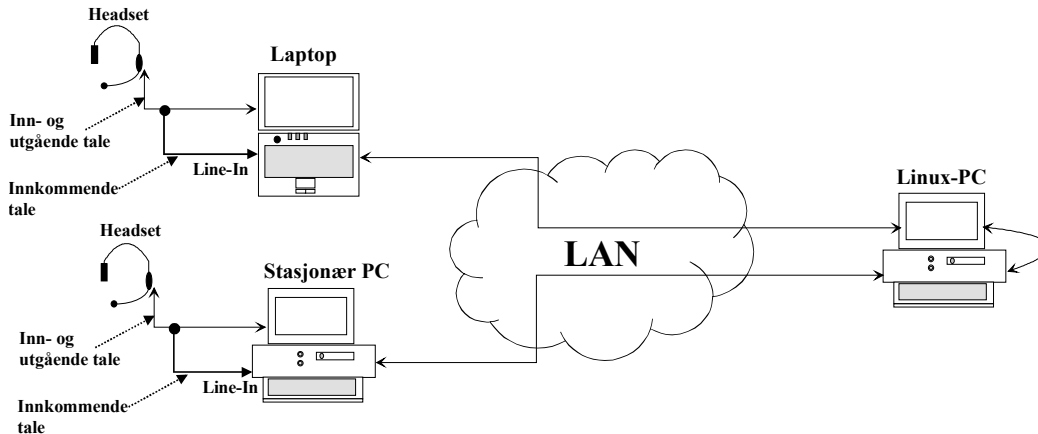
For å simulere nettverksforholdene IP-pakkene, som frakter talen, kan bli utsatt for i reelle IP-nettverk, har det blitt utviklet en socketserver som baserer seg på ”*User Datagram Protocol*”⁴⁰ (UDP), kalt ”*dgramServer*”, i programmeringsspråket C. ”*dgramServer*” ble implementert og kjøres på en Linux-PC. Årsaken til at det ble avgjort å kjøre testene med simulerte nettverksforhold, er fordi kun dette gir kontrollerbare forhold og repeterbare forsøk. Skulle ikke dette oppfylles, kunne talen alternativt ha bli sendt via IP-nettverk over varierte avstander og bli loop-et tilbake. På denne måten kunne talekvaliteten bli testet med ulike forsinkelser og med det naturlige pakketapet som kunne oppstå.

³⁸ Utførte både subjektive og objektive talekvalitetstester. Ikke trykket materiale, hentet fra http://www.etsi.org/frameset/home.htm?plugtests/04History/2002_voipsqa.htm.

³⁹ Ikke trykket materiale, se URL: <http://fly.to/kpatouras.gr>.

⁴⁰ En forbindelsesløs transportprotokoll.

Serveren kan konfigureres til å forsinke pakkene i et gitt antall millisekunder eller til å forkaste en gitt prosentandel pakker for å simulere de forsinkelsene og pakketapene som IP-pakkene kan oppleve i et reelt nettverk. I tillegg, kan programmet bytte om rekkefølgen på IP-pakkene for å simulere den omstokkingen av IP-pakker som kan forekomme i et nettverk. Ved å konfigurere forsinkelsen får man simulert at IP-pakkene blir sendt over forskjellige avstander til mottageren.



Figur 1.3: I både laptop-en og i stasjonær PC-en sitter det et IP-telefoni kort og på hver maskin kjører det en IP-telefoni klient. IP-telefonitalen blir sendt over et LAN og via en Linux-PC, hvor IP-pakkene blir manipulert i dgramServer, til mottageren. Slik kan pakketap og -forsinkelse, som IP-pakkene kan bli utsatt for i reelle IP-nettverk, simuleres. I tillegg til dgramServer, kjøres det på Linux-maskinen også en IP-telefoni gatekeeper. Denne setter opp samtalen mellom IP-telefoni klientene. I hver ende vil innkommende tale bli forgrenet og ført til inngangen på PC-ens lyd kort (line-in), slik at lyden som høres i headset-et også blir tatt opp digitalt. Opptakene blir deretter lagt ut på Internett og blir lastet ned av flygeledere som bedømmer talen.

Testoppsettet som ble brukt i dette prosjektet er vist i figur 1.3. Det er installert et IP-telefoni kort i både laptop-en og på stasjonær PC-en, og på hver av maskinene kjøres det en IP-telefoni klient som er kompatibel med IP-telefon kortene. IP-telefonitalen blir sendt over et 10 Mbps LAN og via en mellomliggende Linux-PC på vei til mottageren. På denne maskinen kjører dgramServer og en IP-telefoni gatekeeper, som brukes til å sette opp samtalen mellom IP-telefoni klientene. I hver ende blir innkommende tale forgrenet og ført til "line-in"-inngangen på PC-ens lyd kort. Her blir talen tatt opp og lagret digitalt. Slik blir talen og den tilhørende talekvaliteten som høres i høyttaleren på headset-et tatt opp. Opptakene fra hver ende blir satt sammen til én stereo lydfile, hvor man hører den ene parten i venstre lyd kanal og den andre i høyre. Slik vil det illustreres at ATC-kommunikasjonen ble overhørt av og lyttet til av en tredjepart. Det ble videre laget en hjemmeside for prosjektet, hvor prosjektet presenteres, hvor lydfile i mp3-format og evalueringsskjemaer legges ut. Her kan flygeledere enkelt laste ned lydfilene, og avspille dem i sin favoritt mp3-spiller og bedømme talekvaliteten. Etter at evalueringsskjemaet har blitt fylt ut, returneres det via e-post.

Selve talen ble generert ved at to personer, som ikke har tidligere erfaring fra ATC, deltok i en replikkutveksling på bakgrunn av et manus. Dette manuset var basert på eksempler på ATC-tale hentet fra diverse opplæringsbøker og -websider om temaet.

1.7 Rapportens struktur

Den analoge talen er nødt til å bli samlet, digitalisert og komprimert før den blir pakket inn i IP-pakker og sendt avgårde via et IP-basert nettverk til en mottager. Hos mottageren blir talen rekonstruert og avspilt. Viktige temaer i bakgrunnskapitlene blir derfor IP-telefoni, hva slags metode som bør benyttes i talekvalitetsbedømmelsene og hvilke krav som stilles til luftfartskommunikasjon. Sentrale temaer innenfor IP-telefoni blir ulike arkitekturer og talekoding. I appendiks A finnes det i tillegg en fyldig beskrivelse av signaleringsprotokollene H.323 og SIP, og i appendiks B blir sentrale ITU-dokumenter om subjektiv testing redegjort for. For øvrig, i appendiks C blir forkortelser forklart og kort beskrevet. På bakgrunn av dette er følgende struktur på rapporten valgt, hvorav kapittel 2 til 4 inneholder nødvendig bakgrunnsteori for mine bidrag som beskrives i kapittel 5 til 9:

Kapittel 2 "IP-telefoni arkitekturer"

Forklarer hva IP-telefoni er og hvilke typiske arkitekturer som finnes. Videre blir begrepene "*stillhetsundertrykkelse*" og "*comfort noise*" gjort rede for.

Kapittel 3 "Talekvalitet"

Begrepet talekvalitet blir definert og det blir sett nærmere på hvilke faktorer som påvirker talekvalitet i IP-telefoni.

Kapittel 4 "Talekoding"

Det blir forklart hvilke tre viktige steg som er nødvendig for å få digitalisert talen og hva disse går ut på. Videre blir det sett på hva en codec er og hva som er viktige attributter for dem. Til slutt gis det et eksempel på et forsinkelsesbudsjett.

Kapittel 5 "Bakgrunn for forsøksoppsettet"

Her sees det nærmere på blant annet hvordan ATC-tale arter seg. Kapitlet diskuterer videre generelt hva forsøksoppsett for subjektiv talekvalitetsbedømmelse av IP-telefoni basert ATC-tale må ta hensyn til. Til slutt redegjøres det for det valgte forsøksoppsettet, skalaer og målsetninger.

Kapittel 6 "Implementasjon og testoppsett"

Dokumenterer implementasjonen av "dgramServer" i programmeringsspråket C og det testoppsettet som ligger til grunn for talekvalitetstestene.

Kapittel 7 "Testresultater"

Resultatene fra talekvalitetstestingene presenteres.

Kapittel 8 "Diskusjon"

Overraskelser i resultatene diskuteres, i tillegg til at det sees med kritiske øyne på testmetoden og selve gjennomføringen talekvalitetstesting. Videre blir det diskutert hvilket videre arbeid som kan gjøres for å følge opp trådene fra dette prosjektet.

Kapittel 9 "Konklusjon"

Her blir det resultatene oppsummert og det redegjort for hvilke konklusjoner som kan trekkes ut fra testresultatene.

Appendiks A "Signaleringsprotokoller"

Forklarer hva signalering er og hvorfor det trenges. To av de store signaleringsprotokollene i IP-telefoni, H.323 og SIP, blir grundig gått igjennom med spesielt hovedvekt på arkitekturer.

Appendiks B "Måling av talekvalitet"

Redegjør for hvilke typer subjektive metoder for talekvalitetstesting som blir anbefalt i sentrale ITU dokumenter. Videre blir objektive testinger nevnt kort.

Appendiks C "Forkortelser"

De mest brukte forkortelsene blir forklart og kort beskrevet.

Appendiks D "Kildekode"

Kildekoden til dgramServer og dgramClient er lagt ved her.

Appendiks E "Personlig e-post"

Her finnes det en personlig e-post som omhandler radiokommunikasjons historikk i luftfartssammenheng, og en e-post fra en flygeleder som det refereres til i teksten.

Appendiks F "Teknisk spesifikasjon"

De tekniske dataene på IP-telefoni kortene "Internet PhoneJACK" og "Internet PhoneCARD" gjengis her.

Appendiks G "Manus og evalueringsskjema"

Manus og evalueringsskjemaene brukt i talekvalitetstestingene blir gjengitt.

Appendiks H "Konfigurasjon"

Diverse konfigurasjoner som har blitt brukt i prosjektet gjengis her.

Appendiks I "Excel-ark"

Excel-arket med rådataene ligger her..

Referanser

Kildehenvisningene listes opp her.

Kapittel 2

IP-telefoni arkitekturer

IP-telefoniens inntreden blir ofte assosiert med Internettets vekst, selvom det i mange tilfeller kan være slik at en IP-telefoni applikasjon ikke involverer Internett i det hele tatt, kun den underliggende lag 3-protokollen, "Internet Protocol" (IP) [McK00]. I det følgende skal det oppklares hva IP-telefoni er, hvilke avarter som finnes og hvilke applikasjoner som er de vanligste. Til slutt i kapitlet vil prosessering kort nevnes for å danne et bilde av hvordan IP-telefoni fungerer rent fysisk.

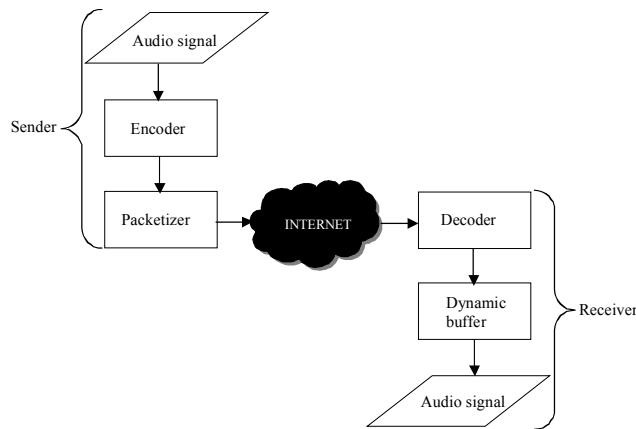
2.1 Hva er IP-telefoni?

Tiden da man klart kunne skille mellom tale og data-applikasjoner har begynt å ebbe ut. Forskjellene blir uklare og disse to typene applikasjoner tilnærmer seg hverandre med stormskritt. Denne tilnærmingen blir ofte i litteraturen kun omtalt som "sammensmeltningen"⁴¹.

Hva er så egentlig IP-telefoni? Forenklet kan man si at IP-telefoni dreier seg om å digitalisere og komprimere analog tale til et format passende for overføring i IP-pakker via Internett eller et annet IP-basert nettverk til en mottager (se figur 2.1). På grunn av at pakkene kan ankomme mottageren med ulik forsinkelse, er man nødt til å buffre opp pakkene før de avspilles for korrekt å gjenskape den analoge talen. Det at pakkene ankommer mottageren med innbyrdes ulik forsinkelse, kalles "*jitter*". "Jitter" vil bli fylligere dekket i kapittel 3 "Talekvalitet".

En fordel med IP-telefoni sammenliknet med PSTN (Public Switched Telephony Network), er at IP-telefoni gjør det mulig å kombinere data- og taletjenester og danne et tjenestespekter som ikke vil være mulig i PSTN [HNA00]. For eksempel vil det være mulig å klikke på en link på en personlig hjemmeside for å ringe innehaveren av hjemmesiden.

⁴¹ På engelsk: "convergence".



Figur 2.1: Den analoge talen må digitaliseres og komprimeres (inngår i "Encoder" i figuren), før den blir pakket inn i IP-pakker og sendt via Internett/IP-basert nettverk til en mottager. Hos mottageren må talen dekodes og buffres opp før den analoge talen gjenskapes og spilles av. (Basert på [KBS98].)

Definisjonene og variantene av IP-telefoni er mange, i denne rapporten skal definisjonene i [McK00] følges, hvor IP-telefoni brukes som et samlebegrep om de ulike måtene å overføre tale, faks og relaterte tjenester over pakkeswitchet nettverk på. Forfatterne sier at selvom man kan dele begrepet IP-telefoni inn i to hovedkategorier, *Internett telefoni* og *"Voice over IP"* (VoIP), finnes det nærmest et uendelig antall forskjellige måter å bruke IP-teknologien på som gir forskjellige type tjenester. Derfor, sier forfatterne, blir man nødt til å klassifisere tjenestene utfra hva slags *terminalenhet* som brukes i tjenesten (f.eks. PC eller telefon), hvor *"gatewayen"* som er bindeledd mellom IP-nettverket og det tradisjonelle telefonnettet (PSTN) er lokalisert, og hvilken underliggende *transmisjonsmetode* som benyttes.

For å kunne klart skille de to nevnte undergruppene av IP-telefoni fra hverandre, må man se på hva slags underliggende transmisjonsmetode som benyttes for de to. [McK00] definerer Internett telefoni til å være IP-telefoni hvor det underliggende transmisjonsnettverket er det offentlige Internettet. Andre betegnelser for dette er f.eks. *"Voice over the Net"* (VON) og *"Internet Phone"*. Videre definerer [McK00] *"Voice over IP"* som IP-telefoni hvor overføringsnettverket (eventuelt: *nettverkene*) er private og kontrollerte⁴² IP-baserte nettverk. Avhengig av det underliggende nettverket kan man f.eks. ha *"Voice over Frame Relay"* eller *"Voice over cable"*. Bruken av begrepet *"offentlig Internett"* i definisjonene over refererer til det globale, IP-baserte meta-nettverket som utgjøres av sammenkoblingen av mange offentlige og private nettverk.

I resten av denne rapporten vil *"Internett"* bli brukt om det offentlige Internettet, og begrepene *"IP-telefoni"* og *"Voice over IP"* vil i påfølgende kapitler kunne bli brukt om hverandre.

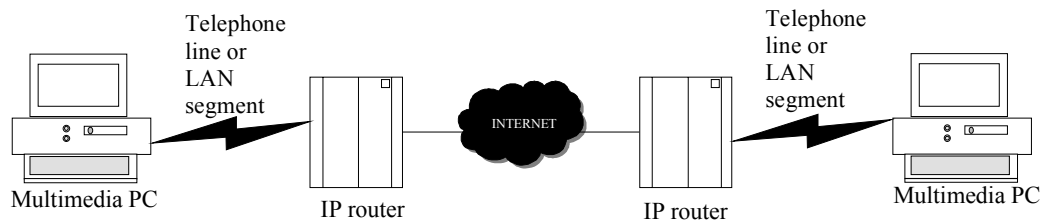
⁴² På engelsk: "managed".

2.2 Arkitekturer

Ved å kombinere ulike plasseringer av ”gatewayen” og ulike terminalenheter, slik som beskrevet på forrige side, fås de tre mest brukte arkitekturerne i IP-telefoni. I de følgende tre delkapitler skal disse sees nærmere på.

2.2.1 PC-til-PC

Denne arkitekturen utgjorde første generasjon av IP-telefoni tjenester og var tilgjengelig fra 1994 [McK00]. Ofte baserte denne tjenesten seg på at brukerne lastet ned billig software fra Internett. Brukerne måtte i tillegg være utstyrt med en PC med lydkort, høyttaler og mikrofon, og ha tilgang til en Internett-forbindelse (se figur 2.2). Samtalepartnerne deres måtte dessuten benytte samme software. Denne tjenesten var avhengig av at samtalen var avtalt på forhånd i og med at det ikke bare var å ringe mottageren. Om vedkommende ikke var online, var det heller ingen måte å nå ham/henne. Derfor var denne tjenesten ideell i en chatte-sammenheng, hvor deltagerne først kommuniserer skriftlig og avtaler å bytte til taleapplikasjonen. I denne typen tjenester er ikke en gateway mot PSTN nødvendig fordi samtalerne ikke blir switchet av PSTN, og den underliggende transmisjonsmetoden er det offentlige Internettet [McK00]. Denne tjenesten var ofte plaget med dårlig lyd kvalitet og var lite brukervennlig, og antas derfor å ha utgjort en minimal trussel mot de tradisjonelle telefonitjenestene [McK00].



Figur 2.2: PC-til-PC arkitektur. Samtalepartnerne måtte bruke samme software og være utstyrt med en multimedia-PC. Ofte ble denne arkitekturen brukt i chatte-sammenheng, hvor man vekslet mellom tekstlig og telefonbasert konversasjon. (Basert på [KBS98].)

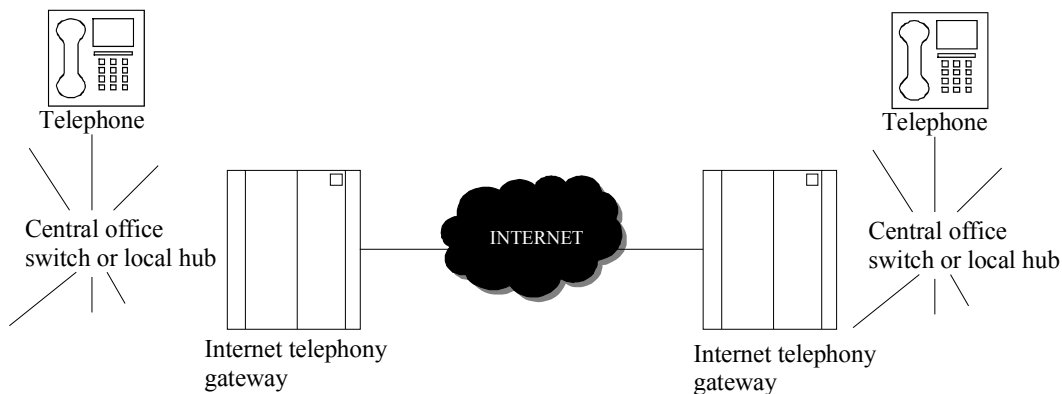
2.2.2 PC-til-telefon

PC-til-telefon (eller: PC-til-faks) var videreutviklingen av PC-til-PC applikasjonen og var tilgjengelig fra rundt 1996 [McK00]. Denne tjenesten ga brukerne en utvidet mengde samtalepartnere, da man nå kunne treffe mennesker på telefon, via faks eller på mobiltelefon [McK00]. For tjenestetilbyderne ble situasjonen mer kompleks, fordi nå måtte samtaler bli tatt betalt for og rutingsavtaler bli inngått. I denne typen arkitektur foregikk konverteringen av analog tale til tale innpakket i IP-pakker i den orginerende⁴³ PC-en, denne prosessen ble reversert i gateway-en til en ”*Internet Telephony Service Provider*” (ITSP). ITSP-en var videre ansvarlig for å slå telefonnummeret til den oppkallede parten og viderebringe talen i begge retninger.

⁴³ Den parten som tar initiativ til samtalen.

2.2.3 Telefon-til-telefon

Denne arkitekturen har eksistert siden 1997 og er den viktigste av de tre hittil nevnte, fordi den involverer flest kommersielle interesser [McK00]. Denne arkitekturen ble også sett på som selve fundamentet for videre utvikling av IP-telefoni, rett og slett fordi folk flest liker å foreta sine samtaler via en vanlig telefon [McK00]. Ulempen med denne arkitekturen, sammenliknet med de to foregående, er at denne krever en del mer investering ettersom det er nødvendig med en orginerende gateway og en terminerende⁴⁴ gateway i PSTN. Det offentlige Internettet kan brukes som underliggende transmisjonsnettverk (se figur 2.3), men det vanligste er at denne arkitekturen baserer seg på lukkede, kontrollerte IP-nettverk og at det eventuelt eksisterer formelle betalingsrelasjoner mellom ITSP-er og telefonselskaper [McK00]. Slik blir denne arkitekturen opererende i parallell med tradisjonell telefoni og har lite med det offentlige Internettet å gjøre, slik som antydnet i innledningen til dette kapitlet.



Figur 2.3: Telefon-til-telefon arkitektur som benytter Internett som underliggende transmisjonsnettverk. I de vanligste VoIP-applikasjonene vil en PABX⁴⁵ utgjøre det som i figuren er benevnt "Central office switch or local hub". (Basert på [KBS98].)

I figur 2.3 er det vist en telefon-til-telefon arkitektur hvor talen fraktes over Internett. For en hjemmebruker vil den orginerende parten ringe via en telefonsentral til nærmeste gateway, som krever en autentisering av brukeren av faktureringshensyn⁴⁶ [KBS98]. Videre blir brukeren bedt om å slå nummeret til mottageren. Dette nummeret blir slått opp og matchet oppmot IP-adressen til den gateway-en mottageren er koblet til. Mottagerens gateway etablerer en forbindelse til mottagerens telefonsentral, som igjen sørger for at det ringer hos mottageren.

Gateway-en er implementert i hardware på "Digital Signal Processor" (DSP) og "Application Specific Integrated Curcuit" (ASIC), og vil som oftest befinne seg i en ruter. I ruterens vil talen bli komprimert og innpakket i IP-pakker, mens i ruterens hos mottageren vil talen tilsvarende bli dekomprimert og gjenskapt. Talen kan enten bli digitalisert i telefonsentralen eller i ruterens, avhengig av om sentralen er digital. Dette vil bli omtalt mer i kapittel 4.

⁴⁴ Den parten som mottar initiativet til samtalen.

⁴⁵ Forkortelse for "Private Automatic Branch Exchange", som er en privat telefonsentral, hussentral, koblet oppmot det offentlige telefonnettet. Blir ofte også betegnet PBX – "Private Branch Exchange".

⁴⁶ På engelsk: "billing purpose".

I en typisk "Voice over IP"-applikasjon i en bedrift, f.eks. mellom et hovedkontor i Oslo og et avdelingskontor i Trondheim, ville telefonsentralen i figur 2.3 utgjøres av en PABX. En bruker i Oslo vil slå f.eks. "9" for å få beslaglegge en VoIP-linje i PABX-en og deretter slås internummeret til kollegaen i Trondheim, som blir matchet med IP-adressen til VoIP-ruteren i Trondheim. Ruteren i Oslo vil rute VoIP-pakkene via lokalnettet til ruteren i Trondheim, som igjen oppretter en forbindelse med den lokale PABX-en. Telefonen hos mottageren vil ringe når en forbindelse mellom PABX-en og telefonen er opprettet.

Dette utgjør også selve kjernen i ideen bak VoIP, nemlig det å anvende allerede eksisterende og muligens avskrevet infrastruktur til ringe bortimot gratis mellom en bedrifts regionale kontorer. De kostnadene som eventuelt må påregnes kan være oppgradering av nettverket (større båndbredde) og nye rutere med VoIP-funksjonaliteter. I Norge har fjerntaksten blitt fjernet, men i andre land med fjerntakst vil VoIP også gi en ekstra gevinst. Dette i form av at man kan ringe via VoIP til et avdelingskontor i en annen landsdel nærmest gratis, og derfra ringe lokaltakst til f.eks. en kunde i den regionen. Dette blir i VoIP-terminologien betegnet som "toll-bypass".

2.3 Stillhetsundertrykkelse og "comfort noise"

Dagens telefoni-nettverk er basert på en toveis⁴⁷ 64 kbps kanal uansett om noen prater, eller ikke [DaP00, side 177]. Da det i normal talekommunikasjon vil være slik at den ene parten prater mens den andre lytter, vil om lag 50% av den beslaglagte båndbredden være bortkastet [DaP00, side 177]. Med IP-telefoni, derimot, åpner det seg muligheter for å unytte slik bortkastet båndbredde til noe mer fornuftig ved hjelp av såkalt *stillhetsundertrykkelse* [Agi00]. Stillhetsundertrykkelse innebærer kort sagt at man sparer båndbredde ved at stillhet i samtalen ikke blir samlet og derfor ikke overført. [ReR01] sier at stillhetsundertrykkelse kan gi en gjennomsnittlig båndbredde-besparelse på omtrent 40 %. Denne funksjonen baserer seg på "Voice Activity Detector" (VAD) som er en delkomponent i en "Digital Signaling Processor"⁴⁸ (DSP) [Per99] implementert i en IP-telefoni ruter eller -terminal, og opererer på sendersiden i en samtale [Agi00].

VAD fungerer på den måten at den detekterer størrelsen på talen i decibel (dB) og avgjør utfra det når den skal avbryte innpakningen av talerammer i IP-pakker [DaP00, side 177]. Når VAD detekterer et fall i amplituden på talesignalet, venter den en fast tid før den avbryter innpakningen. Denne tiden kalles "*hang over*"⁴⁹ (se figur 2.4) og ligger typisk på 200 ms [DaP00, side 177]. VAD-teknologien er belastet med problemer når det gjelder å bestemme når tale begynner og slutter, og å skille mellom tale og bakgrunnsstøy. Dette kalles "*signal-to-noise threshold*" [DaP00, side 177].

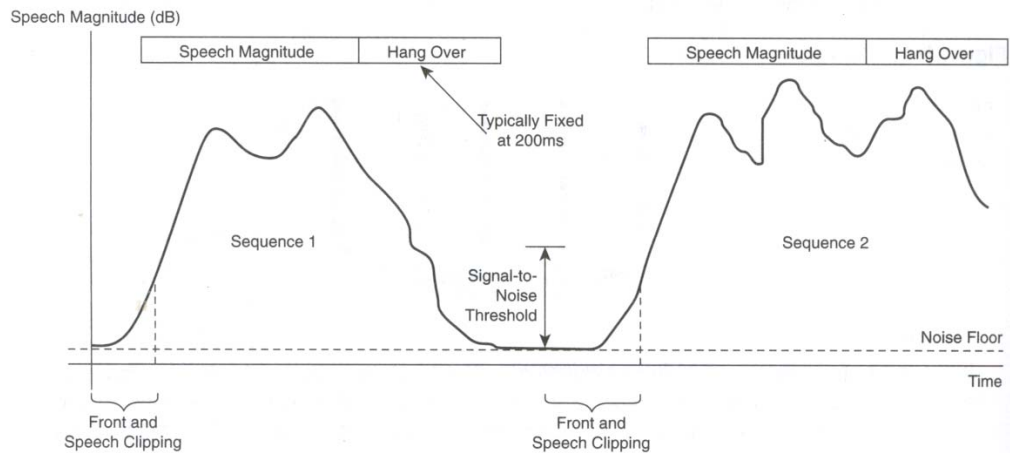
For samtaler i rom med mye støy vil VAD skru seg av selv. Problemet til VAD med å detektere når en samtale starter kan medføre at begynnelsen på setningen blir klippet vekk. Dette kalles "*front-end clipping*". For å optimalisere ytelsen i en gitt samtale, vil VAD kunne adaptivt konvergere mot en hensiktsmessig terskelverdi [Agi00]. Slik vil VAD kunne tilpasse seg varierende nivåer med signaler kontra støy. Om VAD ikke

⁴⁷ Bi-direksjonal.

⁴⁸ DSP vil bli omtalt mer i neste delkapittel

⁴⁹ Også benevnt "holdover time" i litteraturen.

fungerer slik den skal vil den minske forståeligheten på talen og forringe den totale konversasjonskvaliteten [Agi00]. For eksempel vil overdreven "front-end clipping" gjør det vanskelig å forstå hva som sies, mens overdreven "hangover" vil redusere nettverkseffektiviteten. For lite "hangover", derimot vil gjøre at talen føles oppstykket og usammenhengende.



Figur 2.4: VAD detekterer stillhet i talen, som klippes vekk for å spare båndbredde. Problemer med VAD kan være at første del av samtalen kan bli klippet vekk ("front-end clipping"), og å skille mellom støy og tale ("signal-to-noise threshold"). For å unngå å stoppe innpakningen for tidlig, venter VAD en fast tid før den kutter ("hang over"). Hentet ifra [DaP00a].

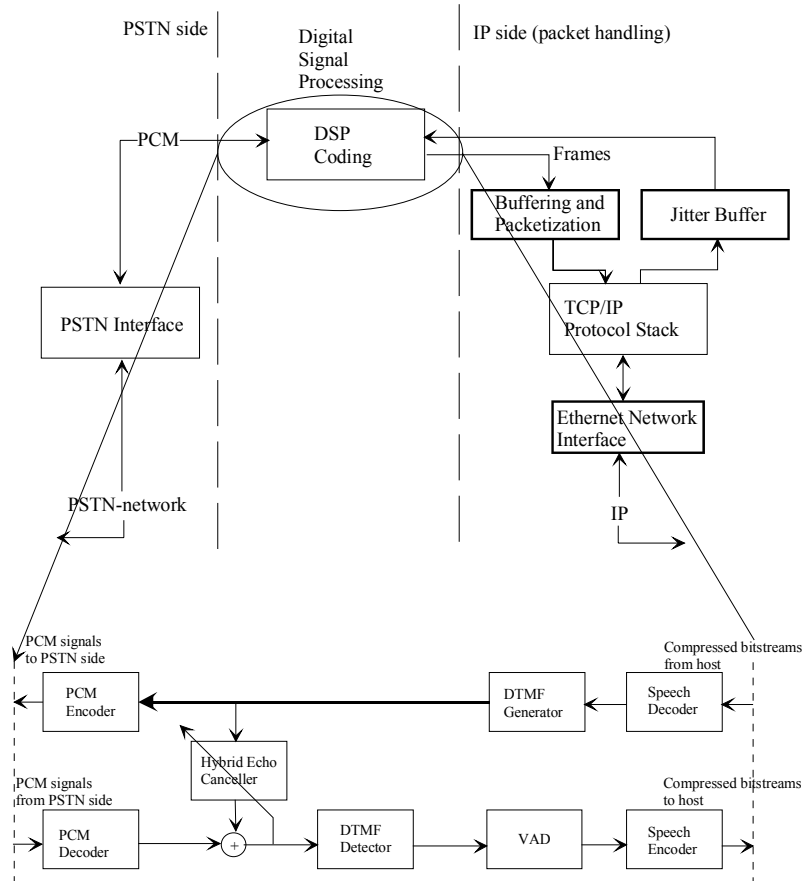
For å utfylle VAD sine egenskaper finnes det på mottagersiden en såkalt "comfort noise generator" (CNG). Den spiller av støy generert på mottagersiden til lytteren der hvor VAD, hos taleren, har kuttet forsendelsen av talepakker på grunn av stillhet i talen [Agi00]. Alternativet til dette, er å la det være fullstendig stille når taleren ikke prater, men dette kan være frustrerende for lytteren da linjen forvekslingsvis virker "død". Likheten mellom den virkelige bakgrunnsstøyen hos mottageren og den genererte, karakteriserer kvaliteten på CNG-en. [ReR01] trekker frem at finnes en potensiell ulempe med CNG, dersom det er en feiltilpasning i støyen når taleren prater og når vedkommende er stille. Dette kan virke forstyrrende og gi inntrykk av at kvaliteten er dårlig.

2.4 Prosessering

Slik som det er vist i figur 2.1 er det en del i faser i IP-telefoni som krever prosessering. I tillegg til sampling, komprimering av tale og innpakning i IP-pakker, vil det være behov for en del andre funksjoner for å få til "ekte" telefoni. Det være seg funksjoner som detektering/generering av DTMF⁵⁰-tone, stillhetsdetektering, generering av såkalt "comfort noise" og ekkoutligning [Per99]. Dette er prosesseringsfunksjoner som blir implementert i IP-telefoni ruterne eller gateway-en, i dedikerte prosessorer⁵¹ samt i tilhørende software.

⁵⁰ DTMF – Dual Tone Multi Frequency: Bruk av to samtidige toner for å slå et telefonnummer. Én tone for rad og én for kolonne. Slik utgjør de 12 tastene på en vanlig telefon en matrise [ToJ99].

⁵¹ Digital Signal Processor, DSP.



Figur 2.5: Tale fra PSTN blir prosessert i DSP-en før de blir forberedt på overføring i IP-pakker (buffering og innpakning) før de føres igjennom TCP/IP-stakken og ut på IP-nettet via ethernet-kortet. Andre veien, blir hodene i IP-pakkene ”strippet av” oppover i stakken før de holdes igjen i et jitterbuffer. Etter prosessering i DSP-en sendes talen via PSTN-nettet til mottageren. Prosesseringen i DSP-en involverer komprimering/dekomprimering, generering/detektering av DTMF-toner, stillhetsundertrykking og ”comfort noise”-generering (i ”VAD”). (Figuren er satt sammen av figurene 3 og 4 i [Per99].)

Nedre del av figur 2.5 viser de ulike prosesseringene som foregår i en IP-telefoni DSP i form av et høynivå skjematisk tegning. Slik som stadfestet tidligere i kapitlet, fungerer en IP-telefoni gateway som et slags bindeledd mellom PSTN- og IP-telefoni nettverket. Dette gjenspeiles også i figuren, hvor signaler fra PSTN-nettet av typen T1⁵² (markert som ”PCM signals from PSTN side” i nedre del av figuren) kommer inn på gateway-en igjennom et grensesnitt (i figur 2.5 benevnt ”PSTN Interface”) mot telefoni-nettverket før de blir prosessert i DSP-en.

I DSP-en blir signalene først utsatt for ekkoutligning før DTMF-tone blir detektert. I figuren inngår stillhetsundertrykkingen og ”comfort noise”-genereringen som ble beskrevet i delkapittel 2.4, i symbolet markert med ”VAD”. Talen blir videre komprimert i en codec, før talerammene blir buffret opp og innpakket i IP-pakker⁵³. IP-pakkene blir deretter ført igjennom TCP/IP-stakken før de blir ført inn på IP-nettet igjennom, i dette tilfellet, et ethernet-kort. IP-pakker inneholdende tale blir i motsatt retning ført igjennom ethernet-kortet, og hodene ”strippes av” opp igjennom stakken før talerammene blir holdt igjen i et jitterbuffer for å oppveie for at IP-pakkene kan

⁵² 1.544 Mbps igjennom det switchete telefon-nettverket og er mest brukt i USA [ToJ99].

⁵³ På engelsk: ”packetization”.

oppleve ulik forsinkelse igjennom IP-nettet. Deretter blir talen prosessert i DSP-en, hvor den blir utsatt for dekomprimering før DTMF-toner genereres. Til slutt føres talen ut igjennom nettverksgrensesnittet i PCM-format til mottageren på PSTN-nettet.

For grundigere informasjon om alle komponentene i en DSP og deres funksjoner henvises leseren til [ToJ99] og [IHM⁺99].

Kapittel 3

Talekvalitet

I dette kapitlet sees det nærmere på talekvalitet og hvilke faktorer som påvirker denne. Det skal også belyses hvordan disse ulike faktorene er med på å forandre den opplevde talekvaliteten og hvordan en samtale blir påvirket av dette. Videre skal det fokuseres på om det finnes noen metoder for å utbedre en "skadeskutt" tale.

3.1 Definisjoner av talekvalitet

For deltageren i en telefonsamtale vil talekvalitet være den karakteristikk han/hun vil sette på den opplevde kvaliteten under telefonsamtalen. Oppfatningen av hva god og dårlig talekvalitet er, er ytterst subjektivt og vil blant annet være avhengig av forventninger, kontekst og humør [Agi00]⁵⁴. Talekvalitetsbegrepet er såpass vidt at [WaS97] betegner det som et paraplybegrep med mange definisjoner.

Dette gjenspeiles også i litteraturen hvor begrepet *talekvalitet* defineres noe ulikt. I [MaL97] defineres kvaliteten på en talekanal som den tiden kanalen er fri for forstyrrelser, forutsatt at denne tiden er lang nok til å overføre segmenter med aktiv tale. I andre artikler inngår dette begrepet i det noe videre begrepet *tjenestekvalitet*⁵⁵. Tjenestekvalitet kan defineres som mangt, det kommer helt an på i hvilket perspektiv man setter det i. For eksempel, er vi interessert i kvaliteten på forbindelsen (man kjører IP-telefoni over) med fokus på blant annet prosentandel tapte pakker eller ønsker vi å karakterisere den opplevde talekvaliteten i IP-telefoni applikasjonen? [ToL00] referer til "International Telegraph and Telephone Consultative Committee" (CCITT)⁵⁶ sin "Recommendation E.800" som sier at tjenestekvalitet kan for brukere også være i hvilken grad de er fornøyd med tjenestens samlede ytelse. [EyS00] sier at brukernes syn på tjenestekvalitet består av to komponenter. Det ene er pålitelighet og tiden det tar å sette opp et kall (definert i [EyS00] som tiden fra siste tall i nummeret er slått til man hører ringing), og det andre er kvaliteten på talen.

⁵⁴ Ikke trykket materiale. Funnet på <http://onenetworks.comms.agilent.com/WhitePapers.asp#IP>

⁵⁵ På engelsk: "Quality of Service" (QoS).

⁵⁶ Heter i dag "International Telecommunication Union", ITU.

[VKBG95] definerer tjenestekvalitet til å være det sett av kvalitative og kvantitative karakteristikk på et distribuert system som er nødvendige for å oppnå funksjonalitetene en applikasjon krever. [Agi00] definerer talekvaliteten som de kvalitative og kvantitative målingene som karakteriserer lyd- og konversasjonskvaliteten på en telefonsamtale. Slik sett stemmer definisjonene til [Agi00] godt overens med den i [VKBG95], men satt i en IP-telefoni kontekst.

I denne rapporten vil begrepet talekvalitet brukes om den lydmessige- og konversasjonsmessige kvaliteten som oppleves i en samtale.

3.2 Talekvalitet er igjen et tema

Internett eller andre IP-baserte nettverk er pakkeswitchete, og er ikke designet for å frakte tidskritisk sanntids⁵⁷ trafikk. Derfor er det en del utfordringer og tekniske problemer som må løses før man kan implementere IP-telefoni med suksess [HNA00]. Blant disse utfordringene kan det nevnes bygeaktig⁵⁸ trafikk og manglende følsomhet for forsinkelser og variasjon i forsinkelsene. Det vanlige telefonsystemet, PSTN, er basert på linjeswitching og tilbyr en garantert tjenestekvalitet til kundene. Dette systemet fungerer ved at en linje settes opp mellom to endepunkter før selve kommunikasjonen begynner.

PSTN-miljøet har for lengst satt talekvalitet på dagsordenen og har gradvis utviklet den til å oppfylle kravene til jitter, lav forsinkelse og fast båndbredde som stilles av tidskritiske taleapplikasjoner. Selvom ikke PSTN tilbyr perfekt kvalitet, har folk blitt vant til denne og PSTN-telefoni betegnes som svært pålitelig. Derfor brukes også PSTN som referansepunkt når talekvaliteten i IP-telefoni/Voice over IP-applikasjoner skal bedømmes [Agi00]. Da disse applikasjonene er basert på tidskritisk levering av pakker i en nettverkstruktur som absolutt ikke er beregnet for dette, kreves det en del viktige mekanismer for at talen i slike applikasjoner skal kunne være sammenlignbare med PSTN. [Agi00] understreker at nettopp det å tilby talekvalitet på nivå med PSTN, er svært viktig for IP-telefoni/Voice over IP for å vinne allmenn aksept. Derfor har også talekvalitet igjen blitt et tema.

3.3 Faktorer som påvirker talekvaliteten

Dette delkapitlet skal se nærmere på hvilke faktorer som påvirker talekvaliteten og om det finnes noen muligheter for å utbedre virkningene av disse faktorene. Hovedvekt vil bli lagt på pakketap, forsinkelser, ekko og jitter.

3.3.1 Pakketap

Siden hver IP-pakke inneholder typisk 40-80 millisekunder (ms) med taleinformasjon⁵⁹, vil pakketap kunne gi stor skade på talekvaliteten [HNA00]. Dette

⁵⁷ På engelsk: "realtime".

⁵⁸ På engelsk: "bursty".

⁵⁹ Noen kilder oppgir 20 ms og oppover. Hvor mye tale en IP-pakke inneholder, er opptil DSP-leverandørene [DaP00, side 168] og kan derfor variere.

kan også være et spørsmål om innstilling av codec⁶⁰ i en IP-telefoni klient. En tapt pakke medfører at deler av taleinformasjonen tapes. Selvom mennesker er i stand til å logisk rekonstruere talen, vil for store pakketap gjøre den talen mottageren hører nærmest meningsløs. Tjenestekvalitet i multimediasesjoner kan derfor bestemmes utfra pakketap og -forsinkelser [JiS99]. Uformelle tester i [KBS98] viste at tilfeldig og uavhengig pakketap på inntil 10 prosent vil ha lite merkbar effekt på tale ved bruk av G.723.1 som codec. [BorS98] sier at kunder er villig til å tolerere en svakt klippet og stagnert tale grunnet tap av en til fire pakker med G.723.1-kodet tale. Avbrudd på flere sekunder vil derimot ikke bli akseptert, og [BorS98] sammenligner dette med at vanlig telefoni plutselig blir brutt. Målingene i [BorS98] viser at det er asymmetri i tapskarakteristikkene i hver ende av en IP-telefoni forbindelse, som kan oppleve radikalt forskjellig tjenestekvalitet. Derfor sier [BorS98] at den totale tjenestekvaliteten må være lik den dårligste av de to unidireksjonelle-retningene, da den personen med best kvalitet kan stadig bli bedt om å gjenta seg selv.

[HNA00] trekker frem en del metoder for å reparere talen:

➤ *"Network upgrade"*

En preventiv metode for å forhindre pakketap i IP-rutere vil være å oppgradere infrastrukturen i IP-nettverket, da pakketap internt i en IP-ruter vil være en direkte årsak av utilstrekkelig link-kapasitet og/eller pakkeprosesseringshastighet.

➤ *"Silence Substitution"*

Når en pakke tapes, kan ikke taleinnholdet i den pakken avspilles. En metode for å opprettholde en sammenhengende samtale, kan da være å erstatte de tapte taleinformasjonene med stillhet. En ulempe med denne metoden er at den medfører "voice clipping", det vil si at enkelte stavelser i talen blir borte, og forringer talekvaliteten betraktelig. Dette gjelder spesielt for store pakker og høye tapsrater. [HHSW95] henviser til studier som har vist at denne metoden kun gir tilstrekkelig god talekvalitet for pakker inneholdende taleinformasjon kortere enn 16 ms ved tapsrate opptil 1 prosent.

➤ *"Noise Substitution"*

En metode som har vist seg å være bedre enn overnevnte er å erstatte tapt taleinformasjon med såkalt "white noise". Dette er definert som "støy med et kontinuerlig og uniformt frekvensspektrum over et spesifisert frekvensbånd. I hele dette frekvensbåndet har støyen lik effekt per hertz"⁶¹. Denne metoden baserer seg på menneskets evne til å selv reparere talen, ved at man skjønner hva neste uttalelse skal være når det kommer støy⁶². Dette skal ifølge [HHSW95] og [PHH98] ikke være mulig om denne støyen blir byttet ut med stillhet.

➤ *"Packet Repetition"*

Avspilling av siste korrekt mottatte pakke kan være en metode for å reparere skadene forårsaket av en tapt pakke. Denne metoden vil ikke klare å reparere skadene som følge av tap av mange påfølgende pakker. En tommelfingerregel sier at G.729 kan tåle et gjennomsnittlig pakketap på 5 prosent for en hel samtale (uten

⁶⁰ Codec-er vil bli nærmere omtalt i kapittel 4

⁶¹ American National Standard for Telecommunications: "Telecom Glossary 2000".

Se URL: <http://www.its.bldrdoc.gov/projects/t1glossary2000/>

⁶² Dette fenomenet kalles "phonemic restoration".

å forringe kvaliteten), fordi den implementerer "packet repetition" [DaP00, side 177]. [DaP00, side 177] opererer med 20 ms tale i hver IP-pakke og sier at med "packet repetition" vil ikke lytteren merke noen forskjell.

➤ *"Packet Interpolation"*

Denne metoden benytter karakteristikken til pakkene enten foran eller bak den tapte pakken for å produsere en erstatning for den tapte pakken. Slik sørges det for at erstatningen følger forandringer i karakteristikken til talestrømmen. Studier har vist at denne metoden gir bedre talekvalitet enn "silence substitution"- og "packet repetition"-metodene [HNA00]. Disse metodene kan enten fokusere på tonehøyde⁶³ eller tidskalaen av et talesignal.

➤ *"Frame Interleaving"*

Metoden innebærer å omstokke den opprinnelige rekkefølgen på de logiske enhetene som talen prosesseres i, såkalte *talerammer*⁶⁴, og spre dem utover flere IP-pakker før transmisjon, slik at påfølgende rammer ikke sendes etter hverandre. Hos mottageren vil disse logiske enhetene bli omstokket igjen tilbake til den opprinnelige rekkefølgen. Ved pakketap medfører dette kun til små glipp i talen spredd utover i tid, istedenfor store glipp som ville vært tilfelle om de logiske enhetene hadde fulgt etter hverandre i den opprinnelige rekkefølgen. På måten ville pakketap forårsake mindre skade på talen, enn om større mengder sammenhengende tale ble tapt. En ulempe er at metoden medfører økt forsinkelse, men skjer dette innenfor forsinkelsesbudsjettet, det vil si den mengden forsinkelse talen kan tåle ende-til-ende, vil dette være en attraktiv metode siden den ikke introduserer ekstra "overhead" i nettet.

➤ *"Forward Error Correction" (FEC)*

Taleinnholdet i en pakke sendes redundant med påfølgende pakker, slik at når pakken eventuelt tapes kan innholdet rekonstrueres fra de påfølgende pakkene. [PCM00] presenterer en FEC-algoritme, kalt "*USF Algorithm*".

3.3.2 Pakkeforsinkelser

Begrepet *forsinkelse*⁶⁵ er tidsforsinkelsen fra et ord uttales til det høres hos mottageren, og måles som regel i millisekunder [Per99]. Denne forsinkelsen kalles også munn-til-øre forsinkelse eller enveisforsinkelse. Effekten av forsinkelse på talen er at en del små uttalelser i en samtale, som bekrefter for taleren at mottageren lytter, blir såpass forsinket at det oppstår forvirring. Forvirringen resulterer i at taleren vil ta pauser for å få disse små tilbakemeldingene fra mottageren, til slutt kan det ende med at flyten i samtalen stadig blir avbrutt. Det som da skjer er at tilstrekkelig forsinkelse blir lagt til mellom to stavelser, slik at mottageren føler at rytmen i talen blir borte. Denne for lange forsinkelsen innvirker på talen på den måten at deltakerne venter på hverandre for å forsikre seg om at den andre er ferdig. Om pausene kommer på riktig så ugunstige tidspunkt, kan det i verste fall medføre at de begynner å snakke i munn på hverandre. For stor forsinkelse kan også medføre at avsenderen hører ekko (mer om ekko i kapittel 3.3.4), da signalet vedkommende sendte avgårde blir reflektert tilbake. Kommer pakkene tilstrekkelig for sent, slik at det ikke er mulig på

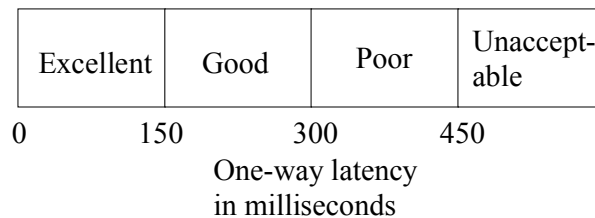
⁶³ På engelsk: "pitch".

⁶⁴ I delkapittel 3.3.2 blir disse forklart litt nærmere.

⁶⁵ På engelsk "Latency". Den norske betegnelsen vil bli brukt i det følgende.

mottagersiden å rekonstruere talen, vil dette gi samme effekt på talen som om pakken ble tapt [Agi00]. Graden av forsinkelse som talen utsatt for, vil være avhengig av de ulike forsinkelsene som kan oppstå i nettet når en talepakke sendes fra originerende side til mottageren.

På mottagersiden holdes den første pakken igjen i et såkalt ”*playout buffer*” eller ”*jitterbuffer*”⁶⁶ slik at man oppnår at flest mulig av talepakkene ankommer før avspilling. På denne måten sørges det for at talerammene i en IP-pakke har ankommet før siste taleramme i den foregående IP-pakken har blitt spilt av. Ankommer pakker etter dette, høres det glipp i talen. Denne holdetiden må være såpass lang at pakker rekker å ankomme før fristen (ellers blir de betegnet som tapte og forkastes), samtidig må holdetiden være kort nok til at konversasjonen holdes sammenhengende [MaL97].



Figur 3.1: Kvaliteten på talen med enveisforsinkelse opptil 150 ms kategoriseres som ”Excellent”, mens 150 til 300 ms betegnes som ”Good”. Fra 300 til 450 ms betegnes talen som ”Poor”, mens enveisforsinkelse over 450 gir en uakseptabel kvalitet. (Basert på [Per99].)

Det er vanskelig å bedømme hvor mye forsinkelse som godtas av brukerne, da de ofte vil avveie forringelsen av talen med det utbytte de har av det aktuelle kommunikasjonssystemet (her: IP-telefoni). Et eksempel er mobiltelefoni hvor brukerne godtar dårligere talekvalitet enn i PSTN. Brukerne prioriterer her fordelene ved å kunne motta samtaler selv når man er mobil høyere enn talekvalitet.

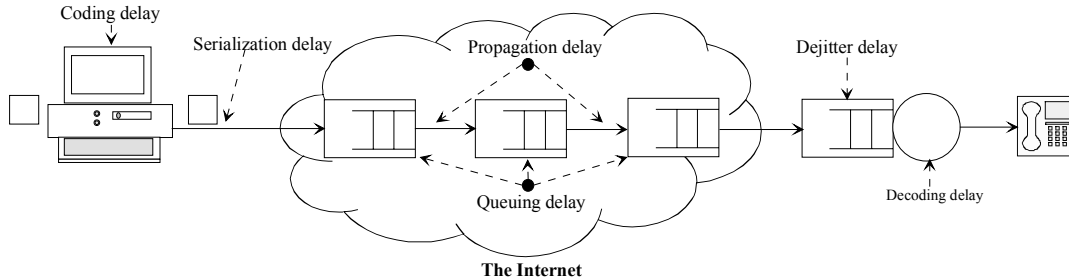
Det oppgis varierende tall i litteraturen for hva som er akseptable enveisforsinkelser. [HNA00] sier at forsinkelser lavere enn 150 ms betegnes generelt som akseptabelt for de fleste applikasjoner, men over dette vil ekko kunne oppleves og deltakerene kan begynne å snakke i munn på hverandre. [KBS98] opererer med 100 ms forsinkelse for å oppnå ”toll quality” og at forsinkelser opptil 150 ms er ”good” og at tale med forsinkelse over 150 ms som ”potentially good”. [Per99] opererer med 150 ms som grense for å oppnå ”excellent” og øvre grense for ”good” ligger på 300 ms (se figur 3.1). [Per99] anbefaler videre at som tommelfingerregel bør enveisforsinkelsen holdes under 200 ms, ellers vil brukeren heller velge vanlig telefoni istedenfor å ringe via IP-telefoni.

Psykologiske studier av dynamikken i konversasjoner har konkludert med at dersom enveisforsinkelsen av talen er større enn 300 ms vil konversasjonene arte seg som halv-dupleks eller ”push-to-talk” [Cox97], slik som i luftfartskommunikasjon hvor den ene snakker ferdig før motparten svarer. Samtidig fant man i disse studiene også at *forventningene* til taleforsinkelsene er av avgjørende betydning. Er man f.eks. psykologisk innstilt på at en lydstrøm⁶⁷ avspilles fra Internett, vil ikke en forsinkelse på 300 ms ha noen innvirkning på brukeren da vedkommende faktisk forventer å måtte vente i noen sekunder. Tilsvarende kan forsinkelser for langdistanse samtaler på

⁶⁶ Også kalt ”dejitter-buffer” i litteraturen. Heretter vil begrepet ”jitterbuffer” brukes.

⁶⁷ På engelsk: ”stream”. Det å avspille en lydstrøm betegnes følgelig som ”streaming”.

mellom 150 og 400 ms være akseptabelt, da deltagerne i samtalen vil være mentalt forberedt på store forsinkelser. Forsinkelser over 400 ms forringer talekvaliteten såpass at det normalt ikke godkjennes, uansett sammenheng. Slike pakker har i praksis samme innvirkning på talen som om de var tapt.



Figur 3.2: De ulike typer forsinkelser som kan oppstå i IP-telefoni. (Basert på [HNA00].)

En av de største tekniske utfordringene i IP-telefoni er forsinkelsene som oppstår under frakten av talepakkene igjennom IP-nettverket. Noen av disse har noenlunde faste verdier og kan kalkuleres med på forhånd, mens andre er høyst variable. De forsinkelser som kan oppstå under transmisjonen er blant annet (se figur 3.2):

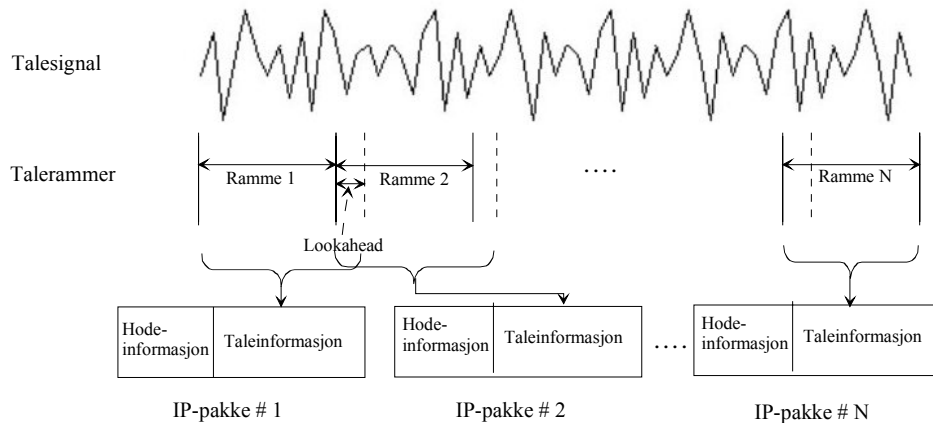
1. Codec-forsinkelse⁶⁸ oppstår i prosessen med å konvertere analog tale til digital og utføre komprimering på den digitaliserte talen for å spare båndbredde. Jo større komprimering desto større total forsinkelse ende-til-ende [Agi00], slik blir komprimeringsgraden en avveining mellom forsinkelse og båndbreddekrav. I tillegg økes prosesseringskostnader⁶⁹ med høyere komprimering. To viktige delkomponenter som inngår i codec-forsinkelse er:
 - "Frame"⁷⁰ processing delay": Forsinkelsen som oppstår i prosesseringen av en enkelt taleramme. Også kalt "processing delay".
 - "Lookahead delay": Forsinkelsen som oppstår i prosesseringen av deler av den påfølgende talerammen for å utnytte likheter i to talerammer. Grunnet at codec-en er avhengig av kunnskap om sample n+1 for å korrekt kunne gjenskapes sample n hos mottager, må det komprimerte samplet n forlenges med litt data om sample n+1 [Cis02], denne datamengden kalles "lookahead". Dette er nødvendig fordi codec-en må vite noe om hvordan lydbølgen for talen i taleramme n+1 ser ut for å kunne skjøte sammen talen i taleramme n med talen i n+1 på en korrekt måte (se figur 3.3).

Andre forsinkelser som inngår i codec-forsinkelsen vil bli nevnt i kapittel 4.2.2.

⁶⁸ Benevnt "coding delay" i figur 3.2.

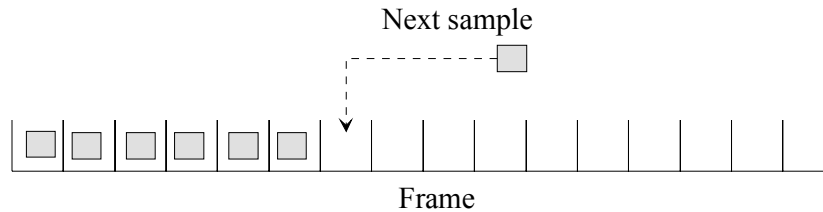
⁶⁹ På engelsk: "computational".

⁷⁰ På norsk: "taleramme". Heretter vil det norske ordet brukes.



Figur 3.3: Codec-en trenger kunnskap om hvordan lydbølgen ser ut i ramme 2 for å kunne gjenskape talen i ramme 1 hos mottager, derfor blir en liten del av ramme 2 bli inkludert i IP-pakke nummer 1. Denne prosesseringsforsinkelsen kalles "lookahead delay".

Ordet "taleramme" er i [BrK99] definert som den logiske enheten som brukes til å prosessere taledataene. Hver taleramme består av et antall råsampler fra den analoge lydbølgen (se figur 3.4). Hvor stor en taleramme er, altså hvor lang tale i millisekunder (ms) hver taleramme består av, er definert i hver codec. Dette er gitt som "frame size" i spesifikasjonene. Når det typisk leveres sampler hvert 0,125 ms fra PCM til den aktuelle codec-en, kan man regne ut hvor mange sampler en ramme består av når "frame size" er kjent. Hvor mange slike rammer som sendes med som last⁷¹ i en IP-pakke er også definert i den enkelte codec-implementasjon.



Figur 3.4: En taleramme ("frame") består av et antall råsampler fra det opprinnelig analoge talesignalet. Antallet sampler som inngår i en taleramme er spesifisert i hver enkelte codec. (Basert på [BrK99].)

