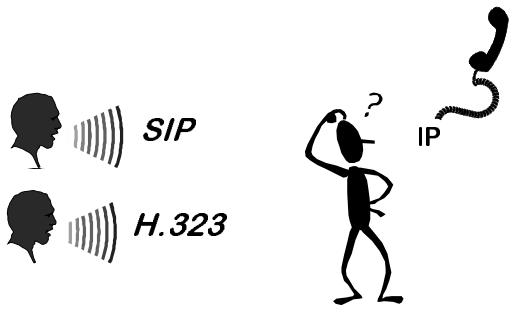
Sammenligning av SIP og H.323

Norwegian Computing Center
APPLIED RESEARCH AND DEVELOPMENT
Norwegian Computing Center / Applied Research and Development

NOTAT/NOTE



IMEDIA/10/98

Eirik Maus
Peter D. Holmes

Oslo
October 1998

IMiS Kernel

Norwegian Computing Center
APPLIED RESEARCH AND DEVELOPMENT

NR-notat/NR Note

Tittel/Title:
Sammenligning av SIP og H.323

Dato/Date: October
År/Year: 1998
Notat nr: IMEDIA/10/98
Note no:

Forfatter/Authors:
Eirik Maus, Peter D. Holmes

Sammendrag/Abstract:

To standarder, SIP og H.323, har dukket opp for å ”ringe mellom PC’er”.

SIP er internett-utviklernes forsøk på å bringe ringe-signalering og format-forhandling inn i internett-verden. H.323 tar televerkenes arbeid med avansert video- og datakommunikasjon over telenettverk inn i data-nettverk-verden. Dette kan delvis ses som en konsekvens av at mange allerede har skaffet seg dyrt og avansert utstyr for multimedia-kommunikasjon: datamaskin med lydkort.

Mens SIP er en protokoll til å signalere til internett-maskiner, er H.323 tenk som en måte å signalere mellom multimedie-terminaler som *for eksempel* multimedie-datamaskiner. H.323 er et forsøk på å bygge bro fra televerkenes eksisterende signalprotokoller osv. til datamaskiner. SIP er et forsøk på å bygge bro den andre veien: bygge audio-, video- og oppringningsfunksjonalitet inn i multimedie-PC’en.

Denne rapporten beskriver funksjonaliteten i de to standardene, og ser på likheter og forskjeller. I tillegg vurderes muligheter for interoperasjon mellom disse to standardene.

Emneord/Keywords: SIP, H.323, IP-telefoni, Voice on the net, konferansesystemer

Tilgjengelighet/Availability: Open

Prosjektnr./Project no.: 28006, IMiS Kernel

Satsningsfelt/Research field: Konferansesystemer

Antall sider/No of pages: 47

Norsk Regnesentral / Norwegian Computing Center
Gaustadalléen 23, Postboks 114 Blindern, 0314 Oslo, Norway
Telefon 22 85 25 00, telefax 22 69 76 60, telex 76518 ncc n

Sammenligning av SIP og H.323

Eirik Maus
Peter D. Holmes

Norsk Regnesentral
October 1998

Innhold

I. Innledning: To teknologier for ett problem

PROBLEMET: INTEGRERT KOMMUNIKASJONSLØSNING

SIP og H.323 er to ulike teknologier for å bygge bro over samme gap. De siste par årene har vi sett en eksplosiv vekst i bruken av internett som kommunikasjonskanal. En viktig faktor til dette er den allmenne utbredelsen av datamaskiner, særlig i arbeidslivet. Praktisk talt all forretningsmessig informasjon, brev, brosjyrer, pristabeller osv, lages på datamaskin før de sendes kunder og partnere. Internett, særlig e-post, har vist seg å være en særlig effektiv måte å dele slik informasjon på. Viktige faktorer er at informasjon kan utveksles raskt og digitalt mellom partenes datamaskiner, og at prisen for bruk av internett er svært lav sammenlignet med f.eks. telefon. Hele verden kan nås til samme pris, og på nær samme tid, som lokal overføring.

Informasjonsutveksling over internett har til nå i hovedsak vært tekstbasert og asynkron. Meldinger kodes, hvilket stort sett betyr skrives, og kan (når de er ferdige) sendes/hentes av andre. Dette vil som regel si å enten sende dem som e-post eller legge det ut til offentligheten gjennom WWW, ftp eller andre lignende tjenester. Det finnes synkrone utvekslingsformer som IRC og Talk der partenes setninger kommer inn mellom hverandre som i vanlige diskusjoner/samtaler. Disse er imidlertid også tekstbasert, noe som gjør det for tungvint å bruke til at det fullt ut kan erstatte samtaler, f.eks. over telefon.

Sammen med World Wide Web har det oppstått et ønske om å utveksle flere medietyper. Lydfiler har blitt mer vanlig, og mange datamaskiner har blitt utstyrt med lydkort og kraftige grafikk-kort. Hva som har manglet for å bruke datamaskiner til interaktiv kommunikasjon er nettopp kontinuerlig utveksling av lyd og evt. bilde, akkurat som telefonen kan. Inntil de siste par årene har det vært mulig å utveksle tekst, og til en viss grad også grafikk mer eller mindre interaktivt og gratis over internett. Straks lyd kan utveksles synkront over internett er PC'en på vei til å bli en billigere telefon som også har muligheter for andre medietyper også. I dag er det to ting i hovedsak som mangler for å få til dette: Det finnes ingen standardisert, synkron metode å gjøre en annen datamaskin oppmerksom på at du ønsker å kommunisere med den (eller brukeren)[1]. Dessuten må det etableres standardiserte måter for datamaskinene å forhandle og bli enige om hvilke medier og formater som skal utveksles. SIP og H.323 er to ulike forsøk på å fylle dette behovet.

BAKGRUNN: TO UTGANGSPUNKT – TO TEKNOLOGIER

For å forstå skillet mellom H.323 og SIP må vi se på historien til de gruppene som står bak.

Internett

Internett har i veldig mange år hatt mulighet for utveksling av tekst. World Wide Web utvidet dette til grafikk og flere medier som lyd og bevegelig bilde. Utbredelsen av WorldWideWeb har bidratt til at etablere har fått seg datamaskin med multimedie-egenskaper, men endret ikke på det faktum at informasjonsutvekslingen i hovedsak er asynkron.

I universitetsmiljøer har det i mange år vært jobbet med utveksling av tidskritiske data som lyd og video over internett. Særlig har synkron utveksling av lyd og bilde, videokonferanser og direkteoverført "TV" bydd på faglige utfordringer. Man har håpet å kunne utvide den støtten internett

[1] Det finnes imidlertid proprietære metoder, som f.eks. Cool Talk, som krever at visse applikasjoner er aktive på maskinen som skal kontaktes.

gir til samarbeid uavhengig av avstand til å gjelde mer enn utveksling av filer. Konferanseprogrammer som Vic og Vat, og verktøy for delt tavle mellom konferansedeltakerne som WB, har blitt utviklet siden starten av 1990-tallet. Noen av disse er laget med fokus på samtale, eller videokonferanser, andre er laget som verktøy for lettere samarbeid på tvers av avstander.

Fire utfordringer står igjen før internett har blitt et fullgodt medium for sanntidskommunikasjon som kan erstatte telefon og kanskje TV. Først trengs systemer for å garantere overføring av de store datamengdene med en gitt kvalitet og uten forsinkelser. Dernest trengs systemer for å kunne ta betalt for bruken, slik at utbygging av nettkapasitet og garanti/reservasjonssystemer kan lønne seg. Siden synkron kommunikasjon forutsetter at begge parter er tilstede samtidig, trengs videre et system for å få tak i en part som ikke vet at du vil snakke med ham, omtrent som ringelyden på telefon. Til sist trengs et system for at to dataterminaler kan bli enige om hvilke av de mange eksisterende medieformater som skal utveksles mellom dem i en samtale/konferanse. SIP er laget av forskerne bak andre viktige multimediestandarder på internett for å løse de to siste problemene. De to første problemene er det andre i internett-miljøet som jobber med.

Telekommunikasjon

I telemiljøer har det i mange år vært forsket på måter utvide telefontjenesten til å dekke flere medier. Særlig har det vært lagt mye arbeid i videotelefoner og videokonferanser, og et par standarder er etablert slik at du kan kjøpe utstyr i butikken.

Felles for disse standardene er at de trenger høyere båndbredde enn det som kan tilbys på vanlige telefonlinjer. En standard for telefonnettverk, ISDN, har blitt etablert, men denne tilbyr ikke spesielt mye større kapasitet. En vil fortsatt trenge flere linjer i parallell for å overføre videotelefoni av brukbar kvalitet. Videre er terminalene ganske avanserte, og dermed ganske dyre. Internett er bygget på nettverksteknologier for datamaskiner. Disse gir stort sett god nok båndbredde til å overføre multimediedata. Videre har veldig mange arbeidsplasser nå dyre, kraftige generelle datamaskiner, som blant annet kan gjengi lyd og bevegelig bilde, samt internettforbindelse med kapasitet til å overføre disse dataene.

To utgangspunkt, to standarder

Avansert teknologi og høy pris er ganske nytt for teleutstyr ment for pulten. Tradisjonelt har telekommunikasjon jobbet siden starten/midten av århundret med å etablere allmenn kommunikasjon på tvers av avstander. En viktig faktor for å tilby dette til alle har vært enkelt og billig brukerutstyr. Som en følge av dette målet har kompleksiteten har blitt liggende i nettverksinfrastrukturen, i store dyre sentraler. Disse har på sin side kunnet utbygges av større foretak, oftest med nasjonal eller offentlig forankring.

Internett ble etablert med et helt motsatt utgangspunkt, nemlig å etablere kommunikasjon mellom allerede eksisterende avanserte datamaskiner. Målet var å finne en sikker måte for disse å kommunisere, uten svake punkter som lett kan rammes i krig. Fleksibilitet, stabilitet og usårbarhet som mål har gjort det naturlig med enkel infrastruktur sentralt i internett. Mange av de vanskeligere oppgavene kan løses av datamaskinene som er brukerutstyret.

At to standarder, SIP[7] og H.323[1] har dukket opp for å ”ringe mellom PC’er” må forstås på denne historiske bakgrunnen. SIP er internett-utviklernes forsøk på å bringe ringe-signalering og format-forhandling inn i internett-verden. H.323 tar televerkenes arbeid med avansert video- og datakommunikasjon over telenettverk inn i data-nettverk-verden. Dette kan delvis ses som en konsekvens av at mange allerede har skaffet seg dyrt og avansert utstyr for multimedia-kommunikasjon: datamaskin med lydkort. Mens SIP er en protokoll til å signalere til internett-maskiner, er H.323 tenk som en måte å signalere mellom multimedieterminaler som *for eksempel* multimedie-datamaskiner. H.323 er et forsøk på å bygge bro fra televerkenes eksisterende signalprotokoller osv. til datamaskiner. SIP er et forsøk på å bygge bro den andre veien: bygge audio-, video- og oppringningsfunksjonalitet inn i multimedie-PC’en.

II. Teknologi-oversikt

H.323

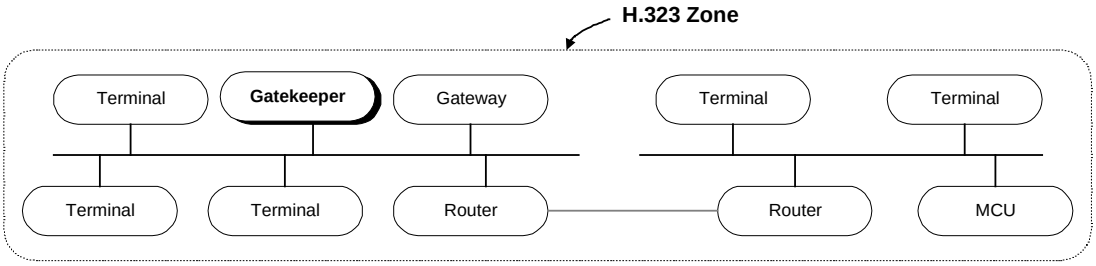
Grunnleggende og bakgrunn

H.323 er en videreføring av den internasjonale teleunionens (ITUs) arbeid med audiovisuell kommunikasjon (bildetelefon). H.323 bygger på tidligere standarder for koding av lyd, bilde og telesignaler på data- og telenett. I tillegg til lyd og bilde er T.120-serien, ITUs standard for kommunikasjon mellom distribuerte dataapplikasjoner som delt whiteboard, anvist en plass i systemet.

Opprinnelig var standardarden laget for kommunikasjon over pakkebaserte lokalnett uten QoS-garantier (les vanlig Ethernet). I seinere versjoner har dette blitt byttet ut med pakkebaserte nettverk generelt, for å kunne passe med internett. Dette har introdusert et par problemer i forhold til adresser ettersom man ikke kan vite om alle adressene på hele internett, og kringkasting til alle for å spørre om adresse er mindre aktuelt.

Bestanddeler og ansvar

H.323 definerer en rekke enheter som opprinnelig var tiltenkt en plass på et lokalnett. Det stilles ingen krav til enheter sentralt i nettverket e.l. Dette er de forskjellige enhetene i et H.323-system.



Figur 1 : Entiteter i H.323 (fra [9])

END SYSTEM / TERMINAL:

- Ringbar enhet med adresse, f.eks telefonapparat eller en multimedia-PC.

GATEWAY

- En gateway er en boks som har kobling mot både datanettet og et telefon- eller annen type nett. Den siste koblingen er ikke vist på bildet.
- Fungerer som oversetter/port mellom pakkebasert nett og for eksempel ISDN/GSTN (vanlig) telefonnett. Må oversette både signalering (signaler for å koble opp etc.) og lyd/bilde-signaler.
- Opererer som End System på begge nettene den bygger bro mellom. Et kall på det ene nettet vil terminere her, og resultere i at gatewayen initierer et kall på det neste nettet fram til endesystemet der.

GATEKEEPER

- Passer på at ingen bruker ressurser uten å få lov først.
- Må:
  - Drive admission control i zonen.
  - Adresseoversette mellom interne/eksterne adresser.
  - Svare på forespørsler om båndbredde, og evt. ta hånd om QoS-reservasjoner.
  - Generell kontroll i Zonen.
- Kan også
  - Ha ansvar for Call Control Signalling
  - Autorisasjonskontroll for innkommende/utgående samtaler.

ZONE

- Lokalnettsegment e.l. som kontrolleres av EN gatekeeper. Hvis det er flere gatekeepere på et nettsegment vil de endesystemene som tilhører den ene gatekeeperen utgjøre en logisk Zone, og de andre en annen.
- Må støtte multicast for at endesystemene skal kunne finne gatekeeper.

MC : MULTIPOINT CONTROLLER

- Systemmodul (gjerne programvare) for å koordinere konferanser med flere parter.
- Kan bygges inn i alle endesystemer, gatekeepere, MCU'er o.l. som kan være i bruk ved konferanser (samtaler med mer enn 2 parter).

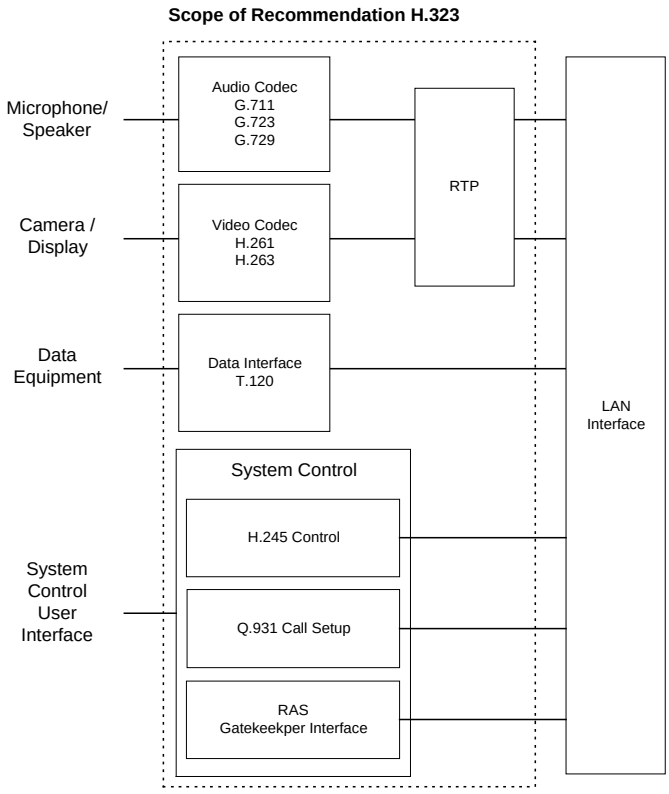
MCU : MULTIPOINT CONTROLLER UNIT

- Kontrollerer konferanser med flere deltakere.
- Inneholder minst én Multipoint Controller.
- Kan inneholde en eller flere Multipoint Processors for å kombinere og spre data fra flere kilder til en, f.eks. mikse lydstrømmene fra de enkelte deltakerne i en konferanse til en lydkilde, og spre denne til alle parter som én lydstrøm (a la RTP multiplexer).

Flere av disse enhetene kan være plasser i samme fysiske boks.

Signalering / kommunikasjon

H.323 er i hovedsak en koordinerende standard som bygger på andre standarder og henviser deres bruksmåte. Figur 2 viser de forskjellige standardene som inngår i kommunikasjonen i H.323. Selve H.323-standarden forteller om hvilke andre standarder som skal brukes hvor. Selve H.323-standarddokumentet bruker mesteparten av plassen på å beskrive fremgangsmåter for å signalere ulik kontrollinformasjon, det vil si signalering innenfor det som er merket "System Control" i figur 2.



Figur 2 : Protokoller og signaler i H.323 (fra [9])

Standardene som inngår i H.323 er:

- H.323 er en del av familien H.32X for videokonferanser over forskjellige typer tele- og datanett.
- G.7* er standarder for koding og komprimering av lyd.
- H.26* er standarder for videokomprimering.
- RTP er IETF/Internet Real Time Protocol (RFC 1889), en standard for å overvåke kvaliteten på lyd/bilde-trafikk på internett. Sender via UDP eller tilsvarende.
- T.120 er en mengde standarder (T.120 – T.129) for distribuert, samtidig datatilgang (”delte applikasjoner”) som definerer synkronisering, kontrollhierarkier m.m.
- RAS er Registration, Admission and Status, et sett av kommunikasjonsprimitiver som utveksles mellom endesystemer og nærmeste gatekeeper for å finne gatekeeper, spørre om tillatelse til å bruke nettet o.l. Bruker UDP eller tilsvarende upålitelig kanal.
- Q.931 er et sett primitiver for å sette opp en samtale/sesjon. Dette innebærer å signalere at man prøver å få tak i motparten, si fra at ”nå ringer det”, ”nå er røret lagt på” osv. Bruker en pålitelig kanal /TCP.
- H.245 er en overordnet kontrollkanal som brukes underveis i sesjonen, bl.a. for å opprette logiske signaleringskanaler, utveksling av ferdigheter (kapabilitet) til å sende og motta ulike medier og kodingsformat, justere båndbredde m.m. Bruker en pålitelig kontrollkanal.
- LAN interface er forutsatt å være i samsvar med standard H.225 om hvordan signalene skal kodes på pakkebaserte nett.

Signalering består i å utveksle pakker kodet etter spesifikasjonene over, pakket i LAN-pakker etter H.225. Kontrollpakkene er beskrevet i ASN.1-syntaks i de ulike standardene som definerer meldingene (i hovedsak H.245). Protokollen er m.a.o. bit-orientert.

H.245 spesifiserer en mengde forskjellige meldinger: 43 forskjellige H.245-signaler er Mandatory (på minst en side av de aktuelle endesystemene), 12 er Forbidden og 56 er Optional på begge sider. Langt færre RAS og Q.931-meldinger er i bruk.

Registrering av terminal i Zonen

Mekanismene for å finne personen eller terminalen det ringes til i H.323 bygger på at alle terminaler må registrere seg hos Gatekeeper. Dette vil typisk skje når de slås på eller når en person logger inn. Straks programvaren for å initiere eller ta imot H.323-samtaler startes, vil den sende en Registration Request (RRQ) til sin Gatekeeper på Gatekeepers RAS-kanal. Denne kan inneholde et email-type navn på brukeren av terminalen, slik at oppringninger kan gjøres til email-type brukernavn og videresendes til riktig terminal av Gatekeeper. Videre inneholder RRQ'en adressen på terminalen RAS-kanal og adressen på terminalens Call-Control-kanal.

Før registrering kan foregå, må terminalen finnet ut gatekeepers adresse. Den kan være fast, eller oppdages med multicast (broadcast) av Gatekeeper Request (GRQ) på "Well-known Gatekeeper Discovery"-adressen. En Gatekeeper som ser meldingen og er villig til å være Gatekeeper for terminalen svarer Gatekeeper Confirm (GCF) og oppgir sin egen adresse.

Både Gatekeeper Discovery og registreringsadressen er en RAS-adresse oppgis som en TSAP, dvs. nettverksadresse + portnummer. Dette gjør Zone-begrepet interessant. Det er mulig for en Gatekeeper å akseptere registrering fra terminaler hvor som helst på internett. En følge av dette er at en Zone blir logisk, heller enn fysisk begrenset. I stedet for å være et lokalnett med en rekke apparater tilkoblet, vil en Zone være en logisk enhet bestående av de terminalene som til enhver tid er registrert ved samme Gatekeeper. I dette lys kan det virke litt merkelig at Gatekeeper har ansvaret for resursreserveringer (båndbreddereservasjon) på vegne av alle enheter i Zonen.

En e-mail-type adresse kan bare være assosiert med én terminal. Dersom en ny terminal prøver å registrere seg med samme identitet, vil den bli avvist. Når man logger ut eller avslutter H.323-applikasjonen må derfor terminalen sende Unregister Request (URQ) til Gatekeeperen. Her er det åpenbart at det kan skje feil hvis terminaler kræsjer e.l. Det finnes mulighet for å sette at RRQ bare skal gjelde for en viss tidsperiode, og må oppfriskes før den utløper, ellers regnes terminalen (brukeren) som avregistrert.

Oppsetting av samtale

Å sette opp en samtale (eller sesjon) gjøres ved å utveksle en rekke meldinger. Noen av disse er for å be om tillatelse fra Gatekeeper. Andre er for å sette opp kontrollkanal, og å utveksle adresser og nødvendig informasjon mellom endesystemene som skal kommunisere.

En sesjon foregår i 5 faser:

A. Call Setup: Få tillatelse til å kontakte partner, åpne kontrollkanal, send ringeindikator og avgjør om han kan/vil snakke med deg.

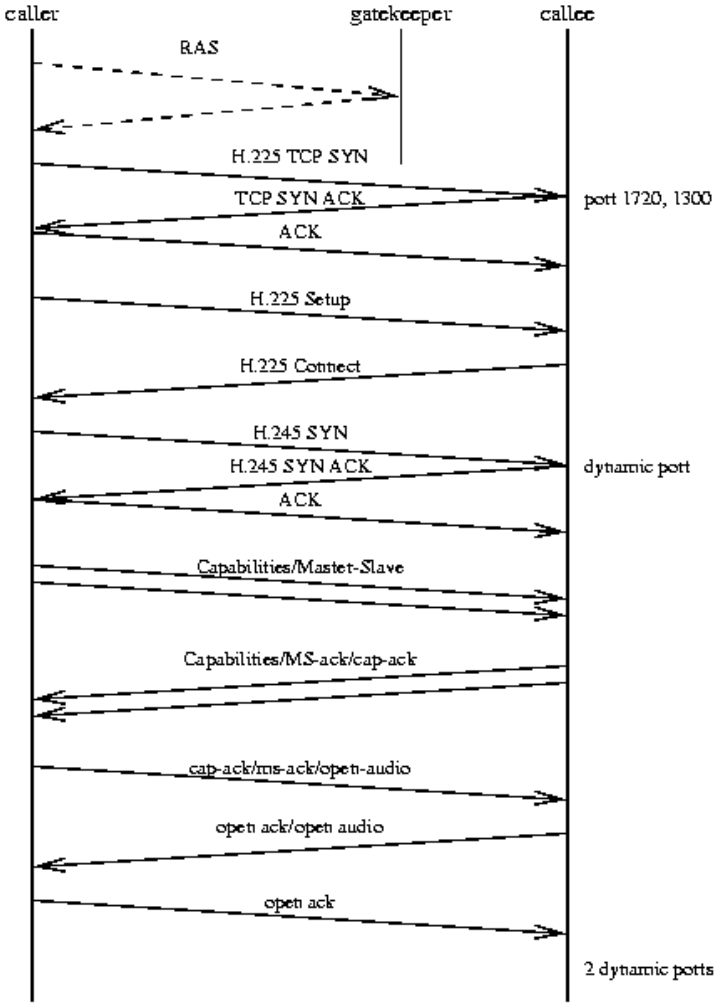
B. Capability exchange: Bli enige om hvilken medier som skal utveksles.