2. På avsendersiden oppstår det i tillegg til codec-forsinkelsen en innpakkingsforsinkelse⁷², som er tiden det tar å "fylle opp" en IP-pakke med taleinformasjon [Agi00]. Innpakkingsforsinkelsen, p , kan matematisk uttrykkes som $p=s/r$, hvor s er størrelsen på talepakken⁷³ i bytes og r er bitraten til codec'en i bit/sekund [MMG98]. Denne bitraten angir hvor fort codec'en jobber og "mater" ferdig prosesserte talerammer for innpakking.

⁷¹ På engelsk: "payload".

⁷² På engelsk: "packetization delay".

⁷³ Dette ordet vil brukes om IP-pakker inneholdende tale.

IP	UDP	RTP	Payload
20 bytes	8	12 byte	20 til 160

Figur 3.5: For å frakte talesampler (i IP-pakkens "payload"-del) igjennom et IP-basert nettverk er IP-, UDP- og RTP-hode nødvendig [WEL01]⁷⁴. Disse hodene er hhv. 20, 8 og 12 byte lange, hvilket utgjør en total hodestørrelse på 40 byte. Lasten ("payload") utgjør 20 til 160 byte [DaP00, side 193].

Lengden på talepakkene blir en avveining mellom følgende: Jo lengre talepakkene er, desto større forsinkelse. Men, samtidig, jo kortere talepakkene er, desto mer redusert nettverkseffektivitet oppnås, da flere pakker må sendes med tilnærmet lik og dermed redundant hodeinformasjon. Dessuten vil disse pakkene sendes med stor overhead. Dette fordi IP-hodet har en fast lengde, 40 byte (se figur 3.5), uansett mengden tale som fraktes i hver pakke.

Ved å benytte "RTP⁷⁵ Header Compression" (cRTP) på hodene vist i figur 3.5 vil de kunne komprimeres ned til 2-4 bytes avhengig om UDP⁷⁶-sjekksnummer benyttes [DaP00, side 192-194]. Den beslaglagte båndbredden for en samtale basert på codec-en G.729 vil da ifølge [DaP00, side 192-194] kunne reduseres fra 24 kbps til 11.2 kbps. For en linje med liten båndbredde, vil dette ha stor betydning. På en 54 kbps linje vil det være mulig å føre fire slike samtaler samtidig. For mer om hodekomprimering, se IETF "Request for Comment (RFC) 2508" eller [DaP00, side 192-194].

Dekodingsforsinkelsen (inkludert dekomprimering) hos mottager kan som tommelfinger-regel sies å være halvparten av kodingsforsinkelsene hos avsender [KBS98].

3. Serialiseringsforsinkelsen er den tiden det tar å få pakkene ut på transmisjonslinjen og bestemmes utfra hastigheten på den. Ved høyere hastigheter på linjen kan denne forsinkelsen reduseres kraftig. Denne forsinkelsen er også avhengig av rammestørrelsen (det vil si mengden taleinformasjon i rammen) som brukes av codec-en. Desto lengre rammene er, jo større forsinkelser oppstår i overføringen av pakken. [DaP00, side 168] nevner bare såvidt serialiseringsforsinkelser, med den begrunnelsen at dens innvirkning på den totale forsinkelsen antas å være så minimal at den kan neglisjeres.
4. Køforsinkelse oppstår i rutere og gateway-er hvor talepakker blir ventende bak andre pakker som igjen venter på å bli sendt ut på en bestemt utgående link. I tillegg kommer det en viss switching/rutingsforsinkelse i forbindelse med å analysere pakkehodet, slå opp i rutings Tabellen og rute pakkene til riktig utport [Agi00]. Denne forsinkelsen vil være avhengig av arkitekturen på ruter og størrelsen på rutings Tabellen. Køforsinkelse bør helst holdes lavere enn 10 ms ved å gjøre bruk av køingsmetoder som er optimale for det aktuelle nettverket [DaP00, side 169].

⁷⁴ Ikke trykket materiale. Hentet fra <http://www.erlang.com/bandwidth.html>.

⁷⁵ Real-Time Protocol. RTP vil bli sett mer på i appendiks A "Signaleringsprotokoller".

⁷⁶ User Datagram Protocol.

[HNA00] nevner videre at det i IETF har blitt jobbet med å bruke "*Differentiated Services*" (DiffServ)⁷⁷ og "*Resource Reservation Protocol*" (RSVP)⁷⁸ til å prioritere taletrafikk foran annen type trafikk. I tillegg kan mekanismer slik som "*Weighted Fair Queuing*" (WFQ) for prioritering, "*Multiprotocol Label Switching*" (MPLS)⁷⁹ for flytkontroll og bruk av "*Type-of-Service*" (ToS)-feltet i IP-hodet benyttes [Agi00]. Ved å sette "*IP precedence*"-bitet i ToS-feltet kan man sette at en type trafikk skal ha høyere prioritet enn annen type trafikk [ToJ99]⁸⁰. ToS-feltet gir tre valg for hvor viktig en type trafikk er: "*lav forsinkelse*", "*stor gjennomstrømning av data*"⁸¹ og "*høy pålitelighet*"⁸². Videre vil egenskapene til "*IP versjon 6*" (ipv6) (IETF RFC 1883)⁸³ i forhold til sanntids prioritering av trafikk kunne forbedre tjenestekvaliteten til IP-telefoni betraktelig. Av avgrensings hensyn vil det ikke bli gått grundigere inn på de ulike nevnte mekanismene for køprioritering og ressursreservering. Leseren kan konsultere [GüG01] som tar for seg de fleste aspektene rundt tjenestekvalitet i forbindelse med IP-telefoni. RSVP blir forklart på en enkel og lettfattat måte i [ToJ99]. Videre går [Fin02] inn på RSVP, MPLS, DiffServ og prioritering på en oversiktlig måte.

- 5 Forplantningsforsinkelse⁸⁴ er den forsinkelsen som trengs for at et signal skal gå fra et punkt til et annet i et fysisk medium. Denne forsinkelsen kan variere for ulike transmisjonsmedium og kan ha mye å si når avstandene blir store.

3.3.3 Jitter.

Jitter er en kanskje større trussel mot talekvaliteten i IP-telefoni enn de overnevnte forsinkelser [HNA00]. Nettverks jitter er de innbyrdes forskjellige forsinkelsene til ankomne pakker som oppstår som følge av forskjellige køings- og forplantningsforsinkelser i nettet. Dessuten kan pakker tilhørende en pakkestrøm ta ulike veier igjennom nettet og derigjennom oppleve ulike forsinkelser. For å tillate varierende ankomsttider og samtidig opprettholde en konstant strøm av pakker, brukes jitterbuffer hvor pakkene holdes igjen før de avspilles, den ekstra forsinkelsen dette påfører en pakke kommer i tillegg til den samlede forsinkelsen pakken opplever igjennom nettverket. Denne holdetiden indikerer også bufferets størrelse og gjør den totale forsinkelsen til en pakke tilsvarende lengre.

Utfordringen ligger i å finne et optimalt buffer som balanserer fjerningen av jitter og samtidig minimerer den ekstra forsinkelsen. Er bufferet for kort tapes pakker, er det for stort blir forsinkelsene ekstra store. Typisk vil bufferstørrelsene variere mellom 50 ms

⁷⁷ RFC-ene i tilknytning til DiffServ er listet opp her: <http://www.ietf.org/>.

⁷⁸ RFC-ene i tilknytning til RSVP er listet opp her: <http://www.ietf.org/html.charters/rsvp-charter.html>.

⁷⁹ RFC-ene i tilknytning til MPLS er listet opp her: <http://www.ietf.org/html.charters/mpls-charter.html>.

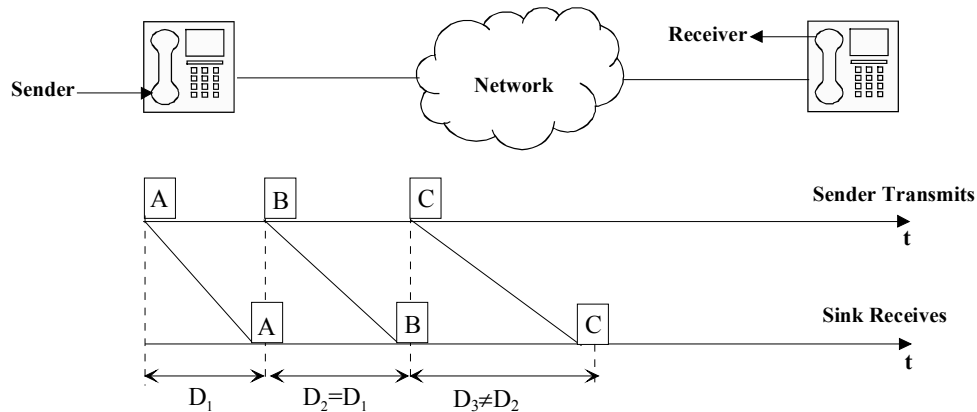
⁸⁰ Side 59 i appendiks A.

⁸¹ På engelsk: "high throughput".

⁸² Mer detaljer om ToS-feltet kan leses i [ToJ99].

⁸³ <http://www.ietf.org/html.charters/ipv6-charter.html>. Finnes også linker til andre RFC-er i tilknytning til ipv6.

⁸⁴ På engelsk: "propagation delay".



Figur 3.6: Pakker ankommer mottageren med innbyrdes ulik forsinkelse. I figuren er forsinkelsen D_3 ulik forsinkelsene $D_2=D_1$. (Basert på [DaP00, side 168].)

til 100 ms, ideelt bør bufferstørrelsen dynamisk kunne varieres utfra nettverksforholdene [HNA00]. Denne kan således bestemmes utfra forholdet mellom antall forsinkede pakker og antall presis ankomne pakker. Den optimale intielle holdetiden⁸⁵ i et jitter-buffer vil være summen av alle de variable forsinkelsene mellom endepunktene [Cis02]. For å unngå "overflow" i bufferet opererer videre [Cis02] med en maksimal bufferstørrelse på normalt halvannen til to ganger den intielle holdetiden. Påfølgende pakker kan da ikke bli mer forsinket enn verdien på den intielle holdetiden om glipp i talen skal unngås. Om pakke nummer to forsinkes mer enn den første, avspilles den første og bufferet tømmes. Dette vil resultere i at pakkenummer to holdes igjen like lenge som den aller første pakken ble for å "resette" bufferet [Cis02]. Dette vil resultere i en glipp i talen med like lang varighet som holdetiden for pakkenummer 1.

I tillegg til jitterbuffere vil bruk av diverse prioriteringsmekanismer (se punkt 4 ovenfor) for prioritering av taletrafikk kunne være med på å redusere jitter [Agi00]. Med jitterbuffere kan det ofte være nødvendig med et tidsstempel for å forsikre at talepakkene blir avspilt for lytteren med samme "timing" relativt til andre taleelementer i den opprinnelige talen, og "Real Time Protocol" (RTP) kan brukes til dette [Cis01c].

3.3.4 Ekko

Ekko er et fenomen som gjør at talesignalene reflekteres tilbake, slik at når man prater i telefonen kan man høre sin egne uttalelser litt forsinket. Hvorvidt man vil legge merke til de reflekterte ordene er avhengig av hvor forsinket de er. I vanlig telefoni, PSTN, er som regel ikke denne forsinkelsen lang nok til at man legger merke til ekkoet. Men på grunn av de nokså store forsinkelsene som kan oppstå i IP-telefoni, slik som beskrevet i delkapittel 3.3.2, vil man ofte kunne legge merke til ekkoene [Agi00]. [Agi00] slår videre fast at om tiden fra ord uttales til de reflekteres tilbake til taleren er kortere enn 25-30 ms eller om volumet er lavt (cirka -25 desibel (dB)), vil ikke ekko være til forstyrrelse for samtalepartnerne. Dette bekreftes også i [ReR01]. [GiK01] sier at om brukerne vil klage på ekkoet, er avhengig av ekkoattenueringen⁸⁶

⁸⁵ [Cis02] definerer dette til å være holdingen av den første pakken i en gitt periode.

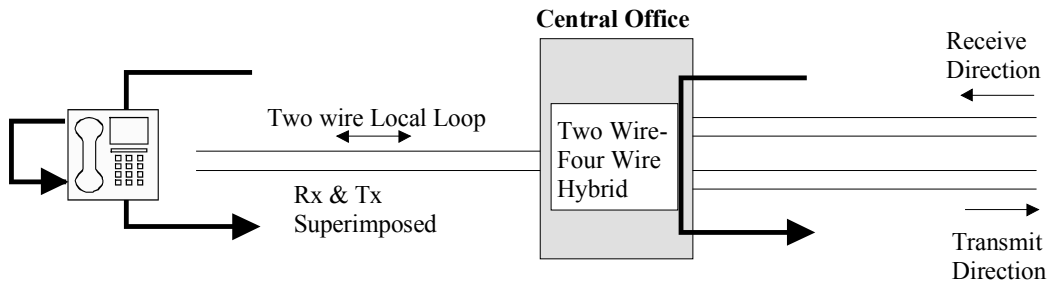
⁸⁶ Avtagning av effekt.

som funksjon av forsinkelsen. Om forsinkelsen er større enn 250 ms, må den relative attenueringen være på minst 46 dB.

Selvom signalene har avtatt en del i signalstyrke, blir som regel de ekkoene man hører reflektert på mottagersiden. Disse ekkoene vil ha en forsinkelse to ganger ende-til-ende forsinkelsen, altså lik tur-retur forsinkelsen⁸⁷.

Man ønsker allikevel et visst ekko i en samtale med godkjent talekvalitet, nemlig det man på engelsk kaller "side tone". Dette er et ekko med en forsinkelse på omtrentlig 28 ms [Agi00], som gjør at man opplever å høre sin egen stemme samtidig som man prater. Dette forsikrer den som prater om at linjen ikke er "død".

Ekko oppstår som følge av en elektrisk mistilpasning mellom den analoge telefonen og transmisjonsmediet i en del av nettverket som kalles "*tail circuit*" [Agi00]. Denne delen inkluderer alle enheter tilknyttet PSTN-siden av IP-telefoni gateway-en. Mistilpasningen skjer i konverteringen, i lokalsentralen⁸⁸, fra fire tråder til to tråder som går ut til mottagerens telefon, eller mellom en digital overføringskanal og to tråder i lokalsentralen [Agi00]. Denne konverteringen gjøres i en enhet, kalt *hybrid* (se figur 3.7), som separerer signalene i sende- og mottaretningene fordi de skal fraktes på to ulike trådpar eller ulike overføringskanaler. Metodene som brukes for å utføre disse separeringene er ikke ideelle, og noen av signalene i motta-retningen "smitter" eller "lekker" over på senderetningen og oppleves som ekko.



Figur 3.7: Ekko oppstår i overgangen fra to til fire tråder i en såkalt "hybrid" i PSTN-nettet. Taleren reflekteres tilbake til taleren som hører sin egen stemme "lenge" etter at vedkommende uttalte ordene. (Basert på DaP00a)

Ekko kan også oppstå på grunn av akustiske koblingsproblemer mellom telefonens høytaler og mikrofon. Slik type ekko kalles gjerne *akustisk ekko*. Akustisk ekko eller "feedback" er vist i figur 3.8 i en telefon-til-PC arkitektur. Dette oppstår som følge av at mikrofon og høytaler på PC-siden står for nærme hverandre, slik at talen fra telefon-siden oppfanges av mikrofonen og sendes som ekko tilbake til telefon-siden [KBS98]. Mikrofonen kan også fange opp lyd fra høytaleren som har blitt reflektert fra tak og vegger⁸⁹, dette er mye vanskeligere for en ekkoutligner⁹⁰ å rette opp enn "vanlig" ekko.

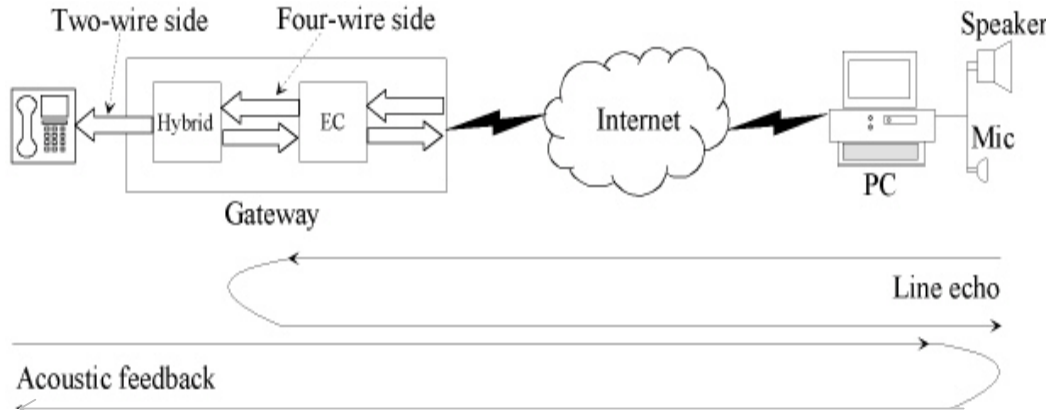
Man kan fjerne ekko ved bruk av *ekkoutlignere* plassert nærmest mulig den "tail circuit" som forårsaker ekkoene, enten i den lokale sentralen, VoIP gateway-en eller i VoIP-baserte PC-terminaler [Agi00]. Ekkoutlignere fungerer på den måten at de

⁸⁷ På engelsk: "roundtrip delay".

⁸⁸ På engelsk: "Foreign Exchange Office" (FXO) eller "Central Office".

⁸⁹ "Multi-path echo."

⁹⁰ På engelsk: "echo canceller".



Figur 3.8: Linjeekko og akustisk ekko i en telefon-til-PC arkitektur. Talesignalene fra PC-siden blir reflektert tilbake til avsenderen i hybriddelen i telefonsentralen. Ekkoutlignereren ("EC") trekker ifra ekkoene på fire-tråds siden. Akustisk ekko oppstår som følge av at mikrofon og høytaler på PC-siden står for nærme hverandre, slik at talen fra telefon-siden oppfanges av mikrofonen og sendes som ekko tilbake til telefon-siden. Basert på [KBS98].

danner en matematisk modell av den "tail circuit" den overvåker. Denne modellen, sammen med talesignalene som forventes å bli reflektert, brukes for å danne et estimat over ekkoet man forventer. Det estimerte ekkoet blir subtrahert bort fra det talesignalet som oppsto på "tail circuit"-siden av ekkoutlignereren hos mottageren. På denne måten får normal tale passere igjennom, mens ekko blir fjernet. Moderne ekkoutlignere har evnen til å tilpasse seg signal- og "tail circuit"-forholdene, slik at disse ved oppstart kan konvergere mot det forventete ekkoet (med opprinnelse fra den fjerntliggende samtalepartneren). Slik vil man i starten av en samtale høre ekko, men langsomt vil den forsvinne ettersom ekkoutlignereren konvergerer.

Ekkoutlignere fungerer ikke når samtalepartnerne prater i munn på hverandre, fordi da oppfattes "tail circuit" som ikke-lineær og talen ender opp med å inneholde forstyrrelser. Ekkoutlignerene antar nemlig at "tail circuit" er lineær og tidsuavhengig⁹¹. Selvom et VoIP-system ikke er tilknyttet PSTN og således ikke kan være utsatt for ekko i hybrid, bør ekkoutlignere brukes for å fjerne akustisk ekko [ReR01].

3.4 Klarhet i talen

Klarhet i talen, i en målingskontekst, beskriver den opplevde⁹² lydgjengivelsen⁹³, hvor klar talen oppleves og i hvilken grad talesignalet oppleves som fri for forstyrrelser [Agi00]. Samtidig trekker artikkelen frem at klarhet også kan beskrive hvor forståelig⁹⁴ talen er, altså hvor mye informasjon det er mulig å trekke ut av konversasjonen. Det blir her poengtert at en tale kan være fullt forståelig og allikevel ha en dårlig klarhet. Derfor påpeker artikkelen at det bør skilles mellom klarhet og forståelighet. [WaS97] understreker at man ikke ene og alene bør forbinde forståelighet med kvalitet, da det er en del andre faktorer (for eksempel lytteinnsats, volum på talen) enn forståelighet som kan spille inn på kvalitetsoppfatningen

⁹¹ På engelsk: "time-invariant".

⁹² På engelsk: "perceptual".

⁹³ På engelsk: "fidelity".

⁹⁴ På engelsk: "intelligibility".

[PvT95]. Artikkelen fastslår, likevel, at forståelighet er den største bestemmende faktoren for talekvalitet.

Hvordan klarhet i talen blir bedømt av mennesker er avhengig av en del faktorer. Blant annet vil frekvensbåndet til talen være viktig. Forstyrrelser og tap i effekt⁹⁵ i 1000-1200 Herz (Hz)-båndet vil sannsynligvis oppleves å minske klarheten og forståeligheten mer enn tilsvarende forringelser i 250-800 Hz båndet [Agi00].

Videre, i et PSTN-til-VoIP scenario, vil PSTN-telefonen påvirke klarheten i talen ved blant annet kvaliteten på både høytalere og mikrofon, og akustisk ekko som oppstår mellom disse [Agi00]. IP-telefoni gateway-en som forbinder PSTN-nettet og *IP*-nettet vil også introdusere forringelse av klarheten igjennom codec-ene, stillhetsundertrykkelse⁹⁶ og ”comfort noise”-generering (disse to begrepene ble forklart i kapittel 2.3) [Agi00]. VoIP-terminalen vil også påvirke klarheten på talen igjennom codec-er, stillhetsundertrykkelse tillegg til kvaliteten på høytalere og mikrofon [Agi00].

⁹⁵ På engelsk: ”attenuation”.

⁹⁶ På engelsk: "silence suppression". Den norske betegnelsen vil bli brukt i det følgende.

Kapittel 4

Talekoding

I dette kapitlet skal prosessene som inngår i digitaliseringen av analog tale redegjøres for. Videre skal det belyses hvilke attributter som er viktig for valg av riktig codec til en applikasjon, før de tre mest omtalte codec-ene i IP-telefoni litteraturen (G.723.1, G.729 og G.729A) skal sees litt nærmere på med hensyn på de viktige attributtene. Til slutt i kapitlet blir et eksempel på et forsinkelsesbudsjett presentert.

4.1 Digitalisering av analog tale.

Slik det er vist i figur 2.1 i kapittel 2 ”IP-telefoni arkitekturer”, utgjør digitalisering og komprimering av analog tale en helt essensiell del av prosessen med å sende tale i IP-pakker over ett nettverk til en mottager. Uten disse to fasene vil det ikke være noen tale å sende avgårde. I dette delkapitlet skal de ulike stegene i digitaliseringen av analog tale påpekes og forklares på en lettfattat måte.

”Pulse Code Modulation” (PCM) benytter seg av såkalt ”waveform coding” [Cis01]⁹⁷. Det vil si at den lager en kodet versjon av den originale lydbølgen. I [Cis01] beskrives tre steg for å konvertere analog til digital tale. Disse skal beskrives i de påfølgende delkapitler.

4.1.1 Filtrering

Den analoge filtreringen gjøres for å eliminere uønskete signaler fra talen, støy med andre ord, som kan forstyrre konversasjonen eller som kan skape feil i kontrollsignaler [Cis01b]⁹⁸. Signalene som passerer igjennom filteret sies å tilhøre ”passbåndet”. Transmisjon av tale krever ikke hele båndbredden, passbåndet for tale er derfor begrenset til 300-3300 Hz [Cis01b]. Studier har nemlig vist at det meste av energien i muntlig tale ligger mellom 200-300 Hz til rundt 2700-2800 Hz. Allikevel,

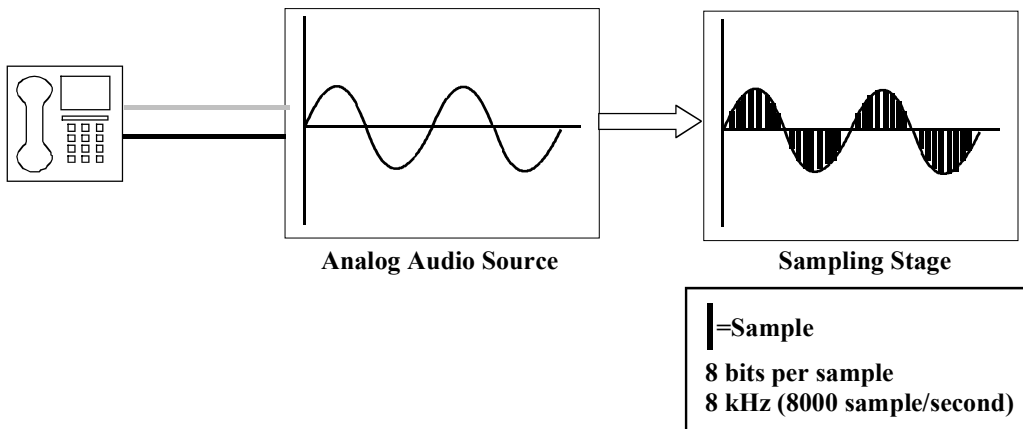
⁹⁷ Ikke trykket materiale, hentet fra http://www.cisco.com/warp/public/788/signalling/waveform_coding.html.

⁹⁸ Ikke trykket materiale, hentet fra http://www.cisco.com/warp/public/788/signalling/define_analog_voice.html.

er båndbredden⁹⁹ til passbåndet for en talekanal i telefonsystemer, såkalt "VF channel", avgrenset til 4000 Hertz (Hz). Dette er for å unngå et fenomen som kalles "antialiasing" [Cisco01]. Dette innebærer at deler frekvensspekteret overlapper og medfører at spekteret ikke kan separeres ved filtrering [RoC95]. For å unngå "antialiasing" må det sørges for at signalspekteret kuttes ved en frekvens x (her: 4000 Hz) og at samplingsfrekvensen minst må være $2x$ (samplingsfrekvens blir mer omtalt i neste delkapittel) [RoC95].

4.1.2 Sampling

Det neste steget er å sample¹⁰⁰ det filtrerte signalet med en konstant frekvens. Dette gjøres ved hjelp av en metode som kalles "Pulse Amplitude Modulation" (PAM). I PAM blir amplituden til en serie av pulser, såkalt *pulstog*, variert med et modulerende signal [RoC95]. Det vil si det signalet som former amplituden til pulstogene, som i dette tilfelle vil være det analoge talesignalet. Resultatet vil på utgangen av PAM være en serie av pulser hvis amplituder varierer i samsvar med det analoge signalet [RoC95]. Pulstoget har en konstant frekvens, kalt *samlingsfrekvens* (f_s). Denne er gitt ved $f_s=1/T_s$, hvor T_s er perioden¹⁰¹ til pulstoget [RoC95]. Denne er definert i Nyquist's teorem til å være minst to ganger den høyeste frekvensen i det analoge talesignalet. Nyquist fant ut at dette gir nok samplinger til at talen kan gjenskapes i et lavpass filter hos mottageren. Likevel, om IP-pakkene tapes på vei til mottager vil det oppleves glipp i talen, det vil si at i korte øyeblikk blir talen borte. For store pakketap vil kunne gjøre talen bortimot meningsløs. Som nevnt i kapittel 3.3.1 finnes det en del metoder for å reparere talen som følge av pakketap.



Figur 4.1: Det analoge signalet samples med en konstant samplingsfrekvens på 8000 Hz. Hvert sample (skissert som vertikale søyler i figuren) tilordnes et digitalt kodeord utfra hvilket kvantiseringsnivå det analoge signalet tilsvarer. Se delkapittel 4.1.3. (Basert på [Cis01].)

⁹⁹ Differansen mellom høyeste og laveste frekvens.

¹⁰⁰ Punktprøve på norsk. Siden "sampling" er en såpass mye brukt fagbetegnelse, vil denne også brukes i det følgende.

¹⁰¹ Differanse i tid fra begynnelsen av en puls til begynnelsen av neste.

4.1.3 Kvantisering

Prosesen med å digitalisere samplene fra forrige fase kalles "Pulse Code Modulation" (PCM). Dette er en variant av PAM, hvor pulsens amplituder blir overført i binær kode [RoC95]. Disse kodeordene representerer altså amplituden på lydbølgen på det tidspunktet samplet ble tatt [Cis01c]¹⁰². På denne måten fungerer PCM som en analog-til-digital konverterer på avsendersiden og som digital-til-analog konverterer hos mottageren.

Teknikken som PCM bruker til dette, kalles *kvantisering*. Dette er den prosessen som avrunder samplenes amplitudeverdier til visse forhåndsdefinerte nivåer, slik at analog-til-digital konverteren (som utfører selve digitaliseringen) får en endelig og håndterbar mengde nivåer å jobbe på [RoC95]. Hvis ikke, ville nivåene kunne bli tilordnet enhver verdi i hele spekteret¹⁰³ på det analoge signalet, hvilket i verste fall kunne medføre uendelige antall nivåer [RoC95]. I kvantiseringsprosessen blir hele spekteret på det analoge signalet delt inn i subnivåer (se figur 4.2a), hvor midtverdien av hvert slikt subnivå utgjør kodeordet for det subnivået [RoC95]. Komparatorer brukes for å bestemme hvilket subnivå en gitt pulsamplitude tilhører og genererer utfra dette kodeordet for akkurat det nivået [RoC95]. I figur 4.2a vises et forenklet tilfelle med 8 nivåer med tilhørende 3 bits binære verdier¹⁰⁴. Det første bitet angir polariteten på det analoge signalet, "1" for en amplitude større enn 0 Volt og "0" for en amplitude mindre enn 0 Volt. De to neste bitene angir hvilket subnivå samplet ligger i. For eksempel vil et sample med negativ polaritet og som ligger i subnivå tre gi et kvantisert sample med det binære kodeordet 011 (se figur 4.2a) [RoC95]. Figur 4.2b viser de binære representasjonene av de syv samplingene i figur 4.2a.

Selv om kvantiseringen sees på som en egen fase, vil den i praksis være integrert i den analoge-til-digitale konverteringen og det finnes Integrerte Kretser¹⁰⁵-er som implementerer alle fasene i konverteringen til digital tale [RoC95]. Slik som vist i figur 4.1 sampler PCM 8000 ganger i sekundet¹⁰⁶, med en samplingsfrekvens på $2 \cdot 4000$ Hz i henhold til Nyquist's teorem. Om hvert kodeord for eksempel er 8 bit lange, blir maksimal bitrate for PCM lik $8 \text{ bit} \cdot 8000 \text{ Hz}$ ¹⁰⁷ = 64000 bps = 64 kbps.

Når subnivåene er dynamisk fordelt igjennom hele spekteret på det analoge signalet, kalles dette *uniform* kvantisering [Cis01]. I slik kvantisering blir samplene tildelt et kvantiseringsnivå som er nærmest mulig amplitudens høyde. Feil blir introdusert i PCM-fasen om kvantiseringsnivåene tildelt samplene ikke tilsvarer den reelle høyden på amplituden (se figur 4.2a). Denne feilen kalles *kvantiseringsstøy* og tilsvarer den tilfeldige støyen som påvirker signal/støy-forholdet, SNR¹⁰⁸, for et talesignal [Cis01]. [RoC95] sier mer presist at denne kvantiseringsstøyen oppstår som følge av differanse mellom kvantiseringsnivåene og den analoge input-en, lydsignalet med andre ord. En metode for å redusere kvantiseringsstøyen er å øke antall kvantiseringsnivåer, slik at

¹⁰² Ikke trykket materiale, hentet fra http://www.cisco.com/warp/public/cc/so/cuso/epso/packv_in.htm.

¹⁰³ På engelsk: "range". En annen norsk oversettelse kunne å ha vært "spenn".

¹⁰⁴ I digital telefoni vil normalt 8 bits (256 forskjellige nivåer) eller 16 bits (65536 nivåer) kodeord brukes.

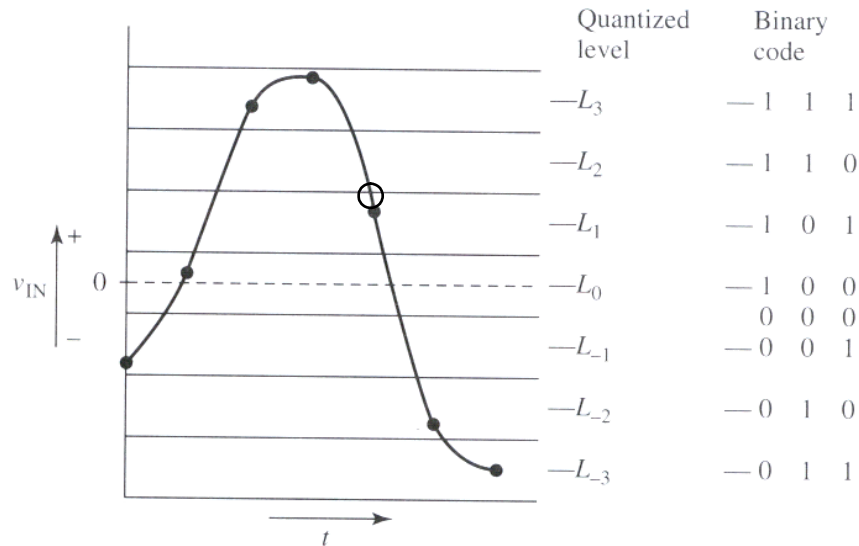
¹⁰⁵ "Integrated Circuits" (IC) på engelsk.

¹⁰⁶ Forutsatt her at PCM betegner hele prosessen med å digitalisere talen, inkludert alle de tre nevnte stegene.

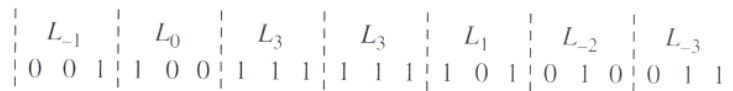
¹⁰⁷ Frekvens=1/T, hvor T er perioden på signalet i sekunder. Måleenhet for frekvens blir da 1/s=Hz.

¹⁰⁸ Av den engelske betegnelsen: "Signal-to-noise ratio". SNR måles i decibel (dB). Jo høyere SNR-verdi, desto bedre talekvalitet.

differansen mellom amplitudeverdien og kvantiseringsnivået minker [Cis01]. Dette forutsetter også at antall kodeord øker proporsjonalt med økningen i nivåer.



(a)



(b)

Figur 4.2: Det analoge signalet deles inn i subnivåer, hvor midtverdien av hvert slikt subnivå utgjør kodeordet for det subnivået (a). Det første bitet i kodeordet angir polaritet, mens de to neste angir hvilket subnivå samplet tilhører. De binære representasjonene av de syv samplingene i a) er vist i b). Om ikke kvantiseringsnivåene tildelt et sample tilsvarende den reelle høyden på amplituden til det analoge signalet blir feil introdusert under kvantiseringen. Dette er markert med sirkelen i figuren. De svarte punktene tilsvarer søylene brukt i figur 4.1 til å skissere sampler. Figuren er hentet fra [RoC95].

Et annet problem som kan oppstå siden det blir brukt dynamisk fordelte kvantiseringsnivåer for hele spekteret i det analoge signalet, er at signaler med lav amplitude vil ha lav SNR, mens signaler med høy amplitude vil ha høyere SNR [Cis01]. Dette er lite hensiktsmessig, da de fleste signalene er lave. For å utbedre talekvaliteten ble det derfor utviklet en metode som kalles "companding" som er ikke-uniform. "Companding" er en sammenslåing av ordene "compressing" og "expanding", og går kort ut på å komprimere samplene hos avsenderen til logaritmiske segmenter før hvert segment blir kvantisert og kodet ved hjelp av uniform kvantisering [Cis01]. Komprimeringen er logaritmisk på den måten at komprimeringen øker ettersom signalets størrelse øker. Hos mottageren blir signalet dekomprimert til sin originale størrelse [Cis01]. Det finnes to typer "companding"-metoder, "A-lov" brukt i Europa og " μ -lov" brukt i Nord-Amerika og Japan. De matematiske likningene som forklarer forskjellen mellom A-lov og μ -lov er redegjort for i [RoC95].

Det finnes to andre versjoner av PCM, "Differential PCM" (DPCM) og "Adaptive DPCM" (ADPCM). DPCM sender kun differansene mellom to sampler istedenfor å

sende hele samplet, mens ADPCM innretter seg etter kvantiseringsnivåene på differansene [Cis01]. Det finnes allikevel en ulempe med disse metodene for å digitalisere analog tale, nemlig at for å spare inn på båndbredde må den kodes med færre bit [Cis01c]. En 8 bits kode gir 256 forskjellige amplitudenivåer, mens 4 bits kode kun gir 64 forskjellige nivåer. Slik kan det digitaliserte signalet bli ulikt det analoge og følgelig kan talekvaliteten degraderes.

”*Linear Predictive Code*” (LPC) er en annen type kodingsmetode, som baserer seg på modeller av den menneskelige talen virkelige fungerer [Cis01c]. Dette er en type koding som gir bedre komprimering av talen samtidig som talekvaliteten bevares. Det er vanlig at disse typene metoder tar inn 64 kbps PCM på inngangen. De nyeste codec-ene tilhører denne gruppen, tre av dem vil bli sett litt nærmere på i neste delkapittel.

For mer om PCM og ADPCM bes leseren om å henholdsvis konsultere følgende ITU-T standarder: G.711 og G.726. [Cis01] forklarer også DPCM og ADPCM på en kort og lettfattet måte. DPCM blir også forklart i [RoC95]. For inngående beskrivelser av talekoding kan ”*Speech Coding: A Tutorial Review*” av A.S Spanias [Spa01] og ”*Speech coding and synthesis*” av W.B. Kleijn og K.K. Paliwal [KIP95] anbefales.

4.2 Codec-er

Talekoding eller talekomprimering er det fagområdet som sørger for kompakt representasjon av talesignaler med det for øyet å oppnå effektiv transmisjon eller lagring [Spa01]. Målsetningen med talekoding er å representere tale med et minimum av bits, samtidig som kvaliteten bevares [Spa01].

I litteraturen blir begrepene ”*vocoder*”¹⁰⁹, ”*speech coder*” og ”*codec*”¹¹⁰ noen steder brukt om hverandre, mens i [Spa01], for eksempel, blir ”*vocoder*” brukt om en ”*speech coder*” som benytter talemodeller, og som ikke nødvendigvis prøver å gjenskape selve lydbølgen¹¹¹. Tilsvarende blir PCM i noen kilder benevnt som en codec, mens andre steder ikke. [Spa01] beskriver 64 kbps PCM som en ikke-parametrisk kvantiserer (det vil si lydbølgebasert). Altså en talekodingsmetode som lager en binær representasjon av talesamplene, slik som beskrevet i forrige delkapittel. [Spa01] sier også at det er vanlig å betegne 64 kbps PCM-digital tale som ukomprimert og at den ofte blir brukt som referanse i sammenlikninger med komprimeringsmetoder. Definisjon i [Spa01] av PCM som en kvantiserer stemmer godt med den generelle beskrivelsen av G.723.1 gitt i ITU-T Recommendation G.723.1 [ITU96b]:

”This coder is designed to operate with a digital signal obtained by first performing telephone bandwidth filtering (Recommendation G.712) of the analogue input, then sampling at 8000 Hz and then converting to 16-bit linear PCM for the input to the encoder.”

¹⁰⁹ Forkortelse for ”*voice coder*”.

¹¹⁰ Forkortelse for ”*coder-decoder*”. Noen kilder bruker også ”*compressor-decompressor*”.

¹¹¹ På engelsk: ”*waveform*”.

Slik det fremgår av sitatet, tar G.723.1 inn PCM på inngangen til koderen og jobber med det som råmateriale i produksjonen av komprimert tale på utgangen. Derfor vil, til tross for sprikende kilder, PCM bli omtalt som en ”kvantiserer” i det følgende, hvor talen er ukomprimert. Begrepet "codec" vil i det følgende bli brukt om talekodingsmetoder, slik som G.723.1, G.729 og G.729A, hvor talen blir komprimert.

Nøkkelen til å finne riktig codec for en applikasjon ligger i dens attributter. Ut fra hvilke krav den aktuelle applikasjonen stiller til en codec, kan visse attributter vektlegges spesielt mens andre kan det tas litt lettere på [Cox97]. [Cox97] deler de nevnte attributtene inn i fire hovedkategorier: *bitrate*, *forsinkelse*, *kompleksitet* og *kvalitet*.

I det følgende skal det sees litt nærmere på disse fire kategoriene, deretter skal det belyses hvilke attributter de vanligste codec-ene i VoIP har.

4.2.1 Bitrate

Bitrate er muligens den mest kjente attributten ved en codec. Bitraten angir hvilken ”oppløsning” lyden (her: talen) gjengis med og måles i bit per sekund (bps). I delkapittel 4.1.3 ble det gitt et eksempel på hvordan bitrate regnes ut. Desto større bitrate, jo bedre kvalitet er det på den digitale gjengivelsen av det opprinnelige analoge talesignalet. Bitraten gjenspeiler også hvilken overføringskapasitet eller båndbredde som kreves for å få overført selve talerammen. Samtidig er det viktig å huske på at enkelte codec algoritmer kan komprimere ned bitraten, som er nødvendig for å gjengi et sample, ved å kun sende avgårde forenklete parametre som representerer den opprinnelige talen. Slik sett vil ikke bitraten alltid i seg selv nødvendigvis angi hvor god kvalitet talesamplene holder. Slike codec-er kalles ofte ”*source-codec*” eller ”*parametric codec*” i litteraturen, mens codec-er som utnytter karakteristikker i den opprinnelige lydbølgen ofte kalles ”*waveform-codec*” (PCM og ADPCM trekkes som frem som eksempler på dette i litteraturen).

4.2.2 Forsinkelse

Forsinkelse spiller som tidligere nevnt en stor rolle for hvordan en IP-telefoni samtale utarter seg. En viktig faktor her er også *enveis codec-forsinkelse*, som kan deles inn i fire separate komponenter [KIP95]:

1. ”*Algorithmic Delay*” er den tiden det tar å akkumulere en hel taleramme før prosesseringen av den kan starte. Eller sagt på en annen måte: den tiden det tar å fylle opp codec-ens input-buffer før selve kodingen kan skje. Denne forsinkelsen kan således sees på som codec-ens buffringsforsinkelse [Che95]. Om codec-en krever en ”lookahead”, vil den totale ”algorithmic delay” være summen av rammelengden og ”lookahead” (begge i millisekunder). Videre vil denne forsinkelsen være avhengig av codec-algoritmen, og ikke av hvordan den er implementert og brukt i et kommunikasjonssystem [Che95].
2. ”*Computational delay*¹¹²” er den tiden selve prosesseringen av en taleramme tar. Denne forsinkelsen vil som regel være lik en rammelengde eller mindre. Denne

¹¹² Også benevnt som ”processing delay” i noen kilder.

forsinkelsen vil være avhengig av hardware-en eller DSP-en codec-algoritmen er implementert på [Che95].

3. ”*Multiplexing delay*” er den tiden det tar å samle opp en blokk av bits tilsvarende en taleramme før transmisjon. Tilsvarende vil blokker av bits bli samlet opp hos mottager før de blir dekodet. Beskrivelsen av innpakkingsforsinkelse i en del litteratur passer inn her, men det er mye sprik i kildene når det gjelder definisjonene av de ulike codec-relaterte forsinkelsene.
4. ”*Transmission delay*” vil være avhengig av om talen skal dele en overføringskanal med annen trafikk, eller om den har en dedikert kanal.

Det antas i [KIP95] at det er rimelig å anslå at ”multiplexing delay” og ”Transmission delay” utgjør opptil en rammelengde om applikasjonen har en dedikert kanal til disposisjon. Dermed vil en tommelfingerregel være at total codec-forsinkelse vil være på minst tre ganger talerammens lengde (i millisekunder, ms), men vil kunne være betydelig lengre for codec-er som krever ”lookahead”. PCM og ADPCM koder talen sample for sample inn i få bits som overføres øyeblikkelig. Slik kan codec-forsinkelsene sees bort ifra og derfor har det heller ikke historisk sett vært fokus på forsinkelse for codec-er med høy bitrate [Che95]. For lav-bitrate codec-er på eller mindre enn 16 kbps, derimot, er dette svært aktuelt fordi de jobber på hele talerammer som inneholder mange sampler.

4.2.3 Kompleksitet

Kompleksitet angir hvor mange prosesseringsinstruksjoner implementasjonen av codec-algoritmen krever av ”Digital Signaling Processor” (DSP) eller ”Central Processing Unit” (CPU) [Cox97]. Denne kompleksiteten angis i ”millioner av instruksjoner per sekund” (MIPS). Andre mål for DSP-kompleksitet er hvor mye ”Random Access Memory” (RAM) og hvor stor ”Read Only Memory” (ROM)-lagringsplass som kreves. For PC-er eller arbeidsstasjoner er kun antall instruksjoner per sekund eneste relevante måleenhet [Cox97]. Antall instruksjoner som kreves av en codec-implementasjon er også avhengig av arkitekturene på DSP-ene fra de ulike leverandørene, hvilket medfører ulik effektivitet i implementasjonen av én codec fra en DSP-leverandør til en annen [Cox97].

4.2.4 Kvalitet

Kvalitet på den dekodete lyden (hos mottager) er avhengig av mange faktorer, slik som beskrevet kapittel 3. En viktig faktor i denne sammenhengen for en codec, er dens robusthet mot bakgrunnstøy¹¹³ eller annen ”fremmed” lyd, hvilket kan kraftig forringe yteevnen¹¹⁴ til lavbitrate codec-er [Cox97]. Robusthet mot bakgrunnstøy vil ha stor betydning for valget av codec til bruk i ”Air Traffic Control” (ATC), da det nødvendigvis vil være en viss bakgrunnstøy i flycockpiter. (Omgivelsene rundt ATC vil bli omtalt mer i kapittel 5.)

¹¹³ På engelsk: ”robustness to background noise”.

¹¹⁴ På engelsk: ”performance”.

4.2.5 IP-telefoni codec-enes attributter

I tabell 4.1 blir de overnevnte attributtene til codec-ene G.723.1 Multi Pulse-Maximum Likelihood Quantization (MPMLQ), G.729 Conjugate Structure-Algebraic Code Excited Linear Prediction (CS-ACELP) og G.729A¹¹⁵ (CS-ACELP) listet opp. G.711 Pulse Code Modulation (PCM) er tatt med i tabellen kun som referanse, fordi bitratene oppført i tabellen er de bitratene man får på utgangen av de nevnte codec-ene med 64 kbps PCM-digitalisert tale på inngangen [KBS98]. "Frame Length" er lengden på hver taleramme i bytes (uten noen hoder).

G.723.1 Annex A beskriver "single-user silence compression" som komprimerer stillhetspartiene i en samtale og følgelig sparer på den måten overføringskapasitet [Cox97]. "Voice activity detector" (VAD) og "comfort noise generator"¹¹⁶ er også definert her. Det tilsvarende for G.729, er definert i G.729 Annex B.

G.729 Annex A inneholder en del forenklinger i forhold til G.729 når det gjelder innsparing av både MIPS og RAM. Dette er ment for applikasjoner hvor den ekstra kompleksiteten i G.729 og G.723.1 utgjør en betydelig byrde. Disse forenklingene innebærer en målbar forringelse i ytelse, men er ikke statistisk signifikante. Dette kommer også frem av siste rad i tabell 4.1.

Attributt	G.711	G.723.1	G.729	G.729A
Bitrate (kbps)	64	5,3 og 6,3 ¹¹⁷	8	8
Frame size ¹¹⁸ (ms)	0.125	30	10	10
Lookahead (ms)	0	7,5	5	5
Prosessering ¹¹⁹ (ms)	-	30	10	10
Total codec-forsinkelse	-	67,5	25	25
Frame Length (byte)	-	20 og 24	10	10
Kompleksitet (MIPS)	<<1 ¹²⁰	16	20	10,5
Kompleksitet (RAM words)	1 byte ¹²¹	2200	3000	2000
Mean Opinon Score (MOS)-resultat	4,1	3,65 og 3,9	3,92	3,7

Tabell 4.1: Sammenlikning mellom G.711 PCM og codec-ene G.723.1, G.729 og G.729A på bitrate, forsinkelser, kompleksitet og kvalitetsrating (MOS). Kvalitetsratingen foregår ved hjelp av subjektiv testing hvor lytterne bedømmer kvaliteten på en skala fra 1 til 5¹²². Rutene med '-' antyder at punktene ikke var spesifisert i kildene. Hentet fra [KBS98] og [Cox97]. MOS-resultatene er hentet fra [DaP00, side 174]. Forskjellen mellom G.729 og G.729A ligger på innspart kompleksitet i G.729A [ITU96c].

[Cox97] trekker frem at den største fordelene med G.723.1 er de lave bitratene, men at den store totale codec-forsinkelsen (67,5 ms jamfør tabell 4.1) innebærer at ikke andre forsinkelser kan bli så veldig store før konversasjonsdynamikken vil bli påvirket. G.729A sitt store fortrinn er dens lave kompleksitet i forhold til G.723.1 og G.729. [ReR01] sier at G.729 sparer inn 87,5% av båndbredden sammenliknet med G.711 PCM og når de ulike pakkehodene tas med oppnås en besparing i båndbredde på 60%. Med hodekomprimering ville besparelsen være høyere [ReR01].

¹¹⁵ Standarderens "Annex A".

¹¹⁶ VAD og "comfort noise" ble nærmere beskrevet i kapitel 3.

¹¹⁷ [KBS98] og [Cox97] oppgir 6.4 kbps, mens standarden oppgir 6.3 kbps [ITU96b]. Antar 6.4 kbps er skrivefeil, og vil bruke 6.3 kbps som ITU-T oppgir.

¹¹⁸ Utgjør den forsinkelsen noen tekster benevner "Packetization delay" eller "innpakningsforsinkelse" på norsk.

¹¹⁹ Også benevnt "Handling delay" i noen tekster.

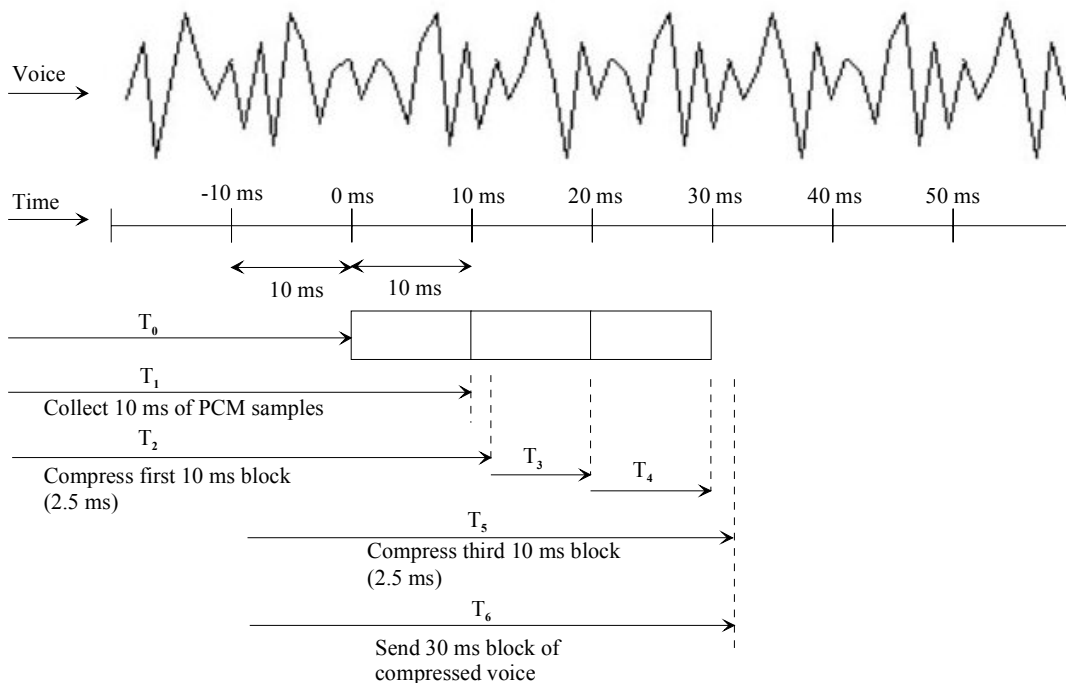
¹²⁰ Hentet fra [Che95].

¹²¹ Hentet fra [Che95].

¹²² Subjektive metoder for talekvalitetsbedømmelse omtalt i Appendiks B.

Tallene i tabell 4.1 viser at den totale codec-forsinkelsen til G.729 er 25 ms for en taleramme. For konfigurasjoner hvor tre talerammer fraktes i IP-pakken, vil ikke den totale codec-forsinkelsen være $3 \cdot 25 \text{ ms} = 75 \text{ ms}$. Den vil imidlertid være langt mindre, fordi generelt vil prosesseringstiden (komprimeringen) for taleramme n og innpakkingsforsinkelsen (samle opp sampler tilsvarende 10 ms tale) for $n+1$ vil overlape for $n > 1$. Dette er vist i figur 4.3, hvor et eksempel med G.729 er gitt. Det er i eksempelet regnet med en prosesseringstid på 2.5 ms, mens "lookahead" ikke er tatt med.

Slik det fremkommer av figuren, vil komprimeringen av den første talerammen overlape med innsamlingen av sampler til den andre talerammen. Tilsvarende gjelder for taleramme to og tre. Slik ender man opp med 10 ms forsinkelse i innsamlingen av sampler til taleramme 1, mens prosesseringen av ramme 2 og 3 overlapper med innsamlingen av sampler til henholdsvis ramme 3 og 4. Disse prosesseringstidene vil således ikke bidra ytterligere til en større total codec-forsinkelse, kun prosesseringen av den siste rammen bidrar i så måte. Dermed ender man opp med 30 ms innpakkingsforsinkelse for de tre rammene pluss 2.5 ms prosessering for den siste rammen, hvilket gir en totalsum på 32.5 ms. Hadde man regnet 10 ms prosesseringstid, hvilket er "worst-case", ville man ha endt opp med en total codec-forsinkelse på 40 ms for tre talerammer.



Figur 4.3: På grunn av at sampler må samles opp for å "fylle opp" en taleramme på 10 ms (såkalt "pipelining"), vil ikke prosesseringstiden (komprimeringen) for pakke to og tre bidra med ekstra total codec-forsinkelse. På denne måten overlapper prosesseringstiden for taleramme n og innpakkingsforsinkelsen til ramme $n+1$ når $n > 1$. (Figuren er basert på [Cis02].)

4.3 Forsinkelsesbudsjett

I dette delkapitlet presenteres et eksempel på et ende-til-ende forsinkelsesbudsjett¹²³ med G.729 som codec (se tabell 4.2).

I eksempelet i tabell 4.2 er innpakkingsforsinkelsen (beskrevet i kapittel 3.3.2) ikke innregnet i codec-forsinkelsen, men kommer i tillegg. [DaP00, side 191], derimot, innregner innpakkingsforsinkelsen i codec-forsinkelsen, slik at den for en IP-pakke inneholdende to talerammer blir på totalt 25 ms. Forsinkelsene fordeler seg da slik: 10 ms prosessering av den første talerammen (det vil si samling av ferdig digital tale fra PCM og komprimering), 5 ms lookahead og 10 ms prosessering for den andre talerammen. [Cis02], derimot, som vist i tabell 4.2 innregner ikke innpakkingsforsinkelsen i codec-forsinkelsen.

Det ser ut til at [DaP00, side 191] av uklare årsaker ser bort ifra innpakkingsforsinkelsen. Det er tydelig sprik i hvordan litteraturen beregner forsinkelsesbudsjettene. I dette kapitlet benyttes beregningsmetoden i [Cis02], hvilket gir en klar sammenheng med dataene i tabell 4.1.

For tallene gitt i tabell 4.2 som angår forsinkelser tilknyttet switche- og rutingsforsinkelser og køforsinkelser i nettverket, vil totalsummen være avhengig av antall hopp mellom endepunktene talen fraktes i mellom. Dekodingsforsinkelsene hos mottager er i tabellen oppgitt til 10% av kodings-/komprimeringsforsinkelsen¹²⁴ (som for G.729 er på 10 ms i henhold til prosesseringsforsinkelsen oppgitt i tabell 4.1) hos avsender multiplisert med antall talerammer som ble fraktet i den aktuelle IP-pakken. I delkapitel 3.3.2 ble det nevnt at dekodingsforsinkelsen er halvparten av codec-forsinkelsene, som omfatter "frame size", "lookahead" og prosesseringstiden [KBS98]. Den første metoden vil for tre talerammer gi en dekodingsforsinkelse på 3 ms, mens den andre vil gi en verdi på 22,5 ms (med tall fra slutten på kapittel 4.2.5: $[40 \text{ ms (total codec-forsinkelse}^{125}) + 3*5 \text{ (lookahead per ramme)}]/2$). En mulig årsak for dette avviket, kan være at den første metoden ikke tar med prosesseringsforsinkelsen i kalkuleringen av codec-forsinkelsen. Summen av de variable forsinkelsene i tabell 4.2 er med på å utgjøre det jitteret som påvirker talekvaliteten hos mottageren.

¹²³ På engelsk: "delay budget".

¹²⁴ I tabell 4.1 benevnt "prosessering".

¹²⁵ Lookahead var ikke inkludert i dette tallet.

			Fast forsinkelse (ms)	Variabel forsinkelse (ms)
Samlet forsinkelse hos avsender	Codec- forsinkelse	Komprimering Lookahead	15	
	Innpaknings- forsinkelse		10	
	Køforsinkelse			$<10^{126}$
Total rutings- forsinkelse avhengig av antall hopp	Serialiserings- forsinkelse		3^{127}	
	Propagasjons- forsinkelse		$0,006/\text{km}^{128}$	
	Switche/rutings- forsinkelse			25^{129}
Samlet forsinkelse hos mottager	Køingsforsinkelse			$0,1-0,2^{130}$
	Jitterbuffer		$50-100^{131}$	
	Dekodings- forsinkelse		10% av komprimerings- forsinkelsen * antall talerammer per IP- pakke	

Tabell 4.2: Et eksempel på forsinkelsesbudsjett tatt utgangspunkt i G.729 codec-en. Tallene er hentet fra [Cis02]¹³², med mindre annet er spesifisert.

G.729 får levert blokker med 10 ms digitalisert tale fra G.711 PCM. Det er altså PCM som utfører selve konverteringen fra analog til digital tale, kvantiseringen, slik som beskrevet tidligere i kapitlet. Den tiden G.729 bruker på å komprimere¹³³ den digitale talen utgjør codec-forsinkelsen [Cis02]. I tabell 4.2 har også lookahead-forsinkelsen blitt tatt med i codec-forsinkelsen, fordi denne inngår i den prosesseringen codec-en utfører. Dette skjer for hvert talesample og akkumulerer en total ekstra forsinkelse på 5 ms for G.729 [Cis02].

Hovedsakelig finnes det to scenarier i VoIP for hvor den analoge-til-digitale konverteringen og komprimeringen gjøres, dette er vist i figur 4.4a og b. Enten kan den analoge talen bli digitalisert i en digital hussentral¹³⁴ og komprimert i IP-telefoni ruterer (figur 4.4a), eller kan begge disse prosessene skje i ruterer (figur 4.4b) [Cis02].