C. Etablering av audiovisuell kommunikasjon: åpne kanaler for mediedata og start utveksling.

D. Tjenester under sesjonen: F.eks. justere båndbredde, gå over til konferanse osv.

E. Terminering av sesjon.

Det er etter fase C samtalen foregår. De foregående er nødvendige for å få tillatelse til å kommunisere og bli enige om medier og formater som skal utveksles.



Figur 3: Oppsetting av samtale i H.323 (fra [11])

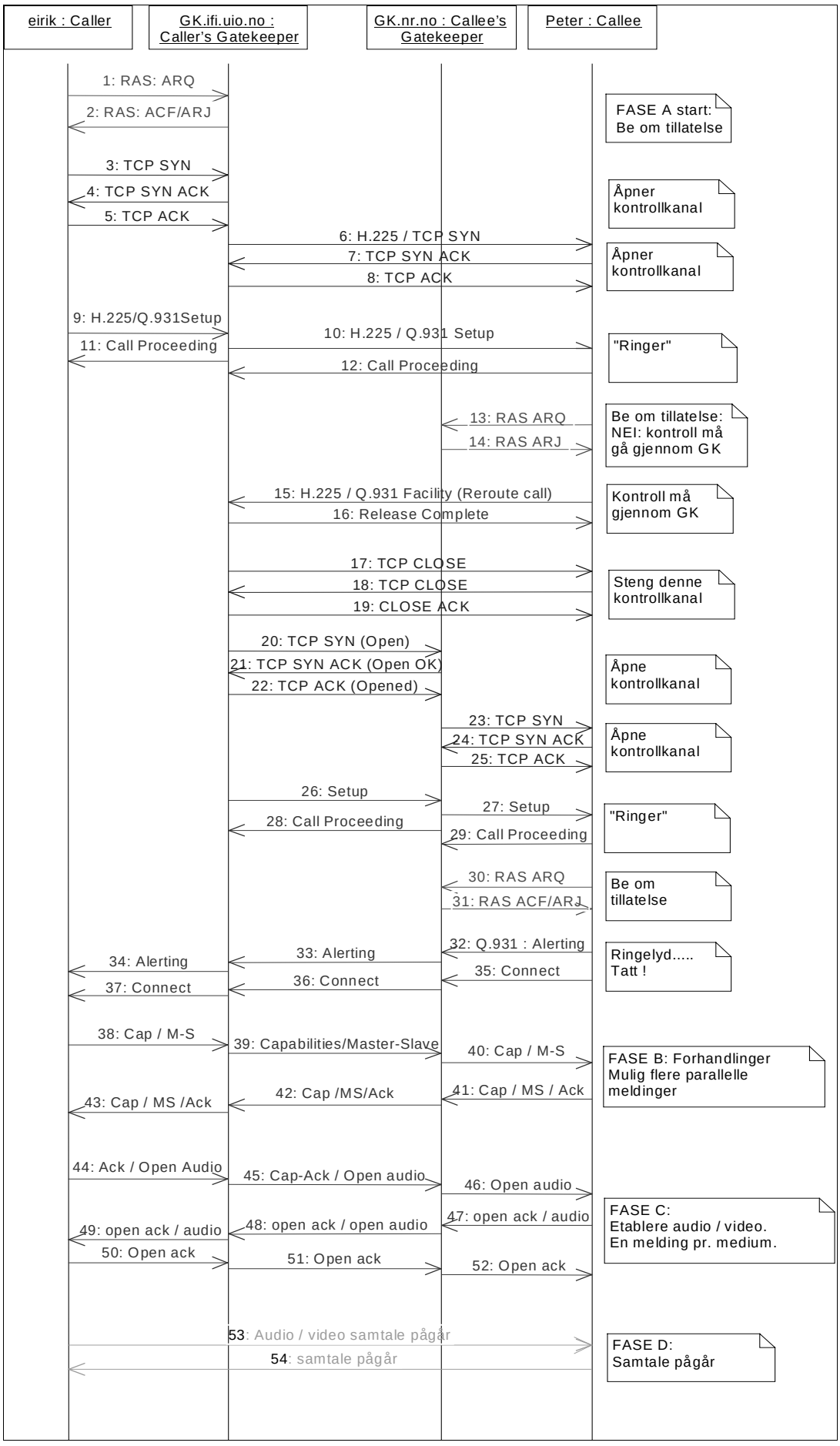
H.323 spesifiserer ulike måter å konfigurere utstyret på avhengig av hvor stor kontroll man ønsker å ha med terminalene/klientene. Gatekeeper, hvis man har dette installert, har oversikt over og ansvar for den kommunikasjonen som foregår i Zonen. Systemet kan settes opp få all kontrollsignalering fra terminal til terminal går gjennom Gatekeeper på en eller begge sider.

Figur 3 viser et oppsett av en samtale der all kontrollinformasjon går direkte mellom endepunktene. Det eneste som kontrolleres av noen gatekeeper er at den som ringer får lov å ringe ut. Bortsett fra det tilfellet der det ikke er noen gatekeeper i det hele tatt, er dette et eksempel på minimal kontroll i H.323. RAS-signaliseringen er merket i store bokstaver er RAS-meldinger for å få lov til å samtale (ut). Store bokstaver forøvrig er meldinger nødvendige for å sette opp en sikker kanal når TCP brukes for sikker overføring (hvilket det i praksis vil være over internett). Små bokstaver er kontrollmeldingene som brukes i H.323 for å sette opp samtalen og forhandle om formater o.l. (Etter Q.931 og H.245)

Figur 4 viser en strengere variant hvor begge endesystemer er registrert hos hver sin gatekeeper (registreringen antar vi er gjort) og hvor begge gatekeepere vil at kontrollinformasjonen skal gå gjennom dem. I motsetning til illustrasjonene i H.323-standard-dokumentet, er også nødvendig utveksling av pakker for å opprette pålitelige kanaler tatt med (TCP).

Peters terminal får Access Reject (ARJ) når den ber om tilgang til samtale med Eiriks terminal. Gatekeeper GK.nr.no vil at kontroll skal gå gjennom den. Da må det opprettes nye kontrollkanaler, og samtalen settes opp på nytt. Som vi ser, må vi gjennom en ikke ubetydelig sekvens av meldinger før samtalen kan komme igang. Dette er særlig fordi oppretting og stenging av TCP-kanaler bruker halvannen pakke-rundtur.

Den endelige versjon av H.323 v2 [10] har tillatt ”Fast Start” signalering for å redusere oppsettingstiden. Dette innebærer å sette data fra de fleste fasene sammen til ett ”jumbo-datagram” og sende dette over i en melding. Fast-Start vil betydelig redusere antall nødvendige pakke-rundturer før samtalen starter. En stor del av signaliseringen vil likevel gå til å opprette pålitelige kontrollkanaler (TCP).



Figur 4 : H.323-samtale gjennom to gatekeepere (etter [1] og [11])

SIP : SESSION INITIATION PROTOCOL

Grunnleggende og bakgrunn

I flere år har programvare for videokonferanser over internett eksistert. Så lenge båndbredde og kvalitet et tilstrekkelig tilstede, gjør økonomi dette langt å foretrekke framfor telefonbaserte videokonferanseløsninger. Det finnes imidlertid ikke noen protokoll for å "ringe", dvs. få tak i samtale-/konferansepartnere som ikke vet at du vil snakke med dem. SIP er laget for å fylle dette gapet.

Et av målene har vært en enkel protokoll det er lett å implementere, og som gir raskere oppkoblingstid enn konkurrenten H.323. SIP er laget så lik HTTP-protokollen for WWW som mulig, men den spør om kontakt med multimedieapplikasjoner, ikke filer på en disk. Flere avsnitt i spesifikasjonen henviser bare til hvordan problemet er løst i HTTP/1.1.

Et viktig problem ved videokonferanser er forhandlinger om hvilke medier og hvilke formater for komprimering av lyd og bilde som skal benyttes. I IETFs arbeid er det satt av plass i SIP-standarden til meldinger om partenes preferanser, men selve formatet på medie- og formatspesifikasjonene er definert i en egen standard kalt SDP (Session Description Protocol, RFC 2327). Denne er eldre enn SIP, og laget for å kunne brukes sammen med Session Announcement Protocol (SAP, ikke ferdig) for å annonsere eksistensen av multimediesesjoner det er mulig å ta imot (omtrent som TV-kanaler).

Bestanddeler og ansvar

Delene i et SIP-system er laget for å være ganske enkle klient-tjener-enheter mest mulig likt webservere og –klienter. Det må imidlertid gjøres endringer fordi det er mennesker, ikke filer, man skal ha tak i, og disse flytter seg stadig rundt.

USER AGENT SERVER

- Program som er server på vegne av en person, og som prøver å gjøre han oppmerksom på hendelsen når noen prøver å initiere en sesjon (f.eks. ved å lage ringelyd). Kjører på brukerens PC/arbeidsstasjon.

KLIENT

- Program som brukes til å initiere multimediesesjoner ved å kontakte den andre partens User Agent Server. Dette kan f.eks. bygges inn i fremtidige web-browsere for å håndtere URL'er som `sip:bruker@mitt.domene.her`, ettersom funksjonaliteten og protokollen er veldig lik HTTP.

SERVER

- Program står på fast adresse, og som benyttes for å kontakte en person som vanligvis befinner seg innenfor denne servers domene. Er enten en proxy server eller en redirect server.

PROXY SERVER

- Server som svarer på vegne av en bruker som ikke er tilstede, og som lager en ny forbindelse til brukeren der han egentlig er, uten at klienten merker forskjell.

REDIRECT SERVER

- Server som melder fra at en bruker (midlertidig) ikke er på denne adressen, og som sier fra hvor han heller kan kontaktes. Klientprogrammet kan da gjøre et nytt forsøk mot den nye adressen.

REGISTRAR

- Kan ta imot REGISTER-Requester fra en bruker som sier at han (midlertidig ?) er på en (annen) adresse, og at kontaktforsøk skal rettes dit. Typisk en Proxy eller Redirect Server eller en database i tilknytning til dette.

Det er ikke urimelig å anta at kommende SIP-servere vil være både Registrar, Server og enten Proxy eller Redirect Server i samme program, kan hende alt sammen med valgfritt Proxy/Redirect.