¹²⁶ Denne forsinkelsen vil være avhengig av hastigheten på transmisjonslinjen og tilstanden på køen. Siden talepakker skal ha prioritet foran datapakker, vil talepakken kun vente bak andre talepakker og evt. datapakker som er igang med å bli sendt avgårde. Slik sett vil serialiseringsforsinkelsen spille inn, altså hvor hurtig pakker kan bli kloknet ut på nettverkskortet [Cis02]. [DaP00, side 168] sier at denne forsinkelsen bør holdes under 10 ms. For en 64 kbps linje anslår [DaP00, side 191] denne til 6 ms.

¹²⁷ Tallet gjelder for en 64 kbps linje. Hentet fra [DaP00, side 191].

¹²⁸ Dette er kun et estimat, da en del faktorer i bærernettet påvirker dette tallet, slik som utstyr for multipleksing og mikrobølge-linker. Dette estimatet har [Cis02] funnet i ITU-T standard G.114. Paperet henviser til dataene for US Frame Relay som ifølge forfatteren oppgir et typisk "worst case"-tall på 40 ms.

¹²⁹ Dette er en faktor som lar seg måle i private nettverk eller utlede estimat for i WAN. [Cis02] henviser til dataene for US Frame Relay, som ifølge forfatteren oppgir et typisk "worst case"-tall på 25 ms.

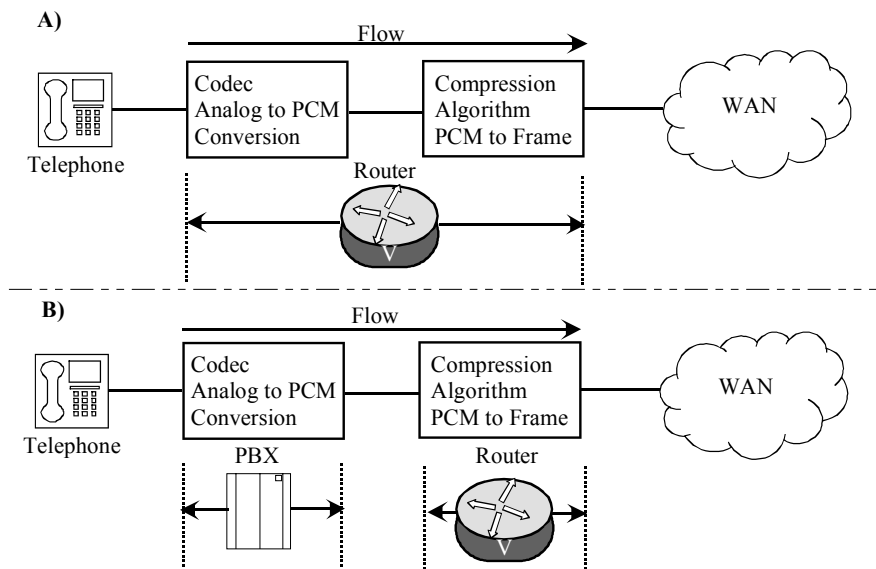
¹³⁰ I eksempler gitt i [Cis02] oppgis både 0,1 og 0,2 ms køforsinkelse for ruterer inne i nettverket.

¹³¹ Tallet er hentet fra [HNA00].

¹³² Ikke trykket materiale, hentet fra www.cisco.com/warp/public/788/voip/delay-details.html.

¹³³ Ifølge [IHM⁺99] klarer G.729 å komprimere ned en 80 byte stor talepakke ned til 10 byte.

¹³⁴ PABX/PBX



Figur 4.4: Talen blir først digitalisert og deretter komprimert ned i en ruter før talen sendes avgårde i en IP-pakke (A). Alternativt kan talen digitaliseres i en hussentral (PBX) og deretter bli komprimert i en ruter (B). (Basert på [Cis02].)

Kapittel 5

Bakgrunn for forsøksoppsettet

For å ta en avgjørelse på hvordan eksperimentene skal bygges opp, er man nødt til å se på hva litteraturen sier om hvordan "Air Traffic Control" (ATC) arter seg og hvilke regler den må følge. Deretter er man nødt til å ta utgangspunkt i de ITU-anbefalinger og råd som gjelder for subjektiv testing i telekommunikasjonsverdenen, ikke minst hvordan forskning på dette området har gjort det. Det vil også være nyttig å se til forskning på subjektiv kvalitetstesting av ATC-kommunikasjon. Selvom disse har hatt andre målsetninger med testingen enn dette prosjektet, vil det være interessant å se hvordan de la opp eksperimentene og tok hensyn til de viktige faktorene i ATC-kommunikasjon. Eksperimentoppsettet i dette prosjektet bør således ha de gjeldende råd og anbefalinger for subjektiv testing som basis, samtidig som testingen må tilpasses slik ATC-kommunikasjon utøves i praksis og de strenge sikkerhetsreglene som stilles til slik kommunikasjon.

I dette kapitlet vil det blant annet sees på hvordan ATC-tale arter seg, hvilke sikkerhetsaspekter som gjelder for ATC. Deretter vil det på generelt grunnlag bli diskutert hva forsøksoppsett for testing av VoIP-basert ATC bør ta hensyn til. Diskusjonen blir deretter kort oppsummert med hensyn på fordeler og ulemper. Til slutt vil det valgte eksperimentoppsettet i dette prosjektet bli redegjort for og begrunnet. Det blir også redegjort for valg av skalaer og målsetning med forsøkene. Beskrivelsene av det valgte eksperimentoppsettet vil i dette kapitlet være av en overordnet karakter, mens *implementeringen* av oppsettet blir grundig redegjort for i kapittel 6.

5.1 ATC taleteknikk og -kvalitetsbedømmelse

[Pat01]¹³⁵ presenterer de grunnleggende reglene som gjelder for taleteknikk i kommunikasjon mellom tårn og fly. Disse, blir det understreket, må alltid overholdes for til enhver tid å sørge for effektivitet i kommunikasjonen og at forståelsen mellom kommunikasjonspartnerne er maksimal. Beskjedene som gis skal være korte, lett

¹³⁵ Ikke trykket materiale, se URL: <http://fly.to/kpatouras.gr>.

forståelige og i en dagligdags tone. For å underbygge det siste punktet blir det presisert at hverken følelseladde utbrudd eller ekstrem høflighet er akseptert oppførsel. Videre skal snakkehastigheten alltid være jevn og dens øvre grense er på 100 ord per minutt. Ved formidling av tallstørrelser eller kritisk informasjon som skal noteres av mottageren, bør hastigheten reduseres og en pause tas. Mikrofonen skal alltid være i en fast avstand til munnen og talevolumet skal holdes jevnt under kommunikasjonen. Om selve plasseringen av mikrofonen sier [FAA01]¹³⁶ at den skal være tett inntil leppene. Videre blir det også presisert at man bør vente litt etter ”push-to-talk”-knappen har blitt trykket for å sikre at også det første uttalte ordet blir overført. Dette problemet er det samme som man i VoIP-terminologien kaller ”first-word clipping” (dette ble omtalt i kapittel 3).

En ATC-beskjed består av et kallesignal¹³⁷ og et spørsmål, en instruksjon eller informasjon [MGNM90]. Videre når en flygeleder stiller et spørsmål om en parameter, skal piloten alltid svare med verdien på den aktuelle parameteren. Når en flygeleder gir en instruksjon, skal piloten repetere instruksjonen for å vise at vedkommende har forstått den. En pilot kan også selv stille spørsmål eller anmode om autorisasjon fra flygeleder om et eller annet.

En sjekk av kvaliteten på kommunikasjonen blir initiert av enten en pilot eller flygeleder, og kalles på ATC-terminologien for ”radio check”. En slik sjekk bedømmer overføringen utfra tre kriterier: (lyd-) styrke, hvor lett den er å forstå¹³⁸ og (total) kvalitet [Pat01b]¹³⁹. Den ene parten tar kontakt med den andre og ber om en bedømmelse på en skala fra 1 til 5. Skalaen er vist i tabell 5.1.

Skala	”Readability”	Beskrivelse
1	”Unreadable”	Veldig dårlig, sterk bakgrunnstøy, forferdelig
2	”Readable” i korte øyeblikk	Dårlig, forstyrret av veldig dårlig overføring. Om man snakker sakte eller gjentar hvert ord er det mulig å få en viss forståelse.
3	”Readable” med vanskeligheter	Rimelig bra (”fairly good”) overføring og forståelig, men ikke sterk.
4	”Readable”	Veldig bra, rimelig bra styrke.
5	”Perfectly Readable”	Utmerket.

Tabell 5.1: Skalaen som blir brukt i ATC-sammenheng til å bedømme graden av forståelighet [Pat01b].

Ut fra skalaen i tabell 5.1 vil det være naturlig å slutte at forståelighet, lydstyrke og bakgrunnstøy er de faktorene som spiller mest inn på ATC-kommunikasjon og som man legger mest vekt på i bedømmelsen av hvilken kvalitet en forbindelse holder. Videre minner denne skalaen noe om de brukt i lyttetester, med elementer fra både innsatsbedømmelsen i ”Listening effort”-skalaen (MOS_{LE}) og kvalitetsbedømmelsen i ”Listening-quality”-skalaene (begge er beskrevet i appendiks B og er også gjengitt i tabell 5.2).

¹³⁶ Ikke trykket materiale, se URL: <http://www2.faa.gov/ATPubs/AIM/Chap4/aim0402.html>.

¹³⁷ På engelsk: ”Call-Sign”.

¹³⁸ På engelsk: ”readability”. Den norske betegnelsen vil bli brukt i det følgende.

¹³⁹ Ikke trykket materiale, se URL:

<http://fly.to/kpatouras.gr>.

5.2 Sikkerhetsaspekter i ATC og faktorer som påvirker talekvaliteten

Det er et absolutt krav at når en av deltagerne i en ATC-kommunikasjon ønsker å formidle en beskjed til den andre, skal denne oppfattes korrekt hos mottageren. Dette betyr at forholdene må ligge til rette for dette, både i form av kvalitet på talen og at talen garantert kommer frem til mottageren. Det kan være svært kritisk, sågar livstruende om beskjeder som gis ikke kommer frem, bare delvis kommer frem eller kun er delvis forståelig. I dette perspektivet er det enten- eller, enten kommer beskjeden frem til mottageren så tydelig at den kan forstås uproblematisk eller, i et ”worst case”-scenario, er det kanskje bedre at den ikke kommer frem. Dette ville i så fall tvinge frem en gjentagelse av meldingen. Men, det er viktig å understreke at det er et ufravikelig krav at talen skal komme frem og holde en slik standard at det ikke er tvil om hva som sies.

En utfordring ved å gjøre ATC-kommunikasjon VoIP-basert blir å kunne forsikre at talen inneholdt i en IP-pakke virkelig kommer frem til mottageren. Slik som forklart i kapittel 3, er IP-pakker utsatt for å bli forsinket såpass lenge at talen ikke lar seg rekonstruere eller de kan bli tapt slik at viktige deler av talen blir borte. En løsning på dette kan være å sende pakkene redundant flere veier for å virkelig være sikker på at pakkene vil komme frem. Prioriterings- og ressursreserveringsmekanismer vil også være en løsning på dette, slik som nevnt i kapittel 3. Videre vil ATC-talen sendt i IP-pakker være utsatt for en del andre faktorer, slik som ekko og jitter (se egen omtale). Følgelig vil det i talekvalitetstestinger i utgangspunktet være aktuelt å utsette IP-pakkene for en del varierende forhold når det gjelder pakkeforsinkelse¹⁴⁰, ulike tapsforhold, jitter og ekko for å se hvordan dette påvirker kvaliteten. På en annen side, slike eksperimenter bør kun teste ut forhold som er realistisk relatert til de talekvalitetskrav som gjelder i den sammenhengen talen skal brukes. Er det for eksempel en nulltoleranse for ekko, er det ikke noen grunn til å teste ut hvilke innvirkninger dette gir på talekvaliteten, fordi det i så fall ikke skal være ekko tilstede.

Et annet interessant spørsmål vil også være hva slags nettverksforhold VoIP-basert ATC vil bli utsatt for i reelle anvendelser. For eksempel, vil det være dedikert kapasitet for slik ATC-kommunikasjon eller vil denne konkurrere med annen datatrafikk? Nettverksforholdene vil slik sett ha stor innvirkning på talekvaliteten og derfor ville det være interessant å se hvor grensen for akseptabel talekvalitet i ATC-sammenheng går. En måte for å finne ut av dette vil være å simulere at talen blir utsatt for varierende nettverksforhold for å se hvor mye trafikk talekvaliteten kan tåle og fortsatt møte kvalitetskravene.

Hvilke nettverksforhold det skal testes på i dette prosjektet, vil bli diskutert mer i kapittel 5.5.

¹⁴⁰ På engelsk: ”delay” eller ”latency”.

5.3 Viktige faktorer forsøksoppsett bør ta hensyn til

I det følgende vil det på *generelt* grunnlag bli diskutert fordeler og ulemper i talekvalitetstestinger av VoIP-basert ATC-tale på følgende punkter:

- Type tale
- Talemateriale
- Språk
- Aktuelle typer tester

Hvilke avgjørelser som har blitt tatt på disse punktene i dette prosjektet og hvorfor, vil også bli besvart. Til slutt vil diskusjonen i dette delkapitlet bli kort oppsummert med hensyn til fordeler og ulemper.



Figur 5.1: Flygeledere på jobb i et flyplass tårn. Hentet fra <http://www.howstuffworks.com/air-traffic-control.htm>

P.800 stiller, som beskrevet i appendiks B, krav til lydtette lyttebokser i konversasjonstester og begrensede verdier for romstøy og romklang i lyttetester. I flygelederens daglige arbeidsmiljø vil det antageligvis være en viss bakgrunnsstøy fra andre kollegaer og maskinstøy og så videre. Operatørposisjonene kan være relativt tett plassert (se figur 5.1) og vil således være utsatt for dette, derfor vil det å benytte lyttetette lytterbokser ikke være realistisk i forhold til flygelederens daglige arbeidssituasjon. Videre må det kunne antas at romklangen i rommet de sitter i er såpass liten at den kan neglisjeres. Dessuten vil piloter i sivile fly ha en viss bakgrunnsstøy, militære fly enda mer, som vil ha innflytelse på talekvaliteten. Derfor vil det i så fall være mer nærliggende å kjøre tester hvor en testperson "leker" pilot og har en bakgrunnsstøy på nivå med det man finner i autentiske cockpiter, slik som det ble gjort i [SDCH97] hvor det ble testet ut to sett av codec-er tiltenkt brukt i ATC-kommunikasjon.

[LSBH97] har gjort talekvalitetstester for tale i ATC-sammenheng. Det ble testet på to "vocoder"¹⁴¹ for å sjekke om de holdt mål. Forsøkspersonene hadde sitt daglige virke innen ATC. Testene ble kjørt med tre forskjellige typer bakgrunnsstøy, fordi [LSBH97] anser bakgrunnsstøyen for å ha en svært betydningsfull effekt på ytelsene til en "vocoder". Talen ble begrenset til 30 stavelser. Det ble utført subjektive tester, hvor det ble testet på forståelighet og akseptabelhet (definert til å bestå av følgende delkomponenter: kvaliteten på en beskjed, innsats som trenges for å forstå en beskjed og den potensielle innflytelsen bakgrunnsstøyen har). Det ble også utført såkalte "Message Completion Test", hvor forsøkspersonene fyller inn manglende ord i en tekst utfra hva de hadde hørt. Videre utførte [LSBH97] også "Audio Preference Test" hvor forsøkspersonene sa ifra hvilken av de to "vocoder" de foretrakk i en gitt testkonfigurasjon. De måtte også begrunne hvorfor.

¹⁴¹ Forkortelse for "voice coder".

5.3.1 Type tale

I [SDCH97] ble det videre brukt autentisk "ATC phraseology", det vil si en autentisk flygelederkommunikasjon, fordi testpersonellet i disse eksperimentene bestod av profesjonelle flygeledere og piloter. I en slik testsituasjon kan det argumenteres at kommunikasjonen som finner sted under testingene bør være såpass realistiske at de ikke forstyrrer konsentrasjonen til flygelederne, og derigjennom påvirker bedømmelsen av talen. Det kan jo allikevel diskuteres hvorvidt man skal tilstrebe og bruke ATC-tale om omgivelsene rundt ikke i tilstrekkelig grad klarer å simulere arbeidsplassen til en flygeleder. Bør man i så fall heller bruke ikke-ATC tale og solid markere at det faktisk er en viss forskjell mellom omgivelsene rundt testene og deres daglige arbeidsplass? Faren kan da være at det kanskje kan føles forvirrende for flygelederne å terpe på at dette ikke er en arbeidssituasjon, samtidig som de skal bedømme om talen holder en godkjent kvalitet for ATC-bruk.

I en situasjon hvor testpersonellet utgjøres av vanlige mennesker som ikke har/hatt en jobb innenfor ATC, er situasjonen en helt annen fordi de allikevel ikke vet hvordan ATC-kommunikasjon arter seg. Man må i så fall "kun" forholde seg til de retningslinjer P.800 og "Handbook on Telephony" (se appendiks B) gir for subjektive lytte- og konversasjonstester. Den avgjørelsen som blant annet må tas er hvilken lengde man skal kjøre på talesekvensene for begge typer tester. Skal man kjøre en standard konversasjonsoppgave (også beskrevet i appendiks B) som vil gi konversasjoner på omtrentlig 3 minutter, eller skal man heller benytte korte setninger som sannsynligvis vil være mer likt med varighetene man vil finne i ATC-kommunikasjon? For eventuelle lyttetester må man i tillegg bestemme hva slags materiale man skal lese inn på bånd. Tilsvarende må man for konversasjonstester avgjøre hva som skal drive konversasjonen.

I en situasjon hvor man har et todelt testpersonell, med f.eks. 50% "vanlige" mennesker og 50% profesjonelle flygeledere, kan man si at situasjonen får en liknende karakter som når kun flygeledere benyttes. Derfor kan det her argumenteres med at det bør kjøres på med ATC-kommunikasjon, selvom ene halvparten av testpersonellet ikke kjenner til denne kvaliteten. Slik kan man i resultatene av testene få en ekstra bonus i form at man kan sammenlikne resultatene fra de "vanlige" menneskene og de profesjonelle flygelederne. Sannsynligvis vil man her finne at flygelederne er mye strengere i deres kvalitetskrav enn det øvrige testpersonellet.

5.3.1.1 Valgt type tale

I dette prosjektet vil ATC-tale brukes i testopplegget, da det er flygeledere som skal bedømme talen. Talen bør være kort, informative og halv-dupleks¹⁴², da er det mest naturlige, i tillegg til det nevnt i diskusjonen over, å velge realistisk ATC-tale. Videre er det slik type konversasjon IP-telefoni skal brukes til i ATC-sammenheng. Dette er også i henhold til nevnte diskusjon om å skape mest mulig realistiske forsøk utfra hvordan flygelederens daglige arbeid arter seg. Videre vil flygelederne, som skal bedømme talen, være klar over at dette prosjektet vil *simulere* en ATC-situasjon.

¹⁴² En prater adgangen.

5.3.2 Talemateriale

Et interessant spørsmål er hva slags talemateriale som skal benyttes. Annen forskning har benyttet utdrag fra for eksempel studentaviser (i lyttekvalitetstestene i [WaS97]) eller annen ikke-teknisk litteratur. [WaS97] har i sine forståelighetstester brukt såkalt "Harvard Phonetically Balanced"-ord, [HHSW95] har i sine subjektive tester også gjort bruk av fonetiske balanserte ord, mens andre igjen har brukt såkalte "Rhyme Test" (rimetester) [Voi83]. Men siden det, som stadfestet tidligere, stilles krav til at ATC-talen skal være "kort, lett forståelig og i en dagligdags tone" bør det antageligvis tilstrebes å bruke en tale som ligger så nærme vanlig tale som mulig. Dermed utelukker de to siste av de overnevnte metodene seg selv, da de blir litt vel kunstige. Denne tankegangen vil være gjeldende for såvel lytttematerialet i lytte tester som i konversasjonstester.

For konversasjonstester, blir et sentralt spørsmål: Hva skal drive konversasjonen? Med flygeledere som deltagere i et eksperiment, er det i henhold til diskusjonen ovenfor helt naturlig å bruke ATC-konversasjon. I testsituasjoner hvor ikke kun flygeledere deltar, blir spørsmålet verre å svare på. Skal man følge rådene og bruke prospektlignende kort, slik det er gitt eksempler på i appendiks B, for å stimulere konversasjonen i henhold til P.800? Problemet blir da at talen antagelig ikke blir noe i nærheten av slik autentisk ATC-kommunikasjon vil arte seg. Sannsynligvis vil man ikke få like korte og svært informative setninger slik som i ATC, men heller en del lengre og i dagligdags form. Muligens vil prating i munnen på hverandre også kunne forekomme. Dette vil ikke være tilfelle i ATC-tale, da denne er "pusht-to-talk" og derigjennom halv-dupleks.

Et annet alternativt er å ha et ferdig manus bestående korte setninger (med autentiske ATC-fraser når flygeledere deltar), som deltagerne leser opp for hverandre slik som i skuespilløvelser. Et potensielt problem her, om deltagerne skal bedømme kvalitet *samtidig* som de deltar i replikkvekslingen, kan være at deltagerne eventuelt kan bli mer opptatt med replikkvekslingen enn å fokusere på talekvaliteten. [WaS96] sier at en ulempe med tester som bruker kort tale, er at den rett og slett er for kort til å gi et reelt bilde av konversasjonskvaliteten. Men man må da huske på at målsetningen med VoIP-basert ATC-konversasjon kun er å holde en akseptabel talekvalitet i perioder ikke lengre enn kanskje et par minutter. Valgene som har blitt tatt i dette prosjektet med hensyn til talemateriale, vil bli omtalt i delkapittel 5.4 på bakgrunn av diskusjonene og tilhørende valg i de påfølgende delkapitlene.

5.3.3 Språk

Ved bruk av ATC-tale vil engelsk være det naturligste språket, fordi "International Civil Aviation Organization" (ICAO) anbefaler at ATC-tale bør foregå på engelsk [ICAO95]. Dessuten vil dette også være i henhold til diskusjonen over om å gjøre eksperimentene så realistiske som mulig. Men, for ikke-ATC tale, skal man da benytte tale på engelsk fordi ATC reelt sett ville foregå på det språket? Eller, om norsktalende personer vil delta i testene, bør talen heller være på norsk? Begrunnelsen for det siste kan være at det bør være et mål i seg selv i størst mulig grad å unngå forstyrrende faktorer, som tar fokuset til testpersonene vekk fra selve bedømmelsen. I en konversasjonstest, vil dette kunne være særlig relevant. Hovedsakelig fordi det kan være at ikke alle deltagerne er like komfortable med å prate på engelsk. For slike

personer kan det å prate på engelsk i en slik testsammenheng være svært forstyrrende, fordi det ville være menneskelig å ønske og ha best mulig uttale. Kanskje vil noen også vegre seg for å prate, og man ender opp med en tale som ikke blir naturlig. P.800 og "Handbook on Telephonometry" legger til en viss grad opp til en selektiv utvelgelse av testpersonene, som selvsagt bør følges for å få testene mest mulige i henhold til standardene. Men det kan bli litt for selektivt å aktivt forhøre seg om forsøkspersonenes engelskkunnskaper.

5.3.3.1 Prosjektets språkvalg

Det har blitt valgt å benytte engelsk i talen som utveksles i eksperimentoppsettet. Dette strider mot diskusjonen om at bruk av engelsk kan skape en unaturlig tale og en viss snakkevegring, da også vanlige mennesker vil delta i forsøksoppsettet. Denne avgjørelsen har likevel blitt tatt, på bakgrunn av at det har blitt besluttet å benytte ATC-tale. Dermed blir det også riktigst å følge anbefalingen fra ICAO.

Om engelskkunnskapene blant de "vanlige" folkene som deltar i testene ikke holder helt Oxford-standard, skulle ikke det være kritisk for testresultatene. Det kan være mulig at flygeledere er mer vant til å bruke engelsk i det daglige enn mange andre, men det er likevel det er lite trolig at deres språkkunnskaper er så overlegne at andre deltagendes eventuelle mangelfulle ferdigheter vil virke forstyrrende på testingen av talekvaliteten. Dessuten vil det være varierende engelskkunnskaper blant såvel flygeledere som andre folk. Videre er det slettes ikke sikkert at flygelederne i deres daglige arbeid kun møter perfekt uttalt engelsk fra alle pilotene de er i kontakt med.

På grunn av at talen foregår på engelsk har det også blitt besluttet at skalaene (valg av skala(er) blir diskutert nærmere i neste delkapitel) i eksperimentoppsettet også skal være på engelsk. Slik unngår man også oversettingsproblematikken med "Listening-quality"-skalaene som ble presentert i kritikken mot P.800 i appendiks B.1.3.

5.3.4 Aktuelle typer tester

Det kan eventuelt diskuteres hvor mye mer relevant konversasjonstester er enn lyttetester, da ATC-kommunikasjonen er halv-dupleks (som beskrevet overfor). I prinsippet vil en slik kommunikasjon utarte seg slik: den ene prater, den andre lytter, den andre prater og den første lytter og så videre. I og med at slik kommunikasjon ikke arter seg som en vanlig telefonkonversasjon, kan kanskje lyttetester være tilstrekkelige for å få simulert og bedømt ATC-kommunikasjon.

Et alternativ til lytte- og konversasjonstester er blant annet forståelighetstester¹⁴³ hvor deltagerne enten skriver ned hva man har hørt eller ringer rundt nøkkelord man mener å ha hørt. I etterkant bedømmer testpersonellet i hvilken grad testdeltagerne virkelig forsto det som ble sagt. Det å teste om kvaliteten er god, innebærer her en antagelse om at testpersonellet vet hva som trenges av kvalitet for at talen skal være forståelig. For talekvalitetstester i ATC-sammenheng, er det kun flygeledere som er i stand til å eksakt definere dette.

I en testsituasjon hvor testpersoner uten ATC-bakgrunn også inngår, kan det diskuteres om det er bedre å teste om det som sies blir forstått enn å teste selve

¹⁴³ Omtales mer i appendiks B.

kvaliteten. Med bakgrunn i den antagelsen om at de ikke vet hva slags kvalitet man jakter på. Slik får man i alle fall testet om talen ble oppfattet riktig, som er det viktigste kravet i ATC-kommunikasjon. Det kan da diskuteres hvorvidt lytte- og konversasjonstester rett og slett er uegnet, særlig når testpersonene likevel ikke vet hva de skal lytte etter.

5.3.5 Oppsummering av diskusjonen

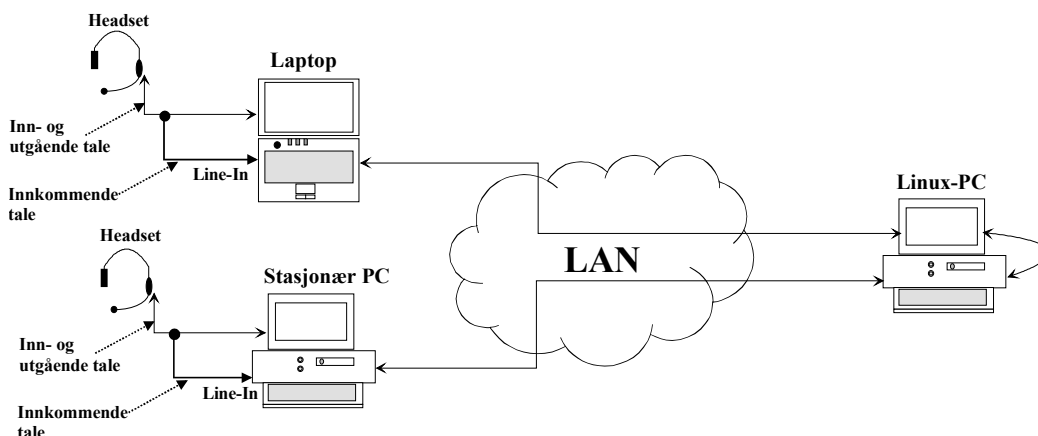
Diskusjonen vil i dette delkapitlet bli kort oppsummert med fokus på fordeler og ulemper med de ulike momentene som ble tatt opp.

		Fordeler	Ulemper
Type tale:	ATC-tale	Økt realisme i testoppsettet når flygeledere deltar i forsøksoppsettet.	Kan skape forvirring, da testomgivelsene avviker fra flygeledernes reelle arbeidssituasjon.
	Vanlig tale	Fordelaktig når vanlige personer deltar i oppsettet, da disse ikke er familiær med ATC-tale.	Uheldig når også flygeledere deltar, mister noe av realismen i oppsettet. Talen blir heller ikke slik VoIP er ment å fungere under i ATC.
Talemateriale:	Utdrag fra aviser, bruk av prospekt kort og lignende	Benyttet i en del annen forskning og nevnt i diverse ITU-anbefalinger og råd.	Tilfredsstillende ikke kravene til hvordan ATC-tale skal arte seg.
	Manus med ATC-fraser	Det er slik type tale som vil bli brukt i reell VoIP-basert ATC.	Deltagerne kan bli for fokusert på replikkvekslingen i en situasjon hvor de <i>samtidig</i> skal bedømme talekvaliteten.
Språk:	Engelsk	Naturligst valg når ATC-tale benyttes, siden ICAO anbefaler dette. Gjør også forsøkene mest mulig realistisk.	Kan virke forstyrrende om de vanlige menneskene som deltar i forsøkene ikke er komfortabel med å snakke engelsk.
	Norsk	Med norsktalende personer deltagende i forsøkene vil dette fjerne eventuelle forstyrrende faktorer ved å snakke på engelsk.	Ikke et naturlig valg ved bruk av ATC-tale, da ICAO anbefaler bruk av engelsk. Realismen blir borte.
Aktuelle typer tester:	Lytt- og konversasjonstester	Mest egnet når flygeledere skal delta i bedømmelsene, da disse vet hva som kreves av kvalitet.	Uegnet når ikke-ATC folk skal bedømme kvaliteten, da disse ikke vet hvilke kvalitetskrav som stilles til ATC-tale.
	Forståelsestester	Tester det som er viktigst i ATC, nemlig at talen forstås. Fordelaktig når vanlige folk deltar, i og med at de ikke vet hvilken kvalitet som tilfredsstillende ATC-kravene.	Skal avgjørelsene i etterkant om talen ble forstått gi mening, må flygeledere gjøre dette. Således må det rekrutteres både vanlig- og ATC-folk.

Tabell 5.2: Oppsummering av diskusjonen med hovedvekt på fordeler og ulemper.

5.4 Valg av eksperimentoppsett

I dette prosjektet vil ikke alle hensynene i ITU sin P.800 og ”Handbook on Telephony” (disse er fylldig omtalt i appendiks B) om subjektive lyttetester følges slavisk. Disse har, derimot, dannet en faglig bakgrunn for et eksperimentoppsett som er tilpasset det ATC-scenariet disse eksperimentene vil ta utgangspunkt i. På andre områder igjen vil standardene i større grad følges, særlig med hensyn til layout på evalueringsskjemaene.



Figur 5.2: I eksperimentoppsettet som er valgt, vil IP-telefonitalen bli sendt over et LAN, via en Linux-PC, til mottageren. På Linux-PC'en vil det kjøres et program som gjør det mulig å manipulere IP-pakkene med hensyn på pakketap og -forsinkelse. I hver ende vil innkommende tale bli forgrenet og ført til inngangen på PC-ens lydkort (line-in), slik at lyden som høres i headset-et også blir tatt opp digitalt. Opptakene blir deretter lagt ut på Internett og lastet ned av flygeledere som bedømmer talen.

Det blitt avgjort å satse på eksperimentoppsettet vist i figur 5.2. Her vil to personer delta i en replikkutveksling utfra et manus basert på engelsk ATC-tale. Disse to personene er ikke flygeledere og har heller ingen tidligere erfaring med luftfartskommunikasjon. Innholdet i dette manuset er hentet fra eksempler på ATC-replikker gitt i ”Radiotelephony Manual” [CAA96], på ”The Federal Aviation Administration” (FAA¹⁴⁴) sin webside om ”Radio Communications Phraseology and Techniques”¹⁴⁵ og fra en gresk flygeleders webside¹⁴⁶ om opplæring i ATC. Et eksempel på slik replikkveksling¹⁴⁷ er:

A:	Hello Cairo, copy estimate over Kilo Uniform Mike Bravo India.
B:	Go ahead.
A:	Mike Sierra Romeo eight eight zero.
B:	Mike Sierra Romeo eight eight zero, Airbus three zero zero, destination Cairo.
A:	Correct, estimation Kilo Uniform Mike Bravo India at two three zero six, Flight Level two niner zero.
B:	Copied, over Kilo Uniform Mike Bravo India two three zero six, Flight Level two niner zero.

¹⁴⁴ Amerikanske luftfartsmyndigheter.

¹⁴⁵ Ikke trykket materiale, se <http://www.faa.gov/ATPubs/AIM/Chap4/aim0402.html>.

¹⁴⁶ http://users.hellasnet.gr/kpatour/General_knowledge/_Toc487237291/

¹⁴⁷ Manuset brukt i testene er i sin helhet gjengitt i appendiks G.

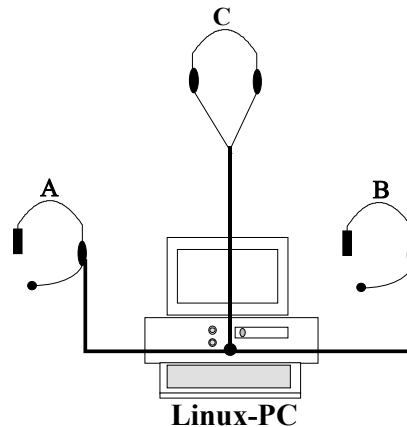
IP-telefonitalen blir sendt over et LAN og via en Linux-PC på vei til mottageren. I Linux-maskinen vil et program som har blitt utviklet i dette prosjektet, *dgramServer*, kjøre. Dette programmet gjør det mulig å simulere pakkeap og pakkeforsinkelse som talepakkene kan bli utsatt for ved traversering av et IP-nettverk. Ved å manipulere med forsinkelser får man simulert forsendelse av pakker over ulike avstander. Programmet gjør det også mulig å omstokke talepakkene for å simulere den omstokkingen IP-pakker kan bli utsatt for i et IP-nettverk. I kapittel 6 blir det gitt mer informasjon om *dgramServer*.

I hver ende ble innkommende tale, altså det samtalepartneren uttalte, forgrenet slik at lyden som høres i høytaleren på headset-et også ble sent til line-in på PC-ens lydkort for digitalt opptak. Hensikten var å slå sammen lyden som ble tatt opp i hver ende til én stereo lydfil. I denne lydfilen ble talen fra den ene parten lagt i det venstre lydsporet og lyden fra den andre i det høyre lydsporet. Dette ble gjort for å illustrere at man overhører en samtale, hvor man sitter midt mellom dem som prater. Disse lydfilene ble lagt ut på en hjemmeside for dette prosjektet, hvor det ble gitt et kort sammendrag av prosjektet i tillegg til at flygeledere kunne laste ned både lydfiler og evalueringsskjema. Skjemaene ble returnert via e-post. Mer informasjon om organisering av lydfilene blir gitt i kapittel 6.4.

Alternativt var det planlagt å bruke eksperimentoppsettet vist i figur 5.3. I dette oppsettet ville det for hver testrunde bli benyttet to "vanlige" personer (A og B) og én flygeleder (C). Det forenklete bildet i figur 5.3 tar ikke hensyn til en del tekniske detaljer som opplagt må være tilstede for at talen skal komme frem til mottageren. Dette eksperimentoppsettet ble forkastet, da det blant annet ble funnet ut at det ville være mer tidseffektivt å ta opp all tale i løpet av én testsesjon og deretter distribuere disse opptakene rundt til flygeledere, enn å arrangere flere sesjoner med flygeledere og testpersoner til stede. Det valgte oppsettet gjorde det også mulig å få inn evalueringer hurtigere og ikke minst åpnet det for å få inn *flere* evalueringer i løpet av samme eller kortere tid sammenlignet med det alternative oppsettet. Kanskje viktigere var det at man med det valgte oppsettet unngikk at flygeledere måtte være tilgjengelig under forsøkene, samt at forsøkene måtte gjentas for hver ny flygeleder som skulle lytte. Med en slik fremgangsmåte ville ikke forsøkene være 100% repetérbare og muligens ville testpersonene lese bedre og bedre med tiden, hvilket kunne påvirke testresultatene. Dette kunne selvfølgelig unngås ved at det ble brukt forskjellige manus for hver av testsesjonene, men da ville man ha mistet sammenligningsgrunnlag for forsøkene.

Meningen var at A og B skulle lese opp for hverandre fra et utlevert manus basert på ATC-tale. Dette manuset skulle inneholde replikker på engelsk som ville bli utvekslet slik som i det valgte eksperimentoppsettet (beskrevet på forrige side). Videre kunne B i manuset bli bedt om å gjenta hva A sa, uten å ha det skriftlig hva A sin replikk var. A ville i sitt evalueringsskjema ha et avkrysningsfelt for om B gjentok korrekt og dermed forstod hva A sa. Dette ville således sjekke forståeligheten. En klar ulempe med dette kunne likevel være at testene ville bli sårbar for manglende konsentrasjon, som i verste fall kunne gi feil gjentakelser fordi det ikke ble fulgt med. Uansett, fremgangsmåten er i henhold til hvordan ATC-konversasjon utarter i og med at parametre skulle verifiseres ovenfor motparten. Man kunne også variere på hvem som skulle gjenta siste replikk. Ideen var at C skulle være analogt koblet til A og B. Slik ville C høre både hva A hører (når B prater) og hva B hører (når A prater). På denne

måten kunne C overvåke replikkvekslingen og som profesjonell bedømme hvorvidt talekvaliteten på replikkvekslingen er tilstrekkelig for ATC-bruk.



Figur 5.3: Et forenklet bilde av det alternative eksperimentoppsettet. A og B (vanlige mennesker) deltar i en replikkutveksling og kan i manuset bli bedt om å gjenta hva samtalepartneren sa for å sjekke forståeligheten. Det var også muligheter for å be A og B bedømme talekvaliteten slik at deres bedømmelser kunne bli sammenlignet med C (flygeleder) sine. C ville være analogt tilkoblet A og B, og kunne på denne måten overhøre konversasjonen. IP-telefonitalen skulle gå over et LAN via en Linux-maskin, hvor manipulering på talepakkene skulle utføres.

Ved å også be A og B bedømme talen kunne man eventuelt få sammenliknet kvalitetsbedømmelsene til C med A og B – om dette skulle være ønskelig. For å kunne sammenlikne forståeligheten til C med A og B, ble det tenkt at vedkommende i noen eksperimenterunder kunne bli bedt om å skrive ned hva han/hun hørte. Alternativt så man også for seg å ta opp det C hørte, slik at det i ettertid ville være mulig å sjekke det C hørte oppmot det som ble sagt av enten A eller B i henhold til manus. Dette kunne også åpne for avspilling av den eksakte talen for et større antall flygeledere for å hypotesetestede bedømmelsene fra den ene flygelederen.

I det alternative eksperimentoppsettet ville man også være sårbar for eventuelle feil som kunne oppstå i og med at man sannsynligvis kun hadde en flygeleder tilgjengelig for en kort periode adgangen. Slik kunne testingen unødig bli heftet ved at dyrbar tid ble brukt til feilretting og man eventuelt måtte vente med å fullføre eksperimentene til neste gang en flygeleder hadde anledning. Resultatmessig gir det valgte eksperimentoppsettet også mest mening, da man får anledning til å få inn flere bedømmelser fra flygeledere. Det alternative oppsettet tok også utgangspunkt i bedømmelser fra vanlige personene. I ettertid gir egentlig ikke dette så mye mening når IP-telefonitalen skal testes i ATC-sammenheng, fordi det er kun flygelederne som vet hvilken kvalitet den bør holde. Videre, gir det heller ikke så mye mening å sammenlikne resultatene fra A og B med C, fordi referansene er totalt forskjellige. A og B vil antagelig sammenlikne talekvaliteten med kvaliteten på vanlig fastlinje-telefoni, mens referansen det er ment å teste oppmot i dette prosjektet er det eksisterende kommunikasjonssystemet som flygelederne bruker i deres daglige arbeid.

Det var tiltenkt å gjøre opptak av det C hørte og la flere flygeledere høre på dette for å få et bredere resultatgrunnlag. Dette oppnås også i det valgte eksperimentoppsettet, men mye mer *effektivt*. På lik linje med det alternative testoppsettet, var det valgte også basert på replikkutveksling. Dette oppsettet tillot imidlertid ikke funksjonaliteten

i det alternative oppsettet med å la A og B fylle inn manglende ord i manus for å teste forståeligheten. I stedet for vil flygelederne bli bedt om å si ja eller nei til om kvaliteten på IP-telefonien holdt en kvalitet tilstrekkelig for ATC-bruk. Dette er egentlig viktigere og utgjør mye av hensikten med testingene i dette prosjektet. Utover dette, vil evalueringsskjemaet være det samme som tiltenkt i det alternative oppsettet. Evalueringsskjemaet vil omtales mer i delkapittel 5.4.2.

Måten lydfilene er laget på, med lyden av den ene parten i venstre lydkanal og den andre i høyre, ble gjort for å simulere den alternative ideen om at C skulle være analogt tilkoblet A og B (jamfør figur 5.3). Slik sett oppnås akkurat det samme med det valgte oppsettet, men uten at man er sårbar for eventuelle feil som kunne oppstå under flygeledernes lytting. Om lyttingen skjer "live" eller i opptak har ikke noe å si for bedømmelsene, så lenge opptakene gjenspeiler kvaliteten som ble hørt i headsetene.

Det valgte eksperimentoppsettet åpner dessuten for en mulighet å eventuelt på sikt etablere en webressurs for talekvalitetstesting av VoIP-basert ATC, hvor interesserte kan lese mer om dette og laste ned taleopptakene som ble gjort. Intensjonen er uansett at rapporten sammen med lydopptakene skal legges ut offentlig på Universitetet i Oslo sine biblioteksider på Internett, slik at interesserte kan få ekstra utbytte av rapporten. Slik kan det være mulig å høre effekten av for eksempel 20% pakketap i tillegg til å lese hvilke bedømmelser dette fikk.

5.4.1 Opptaks- og avspillingsrekkefølge

Hovedsakelig ble det vurdert fire alternativer for hvordan lydopptakene skulle organiseres med hensyn på opptaks- og avspillingrekkefølge og variasjon av manus (jamfør figur 5.4):

1. Ett manus

Om man hadde brukt *ett* manus for alle opptakene av alle parametrene, kunne det ved avspilling være en fare for at flygelederne kunne lære seg manus utenat. Det som da kunne skje er at hukommelsen "fyller inn" ord som eventuelt mangler i talen og hjernen oppfatter den som komplett. En annen ulempe vil også kunne være at opptakene blir utsatt for om en deltager i testopplegget har vanskeligheter med å si et ord eller liknende. Dette vil jo gå igjen. Videre kan det kanskje kan bli litt kjedelig å lytte til i lengden. En fordel med å ha ett manus for alle opptakene, kan likevel være at slik vil ikke forandring i innholdet av manus har noen påvirkning.

a)

		Opptak nummer				
		1	2	3	4	5
Parameter	0%	A	A	A	A	A
	5%	B	B	B	B	B
	10%	C	C	C	C	C
	15%	D	D	D	D	D
	20%	E	E	E	E	E
	25%	F	F	F	F	F

b)

		Opptak nummer				
		1	2	3	4	5
Parameter	0%	A	B	C	D	E
	5%	A	B	C	D	E
	10%	A	B	C	D	E
	15%	A	B	C	D	E
	20%	A	B	C	D	E
	25%	A	B	C	D	E

Figur 5.4: Hver av bokstavene i figurene symboliserer ett unikt manus. I a) var det tenkt på to alternativer for hvordan opptakene og avspillingen skulle organiseres: (1) horisontal organisering av både opptak og avspilling, eller (2) horisontal organisering av opptak og avspilling orientert i serier vertikalt i figuren. Et tredje alternativ var organisering av både opptak og avspilling i serier som vist vertikalt i b).

2. Horisontal organisering

Alle fem opptakene av *én* parameter organiseres horisontalt i figur 5.4a, når det gjelder både opptaks- og avspillingrekkefølge. På denne måten kan det være noenlunde sikkert at alle fem opptakene har blitt utsatt for samme nettverksforhold, i og med at de alle er tatt opp rett etter hverandre over en kort tidsperiode. Avspillingen skjer også i denne rekkefølgen. Flygelederne hører *alle* taleprøvene av 0% og bedømmer hver og en av dem, før de går videre med å høre taleprøvene for 10% osv. Med fremgangsmåten i dette punktet, kan det begrenses at flygelederne blir lei av å høre samme manus flere titalls ganger og at hjernen komplimenterer talen. Allikevel kan det hende at for enkelte lyttere, vil det å høre samme manus fem ganger fortone seg som kjedelig. En ulempe kan muligens være at hjernen også begynner å huske innholdet i talen etter å ha hørt tre til fire opptak med samme manus.

3. Vertikal organisering

Opptakene og avlyttingene organiseres i serier vertikalt i figur 5.4b, hvor samme manus brukes i alle opptakene av hver serie. I en slik organisering vil det kunne være sannsynlig at nettverksforholdene for hvert opptak i en serie har vært noenlunde like. Her oppnås det samme som i tilfelle 2, bortsett fra at det antagelig oppnås et sammenlikningsgrunnlag som ikke er mulig i tilfelle 2. Dette sammenlikningsgrunnlaget oppstår ved at den samme talen høres for gradvis forverrende forhold. På denne måten vil det kanskje være lettere for hjernen å tenke "var denne bedre/verre enn forrige?". I et oppsett hvor bedømmelsene skjer etter avspilling av *alle* opptakene, burde det verste forholdet bli avspilt først. Grunnen til dette er at hjernen har en tendens til å legge mest vekt på det den hører sist, slik som [BSD01] nevner.

Horisontal organisering i figur 5.4b for avspilling ble aldri vurdert som et alternativ, da det ikke ble funnet noen spesielle fordeler med at lytteren hører fem forskjellige manus for én parameter. Det å ta opp fem opptak med fem forskjellige manus ville også ha gjort opptakene ekstra tungvinte.

4. Horisontal opptaksrekkefølge og vertikal avspilling

Opptakene organiseres horisontalt i figur 5.4a, slik som i tilfelle 2), men hvor avspillingen orienteres vertikalt i figuren. På denne måten høres fem serier med A-B-C-D-E-F, hvor hver bokstav angir et unikt manus. I et slikt opplegg er det ikke helt sikkert at for eksempel 0% og 10% har blitt tatt opp under helt like nettverksforhold, da disse to parametrene har blitt tatt opp over en lengre tidsperiode. Denne organiseringen høyner i alle fall sannsynligheten for at nettverksforholdene innbyrdes for alle opptakene av én parameter har vært noenlunde like. Med dette menes at det ikke skal være store forskjeller i talekvaliteten fra ett opptak med for eksempel 10% pakkeap til et annet. Likevel, når talen i dette prosjektet ble sendt over et 10 Mbps-nettverk på en stille søndags ettermiddag, kan det antas at alle opptakene møtte noenlunde like nettverksforhold.

Den største gevinsten med dette alternativet, er at lyttingen blir mer variert enn alternativ 3 og 4, og at bedømmelsene blir mer uavhengig av hva lytterne tidligere har hørt. Innad i hver serie kan rekkefølgen på opptakene varieres. På denne måten kan man unngå at lytterne blir vant til en kvalitet¹⁴⁸ og begynner å

¹⁴⁸ Slik som i alternativ 1, hvor lytteren hører fem like opptak etter hverandre.

bedømme talekvaliteten av ”gammel vane”. I en slik situasjon kan det være en fare for at lytterne kan bedømme kvaliteten uten å egentlig ha lyttet til opptakene. Når opptakene høres i serier og rekkefølgen er variert, blir det ikke så lett å danne seg et bilde av et visst mønster i opptakene. På denne måten blir kanskje bedømmelsene riktigst i forhold til hvordan talekvaliteten faktisk opplevdes øyeblikkelig etter avspilling. En annen mulig fordel med denne organiseringen kontra tilfelle 2, kan være at man oppnår et sammenlikningsgrunnlag i avspillingen mellom hvert opptak. Lytterne kan spørre seg om et opptak var bedre eller verre enn forrige i og med opptakene i seriene holder varierende kvalitet.

På grunn av de nevnte fordelene ble dette alternativet valgt.

5.4.2 Valg av skalaer

Etter at hvert opptak har blitt avspilt, skal den enkelte flygelederen subjektivt bedømme talekvaliteten ved hjelp av et evalueringsskjema. Dette vil inneholde skalaer definert i P.800. Hvilke skalaer det skal benyttes, vil bli diskutert i det følgende.

Konversasjonen som flygelederen overhører er en halv-dupleks replikkutveksling. Dermed er det også nødvendig å benytte en variant av en lytteskala. I delkapitlet 6.2 ble det slått fast at det er et absolutt krav for ATC at talen kommer frem og at kvaliteten skal være slik at det ikke er noen tvil om hva som blir sagt. Altså, blir forståeligheten på talen en viktig medvirkende faktor under bedømmelsene. Slik sett kan det argumenteres at talekvaliteten ikke behøver å være krystallklar og feilfri (med hensyn på blant annet forstyrrelser og ekko), så lenge det som blir sagt blir oppfattet av mottageren. Det er dermed ikke sikkert at det blir like viktige hvor feilfri talen er, som i for eksempel en telefonsamtale, hvor kundene ønsker en mest mulig ”komfortabel” konversasjon for pengene. Derfor vil trolig ikke ”Listening-quality”-skalaen (gjennomsnittsverdien kalles ”Mean Opinion Score” (MOS), jamfør tabell 5.3) alene gi et riktig bilde av hvor egnet talekvaliteten er for ATC. ”Listening-effort”-skalaen (gjennomsnittsverdien kalles her ”Mean Listening-effort Opinion Score”¹⁴⁹ og forkortes MOS_{LE}) sier derimot noe om hvor mye innsats som må til for å forstå meningen med talen, men ikke noe om kvaliteten på talen.

”Listening-effort”		”Listening-quality”	
Effort required to understand the meanings of the sentences.	Score (MOS _{LE})	Quality of the Speech.	Score (MOS)
Complete relaxation possible; no effort required.	5	Excellent	5
Attention necessary; no appreciable effort required.	4	Good	4
Moderate Effort required.	3	Fair	3
Considerable effort required.	2	Poor	2
No meaning understood with any feasible effort.	1	Bad	1

Tabell 5.3: ”Listening-effort”- og ”Listening-quality”-skalaen definert i ITU P.800-standarden [ITU96]. Gjennomsnittsverdiene heter henholdsvis ”Mean Listening-Effort Opinion Score” (MOS_{LE}) og ”Mean Opinion Score” (MOS).

¹⁴⁹ Både ”Listening-quality”- og ”Listening-effort”-skalaen inngår i såkalte ”Listening Opinion”-tester. Dette er mer beskrevet i appendiks B.1.2.

Derfor vil en kombinasjon av begge være heldig, da man for en testparameter får svar på hvilken kvalitet talen holdt, samtidig som man får vite hvor mye lytteinnsats som måtte til for den konfigurasjonen. Slik er det også mulig å få svar på om ”Listening-quality”-skalaen virkelig gir et riktig bilde av hvor egnet talekvaliteten er for ATC, hvilket det i diskusjonen på forrige side ble sådd tvil om. Om man for eksempel får et resultat med kvalitetsverdi (MOS) lik 3 og lytteinnsatsverdi (MOS_{LE})¹⁵⁰ lik 5, indikerer dette at denne skalaen kanskje ikke gir det. Grunnen til dette er at flygelederne i så fall bedømmer talekvaliteten dit hen at det *ikke* kreves noen form for lytteinnsats for å forstå meningen med talen, lytteinnsatsverdi (MOS_{LE}) lik 5, *samtidig* som de bedømmer lyttekvaliteten til å være kun *middels*, kvalitetsverdi (MOS) lik 3 (jæmfør tabell 5.3). I og med at det ikke krevdes noen lytteinnsats, ville det ha vært naturlig å vente at talen også ville få toppkarakterer på ”Listening-quality”-skalaen. Dette gir en logisk motsetning i bedømmelsene som dermed kan antyde at den *reelle* talekvaliteten som flygelederne hørte er bedre enn det resultatet for kvalitet (MOS) og derigjennom ”Listening-quality”-skalaen indikerte. I så fall kan dette tyde på at ”Listening-quality”-skalaen ikke gir et riktig bilde av hvor egnet talekvaliteten er for ATC og at det i fremtidige tester bør vurderes å benytte andre skalaer enn ”Listening-quality”-skalaen. Tilsvarende kan en lavere lytteinnsatsverdi (MOS_{LE}) enn kvalitetsverdi (MOS) indikere at den reelle talekvaliteten er dårligere enn det kvalitetsverdien (MOS) tilsier.

Selvom flygelederne har den faglige kompetansen til å benytte ATC-skalaen i tabell 5.1, vil det ikke være hensiktsmessig å bruke den. Grunnen er at det vil være fordelaktig å følge ITU sine standardiserte skalaer for subjektiv bedømmelse av talekvalitet, siden det er de som er rådende på feltet og har vitenskapelig anerkjennelse.

Det vil også være plass i evalueringsskjemaet til å komme med kommentarer til lydopptakene, men dette er frivillig. Slike kommentarer vil kunne gi et bilde av hvordan flygelederne tenkte under bedømmelsene og kan muligens også gi en del interessante sitater som kan illustrere eller utdype enkelte funn i resultatene.

Evalueringsskjemaet brukt i bedømmelsene ligger vedlagt i Appendiks F.

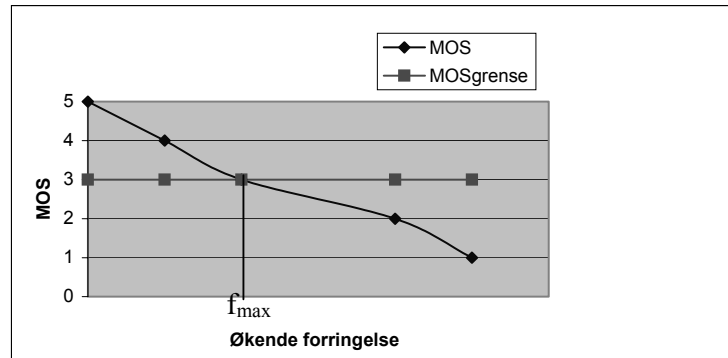
5.5 Målet med eksperimentene

Det har blitt foretatt en del subjektive testinger av talekvalitet i IP-telefoni tidligere. Det som gjør eksperimentene i dette prosjektet helt unike, er at det ifølge vår kjennskap ikke tidligere har blitt gjort forskning på bruk av IP-telefoni i ATC-sammenheng med flygeledere som profesjonelt og systematisk kan bedømme om talekvaliteten holder mål i ATC-bruk.

Målet med eksperimentene er å få frem grafer som viser bedømmelsene som funksjon av gradvis økende forringelse av den enkelte parameteren det testes på. Et eksempel på dette med kvalitetsverdier (MOS) er gitt i graf 5.1. Hovedresultatene man ønsker å finne er krysningspunktet mellom nedre grense for godkjent kvalitet, kalt MOS_{grense} i

¹⁵⁰ For å enklere skille mellom MOS- og MOS_{LE} -verdier, vil disse i det følgende heholdsvis benevnes kvalitetsverdi (MOS) og lytteinnsatsverdi (MOS_{LE}).

graf 5.1, og nevnte kurve. Krysningpunktet vil gi svarene på hvilken konfigurasjon av parametrene, maksimal forringelse (f_{\max}), som gir MOS_{grense} . Ut fra skalaene i tabell 5.2 og kravene til forståelighet vil det være grunn til å anta at $MOS_{\text{grense}}=3$.



Graf 5.1: Et eksempel på en graf basert på testresultater. Krysningpunktet viser den maksimale forringelsen, f_{\max} , som gir den nedre grensen for godkjent talekvalitet, MOS_{grense} .

De parametrene som i utgangspunktet kunne være aktuelle å teste på, er jitter, pakkeforsinkelser og -tap. Det er nettopp disse parametrene annen forskning har testet på. [WaS97] testet på pakketap og opererte med 0, 10, 20..50 % pakketap. [HHM99] testet på jitter med 0, 20, 40, 60..160 ms innbyrdes forsinkelse mellom pakkene. I dette prosjektet vil det antas at jitter vil ha samme effekt på lytterne som pakketap. Grunnen til dette, er at når pakker med innbyrdes ulike forsinkelser¹⁵¹ større enn buffringstiden i jitterbuffret vil bli de forkastet fordi de er så forsinket at de ikke kan brukes til å gjenskape talen. Dermed kan det sies at jitter har samme påvirkning på talen testpersonalet hører som pakketap, og i denne testsammenhengen vil vi av avgrensningshensyn kun konsentrere oss om pakketap og -forsinkelser.

¹⁵¹ Jamfør definisjonen av jitter i kapittel 3.

Kapittel 6

Implementasjon og testoppsett

I dette kapitlet skal implementasjonen av det socket-baserte serverprogrammet *dgramServer* i programmeringsspråket C belyses. Serveren ble benyttet i talekvalitetstesting av "Air Traffic Control" (ATC)-tale for å manipulere IP-pakkene som frakter talen. Temaer som skal berøres angående implementasjonen er hvordan serveren fungerer, hvordan den er bygget opp og hvordan den har blitt benyttet i talekvalitetstesting.

Videre skal det valgte testoppsettet og -plattformen belyses. Temaer som vil bli tatt opp her, vil blant annet være beskrivelser av benyttete datamaskiner, den fysiske audiooppkobling og hvordan talen ble tatt opp.

6.1 Serverens oppbygning og funksjonaliteter

Socket-serveren, *dgramServer*, er utviklet i henhold til *socket*- og *pthread*-bibliotekene i POSIX-basert Linux. For mer informasjon om funksjoner hentet fra disse bibliotekene som nevnes i det følgende eller som har blitt brukt i kildekoden, henvises det til "The GNU C Library"¹⁵² eller manualsidene¹⁵³ i Linux med mindre annet er oppgitt.