Signalering / kommunikasjon

SIP er bygget på utvekslinger av tekstmeldinger ved Request – Response, akkurat som HTTP. Meldingene består av en rekke linjer tekst. Tekstmeldingene utveksles primært over UDP/upålitelig kanal, men TCP/pålitelig kanal kan også brukes. Adressene finnes ved vanlig DNS. En Request har først en linje tekst som forklarer hva slags Request det er, og hvilken versjon av SIP den refererer til. Deretter følger et antall linjer med header-felter. Til sist (etter en blank linje) kan det komme en body med innhold annet enn SIP, gjerne en SDP-melding for å forhandle om dataformater. Respons-meldinger har samme format, men første linje inneholder en tallkode og tekst for å beskrive virkningen av Requesten, f.eks. ”200 OK” eller ”302 Moved temporarily”.
Strukturen på en Request blir da:

```
Request-Line
*Header-lines
<CR LF>
Body
```

Response er helt lik, men starter med Status-Line istedenfor. Noen få headere som ”From”, ”To”, ”Via”, CSeq” og ”Call-ID” må være med i alle meldinger, men ellers er det ingen regler for rekkefølge eller hvor mange headere som kan være med. Headere er slutt når det kommer en tom linje (inneholder bare linjeskift), akkurat som i e-mail og HTTP. Header-feltene kan deles i klasser på denne måten (fra draftet):

Klasse	Betyr	Eksempel
Request-Line	Metode og adresse for Requester. Dette må være første linje i en Request. Finnes ikke i Response Format : <metode> <Request-URI> <SIP-versjon>	INVITE sip:emaus@nr.no SIP/2.0
Status-Line	Svar-linje: Første linje i Response. Ikke i Request. Format: <SIP-versjon> <Kode + tekst> Koder for response-typer nedenfor. Samme som HTTP, men utvidelser for ting som ikke finnes der.	SIP/2.0 181 Call Is Being Forwarded
General Header	Tilhører selve sesjonen.	To, From, Date, Call-ID, CSeq, Via, Encryption etc.
Entity Header	Om innholdet i Body (som finnes hvis Content-Length er større enn 0). Brukes for å spesifisere innholdet, akkurat som MIME.	Content-Length Content-Type Content-Encoding

Request Header	Tilleggsinformasjon om Requesten eller Klienten. Header "Subject" finnes her. Videre "Accept" som brukes til å oppgi hva slags data som forventes tilbake (i body) (definert i HTTP/1.1-standarden). Vanlig her vil være data av type application/sdp som forhandling om hvilke medier og formater som skal utveksles.	Accept Authorization Require etc. (mange)
Response Header	Informasjon om Responsen, som tillegg til status-line, f.eks. ny adresse. Kan brukes til challenge ved negativt svar der Autentisering trengs.	WWW-Authenticate eller følgende to linjer fra svar på Invite: SIP/2.0 302 Moved Temporarily Contact sip:eirikma@ifi.uio.no

Det er definert seks ulike Requester:

Request (metode)	**Betyr/bruk**	**Gir respons?**	**Body**
INVITE	Inviterer partner til samtale (”ringer”)	Ja	Kan ha body for å beskrive ønsket sesjon (medier og formater), for eksempel i SDP.
ACK	Sendes av initiativtaker til samtale (den som sendte INVITE) når endelig 2xx-svar er mottatt for INVITE. Begynn samtale.	Nei (men gir jo samtale)	Kan ha sesjonbeskrivelse i SDP (siste forslag).
CANCEL	Nei, vil ikke vente på svar lenger. Avbryt samtaleforsøk.	Ja	Nei
BYE	Legg på røret. Samtale slutt.	Kan gjøre det	Nei
OPTIONS	Forespørsel om evne til å ta imot mulige medier/formater. OPTIONS uten body = ”hva kan du ta imot?"	Ja	Kan ha format-ønsker i SDP.
REGISTER	Registrer flytting av person til adresse. Som å sette over telefonen til annen terminal.	Ja	Foreløpig tillatt, men ingen vet hva den skal inneholde. Ny adresse er i header.

Response-typer og koder følger stort sett som for HTTP, men det har vært nødvendig med enkelte utvidelser, f.eks. ”180 Ringing”. Kodene er gruppert etter første siffer. En minimumsimplementasjon behøver bare forstå første siffer i Response-koden.

Nummergruppe	**Betyr**	**Eksempel**
1xx	Informasjon	”100 Trying”, ”181 Call is being forwarded”
2xx	Suksess	”200 OK”
3xx	Omdirigering	”302 Moved Temporarily”
4xx	Request feilet	”401 Unauthorized”, ”404 Not Found”
5xx	Server feilet	”501 Not Implemented”, ”504 Gateway timeout”
6xx	Global feil	”600 Busy”, ”603 Decline”

SDP: Session Description Protocol

SDP er en enkel protokoll for å beskrive en multimediesesjon. En SDP-melding består av en rekke tekstlinjer, der hver linje inneholder <parameter>=<verdi>. Alle parametere har navn på en bokstav, og det er strengt bestemt hvilken rekkefølge de må komme i. <Verdi> er resten av linja, og kan godt være flere ord. Meldingen består av en generell sesjonsbeskrivelse først. Deretter kommer en beskrivelse av hver av mediene som formidles i sesjonen. Sesjonsbeskrivelsen inneholder felter som identifikator, tidsperiode sesjonen er aktiv og nettverkstype/adresse, og en rekke ikke-obligatoriske felter som avsenders e-mail og ”informasjon” (som f.eks. kan være filmtittel). Mediebeskrivelsen inneholder felter for medietype (audio/video etc.), tilgjengelige formater, transportform (UDP/TCP/RTP), adresse, portnummer og andre attributter.

To felt er særlig viktige: c-feltet i sesjonsbeskrivelsen og/eller mediebeskrivelsen, og m-feltet i mediebeskrivelsen.

c-feltet spesifiserer adresse for forbindelsen. Denne har formatet
```
c= <nettverktstype> <adressetype> <adresse>
```
F.eks:
```
c= IN IP4  156.16.2.178
```
Alle adresser har dette formatet (f.eks. i feltet o= <origin address>).
c-feltet kan være i sesjonsbeskrivelsen. Hver mediebeskrivelse kan også ha et c-felt hvis de er på en annen adresse. Har alle mediebeskrivelsene c-felt, kan c-feltet i sesjonsbeskrivelsen droppes.

m-feltet beskriver et medium, og er eneste obligatoriske felt i mediebeskrivelsen. Format:
```
m= <media-type> <port> ”/” <antall porter> <transportmåte> <mulige formater>
```
Eks:
```
m=video 49170/2 RTP/AVP 31
```
beskriver video overført med RTP audio-video-profile (over UDP) med RTP på port 49170 og RTCP på 49171, format H.261. ”31” referer til payload-type 31 i RTP/Audio Video Profile (RFC 1890), H.261 ved 90000 Hz. Det kan godt listes opp mange formater. Disse tolkes da som alternativer i foretrukket rekkefølge.

Registrering av terminal/bruker

I en vanlig installasjon av SIP, for eksempel i en institusjon av Regnesentralens størrelse, vil det typisk være satt opp en sentral SIP-server. På alle brukeres terminaler vil det være en User Agent Server som startes opp når brukeren logger inn. Når en bruker sitter ved en terminal, kan meldingen REGISTER sendes til en Registrar som tar i mot slike meldinger på vegne av den sentrale SIP-serveren (antakelig vil serveren selv ta imot disse meldingene). REGISTER-emldingen kan inneholde informasjon om hvor (hvilken terminal) brukeren sitter på. En som ønsker kontakt via SIP, kan nå initiere samtaler mot brukerens User Agent Server hvis han vet hvilken terminal (internett-host) brukeren sitter ved. Hvis ikke kan han initiere samtaler mot den sentrale SIP-serveren, som vil sørge for å formidle kontakten videre til terminalen brukeren sitter på. Dette har en viss parallell til forskjellen mellom direkte innvalg og å ringe til sentralbordet for vanlig telefoni.

REGISTER-meldinger kan sendes når som helst, og kan godt inneholde mer, eller annen, informasjon enn at ”bruker X er logget inn på terminal Y nå”. Det finnes en header ”expires” som kan brukes til å uttrykke hvor lenge adressen er gyldig. En bruker kan godt være registrert med flere adresser.

Adressen i en REGISTER-melding vil ikke bare være begrenset til hvilken maskin og port han ønsker å motta samtaler på. Alle slags URI’er kan benyttes (f.eks. `mailto:navn@host)`. Man kan videre oppgi flere ulike URI, det går an å knytte en prioriteringsverdi til hver av URI’ene som oppgis. Videre kan man spesifisere hva man ønsker serveren skal gjøre (proxy eller redirect). URL’er for å uttrykke telefonnummer[6] vil også kunne brukes. Hvilke av disse mulighetene eventuelle oppringere kan benytte seg av vil avhenge av muligheter til å håndtere de ulike URI’ene i brukerprogrammet (redirect-mode) eller serveren (proxy-mode).
SIP antar at SIP-serveren som skal motta REGISTER-meldingen finnes på en fast adresse. Det er imidlertid definert en multicast-adresse for ”alle SIP-servere”. User Agent’er som vil registrere seg kan sende REGISTER på multicast med TTL satt til 1. Da vil denne nå alle SIP-servere på lokalnettet.

Oppsetting av samtale

Oppsetting av samtale kan gjøres med ned til 3 utvekslinger av pakker, flere hvis proxyer, omdirigering eller autentisering er med i bildet. En samtale eller sesjon settes opp ved at en initiativtaker sender INVITE til (SIP-User-Agenten eller SIP-serveren til) den han ønsker å invitere til sesjon. Serveren kan gi flere tilbakemeldinger, bl.a. for å si at "nå ringer det". Når den oppringte brukeren "tar telefonen" (eller tilsvarende) sendes responsen 200 OK tilbake til initiativtaker. Initiativtaker sender ACK, og samtalen/sesjonen startes med avtalte medier. Utvekslingen kan bli lenger hvis initiativtaker må henvises til annen adresse, bekrefte identitet e.l., men mønsteret er alltid det samme: INVITE – <evt. mellomliggende responser/dialoger> – 200 OK – ACK – <samtale> – BYE.

Draftet til standarden har en del eksempler, her er oppsetting av enkel samtale mellom to parter (komplett datautveksling):

***Request: Klient til Server**:*

```
INVITE sip:watson@boston.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP 169.130.12.5
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. Watson <sip:watson@bell-tel.com>
Call-ID: 2d978243-b270-33dc-a261-d1fe3e2aa05a@kton.bell-tel.com
Subject: Mr. Watson, come here.
CSeq: 17 INVITE
Content-Type: application/sdp
Content-Length: 97

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
c=IN IP4 135.180.144.94
m=audio 3456 RTP/AVP 0 3 4 5
```

Response: Server til Klient

```
SIP/2.0/UDP 100 Trying
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. Watson <sip:watson@bell-tel.com>
Call-ID: 2d978243-b270-33dc-a261-d1fe3e2aa05a@kton.bell-tel.com
CSeq: 17 INVITE
Content-Length: 0
```

Response: Server til Klient

```
SIP/2.0/UDP 180 Ringing
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. Watson <sip:watson@bell-tel.com>
Call-ID: 2d978243-b270-33dc-a261-d1fe3e2aa05a@kton.bell-tel.com
CSeq: 17 INVITE
Content-Length: 0
```

Response: Server til Klient

```
SIP/2.0/UDP 182 Queued, 1 caller ahead
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. Watson <sip:watson@bell-tel.com>
Call-ID: 2d978243-b270-33dc-a261-d1fe3e2aa05a@kton.bell-tel.com
CSeq: 17 INVITE
Content-Length: 0
```

Response: Server til Klient

```
SIP/2.0/UDP 200 OK
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: sip:watson@bell-tel.com
Call-ID: 2d978243-b270-33dc-a261-d1fe3e2aa05a@kton.bell-tel.com
CSeq: 17 INVITE
Contact: sip:watson@boston.bell-tel.com
Content-Length: ...

v=0
o=watson 4858949 4858949 IN IP4 192.1.2.3
c=IN IP4 135.180.161.25
m=audio 5004 RTP/AVP 0 3
```

The example illustrates the use of informational status responses. Here, the reception of the call is confirmed immediately (100), then, possibly after some database mapping delay, the call rings (180) and is then queued, with periodic status updates.

Watson can only receive PCMU and GSM. Note that Watson's list of codecs may or may not be a subset of the one offered by Bell, as each party indicates the data types it is willing to receive. Watson will send audio data to port 3456 at 135.180.144.94, Bell will send to port 5004 at 135.180.161.25.

By default, the media session is one RTP session. Watson will receive RTCP packets on port 5005, while Bell will receive them on port 3457.

Since the two sides have agree on the set of media, Watson confirms the call without enclosing another session description:

Request: Klient til Server

```
ACK sip:watson@boston.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP 169.130.12.5
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. Watson <sip:watson@bell-tel.com>
Call-ID: 2d978243-b270-33dc-a261-d1fe3e2aa05a@kton.bell-tel.com
CSeq: 17 ACK
Content-Length: 0
```

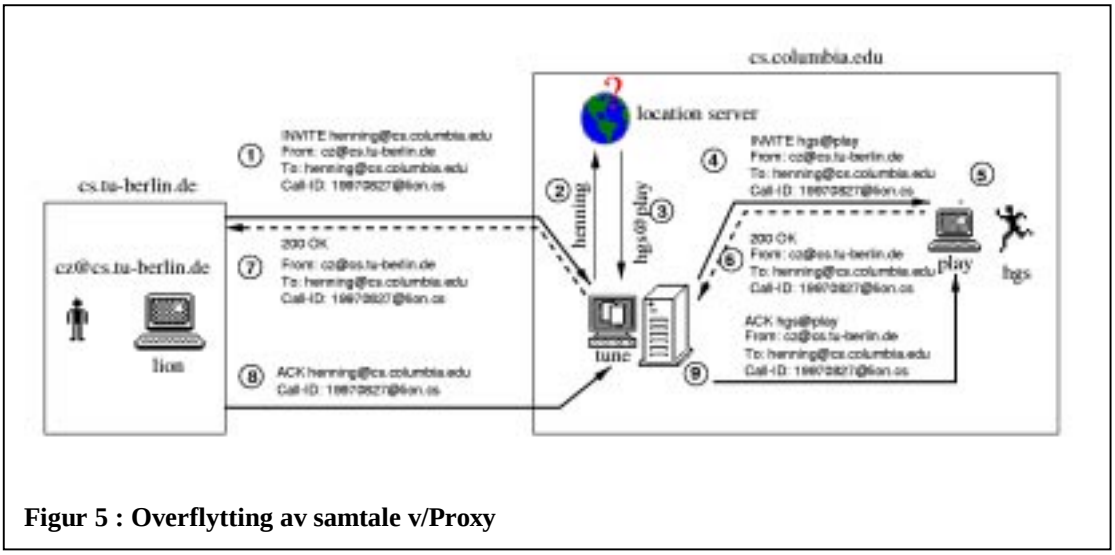
Etter dette kan samtalen begynne, dvs medieapplikasjone kan starte å sende og motta data på de porter og adresser som ble avtalt i SDP. Samtalen kan avsluttes slik:

Vilkårlig hvilken retning:

```
BYE sip:watson@boston.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP 169.130.12.5
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. A. Watson <sip:watson@bell-tel.com;tag=37462311>
Call-ID: 3298420296@kton.bell-tel.com
CSeq: 18 BYE
```

Kontakt via sentral SIP-Server: Proxy og Redirect

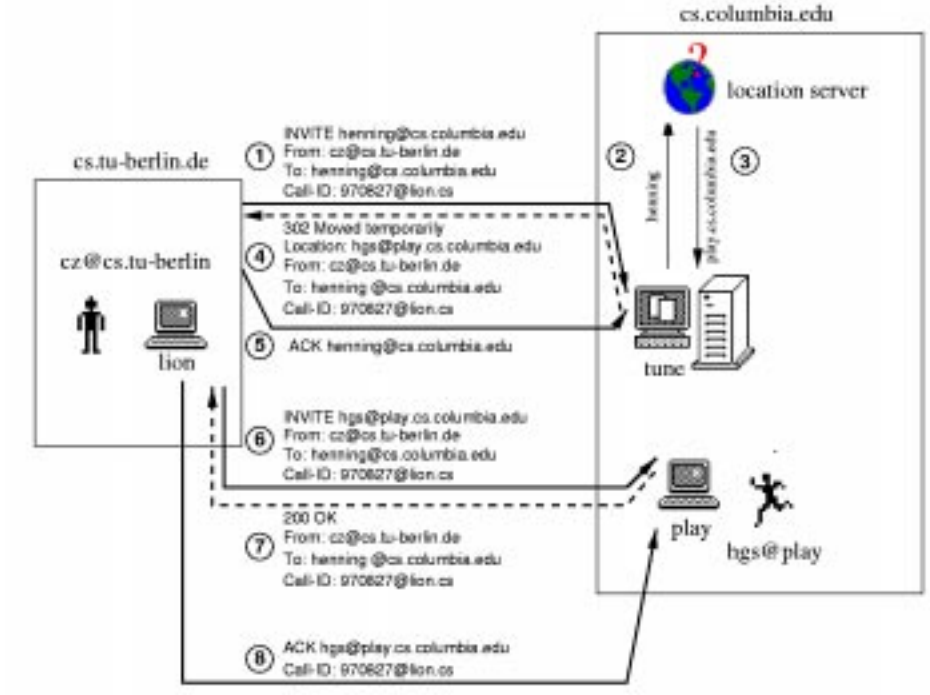
Som nevnt overfor, kan brukere registrere User Agent'en sin hos en sentral SIP-server. Det er videre mulig å presisere hva brukeren ønsker serveren skal gjøre hvis det kommer en innkommende samtale til ham. To valg er mulige: Serveren kan fungere som Proxy, og ta imot samtalen på vegne av klienten, opprette en ny samtale "på baksiden" fra Serveren til User Agent'en på terminalen brukeren virkelig befinner seg. Se figur 5. I såfall vil all signalering gå gjennom Proxy-serveren. Mediestrømmene vil allikevel bli sendt direkte mellom endeterminalene. Alle proxyer setter sin adresse inn i SIP-meldingene med feltet "Via". Siden den neste adressen også kan være en Proxy, må proxyer passe på at de ikke sender i ring. Dette gjøres ved å se om proxyens adresse allerede finnes blant Via-feltene i meldingen.



Figur 5 : Overflytting av samtale v/Proxy

Alternativet til samtaleoverføring ved Proxy er omdirigering (Redirect). Her vil serveren svare at brukeren sitter ikke ved denne maskinen, og oppgi den korrekte adressen brukeren kan kontaktes på. Se figur 6. Legg merke til at headeren ”Location” har byttet navn til ”Contact” siden illustrasjonen ble laget.

Figurene 5 og 6 viser bare tilfellet der brukeren er registrert på en adresse. Hvis brukeren er registrert på flere adresser, vil Proxy-serveren prøve alle disse i parallell og se om det er positivt svar ved noen. Redirect-serveren vil returnere mange ”Contact”-headere. Det blir opp til klientprogrammet å avgjøre om alle disse skal forsøkes kontaktet og måten dette eventuelt skal skje på.



Figur 6 : Overflytting av samtale ved omdirigering

III. Løsning på grunnleggende oppgaver:

Etter å ha gått gjennom den grunnleggende teknologien i forrige kapittel, tar de neste kapitlene for seg forskjellige områder som er viktige for å lage et salgbart produkt som tilbyr multimedietelefoni mellom internett-maskiner og til PSTN. Kategoriene er valgt ut etter skjønn, kan hende flere burde vært med. Særlig tre ting har det vært viktig å få dekket: inntjening av penger, sikkerhet og tjenestetilbud.

Å FÅ INN PENGER

Kontrollere hvem som ringer ut (brukerkontoer)

For tilbydere av telekommunikasjonstjenester er det helt grunnleggende å ha en kontroll over hvem som bruker tjenestene. Det er ikke fritt for enhver å ringe fra en hvilken som helst telefon eller koble seg til internett gjennom en hvilken som helst ISP. Særlig fordi det koster penger å produsere kommunikasjonstjenestene, må man ha kontroll over hvem som bruker dem. I telefonverden regner vi det som helt naturlig at vi får tilsendt telefonregning jevnlig. Private internett-tilbydere gjør det samme, selv om prisingen er forskjellig. Felles for begge er at man ikke får tilgang til utstyr og tjenester de leverer uten en form for ”konto” eller tilgangskontroll. For internett har det imidlertid vært vanlig at straks du er ”på”, kan du fritt og gratis sende data til alle andre maskiner som er på nettet. For telefon har man måttet be en sentral sette opp samtalen, og overføring av data (tale) har dermed lettere kunne prises etter lengde og distanse. Vi må anta at lignende mekanismer må kunne brukes på telefonlignende kommunikasjon over internett. På en eller annen måte må jo telelinjene betales.

H.323

H.323 bygger på at man i utgangspunktet har en lokal Gatekeeper som kontrollerer tilgang til nettet. Det går riktignok an å lage konfigurasjoner uten Gatekeeper, men standarden bygger på at man bør ha en, og at klienter registrerer seg hos gatekeeper f.eks. under pålogging.

Systemer kan konfigureres forskjellig, men standarden legger opp til at endesystemene skal spørre Gatekeeper om lov til å aksessere nettet før de ringer ut. De kan også settes opp til å spørre om tillatelse til å motta innkommende samtaler. Systemet kan konfigureres slik at all signalering må gå gjennom gatekeepere. Da får man kontroll med hvem som ringer ut og når.

Det er grunn til å stille spørsmål om hva man gjør med programmer som ikke registrerer seg, men som sender data direkte til mottakers endesystem. Antakelig kan man forhindre at dette skjer med en brannmur.

Særlig attraktivt er det nok å prøve anonyme oppringninger til Gatewayer mellom Internett og telefonsystemet, for på den måten å kunne ringe gratis. Vi kan anta de fleste Gatewayer vil kreve autorisering evt. fra sin Gatekeeper av alle samtaler.

SIP

I SIP er det i utgangspunktet ingen kontroll med utgående samtaler. SIP er laget mest mulig likt HTTP, og på den offisielle SIP-mailinglista har det vært snakket om å sy SIP-funksjonalitet inn i web-browsere og mulig sammenslåing av standardene på lang sikt. WWW-klienter går som kjent ikke

gjennom noen lokal myndighet for å hente web-sider, den oppretter direkte kontakt med en server i den andre enden.

Det er meningen SIP-klienter skal fungere på samme måte som web-klienter: de skal ta direkte kontakt med User Agent Server på den andre siden. For å få startet klienter må man riktignok logge inn de fleste steder med internett-tilgang. Et system med brukernavn vil sikre en viss kontroll med hvem som har tilgang til å starte SIP-klienter, men gir ikke kontroll over når og hvor de påloggede initierer samtaler ut. En brannmur kan nok kontrollere dette ved å kreve at SIP-meldinger går gjennom en lokal Proxy. Dette vil fungere som Gatekeeper-routed call control i H.323.

Tjene penger på forbruk

For kommersielle tjenestetilbydere i internett-domenet er det et must å kunne ta betalt for tjenestene. På samme måte som telefon over telenettet i dag koster penger å bruke, er det sannsynlig at firmaer som tilbyr telefon over IP ønsker å ta betalt for selve tjenesten.

I vanlig telefoni er det to ting som bidrar til prisen på en samtale: distanse og varighet. Denne prisingen følger mengden ressurser telefonsamtalen legger beslag på. Samtaler krever at det blir satt av ressurser i alle linjer og sentraler langs veien. Jo lengere distanse, jo flere ”dingser” må avsette ressurser. Mengden ressurser lagt beslag på ganges med hvor lenge brukeren legger beslag på ressursene.

Prising av vanlige telefonsamtaler er langt lettere enn prising av dataoverføring på internett. Alle telefonsamtaler bruker den samme båndbredde, og får den samme type tjeneste utført. Lengden av samtalen blir dermed bestemmende for ”datamengden” eller størrelsen på den tjenesten kunden får utført.

Tradisjonelt har det ikke vært tatt betalt for bruk av internett, bare for tilknytning. På internett finnes det ulike tjenester og ulike mengder data som skal overføres. Ofte vil kvaliteten variere, med forsinkelser og tap av data. For multimedieoverføring betyr dette at lyd eller bilde kan gå tapt i perioder eller være ustabile. For tjenester som filoverføring, der det er viktig at alle deler kommer fram, betyr dette at biter må sendes på nytt, og at tiden det tar å få utført overføringen blir lenger. Under slike forhold vil noen oppfatte det som urettferdig å betale etter overføringstid, når de som får dårligst tjenestekvalitet levert bruker ekstra lang tid (og dermed må betale ekstra mye). Når man legger til at båndbredden for lyd og bilde med dagens kompresjonsstandarder varierer med mengden detaljer og bevegelse i bildet, blir det er vanskelig å si (i det minste på forhånd) nøyaktig hvor stor båndbredde som kreves. Dermed skulle det være klart at prising på internet-sesjoner er langt vanskeligere enn telefonsamtaler.