En "socket" er et endepunkt for en interprosess kommunikasjonskanal [Bes87]¹⁵⁴. Denne kommunikasjonen foregår mellom to prosesser ved at hver prosess etablerer hver sin socket og sender informasjonen mellom disse. Dette forutsetter at socketene er av samme type og i samme adressedomene [Sock02]¹⁵⁵. Det finnes to mye brukte adressedomener, Unix domene¹⁵⁶ og Internett domene [Sock02]. Videre er "stream socket" og "datagram socket" de mest brukte socket-typene [Sock02]. "Stream

¹⁵² Ikke trykket materiale. Hentet fra <http://www.gnu.org/manual/glibc-2.2.5/libc.html>.

¹⁵³ De fleste av disse er også tilgjengelig på Internett. Søk på "man + navn på funksjon".

¹⁵⁴ Ikke trykket materiale. Hentet fra <http://www.ifi.uio.no/inf103/Litteratur/UNIXsockets.pdf>.

¹⁵⁵ Ikke trykket materiale. Hentet fra <http://www.cs.rpi.edu/courses/sysprog/sockets/sock.html>.

¹⁵⁶ To prosesser som deler et felles filsystem kommuniserer igjennom et slikt domene [Sock02]

socket” baserer seg på TCP og er forbindelsesorientert, mens ”datagram socket” er basert på UDP og er forbindelsesløs. Det er valgt å benytte datagram socket i serveren fordi i H.323¹⁵⁷ brukes RTP (Real-Time Protocol) til å frakte talen. RTP er, som beskrevet i appendiks A, igjen basert på UDP.

En nyetablert socket har ingen adresse [GNU96]¹⁵⁸, derfor må en socket bli tilordnet en adresse via **bind()**¹⁵⁹-funksjonen. En slik adresse består av vertsmaskinens IP-adresse pluss portnummeret serveren kjører på i vertsmaskinen [Sock02]. To sockets kan ha samme IP-adresse, såfremt de ikke deler samme portnummer. I datagram sockets får servere og klienter henholdsvis sendt og mottatt informasjon via **sendto()**- og **recvfrom()**-funksjonene. For mer inngående informasjon om sockets i Unix, se [Bes87], [GNU96], [Sock02] og [Hal01].

Serveren har blitt utviklet med det mål for øyet at den skal utøve en viss manipulasjon på IP-pakkene som frakter talen. Serveren har derfor tre ulike måter å utøve denne manipulasjonen på:

- Serveren kan simulere hvordan IP-pakker tilfeldig blir omstokket og forsinket i et reelt IP-nettverk.
- Serveren kan simulere pakketap ved at en viss prosent pakketap er konfigurert.
- Tilsvarende kan serveren simulere pakkeforsinkelse ved at en viss forsinkelse i millisekunder (ms) er konfigurert.

På kommandolinjen til programmet kan man derfor angi enten hvor mange prosent pakketap eller millisekunder forsinkelse pakkene skal utsettes for. Kommandolinjen som man starter opp programmet med, er bygget opp på følgende form:

```
dgramServer X Y
```

Her angir X hvor mange prosent pakketap og Y hvor mange millisekunder pakkeforsinkelse som sendes med som parameter til programmet. Programmet er laget slik at det er kun lov å utsette pakkene for én av disse to parametrene, derfor gis det inn 0 for den parameteren som ikke skal brukes i programmet. Simulering av omstokking av pakker i IP-nettverk kan skrues av og på ved å henholdsvis kommentere vekk eller fjerne kommentering av linjen i kildekode som tar seg av dette (jamfør Appendiks D ”Kildekode”) og compilere programmet pånytt.

Serveren oppretter to ”threads”¹⁶⁰ som utfører prosessering av IP-pakkene, én for innkommende pakker (tråd 1) og én for utgående pakker (tråd 2). Trådene etableres med **pthread_create()**-funksjonen [Pro02]¹⁶¹ innebygget i pthread-biblioteket for POSIX-basert Linux. Videre har **pthread_join()**-funksjonen blitt benyttet for å sørge for at tråd 2 ikke avbryter tråd 1, når den jobber, før den selv avgir kontrollen og visa versa. Tråd 2 har blitt gitt en høyere prioritet enn tråd 1, 99 mot 1, slik at pakkene som er klare til å bli sendt fra serveren ikke skal bli ventende for lenge.

¹⁵⁷ En viktig signaleringsprotokoll i IP-telefoni. Denne omtales grundig i appendiks A. IP-telefoni applikasjonene som ble brukt i dette prosjektet var basert på H.323.

¹⁵⁸ Ikke trykket materiale. Hentet fra <http://www.gnu.org/software/libc/libc.html>.

¹⁵⁹ Fet skrifttype vil heretter bli brukt på funksjonsnavn, navn på variabler og lignende.

¹⁶⁰ *Tråder* på norsk. Heretter vil det norske uttrykket brukes.

¹⁶¹ Ikke trykket materiale. Hentet fra http://www.planetoid.org/technical/pthreads/pthreads_man.html#BASIC.

Slik som for sockets, finnes det veldig mye informasjon om pthread på nettet. I tillegg kan kapitlet om pthread i boken ”Linux Unleashed” av Kurt Wall et al. [Wal00] anbefales for en introduksjon til temaet.

Slumptallsgeneratorene¹⁶² i `dgramServer` blir initialisert med klokken. Kvaliteten på slike kan videre være veldig variable. Det har likevel blitt valgt å stole på at kvaliteten på standardfunksjonene for dette i Linux-bibliotekene er tilstrekkelig for formålet i dette prosjektet.

Ved bruk av tråder kan det være en fare for såkalt ”race conditions” kan oppstå, i form av at to tråder ”slåss” om den samme ressurs. Slik kan det ende opp med at én tråd for eksempel prøver å skrive til variabel, samtidig som en annen leser fra den samme variabelen. Dette er *ikke* et problem i denne implementasjonen av to grunner:

1. Trådene kjører til de selv avgir kommandoen. Dette oppstår kun ved at `sendto()` mottar innkommende pakker, ved at `recvfrom()` skal sende en pakke eller når `pthread_cond_timedwait()` (denne vil bli omtalt mer i neste delkapittel) skal slippe en pakke som har blitt holdt igjen lenge nok.
2. Programmet ble kjørt på en maskin med kun én CPU.

I det følgende skal de nevnte to trådene iakttas nærmere, jamfør kildekoden i Appendiks D.

6.1.1 Tråd 1

Tråd 1 jobber på `InputPackets()`-funksjonen, som har ansvaret for å håndtere innkommende pakker ved bruk av `recvfrom()`. Denne funksjonen er blokkerende av natur, det vil si at den stopper iterasjonen i `InputPackets()` inntil det ankommer en pakke. Imens overgis kontrollen til tråd 2. (Tilsvarende vil `sendto()` i tråd 2 avgi kontrollen tilbake til tråd 1 når denne funksjonen venter på å ”koble opp” for å sende en pakke.) Kommer det, derimot, inn pakker blir det for hver innkommende pakke allokeret et minneområde for en peker til en `struct packet`. Hver innkommende blir tatt vare på i en slik `struct`.

`Struct packet` er en egendefinert struktur som inneholder følgende:

- En last (`payload`) hvor innkommende IP-pakke legges.
- Størrelsen på lasten (`packetSize`).
- En tid som angir når i fremtiden pakken skal sendes ut av serveren (`packetTime`).
- En peker (`next`) til neste `packet` i en lenket¹⁶³ liste av `packet`-er.

Hver pakke får tildelt en `packetTime` først etter det har blitt undersøkt hvorvidt pakken skal forkastes i henhold til angitt prosentvis pakketap på kommandolinjen eller om pakken skal beholdes. Skal pakken beholdes, testes det om det på kommandolinjen eventuelt er konfigurert en pakkeforsinkelse større enn null. Deretter kalles `setPacketTime()`-funksjonen opp, hvor `packetTime` kalkuleres før pakken plasseres på rett plass i den lenkede listen.

¹⁶² På engelsk: ”random function”.

¹⁶³ På engelsk: ”linked list”.

For å avgjøre om en aktuell pakke skal bli forkastet, blir først det konfigurerte pakketapet konvertert til et tall (**percentageLoss**) ved følgende kodelinje i **main()**-funksjonen:

```
percentageLoss=(long) (((double) packetLoss/100.0) * (double) RAND_MAX);
```

Her er **packetLoss** den konfigurerte prosenten pakketap og **RAND_MAX** det høyeste tallet det er mulig å trekke fra en slumptallsgenerator (ca 2,1 milliarder). Deretter blir det i **InputPackets** trukket et tilfeldig tall mellom 0 og **RAND_MAX** ved bruk av **rand()**-funksjonen. Om det trukne tallet er større enn **percentageLoss**, skal pakken beholdes. Hvis ikke, skal pakken forkastes.

Når forsinkelsen gitt på kommandolinjen er større enn null, blir **packetTime** regnet ut som summen av nåtid (returnert fra en **gettimeofday()**-funksjon) addert med forsinkelsen oppgitt på kommandolinjen. Denne summen blir i **setPacketTime()** omgjort til sekunder og mikrosekunder for å oppfylle definisjonene i **struct timespec**, som brukes av **pthread_cond_timedwait()**-funksjonen for å holde igjen pakker inntil en viss klokke tid nås (mer omtale av denne funksjonen kommer i neste delkapittel).

Med en kommandolinje hvor pakkeforsinkelsen er satt til 0, blir verdien på **packetTime** normalt satt til 0 slik at pakkene sendes umiddelbart. Men er det heller ønskelig, kan pakkene tildeles en tilfeldig utsendelsestid. Da blir **packetTime** tilordnet en verdi mellom 0 og 100 ms trukket fra **drand48()**-funksjonen, og utføres av følgende linje i **setPacketTime()**:

```
randomTime=(long int) (drand48() *100000.0);
```

Denne funksjonen kan skrues av og på ved å henholdsvis kommentere eller fjerne kommenteringen av denne linjen i kildekoden og compilere programmet på nytt. Med denne funksjonen oppnås det at hver eneste innkommende IP-pakke får tildelt en helt tilfeldig **packetTime**. Pakkene sorteres i den lenkede listen utfra hvilken tid pakken skal slippes, i stigende rekkefølge. Den pakken som til enhver tid har den minste **packetTime** skal alltid ligge først i den lenkede listen. Således blir pakkene omstokket slik at de ikke sendes ut i samme sekvensielle rekkefølge som de ankom serveren.

6.1.2 Tråd 2

Tråd 2 jobber på **OutputPackets()**-funksjonen, som har ansvaret for å holde igjen første pakke¹⁶⁴ i den lenkede listen inntil PC-ens klokke har nådd **packetTime** og deretter sende pakken avgårde. Dette gjøres ved hjelp av henholdsvis **pthread_cond_timedwait()**-funksjonen i pthread-biblioteket og **sendto()**-funksjonen i socket-biblioteket. **pthread_cond_timedwait()** er definert til å ”sove”, det vil her si å avgi kontrollen til andre eksisterende tråder, inntil tiden som er gitt som parameter til funksjonen utløper eller til den mottar et signal fra **pthread_cond_signal()**-funksjonen. Dette signalet er et flagg som kan settes når et bestemt forhold oppstår. Når **pthread_cond_timedwait()** oppdager flagget, avbryter den straks ”sovingen”. Disse egenskapene til **pthread_cond_timedwait()** og **pthread_cond_signal()** utnyttet i **dgramServer** på følgende måte:

¹⁶⁴ I kildekoden benevnt "firstPacket".

Det testes i en if-setning om **pthread_cond_timedwait()** returnerer 110, hvilket returneres om tiden har utløpt. Hvis så, har pakken blitt holdt igjen lenge nok og skal sendes avgårde ved hjelp av **sendto()**. Deretter blir pakken fjernet fra den lenkede listen. Om **pthread_cond_timedwait()** ikke returnerer 110, er det fordi et signal har blitt detektert. Dette signalet har i så fall blitt satt av **InputPackets()** fordi det har kommet en ny pakke som har blitt plassert foran pakken som en gang var den første pakken i den lenkede listen, samtidig som **pthread_cond_timedwait()** holder igjen sistnevnte pakke.

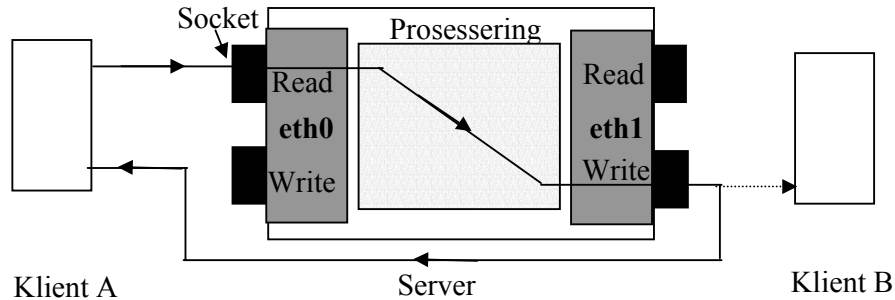
Dette er kritisk, da **pthread_cond_timedwait()** skal holde igjen den pakken som til enhver tid er først i listen. Det er således svært viktig å avbryte **pthread_cond_timedwait()** så fort som mulig for å begynne holdingen av den nye første pakken. Hvis ikke, risikerer det at den nye **firstPacket** blir sendt av gårde etter **packetTime**, fordi den ikke vil bli sendt før etter **pthread_cond_timedwait()** er ferdig med å holde igjen den gamle **firstPacket**. Slik **pthread_cond_timedwait()** er brukt i **OutputPacket()** vil den også, i tillegg til de nevnte tilfellene ovenfor, avgi kontrollen til tråd 1 når **pthread_cond_timedwait()** holder igjen en pakke eller når det ikke er flere pakker å holde igjen.

For at **pthread_cond_timedwait()** skal fungere må **pthread_mutex_lock()**-funksjonen kalles rett før kallet på **pthread_cond_timedwait()**. En slik ”mutex” kan sees på som en slags stafettpinne som **pthread_cond_timedwait()** må ”låses” ved hjelp av **pthread_mutex_lock()** for å fungere. Etter at pakken har blitt holdt igjen nok, låses stafettpinne opp igjen ved hjelp av **pthread_mutex_unlock()**.

6.1.3 Uttesting av serveren

For å gjøre testingen av serveren under utviklingsprosessen enklest mulig, ble det også utviklet en datagram-klient (mer informasjon om denne er gitt i kapittel 6.1.3.1). Dette ble gjort for å kunne sjekke at serveren fungerte og at det ble satt opp kommunikasjon. Under utviklingen av **dgramServer** ble det operert med det oppsettet beskrevet i figur 6.1. Linux-maskinen¹⁶⁵, som serveren ble kjørt på i den fasen, hadde to ethernet-kort, **eth0** og **eth1**. Det ble videre etablert en **read-** og **write-socket** for hver av disse to ethernet-kortene, som tok seg av henholdsvis innkommende og utgående trafikk. Under uttestingen ble pakkene, av ressursmessige grunner, ”loopet” tilbake til samme klient etter å ha vært innom serveren, selv om IP-pakkene reelt sett ville bli sendt til en annen mottagende klient, markert som klient B i figur 6.1.

¹⁶⁵ En Linux Redhat 7.0-basert PC av merket ”Digital Celebris GL 6200” med Intel Pentium Pro 200 MHz prosessor, 2 GB hardisk og 64 Megabyte minne.



Figur 6.1: Under uttestingen av serveren ble det operert med fire sockets, én inngående ("read") og én utgående ("write") for hvert av de to ethernet-kortene i Linux PC-en. Klient A sender IP-pakker til serverens "read"-socket på eth0 som blir utsatt for prosessering i serveren, før de blir sendt ut på "write"-socket på eth1 og returneres til klient A. Reelt ville talen bli sendt til en klient B fra serveren (markert med den stiplede pilen).

Med oppsettet beskrevet ovenfor, ble serveren testet for å sjekke om den holdt mål. Først ble det gjort testinger på hvor lang tid **InputPackets()** bruker på én iterasjon. Dette ble gjort ved å lese av et tidsstempel rett etter **recvfrom()** og ett før, når programmet kom dit igjen. Slik ble tiden målt før og etter **recvfrom()**, fordi denne funksjonen blir stående å vente på innkommende pakker, dermed ville tiden for en iterasjon ikke bli reell dersom denne ventetiden ikke ble trukket ifra. Si at tidsstemplene før og etter **recvfrom()**, kalles henholdsvis t_1 og t_2 . Differansen $t_2 - t_1$ ga således tiden for en iterasjon og ble funnet til å være mindre eller lik 100 μ s.

Tilsvarende test ble også utført på **OutputPackets()**. Tidsstempler ble, på lignende måte som i **InputPackets()**, plassert rundt **sendto()**. Det ble også her funnet at én iterasjon tar mindre eller lik 100 μ s.

Videre ble det, ved bruk av tidsstempler, testet om pakkene ble sendt til riktig tid og eventuelt hvor lang tid etter **packetTime** pakkene faktisk ble sendt ut. Resultatene viste at pakkene ble sendt ut omlag 20 ms etter opprinnelig tid. Dette betyr at pakkene blir utsatt for en ekstra forsinkelse i serveren på totalt cirka 20 ms. Her er det da antatt at de to forsinkelsene på 100 μ s kan neglisjeres. Det må understrekes at disse målingene ikke er helt 100% nøyaktige og var kun ment for å gi en indikasjon på om serveren var rask nok. Denne total forsinkelsen kan tillegges trådswitching i serveren og andre ressurskrevende prosesser (som ikke kan skrus av) i PC-en som tar CPU-tid fra server-programmet. Alle tidsdifferansene ble beregnet utfra den tiden **gettimeofday()**-funksjonen returnerte på de ulike målepunktene.

6.1.3.1 Klient

Som tidligere nevnt, ble det laget en enkel datagrambasert klient, **dgramClient**, for å lettere kunne verifisere under utviklingsprosessen at serveren fungerte som den skulle. I klienten er det opprettet en egendefinert struktur **struct payload** som inneholder et sekvensnummer, **seqNo**, og et tidstempel, **timeStamp**. I klienten kjører to tråder som henholdsvis styrer sending av pakker fra klienten (tråd I) og mottagelse av pakker fra serveren (tråd II). Tråd II har fått høyere prioritet enn tråd I for å unngå at pakker blir ventende unødvendig lenge før de blir behandlet av klienten. Klienten sender et gitt antall pakker avgårde med en instans av **struct payload()** som last og venter en justerbar tid før den sender en ny pakke ut. Tråd II mottar kun pakker og leser ut

dataene i **payload** og finner ”round-trip delay”. Socket-ene og trådene ble etablert i henhold til socket- og pthread-biblioteket i POSIX-basert Linux.

6.2 Muliggjøre forsendelse av tale igjennom dgramServer

På det tidspunktet dgramServer skulle testes ut i kommunikasjon med H.323-baserte IP-telefoni klienter, ble det raskt oppdaget at det ikke lot seg gjøre å få satt opp samtaler mellom to slike klienter via dgramServer. Grunnen var svært enkel: i og med at dgramServer ikke støtter H.323, ble ikke signaleringen fra den ene klienten besvart av den andre, under oppsetningen av et kall via Linux-maskinen. På grunn av at signaleringen ikke gikk i orden, koblet den initierende parten ned kallet og det ble derfor ikke mulig å sende talen via Linux-maskinen med kun dgramServer kjørende på den. Det ble derfor nødvendig å ha en H.323-gatekeeper¹⁶⁶ kjørende i parallell med dgramServer, som kunne sørge for at signaleringen i forbindelse med oppsettet av et kall ble fullført. Slik kunne det dermed la seg gjøre å få klientene til å sende talen via Linux-maskinen til mottagende klient. Videre måtte gatekeeper-en også kunne fungere som en *proxy*, det vil si at den videresender all taletrafikk (RTP-pakker) i tillegg til å håndtere H.323-signaleringen. Det ble valgt at ”*OpenH323 Gatekeeper – The GNU gatekeeper*”¹⁶⁷ versjon 2.0.2 i proxy-modus¹⁶⁸ skulle benyttes for å løse dette.

Ved å kjøre denne proxy-en i parallell med dgramServer lot det seg gjøre å få to IP-telefoni klienter til å sette opp en samtale seg imellom via Linux-maskinen, hvor også talen gikk innom Linux-maskinen på vei til mottageren. Neste utfordring ble å få dirigert talepakkene innad i Linx-maskinen til dgramServer. Løsningen ble å bruke *divert sockets* i stedet for regulære datagram socket-er i dagramServer.

6.2.1 Divert sockets

Ordet ”divert” betyr på norsk ”å avlede”, ”å omlede” eller ”å omdirigere” og navnet er neppe tilfeldig valgt. Divert sockets¹⁶⁹ gjør det nemlig mulig å avskjære IP-pakker igjennom filtrering¹⁷⁰ av dem. Slik blir det mulig å omdirigere pakkene til divert socket-er i ”user space”, hvor man står fritt til å manipulere på pakkene før de blir satt på plass i køen de ble dirigert vekk fra – om det er ønskelig. Man kan nemlig velge om én, alle eller ingen av pakkene skal settes på plass igjen. Om man ikke setter pakkene på plass igjen, vil pakkene bare ikke nå destinasjonen. Dersom det ikke er noen program i ”user space” som benytter divert sockets og som lytter til det portnummeret pakkene blir omdirigert til, vil pakkene følge den opprinnelige veien til destinasjonen. Avskjæring og re-innsetting skjer i implementasjonen av IP under Linux.

¹⁶⁶ Mer informasjon om H.323 og gatekeeper, finnes i Appendiks A.

¹⁶⁷ Heretter kalt gnugk. For mer informasjon, se www.gnugk.org. Denne vil også bli omtalt som *proxy*.

¹⁶⁸ Konfigurasjonsfilen finnes i Appendiks H.

¹⁶⁹ For mer informasjon, se <http://www.anr.mcnc.org/~divert/index.shtml>. Beskrivelsen av divert sockets i dette avsnittet er basert på informasjon fra denne siden.

¹⁷⁰ Til dette brukes iptables, som gjør det mulig å sette firewall-regler for inngående – og utgående trafikk slik at pakker blir filtrert. For mer informasjon, se <http://www.netfilter.org/>.

6.2.2 Endringer i og bruk av dgramServer

For å kunne benytte divert socket i dgramServer har det vært nødvendig med en viss modifisering av kildekoden. Til å begynne med ble de fire socket-ene vist i figur 6.1 etablert med denne kodelinjen:

```
socket(PF_INET, SOCK_DGRAM, 17);
```

hvor **PF_INET** angir at det her etableres en Internett-socket som er datagram-basert (**SOCK_DGRAM**) og basert på protokollnummer 17 (UDP).

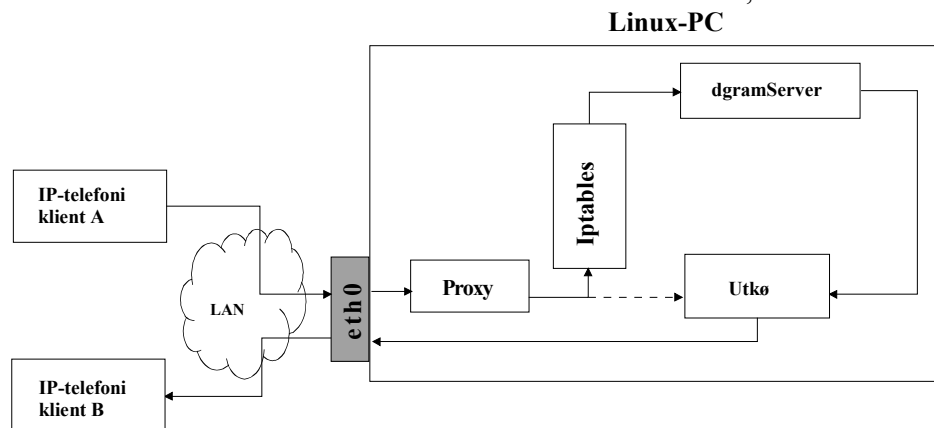
For å opprette en divert socket ble det nødvendig å endre den overnevnte kodelinjen til følgende:

```
socket(PF_INET, SOCK_RAW, IPPROTO_DIVERT);
```

hvor det opprettes en Internett-socket, som opererer på IP-pakker hvor hodene ikke er ”strippet av” (**SOCK_RAW**). **IPPROTO_DIVERT** angir at divert sockets brukes.

På grunn av at det ble nødvendig å benytte en proxy på Linux-maskinen i tillegg til å kjøre dgramServer, trengtes det ikke lengre å bruke to ethernet-kort. Videre ble det også opprettet kun én divert-basert socket til å håndtere både inngående- og utgående trafikk til programmet. Disse endringene gjorde også at scenarioet for bruken av dgramServer ble annerledes enn opprinnelig antatt (hvilket ble vist i figur 6.1).

I figur 6.2 er det vist hvordan dgramServer ble brukt under talekvalitetstestingene. Figuren viser kun hvilken vei talepakkene tar fra ”IP-telefoni klient A” til ”IP-telefoni klient B”, signaleringen mellom de to IP-telefoni klientene via proxy-en er ikke vist¹⁷¹. Figuren tar utgangspunkt i at samtalen mellom klientene har blitt opprettet og at talen blir sendt over et LAN til Linux-maskinens ethernet-kort, eth0.



Figur 6.2: Talen fra ”IP-telefoni klient A” sendes over et LAN via en proxy (kjørende på Linux-PC’en) til mottageren, ”IP-telefoni klient B”. I Linux-maskinen blir talepakkene avledet av en iptables filtreringsregel og sendt til dgramServer, istedenfor å bli plassert rett i utkøen og bli sendt til mottageren (angitt med den stiplede pilen). Etter at dgramServer er ferdig med pakkene blir de plassert i utkøen og sendt avgårde.

Proxy-en videresender konsekvent alle innkomne talepakkene til utkøen for forsendelse til mottagende klient (slik som angitt med den stiplede pilen i figuren). For

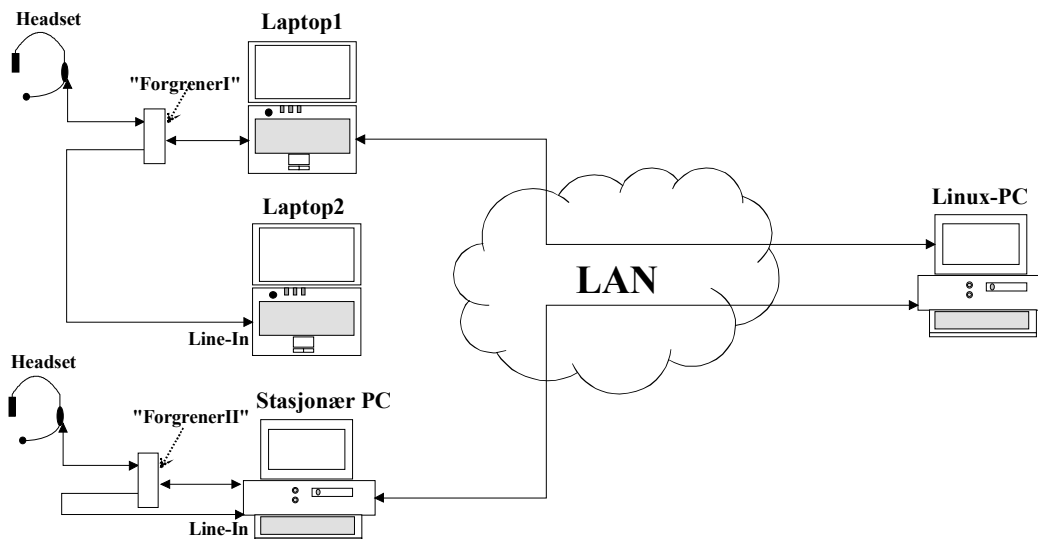
¹⁷¹ H.323-signalering er fylldig omtalt i appendiks A.

å hindre at dette skjer og for å gjøre det mulig at dgramServer kan få tak i disse IP-pakkene, har det blitt laget en filtreringsregel¹⁷² med iptables. Denne filtreringsregelen sier at alle utgående talepakker på portnummer 7600¹⁷³ fra programmet som ble startet med kommandolinjen *gnugk* skal avledes og sendes til et annet internt portnummer, som dgramServer lytter til. Således lykkes det å få sendt talepakkene til dgramServer, slik at det kan utøves manipulasjon på dem. Etter at dgramServer er ferdig med en IP-pakke blir den plassert i utkøen.

Endringene i kildekoden til dgramServer, som beskrevet på forrige side, ble gjort for å tilrettelegge for at programmet skal lytte til og ta til seg avledete pakker. For at "divert sockets" skal fungere må nemlig to funksjonaliteter være på plass. Man må ha en filtreringsregel som avleder pakkene og et program som venter på disse avledete pakkene. Som tidligere nevnt, tar ingen program imot disse pakkene blir de sendt ut som om ingenting har skjedd. Men denne ekstra prosesseringen vil likevel kunne innebære en viss ekstra forsinkelse for IP-pakkene i Linux-maskinen, sammenliknet med om de hadde blitt sendt ut med engang uten å ha blitt avledet til dgramServer. Det er vanskelig å anslå hvor stor denne er, da dette gjøres av operativsystemet og hvordan IP er implementert samt ytelsen til maskinvareplattformen. Disse faktorer ligger utenfor omfanget av dette prosjektet.

6.3 Testoppsett

Under opptakene av talen, som ble brukt i talekvalitetstesting, ble oppsettet vist i figur 6.3 benyttet.



Figur 6.3: Tale mellom laptop1 og stasjonær PC-en blir sendt over et 10 Mbit/s LAN via Linux-PC'en. I hver ende blir innkommende tale forgrenet med henholdsvis "forgrenerI" og "forgrenerII", slik at lyden som høres i headset-et også blir ført til line-in på henholdsvis laptop2 og stasjonær PC-en for digitalt opptak. Detaljer om "forgrenerI" og "forgrenerII" gis i henholdsvis figur 6.4 og 6.5.

¹⁷² Se Appendiks H.

¹⁷³ Brukes av gnugk for forsendelse av talepakker. Denne ble funnet ved å analysere portnumre i bruk på Linux-maskinen ved hjelp av netstat, et verktøy innebygget i Linux. For informasjon om portnumre som brukes i H.323, se <http://www.cs.columbia.edu/~hgs/rtp/h323.html>.

Tale mellom en laptop¹⁷⁴ (*laptop1* i figuren) og en stasjonær¹⁷⁵ PC blir sendt over et 10 Mbit/s LAN, via Linux-maskinen. *dgramServer* og *proxy-en* kjørte på Linux-maskinen slik som angitt i figur 6.2. Den opprinnelig brukte Linux-maskinen (beskrevet i kapittel 6.1.3) ble byttet ut med en ny PC¹⁷⁶ som kjørte Linux 8.0 med 2.4.18 kjerne, da den opprinnelige ikke fungerte under maksimal belastning. Det vil si at den ikke klarte å filtrere innkomne IP-pakker og samtidig kjøre både *dgramServer* og *proxy-en*, når toveis talekommunikasjon ble satt opp igjennom den. Hele maskinen låste seg. Det er fortsatt usikkert på om dette skyldtes hardware-spesifikke årsaker eller om feilen skyldtes problemer med Linux-kjernen som ble brukt.

Innkommende lyd i hver ende blir forgrenet i henholdsvis ”*forgrenerI*” og ”*forgrenerII*” (disse vil bli mer omtalt på neste delkapittel) og ført til line-in, slik at talen som høres i headset-et også blir tatt opp digitalt. Under de første opptakene ble oppdaget en bakgrunnsstøy, som minnet om lyden fra en trafostasjon, på talen generert fra laptop-siden når laptop2¹⁷⁷ ble brukt til å kjøre IP-telefoni klienten. Det ble derfor besluttet å benytte *laptop1* til å kjøre IP-telefoni klienten istedenfor, da det hadde blitt observert at bakgrunnsstøyen forsvant med dette oppsettet. Denne løsningen ble valgt også fordi det ikke var tid til å fordype seg i hva som kunne være årsaken til bakgrunnsstøyen. I ettertid, imidlertid, ansees det som sannsynlig at bruk av strømforsyning på laptop2 kan ha forstyrret audiosignalene og dermed ha generert denne trafoliknende støyen som hørtes i den andre enden.

Laptop1 hadde imidlertid ikke kapasitet nok til å både kjøre IP-telefoni klienten og kjøre nødvendig programvare for å foreta opptakene (denne omtales mer i 6.4), uten at det var fare for klipping i opptakene. Derfor ble *laptop2* brukt til dette.

På både *laptop1* og stasjonær PC-en ble det kjørt en IP-telefoni klient, kalt OpenPhone versjon 1.5.3, fra OpenH323¹⁷⁸. I begge maskinene var det installert et Quicknet¹⁷⁹ IP-telefoni kort. I laptop-en satt det et ”PhoneCARD” PC-card og i stasjonær PC-en et ”Internet PhoneJACK” PCI-kort. Tekniske spesifikasjoner for disse to IP-telefoni kortene finnes i Appendiks F. OpenPhone støttet to hardware codec-er innebygget i Quicknet-kortene, G.723.1 (5,3 kbps) og G.723.1A (6,3 kbps)¹⁸⁰. Av disse ble G.723.1A, med tre talerammer¹⁸¹ per IP-pakke, valgt da denne virket å gi den beste talekvaliteten. Stort sett ble ”default” innstillinger brukt, blant de få endringene som ble foretatt gjaldt ekkoutligning og jitterbuffer. Disse ble justert opp fra henholdsvis ”medium” til ”high” og fra 50 ms til 100 ms, fordi dette syntes å forbedre talekvaliteten. For detaljer om innstillingene av OpenPhone, se Appendiks H.

¹⁷⁴ Siemens Scenic Mobile 501, Pentium II prosessor (266 MHz), 98 MB minne, 4 GB hardisk.

¹⁷⁵ Dell OptiPlex GXpro, Pentium Pro prosessor (200 MHz), 128 MB minne, 20GB hardisk.

¹⁷⁶ En PC fra Komplettdata (www.komplett.no) med AMD Athlon XP200 1.66 MHz 266 MHz bus Socket A (Palomino) prosessor, MSI KT4 Ultra hovedkort, 512 MB minne og 80 GB hardisk.

¹⁷⁷ Dell Latitude CPi D300XT, Pentium II prosessor (300 MHz), 128 MB minne, 6 GB hardisk.

¹⁷⁸ www.openh323.org.

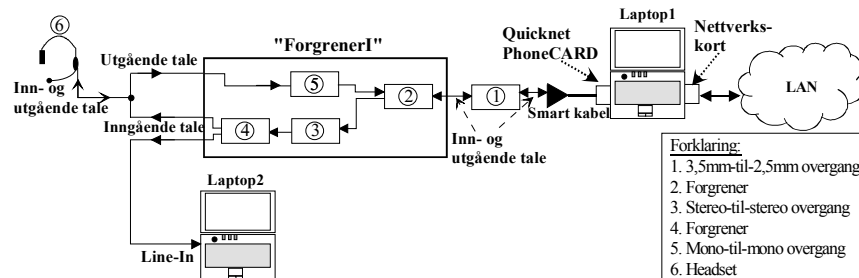
¹⁷⁹ www.quicknet.net.

¹⁸⁰ Tidligere også benevnt som G.723.1 med 6,3 kbps bitrate.

¹⁸¹ Det ble opplyst fra produsenten at en G.723.1-codec med tre talerammer er den beste innstilling om nettverket ikke er mettet (”congested”).

6.3.1 Oppkobling

Dette delkapitlet skal utdype hvordan testoppsettet i figur 6.3 var koblet opp. Figur 6.4 tar for seg laptop-siden av testoppsettet. Det fulgte med en såkalt smartkabel til "Quicknet PhoneCARD" for tilkobling av enten én 2.5 mm stereoplugg eller én RJ-11 telefonplugg. På grunn av at lyden som høres i høyttaleren på headset-et skulle tas opp digitalt, ble det nødvendig med å bruke den førstnevnte tilkoblingen. Dette gjorde det enklest å skille ut og forgrene inngående tale for viderekobling til laptop2.



Figur 6.4: Signaler fra mikrofonpluggen på headset-et (6) går igjennom en mono-til-mono overgang (5) for å filtrere vekk signalene i stereo kontaktpunktet på pluggen. Signalene går videre igjennom en forgrener (2) og en 3.5 mm-2.5 mm overgang (1) før de blir sendt via en smartkabel (som er tilkoblet IP-telefoni kortet Quicknet PhoneCARD) ut til mottageren. Innkommende tale går motsatt vei til forgrener (2), hvor de blir sendt igjennom en stereo-til-stereo overgang (3) for å filtrere vekk signalene i monokanalen. Derfra blir signalene sendt igjennom en forgrener (4), hvor signalene føres til både høyttaleren på headset-et og til laptop2 for digitalt opptak. "Forgrener1" henviser til symbolet brukt i figur 6.3. Detaljer om de ulike komponentene finnes i tabell 6.1 og finnes avbildet i figur 6.5.

En ulempe med denne løsningen var at smartkabelen ikke var laget for bruk av 3.5 mm¹⁸² stereoplugg. 2.5 mm plugger opplevdes ikke å være lett å få tak i. Det lyktes likevel å få kjøpt inn en 3.5mm-til-2.5mm stereo overgang, et såkalt *jackadapter*. Denne er nummerert med 1 i figur 6.4, mer detaljer om de ulike komponentene i figuren finnes i tabell 6.1.

I det følgende skal selve oppkoblingen på laptop-siden forklares (jamfør figurene 6.4, 6.5 (her avbildes komponentene) og tabell 6.1):

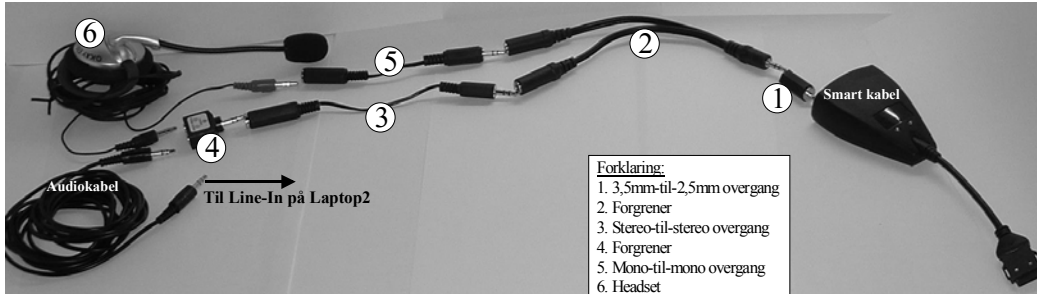
- Jackadapteret (1) var koblet direkte til stereo tilkoblingen på smartkabelen. I den andre enden av jackadapteret var den ene forgreneren (2) tilkoblet. I den andre enden av forgreneren ble den ene "grenen" brukt til å lede utgående signaler (fra mikrofonen) og den andre til å lede inngående signaler (til høyttaleren).
- *Inngående signaler*: Utgående signaler blir "kopiert"¹⁸³ i (2) og finnes følgelig i mono kontaktpunktet i "grenen" for innkommende tale. Derfor måtte (3) kobles inn mellom (2) og (4). Slik ble det forhindret at også mikrofonsignalene ble sendt videre til (4). Mikrofonsignalene kan således sies å ha blitt filtrert vekk. Hadde ikke dette blitt gjort, ville også utgående tale blitt sendt til opptak. Det var, som tidligere nevnt, ikke ønskelig. (3) var videre tilkoblet til en ny forgrener (4). Denne lager to "kopier" av den innkomne talen, hvorav høyttalerpluggen på headset-et (6) var tilkoblet den ene grenen mens den andre ble koblet til line-in på laptop2 via en audiokabel¹⁸⁴.
- *Utgående signaler*: Mikrofonpluggen på headset-et (6) var koblet til (5), som igjen var tilkoblet grenen for utgående signaler i (2). (5) var nødvendig for å

¹⁸² Den vanligste størrelsen på stereopluggen til walkman, cd-spillere, mini-disc og lignende.

¹⁸³ Dette er illustrert i figur F.1 i appendiks F.

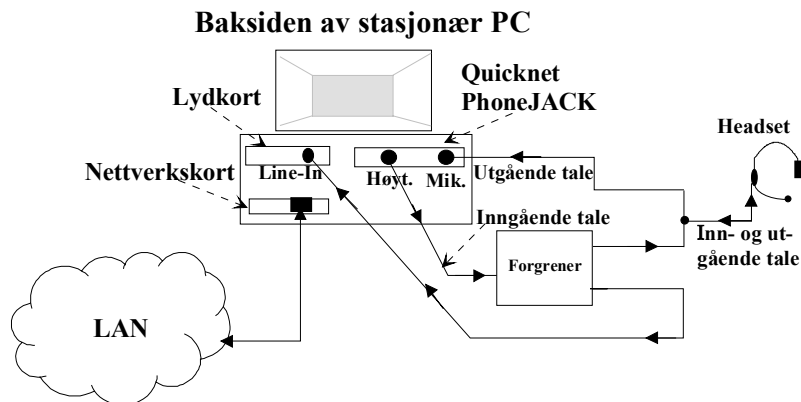
¹⁸⁴ Varenummer 32.7702, Clas Ohlson.

filtrere vekk stereokanalen, siden mikrofonpluggen var loddet med mikrofonlederen til både mono- og stereo kontaktpunktet. Om ikke dette ble gjort ville utgående talesignaler bli sendt til stereo kontaktpunktet i smartkabelen, *samtidig* som inngående tale også ble sendt inn på dette kontaktpunktet. Dette ville ha gitt kaos i signalgangen.



Figur 6.5: Bildet viser oppkoblingen av komponentene slik som vist i figur 6.4.

Figur 6.6 viser hvordan den stasjonære siden av figur 6.3 var koblet opp. Mikrofonpluggen på headset-et¹⁸⁵ ble koblet til mikrofoninngangen på PhoneJACK PCI-kortet. Til høyttalerutgangen på kortet ble det koblet inn en forgrenner¹⁸⁶. Høyttaler-pluggen på headset-et ble koblet til den ene grenen, mens det ble koblet til en audiokabel til den andre. Den andre enden av kabelen ble koblet til line-in inngangen på PC-ens lydkort for opptak.



Figur 6.6: Mikrofonpluggen på headset-et ble koblet rett til mikrofoninngangen på IP-telefoni kortet. Inngående tale, fra høyttalerutgangen, blir forgrenet i *forgrenner* (benevnt "forgrennerII" i figur 6.3) slik at disse signalene blir sendt både til høyttaleren på headset-et og til opptak i line-in på lydkortet.

¹⁸⁵ Samme mikrofon som ble brukt på laptop-siden. Det brukes ellers komponenter som allerede er nevnt i tabell 7.1.

¹⁸⁶ Samme type som (4) i figur 7.4. Inngikk i symbolet "forgrennerII" i figur 7.3.

Nr	Hva	Beskrivelse	Varenr.	Leverandør
1	3,5mm-til-2,5mm overgang	Et jackadapter med hunn-til-han kobling.	303342	OSK Elektrotilbehør A/S, 7038 Trondheim
2	Forgrener	3,5 mm til 2x3,5 mm forgrener. En forgrener gir to parallelle stereosignaler av det opprinnelige stereosignalet. Dette kan sees på som at blir laget to ”kopier” av det opprinnelige signalet. Dette blir illustrert i figur F.1 i appendiks F.	SC3990012424	Expert Strømmen, Strømsveien 49-51, 2010 Strømmen.
3	Stereo-til-stereo overgang	Denne ble laget ved å lodde et stykke vanlig audio kabel mellom en hunn- og han 3.5 mm stereo plugg. Det innerste loddepunktet på slike plugges brukes for signaler i monokanalen og det ytterste for signaler i stereokanalen. Ved å benytte det <i>ytterste</i> loddepunktet i begge ender av kabelen for lederen av stereosignalene og ved å lodde de andre lederne til jord, oppnår man å skille ut høytersignaler (det vil si signalene i monokanalen). Grunnen til at denne og nr. 5 ble laget, var fordi produsenten av IP-telefoni kortene oppga at koblingen for stereopluggen på smartkabelen, var laget slik at mikrofonsignalene ligger i monokanalen og høytersignaler i stereokanalen. Signalene ligger slik til og med komponentnr. 2. Ettersom en forgrener gir to ”kopier” av det opprinnelige, ble det nødvendig å kunne styre hvilke signaler som skulle viderekobles etter en forgrener (2) eller sendt til en forgrener (4).	I. 22.723 II. SC3990010611	I. Clas Ohlson ¹⁸⁷ II. Expert
4	Forgrener	3,5 mm til 2x3,5 mm forgrener	22.728	Clas Ohlson.
5	Mono-til-mono overgang	Tilsvarende som for nr. 3. For å filtrere vekk høytersignaler (signalene i stereokanalen), brukes de <i>innerste</i> loddepunktene i hver ende.	I. 22.723 II. SC3990010610	I. Clas Ohlson II. Expert
6	Headset	”OKAY EM-1001V Single Communication Earset”. Utstyrt med to separate 3.5 mm stereoplugges. Én for mikrofonsignalene og én for høytersignaler. Mikrofonpluggen var loddet med lederen av mikrofonsignalene til både mono- og stereo kontaktpunktet.	32.2763	Clas Ohlson

Tabell 6.1: Stykkliste over komponentene som ble benyttet i oppkoblingen av testoppsettet. Nummereringen refererer til figur 6.4.

¹⁸⁷ www.clasohlson.no

6.4 Opptak av talen

Innkommende tale ble tatt opp i både laptop2 og stasjonær PC-en ved hjelp av lydprogrammet "Sound Forge" versjon 4.5 fra Sonic Foundry¹⁸⁸. Deltagerne i opptakene satt i separate og stille rom. Før opptakene ble gjort, ble mikrofon- og lyttevolum i OpenPhone og volumet på line-in inngangen for hver av PC-ene justert. Dette ble gjort for å hindre at volumet på talen ble for høyt, men samtidig for å sørge for at lyttevolumet på headset-et var høyt nok til å gi komfortabel lytteopplevelse. Lyttevolumet som ble innstilt i OpenPhone virker direkte inn på opptaksvolumet, da OpenPhone således stiller inn volumet som IP-telefoni kortet bruker for gjengivelse av inngående tale.

"Sound Forge" har et innebygget instrument som gjorde det mulig å måle innspillingsvolumet og således gjorde det mulig å stille inn de ulike volumene slik at innspillingsvolumet ville ligge rundt -10 dB (decibel). Slik ble det sørget for at volumet på line-in var såpass høyt at det var mulig å minimalisere susing i lydopptaket. Samtidig ble det sørget for at volumet ikke ble for høyt, hvilket kunne føre til at lyden "sprekker" fordi codec-en kutter.

Codec for innspillingen ble satt til PCM med 8 kHz samplingsfrekvens og med 16 bits oppløsning på opptakene. Dette var den codec-innstillingen som ga riktigst lyd kvalitet ut fra det som hørtes i høytaleren på headset-et. Innstillinger med høyere samplingsfrekvens ga oversampling og medførte en utilfredsstillende mengde med susing, særlig i stille perioder.

Det ble laget fem opptak for hver parameter (de valgte parametrene for pakke tap og pakkeforsinkelse omtales i neste delkapittel) for å være sikker på at man hadde tilstrekkelig antall opptak for bruk i de endelige opptakene som skulle legges på Internett. Dessuten ble dette gjort for å være sikker på at man hadde opptak hvor manus satt godt. De fem opptakene ble gjort sekvensielt for å være sikre på at de ble gjort under tilnærmet like nettverksforhold. Organiseringen av lydopptakene innad i filene som ble lagt ut på Internett, vil omtales mer i kapittel 6.4.3.

6.4.1 Valg av parametre

For å komme frem til et endelig sett med parametre, ble det først gjort en mengde forsøksopptak med ulike parameter-innstillinger i dgramServer. Etter grundige gjennomlysninger og analyse, ble antall parametre i de endelige opptakene begrenset til følgende:

- 0, 5, 10, 15, 20 og 25 prosent pakke tap.
- 0, 200, 300, 400 og 600 ms pakkeforsinkelse.

Pakke tap over 25% ga en så ødelagt tale at det neppe ville ha vært godtatt for vanlig telefoni og dermed ble det fastslått at det også var overveiende sannsynlig at kvaliteten var for dårlig til at den kunne egne seg for ATC-bruk. Derfor ble det besluttet å ikke teste talekvaliteten med pakke tap høyere enn dette. Til sammeligning testet [WaS97] med pakke tap opptil 50% (tidligere nevnt i delkapittel 5.5). Videre ble

¹⁸⁸ <http://www.sonicfoundry.com/>

det observert at 5% og 10%, i tillegg til 0%, burde kunne ha gode sjanser til å godkjennes, men det er nok sannsynlig at de vil kunne få ulike kvalitetsbedømmelser. Det syntes nemlig mulig å høre forskjell på 5% og 10%. Kvalitetene for 0%, 5% og 10% så likevel ut til å ligge nærmere hverandre enn fra 10% og oppover. Tilsynelatende var det også lettere å høre forskjell på 10% og 0%, enn på 5% og 0%. 15% ble nokså naturlig opplevet å ligge som en mellom ting i kvalitet mellom 10% og 20%. Det var likevel vanskeligere å høre en like stor forskjell på 15% og 20%, som på 10% og 15%. Det var en enda mer merkbar forskjell mellom 20% og høyere prosenter. For et utrenet øre vil derfor kvalitetene på 5% og 0% kunne føles tilnærmet like, og tilsvarende for kvalitetene på 15% og 20%.

Forsøk med vanlig telefonkonversasjon, når ulike pakkeforsinkelser ble justert, viste at innstillinger fra 50 ms til 200 ms ikke ga store nok forsinkelser til at samtalen ble ødelagt. Dette var også i henhold til (som tidligere nevnt i kapittel 3.3.2) [HNA00] som sier at forsinkelser under 150 ms generelt betegnes som akseptable og at en tommelfingerregel sier at enveisforsinkelser bør holdes under 200 ms [Per99]. Likevel, testpersonellet opplevde at mengden bortfallende enkeltord i talen økte proporsjonalt med økt forsinkelse. Dette skyldes sannsynligvis at pakkene som frakter disse uttalelsene ble såpass forsinket at de kom for sent til avspilling og dermed ble ansett som tapte. I disse tilfellene hørte man en slags skurring når codec-en prøvde å reparere talen. Dette påvirker *lydkvalitets*komponenten i begrepet talekvalitet (jamfør definisjonen av talekvalitet i kapittel 3.1). Ved 300 ms, derimot, fikk forsinkelsen merkbar effekt på *konversasjonsdynamikken* (den konversasjonsmessige delen av begrepet talekvalitet) og ved 400 ms gikk samtalen i stå. Den ene parten ble utålmodig, da ingen svar kom, og begynte å gjenta i det svaret fra den andre parten kom. Resultatet var at partene snakket i munn på hverandre og hele konversasjonen klappet sammen. Dette er også helt i tråd med [Cox97] (også tidligere nevnt i kapittel 3.3.2) som sier at for forsinkelser større enn 300 ms, vil konversasjonen arte seg som halv-dupleks eller ”push-to-talk”.

Under de endelige opptakene, derimot, med konversasjon basert på ATC-tale var situasjonen en helt annen. På opptakene hører man en økende grad av at enkelte uttalelser blir borte med høyere innstilte pakkeforsinkelsene. Allikevel, det ble funnet at pakkeforsinkelser opptil 600 ms ikke ga negative utslag på konversasjonsdynamikken. Det kan tillegges at ATC-tale er av typen ”push-to-talk”. Det vil si at man henvender seg til en mottager med et spørsmål eller med en beskjed og man *venter* til man får svar, altså en halv-dupleks konversasjon. Etter vår kunnskap, er det strengt innarbeidet i ATC at det ikke snakkes i munn på hverandre og at man ikke forventer diverse små uttalelser som verifiserer at mottageren er på linjen. I vanlige telefonsamtaler, derimot, forventer man små uttalelser slik som ”hmm, hmm”, ”ja, ja” og så videre ellers oppstår det tvil om samtalepartneren fortsatt er på linjen. I så fall begynner man å gjenta.

Det skal ikke gå mer enn 10 sekunder¹⁸⁹ i ATC fra en beskjed gis til svar kommer før beskjeden skal gjentas. Likevel, utfra observasjonene under opptakene virket det ikke som at konversasjonsdynamikken i halv-dupleks tale, slik som ATC, blir påvirket av pakkeforsinkelser opptil 600 ms. Pakkeforsinkelser opptil 300 ms var i utgangspunktet det høyeste tallet interessant for testing, men grunnet det overnevnte

¹⁸⁹ E-post fra Kostas Patouras, gresk flygeleder. E-posten ligger vedlagt i appendiks E.2.

ble det besluttet også å ta med 400 og 600 ms. *Det er min oppfatning at talekvaliteten ved pakkeforsinkelser opptil 400 ms kan bli bedømt å være akseptabelt for bruk i ATC.* Siden ATC-tale er halv-dupleks og IP-telefonitalen også kan karakteriseres slik for pakkeforsinkelser større enn 300 ms, kan det nemlig være at denne kombinasjonen er gunstigere for talekvaliteten i ATC enn vanlige telefonsamtaler.

Videre, det kan hende at jitterbuffre håndterer halv-dupleks tale lettere enn vanlige telefonsamtaler ved høye pakkeforsinkelser. Vanlige telefonsamtaler kan ofte karakteriseres som lengre utgreinger og monolog, hvilket kunne gi en lengre strøm av pakker enn i ATC-tale. I slik tale, derimot, blir de enkelte korte meldingene sendt etterfulgt av lengre tids stillhet før ny melding sendes. Muligens kan dette ha en heldigere effekt på jitter-bufferet.

På bakgrunn av det overnevnte, lot ikke konversasjonsdynamikken seg så lett bedømme. Heretter vil derfor begrepet *talekvalitet* dekke alle andre faktorer *enn* konversasjonsdynamikk som påvirker kvaliteten på talen (som for eksempel lydmessige kvaliteter (blant annet bakgrunnstøy med mer), klarhet og graden av forståelighet).

6.4.2 Forholdene under opptakene

For å hindre at annen trafikk i 10 Mbps nettverket skulle påvirke talekvaliteten, ble opptakene utført på en søndags ettermiddag. Videre ble nettuttak valgt med hensyn til at maskinene skulle enten være tilknyttet samme switch eller switcher som lå nærme hverandre. Etter forsøksopptakene ble klart at det var såpass mye variabel forsinkelse i nettverket og i Linux-maskinen med full-dupleks samtale igjennom den, at det ble besluttet å kutte ut omstokkingen av IP-pakkene i dgramServer. Denne omstokkingen skjer jo, som tidligere beskrevet, ved å legge til en *tilfeldig* forsinkelse i tillegg til den konfigurerte pakkeforsinkelsen. Dette ville gi så mye variabel forsinkelse at det ikke lenger ville være mulig å få reproducerbare forhold. Dessuten ville pakkene kunne få en naturlig omstokking i nettverket. En simulert omstokking gir dessuten mest mening om man opererer på et helt isolert og lukket nettverk. Videre, en slik omstokking får også størst effekt om omstokkingen er såpass omfattende at jitterbufferet ikke klarer å oppveie for dette. Først da får man testet innvirkningene av dette på talekvaliteten i og med at jitterbufferet er laget for å håndtere en viss mengde naturlig omstokking av IP-pakker. I enkelte IP-telefoni klienter, slik som OpenPhone, kan størrelsen på jitterbufferet justeres, men dette kan også variere utfra hva som støttes av hardwaren.

Oppsummert, selvom omstokkingen i dgramServer ble kuttet ut, ble pakkene likevel utsatt for diverse variable forsinkelser i nettverket. Dette er likevel forsinkelser som IP-telefoni mest sannsynlig vil bli utsatt for når talepakkene sendes over et tilsvarende nettverk. Utover dette vil også de ulike faste- og variableforsinkelsene nevnt i kapittel 4 også være gjeldende, men dette er forsinkelser som generelt vil eksistere i IP-telefoni. Forsinkelsene i Linux-maskinen kommer i tillegg til dette. Eksempler på slike forsinkelser vil være forsinkelser i forbindelse med at proxy-en videresender pakker, i filtreringen av pakkene og programforsinkelsen i dgramServer.

6.4.3 Prosessering av opptakene

Etter at opptakene var gjort, ble lydfilene fra hver ende satt sammen til én lydfil. Lyden fra laptop-siden ble lagt i det ene lydsporet og lyden fra stasjonær PC-en i det andre. Slik har det, som nevnt tidligere, blitt forsøkt å illustrere at man sitter mellom to personer og overhører deres samtale, hvor man hører den til venstre for seg i venstre øret og den til høyre i det andre.

Ved prøving og feiling ble det forut for opptakene funnet ut at signalene som skal tas opp bør ha et størst mulig volum, men ikke høyere enn at talen ikke sprekker. Det ga nemlig opptak med minst susing i opptakene, særlig i perioder med tale. Også i stille perioder ble ikke susing så lett oppfanget av øret, da talesignalene var såpass mye høyere enn volumet på susingen. Under opptakene pratet den kvinnelige deltageren¹⁹⁰ lavere enn forventet og de innstilte volumene rettet ikke opp dette tilstrekkelig. Resultatet ble at talen fra den kvinnelige deltageren ga mer susingen i opptakene, spesielt i stille perioder, enn hva som var opprinnelig antatt. For å utjevne volumet på opptakene for kvinne- og mannsstemmen, ble hvert av lydsporene normalisert til -10 dB ved å bruke "Normalize"-funksjonen i Sound Forge. Slik ble susingen naturligvis også forsterket, da volumet på opptakene av kvinnestemmen ble forsterket *opp* til -10 dB. Dette ble ryddet opp i ved bruk av "Noise gate"-funksjonen i Sound Forge. Denne funksjonen fjerner lyd lavere enn et konfigurert volum.

Grunnen til at dette ble rettet opp i, var fordi det var fryktet at volumforskjellen i hver av lydkanalene kunne være en forstyrrende faktor under bedømmingen. I utgangspunktet var det ikke meningen å utføre noen prosessering av opptakene, da tanken var at opptakene skulle gjenspeile, på godt og vondt, kvaliteten på det som hørtes i headset-et. Var det dårlig, skulle opptakene vise dette og visa versa. Ingen prosessering skulle gi opptakene bedre eller verre kvaliteten enn det talen virkelig holdt. Det var nettopp denne begrunnelsen som ble lagt til grunn for avgjørelsen om å prosessere lyden fra den personen som snakket lavest. Dette ga nemlig mer susing i opptaket enn man hørte i headset-et "live" under konversasjonene og gjenspeilte dermed ikke den reelle talekvaliteten. I dette tilfelle, ville en viss prosessering gi et mer riktig bilde av hva som ble hørt i headset-et.

Til slutt ble opptakene konvertert fra det opprinnelige wav-filformatet til 128 kbps mp3, da dette halverte filstørrelsen og ikke syntes å forringe kvaliteten på opptakene. Minst mulig filstørrelse var definitivt en fordel, da lydopptakene skulle lastes ned fra Internett.

6.4.4 Organisering av opptakene

Det ble bestemt at avspillingsrekkefølgen på opptakene skulle organiseres i vertikale *serier* slik som det er gitt et eksempel på i figur 6.7 (hvor hver bokstav angir ett unikt manus) og som tidligere beskrevet i kapittel 5.4.1. Alle parametrene for pakketap ble tatt med i seriene, men for pakkeforsinkelse ble 0 ms kuttet ut. Grunnen til dette var at denne parameteren teknisk sett var samme innstilling i dgramServer som 0 %. For pakketap bestod seriene derfor av seks ulike opptak (én for hver av de seks

¹⁹⁰ Det ble benyttet en kvinne- og en mannsstemme i opptakene.

testparametrene for pakketap) og tilsvarende bestod seriene for pakkeforsinkelse av fire ulike opptak.

Videre ble det besluttet at kun tre av fem opptak for hver av parametrene for pakkeforsinkelse og pakketap skulle legges ut, da dette ville være tilstrekke for å få et bredt resultatgrunnlag. På denne måten ble det også mulig å velge de tre opptakene hvor manus satt best. Slik ble altså for både pakketap og pakkeforsinkelse lagt ut tre serier på Internett, jamfør den vertikale organiseringen i figur 6.7.

		Opptaksnummer				
		1	2	3	4	5
Parameter	0%	A	A	A	A	A
	5%	B	B	B	B	B
	10%	C	C	C	C	C
	15%	D	D	D	D	D
	20%	E	E	E	E	E
	25%	F	F	F	F	F

Figur 6.7: Figuren er en gjengivelse av figur 5.4a. Det ble benyttet samme manus for alle opptakene av én parameter (bokstavene i figuren skal illustrere dette). Opptakene ble lagt ut på Internett som serier, organisert vertikalt i figuren. Dette er illustrert med den svarte rammen. Tilsvarende ble gjort for pakkeforsinkelse, men da med kun fire parametre.

Den totale spilletiden på seriene var 18 minutter. I hver serie ble rekkefølgen opptakene omstokket, slik at ikke ingen av de tre seriene for pakketap og -forsinkelse var like. I den første serien (for både pakketap og -forsinkelse) var avspillingsrekkefølgen omstokket vilkårlig, i den andre ble forholdene gradvis bedre og i den tredje serien ble forholdene gradvis forverret. En grunn til at opptakene ble organisert slik, var at flygelederne skulle få en mest mulig variert lytteopplevelse. Den viktigste grunnen var likevel for å sørge for at det ikke skulle oppleves noen mønster i rekkefølgen av opptakene. Flygelederne skulle ikke få inntrykk av at "her er det en serie med god/dårlig kvalitet" og slik forhåndsdomme, eventuelt bedømme uten å egentlig å ha lyttet til opptakene.

Det ble videre antatt at rekkefølgen utover dette ikke burde ha noen innvirkning på bedømmelsene, da hvert lydopptak skal bedømmes umiddelbart etter gjennomlytting. Slik skal ikke hukommelsen kunne spille inn, i form av man husker best det man hører til sist, hvilket kunne påvirket bedømmingene om *alle* opptakene ble gjennomlyttet før bedømmelse. Videre ble det ikke funnet noen grunn til å etablere referanseforhold for bedømmelsene, det vil si et par opptak med beste kvalitet avspilles før bedømmelsene setter i gang og at talekvaliteten bedømmes oppimot disse referanseforholdene. Referanseforholdet i talekvalitetstestene i dette prosjektet er talekvaliteten på det eksisterende kommunikasjonssystemet flygelederne bruker i sitt daglige arbeid. Således gir det ingen mening å gi et referanseforhold med den beste kvaliteten man klarte å oppnå på talen, fordi det er *de* som skal avgjøre om talekvaliteten er god nok utfra hva de er vant med. Dessuten er det bare de som vet hva slags talekvalitet som er god nok og det ville derfor ikke gi mening at det skulle forhåndsprosederes hva som er *beste* kvalitet.

Til slutt ble det utviklet en enkel hjemmeside for prosjektet, på både engelsk og norsk, hvor prosjektet presenteres og hvor opptakene ble lagt ut. Sammen med hver serie ble det også lagt med en spilleliste-fil, hvilket kan åpnes i de fleste mp3-program, for

enkler avspilling. Opptakene ble gjort tilgjengelige som zip-filer, hvor det var mulig å laste ned alle opptakene som en samlet fil eller som separate filer. På websiden ble instruksjoner for bedømmelsene gitt, og evalueringsskjemaer ble lagt ut på engelsk og norsk og i ulike filformat. Videre ble det inngått en avtale med Avinor Skolesenter¹⁹¹ om at foreleserne der skulle delta i dette prosjektet. Foreleserne der er blant de beste operative flygelederne i landet som har blitt rekruttert til å lære opp flygelederstudenter. Skolesenteret opprett også kontakt med operative flygeledere. Videre ble det også sendt en mail ”European Air Traffic Controllers’ Mailing List¹⁹²” og til en gresk flygeleder, som det har blitt korrespondert med, med henvisning til prosjektets hjemmeside. Dette ble gjort for å ytterligere forsterke talekvalitetstestene ved at flygeledere fra forskjellige nasjoner deltok i testene.

¹⁹¹ Utdanner flygeledere

¹⁹² http://groups.yahoo.com/group/euro_atcontrollers/

Kapittel 7

Testresultater

I dette kapittelet skal testresultatene fra flygeledernes evalueringer av talekvaliteten presenteres. Flygelederne ble bedt om å evaluere hvert av opptakene rett etter avspilling på følgende tre punkter (jamfør evalueringsskjemaet vedlagt i appendiks G.2):

1. **Deres mening om talekvaliteten på konversasjonen.**
2. **Deres mening om hvor mye innsats som var nødvendig for å forstå meningen med talen i lydopptakene.**
3. **Si ja eller nei til om talekvaliteten var tilstrekkelig for bruk i luftfartskommunikasjon.**

I tillegg fikk flygelederne i evalueringsskjemaet mulighet til å komme med kommentarer til lydopptakene. I punkt 1 ble "Listening-quality"-skalaen benyttet, mens i punkt 2 ble "Listening-effort"-skalaen benyttet (begge er gjengitt i tabell 7.1). Gjennomsnittsverdiene for disse skalaene kalles henholdsvis "Mean Opinion Score" (MOS) og "Mean Listening-Effort Opinion Score" (MOS_{LE}). For øvrig ble flygelederne oppfordret til å benytte hodetelefoner under avspilling for best mulig utbytte av lyttingen.

"Listening-effort"		"Listening-quality"	
Effort required to understand the meanings of the sentences.	Score (MOS _{LE})	Quality of the Speech.	Score (MOS)
Complete relaxation possible; no effort required.	5	Excellent	5
Attention necessary; no appreciable effort required.	4	Good	4
Moderate Effort required.	3	Fair	3
Considerable effort required.	2	Poor	2
No meaning understood with any feasible effort.	1	Bad	1

Tabell 7.1: Flygelederne benyttet seg av "Listening Quality"-skalaen til å bedømme kvaliteten på talen, mens "Listening effort"-skalaen ble brukt til å bedømme hvor mye lytteinnsats som trengtes for å forstå meningen med talen. Gjennomsnittsverdiene heter henholdsvis "Mean Listening-Effort Opinion Score" (MOS_{LE}) og "Mean Opinion Score" (MOS). Disse skalaene er tidligere gjengitt i tabell 5.3.

Grunnet at det ikke var lett å få flygeledere til å bruke arbeidstiden til å delta i prosjektet og at så få flygeledere fra mailinglisten meldte sin interesse, lyktes det kun å få innhentet 10 evalueringer. Disse fordelte seg på to greske flygeledere, én spansk og syv norske flygeledere fra Avinor (hvorav én flygeleder var fra skolesenteret og seks i operativ tjeneste¹⁹³). Hver testparameter ble vurdert utfra tre opptak. På denne måten ble hver testparameter bedømt totalt 30 ganger. Rådataene i fra evalueringene ligger vedlagt i appendiks I. Det må understrekes at resultatene i dette prosjektet kun er ment å gi en *indikasjon* på, og ikke absolutter, for kvaliteten på VoIP-basert ATC. Dette prosjektet er for en pilotstudie eller et første initiativ å regne, og om resultatene i dette prosjektet skulle vise seg å være interessante for videre arbeid kan det legges inn mer tid og penger for å oppnå et bredere resultatgrunnlag.

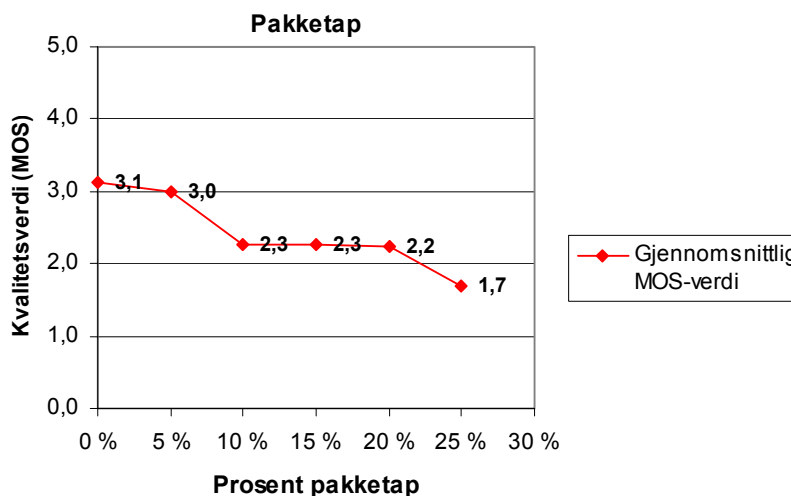
I resten av dette kapittelet vil flygeledernes bedømmelser presenteres og analyseres på overnevnte tre punkter først med pakketap som varierende parameter og deretter med pakkeforsinkelse. Overraskelser i resultatene vil bli mer drøftet i kapittel 8.

7.1 Pakketap

I dette delkapitlet vil resultatene fra flygeledernes bedømmelser av pakketap presenteres.

7.1.1 Talekvalitet

Resultatene viser noenlunde synkende gjennomsnittlige kvalitetsverdier (MOS) for økende prosentar pakketap, slik som forventet (se graf 7.1). Disse verdiene inneholdt likevel noen overraskelser i forhold til hva som var forventet. På bakgrunn av observasjonene ved opptak, var det generelt ventet en en brattere kurve, med høyere kvalitetsverdier (MOS)¹⁹⁴ for 0%, 5% og 10% og litt lavere verdier 15%, 20% og 25%.



Graf 7.1: Grafen viser gjennomsnittlig kvalitetsverdi (MOS) som funksjon av økende pakketap.

¹⁹³ Oslo Kontrollsentral (Oslo ATCC), region øst-Norge

¹⁹⁴ Som tidligere nevnt, vil begrepene kvalitetsverdi (MOS) og lytteinnsatsverdi (MOS_{LE}) brukes om MOS- og MOS_{LE}-verdi

Særlig 0% og 5% fikk overraskende lave kvalitetsverdier (MOS). Likevel, aller mest overraskende var det at 0% pakketap ikke ble bedømt bedre enn 3,1. Det var her ventet en verdi i området 4 til 5 siden denne talen i prinsippet skal være feilfri. For et utrenet øre ville det være trolig at kvalitetsverdien (MOS) til 20% pakketap, $MOS(20\%)^{195}$, skulle ende på et nivå under 2,0.

I kapittel 6.4.1 ble redegjort for observasjonene på lydopptakene. Det vil i det følgende bli sett på disse og sammenliknet med resultatene i graf 7.1:

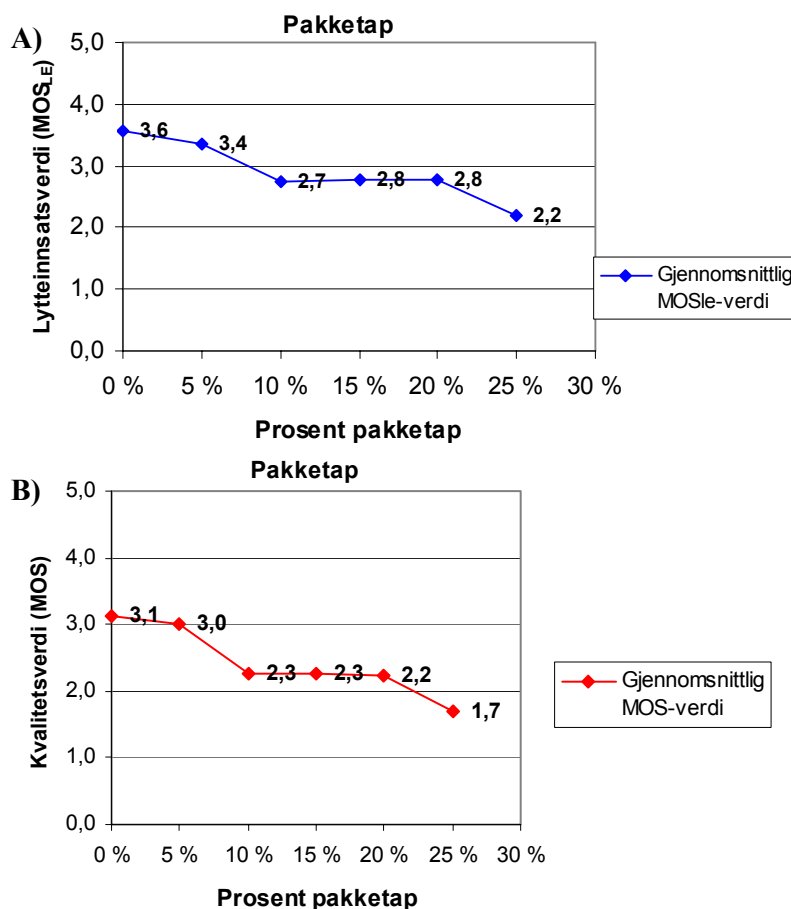
- **”Mulig å høre forskjell på 5% og 10%”**
Dette støttes langt på vei av resultatene ettersom 5% fikk en gjennomsnittlig kvalitetsverdi (MOS) på 3, mens 10% kun fikk 2,3.
- **”Kvalitetene til 0%, 5% og 10% så ut til å ligge nærmere hverandre enn 10% og oppover”**
Denne observasjonen støttes ikke av resultatene. Graf 7.1 viser at 0% og 5% ligger like nærme hverandre som 15% og 20%. Videre fikk 10% samme bedømmelse som 15%. Dessuten viser resultatene en større forskjell mellom 20% og 25% enn forventet.
- **”Lettere å høre forskjell på 10% og 0%, enn på 5% og 0%”**
Resultatene støtter dette siden differansen $MOS(0\%)-MOS(10\%)$ er større enn $MOS(0\%)-MOS(5\%)$.
- **”15% opplevdes som en mellomting i kvalitet mellom 10% og 20%, men likevel var det vanskelig å høre like stor forskjell på 15% og 20% som på 10% og 15%”**
Resultatene støtter ikke noen av disse observasjonene ettersom alle tre parametrene ble bedømt tilnærmet likt.
- **”Enda en mer merkbar forskjell mellom 20% og høyere prosenter”**
Selv om testingene ble avgrenset til 25% som høyeste testparameter støtter resultatene denne observasjonen, da det er 0,5 forskjell mellom $MOS(20\%)$ og $MOS(25\%)$. Dette er en større differanse enn differansene mellom kvalitetsverdiene (MOS) til henholdsvis 10%, 15% og 20%.
- **”For et utrenet øre vil kvaliteten på 5% kunne oppleves å være tilnærmet lik kvaliteten på 0%, og tilsvarende for 15% og 20%”**
Denne observasjonen støttes langt på vei av resultatene, da disse viser at $MOS(5\%) \approx MOS(0\%)$ og $MOS(20\%) \approx MOS(15\%)$.

7.1.2 Lytteinnsats

I graf 7.2A) observeres mye av det samme i graf 7.1, nemlig at lytteinnsatsverdiene (MOS_{LE}) for 0% og 5% ligger lavere enn man kunne forvente og at 20% og 25% fikk verdier litt høyere enn antatt. Ellers lå 10% og 15% på et litt mer forventet nivå. Resultatene i graf 7.2A) viser videre at det generelt for alle parametrene kreves en nokså stor mengde med innsats for at meningen med talen skal forstås. Grafen viser også at 0% med en lytteinnsatsverdi (MOS_{LE}) på 3,6 er den eneste parameteren som har potensiale til å gi en avslappet lytteopplevelse. De nokså lave kvalitetsbedømmelsene i graf 7.1 gjenspeiles også i den mengden gjennomsnittlige lytteinnsatsen som graf 7.2A) antyder er nødvendig for å kunne forstå meningen med talen. Her viser resultatene en klar sammenheng mellom talekvaliteten og lytteinnsats: det kreves en lytteinnsats tilnærmet lik kvalitetsverdien (MOS) på talen.

¹⁹⁵ Denne skriftsformen vil i det følgende også bli brukt om kvalitetsverdien (MOS) til en gitt parameter. Tilsvarende vil også brukes for lytteverdier (MOS_{LE}).

Som nevnt i forrige avsnitt følger kvalitetsverdiene (MOS) og lytteinnsatsverdiene (MOS_{LE}) hverandre tett. Det er for så vidt en naturlig sammenheng, da en dårlig talekvalitet burde kreve mer lytteinnsats enn en bedre én. Dette var på forhånd likevel ikke tatt for gitt. I kapittel 5.4.2 ble det nemlig diskutert at om en gitt parameter blir bedømt til en kvalitetsverdi (MOS) på for eksempel 3 men gis en lytteinnsatsverdi (MOS_{LE}) på 5, kunne dette indikere at "Listening-quality"-skalaen ikke gir et riktig bilde av kvaliteten på talen. Grunnen til dette ville være at en slik bedømmelse i så fall innebar at talen holdt en middelmådig kvalitet, *samtidig* som den *ikke* krevde noen form for lytteinnsats. Den reelle talekvaliteten kunne således være *bedre* enn det "Listening-quality"-skalaen indikerte. Tilsvarende kunne den reelle kvaliteten være dårligere enn det denne skalaen indikerer om talen ble gitt lytteinnsatsverdier (MOS_{LE}) lavere enn kvalitetsverdiene (MOS).



Graf 7.2: Graf A) viser gjennomsnittlig lytteinnsatsverdi (MOS_{LE}) som funksjon av økende pakketap. Grafen i B) er en gjengivelse av graf 7.1 og viser kvalitetsverdiene (MOS) for pakketap.

Denne påstanden støttes ikke av resultatene i graf 7.2A). For å illustrere dette har graf 7.1 blitt gjengitt i graf 7.2B) og plassert under graf 7.2A). Her ser man at kurvene er nesten identiske og tilnærmet parallellforskjøvet. Dette illustrerer godt hvordan kvalitetsverdiene (MOS) og lytteinnsatsverdiene (MOS_{LE}) følger hverandre. Parallellforskyvingen tilsvarer differansen mellom kvalitetsverdiene (MOS) og lytteinnsatsverdiene (MOS_{LE}), og varierer i området 0,4-0,6. Dette er en såpass liten forskjell at påstanden ikke kan sies å bli underbygget av resultatene. Tvert imot, kan

det heller sies at kvalitetsverdiene (MOS) og lytteinnsatsverdiene (MOS_{LE}) er tilnærmet like hverandre. Gitt argumentasjonen i forrige avsnitt, gir dette dermed en grunn til å tro at "Listening-quality"-skalaen gir et *riktig* bilde av talekvaliteten i ATC-sammenheng.

I kapittel 5.4.2 ble det også spekulert i hvilken grad hvor feilfri og krystallklar talen er har betydning for talekvalitetsbedømmelsene. Det ble hevdet at kvaliteten ellers kan være dårlig bare innholdet forstås. Men de lave bedømmelsene i graf 7.1 understøttet av den relativt betydelige mengden lytteinnsats som graf 7.2A) antyder at behøves, kan tyde på at hvor feilfri talen er er en viktig medvirkende faktor for bedømmelsen av talekvaliteten og for forståelsen av innholdet. Lydopptakene for pakkeap kan derfor ikke ha holdt mål på dette punktet. Kommentarene fra flygelederne tyder også på dette.

Eksempler på slike kommentarer er:

- *"Det at opptakene er "hakkete" medfører at viktige detaljer i transmisjonen kan komme bort, ergo upålitelig i denne sammenhengen"*
- *"Sample 1-6 er utydelig, har varierende lydstyrke og er delvis forvrengt"*
- *"Sample 1-5 er litt forvrengt og har varierende lydstyrke. Ikke egnet til luftfartskommunikasjon"*

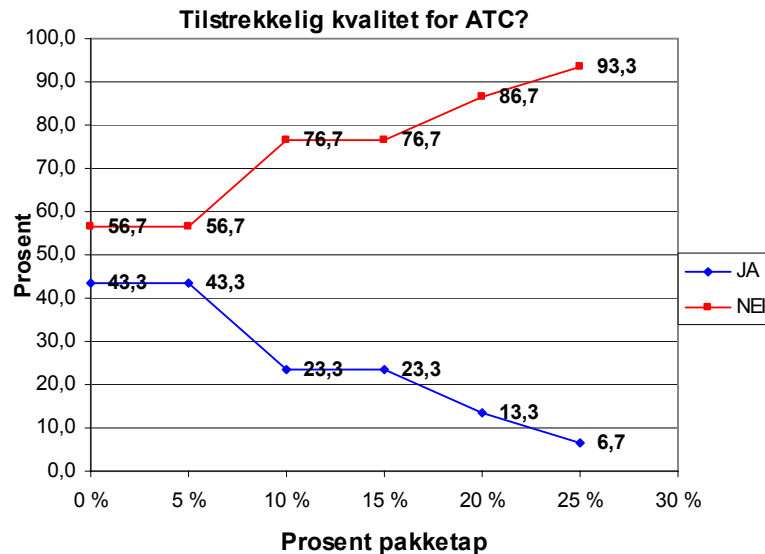
7.1.3 Resultatene svar på om kvaliteten er tilstrekkelig for ATC

Oppslutningene for ja og nei til om talekvaliteten var tilstrekkelig for ATC var som ventet henholdsvis jevnt synkende og økende. Dette er i tråd med de synkende kvalitetsverdiene (MOS) og lytteinnsatsverdiene (MOS_{LE}) i henholdsvis graf 7.1 og 7.2A). Resultatene presentert i graf 7.3 underbygger ytterligere antagelsen om at 0% og 5% kan sies å holde tilnærmet lik talekvalitet i og med at begge disse parametrene fikk samme oppslutning. Ellers observeres det at 10% og 15% fikk samme oppslutning. Dette er også i henhold til kvalitetsbedømmelsene i graf 7.1, hvor begge disse prosentene ble gitt en kvalitetsverdi (MOS) på 2,3. Dessuten ble disse tildelt en tilnærmet lik lytteinnsatsverdi (MOS_{LE}).

Nokså uventet var det likevel at grafene ikke krysset hverandre et sted mellom 0 og 10%, da det var ventet at særlig 0%, 5% og også kanskje 10% skulle få et ja-flertall. Dette krysningspunktet var mest ventet å ligge rundt 5%. Resultatene støtter således heller ikke resultatene i [KBS98] som viste at pakkeap opptil 10% hadde lite merkbar effekt på talen når G.723.1 ble brukt som codec (tidligere referert til i dekkapittel 3.3.1). Dette kan skyldes at [KBS98] ikke testet talen i ATC-sammenheng og at deres tale derfor ikke ble bedømt utfra de strenge sikkerhetskravene som stilles til ATC-tale. Det er også usikkert hvorvidt [KBS98] skilte mellom codec-ene G.723.1 og G.723.1A, disse har som vist i tabell 4.1 forskjellige bitrater. Det er derfor også noe usikkert hvor relevant det er å sammenligne resultatene i dette prosjektet med de i [KBS98].

Resultatene indikerer ganske klart at ingen av parametrene kan sies å holde en kvalitet tilstrekkelig for ATC, da mer enn 50% sa nei til dette. Samtidig viser resultatene likevel at kvalitetene på 0% og 5% ikke er håpløst dårlige ettersom over 40% sier ja. Kommentarer fra enkelte flygeledere underbygger dette. Eksempler på slike kommentarer er:

- ”Sample 6 er ok, men ikke 100%”
- ”Kan til nød godkjennes, liten men dog markert forvrengning”
- ”Sample 5: Beste hittil, alt kan forstås med litt innsats”
- ”Mangler litt på å bli godkjent, noen deler faller bort”
- ”Noen av filene er det rimelig god kvalitet på, men så kommer det deler innimellom med tall som blir uleselig. Helhetsinntrykket er bra, men siden hele filen må være leselig har jeg satt en lav karakter”



Graf 7.3: Grafen viser antall prosent JA og NEI-oppslutning per tapsparameter for om talen holdt en kvalitet tilstrekkelig for ATC.

På bakgrunn av det overnevnte kan pakketap opptil 5% sies å ha potensiale til å bli godkjent. Resultatene i graf 7.1 og graf 7.2A) viser også dette. I så fall ville dette i resultatene tilsvare en laveste godkjente kvalitetsverdi (MOS), MOS_{grense} , på 3 og en maksimal godkjent forringelse, f_{maks} , på 5% slik som antatt i kapittel 5.

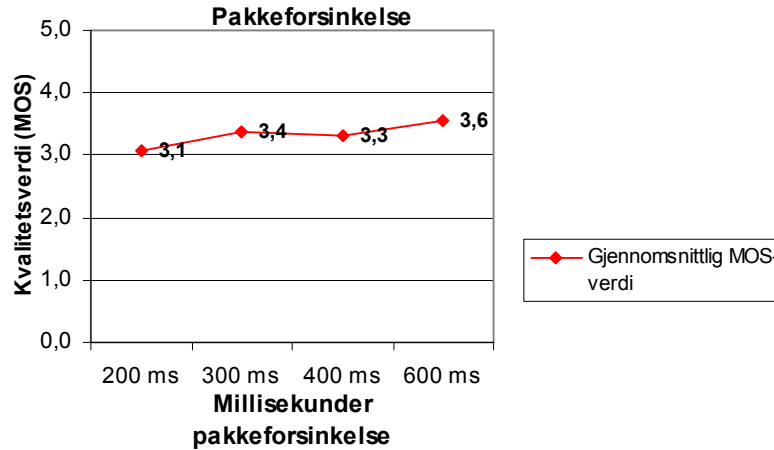
7.2 Pakkeforsinkelse

I dette delkapitlet presenteres og analyseres flygeledernes bedømmelser når antall millisekunder pakkeforsinkelse varierer.

7.2.1 Talekvalitet

Graf 7.4 viser at resultatene ga en svakt stigende kurve med økende pakkeforsinkelser. Dette var overraskende da det var ventet at andelen tapte enkeltord ville øke med økende pakkeforsinkelse. Særlig var det på forhånd ventet at dette ville påvirke talekvaliteten for 400 ms og 600 ms. Spesielt var det uventet at 600 ms lå høyere og 200 ms lavere enn de andre parametrene. Det kan være tilfeldigheter eller noe med selve opptakene som kan skyldes dette. Ellers var det utfra observasjonene på lytteopptakene ventet at bedømmelsene jevnt over skulle gi høyere kvalitetsverdier (MOS). Men resultatene viser likevel at pakkeforsinkelse ikke har like stor negativ påvirkning på talekvaliteten i ATC som pakketap. Dette gjenspeiles i den jevnere og

høyere bedømmelsen pakkeforsinkelsene fikk sammenlignet med pakketap. Videre viser resultatene også at talekvalitetsbedømmelsene ikke i stor grad ble påvirket av økende pakkeforsinkelser. Bedømmelsene ligger relativt tett på "Listening-quality"-skalaen og det er ingen store differanser i kvalitetsverdiene (MOS) for de ulike pakkeforsinkelsene.

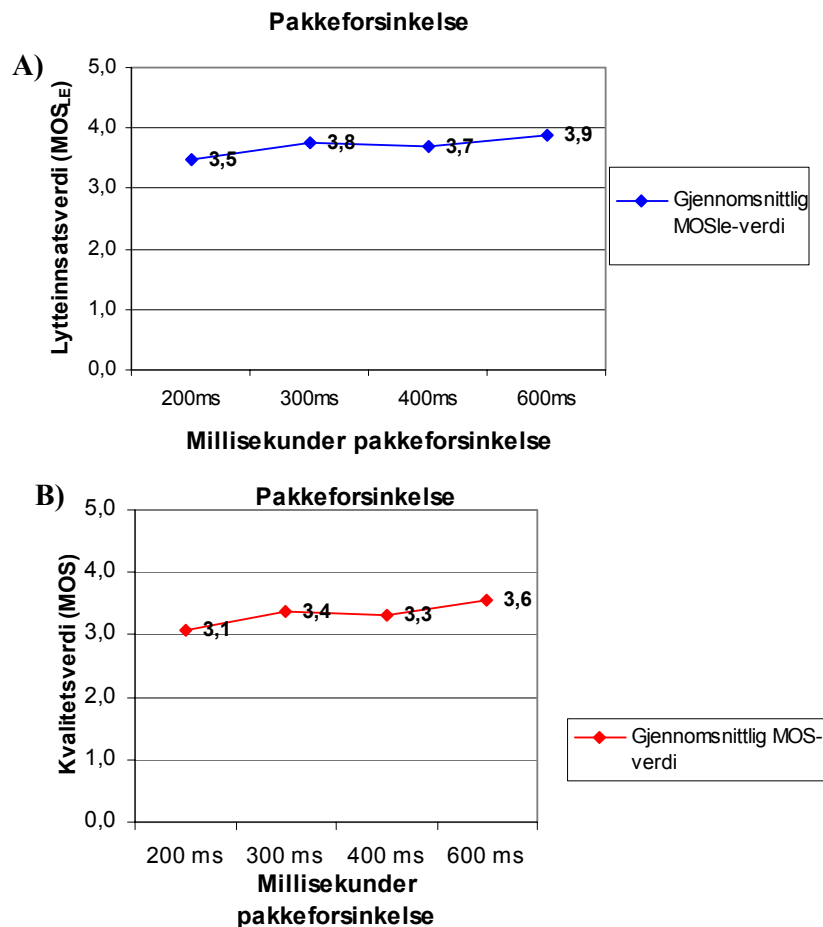


Graf 7.4. Grafen viser gjennomsnittlig kvalitetsverdi (MOS) som funksjon av økende pakkeforsinkelser.

7.2.2 Lytteinnsats

Graf 7.5A) viser at lytteinnsatsverdiene (MOS_{LE}) er fordelt tettere enn tilsvarende graf for pakketap, slik som også kvalitetsverdiene (MOS) i graf 7.4 var. Ellers ligger kurven høyere (rundt 3,5 til 4) og holder et jevnere nivå enn tilsvarende kurve for pakketap i graf 7.2A). Dette indikerer at jevnt over kreves mindre lytteinnsats for å forstå meningen når talen blir utsatt for pakkeforsinkelse sammenliknet pakketap. Dette er også i tråd med de jevnt høyere kvalitetsverdiene (MOS) for pakkeforsinkelse kontra pakketap. Lytteinnsatskurven er også like jevn og svakt stigende slik som kvalitetskurven i graf 7.4.

Videre, ved å sammenlikne grafene 7.5A) og 7.4 (gjengitt i graf 7.5B) observeres det at lytteinnsatskurven ligger parallellforsjøvet med en verdi på 0,4 i forhold til kvalitetskurven. Dette forsterker inntrykket av at "Listening-quality"-skalaen gir riktig inntrykk av talekvaliteten i ATC. Videre kan se ut til at kvalitetskurven og lytteinnsatskurven i større grad konvergerer enn for pakketap ved pakkeforsinkelser større enn 600 ms. Resultatene gir ingen bevis for at disse to kurvene *vil* krysse hverandre og kan være tilfeldig, men om de hadde gjort dette kunne dette i så fall bety at den reelle talekvaliteten er dårligere enn det kvalitetskurven tilsier (som nevnt i delkapittel 7.1.2) når pakkeforsinkelsene blir høyere enn 600 ms.



Graf 7.5: I A) vises gjennomsnittlig lytteinnsatsverdiene (MOS_{LE}) som funksjon av økende pakkeforsinkelser. B) er en gjentakelse av graf 7.4 og viser kvalitetsverdiene (MOS) for pakkeforsinkelse.

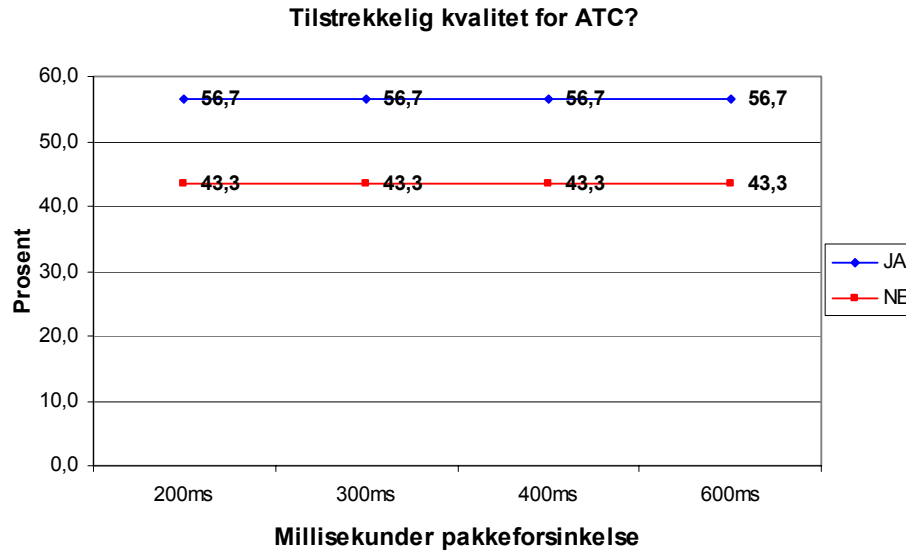
7.2.3 Resultatene svar på om kvaliteten er tilstrekkelig for ATC

Nokså overraskende fikk alle parametrene lik ja-oppslutning. Dette kan kanskje skyldes tilfeldigheter som igjen kan skyldes at antall respondenter i testene har vært for få. Utfra graf 7.4 og 7.6 kan det stadfestes at resultatene støtter antagelsene i kapittel 6.4.1 om at pakkeforsinkelser opptil 600 ms ikke gir negative utslag på talen. Begrunnelsen for dette er at alle parametrene godkjennes med nesten 57% og at kvalitetsverdiene (MOS) og lytteinnsatsverdiene (MOS_{LE}) ligger jevnt mellom 3 og 4. Tilsvarende støttes også antagelsen om pakkeforsinkelser opptil 400 ms ville holde en talekvalitet akseptabel for ATC. Resultatene viser likevel at kvaliteten generelt ikke er helt ideell i og med at en såpass stor andel som 43% sa nei. Dette gjenspeiles også i graf 7.4 og 7.5A) hvor både talekvaliteten og lytteinnsatsen kun ble bedømt til å være *middels* selvom et relativt *stort flertall* sa ja til at talekvaliteten er tilstrekkelig for bruk i ATC. Kommentarer fra flygelederne tyder også på dette. Eksempler på slike kommentarer er:

- ”Mangler litt på kvaliteten totalt sett”
- ”Kutter i starten, litt hakkete”
- ”Bra tale, alt kommer med, litt dårlig lyd kvalitet”

- ”Beste i testen så langt, men mangler litt”
- ”Noen deler faller bort”
- ”Bra nok, men ikke perfekt”

Resultatene gjør det vanskelig å eksakt definere nedre grense for hvor mye pakkeforsinkelse talen kan holde og fortsatt holde en talekvalitet tilstrekkelig for ATC. Hovedårsaken til dette er de vannrette kurvene i graf 7.6, men det kan i alle fall utfra resultatene slås fast at kvalitetsverdier (MOS) over 3 for pakkeforsinkelse er tilstrekkelig for bruk i ATC.



Graf 7.6: Grafen viser antall prosent JA og NEI-opplutning per forsinkelsesparameter for om talen holdt en kvalitet tilstrekkelig for ATC.

Kapittel 8

Diskusjon

8.1 Konversasjonsdynamikk

I dette prosjektet lot det seg ikke så lett gjøre å få bedømt konversasjonsdynamikken. Dette skyldtes delvis den valgte fremgangsmåten for talekvalitetstesting, det kommer ikke frem av opptakene den følelsen en deltager av en konversasjon får når samtalen går i stå. Opptakene kan i beste fall kun illustrere hvordan samtalepartnerne begynner å gjenta og eventuelt til slutt prater i munn på hverandre. Som utenforstående lytter kan man derfor kun bedømme konversasjonsdynamikken utfra dette. Den viktigste årsaken, derimot, er at det har vist seg i dette prosjektet at ATC-talens egenart er av en slik karakter at konversasjonen ikke så lett klapper sammen som vanlige telefonsamtaler (som beskrevet i kapittel 6.4.1). Slik blir det kun lyd kvaliteten (herunder også om innholdet i talen forstås) som lar seg bedømme. Altså, hvordan eventuelle manglende uttalelser eller andre faktorer påvirker kvaliteten på talen. Det at opptakene var basert på manus gjorde det heller ikke lettere å få frem eventuelle forandringer i konversasjonsdynamikken i og med at begge deltagerne visste hvilket svar som skulle komme.

En annen årsak til at ATC-talen under opptakene ble observert å tåle høyere innstilte pakkeforsinkelser bedre enn forsøket med vanlig telefonkonversasjon (som beskrevet i kapittel 6.4.1), kan helt enkelt være at Linux-maskinen, med alt av software kjørende, taklet halv-dupleks tale bedre enn en full-dupleks telefonsamtale. En full-dupleks samtale vil jo innebære to samtidige talestrømmer igjennom maskinen, kontra én ved en halv-dupleks samtale. Talekvaliteten ved høye pakkeforsinkelser kan derfor ha blitt bedre med ATC-tale enn med vanlige telefonkonversasjon.

8.2 Overraskelser i testresultatene

Resultatene presentert i kapittel 7 inneholdt noen avvik fra det som var forventet. For pakketap var det særlig uventet at 0% og 5% ikke ble bedømt bedre og ikke ble godkjent. Jevnt over ble talen imidlertid bedømt til å holde en nokså lav kvalitetsverdi (MOS) – for både pakketap og -forsinkelse. Det kan ikke helt utelukkes at dette kanskje kan skyldes at forsinkelsene i Linux-maskinen, i tillegg til de konfigurerte i `dgramServer`, og kanskje noe i nettverket¹⁹⁶ hadde en større påvirkning på talekvaliteten enn det var mulig å anslå på forhånd. Andre og kanskje mer betydningfulle årsaker for overraskelsene i resultatene beskrives lengre ned i dette kapitlet. Men, de overnevnte forsinkelsene bør ikke helt oversees. Hovedgrunnen til dette er at [Cox97]¹⁹⁷ sier at forsinkelsene skal ikke bli særlig store før dette får konsekvenser for kvaliteten på talen når G.723.1A brukes som codec. Grunnen til dette er den store codec-forsinkelsen i G.723.1A. Det er usikkert hvor stor programforsinkelsen i `dgramServer` var under testene. I delkapittel 6.1.3 ble operert med et tall på 20 ms, men denne ble målt på den opprinnelige Linux-maskinen. Den nye Linux-maskinen var mye kraftigere enn den opprinnelige og således burde programforsinkelsen være mindre på den nye maskinen. På en annen side ble det kjørt IP-tables filtrering og gatekeeper i tillegg til `dgramServer` på den nye Linux-maskinen, det ble det ikke gjort på den opprinnelige. Derfor er det ikke sikkert at programforsinkelsen i `dgramServer` kjørt på den nye Linux-maskinen var forskjellig fra 20 ms.

Grunnen til at programforsinkelsen på den nye maskinen ikke ble målt, var fordi det ble nødvendig å få til en testplattform som tillot oppsett av forbindelser med H.323-signalering via proxy-en. Derigjennom ble det også nødvendig å modifisere `dgramServer`. Dette var et større arbeid som ble ansett å være på siden av prosjektets hovedfokus. Da mine lytte tester ende-til-ende ved vanlig telefoni viste seg å holde tilstrekkelig kvalitet, ble det derfor antatt at eksakte målinger av programforsinkelsen i `dgramServer` kjørt på den nye maskinen ikke var absolutt nødvendig da det var god margin i programforsinkelsen for den opprinnelige implementasjonen og fordi den nye maskinvareplattformen hadde en bedre ytelse.

De jevnt lave lytteinnsatsverdiene (MOS_{LE}) for pakketap indikerer også at talekvaliteten på talen ikke kvalifiserte seg til bruk i ATC av mer helsemessige årsaker, grunnet at det generelt krevdes for mye innsats for å forstå meningen med talen. Det å måtte lytte intenst for å få med seg innholdet i en tale, ville i lengden være for anstrengende for en flygeleder når en betydelig del av jobben går ut på å lytte til tale på heltid. Det er i denne sammenheng fullt forståelig at de ønsker en mest mulig avslappet lytteopplevelse, særlig når arbeidstiden antageligvis til tider er svært hektisk og stressende. Dette gjør det mer åpenlyst hvorfor kvalitets- og lytteinnsatskurvene følger hverandre såpass tett og hvorfor det ble et såpass stort neiflertall for pakketap. Følgende kommentar fra en flygeleder illustrerer dette:

¹⁹⁶ Eventuelle pakketap som kunne ha oppstått under talepakkenes ferd igjennom nettverket, kan muligens også ha påvirket resultatene i begrenset omfang.

¹⁹⁷ Referert til i kapittel 4.2.5.

”Et hakkete lydbilde er veldig anstrengende å jobbe med og derfor ikke under noen omstendighet akseptabel. En bruker for mye energi på å fange opp det som blir sagt. Til sammenlikning bruker man mye mer energi på å forstå en person med talevansker, fremfor én med normale taleevner.”

Ellers var det overraskende at pakkeforsinkelse på 200 ms fikk en lavere bedømmelse enn de andre forsinkelsesparametrene og at 600 ms ble bedømt til å holde høyest kvalitet. At alle forsinkelsesparametrene fikk lik ja- og nei-oppslutning var også oppsiktsvekkende. Ja-overvekten for pakkeforsinkelse, samt den jevne og høyere lytteinnsatsbedømmelsen for pakkeforsinkelse, indikerer i alle fall at testparametrene på dette området gir en lytteopplevelse som er bedre og som ligger nærmere det eksisterende systemet enn for pakketap. Det var også noen tilfeller hvor kurvene tar ”en avstikker” fra en ellers tilnærmet lineær kurve. Et eksempel på dette er graf 7.3, hvor kurvene er relativt lineære fra 15% pakketap og oppover. Det hadde her vært ventet en litt lavere nei-oppslutning (og tilsvarende litt høyere ja-oppslutning) for 10% pakketap, slik at kurvene hadde også hadde vært tilnærmet lineær fra 5%.

Årsaker til overraskelsene i resultatene kan være mange. Blant dem er følgende:

➤ **Subjektive testers natur**

Med subjektive tester kan man få til dels motstridende bedømmelser og kommentarer. Et slikt eksempel er en flygeleder som ga 200 ms en kvalitetsverdi (MOS) lik 4 og de andre forsinkelsesparametrene 5, mens vedkommende ga alle disse parametrene en lytteinnsatsverdi (MOS_{LE}) på 5. Andre godkjente alle forsinkelsesparametrene, mens andre igjen avfeide kvaliteten på de samme parametrene som for dårlige.

Det er naturlig at resultatene i slike tester gjenspeiler de ulike testdeltagernes individuelle meninger og dermed at resultatene også lett kan bli påvirket av avvikende meninger når antall respondenter er få. Blant annet ga kommentarer fra én flygeleder inntrykk av at vedkommende hadde bedømt talen mer utfra hvor autentisk den var og ikke utfra talekvaliteten, selv om om det i testinstruksjonen ble opplyst om at talen *illustrerer* ATC-tale. Det skal ikke mange slike avvikende bedømmelser til, som kanskje også er basert på feil grunnlag, før dette gir synlige innvirkninger på resultatet når antall respondenter er få. På denne måten kan det hende at sluttresultatene ikke gir et riktig bilde av den reelle talekvaliteten. For å få frem dette måtte kanskje antall respondenter være flere titalls ganger så stort. Blant annet ville det da ha vært mulig at ja-/nei-oppslutningene i graf 7.6 hadde sett anderledes ut.

➤ **Avspillingsrekkefølgen**

Avspillingsrekkefølgen på opptakene kan ha ”lurt” og forvirret testdeltagerne. Ettersom disse var omstokket kan flygelederne ha innbilt seg at noen opptak holdt en bedre kvalitet enn den egentlig gjorde, eller omvendt. Et eksempel er en flygeleder som ga 300 ms, 400 ms og 600 ms en kvalitetsverdi (MOS) på 5 i én serie, men ga de samme parametrene en kvalitetsverdi (MOS) på 4 i neste serie. Evalueringene ga også flere slike eksempler, slik at det er tydelig at det lyktes i å sørge for at flygelederne ikke oppfattet noe mønster i avspillingsrekkefølgen. De nevnte to seriene kunne selvsagt ha inneholdt kvalitetsmessige forskjeller, da opptakene ikke ble gjort på helt det samme tidspunktet og derfor kunne nettverksforholdene ha endret seg fra et opptak til et annet. Men en forskjell på *én* (av fem) på ”Listening-quality”-skalaen virker mistenkelig stort, dessuten ble alle

opptakene av én parameter gjort sekvensielt og umiddelbart etter hverandre. På denne måten skulle nettverksforholdene ha endret seg kun minimalt fra et opptak til et annet. Omstokkingen av opptakene kan ha påvirket resultatene, men samtidig kunne dette også ha vært tilfelle om avspillingsrekkefølge hadde vært av en slik art at flygeledeerne hadde sett et mønster i rekkefølgen..

➤ **Avspillingsutstyr**

Flygeledeerne ble i testinnstruksen oppfordret til å benytte hodetelefoner under lyttingen til opptakene for å få best mulig utbytte av lydopptakene. Slik ville man også få mest effekt av at stemmene var lagt i hvert sitt lydspor. Det lar seg ikke å gjøre og etterprøve at denne oppfordringen virkelig ble fulgt av flygeledeerne. Det er derfor vanskelig å si hva slags utstyr de brukte under avspillingen. Et par PC-høytalere ville for eksempel gi en annen lytteopplevelse enn om hodetelefoner ble benyttet. Avspillingsutstyret og kvaliteten på det kan således ha påvirket bedømmelsen.

➤ **Uprofesjonell innlesning**

Snakkingen på opptakene kan ha påvirket resultatene i og med at ikke-profesjonelle ble brukt under opptakene. Dette kan ha forstyrret flygeledeerne i deres bedømmelse da talen ikke var autentisk.

➤ **Opptakenes beskaffenhet**

Opptakenes beskaffenhet kan også ha påvirket resultatene. I og med at det var kun uproffesjonelt audioutstyr tilgjengelig i dette prosjektet kan dette ha påført opptakene mer forvrengning og støy og liknende enn det "live"-talen holdt. Dessuten ble det ikke brukt autentiske ATC-operatørposisjoner under opptakene.

➤ **MP3-konverteringen**

Konverteringen av lydopptakene fra wav-format til mp3-format mer enn halverte den samlede filstørrelsen¹⁹⁸ på lydopptakene. Totalt ble flygeledeerne bedt om å laste ned 30 mp3-filer, som hadde en total filstørrelse på 17 MB før zip-komprimering. Denne konverteringen kunne muligens gi en forringelse av lyd kvaliteten på lydopptakene i og med at mp3-formatet innebærer en komprimering av filstørrelsen. Videre kan kanskje kvaliteten på mp3-dekodere variere. Men, lydopptakene ble konvertert til et mp3-format med 128 kbps bitrate, hvilket ofte blir betraktet som "cd-kvalitet". Det ble valgt å satse på mp3-formatet da det, som tidligere nevnt, *ikke kunne høres noen forringelse* av lydopptakene på mp3-filene som følge av konverteringen. Det ble derfor antatt at mp3-formatet ga en kvalitet tilfredsstillende for formålet i dette prosjektet. Dessuten gjorde eksperimentformen det tvingende nødvendig å velge et tilfredsstillende lydformat som krevde minst mulig filstørrelser. Grunnen til dette var at flygeledeerne ble bedt om å *laste ned* lydfilene og det var usikkert hva slags Internett-forbindelse den enkelte flygeleder hadde tilgjengelig. For da å sikre at flygeledernes interesse og engasjement i prosjektet ikke skulle avta som følge av en tung og lang nedlastelsesprosess, ble det derfor valgt å satse på mp3-formatet.

➤ **Flygeledernes humør og innstilling til prosjektet**

Flygeledernes humør under bedømmelsene og deres innstilling til prosjektet kan også ha påvirket utfallet av testene. Blant annet ser [Agi01] problemer i at resultatene fra subjektive testinger i stor grad vil være avhengig av en del ukontrollerte faktorer som følge av testpersonenes humør, holdninger og kulturelle bakgrunn. I dette prosjektet gjelder dette spesielt for flygeledeerne i Avinor. For disse kunne det være en fare for at de følte seg "beordret" av overordnede til å

¹⁹⁸ Den totale filstørrelsen for wav-filene, det opprinnelige filformatet, var på cirka 40 MB.

delta, eller at de generelt hadde motforestillinger mot å delta fordi de for eksempel ikke likte å bruke arbeidstiden til å delta i dette prosjektet. Det må her understrekes at *ingen* av kommentarene fra flygelederne tydet på at de i utgangspunktet virkelig var negativt innstilte til prosjektet. Tvert imot, er det mitt inntrykk at de fleste av flygelederne som jeg var i direkte e-post korrespondanse med syntes at prosjektet var spennende og interessant. På en annen side var det noen som ikke ga noen kommentarer til bedømmelsen og som samtidig ga dårlige karakterer, det kan derfor ikke helt utelukkes at disse *potensielt kunne* ha vært negativt innstilte til prosjektet. Flygeledernes humør og innstilling under bedømmelsene kan dessuten ha blitt påvirket av det ikke var flygeledere som pratet på lydopptakene og at talen derfor ikke var 100% autentisk.

➤ **Nettverksforholdene**

Nettverksforholdene i det benyttete LAN-et er som tidligere nevnt neppe hovedkilden til overraskelsene i resultatene, de overnevnte punktene kan nok ha hatt større påvirkning. Det som eventuelt kunne tyde på nettverksforholdene kunne ha påvirket resultatene, er de to siste kommentarene nevnt i kapittel 7.1.3. Her blir det sagt at deler av lydopptakene er bra inntil det plutselig kommer et parti av talen hvor kvaliteten blir dårligere. Dette kunne eventuelt ha skyldtes at tilfeldig pakketap har rammet talen eller at talepakker har blitt såpass forsinket at de ikke har latt seg avspille. Om dette skulle være forårsaket av nettverksforhold, kan dette antageligvis ha oppstått i switchene ved at pakker har "kollidert" som følge av høy belastning. Likevel, disse faktorene kan naturligvis også ha vært forårsaket av de innstilte pakketapene og -forsinkelsene i dgramServer, eller av én eller flere av de overnevnte årsakene (spesielt uprofesjonell innlesning eller opptakenes beskaffenhet).

Denne plutselige forandringen i kvalitet er likevel en faktor som IP-telefoni i ATC meget vel kan bli utsatt for i et nettverk hvor annen trafikk konkurrerer med talen. Dermed kan det definitivt stilles spørsmålsteget om egnetheten ved IP-basert ATC om det ikke sikres mot dette. Resultatene gjenspeiler også dette, da denne talen ble sendt over et nettverk hvor talen potensielt kunne ha konkurrert med annen trafikk. Et klart flertall stemte i alle fall nei til at alle parametrene for pakketap kunne godkjennes. Dessuten ble ikke den høyeste gjennomsnittelige bedømmelsen for pakketap høyere enn 3,1. For konvensjonell bruk, kunne det likevel hende kvaliteten under slike forhold ville godkjennes. Disse relativt lave bedømmelsene i ATC-sammenheng vil derfor være en direkte følge av sikkerhetskravene som stilles til IP-telefoni i en slik sammenheng og verifiserer også antagelsen om de høye sikkerhetskravene i ATC.

I neste delkapittel skal potensielle svakheter i testingene drøftes ytterligere, hvor noen av overnevnte punkter også inngår.

8.3 Svakheter

I dette delkapitlet vil mulige svakheter ved dette prosjektet påpekes:

- Konversasjonodynamikk lot seg ikke bedømme. Dette kunne ha vært lettere å få til i det alternative oppsettet, hvor det var tiltenkt å benytte manus uten at svarene i konversasjonen var oppgitt. Når deltagerne i konversasjonen ikke vet hva de kommer til å høre, kan det være mulig at høye forsinkelser kan ha en annen påvirkning på konversasjonodynamikken og oppfattelsen av innholdet i talen. På en annen side, et slikt opplegg kan (som diskutert i kapittel 5) være sårbart om deltagerne ikke følger med i konversasjonen
- Et par kommentarer fra flygeledere i evalueringsskjemaene gikk på at talen på opptakene ble lest inn med for lav hastighet og ikke med riktig vektlegging av enkeltord. En skrev at flyplassbeskjeder uttales raskere enn talen på opptakene. Et par andre skrev at talen var ”hakkete” innlest og at det derfor var vanskelig å anslå hvorvidt det var kvaliteten på IP-telefoni som ga denne hakkingen. Det er vanskelig å bedømme hvor utslagsgivende dette vil være for talekvalitetsbedømmelsene, da det var kun noen få som ga uttrykk for dette. Likevel, dette kan ha vært en forstyrrende faktor da det er profesjonelle flygeledere som ga uttrykk for dette og som tydeligvis hadde forventet å høre ATC-tale uttalt av fagfolk. På en annen side, denne hakkingen kan like vel skyldes at noen ord i talen ble borte grunnet pakketap eller høye pakkeforsinkelser.

Det ble nevnt i kapittel 6 at omstokkingen i dgramServer måtte slås av da Linux-maskinen ikke håndterte belastningen godt nok. Eventuelt kan det også ha blitt introdusert hakking i talen fordi denne maskinen ikke klarte å håndtere talepakkene raskt nok. Den totale forsinkelsen som talepakkene ble utsatt for i Linux-maskinen, kan således ha medført at de ankom mottager for sent til avspilling hos mottageren.

Kanskje kunne en enda kraftigere Linux-PC ha håndtert belastningen bedre og dermed kunne potensielle følger for talekvaliteten av hardwarespesifikke grunner ha vært unngått. Aller helst burde det ha blitt benyttet en dedikert ruter med IP-telefoni kapabiliteter til å utføre jobben Linux-maskinen utførte. Det kan ikke forventes at en kraftig hjemme-PC skal kunne gi samme ytelser som en dedikert ruter. Det er de økonomiske rammene for dette prosjektet, som ble begrensningen i denne sammenheng.

- Et par kommentarer gikk på at den kvinnelige stemmen i opptakene hadde mer bakgrunnsstøy enn den mannlige og lurte på om det kunne være mikrofonteknikken. Dette kan nok like godt skyldes at innspillingsvolumet på den kvinnelige stemmen ble litt for lav og at det dermed ble mer susing i opptaket enn forutsatt (som omtalt i kapittel 6). Videre ble det benyttet uskjermete audiokabler, ”hjemmesnekrete” overganger og ikke de mest avanserte mikrofonene, hvilket kunne bidra til eventuell støy i opptakene. Støyen fra strømforsyningen til den ene laptop-en og dens innvirkning på talen er et godt eksempel på dette

- Det at talen ble sendt via et bedriftsnettverk gjør det vanskelig å anslå hva slags nettverksforhold talen møtte, og derigjennom hvor mange prosent pakketap eller millisekunder pakkeforsinkelse talepakkene reelt ble utsatt for.

8.4 Videre arbeid

I videre arbeid *bør* det benyttes flygeledere i konversasjonene som tas opp. Dette for å være sikker på at måten talen lest inn på opptakene ikke blir en forstyrrende faktor for bedømmelsene. Slik vil det også sikres at mikrofonteknikken ikke påvirker talekvaliteten. Videre ville man på denne måten også være sikre på at talen får den rytmen og den riktige vektleggingen av enkeltord som benyttes i ATC. En annen fordel med å benytte flygeledere under opptakene, vil være at for dem gir innholdet i ATC-talen mening. Det gjorde den ikke for deltagerne i konversasjonene i dette prosjektet. Flygelederne forventer en viss type svar på ulike henvendelser, dette kunne man utnytte ved å utelate svar i manus. På denne måten kan ATC-konversasjonen bli mer reell og kanskje blir det slik lettere å få illustrert påvirkningen av høye pakkeforsinkelser på konversasjonsdynamikken. Grunnen til dette er at flygelederne vil forvente et svar innenfor en viss tidsfrist. Kommer ikke et svar innen en viss tid, vil den ene parten bli nødt å gjenta. Ved bruk av flygeledere bør det også benyttes reelle operatørposisjoner og ”push-to-talk”-knapp for å gi en mest mulig autentisk ATC-tale.

For å oppnå best ytelse i fremtidige talekvalitetsbedømmelser, bør funksjonene i dgramServer implementeres for, og kjøres på, en dedikert *ruter* med IP-telefoni kapabiliteter. Slik kan det bedre sikres mot at unødvendig prosesseringsforsinkelser introduseres og dermed at talekvaliteten unødig forringes av dette. Alternativt bør det vurderes hvorvidt en mer effektiv datastruktur i dgramServer kan implementeres og om en enda kraftigere maskin bør brukes til å kjøre dette og annen tilhørende software. Dersom omstokkingsfunksjonaliteten i dgramServer ønskes benyttet i fremtidige forsøk, bør det også vurderes om talen skal sendes over helt isolerte nettverk, mellom to PC-er via en hub/ruter. Slik vil det være lettere å kontrollere de variable forsinkelsene og å hindre at annen trafikk konkurrerer med IP-telefonitalen.

Videre bør det i fremtidig arbeid vektlegges å benytte profesjonelt opptaksutstyr og kabler, noe liknende det som brukes i lydstudioer. Slik kan det med riktige måleinstrumenter sikres at innspillingsvolumet blir tilstrekkelig høyt, og generelt få bedre og renere digitale opptak. Videre bør det også benyttes skjermete kabler for å sikre at omkringliggende apparater ikke påvirker talesignalene.