Dette kapittelet vil se på hvordan leverandører kan ta betalt for bruken av teletjenester over internett. Viktige spørsmål blir da hva det kan tas betalt for og hvor det skal måles hen når det er avvik mellom hva som blir sendt og hva som blir mottatt.

H.323

Så lenge man har en implementasjon / oppsett av H.323 hvor terminalene registrerer seg hos Gatekeeper og lar signalering gå gjennom Gatekeeper, skulle det være mulig å ha en viss kontroll med samtalens lengde og ressursforbruk. Dette bør gi muligheter for å belaste forbruk. Dessverre tar det ganske lang tid å sette opp telefonsamtaler på denne måten (se fig. 4). Siden dette likevel er måten å få tak i informasjon om samtalen, kan vi anta at de fleste installasjoner vil benytte Gatekeeper og Gatekeeper-routed call-control.

Ericsson sa de kunne monitorere RTCP-pakker som utveksles om dataflyten på mediekanalene (lyd og bilde). RTP/RTCP-standarden nevner også eksplisitt at den kan brukes av en tredjepart for å monitorere mengde og kvalitet på dataoverføring. Man kan få tilsendt RTCP-pakker ved selv å sende

en RTCP-pakke til den porten de utveksles på. Får man tak i RTCP-informasjon finnes det grunnlag for mange ulike måter å ta betalt for forbruk på.

Uansett hvilken måte man velger å ta betalt for forbruket på trengs det en slags brukerdatabase i tilknytning til registrering av terminalene og monitorering av bruk. Denne må ha et konto- eller forbruksberegningssystem implementert. Siden registrering av terminaler gjøres til Gatekeeper, og det er Gatekeeper som har ansvar med adgangskontroll og ressursreservasjoner (det siste er frivillig), er det naturlig at brukerkontoer også håndteres av denne.

Ericsson (ETO) har laget et system der støtte-servere kan abonnere på hendelser (”events”) fra Gatekeeper, som genererer en mengde forskjellige eventer. En støtte-server for betaling kan for eksempel abonnerer på ulike samtale-events og beregne avgift på en samtale på bakgrunn av forskjellige attributter ved samtalen, bl.a. varighet, samt data om hva slags telefonabonnement denne abonnenten har osv. En annen støtte-server kan f.eks. abonnere på registrerings-eventer og sjekke om brukeren som prøver å registrere terminalen mot gatekeeper virkelig er kunde ved denne tilknytningsleverandøren.

SIP

I SIP skal ikke klienter ”si fra” til noen lokal entitet før de initierer sesjoner. Klientprogrammer sender initieringssignaler direkte til mottakers User Agent Server. Det er derimot muligheter til å spesifisere krav til sesjonen ved hjelp av header-feltet ”Require”. SIP-draftet har følgende eksempel:

```
Klient => UAS:
      INVITE sip:watson@bell-telephone.com SIP/2.0
      Require: com.example.billing
      Payment: sheep_skins, conch_shells
      ... <flere> . . .

UAS => Klient:
      SIP/2.0 420 Bad Extension
      Unsupported: com.example.billing
```

Legg merke til at ”Payment” er en del av ”Require”-headeren, bestemt av de parametere som må til for Require-type ”com.example.billing” (selv om den står på egen linje). Servere som ikke forstår innholdet av en Require-heading må svare ”420 Bad Extension”.

Hvordan betalingen nå foregår er avhengig av betydningen av ”com.example.billing”. Kanskje dette er en egen applikasjon som blir startet av SIP-serveren eller klienten. ”Require” kan også ønskes av den som blir kalt opp. Det er også mulig å kreve at den andre parten skal identifisere seg skikkelig (header WWW-Authenticate).

Vi skal her legge merke til at ”Require” her må kreves av en av partene i samtalen. Dette avviker fra H.323 hvor Gatekeeper typisk vil kreve autentisering for deretter automatisk å beregne pris på samtalen.

En annen måte for å få betalt for tjenesten i SIP kan være å bruke Proxy i kombinasjon med brannmur slik at utgående samtaler må gå via Proxy. Dette vil likne på Gatekeeper i H.323. Proxyer kan stille krav om at brukere skal bekrefte identitet (Proxy-Authenticate). Dette kan brukes til å logge når samtaler settes opp. Da kan man få vite om adresser for RTCP-pakker også. Proxyer kan også stille egne krav til initiativtaker, med headere som ”Proxy-Require” og ”Proxy-authenticate”.

Oppsummering av inntjeningskontroll

For å få kontroll over hvem som foretar utgående samtaler trenger begge standarder en eller annen form for brukerkontoer. Til å belaste brukerkontoer etc. kan for eksempel RADIUS (RFC 2138) eller DIAMETER (draft) benyttes, eller man kan (i H.323) koble gatekeeper-funksjonaliteten til en database/støtte-server.

Selve samtalene er det ikke like lett å få kontroll over. I begge standarder kan dette gjøres ved at en eller annen sentral enhet, i H.323 Gatekeeper, er involvert i samtaleoppsettet. I SIP kan man signalere gjennom proxyer og ha brannmurer som sperrer for å gå utenom. Dette vil øke oppsettingstiden for samtaler, men vil kanskje være nødvendig i miljøer hvor de ulike brukerne selv kan installere/konfigurere programvaren på terminalene.

Begge protokoller kan benytte RTCP til å få oversikt over mengde og kvalitet på datautvekslingen. Det er imidlertid bare H.323 som spesifiserer at RTP skal brukes til utveksling av data for sanntidsmediene. I praksis vil det nok bli brukt i SIP også, bl.a. fordi SDP er best utviklet til å annonsere sesjoner som overføres med RTP. Problemet blir å få tak i hvilken port man skal be om disse på. Her kan man sikkert benytte brannmurteknologi til å lytte etter RTP/RTCP-pakker, selv om dette har et preg av avlytting.

I det sivile samfunn er flere betalingsformer med lav sikkerhet forholdsvis utbredt, f.eks. VISA-kort. Systemet bygger på at folk i utgangspunktet oppfører seg skikkelig, og at antallet forbrytelser er lavt. Vi kan anta det samme gjelder for denne typen tjenester: I hovedsak vil folk benytte de programmene de får anvist. Svært få vil benytte programmer laget spesielt for å gi falske lave verdier i RTCP-meldinger, sette opp samtaler uten å registrere seg, dytte betalingen over på fremmede eller snike seg unna betaling på andre måter. Enda færre vil lage slike programmer, selv om dette er fysisk mulig.

Når alt kommer til alt, er ikke sikkert at det er så farlig om det er vanskelig å få helt sikker oversikt over datautvekslingen og hvem som ringer når og hvor lenge. Selve oppsettingen av samtalene er lite ressurskrevende, bare litt mer enn å hente web-sider. Det som tar ressurser er først og fremst overføring av lyd- og bilde-data. Dette tar særlig ressurser hvis man benytter ressursallokeringssystemer som for eksempel RSVP til å få garantier for kvaliteten på overføringen. I dag er det få som benytter seg av RSVP, men vi kan regne med at utbredelsen vil følge utbredelsen av telefoni over internett for å sikre og stabilisere kvaliteten på sesjonene. Siden det er garantier for overføringer av store mengder sanntidsdata som er dyrt og vanskelig, er det sannsynlig å anta at det vil følge prisingsmekanismer med selve reservasjonssystemet, og at ingen programmer kan reservere båndbredde uten å bekrefte brukeridentitet og betalingsvilje. I såfall vil tjenesteleverandører være sikret betaling for den delen av tjenesten som er ressurskrevende. Da er det kanskje ikke så farlig at det er mulig å snike seg til gratis samtaler bygget på best-effort-overføring.

SIKKERHET OG PRIVATLIV

Sikkerhet er et tema som ofte kommer opp i internettsammenheng. Det er lite tenkelig at kommersielle aktører kan markedsføre telefoni over internett uten å dokumentere at ”sikkerheten er på topp”, og at systemet er ”sikkert”.

Dette kapittelet vil fokusere på noen aspekter vi kan knytte til sikkerhet: Kontroll med hvem som ringer til deg, og at de faktisk er den de utgir seg for å være, muligheter for kryptering av viktig informasjon og muligheter for avlytting.

Hvem ringer?

Mange ønsker å vite hvem som ringer før de velger om de skal svare eller ikke. I tillegg kan det tenkes at noen ønsker å filtrere samtaler på bakgrunn av identiteten til den som ringer eller adressen

det ringes fra. ISDN har gjort denne teknologien mulig for alminnelige telefonapparater, man kan få opp nummeret det ringes fra på skjermen før man bestemmer seg for om man vil svare eller ikke.

H.323

H.323 støtter dette ettersom avsenderens nettadresse må være med i pakkene. Det er meningen at den oppringte part skal kunne være konfigurert slik at den må spørre Gatekeeper om lov til å ta imot en samtale. Det kan hende en implementasjon vil spørre brukeren også, slik mange telefonapparater gjør i dag ved å vise nummeret det ringes fra.

SIP

I SIP må også avsenders navn og adresse være med i meldingene. Her skal man legge merke til at det kan være flere adresser assosiert med en "avsender". Det ene er adressen som brukes som avsenderadresse, og som kan brukes hvis du skal ringe tilbake en annen gang. Denne refererer gjerne til domenet, på samme måte som man vil sette navnet til bedriften, ikke nummeret på kontoret, som avsender på et brev. Den andre adressen er navnet på den terminalen du for tiden sitter ved og som data skal sendes til. Dette må være adressen til en spesifikk terminal, og vil være annerledes hvis du ringer fra en annen terminal en annen gang. Dette kan også referere til en terminal helt utenfor det vanlige domenet ditt, f.eks. en "telefonkiosk". Standarden sier at den første adressen i det minst bør vises før brukeren velger om oppringningen skal aksepteres. Et tredje aspekt er Via-feltene i headeren som indikerer veien gjennom SIP-servere, proxyer, gatewayer osv. meldingene skal følges. Disse kan krypteres på per-strekning-basis slik at hver proxy/gateway bare kan spore navnet på den nærmeste i kjeden. Hvis SDP-beskrivelsene av hvor medie-data skal sendes indikerer multicastadresser eller adresser på tallformat som ikke kan oversettes til domenenavn på tekstform, er det svært vanskelig å ha noen håndfast avsenderadresse som kan bekreftes. I slike tilfeller kan man imidlertid avslå samtaleinvitasjonen og prøve å ringe tilbake til avsender-adressen i From-feltet.

Kontrollere tilgang til og fra terminaler

Mange ønsker å ha kontroll med hvem som ringer inn til en terminal på en strengere måte enn bare å få vite hvilket nummer det ringes fra eller hvem oppringeren utgir seg for å være.

- I forbindelse med salg kan det være nødvendig å få bekreftelse på at den som ringer faktisk er den han utgir seg for å være.
- Andre selger tjenester som hjelp og kundestøtte bare til personer med spesielle abonnementer e.l. I såfall gjelder det å forhindre at personer som ikke er kunder kan ringe inn.
- Et tredje tilfelle tilsvarer "hemmelige telefonnummer". Kjente personer, toppledere o.l. kan nok antakelig tenke seg en begrensning i hvem som får lov til å ringe dem. Innkommende telefonsamtaler må bekrefte at de kommer fra "godkjent venn" før de slippes gjennom.
- Et fjerde tilfelle er ønske om å forhindre brukere av et nett å komme i kontakt med visse adresser, også ved oppringning. For eksempel for å forhindre at studentene bruker universitetets anlegg til kontakt med grupper som driver lyssky aktiviteter, eller forhindre at anlegget blir brukt til underholdning på formiddagstid.

I de tre første tilfellene handler det om at innringer har et spesielt passord eller at han kan bekrefte at han vet om en hemmelighet (for eksempel "private key") som deles mellom han og den som blir oppringt. I det siste vil man avskjære visse adresser evt. bare i visse perioder. Vi skal også huske på at personer vil kunne kjennes igjen på stemme og/eller bilde i motsetning til bokstavbaserte medier over internett.