I fremtidig arbeid bør det også vurderes å teste på flere parametre, slik som ekko og jitter, og det *bør* innhentes flere bedømmelser. På denne måten får man sikret at testresultatene ikke så lett blir påvirket av individuell synsing og motstridende bedømmelser. Ideelt sett burde det i videre arbeid benyttes objektive tester (omtales kort i appendiks B.2) for å være *helt* sikker på at synsingen ikke påvirker testresultatene og at disse *virkelig* gjenspeiler den reelle talekvaliteten. Et problem med en slik fremgangsmåte er at man ikke får nyttegjort seg av flygeledernes kompetanse. For å få verifisert eller avkreftet at individuell synsing i subjektive tester av IP-telefoni basert ATC påvirker testresultatene, kan det heller være tilrådelig at det blir foretatt *både* subjektive og objektive tester. På denne måten kan man finne ut i

hvilken grad synsingen eventuelt påvirker testresultatene, *samtidig* som man får dratt nytte av flygeledernes kompetanse. Slik vil det også være mulig å sammenligne resultatene med annen forskning som har brukt en lignende fremgangsmåte for vanlige IP-telefoni anvendelser for å sjekke i hvilken grad ATC-tale gir noen forskjell i resultatene. Med en slik fremgangsmåte kan man også undersøke i hvilken grad den objektive testingen og flygeledernes bedømmelser er forskjellig.

Kapittel 9

Konklusjon

Resultatene presentert i kapittel 7 gir ikke noen entydige svar på hvilken høyeste forringelse, f_{maks} , som gir en minste godkjente kvalitet, $\text{MOS}_{\text{grense}}$ – verken for testresultatene av pakketap eller pakkeforsinkelse. For pakketap ble ingen av testparametrene bedømt til å holde en kvalitet tilstrekkelig for ATC, dermed kan det heller ikke påpekes noen minste godkjente kvalitet. Det kan likevel trygt sies at pakketap opptil 5% har *potensiale* til å bli godkjent. I følge resultatene i dette prosjektet ville dette ha tilsvart en kvalitetsverdi (MOS) på 3.

For pakkeforsinkelse er det nesten like vanskelig å peke på en klar høyeste godkjente forringelse siden alle ble tildelt samme prosent ja-oppslutning. Den laveste gjennomsnittlige kvalitetsverdien (MOS) som ble utdelt for pakkeforsinkelse var 3,1. På denne måten kan det slås fast at kvalitetsverdier (MOS) over 3 er godkjent for ATC med pakkeforsinkelser i intervallet 200 ms til 600 ms. Til tross for dette viser tallene for lytteinnsats (MOS_{LE}) at den kvaliteten dette prosjektet klarte å fremskaffe på IP-telefonitalen krevde for mye lytteinnsats til at dette umiddelbart kan sies å egne seg for bruk i et intenst og krevende arbeid slik som flygeledelse. Dessuten er en nei-oppslutning på over 40% et for stort tall til å entydig kunne konkludere med dette. Selv om kvaliteten på pakketap opptil 5% ble underkjent med over 50% oppslutning, sier likevel ja-oppslutningen på over 40% at kvaliteten *ikke* er så håpløst dårlig at det ikke vil være mulig å få til en implementasjon som gir en kvalitet som kan bli funnet å være tilstrekkelig for ATC. Resultatene viser, derimot, klart at pakketap har en langt større negativ innvirkning på talekvaliteten enn pakkeforsinkelse.

Resultatene antyder også at ”Listening Quality”-skalaen i ATC-sammenheng gir et riktig bilde av den reelle talekvaliteten, og at hvor feilfri talen er har innvirkning på bedømmelsen og på forståelsen av innholdet i talen.

Til tross for at alle parametrene for pakkeforsinkelse ble godkjent, gir ikke dette noe grunnlag for å si klart ja til om talekvaliteten på IP-telefoni *generelt* egner seg for ATC. Til det er resultatgrunnlag altfor tynt. Det kreves flere enn 10 evalueringer for å kunne trekke en slik slutning. Dessuten det at *ingen* av parametrene for pakketap ble godkjent, gir i alle fall ikke noen belegg for å si at talekvaliteten på IP-telefoni er egnet, da pakketap vil kunne forekomme. Dermed vil det være nødvendig med ytterligere forskning som *kan* fastslå at kvaliteten på VoIP-basert ATC virkelig tåler en viss mengde pakketap og likevel kan tilfredsstillende sikkerhetskravene i ATC før det kan si ja til dette. *Konklusjonen blir derfor at kvaliteten på IP-telefoni som dette prosjektet kom opp med ikke er egnet for ATC, men at tester med et større budsjett, mer profesjonelt utstyr, flygeledere i opptakene og med flere individuelle evalueringer kan finne at IP-telefoni kan godkjennes for bruk i ATC.*

Appendiks A

Signaleringsprotokoller

Signalering er en av de viktigste funksjonene i telekommunikasjonsinfrastrukturen [LiM00], fordi det tillater ulike nettverkskomponenter å kommunisere med hverandre for å *sette opp og koble ned* f.eks. samtaler.

For å plassere et kall i IP-telefoni er signaleringsprotokoller påkrevet for å lokalisere parten(e) man ønsker å kommunisere med og varsle den/de om den ønskete kommunikasjonen [deC99]. Når mottagerne er funnet, må signaleringsprotokollene reservere ressurser og forberede forsendelse av mediainnholdet. De fleste signaleringsprotokollene konsentrerer seg først og fremst om ”*connection control*”¹⁹⁹ og ”*call control*”²⁰⁰ og blander seg ikke inn i overføringen av selve mediainnholdet. På denne måten blir ikke endepunktene som ønsker å kommunisere avhengig av at informasjonstrømmene må inntreffe i en tredjepart. Denne ekstra prosesseringen ville i verste fall kunne bidra til øket transmisjonsforsinkelse.

En betingelse for å oppnå en altomfattende aksept for IP-telefoni er en talekvalitet på nivå med tradisjonell fastlinje telefoni [DaF99], såkalt "Public Switched Telephone Network" (PSTN), hvilket innebærer ende-til-ende forsinkelse og klarhet i talen (viktige aspekter her er hvilke digitaliserings- og komprimeringsmetoder som brukes og antall forsinkete og tapte pakker). En annen er minst en "ease of operation" og funksjonalitet på nivå med PSTN, hvilket innebærer at signaleringen i IP-telefoni arkitekturen må minst gi de samme mulighetene²⁰¹ og egenskaper²⁰² som Signalerings System nr. 7 (SS7), som er signaleringsystemet som brukes i PSTN/ISDN²⁰³.

¹⁹⁹ Vedlikeholder forbindelsen.

²⁰⁰ Vedlikeholder selve kallet.

²⁰¹ På engelsk: "capabilities".

²⁰² På engelsk: "features".

²⁰³ Integrated Services Digital Network

Konkret betyr dette at signaleringsprotokollene må

- bidra med diverse funksjoner for å sette opp, håndtere og koble ned kall og forbindelser.
 - Av disse funksjonene, nevner [ScR99a] navneoversettelser og brukerlokalisering, forhandling av egenskaper (som tillater endepunktene å bli enige om et sett av parametre for mediautvekslingen), medlems håndtering²⁰⁴ (for å invitere nye deltagere eller å sette eksisterende deltagere på vent under selve kallet), og det å forandre egenskapene²⁰⁵ underveis i et kall som de viktigste.
- være skalerbar med hensyn på å støtte et antall endepunkter i størrelsesorden milliarder på verdensbasis og flere millioner samtidige kall.
- bidra med mekanismer for å formidle "Quality of Service" (QoS)-forespørsler fra endepunkter og sette opp disse.
- støtte diverse nettverks håndteringsegenskaper for blant annet policy-kontroll, bokføring²⁰⁶ og fakturering²⁰⁷.
- være mulig å utvide for lett å kunne implementere nye egenskaper.
- støtte interoperabilitet mellom ulike signaleringsprotokoller, mellom ulike versjoner av signaleringsprotokollene og mellom de ulike leverandørenes implementasjoner.

I litteraturen om signalering innen IP-telefoni er det først og fremst ”*International Telecommunication Union*” (ITU) sin ”*H.323*”-standard og ”*Internet Engineering Task Force*” (IETF) sin ”*Session Initiation Protocol*” (SIP) som omtales. Inntil 1996 var SIP den eneste protokollen for multimedia konferanser over Internettet, og var mye brukt i en del freeware og shareware konferansesoftware [LiM00]. Protokollen gjennomgikk mange forandringer før IETF godkjente den som en foreslått standard²⁰⁸ i mars 1999. ITU begynte standardiseringsarbeidet av en signaleringsprotokoll for IP-telefoni i mai 1995 som løsning på et problem som preget denne perioden av IP-telefoniens historie. Problemet besto i at hver leverandør av IP-telefoni software for PC-til-PC multimedia kommunikasjon hadde sine individuelle proprietære signaleringsprotokoller for opp-/nedkopling av kall. I juni 1996 var versjon 1 av H.323 ferdig, en protokoll som ofte blir referert til som en sanntids²⁰⁹ videokonferanseprotokoll beregnet for bruk over ”*Local Area Network*” (LAN) uten QoS-garantier.

Etterhvert som tjenestetilbydere mer og mer benyttet nettverk av H.323- gateway til å tilby VoIP-tjenester, ble det raskt funnet ut at H.323 versjon 1 hadde en del begrensninger. Spesielt gjaldt dette at H.323 antok at gateway-ene skulle håndtere signaleringskonvertering, kontroll av kall, transkoding av mediastømmene i én og samme boks. Dette ville kunne medføre skaleringsproblemer i storskala implementering [LiM00]. Dessuten tilbød ikke denne versjonen av H.323 støtte for SS7-forbindelser som kunne hindre sømløs integrering med PSTN. Som en løsning på dette ble det i mai 1998 introdusert en protokoll, ”*Media Gateway Control Protocol*” (MGCP), som innebar en todeling av gateway-en. Den ene delen fikk ansvar for kontroll av kall og ble kalt ”*Media Gateway Controller*”, mens den andre fikk ansvaret for transformering av media og fikk derfor navnet ”*Media Gateway*” [LiM00]. I juni

²⁰⁴ På engelsk: "call participant management".

²⁰⁵ På engelsk: "feature change".

²⁰⁶ På engelsk: "accounting".

²⁰⁷ På engelsk: "billing".

²⁰⁸ På engelsk: "proposed standard".

²⁰⁹ På engelsk: "real-time".

2000 definerte ITU og IETF i samarbeid en standard for kontroll av media gateway-er, kalt "*H.248*" (på ITU-terminologi) eller "*Megaco*" (på IETF-terminologi). For mer om H.248/Megaco bes leseren å konsultere [LiM00], ITU-T Recommendation H.248 eller IETF Request for comment (RFC) 3015.

Slik som nevnt ovenfor er H.323 utviklet og standardisert av ITU, mens SIP er utarbeidet av IETF. Disse protokollene gjenspeiler også forskjellene i motivet for de to organisasjonene i utviklingen av disse protokollene. IETFs utgangspunkt i sin tilnærming til multimediatelefoner var å opprette multimediakommunikasjon mellom to eksisterende datamaskiner klargjør for multimedia, mens ITUs utgangspunkt var å tilby allmenn kommunikasjon på tvers av avstander [Sch99]. ITUs filosofi innebar rimelig og enkelt brukerstyr, mens kompleksiteten skulle bli liggende i store og dyre sentraler. Protokollene er videre også preget av hvilken forretningsmessige tanke som ligger til grunn for organisasjonene. En av drivkraftene bak IETFs arbeid er teknologiforskning fra universiteter, og det eksisterer ingen utbredt tanke om å tjene penger på arbeidet [Sch99]. Det gir standarder med veldig god funksjonalitet. ITU, på sin side, drives mer kommersielt med en grunntanke man skal betale for de resursene som brukes og den tjenestekvaliteten man får levert [Sch99]. I tillegg, er *interoperabilitet* viktig for ITU, alle skal nemlig som minimum kunne føre en ordinær samtale uavhengig av hva slags utstyr som blir benyttet i endepunktene eller hvem som har laget utstyret [Sch99]. Likevel, den viktigste forskjellen mellom ITU og IETF, er at tradisjonelt har ITU arbeidet med telekommunikasjonsstandarder. IETF, derimot, er den ledende standardiseringsorganisasjonen innen datakommunikasjon [Sør00].

I dette vedlegget skal H.323 og SIP sees nærmere på med hovedvekt på arkitektur. Til slutt vil H.323 og SIP bli sammenholdt med hensyn på funksjonalitet, skalerbarhet, enkelhet, utvidelsesmuligheter og modularitet.

A.1 H.323

Siden kun tale er påkrevet i H.323 og alle andre medier er valgfrie, spiller H.323-terminalene en nøkkelrolle i IP-telefoni [Tri199]²¹⁰. En H.323-terminal kan enten være en PC eller en enkeltstående enhet som kjører H.323-stakken og multimediaapplikasjoner, og hovedhensikten med H.323 er samhandling med andre multimedia terminaler som for eksempel H.310-terminaler i B-ISDN eller H.320 i ISDN.

A.1.1 Arkitektur

De ulike komponentene i et H.323-nettverk (se figur A.1) har følgende funksjonalitet:

- Terminal er et endepunkt i nettverket som støtter toveis sanntids kommunikasjon med enten andre H.323-terminaler, gateway-er eller Multipoint Control Unit (MCU)-er. En terminal kan sette opp kall til andre terminaler enten direkte eller med hjelp fra en gatekeeper. H.323-terminaler må støtte G.711 Pulse Code Modulation (PCM) audio-codec, mens støtte av video codec-er er valgfritt.
- Gatekeeper er administratoren i sonen. På grunn de sentraliserte håndterings²¹¹- og kontrolltjenestene den tilbyr blir den ofte sett på som intelligensen i H.323-nettverket [Kar99]. Gatekeeper-en gjør adresseoversettelser mellom for eksempel telefonnumre og nettverksadresser, og kontrollerer tilgangen til nettverket for terminalene, gateway-er og MCU-er. Gatekeeper-en kan også tilby andre tjenester slik som båndbreddehandtering og lokalisering av gateway-er.

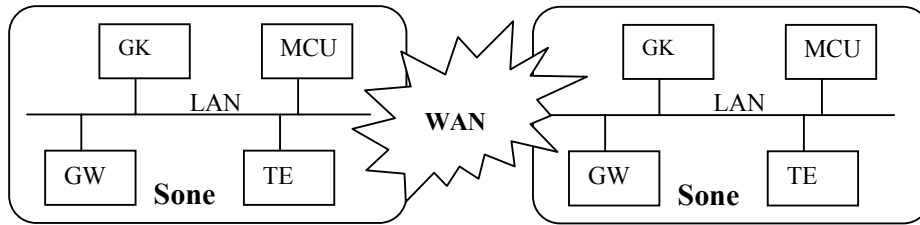
En valgfri funksjon er ruting av kallsignaler, hvor gatekeeper-en ruter kallsignaleren til destinasjonsendepunktet. Om denne funksjonen ikke velges kan endepunktene sende kallsignaleringsmeldinger direkte til likeverdige²¹² enheter. Men [Tri199] påpeker viktige fordeler ved denne rutingsfunksjonen i form av bedre kontroll med kallene i nettet og bedre ytelse, ettersom gatekeeper-en kan ta rutingsavgjørelser utfra en del faktorer (slik som lastbalansering av gateway-ene). Gatekeeper-ens funksjon er valgfri i H.323, men når den først er tilstede må terminalene, gateway-ene og MCU-ene registrere seg hos den.

- Gateway er et H.323-endepunkt som tilbyr toveis kommunikasjon mellom terminaler på det H.323-baserte nettet og terminaler i andre typer nett, slik som PSTN. Gateway-ene utfører "call setup" og kontroll på både det pakkebaserte nettet og det linjeswitchete nettet, og kan oversette mellom ulike overføringsformater og mellom ulike kommunikasjonsprosedyrer [DaF99]. Noen gateway-er kan dessuten også oversette mellom ulike audio- og/eller video-codec standarder, såkalt transkoding, for å minimere båndbredden mediastråmmene beslaglegger om båndbredden i det linjeswitchete nettet er begrenset. Gateway-en er ikke påkrevet så lenge det ikke er noen forbindelser til andre nettverk.

²¹⁰ Ikke trykket materiale, funnet på <http://www.iec.org>

²¹¹ På engelsk: "management".

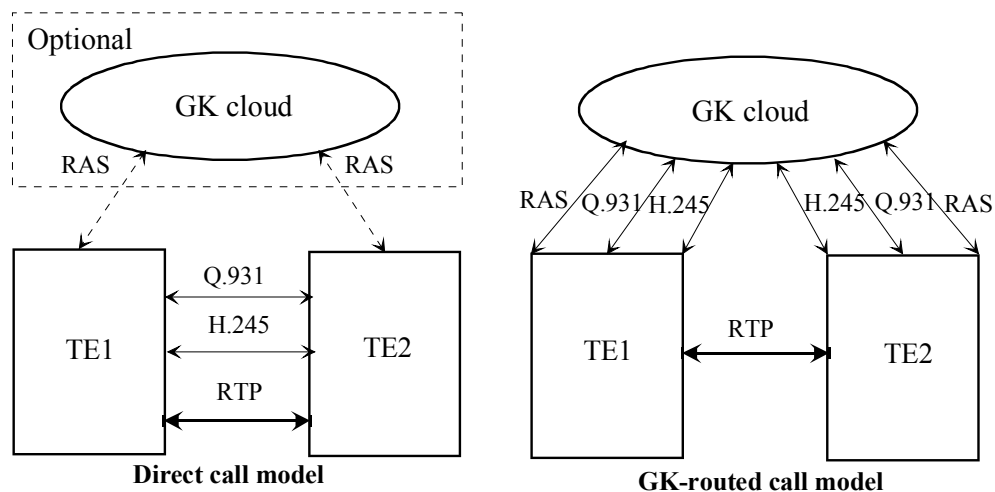
²¹² På engelsk: "peer".



Figur A.1: Et H.323 nettverk bestående av soner koblet sammen via WAN [LiM00]. Hver sone består av minst ett terminal endepunkt (TE), "Multipoint Control Unit" (MCU), et antall gateway-er (GW) og kun én valgfri gatekeeper (GK) som er koblet sammen over et LAN. (Basert på [LiM00].)

- Multipoint Control Unit (MCU) tillater tre eller flere terminaler og gateway-er til å delta i konferanser. MCU-en består av en "Multipoint Controller" (MC), og null eller flere "Multipoint Processor" (MP). MP utfører den nødvendig prosesseringen av mediastrømmene som inngår i konferansen, typisk miksing av lyd og switching mellom lyd og video. MC forhandler mellom endepunktene i konferansen for å oppnå enighet om hvilke audio- og/eller video-codec-er som skal benyttes og for å bestemme felles egenskaper for prosesseringen av audio og video [DaF99]. MC kan også brukes i et punkt-til-punkt kall, som senere kan bli utvidet til en konferanse.

Et H.323-nettverk er typisk bygget opp av flere såkalte soner [LiM00], som er koblet sammen via et "Wide Area Network" (WAN) (se figur A.1). Hver sone består av minst ett terminal endepunkt (TE) [Tri199] og kan bestå av et antall gateway-er (GW) og Multipoint Control Unit (MCU) koblet sammen over et LAN. Om gatekeeper (GK) inngår i nettverket, denne er valgfri, kan det kun være én av denne. En sone kan strekke seg over flere LAN eller kun gjelde innenfor et eneste LAN. Gatekeeper, gateway og MCU er logiske separate komponenter, men kan likevel bli implementert i én fysisk enhet [Tri199]. Videre benevnes ofte terminaler, gateway og MCU-er kollektivt som endepunkter [Kar99]²¹³.



Figur A.2: Det støttes to typer modeller for "call kontroll", "direct call model" (til venstre) og "Gatekeeper (GK)-routed call model" (til høyre), i H.323. (Basert på [LiM00].)

²¹³ Ikke trykket materiale, funnet på <http://www.cis.ohio-state.edu/~jain/cis788-99/h323/index.html>

I figur A.2 er det vist de to "call control"-modellene som støttes i H.323, "direct call model" og "Gatekeeper-routed call model". I den direkte modellen blir signalering (H.225.0 Q.931- og H.245-meldinger) og selve mediastrømmen (Real-Time Protocol, RTP) utvekslet direkte mellom to terminaler (TE1 og TE2) [LiM00]. Gatekeeper-skyen og "Registration, Admission and Status" (RAS)-signalering er valgfrie i denne modellen (H.225.0 RAS, H.245, H.225.0 Q.931 og RTP vil bli sett nærmere på i neste delkapittel), så lenge den oppkallende parten vet transportadressen til den oppkallede parten kan den sette opp en forbindelse direkte dit [LiM00]. Dette er en "call control"-modell mye brukt i en PC-til-PC arkitektur (se kapittel 2) for gratis samtaler over Internett. Om gatekeeper-en brukes kan den oppkallende parten be om adresseoppslag fra sin gatekeeper, mens den oppkallede parten kan be om tillatelse fra sin gatekeeper om å motta kallet. Denne modellen har begrensinger i storskala-sammenheng, da tjenestetilbydere ikke har kontroll over antall kalloppsett og kan derfor bli hindret i å tilby nok ressurser og ta betalt for kallene [LiM00]. I "gatekeeper-routed call model" blir alle signaleringsmeldinger rutet via gatekeeper-en, dermed blir også RAS-signalering nødvendig [LiM00]. Dette er nødvendig for at terminalene blant annet skal kunne finne en gatekeeper og registrere seg hos den. Gatekeeper-skyen tilbyr adresseoppslag og ruting av kall til endepunktene, samtidig gir denne modellen gatekeeper-en mulighet til å utøve tilgangskontroll og allokering av båndbredde i sonene den dekker. Dette er en attraktiv modell for IP-telefoni tilbydere, fordi dette gir dem mulighet til å kontrollere nettverket, og utøve bokføring og fakturering [LiM00].

A.1.2 Protokoller i H.323-paraplyen

H.323-paraplyen består av fire protokoller:

- H.225.0 Registration, Admission and Status (RAS)
- H.225.0 Call Signaling (subsett av signaleringsprotokollen Q.931 som brukes i ISDN. Denne protokollen er benevnes ofte Q.931 eller H.225 Q.931, deriblant i figur A.1 og A.2)
- H.245 Control Signaling
- Real-Time Transmission Protocol (RTP).

Tabell A.1 viser gangen i oppsettet og nedkobling av et kall mellom to endepunkter via en gatekeeper (se "gatekeeper-routed call model" i figur A.2), og i hvilke faser de overnevnte protokollene benyttes. Hensikten med tabellen er å gi en oversikt over hva de ulike protokollene brukes til, før de blir nærmere beskrevet utover i kapitlet. Delkapittelet avrundes med en beskrivelse av tilleggstjenestene definert i H.450.X. For inngående informasjon om de faktiske meldingene som utveksles, se [Agi00a] og [Fin02].

1	"Gatekeeper discovery" og registrering av endepunkt.	H.225-RAS
2	Ruting av kall signalering mellom terminalene via GK.	H.225-RAS H.225-Q.931
3	Innledende kommunikasjon, hvilket innebærer detektering av "master/slave"-forhold, og kapabilitetsutveksling.	H.245
4	Etablering av audio-kommunikasjonen, deriblant åpne den logiske kanalen.	H.245
5	Overføring av audio.	RTP/RTCP
6	Lukking av de logiske kanalene ("Channel closing").	H.245
7	Nedkobling av kallet ("Call teardown").	Q.931
8	Frigjøring av ressurser som var beslaglagt under kallet ("Call disengage").	H.225-RAS

Tabell A.1: Gangen i oppsettet av kall i H.323 for overføring av tale, og hvilke protokoller i H.323-stakken som involveres (basert på [Agi00a]). De tre nederste radene viser nedkoblingen av et kall (basert på [LiM00]).

A.1.2.1 H.225.0 RAS

"Registration, Admission and Status" (RAS) er en transaksjonsorientert protokoll (det vil si at en forespørsel til en mottager blir prosessert og besvart med en respons tilbake) som brukes mellom et endepunkt (enten en terminal eller en gateway) og en gatekeeper.

Endepunktene bruker RAS for å

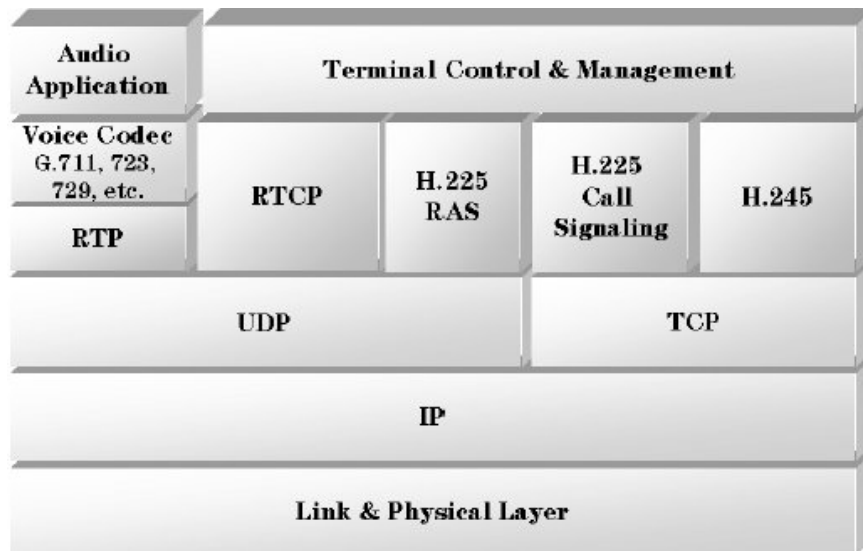
- finne en gatekeeper ("Gatekeeper Discovery").
- registrere seg hos eller stryke seg fra listen i en gatekeeper.
- be om tilgang til å etablere et kall²¹⁴.
- få tilordnet båndbredde.
- koble ned et kall.

Videre er det også innebygget mekanismer for at en gatekeeper kan lokalisere et endepunkt, få statusinformasjon om et endepunkt og for flere gatekeeper å kommunisere seg imellom på tvers av soner for å utføre adresseoppslag²¹⁵. H.255.0 RAS er kun nødvendig når en gatekeeper eksisterer i nettet [Kar99], og baserer seg på en upålitelig²¹⁶ transportprotokoll. I IP-baserte nett baserer RAS seg følgelig på User Datagram Protocol (UDP) (se figur A.3).

²¹⁴ På engelsk: "call admission".

²¹⁵ På engelsk: "address resolution".

²¹⁶ På engelsk: "unreliable". Ingen garantier for levering av datagrammene, ingen kvitteringer og uordnet levering.



Figur A.3: H.323-protokollstakk, populært også kalt "H.323-paraplyen" siden H.323 definerer et sett av protokoller. Hentet fra [Pra00].

A.1.2.2 H.225.0 Call Signaling

"Call Signaling" er en grunnleggende nødvendighet for det å sette opp og koble ned kall mellom to endepunkter [Kar99]. H.225.0 bruker et subsett av signaleringsprotokollen brukt i ISDN²¹⁷ til dette formålet. Denne integreres i H.323 via meldingsformatet for H.225.0. Signaleringsmeldingene sendes direkte mellom endepunkter såfremt ingen gatekeeper eksisterer i nettet. Skulle en gatekeeper eksistere i nettet, kan disse meldingene sendes via gatekeeper-en. Disse meldingene vil inneholde informasjon om kall kontroll og kontroll av tilleggstjenester [DaF99]. Når kallet først er etablert vil transportadressen til H.245 "Control Channel" (se neste delkapittel) sendes over H.225 Call Signaling-kanalen. [LiM00] trekker frem enkel samhandling med PSTN og andre linjeswitchete konferansestandarder som årsaken til at H.323 har innebygget kompatibilitet med signaleringsprotokollen i ISDN. Denne samhandlingen med PSTN var ønskelig fordi, på det tidspunktet første versjon av H.323 kom ut, ble det å unngå fjerntakst sett på som den største pådriveren for *Voice over IP* (VoIP). Dette foregikk, som beskrevet i kapittel 2, ved å ringe via et datanettverk til en aktuell landsdel for deretter å ringe lokaltakst i den gjeldende regionen²¹⁸.

A.1.2.3 H.245 Control Signaling

H.245 brukes for å kontrollere forbindelsene, hvilket tillater endepunktene til å forhandle om hvilke mediaprosesserings-egenskaper (for eksempel audio/video codec-er) som skal gjelde for hver av forbindelsene dem imellom. I tillegg til denne kapabilitetsutvekslingen, brukes H.245 i H.323-sammenheng (H.245 er en fellesprotokoll for alle protokollene i H-serien) også for å bestemme "master/slave"-forhold mellom endepunkter, for å åpne og lukke logiske forbindelser for overføring av mediastrømmene mellom to endepunkter og for flytkontroll.

²¹⁷ ITU-T Recommendation Q.931.

²¹⁸ På VoIP-terminologien omtalt som "toll bypass".

A.1.2.4 RTP

H.323 bruker ”*Real-Time Transmission Protocol*²¹⁹” (RTP) som transport protokoll for ende-til-ende levering av sanntids audio/video [Tri199] (se figur A.2). Hver mediatype fraktes i et par unidireksjonelle logiske kanaler, én for hver retning, ved bruk av RTP og Real-Time Control Protocol (RTCP), som kontrollerer transporten av real-time mediene [DaF99]. RTP igjen kjøres over UDP (se figur A.2), som sammen gir transportprotokoll funksjonaliteter [Tri199]. I det følgende skal det sees litt nærmere på RTP og RTCP, med hovedvekt på hvilke funksjonaliteter de tilbyr.

For å kunne transportere sanntids mediestrømmene, må bitstrømmene generert av codec-ene deles opp i passe lengder og tilpasses RTP- pakkene før pakkene sendes over til mottageren, hvor den originale bitstrømmen gjenskapes [ScR99b]. Dette er en komplisert prosess, hvor pakker kan tapes, omstokkes og forsinkes. For å mest mulig unngå dette, er det definert en del funksjonaliteter:

- *Sekvensnummer* brukes for å detektere tap og for gjenskaping av mediestrømmen.
- *Intermedia synkronisering* tilbyr de nødvendige tidsstemplene som er nødvendig for bruk av jitterbuffer for å kompensere for jitter.
- *"Payload Identification"* for å dynamisk kunne forandre kodingsformen på det overførte mediet (altså lasten i RTP-pakken), ettersom nettverksforholdene forandres under transporten. Dette bunner ut i at enkelte codec-er har varierende evne til å fungere normalt under visse tap- og forsinkelsesforhold.
- *"Frame Indication"* brukes for å indikere for mottageren hvor en taleramme starter eller slutter. Et "framebit" markerer dette. Hensikten er å gjøre synkroniseringen lettere for levering av talerammene til protokoller på høyere lag.
- *"Source Identification"* indikerer hvilke av deltagerne i en konferanse som sendte en pakke.

En viktig bemerkning om RTP er at den ikke reserverer ressurser i nettet for å unngå tap og jitter, men at den, derimot, legger forholdene til rette for at mottageren kan gjenskape mediestrømmene under slike nettverksforhold. Et annet poeng er sammenhengen mellom signaleringsprotokollene og RTP; signaleringsprotokollene etablerer de parametre som skal gjelde for RTP-transporten [ScR99b]. Altså signaleringen setter opp forbindelsen hvor RTP-pakkene skal fraktes over.

RTCP er RTP sin "samarbeidsprotokoll", som tilbyr følgende kontrolltjenester:

- *QoS-tilbakemeldinger* fra mottagere til hver av avsenderne om hvordan de opplevde kvaliteten på mottatte mediestrømmer. Disse tilbakemeldingene inneholder informasjon om antall tapte pakker, jitter og "roundt-trip"-forsinkelse.

²¹⁹ For mer informasjon, se IETF RFC 1889.

- *Intermedia synkronisering* brukes for å synkronisere lyd og bilde i de tilfeller hvor lyd og video sendes over separate pakkestrømmer. Dette lar seg også gjøre når disse pakkestrømmene har ulike kilder.
- *Identifisering* brukes for at sesjonsdeltagere skal kunne få vite identiteten til andre deltagere. Identifiseringen inneholder informasjon slik som epost-adresse, telefonnummer og fullt navn.
- *Sesjonskontroll* tillater deltagere i en sesjon å indikere at de er i ferd med å forlate sesjonen og sende små meldinger til hverandre.

RTCP krever at alle sesjonsdeltagere, både de som sender og de som lytter, periodisk må sende ut pakker som inneholder de overnevnte informasjonene. Disse pakkene sendes til samme adresse som RTP-pakkene, men innpå en annen port. Grunnen til at disse pakkene må sendes ut periodisk, er blant annet fordi at informasjonen i disse pakkene vil kun være gyldig i en kort periode, da nettverksforholdene vil endres med tiden. Derigjennom vil blant annet mottatt kvalitet (hos en av deltagerne i et kall) kunne endre seg kontinuerlig som følge av endrete nettverksforhold.

A.1.2.5 H.450.X Tilleggstjenester

H.323 versjon 1 definerte kun grunnleggende kall kontroll og signalering for å sette opp multimedia konferanser med flere deltagere, og tok ikke for seg avanserte tjenester. Derfor opprettet ITU-T H.450-serien for å tilby avanserte tjenester liknende tilleggstjenestene²²⁰ i PSTN [LiM00]. Med H.323 versjon 2 ble tre H.450-protokoller definert (se figur A.4):

- H.450.1: En generell funksjonell protokoll på toppen av Q.931 gjeldende for alle tilleggstjenester. Denne er en ende-til-ende signaleringsprotokoll avledet fra QSIG-protokollen²²¹ som brukes for sammenkoblinger av "Private Branch Exchange" (PBX). Således kan tilleggstjenestene i H.323 sees på som en tilpasning av PBX-tjenester til IP-domenet [LiM00].
- H.450.2: "*Call Transfer*", en funksjon for å sette over kall til en tredje part.
- H.450.3: "*Call Diversion*" inneholder ulike varianter for viderekobling²²² av kall.

I versjon 3 av H.323 ble ytterligere fem nye tjenester lagt til:

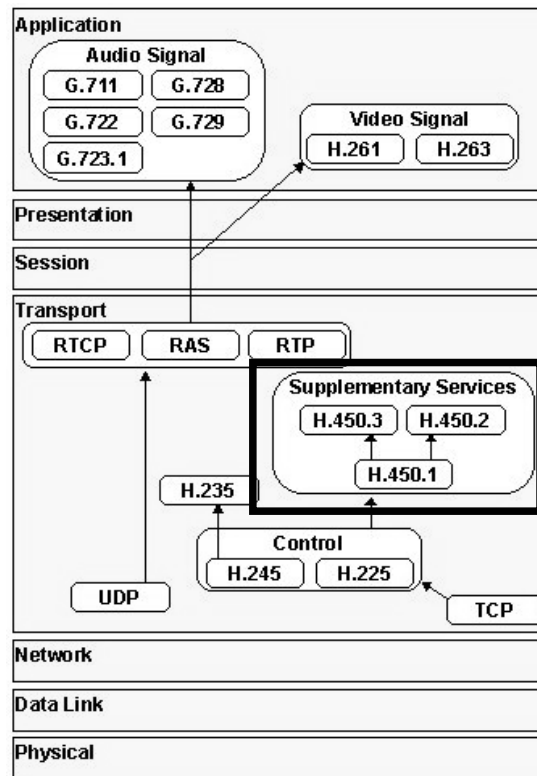
- H.450.4 "*Call Hold*", en funksjon for å sette et kall på vent.
- H.450.5 "*Call park and pickup*". "*Call park*" er en funksjon for å sette en samtale på vent og gjenopprette samtalen fra en annen telefon. "*Call pickup*" er en funksjon for å besvare en annen ringende telefon i en såkalt "*call pickup group*".
- H.450.6 Indikerer at en beskjed venter.
- H.450.7 "*Call Waiting*", samtale venter.

I løpet av år 2000 jobbet ITU med enda ytterligere fem tilleggstjenester til versjon 4, deriblant H.450.8 navne identifikasjon.

²²⁰ "Supplementary Services".

²²¹ For mer informasjon, se <http://www.qsig.ie/qsig/index.htm>.

²²² "Call Forwarding".



Figur A.4: H.323 versjon 2-stakken og OSI-modellen. Plassering av tilleggstjenestene (svart ramme) i stakken. (Hentet fra [Pro01]²²³.)

Da H.450 er en ende-til-ende protokoll, innebærer dette at begge terminalene i en sesjon må skjønne tjenestelogikken for at tilleggstjenestene skal fungere. Videre kreves det at terminalene også kjører det meste av logikken, hvilket innebærer en avsporing fra PSTN hvor logikken er implementert i switchene og ikke i telefonene. Dette kan by på problemer for storskala implementeringer, hvor terminalene har implementert ulike versjoner av H.323 [LiM00]. En annen ulempe med H.450-baserte tjenester som også trekkes frem i [LiM00], er at nye tjenester må spesifiseres av ITU og at terminalene må oppgraderes før en tjeneste kan igangsettes. Dette vil kunne forsinke igangsettelse²²⁴ av nye tjenester. En løsning som foreslås er å implementere tjenester i gatekeeper-ene. Disse tjenestene vil sannsynligvis være proprietære, men når VoIP-industrien modnes ytterligere vil det trolig etter hvert komme standardiserte tilnærminger til denne løsningen [LiM00].

²²³ Ikke trykket materiale, hentet fra URL: <http://www.protocols.com/pbook/h323.htm> 7/11-01. Figuren viser et utsnitt av bildet på nevnte URL. Den svarte rammen er tegnet inn.

²²⁴ På engelsk: "deployment".

A.2 SIP

”Session Initiation Protocol²²⁵” (SIP) brukes til å initialisere sesjoner mellom brukere, og dens fremste styrke ligger i å lokalisere brukere [ScR99b]. Andre tjenester som SIP tilbyr er etablering av kall²²⁶, håndtering av kallets medlemmer²²⁷ og begrenset muligheter for aktivering av egenskaper²²⁸. SIP har valgt epost-liknende adresser på formen [bruker@domene](#), [bruker@host](#), [bruker@IP_adresse](#), eller [telefonnummer@gateway](#). Disse adressene brukes i SIP-URL, som for eksempel `sip:o.nordmann@berget.no`, og kan eventuelt plasseres på en hjemmeside, slik at et klikk på den vil starte et kall til den adressen.

Det er ikke definert i SIP hva slags sesjoner som skal opprettes, slik at den kan opprette alt fra spillsesjoner til audio-/videokonferanser. SIP gjør minimale antagelser om underliggende transportprotokoll. Det eneste kravet er at enten må SIP-forespørsler og responser leveres hele, eller ikke bli levert i hele tatt [ScR98a]. [ScR00] trekker frem at siden SIP gjerne kan kjøre over UDP, må den selv sørge for pålitelighet. Invitasjonsforespørselene er litt spesielle, da det kan gå flere sekunder fra forespørselene ankommer en tjener til en endelig respons blir sendt tilbake (som for eksempel ved å ta av røret). For å løse pålitelighetsproblemet i dette scenarioet, kan klienter retransmittere invitasjonsforespørsler inntil en foreløpig respons ankommer, og tjenere kan retransmittere responser inntil de blir bekreftet.

A.2.1 Overblikk

SIP er en klient/tjener-basert protokoll som har arvet en del syntaks og semantikk fra ”Hyper Text Transport Protocol” (HTTP). Dette innebærer at meldingsutvekslingen er transaksjonsorientert. Responser i SIP følger samme sett av tjenere, som en forespørsel var innom på vei til mottageren, tilbake til avsenderen av forespørselen. Dette lar seg gjøre fordi hver tjener en forespørsel er innom, registrerer sin adresse og portnummer i et VIA-felt i SIP-hodet. Hver SIP-forespørsel består av en rekke felter i hodet som beskriver selve kallet som helhet, etterfulgt av en beskrivelse av de enkelte mediasesjonene som utgjør kallet. I dag er det ”Session Description Protocol²²⁹” (SDP) [SDP98] som brukes til det sistnevnte [ScR99a]. Initialiseringen av en sesjon er avhengig av den oppkallede partens tilgang til god nok informasjonen om den, slik at han/hun kan ta en avgjørelse om delta i eller avstå fra sesjonen [ScR00].

I endepunktene er det i SIP implementert en software, kalt "User Agent", som samhandler²³⁰ med brukerne. En "User Agent" er todelt, én "User Agent Client" (UAC) og én "User Agent Server" (UAS). I typiske IP-telefoni sammenheng vil en bruker ha både en UAS og en UAC, da en bruker både kan motta kall og selv kan initialisere nye kall. Altså, en UAS besvarer kallforespørsler, mens en UAC har ansvaret for å etablere nye kall.

²²⁵ IETF Proposed Standard Request for Comment (RFC) 2543.

²²⁶ På engelsk: "call establishment".

²²⁷ På engelsk: "call participant management".

²²⁸ På engelsk: "feature invocation".

²²⁹ IETF RFC 2327.

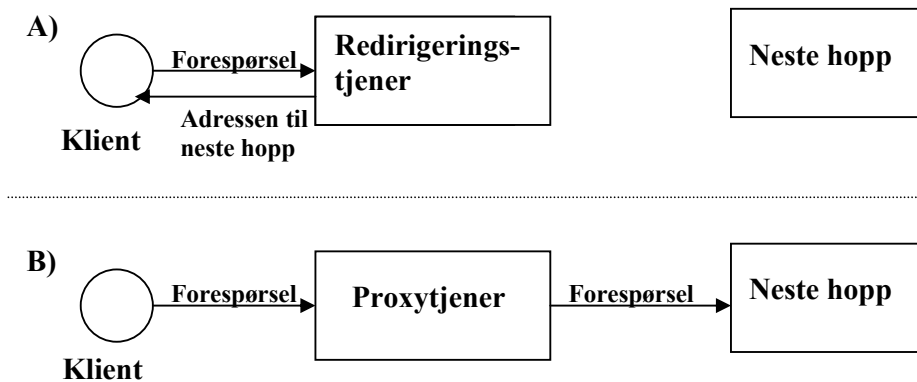
²³⁰ På engelsk: "interacts".

A.2.2 Tjenere

I SIP er det definert tre typer tjenere; registrerings-, proxy- og redirigeringstjenere.

Redirigeringstjeneren mottar en forespørsel og bestemmer neste hopp for forespørselen. I motsetning til proxy-tjeneren, returnerer redirigeringstjeneren adressen til nestehopp-tjeneren (som har mer nøyaktig informasjon om den oppkallede parten) tilbake til klienten (se figur A.5a). Proxy-tjeneren, derimot, videresender selve forespørselen til nestehopp-tjeneren (se figur A.5b). Felles for både proxy- og redirigeringstjeneren er at deres hovedoppgave er ruting av kall²³¹, dvs å bestemme det settet av tjenere som forespørselen må traversere for at kallet skal kunne fullføres. For å oppnå dette kan de benytte alle tilgjengelige midler, som for eksempel kjøre programmer og gjøre oppslag i databaser. I tillegg har proxy-tjeneren "fork"-egenskapen, dvs at den kan videresende en forespørsel til mange mottagere og på denne måten få prøvet ut mange lokasjoner samtidig for å finne den oppkallede parten. Den tjeneren som vet den gjeldende lokasjonen til den oppkallede parten, blir koblet opp mot denne.

Registreringstjeneren blir oppdatert med brukerens lokasjoner, via registreringsmeldinger. (I kapittel A.2.3 vil meldinger bli nærmere behandlet.)



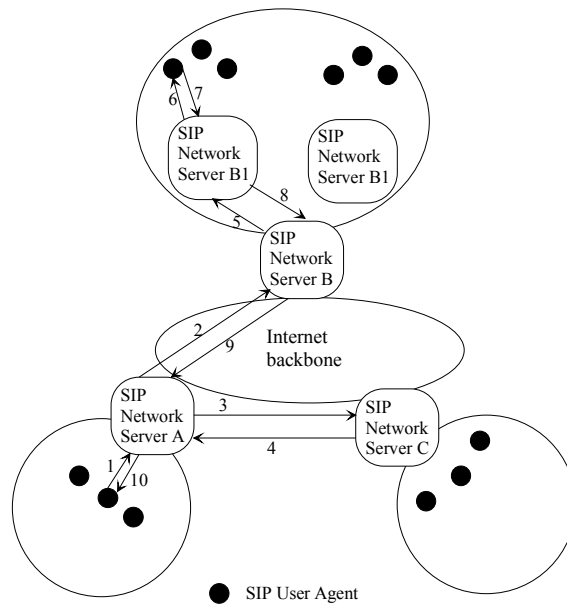
Figur A.5: Redirigeringstjeneren returnerer adressen til nestehopp-tjeneren tilbake til klienten (A), mens proxytjeneren sender selve forespørselen videre til nestehopp-tjener (B).

En SIP-proxy eller -redirigeringstjener har også mulighet for å veksle mellom å holde tilstander eller ikke²³², dette kan varieres fra kall til kall av en administrator [ScR98b]. Grunnen til dette er at INVITE-meldingen (denne og flere meldinger vil bli sett på i neste delkapittel) inneholder en adresse som en UAC kan bruke for å kommunisere med en UAS. For eksempel kan en stor, sentralisert SIP-tjener være tilstandsløs mens en mindre, lokal kan holde tilstander. En tilstandsløs tjener vil her motta en forespørsel og enten videresende den (til den ene adressen angitt i VIA-feltet i hodet, den eneste påkrevete tilstanden) eller generere en respons. Deretter "glemmer" den alt. På denne måten oppnås god skalerbarhet og pålitelighet, da kjøring av feilmodier og normalt virke fungerer med en fast mengde minne. Dessuten vil en tjener kunne gå ned og komme seg igjen (eller at en "backup"-server kan bli satt inn) uten at kallene initiert igjennom den blir affektert [ScR98b].

²³¹ På engelsk: "call routing".

²³² På engelsk: "statefull" og "stateless".

Figur A.6 viser et typisk SIP-tilfelle med tre domener (A, B og C), som hver har en aksesstjener som styrer trafikk inn og ut av deres nettverk. En UAC i domene A vil lokalisere og plassere et kall til en annen bruker (heretter: X). Kall-invitasjonen blir videresendt til A sin aksesstjener (1), som prøver å finne den oppkallede X i domene B og C ved hjelp av såkalt ”forking”. Forespørslene ankommer henholdsvis domene B (2) og C (3). Siden X er ukjent i C, returneres en feilmelding tilbake (4). I domene B, derimot, er X kjent og forespørselen blir videresendt til en lokal tjener (5), hvor den ønskete UAS-en blir funnet (6). Responsen fra X blir videresendt langs den samme ruten tilbake til den orginerende parten (7,8,9,10).

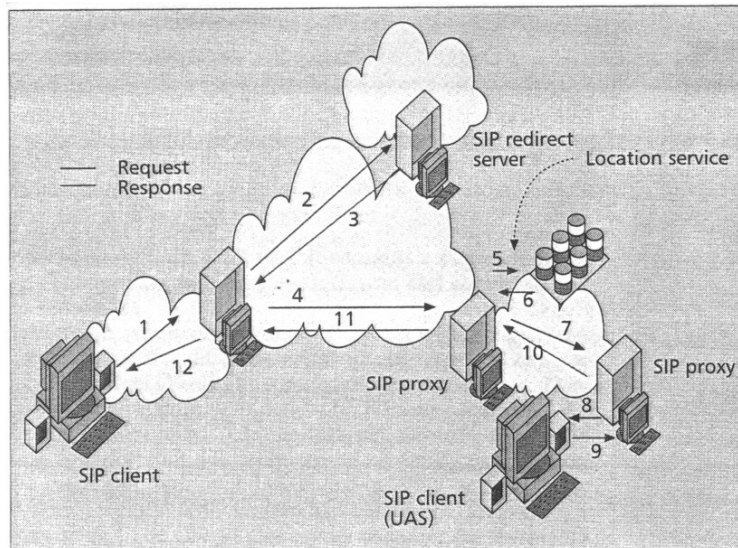


Figur A.6: Et typisk tilfelle med SIP hvor en bruker ønsker å finne og kommunisere med en annen bruker i et annet domene. Basert på [ScR98b].

[ScR99c] gir et tilsvarende eksempel som det vist i figur A.6, men her illustreres bruken av proxy- og redirigeringsstjenere bedre. Dette eksemplet er gjengitt i figur A.7, hvor en UAC oppretter en INVITE-forespørsel som skal sendes til sip:joe@company.com (meldingsformatet på INVITE-forespørselen blir sett nærmere på i neste delkapittel). Denne blir videresendt til den lokale proxytjeneren (1), som slår opp company.com i ”Domain Name Server” (DNS) og får slik greie på IP-adressen til den tjeneren som håndterer SIP-forespørsler i det domenet. Forespørselen sendes til denne tjeneren (2), som kjenner til Joe, men siden Joe i øyeblikket er logget inn som j.user@university.edu blir proxytjeneren omdirigert (3) til å prøve denne adressen isteden. Proxy-en gjør et nytt DNS-oppslag og sender forespørselen til SIP-tjeneren (heretter: Z) som betjener domenet til univeristy.edu (4). Denne tjeneren sjekker sin lokale database etter j.user@university.edu (5), som gir svar tilbake om at brukeren er kjent lokalt som j.smith@cs.univeristy.edu (6). Z sender så forespørselen til tjeneren som betjener cs.univeristy.edu (7), denne kjenner IP-adressen til maskinen brukeren er logget inn på, og videresender forespørselen dit (8). Brukeren aksepterer kallet og responsen følger samme ruten tilbake til den orginerende parten (9, 10, 11 og 12).

Dette eksemplet er veldig likt det i figur A.6, bortsett fra at dette også illustrerer personlig mobilitet, i tillegg til hvordan proxy- og redirigeringsstjenere samspiller. Figur A.6, på sin side, illustrerer ”forking”-egenskapene til SIP, hvilket eksemplet i

figur A.7 ikke gjør. Liknende eksempler med mer innflokke scenarier finnes i [ScR98b], [ScR99b] og i [MMTG00] gis en del eksempler på bruk av SIP i dagliglivet, slik som å styre en vekkerklokke.



Figur A.7: Et eksempel med SIP-transaksjoner, hvor samspillet mellom proxy- og redirigeringsjenere og personlig mobilitet illustreres. Hentet fra [ScR99c].

A.2.3 Meldinger

På samme måte som i HTTP vil klient-forespørsler aktivisere kall på metoder i tjeneren. Videre er forespørsler og responser tekstbaserte, og har feltet i hodet som inneholder egenskaper²³³ om kallet og tjenesteinformasjon [ScR98a]. SIP har arvet en del av feltene i HTTP-hodet, denne gjenbruken av kode bidrar til lettere integrering av SIP-tjenere med web- og mailtjenere.

```

INVITE sip:ola.nordmann@berget.no SIP/2.0
Via: SIP/2.0/UDP berget.no:5060
From: Kari <sip:kari.dunk@berget.no>
To: Ola <sip:ola.nordmann@berget.no>
Call-ID:12345600@berget.no
Cseq: 1 INVITE
Subject: Talekvalitet
Contact: Kari <sip:kari.dunk@berget.no>
Content-Type: application/sdp
Content-Length: 147
    
```

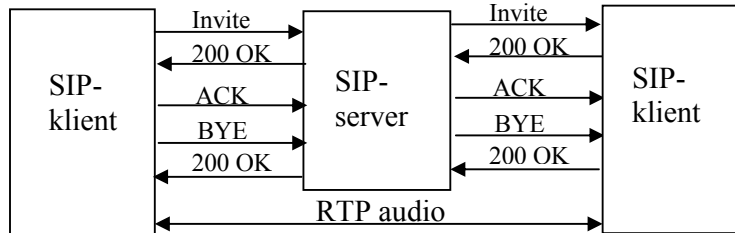
Figur A.8: Et eksempel på hodefelter i en INVITE-melding.

²³³ På engelsk: "properties".

Det er definert syv metoder i SIP:

- INVITE:** Inviterer nye deltagere til sesjonen. Denne metoden kan også brukes for å forandre nettverksadressen til destinasjonen for mediastrømmene, legge til/fjerne en mediastrøm til/fra sesjonen ved å sende en INVITE-forespørsel på nytt [ScR00]. De viktigste feltene i hodet er "To" og "From" (se figur A.8), som indikerer hhv. oppkallerens og den oppkallede sin adresse [ScR99b]. Videre forteller "Subject"-feltet temaet for sesjonen, "Call-ID" er kallets unike identifikator og "Cseq"-feltet inneholder et sekvensnummer. Contact-feltet inneholder en liste over adresser hvor en deltager kan bli kontaktet og "Require"-feltet brukes for forhandlinger om protokoll-egenskaper. "Content-Length" og "Content-Type" holder informasjon om kroppen²³⁴ til meldingen, som inneholder en beskrivelse av sesjonen som skal opprettes. SDP er vanlig å bruke til dette formålet. (SDP blir nærmere behandlet i kapittel A.2.4) Det er i figur A.9 vist et eksempel på meldingsutveksling mellom SIP-klient og -server.
- BYE:** Avslutter en forbindelse mellom to deltagere i en sesjon.
- OPTIONS:** Denne har ikke andre funksjoner enn å anmode om informasjon om kapabiliteter.
- STATUS:** Informerer andre tjenere om progresjonen på signalerings-handlingene forespurt i "Also"-hodet. Dette hodet inneholder en URL (som regel en SIP-URL) som en tjener skal etablere et kall til ved å sende en INVITE. "Also"-hodet kan inneholde flere URL-er, i så fall skal tjeneren sende en INVITE til dem alle. Dette hodet vil også inneholde et "Requested-by"-felt som indikerer hvilken klient som opprinnelig sendte ut forespørselen.
- ACK:** Sørger for pålitelig utveksling av meldinger.
- CANCEL:** Brukes for å terminere et ventende kall, men kan ikke omgjøre et ferdig etablert kall. Dvs at CANCEL-metoden ikke har noen virkning på en UAS om den har besvart et kall, men om en UAS ikke har besvart kallet skal den la være å gjøre det [ScR98b].
- REGISTER:** Frakter lokasjonsinformasjon til en SIP-tjener. Slik kan en bruker gi beskjed til en SIP-tjener om hvordan den skal konvertere en innkommende adresse til en utgående adresse, eller hvilken proxy som vet lokasjonen til den aktuelle brukeren. Kroppen til en REGISTER-respons kan være hva som helst, men som oftest inneholder den komplekse programmer for navneoversettelser og kan dessuten inneholde konfigurasjonsinformasjon som User Agent kan benytte seg av [ScR98b].

²³⁴ På engelsk: "body".



Figur A.9: Meldingsutveksling mellom klient og proxy-server. Hver forespørsel og respons består av både hode og kropp slik som deres motstykker i HTTP. Meldingen "200 OK" indikerer suksess. Basert på [deC99].

INVITE, BYE og OPTIONS betraktes som forespørselsmetoder og brukes som verktøy for SIP-klienter til å opprette tjenester [ScR98a].

SIP-klienter har også et sett av hodefelter, spesielt "Call-Disposition"²³⁵ og "Also". Videre finnes et "Location"-felt som er lik Also-feltet ved bruk i forespørsler, bortsett fra at Location lister opp alternativer som benyttes inntil en URL gir suksess. Ved flere "Location"-hoder i en forespørsel eller respons, skal mottageren velge én URL og sende INVITE dit. "Location"-adressene kan henviser til forskjellige kommunikasjonsmodier for én person eller til ulike personer. Det finnes også et "Replaces"-felt som indikerer at en bruker skal sende en BYE til de indikerte partene.

Semantikken i de overnevnte hodefeltene og meldingstypene er uavhengige av i hvilke forespørsler eller responser de inngår i. Dette innebærer en stor økning i antall nye tjenester det er mulig å opprette. Ved å kombinere de to overnevnte verktøyene på forskjellige måter kan et antall nye tjenester opprettes.

A.2.4 SDP

Når en UAC vil sette opp et kall mot en UAS, sender den av gårde en INVITE. Denne inneholder en "Session Description Protocol" (SDP)- kropp som beskriver, i tekstformat, de sesjonsparametre som oppkalleren ønsker i prioritert rekkefølge [ScR98b]. Den oppkallede parten på sin side vurderer disse ønskene og returnerer en modifisert liste (som inngår i SIP-responser) tilbake utfra dens kapabiliteter.

SDP inneholder informasjon om det følgende:

Mediastrømmene: En multimedia sesjon kan inneholde flere mediastrømmer, SDP inneholder antall mediastrømmer i sesjonen og hvilken type de ulike mediastrømmene er av. Så langt er audio, video, data, kontroll og applikasjoner de definerte strømmtypene [ScR99b].

Adresser: Indikerer destinasjonsadressene til de individuelle strømmene, for eksempel kan lyden sendes til en enkel telefon, mens en videosekvens sendes til mottagerens multimediamaskin.

Porter: Indikerer individuelle UDP-portnumre for mottak/sending av hver av de enkelte strømmene.

²³⁵ Uttrykker en klient sin ønsker om kall-håndtering, for eksempel at et kall ikke skal viderekobles.

Lastyper:	Inneholder en liste, i IP-telefoni sammenheng kalt "Capability set", over de ulike mediaformatene som kan brukes i sesjonen.
Start/Stopptider:	Inneholder informasjon over start-, stopp- og repretider for broadcastaktige sesjoner. Dette kan for eksempel brukes til å invitere en bruker til en ukentlig TV-sending eller lignende.
"Originator":	Holder informasjon om initiativtakeren ²³⁶ i en broadcastaktige sesjon, slik som navn og hvordan denne personen kan nås.

A.3 Sammenholding av SIP og H.323

Skal i det følgende sammenholde SIP og H.323 med hovedsakelig vekt på enkelhet, funksjonalitet og skalerbarhet.

A.3.1 Funksjonalitet

Kapasitetsutveksling: Hensikten med kapasitetsutvekslingen er å forsikre at overførte multimedia signaler kan mottas og prosesseres på korrekt måte i mottagende terminal [MoP01]. H.323 bruker H.245 som beskriver kapasitetsegenskapene i form av et sett av strukturer. Disse er veldig kompakte og inneholder svært detaljert informasjon om kapasitetene til hver terminal. SIP bruker SDP til dette formålet og en oppkallende part kan sende en OPTION-forespørsel for å få informasjon om den mottagende terminalens kapasiteter.

Interopabilitet: H.323 er totalt "backwards compatible" [MoP01, ScR99c] i form av at tjenerne må støtte alle funksjonene i de tidligere versjonene. Slik slipper klienter som kjører en eldre versjon av H.323 å bli oppgradert hver gang en tjener implementerer nye versjoner. Dette gir sømløs interopabilitet mellom ulike H.323-versjoner [DaF99]. SIP, derimot, fjerner karakteristikker fra versjon til versjon om utviklerne mener disse er unødvendige. Dette gir et mindre antall instruksjoner og mindre kompleksitet enn i H.323, men skader kompatibiliteten mellom versjoner. For å ha et grensesnitt mot tradisjonell telefoni må IP-telefoni støtte PSTN-signalering. H.323 benytter et subsett av meldingene i Q.931 for etablering av et kall. SIP har ingen slik støtte.