H.323

H.323 støtter selektivt mottak av samtaler. I konfigurasjoner hvor terminalene registrerer seg hos Gatekeeper skal terminalene spørre om tillatelse til å motta innkommende samtaler. Gatekeeper kan avvise dette ut fra forskjellige kriterier, noe som er et spørsmål om implementasjon og konfigurering av Gatekeeper. Det later til å være vanskelig å kontrollere terminal-programvare som bryter reglene og ikke spør om tillatelse fra Gatekeeper. Her kan avvisning av resursreservering til visse adresser

eller tradisjonell brannmur-teknologi kanskje avhjelpe problemet. Alle gatewayer til andre nett vil sannsynligvis kreve at samtalekontrollen går gjennom Gatekeeper. Det går videre an å sette opp en brannmur eller ruter som en Gateway mot internett, slik at samtaler ut kan kontrolleres. Dette er imidlertid avhengig av at terminalen befinner seg på innsiden av brannmuren for å virke.

Det finnes også autentiseringsmekanismer og mekanismer for å gjøre kontrollkanalen kryptert. H.235 er en egen standard om sikkerhet i H.323 og lignende systemer.

SIP

SIP støtter Autentisering på samme måte som det finnes i HTTP. I tillegg støttes PGP-autentisering der innringer må kode en melding etter oppgitt standard. Bare hvis både oppringer og oppringt deler har samme nøkkel kan de forstå meldingen. Eventuelt kan en tredjepart signere (kode) meldingen for å verifisere identiteten.

Det går i tillegg an å kryptere mesteparten av SIP-meldingen, f.eks. i Response fra en oppringt User Agent. I såfall vil bare samtalen kunne startes opp hvis den som ringer har nøkkelen til å dekode meldingen.

I spesifikasjonen ”Requirements for SIP Servers and User Agents” (Schultzerinne, 98) anbefales et script-system som gjør at brukere kan gi instruksjoner til serveren om hvordan samtaler til dem skal håndteres. Det betyr at oppringere som ikke har adressen direkte til terminalen til den de ønsker å kontakte, men som må gå gjennom domenets SIP-server for å få kontakt, kan enkelt sorteres etter navn, tidspunkt, adresse osv. Kombinert med et oppsett av User Agent og Server slik serveren må brukes som proxy, kan dette være meget effektivt.

Brukere som installerer programvare som bryter de lokale reglene kan nok by på problemer her. Kontroll av eventuell resursreservering og brannmur mot visse adresser kan kanskje hjelpe, men det blir vanskelig å forhindre kommunikasjon over multicastadresser uten å ramme konferansefunksjonaliteten, siden denne baserer seg på bruk av multicast.

Hemmelighold, kryptering og muligheter for avlytting

Å kontrollere samtaler til og fra terminaler er viktig, men vel så viktig er det å garantere at kommunikasjonssystemet er sikret mot at uvedkommende kan få fatt i sensitiv informasjon. Dette kan de få på flere måter:

1. Svikefull programvare i en av klientene/terminalene sender kopi av samtaledata til tredjepart, eller tredjepart klarer å lure den vanlige programvaren til å gjøre dette.
2. Tredjepart kan snike seg med som lyttende medlem i en konferanse uten at de andre merker det.
3. Svikefulle nettverksforbindelser og/eller rutere langs veien dupliserer pakker og sender til tredjepart. Hvis dette er kontrollpakker, kan tredjepart få tak i nok informasjon til å kunne etterlikne andre brukere.
4. Trafikkovervåkning på nettet gjør det mulig for uvedkommende å spore hvem som snakker med hvem når.

De tre første tilfellene vil nok være de mest alvorlige, og er nok det de fleste forbinder med avlytting. Kryptering av lyd- og bilde-data kan gi en løsning her, men det er vanskelig å finne sikre krypteringsmetoder med tilstrekkelig lave krav til prosessering. Å beskytte informasjon om hva slags adresser disse dataene sendes mellom kan være et effektivt hjelpemiddel for å forhindre tapping av data i disse tilfellene.

Det siste tilfellet kan imidlertid også være alvorlig, og innebærer at informasjon lekker ut på en måter man oftest ikke tenker på. I sektorer som Jus og finans kan det ligge mye informasjon i å vite hvem som snakker med hvem. At multimedie-kommunikasjon gir langt større og mer kontinuerlige datastrømmer mellom partene enn tradisjonell datautveksling gjør det lett å oppdage hvem som snakker med hvem i små deler av nettet med få samtidige samtaler.

Det vil falle utenfor denne tekstens tema å ta opp usikkerhet i selve nettverksteknologien som gjør at IP-pakker, trafikk-informasjon eller reservasjonspakker på avveie blir en sikkerhetsrisiko. De to første av tilfellene over (1. og 2.) er imidlertid relevante for dette notatet.

H.323

H.323 har en forholdsvis sterk sikkerhetsmodell. Det finnes en egen standard H.235 (tidl. H.Secure) om sikkerhet, autentisering og kryptering i H.3xx-systemer. Implementasjon av denne standarden er ikke påkrevet av H.323-systemer, men på sikt vil nok mange støtte den.

H.235 spesifiserer muligheter for autentisering og kryptering av kontrollkanalen. Dette vil gjøre det langt vanskeligere å få tak i adressene mediedata sendes til, og følgelig også mediedataene. Dessuten vil det bli vanskeligere for uønskede parter å "blande seg inn i" utvekslingen av kontrollmeldinger (H.245-meldinger). De som legger vekt på sikkerhet vil nok benytte en konfigurasjon av H.323 med gatekeeper hvor alle terminaler må avklare alle samtaler inn og ut med gatekeeper. I såfall er det vanskelig å sette opp ekstra samtaler i bakgrunnen hvor man sender kopi av konferansen, selv for illojale parter i konferansen som prøver dette med vilje.

Det skal ikke være mulig for programvare som følger spesifikasjonene å bli med som bare lyttende part i en sentralisert konferanse uten at konferansens Multipoint Controller (MC) vet om dette. Så lenge denne kan stoles på, skulle dette være sikkert. Automatisk utvidelse av samtale til konferanse når tredjemann ringer er imidlertid mulig, og kan kanskje virke som en fristende konfigurering for mange. Dette vil nok kunne brukes til avlytting hvis konfigurasjonen er sånn at bruker ikke spørres før tredjeman er med i samtalen. Det finnes dessuten mulighet for at klienter setter opp desentraliserte konferanser over multicast-adresser i H.323. Hvis multicast over internett brukes til dette, kan nær sagt hvem som helst be om å ta imot medie-pakkene. I såfall kan nær sagt hvem som helst bli "blindpassasjerer" i konferansen.

Kryptering av kontrollkanalen kan gjøre det vanskelig å finne ut hvilken multicastadresse som benyttes for desentralisert konferansesesjon. I tillegg kan selve medie-data-pakkene krypteres. I sentraliserte konferanser er det støtte for at hver link mellom MCU og endesystem benytter forskjellige nøkler.

SIP

SIP-standarden gir omtrent de samme muligheter som H.323, med den forskjellen at kontroll av utgående samtaler ikke foretas i SIP (men kan ordnes utenfor standarden likevel).

Å forhindre at uønskede tredjepersoner blir lyttende "blindpassasjerer" i samtaler skal være mulig. SIP spesifiserer riktignok at samtaler kan forwardes automatisk ved å inkludere "Contact" i BYE. Ved å sette Contact til en multicastadresse kan samtaler enkelt og automatisk overføres til konferanse. På multicastadresser kan i prinsippet hvem som helst lytte, med mindre rutere og brannmurer konfigureres spesielt for å hindre dette. I motsetning til H.323 er det ikke lagt opp til automatikk i at invitasjoner til personer som allerede er i en samtale kan føre til at samtalen blir omgjort til konferanse med alle 3. Dette er mulig å lage, men er ikke beskrevet i noen spesifikasjon.

Tredjeparter kan ikke uten videre blande seg inn i samtaler, blant annet fordi de mangler informasjon som bare deles mellom partene i samtalen: Call-ID og CSeq. I SIP finnes det muligheter for å kryptere det aller meste av kontrollinformasjonen i SIP-meldingene, inkludert disse feltene. Krypteringen vil da også gjøre SDP-informasjon om adresser for datautveksling hemmelig. I tillegg kan selvfølgelig medie-dataene krypteres. Det er opp til prosseseringsevnen og programvaren partene benytter til selve medieutvekslingen å bestemme hvordan dette skal foregå.

Utvidelsen "org.ietf.sip.call" for SIP Call Control Services definerer en ny header "Also" som gir mulighet for begrensede konferanser basert på alle-til-alle-unicast. Det er usikkert hvor mange implementasjoner som vil støtte denne utvidelsen. Unicast har nok sikkerhetsmessige fordeler framfor

multicast. På den annen side kan Also brukes til automatiske overføringer av samtaler og inkludering av tredjeperson, noe som kan ha sikkerhetsmessige ulemper hvis ikke brukere bli bedt om å bekrefte før det de sier blir sendt til nye adresser.

Andre angrep – denial of service

Både SIP og H.323 kan møte sikkerhetsmessige problemer i forbindelse med registrering av terminaler hos Gatekeeper/Registrar. Det forventes at ingen Server/Gatekeeper aksepterer registreringer uten en eller annen form for autentisering.

Registreringsproblemet er størst i SIP, der man kan registrere terminaler over hele internett. (Det vil antakelig være implementasjonsavhengig eller konfigurerbart om dette er lov i H.323, se registrering av terminaler i teknologioversikten).

En uvennlig maskin kan (i SIP) registrere sin adresse som brukerens, ved å sette adressen sin inn i en REGISTER-melding. Hvis den samtidig ber hjemme-serverern fungere som Proxy, kan den lett forårsake denial-of-service attacks. Proxyer som får innkommende samtaler skal i følge standarden videresende disse i parallell til alle registrerte adresser. Hvis en uvennlig maskin har fått registrere seg kan den da umiddelbart sende svar i klassen 500 eller 600. Dette kan føre til at hele samtaleforsøket blir oppgitt av Proxyen som foretar parallellsøket. 500/600-svaret vil nesten helt sikkert komme fram til proxyen før et eventuelt 200-OK-svar fra den terminalen brukeren faktisk sitter på, etter som denne siste må vente på brukerens bekreftelse på at samtalen ønskes.

I H.323 er registrering av terminaler utenfor lokalnettet et vanskelig og uklart tema p.g.a. adresse-problemene som oppsto da H.323 ble omdefinert fra lokalnett til generelle pakkebaserte nett (les internett). I utgangspunktet kan man nok bare registrere seg med terminaler som ligger i lokalnettet til hjemme-gatekeeperen. Det kan likevel hende at feil oppstår så en bruker blir assosiert med en terminal han ikke (lenger) sitter ved. Hvis han da prøver å registrere en ny terminal krever standarden at han avvises. Man kan ikke assosieres med to terminaler. Terminal og Gatekeeper kan imidlertid bli enige om Time To Live for en registrering. En terminal som ikke gjenoppfrisker registreringen innen det avtalte antall sekunder kan regnes som utmeldt av Zonen (utregistrert). Gatekeepere som tillater registrering av terminaler utenfor lokalnettet må foreta grundig autentisering for å unngå at uvedkommende registrerer seg under falsk identitet. Uvennlig registrering av ”falsk” terminal vil både føre til at uvedkommende vil få overført oppringningsforsøk til seg og at den som egentlig eier identiteten ikke kan registrere seg hos Gatekeeper. Å tillate registrering utenfor lokalnettet vil derfor føre til vel så store sikkerhetsproblemer i H.323 som i SIP.

FÅ I STAND KOMMUNIKASJON OG TJENESTER

Signalere og sette opp sesjon

Det mest grunnleggende for å produsere tjenester som ligner på telefoni er at det går an å ”ringe opp” en annen part man ønsker å kommunisere synkront med. Helst vil man at dette skal gå unna ganske raskt, så man i høyden må vente et par sekunder. Både H.323 og SIP forsøker å gjøre dette mulig mellom datamaskiner på nett.