Tjenester: SIP sin bruk av URL gjør at man kan finne den oppkallede parten i forskjellige lokasjoner. Videre kan SIP lete i forskjellige maskiner etter en bruker samtidig og en proxy-tjener kan sende forespørsler til flere maskiner samtidig²³⁷. På denne måten akseleres søket. H.323 kan få til noe liknende ved å be en oppkaller å forsøke flere adresser, men dette er svært begrenset. Oppkalleren kan blant annet ikke gi noen liste over ønsker i den initielle kall-invitasjonen. Videre ble ikke H.323 designet for å fungere i WAN. Den støtter riktignok håndtering av kall mellom tjenere, men tillater ikke gatekeep-ere å håndtere forespørsler fra flere tjenere. Ellers

²³⁶ På engelsk: "originating part".

²³⁷ På engelsk: "forking".

betrakter [MoP01] tjenestetilbudene i SIP og H.323 som tilnærmet like, men understreker at det er vanskelig å sammenlikne dette da tjenester stadig blir lagt til begge protokollene.

En SIP-tjener kan opprette/koble ned samtaler hos eksterne tjenere og disse sender periodisk tilbake kontroll- og statusmeldinger. Slik kan den initiativtakende SIP-tjeneren tilpasse seg i sanntid [deC99]. På denne måten er det nesten ikke begrensninger for hva slags tjenester som kan legges til. I H.323, derimot, kreves det inngående kunnskap om protokollen og det er tidskrevende å få implementert det samme [deC99].

Kontroll: H.323 forsikrer kontroll med båndbreddebruken i en konferanse ved hjelp av "gatekeeper" [MoP01], mens SIP så langt ikke kan håndtere dette. Men det pågår en del forskning på dette, spesielt på å ta i bruk "Resource Reservation Protocol" (RSVP) for dette formålet.

Codec-støtte: I SIP blir SDP brukt til å transportere codec-data som støttes av endepunktene i en sesjon [MoP01]. SIP kan således fungere med hvilken som helst codec, bare de har blitt gitt et navn i "Internet Assigned Numbers Authority" (IANA). H.323, derimot, er avgrenset til kun codec-ene utviklet av ITU.

A.3.2 Skalerbarhet

Skalerbarhet er evnen en protokoll har til å justere seg inn som følge av nettverks- og lastforhold [deC99].

Tre aspekter:

- 1) Stort antall domener. H.323 var opprinnelig utviklet for LAN-bruk [MoP01], nyere versjoner definerer soner og prosedyrer for lokalisering av brukere igjennom disse. Likevel lider H.323 av skaleringsproblemer i store domener og komplekse lokaliseringsoperasjoner. Blant annet har ikke H.323 versjon 1 og 2 mekanismer for å detektere "looping" i søk i flere domener. I versjon 3 er det definert et PathValue-felt som indikerer hvor mange gatekeep-ere en melding kan traversere før den skal forkastes, men er ikke like effektiv som algoritmen i SIP [DaF99]. SIP var opprinnelig designet for bruk i WAN og har derfor "loop detection"-algoritmer for å forhindre uendelige løkker [deC99]. Før en proxy videresender en melding, sjekker den VIA-feltet for å se om den står oppført der. Hvis så, har meldingen gått i løkke og meldingen forkastes. Hvis ikke, fører proxy-en seg opp og sender meldingen videre til et endepunkt eller en annen proxy [DaF99].
- 2) Konferansestørrelse. H.323 krever en sentral "Multipoint Controller" (MC) for prosessering av all signalering [MoP01], selv for små konferanser. Dette medfører en flaskehals for større konferanser. Videre er gateway-er og MC-er valgfrie i H.323, hvilket medfører at i enkelte sammenhenger kan ikke H.323 støtte konferanser med tre parter. SIP har ingen krav om slike MC-er siden koordineringen av konferanser er distribuert, hvilket bedrer skalerbarheten og minsker kompleksiteten. Videre kan SIP håndtere alle konferansestørrelser, grunnet SIPs enkle syntaks og at den ikke holder tilstander [deC99]. SIP kan følgelig prosessere flere forespørsler per minutt og kan derigjennom håndtere flere samtidige kall.

- 3) Tilbakemeldinger. H.323 bruker H.245 som definerer prosedyrer for at mottagere kan kontrollere kodingsformen på mottatte medier, overføringsraten og "error recovery". [MoP01] trekker frem at dette gir mening i punkt-til-punkt kall, men ikke gir mening i flerpartskonferanser. I SIP benyttes RTCP for tilbakemeldinger på mottatt kvalitet og for å få tak i lister over deltakere i konferansen. Dette gjøres totalt distribuert.

A.3.3 Enkelhet

[MoP01] trekker frem antall sider i spesifikasjonene av H.323 og SIP som et mål for hvor enkle de er. SIP-spesifikasjonen er på 157 sider, mens H.323-spesifikasjonen er på 736 sider. Videre trekkes frem antall elementer definert i protokollene. H.323 definerer over 100 elementer, mens SIP har kun 37 ulike hoder med begrenset antall verdier og parametre. Til slutt trekkes det frem at SIP-meldinger er tekstbaserte, mens H.323 bruker binære representasjoner på meldingene sine basert på ASN.1 og "Packed Encoding Rules" (PER). SIP sin enkelhet i antall hoder og det at den er tekstbasert, gjør det mulig å relativt enkelt lage en SIP-parser i for eksempel Java eller Perl [deC99].

[deC99] trekker frem en fordel med H.323 sitt binære format, nemlig det at det går raskere å identifisere og prosessere en binær verdi enn en tekst-streng. Men [deC99] understreker at dette ikke utgjør noen stor fordel, da hastigheten for signaleringsmeldingene kun er kritisk i oppsettet av kallet.

[ScR99c] trekker frem at en H.323-gatekeeper eller -gateway må prosessere signaleringsmeldinger for hvert kall. Artikkelen hevder videre at jo enklere signaleringen er, desto flere kall kan en GK eller GW støtte. Da SIP er enklere å prosessere enn H.323, bør SIP kunne tillate flere kall per sekund bli håndtert av en boks enn H.323. Dette er en logikk som forfatterne ikke kan henvise til noen forskning på.

A.3.4 Utvidelsesmuligheter

Både H.323 og SIP kan utvides [deC99], men på to vidt forskjellige måter. SIP fungerer slik som HTTP og SMTP, en SIP-parser hopper over et ukjent hode om den kommer over et. Slik kan nye egenskaper bli lagt til SIP, uten at eldre SIP-parsere stopper å fungere. Videre er feilmeldingene i SIP gruppert i seks logiske klasser, ordnet fra 1xx til 6xx. Disse feilmeldingene klassifiseres utfra det første tallet, dette gjør at også eldre applikasjoner kan klassifisere en nyere feilmelding til én av de seks hovedklassene, og således skjønne hva slags type feil det er snakk om.

Programmere som ønsker å utvide H.323 har tilgang til "*nonstandardParam*"-feltet i ANS.1-datastrukturen. Ønsker man å utvide dette feltet er det nærmest umulig å bestemme hensikten med en tidligere utvidelse, på grunn av feltets binære representasjon, uten å snakke med personen som laget den. Med slike vanskelige utvidelser, kan dette gjøre det kronglete å få til full interoperabilitet.

A.3.5 Modularitet

[deC99] bruker legoklosser som analogi for modulariteten i SIP, hvor grunnleggende kall signalering, registrering og brukerlokalisering utgjør legofundamentet som kan utvides med andre legoklosser i form av andre "default" protokoller slik som RTP og RTCP. Men SIP kan også utvides med andre protokoller for å oppnå kompatibilitet mellom ulike signaleringsprotokoller. For eksempel, kan H.245 legges til SIP uten at det påvirker "legofundamentet". Videre kan man derfor også kombinere H.323 og SIP, slik at man gjør nytte av SIP sine avanserte lokaliseringsegenskaper til å lokalisere en bruker og la den faktiske kommunikasjonen skje med H.323-signalering [MoW01]. Dette kan skje ved f.eks. at en SIP redirigeringsstjener redirigerer responser til en H.323 URL [ScR99c].

En oppsummering av sammenlikningen mellom H.323 og SIP er gitt i tabell A.2.

Tema	H.323	SIP	Kommentar
Funksjonalitet	Full bakoverkompatibilitet. Støtter PSTN-signalering. Kan lete etter en bruker ved å be oppkalleren forsøke flere adresser. Støtter kun codec-er utviklet av ITU.	Fjerner unødvendige karakteristikk mellom ulike versjoner. Støtter ikke PSTN-signalering. Kan lete etter en bruker i flere maskiner samtidig, og sende ut forespørsler til flere maskiner samtidig ("forking"). SIP kan fungere med alle slags codec-er, bare de er definert i IANA	H.323 inkluderer støtte for PSTN-signalering som er en veldig viktig forutsetning for VoIP, da "toll bypass" (se kapittel 2 "IP-telefoni arkitekturer") er det viktigste argumentet for VoIP. I Norge har fjernaksten blitt fjernet, slik at det er andre egenskaper med VoIP som muligens har blitt viktigere. SIP har enklere søk etter den oppkallede parten enn H.323. Kan bli viktig for fremtiden mhp mobilitet, da man kan se for seg muligheten til å viderekoble VoIP-samtaler til en IP versjon6-basert mobiltelefon.
Skalerbarhet	H.323 var ikke tiltenkt brukt i WAN, først i versjon 3 kom "PathValue"-feltet for å hindre looping. [Pac01] hevder likevel at gatekeeper-ere med rutingsfunksjonaliteter kan sjekke to felter, "CallIdentifier" og "destinationAddress", i kallprosesseringsmeldinger. Om kombinasjonen av disse er lik et eksiterende kall, er dette en loop. H.323 krever en sentral "Multipoint Controller" for prosessering av all	SIP var tiltenkt brukt i WAN fra "fødselen" av, og har derfor innebygde "looping detection"-algoritmer. Krever ingen MC fordi koordineringen av konferanser er distribuert, dette gir bedre skalerbarhet og minsker kompleksiteten. Støtter alle konferansesestørrelser. SIP har ingen lastbalansering [Pac01].	SIP har bedre funksjonaliteter for detektering av "looping" enn H.323. Dessuten vil det være rimelig å si at slik egenskapene til SIP og H.323 har blitt beskrevet i litteraturen, er SIP bedre egnet for bruk i konferanser enn H.323 når det gjelder skalerbarhet og støtte av konferansesestørrelser.

	<p>signalering selv for små konferanser. Dette medfører flaskehals for skalerbarheten. Videre støtter ikke H.323 fullt ut treparts-konferanser.</p> <p>H.323 støtter last-balansering av endepunkter over et antall alternative gatekeep-ere for skalere et lokalt "point-of-presence"[Pac01]. Videre kan endepunkter ha muligheter til å rapportere sine tilgjengelige og totale kapasitet, slik at kall kan bli best mulig distribuert mellom de gatekeeper-ene.</p>		
Enkelhet	<p>Spesifikasjonen er på over 700 sider og definerer 100 elementer. Benytter binære representasjoner på meldinger. En fordel med dette er raskere parsing og identifisering enn ved tekststreng. Ifølge [deC99] avgrenses denne fordel kun til å gjelde oppsett av kall.</p>	<p>Spesifikasjonen er på rundt 150 sider og definerer kun 37 ulike hoder med begrensede antall verdier og parametre.</p> <p>Benytter teksbaserte meldinger. Dette gir store meldinger som ikke er egnet for nettverk hvor båndbredde og prosessering er avgjørende faktorer [Pac01].</p>	<p>SIP er mindre omfattende enn H.323 i definisjonene og spesifikasjonene. Dette gjenspeiler definitivt det faktum at H.323 er en <i>paraply</i>-protokoll, mens SIP er <i>én</i> enkelt protokoll.</p>
Utvidelsesmuligheter	<p>Vanskelig å utvide H.323 pga den binære representasjonen brukt i ANS.1-datastrukturen.</p>	<p>Feilmeldingene er inndelt i seks logiske klasser, 1xx.. 6xx.</p>	<p>Den binære representasjonen brukt i H.323 vanskeliggjør full interoperabilitet mellom ulike versjoner.</p> <p>SIP sin løsning med seks logiske klasser gjør at eldre applikasjoner også kan klassifisere nyere feilmeldinger utfra det første sifferet i den logiske klassen.</p> <p>For å unngå en evig "karusell" hvor man er nødt til å implementere nye versjoner av en aktuell signaleringsprotokoll, er det viktig med utvidelsesmuligheter og interoperabilitet mellom</p>

			versjoner. Dette kan muligens også få følger for IP-telefoni sin popularitet som teknologi.
Modularitet	Ikke beskrevet i litteraturen.	"Legofundament" som kan utvides med andre protokoller for å oppnå kompatibilitet mellom ulike signaleringsprotokoller.	

Tabell A.2: Oppsummering av H.323 og SIP i en IP-telefoni kontekst.

A.3.6 Oppsummering

I litteraturen trekkes frem fleksibilitet til å legge til nye egenskaper, enkel implementasjon og "debugging", og støtten for tekstlige beskrivelser av sesjoner som SIP sine fremste fordeler. H.323 har først og fremst fordeler i at den støtter interopabilitet med PSTN, god støtte for kompatibilitet mellom ulike versjoner²³⁸ og for utveksling av kapasiteter. Videre sees funksjonalitets- og tjenestestøtten i H.323 versjon 2 og SIP på som tilnærmet like, bortsett fra at tilleggstjenestene i H.323 er strengere definert og burde derfor medføre færre implementasjonsproblemer enn for SIP. Også i QoS-støtten blir H.323 og SIP likestilt (lik forsinkelse i kalloppsett, ingen reservering av ressurser eller tjenesteklasser), men det trekkes også frem at H.323 versjon 3 tillater signalering av ønsket QoS.

I følge [ScR98a] var H.323 sine svakheter innenfor kompleksitet, skalerbarhet og utvidelsesevne som motivasjonene for at SIP ble opprettet. [deC99] trekker frem at SIP kan ha hatt en fordel ovenfor H.323, da SIP som en "draft" under utvikling lenge har hatt mulighet til å lære fra feilene i H.323. H.323, på sin side, har vært igjennom flere revideringer og sliter med kompatibilitetsproblemer mellom ulike versjoner. [DaF99] og [MoW01], derimot, trekker frem det som positivt at standardiseringsfolkene for de to protokollene kikker hverandre i kortene, da de slik lærer av hverandre og forbedrer seg. På denne måten vil forskjellene mellom de etter hvert forsvinne for hver nye versjon.

SIP benytter en desentralisert tilnærming slik som i HTTP og denne åpne arkitekturen vil i stor grad være i stand til å tilpasse seg til og vokse med et stadig ekspanderende VoIP-marked. H.323 har også muligheter for utvidelser i sin sentraliserte arkitektur, men protokollen kan ikke forlenges i like stor grad som SIP. Derfor tør ikke [deC99] å spå hvilke av de to som vil "vinne", men at muligens vil H.323 gå seirende ut i nærmeste fremtid og at SIP vil være den langsiktige løsningen.

Dette blir langt på vei støttet av [MoW01]:

"Vi er av den oppfatning at SIP er best for å operere mot Internett, mens H.323 virker mer solid i øyeblikket for forbindelser mot PSTN".

Sikkert er det at IP-telefoni med tiden vil erstatte vanlig telefoni og at signaleringsprotokollene må være interopabile for at kundene skal kunne bli tilbudt de tjenester de ønsker.

²³⁸ På engelsk: "backwards compability".

Utover det vil sannsynligvis være en debatt i industrien om VoIP signaleringsarkitektur og opp til markedskreftene å avgjøre den videre utviklingen [LiM00].

Appendiks B

Måling av talekvalitet

I hovedsak kan man si at det finnes to grunnleggende måter å måle talekvalitet på, subjektivt og objektivt. Dette vedlegget skal se nærmere på hvilke metoder som eksisterer for subjektiv måling av talekvalitet, siden det er slike tester som skal benyttes i kvalitetstesting av IP-telefonitale i ATC.

Subjektive tester kan foregå på ulike måter avhengig av testoppsettet, men de vil alltid inkludere en hørselstimulans og nedtegning av forsøkspersonenes responser [ITS94]. Det er i slike eksperimenter svært viktig at miljøet hvor eksperimentene foregår er kontrollerbare, spesielt med tanke på bakgrunnstøy. Derfor blir også ofte lydisolerte rom benyttet. Hørselstimulansen vil ofte være tale eller musikk, og vil gjengis igjennom høytalere, hodetelefoner, telefon-håndsett eller eventuelt annet utstyr som eksperimentoppsettet krever.

Det er ITU-T standard P.800 "Methods for subjective determination of transmission Quality" [ITU96] og ITU-dokumentet "Handbook on Telephonometry" [ITU92] som setter standarden for subjektiv talekvalitetstesting. I det følgende vil det sees på hvilke to hovedgrupper P.800 deler subjektive tester inn i, og hvilke anbefalinger "Handbook on Telephonometry" gir om eksperimentoppsett. Videre vil det refereres til hva man i litteraturen trekkes frem som fordeler og ulemper ved slike tester. I tillegg vil det nevnes hvilke typer objektive tester som finnes.

B.1 ITU-T Recommendation P.800

ITU-T definerer i sin P.800 "Methods for subjective determination of transmission Quality" metoder for subjektiv bestemmelse av talekvalitet. Grovt sett kan man dele standarden i to hovedtyper tester [ITU96]:

1. "*Conversational-opinion*"-tester
2. "*Listening-opinion*"-tester

Disse to vil bli sett mer på i påfølgende delkapitler.

B.1.1 “Conversational-opinion”-tester

“Conversational-opinion” som prøver å gjenskape en autentisk telefonkonversasjon i et laboratoriumsmiljø. For slike tester er det viktig å velge realistiske forhold for testene, plukke ut riktige personer og styre testingen på en fornuftig måte. Forholdene må spesifiseres nøyaktig, settes opp og måles både før og etter testingen.

Forsøkspersonene sitter i lyttbokser (noe liknende de man blant annet brukte i NRK-programmet “Kvitt eller Dobbelt”), som det stilles en del strenge krav til. Blant annet stilles det krav til størrelse, romklang og omgivelsestøy²³⁹. Videre stilles det også krav til at forsøkspersonene ikke har jobber relatert til forsøkene, eller har deltatt i generelle subjektive testinger på minst seks måneder og ikke deltatt i konversasjonstester på over et år. Derimot stilles det ingen krav til fordeling blant kvinner og menn, med mindre eksperimentdesignet fordrer noe annet. Forsøkspersoner blir tilfeldig satt sammen to og to i grupper før forsøkene tar til. I forkant av forsøkene skal forsøkspersonene ha mottatt et brev med ikke-teknisk informasjon om eksperimentene og hva som vil bli forventet av dem. Når så disse personene ankommer på forsøksdagen, blir eventuelle spørsmål og uklarheter angående brevet besvart og ryddet opp i. Lyttboksene og utstyret i dem blir deretter demonstrert, og videre blir det gitt noen flere opplysninger om for eksempel hvor mange samtaler som skal finne sted under forsøkene.

B.1.1.1 Konversasjonstester i laboratorium

I dette delkapitlet skal det sees nærmere på hva [ITU92] sier om hvordan konversasjonstester bør legges opp.

Ifølge [ITU92] gjenskaper konversasjonstester en naturlig telefonkonversasjon bedre enn lyttetester. Det utføres ingen kontroll over snakkingen, eller på lyttenivåene. Helt sentralt i slike i konversasjonstester er den oppgaven, konversasjonsoppgaven, som deltagerne utfører for å få til en samtale.

[ITU92] definerer målsettinger med slike tester til å være å:

- Å gjenskape en situasjon hvor partene har en konversasjon som utarter seg som en autentisk telefonsamtale over forbindelsen som skal bedømmes.
- Få tak i forsøkspersonenes subjektive mening om forbindelsen.
- Observere hvordan testpersonale oppfører seg under testene, for å finne ut av visse "uvaner" som kan bidra til å forringe forbindelsen, som for eksempel hvordan deltagerne holder telefonrøret.

For at dette skal kunne oppnås må følgende oppfylles:

- Valg av linjeforhold som må møte visse krav.
- Prosedyrene i testingen bør oppleves så naturlig og realistisk som mulig for testpersonene. Dette betyr at:
 - temaet som diskuteres bør være av en slik art at det leder til en konversasjon som ruller videre på egen hånd.

²³⁹ Mer om dette finnes i P.800 [ITU96] Annex A side 5-7.

- forsøkspersonene bør ikke bli forventet å bruke utstyr som folk flest ikke er vant med, eller bli bedt om å oppføre seg på en måte som er helt unaturlig å gjøre når man er på telefoner²⁴⁰.
- Forsøkspersonene bør ikke forventes å komme med unaturlige og dype analyser av kommunikasjonsmetoden, eller komme med andre ytringer som er avhengig av personenes kunnskaper om temaet. Dette utelukker en del skalaer som brukes i diagnostisk og psykologiske eksperimenter.

[ITU92] understreker at eksperimenter som bryter med det overnevnte ikke behøver å være feil, men omstendighetene rundt disse eksperimentene må tillate det. [ITU92] sier bare at de anbefalinger som er gitt, er nødvendig for å få til eksperimenter hvor resultatene vil være gyldige for reelle telefonsamtaler.

Det kan også utføres forenklede konversasjonstester²⁴¹ for å få testet på et øket antall scenarier eller for å begrense tidsforbruket. De snareveiene som i så fall tas, vil gå på bekostning av hvor naturlig eksperimentene blir. Eksempler kan være:

- Ikke slå nummer/ringe opp mottageren.
- Hoppe fra en konversasjon til en annen ved å signalisere pause i konversasjonen for å bedømme kvaliteten, og deretter fortsette konverseringen.
- Begrense varigheten av konversasjonene til en forhåndsbestemt øvre grense, for eksempel 3 min.

[ITU92] sier også at testinstruksjonene som gis til testpersonene skal være klare og entydige og skal dekke alle aspektene i testene, inklusiv oppsett av forbindelsen. Videre skal informasjonen forklare konversasjonsoppgaven og hvilken metode som brukes for bedømmelse.

Eksperimentlederen utarbeider en plan²⁴², basert på eksperimentdesignet, som viser hvilke forhold det enkelte forsøkspar skal utsettes for, hvilke sett av bilder²⁴³ de skal benytte seg av i konversasjonen, og som kan gi diverse annen informasjon. Videre bør det være plass på planen til å fylle ut ny informasjon som eventuelt vil kunne dukke opp under eksperimentene. [ITU92] legger vekt på at det er veldig viktig at man ikke gir forsøkspersonene noen informasjon angående linjeforholdene. Svarskjemaene bør heller ikke inneholde noen tall eller kode som er relatert til dette. Slik informasjon settes på etter at skjemaene er samlet inn.

B.1.1.2 Skalaer

Det benyttes to typer skalaer:

- “Conversational opinion”-skalaen består av “Excellent”, “Good”, “Fair”, “Poor” og “Bad”. Disse nummeres fra 5 (tilsvarende “Excellent”) til 1 (tilsvarende “Bad”). Gjennomsnittet av disse tallene gir “Mean Conversational-opinion score” (MOS_C).
- “Difficulty scale” er et ja/nei-spørsmål om noen av samtalepartnerne hadde problemer med å høre eller snakke over en aktuell forbindelse. Prosentandelen “ja”-tilbakemeldinger kalles “Percentage Difficulty” og symboliseres ved ”%D”.

²⁴⁰ For mer om dette, se punkt iii) side 126 i [ITU92].

²⁴¹ På engelsk: “Simplified conversastion tests”.

²⁴² På engelsk: ”schedule”.

²⁴³ Bilder blir ofte brukt som utgangspunkt i konversasjonensoppgaven, dette vil bli mer omhandlet i appendiks B.1.1.3.

Det blir også understreket at i en del testopplegg er det nødvendig med noe vanskelige lytte- og snakkeforhold. I slike sammenhenger er det ofte vanlig å spørre forsøkspersonene om vedkommendes egen beskrivelse av hvordan de oppfattet de vanskelige forholdene.

Det understrekes i [ITU96] at ordlyden og layout-en på disse skalaene er viktige og bør følge standarden som ble til etter mange års erfaring. På en annen side, blir det også sagt at skalaene må oversettes til det nasjonale språket som brukes i det landet eksperimentene gjøres, i og at dette kan innebære små avvik fra ordlyden i den opprinnelig engelske skalaen.

B.1.1.3 Eksempel på testprosedyre og konversasjonsoppgaver

[ITU92] gir følgende eksempel på en testprosedyre for en konversasjonstest:

Først, før hver konversasjon, tar testpersonene frem et sett av bilder og ordner disse i en rekkefølge etter den enkeltes ønske. Deretter blir enkelte passende felter i evalueringsskjemaene fylt ut, for eksempel rekkefølgen på bildene. Etter dette får den ene beskjed om å ringe den andre. Deltagerne bestemmer selv gangen i samtalen, men det er ikke under noen omstendigheter lov til å snakke om selve bedømmelsen av talen. Etter at samtalen er ferdig og bedømmelsen er avgitt, samt at forsøkspersonene har svart på ”difficulty”-spørsmålet, tar eksperimentlederen(e) kontakt med hver av deltagerne for å spørre om hva de svarte på dette spørsmålet. Om svaret er ”ja”, bes deltagerne om å beskrive vanskelighetene. Hensikten med denne samtalen (som gjøres over telefon) er å såvel å avlede deltagerne fra å merke eventuelle justeringer av bakgrunnstøy, som å hindre at forsøkspersonene ikke føler seg isolert og glemte. Slike samtaler bør holdes så ikke-teknisk som mulig. Til slutt får deltagerne beskjed om å legge vekk evalueringsskjemaet i en konvolutt de har fått utdelt og begynne å sortere et nytt bildesett. Eventuelt kan deltagerne også bli bedt om å vente for å bli avløst av nye forsøkspersoner.

[ITU92] gir også eksempler på hvordan selve konversasjonsoppgaven kan legges opp. Blant annet nevnes bruk av like sett med postkort, hvor hver deltager velger en rekkefølge og via konversasjon med partneren blir enige om en felles ordning. Man kan også tenke seg at postkortene er miniatyrtgaver av bilder som skal velges ut og henges opp i for eksempel kantinen på en arbeidsplass. Det finnes også andre varianter av denne oppgaven: man kan nummerere de eksakte like bildesettene ulikt, slik at deltagerne må beskrive bildene for å skjønne hvilket bilde motparten ser på. En annen variant er å ta ut ett bilde av det ene settet og erstatte det med et annet nytt bilde (som den andre parten ikke har) og samme scenario oppstår.

Det blir understreket at samtalene bør ha en naturlig begynnelse og slutt, og det skal tilstrebes å ikke avbryte midt i en samtale. Grunnen til dette er at samtalen skal bli mest mulig realistisk.

"Handbook of Telephonometry" gir også anbefalinger om hvor stort spenn av testforhold det bør testes på, og sier at testforholdene bør varieres fra det en gjennomsnittlig forsøksperson vil kalle ”excellent” og helt ned til ”poor”. Det skal alltid være minst et ”dummy”-forhold, ofte bruker man tre. Hver deltager bør oppleve hele skalaen, i alle fall én gang med veldig gode forhold, én gjennomsnittlig og én

med veldig dårlige forhold. De andre gjennomføringene bør ikke samles rundt tilnærmet den samme kvalitetsverdien (MOS), man ønsker en viss spredning. Om man ikke kan være sikker på forhånd at hele spennet brukes, bør man kjøre noen innledende testrunder og eventuelt gjøre nødvendige endringer i forholdene.

B.1.1.4 Datainnsamling

Talevolum, samtalevarighet og aktivitetsnivå kan hentes utfra båndopptak av samtalene som kan gjøres for å overvåke forsøkene. En annen måte å overvåke forsøkene kan for eksempel være visuell overvåkning ved bruk av videoopptak for å finne ut hvordan forsøkspersonene holder håndsettene. Den mest vanlige metoden for å hente ut data fra båndopptakene, er ved bruk av maskinbaserte måleinstrumenter og lagring på filer.

For hver konversasjon hentes det altså ut to subjektive testresponser per forsøksperson, "conversation-opinion score" og "Difficulty scale"-verdien.

B.1.1.5 Behandling av testresultater

For hvert testscenario kalkuleres gjennomsnittlig meningspoeng²⁴⁴. Videre anbefaler standarden at tillitsgrenser bør utledes om resultatene, samt signifikantester kalkuleres ved bruk av konvensjonelle avviksanalyser²⁴⁵.

B.1.2 "Listening-opinion"-tester

"Listening-opinion"-tester er den testmetoden som det blir antatt er best egnet for subjektiv testing av systemer og tjenester under utvikling. Denne testen foregår ikke fullt så realistisk, slik som "Conversational-opinion", og det blir derfor ikke satt fullt så strenge krav til denne. På en annen side, i denne typen tester stilles det en del krav på områder der "Conversational-opinion"-tester gir større frihet til å velge om kravene skal følges. Det er definert flere forskjellige typer tester i denne kategorien:

- "Absolute Category Rating" (ACR)
- "Quantal-Respon Detectability"
- "Degradation Category Rating" (DCR)
- "Comparison Category Rating" (CCR)
- "Threshold Methods"

I dette delkapitlet vil det bli fokusert nærmere på ACR, i tillegg vil DCR og CCR nevnes kort.

[ITU92] sier at generelle målsetninger med lyttetester er å:

- Kontrollere viktige variable i eksperimentet. Eksempler på dette kan være linjeforhold, egenskaper på inngangssignalet, og valg av talere og talemateriale.
- Få tak i lytterenes subjektive reaksjoner.

Derfor må følgende oppfylles:

- Valg av linjeforhold²⁴⁶ må møte visse krav.

²⁴⁴ På engelsk: "opinion score".

²⁴⁵ På engelsk: "analysis-of-variance".

²⁴⁶ På engelsk: "circuit condition".

- Forsøkspersonenes bedømmelse har vist seg å bli sterkt påvirket av kvalitetspekteret og lyttevolumet²⁴⁷ som brukes i testene, derfor må de inneholde et nødvendig spenn av forhold. Dette bør bestå av både de mest fordelaktige forhold og de verste. Men siden de verste tilfellene ikke tolereres i praksis, er det viktig å ikke ha for mange av de dårligste forholdene i testingen.

Siden forsøkspersonene ikke har andre oppgaver enn å lytte, er det stor fare for at de kan gå lei. Følgelig bør ikke testsesjonene vare lengre enn i 20 minutter. Alternativt bør testingen bestå av sesjoner med pauser innimellom.

[ITU92] anbefaler at man kjører seks referanseforhold²⁴⁸, slik at forsøkspersonene på begynnelsen av testingen får en kvalitetsreferanse og blir kjent med testrutinene. I dette delkapitlet skal "Absolute Category Rating" beskrives nærmere. Til slutt vil også "Degradation Category Rating" og "Comparison Category Rating" bli kort omtalt.

B.1.2.1 "Absolute Category Rating" (ACR)

Det stilles i ACR en del krav til lyd-kilden, og videre for å unngå uønskete variasjoner i talen anbefales det at den bør tas opp på forhånd. Det eksisterer en del krav til systemet som formidler disse opptakene til lytterne, denne kvaliteten skal måles både før og etter testene for å verifisere at systemet oppfyller kravene P.800 stiller. P.800 kommer med tre forslag til avspillingsenheter, tradisjonell båndopptaker med to spor, Digital Audio Tape (DAT) eller digitalt lagringssystem.

Lyttematerialet bør bestå av såkalte "passe" korte setninger tatt fra ikke-teknisk litteratur og satt inn i en tilfeldig rekkefølge. Målet er at setningen skal gi 2-3 sekunder tale. Det anbefales at minimum to setninger, maksimum fem setninger utgjør et talesample²⁴⁹. Tiden mellom hver setning avgjøres av eksperimentlederne. Denne tiden er viktig, fordi i denne perioden vil linjestøy og adaptative prosesser kunne gå over i nye tilstander. De som leser inn talen på bånd skal snakke naturlig og flytende, på en slik måte som føles behagelig for dem og som de kan holde over en lengre tid.

Når det gjelder utvelgelse av personer til å lese inn talen på bånd, sier P.800 at det er viktig å ta hensyn til at noen avanserte prosesser kan påvirke mannlige og kvinnelige stemmer ulikt. Derfor bør man ha flere enn én mannlige og én kvinnelig stemme for å få velbalanserte resultater. Det blir oppfordret til å behandle resultatene for de ulike kjønnene separat, mens gjennomsnittsverdier kan regnes ut om det vil gi helt sentrale resultater og ikke gir interaksjoner som er statistisk forskjellige.

Det bør bli benyttet referanseforhold avhengig av hva slags type tester som skal utføres. Hensikten med dette er å ha noe fornuftig å sammenligne med for påfølgende eksperimenter i samme eller andre laboratorier.

Selvom det ikke mulig å utlede absolutte regler for bruk av ACR-tester, har [Dim93] funnet ut slike tester fungerer godt for Q-verdier (forholdet mellom talenivå og multiplikativ støy-nivå) mindre enn 20 dB. Videre sier [Dim93] at ACR-tester er utsatt

²⁴⁷ På engelsk: "listening level".

²⁴⁸ På engelsk: "dummy condition".

²⁴⁹ For mer informasjon om organiseringen av lyd-materialet, se [ITU92].

for såkalt ”order effect”, det vil si at rekkefølgen talesamplene spilles av vil ha noe si for bedømmelsen av samplet. For eksempel vil et sample karakterisert som ”fair” få en annen bedømmelse avspilt etter et ”good”-sample, enn om det var avspilt etter et sample karakterisert som ”bad”. ACR har blitt brukt til å bedømme ytelsene til G.721 og G.722 ADPCM algoritmene [Ros88].

B.1.2.1.1 Eksperimentdesign

Eksperimentene kan ikke ha større omfang enn at et knippe forsøkspersoner ikke går lei eller blir slitne. Varigheten av en sesjon bør ligge på 20 minutter, i alle fall ikke lengre enn 45 minutter. Blir det slik at en sesjon blir varende lengre enn dette, gjøres det klokt i å dele opp sesjonen i to. Antall testpersoner vil ligge mellom 20 og 100 [ITS94].

B.1.2.1.2 Skalaer

Det finnes tre typer skalaer i ACR:

- “Listening-quality” er en skala som er lik den brukt i “Conversational opinion”, men gjennomsnittsverdien kalles her “Mean Opinion Score” (MOS).
- “Listening-effort”-skalaen inneholder fem karakteristikk på hvilken lytteinnsats som kreves. Den beste karakteristikken er “Complete relaxation possible; no effort required” som har blitt tildelt verdien 5, mens den dårligste med verdien 1 er “No meaning understood with any feasible effort”. Gjennomsnittsverdien kalles “Mean Listening-effort Opinion Score” (MOS_{LE}). Dessuten inneholder denne skalaen en overskrift, “Effort required to understand the meaning of the sentences”. Uten denne gir ikke skalaen noe mening.
- “Loudness-preference”-skalaen inneholder likeledes en overskrift, “Loudness preference”, og har en femdelt-skala hvor den beste karakteristikken er “Much louder than preferred” og har blitt tilordnet verdien 5, mens den dårligste karakteristikken er “Much quieter than preferred” og har verdien 1.

Man bør være oppmerksom på at det kan være vanskelig å oversette navnene på kategoriene brukt på skalaen til forskjellige språk [ITU92]. Disse oversettelsene gjør at man ikke får beholdt de samme sprangene på kategoriene som på originalspråket, engelsk [ITU92]. Derfor er det umulig å gi en universal veiledning om valg av ”opinion scale”²⁵⁰.

B.1.2.1.3 Analyse av testresultater

Gjennomsnittsverdiene bør beregnes for hvert av lytteforholdene (som brukes i eksperimentene) for de ulike lyttevolumene²⁵¹ og bli listet opp. På denne måten kan man finne spesielle resultater forårsaket av for eksempel systeminnvirkninger på kvinnelig- og mannlig stemmer. Tillitsgrenser bør også evalueres og signifikantester utføres ved vanlige teknikker for avviksanalyse.

B.1.2.2 “Degradation Category Rating” (DCR)

Grunnprinsippet er å måle forverringen av kvalitet i forhold til et referansesignal. Her brukes en ”degradation opinion”-skala og hørselsstimulansen avspilles i A-B par,

²⁵⁰ På norsk: menigsskalaer.

²⁵¹ På engelsk: “Listening level”.

hvor A er referansesignalet som skal sette standarden for bedømmelsene og B er den samme hørselsstimulien etter en viss prosessering [Dim93]. A avspilles alltid først, deretter B. Testpersonene bedømmer deretter de prosesserte signalene etter nevnte skala. Resultatene fra denne typen tester kalles "Degradation MOS" (DMOS). For mer detaljer, se "Annex D" i P.800 [ITU96].

B.1.2.3 "Comparison Category Rating" (CCR)

Grunnprinsippet er her å sammenlikne mot et referansesignal og måle forbedring og/eller forverring av signalet. I DCR blir det uprosesserte lydsamplet alltid avspilt først og deretter det manipulerede, men i CCR kan rekkefølge for hvert forsøk velges fritt. Men, i totalt halvparten av forsøkene skal det manipulerede samplet spilles av først, og i den andre halvdel i omvendt rekkefølge. En fordel som trekkes frem med CCR over DCR, er at man i CCR er i stand til å måle både bedring og forverring av signalet, ikke kun forverringen.

B.1.3 Kritikk mot P.800

[BSD01] er kritisk til "Listening-quality"-skalaen både når det gjelder inndelingen av skalaen og hvorvidt den i hele tatt måler parametre som er viktige for brukerne. Artikkelen trekker frem en fordel med "Listening-quality"-skalaen er at den er en standardisert skala hvor kvantitative data, relatert til et kvalitetsnivå kan bli sammenlignet direkte med et annet nivå. Artikkelforfatternes kritikk går hovedsakelig ut på at kvalitative skalaer, slik som MOS, ikke klarer å bedømme brukernes ulikheter eller hvordan en opplevd QoS²⁵² i interaksjonens tidlige fase kan påvirke bedømmelsen av QoS. Eller det omvendte tilfellet: artikkelen henviser til forskning i [ADGHP95] som viste at når varigheten av stimulansen økte fra 10 sekunder til 30, la forsøkspersonene mer vekt på den stimulansen de opplevde mot slutten av sekvensen enn på begynnelsen. Årsaken til dette, sier [ADGHP95], kan skyldes menneskets arbeidende minne er kortvarig av natur. Poenget til [BSD01] er at når man, etter en lydopplevelse, spør forsøkspersonene om deres mening om kvaliteten, klarer ikke slike metoder å fange opp hvilken fase av lydopplevelsen som påvirket bedømmelsen deres.

Løsningen, ifølge [BSD01], er en mer dynamisk måte å evaluere hva brukerne opplever, da forholdene til mediet som måles forandres dynamisk. Grunnen til dette, er at tradisjonelle metoder (slik som "Listening-quality"-skalaen) ikke er tilpasset IP-nettverkens bygeaktige natur, som følgelig kan innebære sterkt varierende kvalitet for brukerne. Et forslag gitt i artikkelen for å oppfylle dette, er såkalt vedvarende bedømmelse²⁵³. Her benytter testpersonene et program hvor de kan flytte en såkalt "slider bar" for å kontinuerlig og dynamisk kunne bedømme kvaliteten. Slik blir man i stand til å registrere hvordan testpersonenes QoS-oppfatning forandres i løpet av interaksjonen.

Et annet forslag som nevnes i [BSD01] er "Task-oriented performance measures" (tmgs), hvor brukerne blir utsatt for varierende kvalitetsnivåer under eksperimentene.

²⁵² Forkortelse for "Quality of Service".

²⁵³ På engelsk: "continous assessment".

Testene måler objektivt evnen brukerne utviser til å fullføre oppgaven²⁵⁴ de bruker systemet til. Artikkelen referanse [KDK99] viser at man slik måler en ytelse som er relevant i forhold til applikasjonen som støtter oppgaven. Ved at metoden er objektiv, slipper man usikkerhetsmomentene ved brukernes selvreflekteringer ved subjektiv bedømmelse. Tmps er spesielt egnet som metode når brukerne blir bedt om å utføre en kompleks oppgave [BSD01].

Et av resultatene av undersøkelsene presentert i [BSD01] er at hensikten med bruken av en konkret applikasjonen, for eksempel lytting til sanntids audio, avgjør brukernes krav til kvalitet. Det blir også sagt om kvalitative metoder, det å spørre testpersonene om å beskrive kvaliteten de opplevde, kan etablere et godt fundament for videre design av eksperimenter. Dette kan også brukes i sammenheng med eksperimentdata for å få en idé om hvilke motivasjoner som lå til grunn for bedømmelsene underveis i interaksjonen (ved bruk av metoden for vedvarende bedømmelse).

[WaS98] kommer med kritikk av "Listening-quality"-skalaen som dekker tre områder: (1) vokabularet brukt på skalaene, (2) lengden på de anbefalte testmateriale, og (3) "conversational difficulty"-skalaen.

Angående (1) begrunner artikkelen med at talen i en multimedial kommunikasjon befinner seg i smalbandet²⁵⁵ av frekvensspekteret og er utsatt for en rekke ødeleggelser som følge av nettverks- og omgivelsesforhold. Derfor kan den eksisterende inndelingen av "Listening-quality"-skalaen være uegnet, fordi det er sannsynlig at bedømmelsene av talen stort sett vil finne seg i den nedre halvdel av skalaen. [WaS96] presenterer resultater som skal verifisere dette. Begrunnelsen for (2) er at den anbefalte varigheten på testmaterialet er på maksimalt 10 sekunder, hvilket er altfor kort til å oppleve de reelle variasjonene i talekvalitet som følge av nettverkets uforutsigbarhet, eller på den andre siden får man heller ikke oppleve hva slags kvalitet systemet er godt for ved lave tapsrater. Kritikken i (3) går på at selv små andeler pakketap kan gi vanskeligheter med å høre/snakke, selv om de er kortvarige.

Videre er det en voksende skepsis blant forskere om hvorvidt man skal stole på "Listening-quality"-skalaen. [WaS98] refererer til en del forskning hvor man har funnet ut at intervallene mellom de fem kategoriene på skalaen ikke er jevnt fordelt på en del språk. Det vil si at avstanden mellom "Good" og "Fair", og "Poor" og "Bad" ikke er like store på noen språk. Altså, kan oversettingen fra engelsk bli et problem i og med at det ikke alltid er like lett å finne dekkende ord på alle språk, som er ensbetydende med de engelske. Som et eksempel viser [WaS98] til forskning der betydningen av ordet "OK" blir plassert på midten av skalaen av amerikanerne, mens italienerne forbinder det med en betydning lik "Good". På dette punktet sier jo også [ITU92] (som nevnt i B.1.2.1.2) at man bør være oppmerksom på at det kan være vanskelig å oversette navnene på kategoriene på skalaen til forskjellige språk.

[WaS98] viser videre til resultater presentert i [WaS97] som viser at umerkede skalaer reduserer forsøkspersonenes tendens til å søke endepunkter av skalaen.

²⁵⁴ På engelsk: "task".

²⁵⁵ På engelsk: "Narrowband".

B.2. Objektive tester

Objektive metoder for talekvalitetstesting går ut på lage en matematisk modell for hvordan det menneskelige øret fungerer, og bruke dette til å finne en fornuftig måte å prøve og forutsi hvordan folk faktisk opplever kvaliteten på en ende-til-ende nettverksforbindelse [BrT02].

[Vor94] sier at selv om subjektive tester ofte vil være tidkrevende og dyre, blir denne typen tester ofte betegnet som den beste måten å teste ut nytt taletransmisjons utstyr. Automatiserte objektive testmetoder kjennetegnes ved at de gir resultater det er mulig å gjenskape og krever betydelig mindre innsats, tid og er mer kostnadsbesparende. Allikevel, vil man ofte finne betydelig avvik mellom objektive og subjektive resultater, hvor de subjektive som regel blir ansett for de mest korrekte. Jo nærmere objektive testresultater treffer de subjektive resultatene, desto bedre ansees de. [Vor94] sier videre at valget mellom objektive og subjektive testmetoder blir en avveining mellom hvor store avvik man kan godta i testingen og ønsket om å kutte kostnader. For testing med svært høye krav til små avvik, blir subjektive testmetoder betegnet som de mest ideelle. [Vor94] presenterer også tre metoder for å sammenligne objektive og subjektive talekvalitets tester.

[Agi01]²⁵⁶ er, i motsetning til [Vor94], utelukkende positiv til objektive metoder for talekvalitetstesting. [Agi01] trekker frem at subjektive metoder er dyre, fordi de krever et stort antall mennesker og detaljert testoppsett. Videre ser [Agi01] problemer i at resultatene fra subjektive testinger vil i stor grad være avhengig av en del ukontrollerte faktorer som følge av testpersonenes humør, holdninger og kulturelle bakgrunn. Dette gjør at subjektive testmetoder ikke er repeterbare og konsekvente²⁵⁷. Dessuten, mener [Agi01], at subjektive testmetoder gjør det ineffektivt og upraktisk å utføre jevnlige testinger, slik som er nødvendig for endring av nettverksdesign og konfigurasjon, og rutinemessig overvåking av nettverk.

En årsak til at [Vor94] er såpass kritisk til objektive testmetoder, mens [Agi01] er utelukkende positivt, kan være at det har skjedd en del utvikling av slike testmetoder de siste årene.

[BoS99b] sier at det ikke er sikkert at man kan anta en sammenheng mellom objektive QoS-nivåer og brukernes subjektive evalueringer av QoS. Derfor spør [BoS99b] om prisingsmekanismer på tjenester må heller fokusere på sammenhengen mellom verdien tilskrevet et QoS-nivå og hvordan denne tjenestekvaliteten blir representert hos brukeren. Det er nemlig ikke nødvendigvis sikkert at den teknisk mest effektive fremgangsmåten treffer brukernes preferanser, men graden av tilbakemeldinger²⁵⁸ brukerne får i sin interaksjon for å utføre en arbeidsoppgave²⁵⁹. Videre sier [BoS99b] at objektive nivåer med QoS ikke er like viktige for brukerne som å motta en forventet mengde QoS som tillater brukerne å fullføre en valgt oppgave.

²⁵⁶ Ikke trykket materiale, hentet fra <http://onenetworks.comms.agilent.com/WhitePapers.asp#IP>.

²⁵⁷ På engelsk: "consistent".

²⁵⁸ På engelsk: "feedback".

²⁵⁹ På engelsk: "task".

Av avgrensingshensyn vil ikke detaljert beskrivelse av objektive testmetoder gis i denne rapporten, men det kan nevnes at det finnes en del varianter og standarder:

- ITU-T Recommendation P.861 "Perceptual Speech Quality Measurements" (PSQM), har i følge ITU-T sine sider på Internett²⁶⁰ blitt trukket tilbake på grunn av mangler og blitt erstattet av følgende oppdatering:
- "Perceptual Analysis Measurement System" (PAMS) utvikles av British Telecom. [BrT02]²⁶¹ sier at teknologien i PAMS har blitt innarbeidet i PESQ (se neste punkt) og for viderutviklingen av metoden ble firmaet "Psytechnics" etablert i 2001.
- ITU-T Recommendation P.862 "Perceptual Evaluation of Speech Quality" (PESQ).

For mer om slike objektive testmetoder, se [Agi01] samt de nevnte standardene. Det kan leses mer om PAMS i [RiH00], [BrT02] og på websidene til "Psytechnics"²⁶² finnes whitepapers om PAMS og PESQ.

²⁶⁰ <http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-P.861>

²⁶¹ Ikke trykket materiale. <http://www.btexact.com/research/pastprojects?doc=42163&page=2>.

²⁶² <http://www.psytechnics.com/technical.html>.

Appendiks C

Forkortelser

ACR	<i>Abosolute Category Rating</i> . For mer informasjon, se Appendiks B.1.2.1.
ASIC	<i>Application Specific Integrated Circuits</i> .
ATC	<i>Air Traffic Control</i> .
CNG	<i>Comfort Noise Generation</i> . Støy blir avspilt hos mottageren hvor VAD har fjernet en del av talen på grunn av stillhet. Det skal forhindre at mottageren opplever at linjen er "død".
CODEC	<i>"Coder/Decoder"</i> eller <i>"Compression/Decompression"</i> . For mer informasjon, se kapittel 3.2.
DCR	<i>Degradation Category Rating</i> . For mer informasjon, se B.1.2.2.
DSP	<i>Digital Signal Processor</i> . En type mikroprosessor som er spesiallaget for sanntids prosessering av digitale signaler som stammer fra den analoge verden.
DTMF	<i>Dual Tone MultiFrequency</i> . Bruk av to samtidige toner for å slå nummeret: én tone for rad og én for kolonne. På denne måten utgjør de vanlige 12 tastene på en telefon en matrise.
HTTP	<i>Hyper Text Transfer Protocol</i> . Denne protokollen brukes av web browsere og -servere til å overføre filer, f.eks bildefiler og html-filer.
ICAO	<i>International Civil Aviation Organization</i> . For mer informasjon, se www.icao.org .
IETF	<i>Internet Engineer Task Force</i> . For mer informasjon, se www.ietf.org .
IP	<i>Internet Protocol</i> . En upålitelig forbindelsesløs pakketjeneste. Levering av pakkene garanteres ikke, men i utgangspunktet skal tjenesten prøve å levere pakkene etter beste evne, såkalt "best-effort delivery".
ITU	<i>International Telecommunication Union</i> . For mer informasjon, se www.itu.org .
ITSP	<i>Internet Telephony Service Provider</i> . Tilbyder av IP-telefoni tjenester over Internett.
ISDN	<i>Integrated Services Digital Network</i> .
ISP	<i>Internet Service Provider</i> . Tilbyder av Internett-tjenester.
LAN	<i>Local Area Network</i> . Høyhastighetsdatanettverk med lav feilrate som dekker et relativt lite geografisk området, som regel opptil et par tusen meter.
MAC	<i>Medium Access Control</i> . Lag 2 i TCP/IP-stakken.
Mbone	<i>Multicast Backbone</i> . En del av Internett som er designet for å støtte multicasting.

MCU	<i>Multipoint Control Unit</i> . Se appendiks A.1.1.
MIPS	<i>Millions of Instructions per second</i> .
MOS	<i>Mean Opinion Score</i> . En fempunktsskala for subjektiv talekvalitetsbedømmelse.
PABX/	<i>Private Automatic Branch Exchange/ Private Branch Exchange</i> . En privat
PBX	telefonsentral koblet opp mot det offentlige telefonnettet.
PAM	<i>Pulse Amplitude Modulation</i> . Sampling av et signal med en konstant frekvens. For mer informasjon, se kapittel 4.1.2.
PCM	<i>Pulse Code Modulation</i> . Prosessen med å digitalisere sampler. For mer informasjon, se kapittel 4.1.3.
PSTN	<i>Public Switched Telephony Network</i> . Den engelske betegnelsen for vanlig, tradisjonell fasttelefoni. Også betegnet som " <i>Plain Old Telephony System</i> " (POTS).
PTT	<i>Push-to-talk</i> . En flygeleder må trykke en knapp for å skru på mikrofonen før de formidler en beskjed.
QoS	<i>Quality of Service</i> . På norsk: tjenestekvalitet. Et mål for kvaliteten på talen som blir levert til en bruker. For mer informasjon, se kapittel 3.1.
RAS	<i>Registration, Admission and Status</i> . Se appendiks A.1.2.1.
RSVP	<i>Resource Reservation Protocol</i> . En nettverkskontroll protokoll som brukes for å reservere en viss QoS på overføringen av datapakker over et nettverk.
RTCP	<i>Real-Time Control Protocol</i> . Se appendiks A.1.2.4.
RTP	<i>Real-Time Protocol</i> . Brukes i "Voice over IP" for sanntidsoverføring av talen. For mer informasjon, se appendiks A.1.2.1.
SDP	<i>Session Description Protocol</i> . Beskriver i tekstformat de sesjonsparametre oppkalleren ønsker i prioritert rekkefølge. For mer informasjon, se appendiks A.2.4.
SIP	<i>Session Initiation Protocol</i> . Initialiserer sesjoner mellom brukere. For mer informasjon, se appendiks A.2.
SNR	<i>Signal-to-noise ratio</i> . Måles i decibel. Jo høyere SNR, desto bedre talekvalitet.
SS7	<i>Signaling System number 7</i> . Et signaleringssystem som blant annet brukes i ISDN.
TCP	<i>Transmission Control Protocol</i> . Forbindelsesorientert transportprotokoll. Ligger på transportlaget og yter en pålitelig full-dupleks overføring av data.
UAC	<i>User Agent Client</i> . I endepunktene er det implementert en <i>User Agent</i> -software. UAC har ansvaret for å etablere nye kall
UAS	<i>User Agent Server</i> . Har ansvaret for å motta kall. Se forøvrig <i>UAC</i> .
UDP	<i>User Datagram Protocol</i> . Forbindelsesløs protokoll som ligger på transportlaget. En enkel protokoll som sender og mottar datagrammer uten kvittering eller garantert levering.
VAD	<i>Voice Activity Detection</i> . Deler av talen med stillheten blir fjernet for å spare båndbredde.
VCCS	<i>Voice Communications and Control System</i> . Det kommunikasjonssystemet som brukes i luftfartskontroll i dag. For mer informasjon, se kapittel 1.3.
VoIP	<i>Voice over IP</i> . Overføring av tale over et IP-basert pakkesvitsjet nettverk med de samme funksjonalitetene, pålitelighet og talekvalitet som i vanlig telefoni. For mer informasjon, se kapittel 2.1.
WFQ	<i>Weighted Fair Queuing</i> . En mekanisme for å løse opphopningsproblemer. Denne algoritmen sørger for å identifisere pakker fra ulike datastrømmer og skiller ut de ulike pakkene. Videre fordeles kapasiteten noenlunde likt ("fairly") mellom datastrømmene

Appendiks D

Kildekode

D.1 dgramServer

Kildekoden for den endelige utgaven av serveren, *dgramServer*, følger under.

```
/*Et serverprogram basert på UDP, som skal kunne lytte til en divert socket. En iptables-regel vil
sende IP-pakkene med tale fra gnugk til dette programmet. Programmet har mulighet for å omstokke
IP-pakkene, slik at den naturlige oppstokingen i et nettverk kan simuleres. Videre kan man inn-
stille antall millisekunder pakkeforsinkelse ELLER en viss prosent pakketap*/

#include <pthread.h>
#include <netdb.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <sys/time.h>
#include <sys/types.h>

#define IPPROTO_DIVERT 254 //Divert har protokollnummer 254
#define DIVERT_PORT 9874 //Portnr prog. lytter til for innkomne pakker via divert

int fromlen; //lengden på mottagerens socket-adresse struktur
int percentageLoss;
int packetLatency;
int pLoss;
int pLatency;
size_t bufferSize; //Størrelsen på input-bufferet
pthread_t InputThread, OutputThread; //To threads
```


Appendiks D: Kildekode

```
struct timespec setReleaseTime(int);
struct sockaddr_in from;

void InputPackets(int);
void OutputPackets(int);

pthread_mutex_t threadMutex=PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t threadCond=PTHREAD_COND_INITIALIZER;

//Funksjon for å gi ut feilmeldinger;
void error (char* msg)
{
    perror(msg);
    exit(0);
}

struct timespec setPacketTime(int TimeParam)
{
    struct timespec tempTime;
    struct timeval now;
    long int randomTime,microSec,modTime,deltaSec,extraTime;

    gettimeofday(&now,NULL);

    if(pLoss>0||((pLoss==0)&&(pLatency==0)))
    {
        //randomTime=(long int)(drand48()*100000.0);
        //extraTime=randomTime;
        extraTime=TimeParam;
    }
    else if(pLatency>0)
    {
        //randomTime=(long int)(drand48()*100000.0);
        //extraTime=randomTime;
        extraTime=(long int)(TimeParam*1000);
    }

    //Omgjør setTime til sekunder og nanosekunder;
    microSec=now.tv_usec+extraTime;
    modTime=microSec%1000000;
    deltaSec=(microSec-modTime)/1000000;
    tempTime.tv_sec=now.tv_sec+deltaSec;
    tempTime.tv_nsec=modTime*1000;
    return tempTime;
}

//Funksjon som gir en pakke tid for utsendelse
//Avsenderens adresse.

//Funksjone som håndterer inngående pakker til progr
//Funksjone som håndterer utgående pakker til progr

//Lager en mutex som brukes for å signalisere outputPackets
//Lager en condition som trigger signaliseringen av outputPackets

//trekker tilfeldig tall mellom 0 og 100ms(=100 000 usec)..
//..brukes for til å simulere omstokkingen i IP-netterk.
//her er TimeParam=0.

//trekker tilfeldig tall mellom 0 og 100ms(=100 000 usec)..
//..brukes for til å simulere omstokkingen i IP-netterk.
//forsinkelsen gitt på kommandolinjen (i usec).
```

Appendiks D: Kildekode

```
//Definisjon av strukturen som inngår i pekerkjeden:
struct packet
{
    int packetSize; //størrelsen på IP-pakken
    unsigned char* payload; //bufferet det skal leses inn til fra eth
    struct timespec packetTime; //pakkens tid for utsendelse
    struct packet* next; //peker til neste struct packet.danner lenket liste.
}*firstPacket;

//En funksjon som håndterer inngående pakker,
//og etablerer en pekerliste av pakker (lenket liste) i den rekkefølgen de skal sendes ut i:
void InputPackets(int sockInput)
{
    int n;
    int dummyVar=0;
    struct packet *newPacket; //En peker til en ny innkommende pakke
    struct packet *searchPacket; //En peker som brukes til å finne riktig plass i listen.
    fromlen=sizeof(struct sockaddr_in); //Initialiserer lengden på adresse strukturen.
    fprintf(stderr,"Venter paa innkomne pakker..\n");
    while(1)
    {
        //Oppretter en ny pakke
        newPacket=(struct packet*)malloc(sizeof(struct packet)); //allokerer minneområde for newPacket
        newPacket->payload=(char*)malloc(bufferSize); //allokerer minneområde for newPacket sin payload
        newPacket->next=NULL; //nestepekeren blir initialisert
        newPacket->packetSize=0; //pakkestørrelsen blir initialisert

        n=recvfrom(sockInput,newPacket->payload,bufferSize,0,
            (struct sockaddr*)&from,&fromlen);
        newPacket->packetSize=n; //recvfrom returnerer mengden bytes som mottas
        fprintf(stderr,"+"); //skriver ut en '+' til skjerm for hver mottatte pakke
        if(n<0) error ("\nError in the reception of the packet.\n");

        //Om nummeret som trekkes er større en packetLoss eller hvis ingen pakketap ønskes, fortsettes det videre..:
        if((percentageLoss==0)|| (rand()>percentageLoss))
        {
            if(packetLatency>0) //har pakkeforsinkelse blitt angitt?...
            {
                newPacket->packetTime=setPacketTime(packetLatency); //tid for utsending blir kalkulert
            }
            else //...Nei..
            {
                newPacket->packetTime=setPacketTime(dummyVar); //ingen forsinkelse beregnes
            }
        }
    }
}
```

Appendiks D: Kildekode

```
//Pakkene blir sortert i den lenkede listen etter stigende tid for utsendelse.Slik blir pakkene omrokkert...
//...hvis det blir lagt til en tilfeldig tid i packetTime:
if(firstPacket==NULL)
    firstPacket=newPacket; //Det finnes ingen pakker fra før
//..fikk nettopp inn én!
else if(((newPacket->packetTime.tv_sec)<(firstPacket->packetTime.tv_sec))|| //Ny firstPacket -ALERT,ALERT!!
        ((newPacket->packetTime.tv_sec)==(firstPacket->packetTime.tv_sec)&&
        (newPacket->packetTime.tv_nsec)<(firstPacket->packetTime.tv_nsec)))
    {
        newPacket->next=firstPacket;
        firstPacket=newPacket;
        pthread_cond_signal(&threadCond); //outputPackets() må varsles umiddelbart!
    }
else //Det finnes pakker i listen fra før av
    {
        searchPacket=firstPacket;
        //Forts pakker i listen OG searchPacket->next->packetTime er mindre enn newPacket sin packetTime:
        while((searchPacket->next!=NULL)&&
              ((searchPacket->next->packetTime.tv_sec<newPacket->packetTime.tv_sec)||
              ((searchPacket->next->packetTime.tv_sec==newPacket->packetTime.tv_sec)&&
              (searchPacket->next->packetTime.tv_nsec<newPacket->packetTime.tv_nsec))))
            {
                searchPacket=searchPacket->next; //Gå videre i listen
            }
        newPacket->next=searchPacket->next; //Plasserer newPacket på riktig plass i listen
        searchPacket->next=newPacket;
    }
}
else
    free(newPacket); //forkaster pakket
}
}

void OutputPackets(int sockOutput)
{
    int n;
    struct packet *tempPacket; //Temp. peker til pakken,blir slettet etter at pakken er sendt
    while(1)
    {
        tempPacket=NULL;
        if(firstPacket!=NULL) //Kun relevant om vi faktisk har pakker fra før av
        {
            pthread_mutex_lock(&threadMutex);
            if(pthread_cond_timedwait(&threadCond,&threadMutex,&firstPacket->packetTime)==110)
                //Pakken vil overholde forsinkelsen, har ingen ny firstPacket
            {
                tempPacket=firstPacket; //Tar vare på nåværende firstPacket
            }
        }
    }
}
```

Appendiks D: Kildekode

```
    firstPacket=firstPacket->next; //Flytter pekeren til nåv. første pakke til plassnr. to i listen
    pthread_mutex_unlock(&threadMutex);
    n=sendto(sockOutput,tempPacket->payload, //Sender pakken avgårde
            tempPacket->packetSize,0,(struct sockaddr*)&from,fromlen);
    fprintf(stderr,"-"); //En pakke ble nettopp sendt avgårde
    if(n<0)
        error("\nError in the sending of the packet to the receiver\n");
    if(n!=tempPacket->packetSize)
        fprintf(stderr,"\nFEIL: Ble innsatt %i bytes, skulle vært %i bytes.\n"
                ,n,tempPacket->packetSize);
    free(tempPacket); //Fjerner firstPacket som nettopp ble sendt fra listen
}
else //har mottatt et SIGNAL,dvs vi har mottatt en ny firstPacket..
    pthread_mutex_unlock(&threadMutex); //..Må avbryte holdingen av pakken øyeblikkelig
}
}

//main-funksjonen som initialiserer socketene og oppretter to threads
int main(int argc, char* argv[])
{
    int divertedPackets, threadReturn,fromlen,packetLoss,sockInput,sockOutput; //Diverse deklarasjoner
    int yes=1;
    int ret, lengthEth_addr;
    pthread_attr_t thread1Attr;
    pthread_attr_t thread2Attr;
    struct sched_param thread1Priority;
    struct sched_param thread2Priority;
    struct sockaddr_in Eth_addr; //serverens socket-adresse struktur for eth.

    firstPacket=NULL; //Initialisering av firstPacket.
    percentageLoss=0;
    packetLoss=0;
    packetLatency=0;
    ret=0;
    srand48(time(0)); //Setter frø til random-funksjonen
    bufferSize=150;

    if (argc<2)
    {
        fprintf(stderr,"Feil kommandolinje: [Prosentvis pakketap] [ms forsinkelse].
        \nValg: f.eks. '10' for ti prosent pakketap eller '15' for 15 ms forsinkelse paa pakkene.\n
        Gi med '0'som parameter om pakketap/forsinkelse ikke ønskes.\n");
        exit(1);
    }
}
```

Appendiks D: Kildekode

```
}
packetLoss=atoi(argv[1]); //leser inn fra kommandolinje
packetLatency=atoi(argv[2]);

if(packetLoss>0 && packetLatency==0)
{
    percentageLoss=(long) (((double)packetLoss/100.0)*(double)RAND_MAX); //konverterer konf. prosent til et tall
    fprintf(stderr,"packetLoss=%d\n",packetLoss);
    fprintf(stderr,"percentageLoss=%d\n",percentageLoss);
    pLoss=1;
    pLatency=0;
}

else if(packetLatency>0 && packetLoss==0)
{
    fprintf(stderr,"packetLatency=%d ms\n",packetLatency);
    pLatency=1;
    pLoss=0;
}

else if(packetLatency==0 && packetLoss==0)
{
    fprintf(stderr,"packetLatency=0 og packetLoss=0\n");
    pLoss=pLatency=0;
}

else
{
    fprintf(stderr,"** Ikke gyldig valg av parameter **\n");
    exit(1);
}

fprintf(stderr,"Oppretter socketen..\n");
divertedPackets=socket(PF_INET, SOCK_RAW, IPPROTO_DIVERT);
lengthEth_addr=sizeof(Eth_addr);
if (divertedPackets== -1)
{
    error("Noe gikk galt i åpningen av socketen for eth0-read\n");
    exit(1);
}
if(setsockopt(divertedPackets,SOL_SOCKET,SO_REUSEADDR,&yes,sizeof(int))== -1) //Sørger for gjenbruk av portnr. til eth0
{
    fprintf(stderr,"Feil under bruk av socket-option\n");
    exit(1);
}

//Setter opp adresse-familie, ip-adr og portnummer:
bzero(&Eth_addr, lengthEth_addr); //Initialiserer adresse-strukturen.
```

Appendiks D: Kildekode

```
Eth_addr.sin_family=AF_INET; //Vi har Internett-socket
Eth_addr.sin_port=htons(DIVERT_PORT); //Setter portnr som det skal opprettes en forb.til
Eth_addr.sin_addr.s_addr=0; //Setter ip-adr til maskinen manuelt

fprintf(stderr,"Eth sin adresse er %s\n",inet_ntoa(Eth_addr.sin_addr));

//Binder socket-adressen til eth0R til socket-deskriptoren:
fprintf(stderr, "Binder socket'en..\n");
ret=bind(divertedPackets,(struct sockaddr*)&Eth_addr,sizeof(struct sockaddr_in));

if (ret!=0)
{
    close(divertedPackets);
    error("Noe gikk galt i bindingen av socketen for eth0");
    exit(2);
}
sockInput=divertedPackets;
sockOutput=divertedPackets;

pthread_attr_setschedpolicy(&thread1Attr,SCHED_RR);
pthread_attr_setschedpolicy(&thread2Attr,SCHED_RR);

//Oppretter to threads: én for å håndtere inngående pakker, én for utgående pakker:
threadReturn=pthread_create(&InputThread,NULL,(void*)InputPackets,(void *)sockInput);
if(threadReturn<0)
    error("Feil i opprettelsen av InputThread:\n");

threadReturn=pthread_create(&OutputThread,NULL,(void*)OutputPackets,(void *)sockOutput);
if(threadReturn<0)
    error("Feil i opprettelsen av OutputThread:\n");

thread1Priority.sched_priority=1;
thread2Priority.sched_priority=99;

pthread_setschedparam(InputThread,SCHED_RR,&thread1Priority);
pthread_setschedparam(OutputThread,SCHED_RR,&thread2Priority);

pthread_join(InputThread,NULL); //OutputThread sover inntil InputThread avslutter.
pthread_join(OutputThread,NULL); //InputThread sover inntil OutputThread avslutter.
}
```

D.2 dgramClient

Kildekoden til klienten, dgramClient følger under:

```
/* UDP client in the internet domain */
#include <pthread.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <sys/types.h>
#include <netinet/in.h>

#define PORT 8686 //Serverens portnr på eth0

void error(char *);
void sendPackets(void);
void receivePackets(void);

int sock, length;
size_t payloadSize;
struct sockaddr_in server, from;
pthread_t SendThread,ReceiveThread;

struct payload
{
    long int seqNo;
    struct timeval timeStamp;
};

void sendPackets()
{
    int r, n;
    struct payload packetPayload;
    struct timeval wait, sendTime;
    packetPayload.seqNo=0;

    wait.tv_sec=0;
    wait.tv_usec=30000;
```

Appendiks D: Kildekode

```
for(r=0;r<15000;r++)
{
    gettimeofday(&sendTime,NULL);
    packetPayload.timeStamp=sendTime;
    packetPayload.seqNo++;
    n=sendto(sock, &packetPayload, payloadSize, 0,
            (struct sockaddr*)&server,length);
    if(n<0) error("Feil i sendingen av pakker til motparten\n");
    select(0,NULL,NULL,NULL,&wait);
}
}

void receivePackets()
{
    int r, p;
    struct payload recvPayload;
    struct timeval recvTime;
    for(p=0;p<15000;p++)
    {
        r=recvfrom(sock, &recvPayload,payloadSize, 0,
                (struct sockaddr*)&from,&length);
        if(r<0)error("Feil i mottagelsen fra serveren\n");
        gettimeofday(&recvTime, NULL);
        fprintf(stderr, "%d\n",
                ((recvTime.tv_sec-recvPayload.timeStamp.tv_sec)*1000
                +(long int)((double)(recvTime.tv_usec-
                recvPayload.timeStamp.tv_usec)/1000.0)-100));
    }
}

int main(void)
{
    int threadReturn1,threadReturn2;
    struct sched_param thread1Priority;
    struct sched_param thread2Priority;

    payloadSize=20; //20 bytes

    sock=socket(AF_INET, SOCK_DGRAM, 0);
    if (sock < 0) error("socket");
    server.sin_family = AF_INET;
    server.sin_addr.s_addr=inet_addr("128.39.213.110"); //IP-adr til eth0

    server.sin_port=htons(PORT);
```


Appendiks D: Kildekode

```
length=sizeof(struct sockaddr_in);

threadReturn1=pthread_create(&SendThread,NULL,(void*)sendPackets,NULL);
if(threadReturn1<0)
    error("Feil i opprettelsen av InputThread:\n");

threadReturn2=pthread_create(&ReceiveThread,NULL,
                             (void*)receivePackets,NULL);
if(threadReturn2<0)
    error("Feil i opprettelsen av InputThread:\n");

thread1Priority.sched_priority=1;
thread2Priority.sched_priority=99;

pthread_setschedparam(SendThread,SCHED_RR,&thread1Priority);
pthread_setschedparam(ReceiveThread,SCHED_RR,&thread2Priority);

pthread_join(SendThread,NULL);
pthread_join(ReceiveThread,NULL);

close(sock);
exit(1);
}
void error(char *msg)
{
    perror(msg);
    exit(0);
}
```

Appendiks E

Personlig e-post

E.1 Fra Norsk Luftfartsmuseum

----- Original Message -----

From: [Per Arvid Pettersen](#)

To: [Flymuseum](#)

Sent: Friday, July 12, 2002 4:36 PM

Subject: Re: [Fwd: Spørsmål]

Hei igjen,

Som tidligere nevnt er dette et spørsmål det er vanskelig å svare eksakt på fordi det ikke er forsket spesielt på emnet. Generelt kan man si:

Utlandet:

- Radiokommunikasjon mellom bakke og fly ble utviklet under siste del av 1. verdenskrig.
- Da Alcock og Brown fløy over Atlanderhavet for første gang i 1919 hadde de radioutstyr, men rekkevidden var svært begrenset og ble ikke brukt som navigasjonshjelpemiddel.
- Inntil 1928 måtte flygere stole på astronavigasjon over vann. Over land ble i tillegg f.eks. jernbanelinjer, bynavn malt på hustak, lyskastere om natten, o.l., brukt til å navigere etter.
- I 1928 ble retningsgivende radio-beacon, spesielt beregnet for fly (retningen vist på instrument i flyet), benyttet med suksess under flyging fra Oakland til Hawaii.

- Rundt 1930 hadde alle store flyselskaper radiopeilere på bakken. Disse kunne på forespørsel gi flygeren retning til senderen.
- Da Howard Hughes i 1938 satte verdensrekort i flyging verden rundt, hadde han organisert en kjede med skip og bakkestasjoner utstyrt med radiofyr. Flyet hadde hele tiden kontakt med minst en stasjon.
- I 1935 utviklet Pan Am et retningsgivende radiofyr med en rekkevidde på 2.400 miles.

I Norge:

- Ca. 1925: Radiopeiling/kommunikasjon: Kystradiopeilesendere bygges ut langs kysten. Disse var spesielt beregnet for sjøfarten, men kunne også benyttes av fly hvis man hadde utstyr for dette.
- I 1927/28/29 fikk ordinære fyrstasjoner som Fuglehuk, Færder og Røst slikt utstyr. Disse ble nå i stand til å peile/fastslå retningen på et radiosignal.
- I 1935, da DNL fikk konsesjon på ruteflyging i Norge, ble det opprettet telegrafstasjoner som skulle kommunisere med luftfarten. Etter 2. verdenskrig ble dette systemet oppgradert til radiotelefoni. DNL installerte radiokommunikasjon mellom flystasjon-fly. Disse stasjonene hadde svært kort rekkevidde (aeradiostasjoner på Sola, Fornebu/Gressholmen, Bergen, Kjevik/Kristiansand).
- I 1936 fikk DNL's fly installert peileutstyr for blant annet å kunne peile NRKs radiostasjoner. På grunn av manglende trening og prosedyrer forhindret ikke dette Norges første alvorlige flyulykke.
- Etter flyulykken i 1936 ble flyradionavigasjonssystemet i Norge bygget kraftig ut.

Dette var noen data. På tross av omtrentligheter og unøyaktigheter håper jeg dette kan brukes. Kilde for opplysningene om norske forhold er:

- Telegrafverkets historie av Torolf Rafto
- Norsk Luftfartsmuseum/DNLs arkiv

Lykke til og god sommer!

Vennlig hilsen
Norsk Luftfartsmuseum
Per Arvid Pettersen
Rådgiver

E.2 Fra flygeleder

----- Original Message -----

From: [Konstantinos Patouras](#)

To: [Thomas Tomter](#)

Sent: Monday, April 14, 2003 11:24 PM

Subject: RE: An ATC-question

Hi Thomas

If I transmit a message to an airplane and I don't get read back I wait for 10 seconds before I retransmit. That is what is recommended in the Radiotelephony manual.

Did you get any assessments by my colleagues?

Regards,

Kostas Patouras

Appendiks F

Teknisk spesifikasjon

I dette appendikset gjengis de tekniske spesifikasjonene for Quicknet²⁶³-kortene "Internet PhoneJACK" og "Internet PhoneCARD". Til slutt vises en teknisk tegning som illustrerer hvordan en forgrener er laget.

F.1 "Internet PhoneCARD"

De tekniske dataene gjengitt her, er oppgitt av produsenten på e-post og på emballasjen til kortet:

PHYSICAL CONNECTORS

- Phone device port, RJ-11 (with SmartCable adapter)
- One 2.5mm stereo jack (with SmartCable adapter): Microphone/Speaker audio in/out.
- Form Factors: PC Card, Type II PCMCIA

ANALOG AUDIO PORT

- Microphone impedance: 600 Ohms nominal
- Speaker output impedance: 32 Ohms nominal

SYSTEM REQUIREMENTS

- Pentium PC or higher
- 32 MB RAM
- Windows 98/98SE/ME/NT or 2000
- Available PCMCIA slot
- Internet Connection

²⁶³ www.quicknet.net

AUDIO

- DSP
- Full duplex
- DTMF decoding & generation
- Call progress tone generation
- Echo cancellation
- H.323/H.324, m-Law/A-Law, G.711, G.723.1, G.728, G.729A & Truespeech

OPERATING ENVIRONMENT

- Operating Temperature: 0-55°C
- Humidity: Up to 80% non-condensing
- Storage: -40°C to 85°C

F.2 "Internet PhoneJACK"

De tekniske dataene gjengitt her, er oppgitt av produsenten på e-post og på emballasjen til kortet:

PHYSICAL CONNECTORS

- Phone device port, RJ-11
- (2) 3.5mm stereo jacks (not available in the Lite version)
- Form Factors: single slot
 - 1/2 size PCI bus card
 - 1/2 size ISA bus card

RJ-11 PORT

- On hook voltage: -24VDC±5VDC
- Nominal impedance: 900 Ohms
- Ring Source: Square Wave 66VDC Nominal@20Hz
- Ring Drive: 3 REN

ANALOG AUDIO PORT

- Microphone impedance: 600 Ohms nominal
- Speaker output impedance: 32 Ohms nominal

SYSTEM REQUIREMENTS

- Pentium PC or higher
- 32 MB RAM
- Windows 98/98SE/ME/NT or 2000
- Available ISA or PCI slot
- Internet Connection

AUDIO

- DSP
- Full duplex

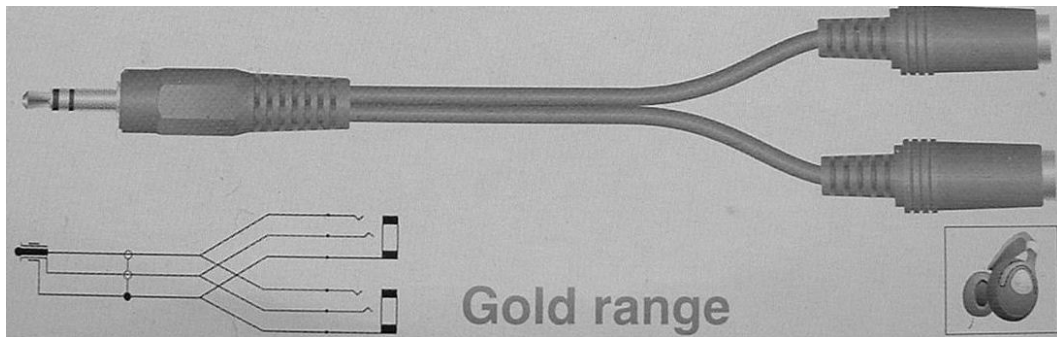
- DTMF decoding & generation
- Call progress tone generation
- Echo cancellation
- H.323/H.324, m-Law/A-Law, G.711, G.723.1, G.728, G.729A & Truespeech

OPERATING ENVIRONMENT

- Operating Temperature: 0-55°C
- Humidity: Up to 80% non-condensing
- Storage: -40°C to 85°C

F.3 Forgrener

Her gis det et bilde som illustrerer hvordan signalene ligger i en forgrener. Bildet er tatt av baksiden på pakningen til komponent 2 i figur 6.4, og viser hvordan et stereosignal på venstre siden av forgreneren blir ”kopierte” til to identiske og parallelle signaler på høyre siden.



Figur F.1: Baksiden av emballasjen til komponent 2 i figur 6.4

Appendiks G

Manus og evalueringsskjema

I dette appendikset vil manuset og evalueringsskjemaene som ble brukt i talekvalitetsbedømmelsene (både på norsk og engelsk) gjengis. Det vil kun bli gjengitt skjemaet for *serie 1* i og med at det samme skjemaet ble brukt for alle seriene.

G.1 Manus

Eksperimentnr.: 1	
A:	Copenhagen Radar, Air France One niner niner two over Alpha Romeo Alpha request descent.
B:	Air France One niner niner two, descend to Flight Level One seven zero, report leaving three seven zero.
A:	Descend to Flight Level one seven zero, leaving now Flight Level three seven zero, Air France One niner niner two.
A:	Air France One niner niner two, contact Copenhagen Approach one one niner decimal one.
B:	One one niner decimal one, Air France One niner niner two.

Eksperimentnr.: 2	
A:	Hello Cairo, copy estimate over Kilo Uniform Mike Bravo India.
B:	Go ahead.
A:	Mike Sierra Romeo eight eight zero.
B:	Mike Sierra Romeo eight eight zero, Airbus three zero zero, destination Cairo.
A:	Correct, estimation Kilo Uniform Mike Bravo India at two three zero six, Flight Level two niner zero.
B:	Copied, over Kilo Uniform Mike Bravo India two three zero six, Flight Level two niner zero.

Eksperimentnr.: 3

B:	Sofia, copy re-clearance regarding Delta Alpha November Four five zero.
A:	Go ahead.
B:	Delta Alpha November Four five zero, recleared Flight Level Three five zero over Tango India Mike Uniform Romeo at zero niner zero two.
A:	Copied, Delta Alpha November Four five zero recleared Flight Level three five zero, Tango India Mike Uniform Romeo zero niner zero two.

Eksperimentnr.: 4

A:	Fastair Three Four Five, cleared take-off runway two eight report airborne.
B:	Cleared take-off runway two eight Wilco. Fastair Three Four Five.
B:	Fastair Three Four Five airborne. Contact Radar One Two One decimal Seven Five.
A:	Radar One Two One decimal Seven Five, Fastair Three Four Five.

Eksperimentnr.: 5

B:	Golf Charlie Delta, report passing Flight Level eight zero.
A:	Report passing Flight Level eight zero, Golf Charlie Delta.
A:	Golf Charlie Delta Passing Flight Level eight zero.
B:	Golf Charlie Delta, maintain two thousand five hundred feet.
A:	Maintaining two thousand five hundred feet, Golf Charlie Delta.

Eksperimentnr.: 6

A:	Golf Charlie Delta, climb Flight Level Seven Zero.
B:	Climb Flight Level Seven Zero, Golf Charlie Delta.
B:	Golf Charlie Delta reaching Flight Level seven zero.
A:	Golf Charlie Delta request descent.
B:	Golf Charlie Delta, descend Flight Level six zero.
A:	Descend Flight Level six zero, Golf Charlie Delta.

Eksperimentnr.: 7

B:	Fastair Three Four Five, after passing North Cross descend Flight Level eight zero.
A:	After passing North Cross descend Flight Level eight zero Fastair Three Four Five.
B:	Fastair Three Four Five, stop descent Flight Level two one zero
A:	Stop descent Flight Level two one zero, Fastair three four five

Eksperimentnr.: 8

B:	Fastair three four five, when ready climb Flight Level two eight zero.
A:	When ready climb Flight Level two eight zero, Fastair three four five.
B:	Fastair three four five, leaving Flight Level two zero zero climbing Flight Level two eight zero.
A:	Fastair three four five.

Eksperimentnr.: 9

B:	Fastair four five six line up and wait- vehicle crossing upwind end of runway
A:	Line-up and wait Fastair four five six. Cleared take-off, Fastair four five six.
B:	Fastair four five six cleared take off

Eksperimentnr.: 10

A:	Golf Charlie Delta hold position, cancel I say again cancel take-off, acknowledge.
B:	Hold position, Golf Charlie Delta.
A:	Fastair five six seven, stop immediately, I say again, Fastair five six seven stop immediately, acknowledge.
B:	Stopping, Fastair five six seven.

G.2 Evalueringsskjema

G.2.1 Norsk

Evalueringsskjema

NB! Det bes om at hvert lydopptak bedømmes umiddelbart etter avspilling. Med "sample number" menes nummeret på lydopptaket.

Serie 1

1. Vær vennlig og sett *ett* kryss per lydopptak for din mening om talekvaliteten på konversasjonen du overhørte:

Quality of the speech	Sample Number					
	1	2	3	4	5	6
Excellent						
Good						
Fair						
Poor						
Bad						

2. Vær vennlig og sett *ett* kryss per lydopptak for din mening om hvor mye innsats som var nødvendig for å forstå meningen med talen:

Effort required to understand the meanings of the sentences.	Sample Number					
	1	2	3	4	5	6
Complete relaxation possible; no effort required.						
Attention necessary; no appreciable effort required.						
Moderate effort required.						
Considerable effort required.						
No meaning understood with any feasible effort.						

3. Sett *ett* kryss per lydopptak for om talekvaliteten var tilstrekkelig for bruk i luftfartskommunikasjon:

Sample Number	Sufficient Quality?	
	Yes	No
1		
2		
3		
4		
5		
6		

4. Eventuelle kommentarer til talekvaliteten, hvordan du opplevde lydopptakene eller andre kommentarer. Vennligst angi for hvilke(t) "sample number" kommentarene gjelder:

G.2.2 Engelsk

Evaluation form

Please assess each sound recording before playing the next one.

Series 1

1. Please check *one* box per sound recording for your opinion about the voice quality in the conversation you just overheard:

	Sample Number					
Quality of the speech	1	2	3	4	5	6
Excellent						
Good						
Fair						
Poor						
Bad						

2. Please check *one* box per sound recording for your opinion about how much effort that was required to understand the meanings of the sentences:

	Sample Number					
Effort required to understand the meanings of the sentences.	1	2	3	4	5	6
Complete relaxation possible; no effort required.						
Attention necessary; no appreciable effort required.						
Moderate effort required.						
Considerable effort required.						
No meaning understood with any feasible effort.						

3. Please check *one* box per sound recording for your opinion about whether the voice quality was sufficient for Air Traffic Control:

	Sufficient Quality?	
Sample Number	Yes	No
1		
2		
3		
4		
5		
6		

4. Please write down any comment you may have regarding the voice quality, how you experienced the sound recordings or other comments. Please state which sample(s) your comments applies:

Appendiks H

Konfigurasjon

I dette appendikset vil konfigurasjonsfilen til gnugk, scriptet som ble brukt til å etablere IP-tables filtreringsreglene og innstillingene i OpenPhone bli gjengitt.

H.1 Gatekeeper

Konfigurasjonsfilen²⁶⁴ som ble brukt i gnugk er som følger:

```
[Gatekeeper::Main]
Fourtytwo=42
TimeToLive=600

[RoutedMode]
GKRouted=1
H245Routed=0
CallSignalPort=1721
CallSignalHandlerNumber=1
RemoveH245AddressOnTunneling=1
AcceptUnregisteredCalls=1
DropCallsByReleaseComplete=1
SupportNATedEndpoints=0
Q931PortRange=30000-39999
H245PortRange=40000-49999

[Proxy]
Enable=1
T120PortRange=50002-59999
RTPPortRange=7600-7600

[GkStatus::Auth]
rule=allow
```

²⁶⁴ For mer informasjon om syntaks og liknende, se www.gnugk.org.

H.2 IP-tables

Scriptet som ble benyttet til å sette filtreringsreglene²⁶⁵ i IP-tables, som sørget for at IP-pakkene ble omdirigert til dgramServer, er gjengitt her:

```
#####
# Et lite firewall-script som sørger for at dgramServer får levert #
# de r e talepakkene som gnugk sender avg rde til utk en. Dette blir#
# gjort ved   avlede pakkene ved hjelp av divert-target'en i      #
# IP-tables. dgramServer lytter til en divertSockets p  port-    #
# nummer 9874. gnugk ble startet opp med kommandoen gnugk.      #
#####

#!/bin/sh

# Fjerner evtentuelle eksisterende regler i alle Iptables-kjeder:
iptables -F

# UDP-pakker fra port 7600 fra GNUgk skal bli avsk ret og sendt til
#dgramServer som lytter til port 9874.

iptables -A OUTPUT -p udp --source-port 7600 -m owner --cmd-owner
gnugk -j DIVERT --div-port 9874

# Reglene blir printet ut til skjerm
iptables -L
```

H.3 OpenPhone

De innstillingene som var valgt i OpenPhone²⁶⁶ er listet opp i det f lgende:

General:

Call intrusion Protection Level: medium

DMTF as H.245 string: enabled

Ikke kryssset av for disabling av Fast-start, H.245 Tunneling eller H.245 in Setup

Network:

LAN 10000 kb/s

TCP port base 3000, max 3049

UDP port base 3000, max 3099

RTP port base 7600, max 7798

IP Type of service: B0(hex)

Gatekeeper:

Use gatekeeper, static host.

Audio Device:

Echo cancellation: high

Codec:

G.723.1A (6,3kbps) {hw} med 3 frames

silence detection: enabled

jitter buffer:100 ms.

Signal: 65 ms

Silence: 400ms

²⁶⁵ For mer informasjon om bruk av IP-tables filtreringsregler, se www.netfilter.org.

²⁶⁶ For mer informasjon om OpenPhone, se www.openh323.org.

Appendiks I

Punkt 1: Quality of the Speech							Punkt 2: Effort required to understand the meanings of the sentences														
Evalueringsnr.	Parameter						200 ms	300 ms	400 ms	600 ms	Evalueringsnr.	Parameter									
	0 %	5 %	10 %	15 %	20 %	25 %						0 %	5 %	10 %	15 %	20 %	25 %	200ms	300ms	400ms	600ms
1	4	3	2	4	3	3	4	5	4	5	1	5	4	2	5	4	3	5	5	4	5
	2	3	3	3	4	4	5	5	4	4		2	4	4	4	5	5	5	5	4	5
	5	5	4	4	4	2	3	2	2	2		5	5	5	4	3	2	4	2	3	2
2	3	3	1	1	2	1	2	3	3	3	2	3	1	2	3	3	1	2	3	4	3
	2	2	1	2	2	1	2	2	3	3		3	3	1	2	2	1	3	3	3	3
	3	2	2	2	1	1	2	2	3	3		3	3	2	2	2	1	2	3	3	3
3	1	1	1	1	1	1	2	2	2	3	3	3	2	2	2	3	2	3	4	4	4
	2	2	2	2	1	2	2	2	2	3		4	3	4	4	3	3	4	4	5	4
	3	2	2	2	2	1	2	2	2	2		4	4	4	4	4	3	4	4	4	4
4	3	4	4	1	2	1	4	5	4	5	4	4	3	4	1	2	2	5	5	5	5
	5	5	1	4	3	3	5	4	4	5		5	5	2	4	2	3	5	4	5	5
	4	5	3	4	3	2	5	4	5	5		4	5	3	4	4	2	5	4	5	5
5	3	2	2	1	1	1	2	3	3	4	5	3	2	2	1	2	2	2	4	3	3
	3	3	2	2	3	1	3	3	4	4		3	3	2	2	2	1	3	3	4	4
	4	4	3	2	3	1	4	4	4	4		3	3	2	2	2	1	3	4	4	4
6	3	2	2	2	2	2	2	3	3	3	6	3	2	2	2	2	2	2	3	3	3
	3	3	2	3	2	2	1	2	2	2		3	3	2	3	2	2	1	2	2	2
	3	3	3	2	2	1	2	2	3	2		3	3	3	2	2	2	3	3	3	3
7	1	2	2	1	1	1	3	3	3	3	7	2	3	3	2	3	2	3	3	3	3
	3	3	1	2	2	1	3	3	3	4		4	4	2	3	3	2	3	3	3	4
	4	3	2	2	2	2	4	4	3	3		4	3	2	2	2	2	4	4	3	3
8	3	2	2	1	2	2	4	4	4	4	8	4	3	3	2	3	3	4	4	4	4
	4	3	3	3	3	3	4	4	4	4		4	3	3	3	3	3	4	4	4	4
	4	4	4	3	3	1	4	5	5	5		4	4	4	3	3	2	5	5	5	5
9	3	3	3	2	3	2	3	4	4	3	9	4	4	4	2	4	2	3	4	4	4
	4	4	4	3	3	3	3	4	4	4		4	4	4	4	4	4	3	3	4	4

Appendiks I: Excel-ark

	4	3	3	2	2	2	3	4	3	4		4	3	3	2	2	2	3	4	3	4
10	3	3	1	2	2	2	2	4	2	5	10	4	4	2	2	2	3	2	5	2	5
	2	3	1	3	2	1	3	3	4	4		3	4	1	4	3	2	4	5	5	5
	3	3	2	2	1	1	4	4	3	2		3	4	3	3	2	1	5	4	3	4
Sum	94	90	68	68	67	51	92	101	99	107	Sum	107	101	82	83	83	66	104	113	111	116
Minimum	1	1	1	1	1	1	1	2	2	2	Minimum	2	1	1	1	2	1	1	2	2	2
Gjennomsnitt	3,1	3,0	2,3	2,3	2,2	1,7	3,1	3,4	3,3	3,6	Gjennomsnitt	3,6	3,4	2,7	2,8	2,8	2,2	3,5	3,8	3,7	3,9
Maksimum	5	5	4	4	4	3	5	5	5	5	Maksimum	5	5	5	4	4	4	5	5	5	5

Punkt 3: Suffient quality?

Parameter

Evalueringnr.	0 %		5 %		10 %		15 %		20 %		25 %		200ms		300ms		400ms		600ms	
	yes	no	yes	no	yes	no	yes	no	yes	no	yes	no	yes	no	yes	no	yes	no	yes	no
1	1		1			1	1		1		1		1		1		1		1	
		1		1		1		1		1		1		1		1		1		1
	1		1		1		1		1			1	1			1		1		1
2		1		1		1		1		1		1		1		1		1		1
		1		1		1		1		1		1		1		1		1		1
		1		1		1		1		1		1		1		1		1		1
3		1		1		1		1		1		1		1		1		1		1
		1		1		1		1		1		1		1		1		1		1
		1		1		1		1		1		1		1		1		1		1
4	1		1		1			1		1		1	1		1		1		1	
	1		1			1	1			1		1	1		1		1		1	
	1		1		1		1		1			1	1		1		1		1	
5		1		1		1		1		1		1		1		1		1		1
		1		1		1		1		1		1		1		1		1		1
		1		1		1		1		1		1		1		1		1		1
6		1		1		1		1		1		1		1		1		1		1

Appendiks I: Excel-ark

		1		1		1		1		1		1		1		1		1		1	
		1		1		1		1		1		1		1		1		1		1	
7		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
		1		1		1		1		1		1		1		1		1		1	
8		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
		1		1		1		1		1		1		1		1		1		1	
9		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
		1		1		1		1		1		1		1		1		1		1	
10		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
		1		1		1		1		1		1		1		1		1		1	
Sum		13	17	13	17	7	23	7	23	4	26	2	28	17	13	17	13	17	13	17	13
Prosent	JA	43,3		43,3		23,3		23,3		13,3		6,7		56,7		56,7		56,7		56,7	
	NEI		56,7		56,7		76,7		76,7		86,7		93,3		43,3		43,3		43,3		43,3

Referanser

- [ADGHP95] R.Aldridge, J.Davidoff, M.Ghanbari, D.Hands, D.Pearson. "Measurement of scene-dependent quality variations in digitally coded television pictures", IEE Proceedings - Vision, Image and Signal Processing, Vol. 142, No. 3, June 1995.
- [Agi00] Agilent Technologies ved Stefan Pracht og Dennis Hardman. "Voice Quality in Converging Telephony and IP Networks", september 2000.
<http://onenetworks.comms.agilent.com/WhitePapers.asp#IP>
- [Agi00a] Agilent Technologies ved Stefan Pracht. "Troubleshooting H.323 Signaling". October, 2000.
<http://onenetworks.comms.agilent.com/WhitePapers.asp#IP>
- [Agi01] Agilent Technologies ved John Anderson. "Methods for Measuring Perceptual Speech Quality". May, 2001.
<http://onenetworks.comms.agilent.com/WhitePapers.asp#IP>
- [Bes87] Lawrence Besaw: "BSD Socket Reference: Berkeley UNIX System Call and Interprocess Communication", 1987. Revised January 1991 by Marvin Solomon. Hentet fra
<http://www.ifi.uio.no/inf103/Litteratur/UNIXsockets.pdf>.
- [BorS98] Michael S. Borella og Debbie Swider. "Internet Packet Loss: Measurements and Implications for End-to-End QoS", Proceedings, International Conference on Parallel Processing, Aug. 1998.
URL: <http://citeseer.nj.nec.com/borella98internet.html>.
- [BoS99] A. Bouch og M.A Sasse. "Network Quality of Service- An Integrated Perspective". Proc. RTAs'99, Vancouver 1- 3 June, 1999.
- [BoS99b] Bouch og M.A. Sasse. "Network Quality of Service: What do users need?", In Proceedings of the 4th International Distributed Conference (IDC'99), 1999.
<http://citeseer.nj.nec.com/336987.html>
- [BrK99] Brooktrout Technology ved Alan Percy. "Understanding Latency in IP Telephony". <http://www.brooktrout.com>, februar 1999.
- [BrT02] British Telecommunications. "Perceptual analysis modelling", 2002. Hentet fra
<http://www.btexact.com/research/pastprojects?doc=42163&page=2>.
- [BSD01] A. Bouch, M.A. Sasse, H. DeMeer. "Of packets and people: A user-centered approach to Quality of Service" in Proc. IWQoS 2001.
<http://www.cs.ucl.ac.uk/staff/A.Sasse/pub.html>.
- [CAA96] Civil Aviation Authority: "Radiotelephony Manual". Sixth Edition January, 1996. ISBN 0 86039 644 4
- [Che95] Juin-Hwey Chen. "Low-Delay Coding of Speech", kapittel 6 i "Speech Coding and Synthesis", Elsevier Science B.V, W.B Kleijn og K.K. Paliwal (editors), Amsterdam, 1995.

- [Cis01] Cisco Tech Notes. "Waveform Coding Techniques". Cisco Systems Inc. 2001. Hentet fra http://www.cisco.com/warp/public/788/signalling/waveform_coding.html, 26/3-02.
- [Cis01b] Cisco Tech Notes. "Defining Analog Voice". Cisco Systems Inc. 2001. Hentet fra http://www.cisco.com/warp/public/788/signalling/define_analog_voice.html, 26/3-02.
- [Cis01c] Cisco Solutions Guide. "Introduction to Packet Voice Networking in the Wide Area Network". Cisco Systems Inc. 2001. Hentet fra http://www.cisco.com/warp/public/cc/so/cuso/epso/packv_in.htm. 26/3-2002
- [Cis02] Cisco White Paper: "Understanding Delay in Packet Voice Networks". Ikke trykket materiale, hentet fra www.cisco.com/warp/public/788/voip/delay-details.html 26/3-02.
- [Cox97] Richard V. Cox. "Three New Speech Coders from the ITU Cover a Range of Applications". IEEE Communications Magazine side 40-47, september 1997.
- [DaF99] Ismail Dalgic og Hanlin Fang. "Comparison of H.323 and SIP for IP Telephony Signaling". Part of the SPIE Conference on Multimedia Systems and Applications II, Boston, Massachusetts, September 1999. SPIE Vol. 3845, pp 106-122..
- [DaP00] Jonathan Davidson, James Peters. "Voice over IP Fundamentals", Cisco Press, 2000. ISBN: 1-57870-168-6.
- [deC99] Linden deCarmo."Internet Telephony Protocols: H.323 versus SIP". Dr.Dobb's Journal, pp 30-39, July 1999.
- [Dim93] S. Dimolitsas. "Subjective assessment methods for the measurement of digital speech quality". In B. Atal, V. Cuperman and A. Gersho (Eds.), Speech and Audio Coding for Wireless and Network Applications, 43-54.Kluwer Academic Publishers, Boston. 1993.
- [ETSI02] European Telecommunications Standards Institute (ETSI). "2nd ETSI TIPHON VoIP Speech Quality Test Event: Test Specification Version 2.0.1" http://www.etsi.org/frameset/home.htm?plugtests/04History/2002_voipsqa.htm, lastet ned 27/7-02.
- [EyS00] Tony Eyers and Henning Schulzrinne, "[Predicting Internet Telephony Call Setup Delay](#)," in *Proceedings of the 1st IP-Telephony Workshop (IPtel 2000)*, (Berlin, Germany), Apr. 2000.
- [FAA01] Federal Aviation Administration (FAA). "Aeronautical Information Manual", Chapter 4, Section 2: "Radio Communications Phraseology and Techniques". Ikke trykket materiale, se URL: <http://www2.faa.gov/ATPubs/AIM/Chap4/aim0402.html>
Informasjon hentet 15/10-01.
- [Fin02] Victoria Fineberg. "A practical Architecture for Implementing End-to-End QoS in an IP Network", side 122-130 IEEE Communications Magazine January 2002.
- [GiK01] Hans W. Gierlich og Frank Kettler. "Conversational Speech Quality-The Dominating Parameters in VoIP Systems", Internet Telephony Workshop 2001, (New York), April 2001.

- [GNU96] Sandra Loosemore with Richard M. Stallman, Roland McGrath, Andrew Oram, and Ulrich Drepper. "The GNU C Library", kapittel "Sockets". Edition 0.07 DRAFT last updated 4 Oct 1996 for version 2.00 Beta Copyright (C) 1993, '94, '95, '96, '97 Free Software Foundation, Inc.
Hentet fra <http://www.gnu.org/software/libc/libc.html>.
- [GüG01] Gürkan Gür og Özgür Güler. "VoIP With An Emphasis on QoS". EE540-Computer Networks Term Report, Electrical/Electronics Eng. Dept, Bogazici Univeristy- Istanbul, Turkey. Finnes på internett:
http://www.busim.ee.boun.edu.tr/~sayli/ece260/EE540_files/VoIP_With_An_Emphasis_On_QoS/540reportgur_gulersubmitted.doc
28.desember 2001.
- [Hal01] Brian Hall. "Beej's Guide to Network Programming: Using Internet Sockets". Revision Version 2.3.1 October 8, 2001. Hentet fra <http://www.ecst.csuchico.edu/~beej/guide/net/>.
- [HHM99] Olof Hagasand, Kjell Hanson, og Ian Marsh. "Measuring Internet Telephony Quality: Where are we today?". Conference Record/IEEE Global Telecommunications Conference, Volume 3, 1999, pages 1838-1842.
- [HHSW95] Vicky Hardman, Mark Handley, Martina Angela Sasse, Anna Watson. "Reliable Audio for Use over the Internet", Proc. INET'95, Hawaii, 1995.
- [HNA00] M.Hassan, A. Nayandoro og M. Atiquzzaman. "Internet Telephony: Services, Technical challenges, and Products". IEEE Communications Magazine, April 2000.
- [ICAO95] International Civil Aviation Organization (ICAO). "Annex 10, Volume 2 - Aeronautical Telecommunications", paragraph 5.2.1.1.2, page 39. 1995.
- [IEEE69] "IEEE Recommended Practice for Speech Quality Measurements", IEEE Transactions on Audio and Electroacoustics, Vol AU-17(3), 1969, 225-246.
- [IHM⁺99] Mitsunori Iбата, Masako Hida, Yukio Mitani, Masashi Kugou og Masayoshi Hiraguchi. "An Architecture and Design of IP Telephony Gateway", NEC research & development, Volum 40, Hefte 1, side 108-113, 1999.
- [ITS94] ITS Audio Quality Research Program. "Listening Experiments", 1994.
<http://www.its.bldrdoc.gov/home/programs/audio/subj.htm>
- [ITU92] ITU "Handbook on Telephonometry", Geneva, 1992.
- [ITU96] ITU-T Recommendation P.800 "Methods for subjective determination of transmission Quality", 08/1996.
- [ITU96b] ITU-T. "ITU-T Recommendation G.723.1. Dual Rate Speech Coder for multimedia Communications transmitting at 5.3 and 6.3 kbit/s". G.723.1 (03/96).
- [ITU96c] ITU-T. "ITU-T Recommendation G.729 – Annex A. Coding of speech at 8 kbit/s using conjugate structure algebraic-code-excited linear-prediction (CS-ACELP). Annex A: Reduced complexity 8 kbit/s CS-ACELP speech codec." G.729 Annex A (11/96).

- [JiS99] Wenyu Jiang og Henning Schulzrinne. "QoS Measurement of Internet Real- Time Multimedia Services". Technical Report CUCS-015-99, Department of Computer Science, Columbia University, December 1999.
- [Kar99] Asim Karim, Ohio State University. "H.323 and associated Protocols", 1999. <http://www.cis.ohio-state.edu/~jain/cis788-99/h323/index.html>
- [KBS98] Thomas J. Kostas, Michael S. Borella, Ikhtlaq Sidhu, Guido M. Schuster, Jacek Grabiec, and Jerry Mahler. (3COM). "Real-Time Voice Over Packet-Switched Networks". IEEE Network, January/February 1998, vol 12, no. 1, pp 18-27.
- [KDK99] Hendrik Knoche, Hermann G. De Meer, David Kirsch. "Utility Curves: Mean Opinion Scores Considered Biased". 1999 7th IEEE/IFIP International Workshop on Quality of Service (IWQoS'99). London, U.K., June, 1999.
- [KIP95] W.B Kleijn og K.K Paliwal. "An Introduction to Speech Coding" kapittel 1 i "Speech Coding and Synthesis", Elsevier Science B.V, W.B Kleijn og K.K. Paliwal (editors), Amsterdam, 1995.
- [LiM00] Hong Liu og Petros Mouchtaris. "Voice over IP Signaling: H.323 and beyond". IEEE Communications Magazine, pp 142-148, October 2000.
- [LSBH97] James LaDue, Randy L. Sollenberger, Bill Belanger, Annemarie Heinze. "Human Factors Evaluation of Vocoders for Air Traffic Control Environments Phase I: Field Evaluation", U.S Department of Transportation, Federal Aviation Administration. Rapportnummer: DOT/FAA/CT-TN97/11. September 1997.
- [MaL97] N.F.Maxemchuk og S.Lo. "Measurement and Interpretation of Voice Traffic on the Internet". Conference Record of the International Conference on Communications (ICC), Montreal, Canada. June 1997.
- [McK00] Craig McTaggart, Tim Kelly. "Background Issues Paper", ITU IP Telephony Workshop 14-16 June 2000. Document: IPTEL/03, 29 May 2000.
- [MGNM90] K. Matrouf, J.L. Gauvain, F. Neel, og J. Mariani. "An oral task-oriented dialogue for air- traffic controller training". Proceedings of the SPIE (The International Society for Optical Engineering), Vol. 1293, pt. 2, p. 826- 837, 1990.
- [Micro02a] MicroWarehouse ([URL:www2.warehouse.com](http://www2.warehouse.com)). Tekniske spesifikasjoner for Quicknet Technology Inc. sitt IP-telefonkort "Internet PhoneCARD":
[URL:http://www2.warehouse.com/product.asp?pf_id=DMD4720&prod=overview&home=pc&cat=pc&blind=&view=](http://www2.warehouse.com/product.asp?pf_id=DMD4720&prod=overview&home=pc&cat=pc&blind=&view=)
 Spesifikasjonene er gjengitt i Appendiks F og hentet ned fra hjemmesiden 12/8-2002.

- [Micro02b] MicroWarehouse (<http://www2.warehouse.com>). Tekniske spesifikasjoner for Quicknet Technology Inc. sitt IP-telefonkort "Internet PhoneJACK":
<http://www2.warehouse.com/product.asp?pf%5Fid=EX10648&cat=pc&blind=>
Spesifikasjonene er gjengitt i Appendiks F og hentet ned fra hjemmesiden 12/8-2002.
- [MMG98] Jed Martens, Anthony McGregor og Ian Graham. "Voice over IP: The impact of RSVP". Part of the SPIE Conference on Routing in the Internet: New Concepts. Boston, Massachusetts, November 1998. SPIE vol. 3529.
- [MMTG00] Stan Moyer, Dave Maples, Simon Tsang og Abrahajit Ghosh. "Service Protability of Networked Appliances", IEEE Communications Magazine, January 2000.
- [MoP01] Ibrahim Mortada og Wilfried Probst. "Internet telephony signaling". Telematics and Informatics, pp 159- 194, 18 (2001)
- [NFR01] Norsk Forskningsråd (NFR)-søknad 2001 (konfidensiell) ved Navia Aviation (nå: ParkAir Systems). Skrevet av Martin Hasselknippe, Bjørn Wasstøl, Roger Lee, Per Løvstad og Geir Horn.
- [NoT98] Norsk Telemuseum. "Telehistoriske glimt -Telekommunikasjon i fortid og nåtid", ISBN:82-91335-17-6, 1998.
- [Pac01] Pacektizer. "H.323 versus SIP: A Comparison".
http://www.packetizer.com/iptel/h323_vs_sip. Printet 7/11-01.
- [Pat01] Kostas Patouras,"Air Traffic Control, Chapter 1.9 Speaking Technique". Ikke trykket materiale, se URL: <http://fly.to/kpatouras.gr>. Informasjon hentet 15/10-01.
- [Pat01b] Kostas Patouras,"Air Traffic Control, Chapter 1.4 Radio Check". Ikke trykket materiale, se URL: <http://fly.to/kpatouras.gr>. Informasjon hentet 15/10-01.
- [PCM00] Chinmay Padhye, K. Christensen, W. Moreno. "A New Adaptive FEC Loss Algorithm for Voice over IP Applications". In Proceedings of IEEE International Performance, Computing and Communication Conference, Feb. 2000. URL: <http://citeseer.nj.nec.com/padhye00new.html>.
- [Per99] Alan Percy, Brooktrout Technology Inc. "Understanding Latency in IP Telephony", 1999. <http://www.brooktrout.com/whitepapers/>
- [Pett02] Per Arvid Pettersen, Norsk Luftfartsmuseum. Personlig brev, se appendiks E, 12/7-2002.
- [PHH98] Colin Perkins, Orion Hodson, og Vicky Hardman. "A Survey of Packet Loss Recovery Techniques for Streaming Audio", IEEE Network, September/October 1998.
- [Pra00] Stefan Pracht. "Troubleshooting H.323 Signaling." White Paper, Agilent Technologies. October, 2000.
URL: <http://onenetworks.comms.agilent.com/WhitePapers.asp#IP>.
- [Pro01] Protocols.com, Protocol Directory: "H.323". Ikke trykket materiale, hentet fra URL: <http://www.protocols.com/pbook/h323.htm> 7/11-01.

- [Pro02] Christopher Angelo Provenzano. "Pthread: Basic Routines". Hentet fra http://www.planetoid.org/technical/pthreads/pthreads_man.html#BASIC. 8/2-2002.
- [PvT95] J.E. Preminger & D.J. Van Tasell, "Quantifying the relationship between speech quality and speech intelligibility", *Journal of Speech and Hearing Research*, 38, 1995, 714-725.
- [ReR01] R.J.B. Reynolds og A.W.Rix. "Quality VoIP – an engineering challenge", *BT Technol J* Vol 19, No 2 April 2001. Finnes også online: <http://www.psytechnics.com/papers>.
- [RiH00] Antony W. Rix and Michael P. Hollier." The Perceptual Analysis Measurement System For Robust End-to-End Speech Quality Assessment". Presentert på IEEE ICASSP, 5–9 June 2000. Finnes på Internett: <http://www.psytechnics.com/papers/2000-P01.pdf>.
- [Rik98] Tore Riksaasen. "Telematikknett", Universitetsforlaget, 2.opplag, 1998. ISBN: 82-00-41489-2.
- [RoC95] Dennis Roddy og John Coolen. "Electronic Communications", 4th Edition, Prentice Hall, 1995. Kapittel 11: "Pulse Modulation", side 383-412. ISBN: 0-13-312083-X.
- [Ros88] J.R Rosenberger. "Quality assessment methods for speech coding", *Telecommunications Journal*, Volum 55, Heftenummer 12, side 820- 825, 1988.
- [Sch99] Rune Schumann. "SIP/H.323 Gateway Løsning basert på abstraksjon", Hovedfagsoppgave, Universitetet i Oslo, Institutt for Informatikk, 1.november 1999.
- [Schul99] Henning Schulzrinne. "Converging on Internet Telephony Service for Telecom Version 2 - Guest Editor's Introduction", [IEEE Internet Computing](http://www.ieee.org/publications_standards/publications/free_content/contributors/contributors.cfm) 3(3): 40-43 ,1999.
- [ScR98a] Henning Schulzrinne og Jonathan Rosenberg. "Signaling for Internet Telephony". Technical Report CUCS-005-98, Feb. 1998, Columbia University, New York, New York.
- [ScR98b] Henning Schulzrinne og Jonathan Rosenberg. "The Session Initiation Protocol: Providing Advanced Telephony Services Across the Internet", *btj*, vol 3, no. 4, pp 144-160, October- December 1998. http://www.cs.columbia.edu/~hgs/papers/Schu9810_Session.ps
- [ScR99c] Henning Schulzrinne og Jonathan Rosenberg. "A comparison of SIP and H.323 for Internet Telephony" in *nossdav* (Cambridge, England), pp 83- 86, Jul. 1998.
- [ScR99a] Henning Schulzrinne og Jonathan Rosenberg. "Internet Telephony: Architecture and Protocols- an IETF perspective". *cnis*, pp 237-255, vol 31, no 3, Feb. 1999.
- [ScR99b] Henning Schulzrinne og Jonathan Rosenberg. "Tutorial: The IETF Internet Telephony Architecture and Protocols", *IC Online*, <http://computer.org/internet/telephony/w3schrosen2.htm>, 1999.

- [ScR99c] Henning Schulzrinne og Jonathan Rosenberg. "The IETF Internet Telephony Architecture and Protocols". IEEE Network, May/June 1999.
- [ScR00] Henning Schulzrinne og Jonathan Rosenberg. "The Session Initiation Protocol: Internet-Centric Signaling". IEEE Communications Magazine, October, 2000.
- [SDCH97] Randy L. Sollenberg, James La Due, Brian Carver og Annmarie Heinze. "Human Factors Evaluation of Vocoders for Air Traffic Control (ATC) Environments Phase II: ATC Simulation". ATC Simulation (DOT/FAA/CT-TN97/25). Atlantic City International Airport: Federal Aviation Administration William J. Hughes Technical Center, 1997.
http://www.airtrafficcafe.com/training_info.shtml
- [SDP98] M. Handley og V. Jacobson. "SDP: Session Description Protocol". Request for Comments (Proposed Standard) 2327, Internet Engineering Task Force. April 1998.
Se URL: <http://www.ietf.org/rfc/rfc2327.txt?number=2327>
- [Shil91] F.L.Shilling. "Safety in civil aviation through telecommunications". Telecommunication Journal, Vol. 58, VIII, 1991
- [SIP99] M. Handley, H. Schulzrinne og J. Rosenberg. " SIP: Session Initiation Protocol". Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force. March 1999. Se URL: <ftp://ftp.isi.edu/in-notes/rfc2543.txt>.
- [Sock02] Ukjent forfatter. "Sockets Tutorial". Hentet fra <http://www.cs.rpi.edu/courses/sysprog/sockets/sock.html> 10/1-2002.
- [Spa01] A.S. Spanias. "Speech Coding: A tutorial Review", Proceedings of the IEEE Volume 82, No. 10, page 1541-1582, October 1994.
<http://www.eas.asu.edu/~spanias/papers/spanias94tutorial.pdf>.
- [Sør00] Steinar Sørli. "Integrasjon av multimediatrminaler i telefonnett", Hovedfagsoppgave Universitetet i Oslo, Institutt for Informatikk, 1. august 2000.
- [Tan96] Andrew S. Tanenbaum. "Computer Networks", Third Edition, Prentice Hall, 1996. ISBN: 0-13-349945-6. Kapittel 7.7.5, side 756-760.
- [ToJ99] Thomas Tømter og Rune Jordet. "Voice over IP", Oppgavenummer T08-99, 1999. Hovedprosjekt ved Høgskolen i Oslo, Avd. for Ingeniørutdanning, Seksjon for elektrofag.
- [ToL00] Henri Tobiet og Pascal Lorenz. "Performance Measurement Methodologies and Quality of Service Evaluation in VoIP and Desktop Videoconferencing Networks". Mobile and Wireless Communications Networks (MWCN) 2000, Lecture Notes in Computer Science (LNCS) 1818, pp 92- 107, 2000. Springer Verlag Berlin- Heidelberg.
- [Tri199] Trillium. "H.323", Web ProForum Tutorials, <http://www.iec.org>, 1999.
- [VKBG95] Andreas Vogel, Brigitte Kerhervé, Gregor von Bochmann og Jan Gecsei. "Distributed Multimedia and QoS: A Survey". IEEE Multimedia, pp 10- 19, 1995.

- [Voi83] William D. Voiers. "Evaluating Processed Speech using the Diagnostic Rhyme Test". "Speech Technology", volum 1, heftenummer 4, side 30-39, 1983. ISBN/ISSN: 0744-1355
- [Vor94] Stephan Voran. "Techniques for Comparing Objective and Subjective Speech Quality Tests". Proceedings of the Speech Quality Assessment Workshop, November 10- 15, 1994, Bochum, Germany.
www.its.bldrdoc.gov/home/programs/audio/bochumabs.htm
- [Vor98] Stephen Voran. "A Subjective Conversation Test to Evaluate Two Telecommunications Systems", U.S Department of Commerce's National Telecommunications and Information Administration, Institute for Telecommunication Sciences, System Performance Standards Group. 325 Broadway, Boulder, Colorado 80303. September 28, 1998. (På oppdrag for: Netrix Corporation, Voice Technology Center, 2402-A Clover Basin Road, Longmont, Colorado 80503)
- [Wal00] Kurt Wall et al. "Linux Programming Unleashed (from knowledge to master)", SAMS Publishing, 2000. ISBN: 0-672-31688-9.
- [WaS96] Anna Watson og Martina Angela Sasse. "Evaluating Audio and Video Quality in Low-Cost Multimedia Conferencing Systems" in Interacting with Computers, vol 8, pp 255- 275, 1996.
- [WaS97] Anna Watson og Martina Angela Sasse. "Multimedia Conferencing via Multicast: Determining the Quality of Service Required by the End User". Proc. International Workshop on Audio-Visual Services over Packet Networks (AVSPN), 15- 16 September 1997, Aberdeen.
<http://citeseer.nj.nec.com/357437.html>
- [WaS98] A. Watson og M.A. Sasse. "Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications" in Proceedings of the ACM Multimedia Conference, pp. 55- 60, Bristol, UK, September 1998.
<http://citeseer.nj.nec.com/watson98measuring.html>
- [WEL01] Westbay Engineerings Limited: "Bandwidth Requirements for Voice over IP transmission". Copyright 2001. Ikke trykket materiale, hentet fra <http://www.erlang.com/bandwidth.html> 12/3-02.