H.323

H.323 bruker ASN.1-definerte meldinger i hovedsak bestående av tall og bitfelt. H.245 definerer en mengde slike meldinger (nær 100). Hver melding har en klar og avgrenset oppgave/betydning, og et begrenset antall felt med data. Dette gjør det klart enklere å lese hver enkelt melding isolert sett.

Ulempen er at siden store mengder data om adresser, preferanser om mediers kanaler og formater osv. må utveksles før samtale kan komme igang, må forholdsvis lange sekvenser av meldinger utveksles før samtalen er igang. Dette tar tid. Å sette opp en sesjon (eller samtale) i H.323 er en noe omstendig affære, og kan ta ganske lang tid sammenlignet med SIP. Tre ting bidrar til at det blir omstendig. For det første settes det opp sikre kanaler (les TCP) for kontroll/Call setup. Dette innebærer en hel pakke-rundtur for å sette opp kanalen før noe kan sendes over den. For det andre går i utgangspunktet all kontrollinformasjon gjennom gatekeeper, og dette fører til at en mengde kontrollkanaler og tillatelser må komme i orden før man kan sende. For det tredje utveksles lite informasjon av gangen gjennom meldingene, så mange sett av meldinger må sendes fram og tilbake før sesjonen kan begynne.

H.323v2 har definert "fast-start" hvor alle pakker samles til et "Jumbogram"[10]. Dette sparer pakke-rundturer, og vil føre til at oppsetting av samtaler går fortere. Figur 4 viser at mesteparten av pakke-utvekslingen likevel kan gå med til å sette opp og ta ned kontrollkanaler.

SIP

SIP sender lange tekstmeldinger mellom oppkalt server og initiativtaker, der hver linje (header-felt) i meldingen inneholder omtrent det samme som en melding eller meldingsfelt i H.323/H.245. Med mange headere i hver melding, blir sekvensen av meldinger kort. En samtale kan oftest starte etter 1,5 – 3 runder med meldinger, mens 4 - 10 er vanlig i H.323. At informasjonsfeltene som utveksles er tekstbasert, gjør at felt for informasjon man ikke ønsker å uttrykke kan utelates. Dette forhindrer at meldingene blir unødvendig store. For de vanligste headere er det videre definert navneforkortelser på én bokstav. Det er antatt at overheadet i utvekslet datamengde og prosesseringstid for å parsere teksten vil være helt ubetydelig i forhold til en bit-orientert løsning.

Både SIP og H.323 baserer seg på at det finnes programvare som venter på innkommende samtaler på en forutbestemt port.

Finne adresser og personer

Både SIP og H.323 tillater at det går an å initiere kall direkte til den terminalen en person sitter ved, så lenge man vet adressen, og systemet ikke er konfigurert til å stoppe slike samtaleforsøk. Begge skiller også mellom personidentitet og terminal, og at det finnes en sentral instans/entitet som vet hvilken terminal en gitt bruker sitter ved. Invitasjoner til samtale kan da rutes gjennom denne sentralinstansen eller spørre denne om adressen til en bruker, og dermed bli satt direkte over til terminalen brukeren for tiden sitter ved. Denne arkitekturen bygger på to ting for å gjøre det mulig å ringe til personer og terminaler utenfor et begrenset område. For det første må en sentral enhet ha en oversikt over "sine" lokale brukere. For det andre må det være mulig å identifisere hvilke annen slik sentral enhet som skal kontaktes for å få tak i terminalen til en bruker som tilhører et annet "domene".

H.323

H.323 bygger på at Gatekeeper skal oversette mellom "interne" og "eksterne" adresser og mellom "e-mail-type" adresser og terminalens Call-Control-adresse (host + port) ([1], kap. 7.1.3).

FINNE PERSON I EGEN ZONE

H.323 spesifiserer at elle tilgjengelige brukere registrerer seg hos Gatekeeper når de logger seg inn, skrur på terminalen e.l. I tilknytning til Gatekeeper vil det nok derfor være et tabellsystem, en database e.l. som gjør at Gatekeeper kan finne adresser, gitt en H.323-personidentifikator (e-mail-type adresse e.l.).

Endepunkter eller en Gatekeeper som ønsker å få vite terminal-adressen til en person (H.323-identifikator) gjør dette ved å sende Location Request (LRQ) på Gatekeepers RAS-adresse. Gatekeeper

svarer på dette med Location Confirm (LCF). LCF inneholder den adressen som skal brukes til Call-control. Hvis Gatekeeper vil at Call-control skal gå gjennom Gatekeeper, setter den altså sin egen adresse inn i denne pakken, ellers settes terminalens.

FINNE PERSON I EN ANNEN ZONE

Terminaler som vil ringe bruker Gatekeeper til å finne adressen. Gatekeeper vil etter all sannsynlighet bare vite om adressene til personer som er registrert, eller pleier/har tillatelse til å registrere seg i Gatekeepers Zone. En Gatekeeper som ønsker (evt. på vegne av en terminal) å finne adressen til en person i en fremmed Zone må derfor sende LRQ til Gatekeeper for brukerens Zone. Men hvordan finner man ut hvilken Gatekeeper dette er snakk om og hvilken adresse den er på?

I dette spørsmålet merker man at H.323 i utgangspunktet ble utviklet for lokalnett. Det finnes nettopp ingen spesiell løsning for hvordan man finner ut hvilke navn som hører til hvilke Gatekeepere/Zoner. H.323 ble opprinnelig laget med hver Zone som et lokalnett-segment og en forståelse av at Gatekeepere som skulle (behøve å) kommunisere "visste om hverandre".

Det er ingen måte for en Gatekeeper å finne ut hvilket domene e.l. et navn hører til, hvis det ikke er dens eget. Gatekeepere har imidlertid en "Well-known Gatekeeper Discovery multicastadresse" ([1], kap. 7.2.3), og LRQer kan sendes på denne adressen. Alle gatekeepere må motta på denne adressen. Den Gatekeeper som trenger å få adressen for et navn sender altså LRQ på multicast, og den gatekeeper som "kjenner igjen navnet" som ett av sine "egne" svarer LCF ([1], kap. 8.1.6) med endepunktet eller sin egen adresse.

Å bruke multicast som adresseoppslagsmekanisme på internett kan virke, men er en løsning med mange problemer. For det første bli TTL på multicastpakker kraftig diskriminert på Mbone multicast på internett. Det er vanlig å trekke 128 "poeng" (av 255 mulige) for forbindelser mellom kontinenter. Hvis andre linker også trekker mange "poeng" vil kanskje ikke LRQ-meldingen nå fram. Grunnen til at pakker forsøkes stoppet tidligst mulig, er for å prøve å begrense hvor stor del av nettet som skal belastes. Hvis alle adresseoppslag skal spres til nesten hele internett, vil nettet etter all sannsynlighet til å bli sterkt belastet, bare med adresseoppslag. Denne metoden skalerer overhodet ikke til millioner av installasjoner verden over. En forutsetning for at denne teknologien skal virke blir dermed at den har begrenset utbredelse.

SIP

SIP er utviklet i samme miljø som mange andre internettprotokoller, og benytter eksisterende internett-teknologier til å finne fram på nettet.

FINNE PERSON LOKALT

En SIP-server er koblet til en Location-service lokalt. Denne Location service henger sammen med en Registrar, som tar imot REGISTER-meldinger fra brukere som logge seg inn. Antakelig vil alle disse tre henge sammen som en SIP-server med mulighet til å ta imot REGISTER-meldinger og en database til å holde oversikt over hvor de forskjellige brukerne er registrert hen. Hele denne arkitekturen er avhengig av at brukerprogrammet (User Agent Server) kan finne SIP-serveren. Det er definert en multicastadresse som kan brukes lokalt hvis SIP-serveren ikke er på noen fast adresse eller User Agent'en ikke vet hva denne adressen er.

Når Serveren mottar INVITE vil den sjekke med Location service hvor brukeren er registrert. Hvis serveren er en Proxy vil den akseptere INVITE-requesten og sende en ny INVITE med de samme parametere videre til User Agent Serveren på terminalen brukeren sitter ved. All signalering vil da gå gjennom Serveren, akkurat som signalering kan gå gjennom Gatekeeper i H.323. Hvis serveren er en Redirect server vil den svare "302 Moved Temporarily" med en Contact header som inneholder adressen (URL) brukeren er registrert ved.

Hvis brukeren er registrert ved flere adresser (ikke bare maskinnavn, se s. 14) vil en proxy-server prøve å kontakte alle adresser i parallell, mens redirect-servere bare vil returnere alle adressene for at brukerprogrammet skal finne ut hvordan de skal behandles. Antakelig vil en proxy-server bare forsøke videre på SIP-URIer. Proxy for ”mailto:” virker helt meningsløst.

Her kan vi legge merke til at standarden ikke beskriver hva slags Location Service som skal brukes og hvordan serveren skal bruke denne. Ett eller annet sted bør det være noen som tar imot REGISTER-meldinger. Disse kan være til hjelp for å finne hvilken terminal en person sitter ved. Andre måter kan også brukes, som ’finger’. Hvilken måte som velges er opp til implementasjonen av serveren å bestemme. Leverandører vil kanskje gi kunder mulighet til å velge mellom flere alternative Location-service-løsninger.

FINNE PERSONER I ANNET DOMENE

SIP bygger på adresser som ligner på mail-adresser, nemlig `sip:bruker@maskin.domene`. Delen etter ’@’ er det tekstlige navnet på maskinen hvor brukerens SIP-server kjører. Denne blir oversatt til IP-adresse ved hjelp av vanlig Internett Domain Name Service (DNS). SIP prioriterer de nye SRV DNS recordene, og UDP framfor TCP, men ellers er det likt som e-mail eller HTTP. Straks et klientprogram finner adressen til brukerens SIP-server kan denne kontaktes og videre adresseoppløsning fortsetter som over.

Forhandlinger

Mye forskjellig er gjort innen multimedia på data, og mange ulike praksiser og standarder finnes. Særlig gjelder dette kodingsformater for lyd og bilde. For å få til kommunikasjon mellom tilfeldige datamaskinterminaler er det nødvendig at disse kan bli enige om hvilken standard man skal følge, hva man skal utveksle og lignende. Andre funksjoner, særlig innen de standardene som defineres av ITU, krever at spesielt koordineringsansvar e.l. tillegges en av partene. I såfall må man bli enige om hvem som skal ha dette ansvaret.

H.323

Forhandlinger i multimediesystemer handler stort sett om at den ene part formidler sine formatønsker i prioritert rekkefølge til den annen part, som så velger det første den støtter. I så måte har H.323 en forholdsvis sterk forhandlingsmekanisme for hvilke medier og formater som skal utveksles.

Innerst i datastrukturene som utveksles er det lister av alternativer den andre kan velge en av, f.eks. forskjellige kodingsformater for samme lydstrøm. Disse en-av-listene pakkes inn ”samtidig-strukturer” som uttrykker at en fra hver av disse en-av-listene kan presenteres samtidig. Disse pakkes i sin tur inn i alternativ-lister som betyr ”enten disse eller disse”.

Eksempel på betydning (fra H.323-standarden):

```
Enten:
        Samtidig:
                En av: G.711 eller G.723.1 eller G.728
        Og
                En av: H.261 eller H.263
        Og
                En av: H.261
Eller:
        Samtidig:
                En av: H.262
        Og
                En av: G.711
```

Detaljer innen kodingen av medietypene kan forhandles, men bare allerede definerte standardverdier kan forhandles. Det kan bare velges mellom de medieformatene som er definert i H.245. Portnummerene utveksles ikke før i senere meldinger. Det finnes prosedyrer for å reforhandle eller åpne nye medier underveis i sesjonen.

H.323 krever dessuten at en av terminalenes Multipoint Controller (MC) skal være ”master” i tilfelle konferanse blir aktuelt. Den/de andre blir da ”slave”. Det ”forhandles” om dette ved å utveksle en tallverdi for terminaltype MC’en befinner seg i. Verdiene er definert i standarden slik at MC’er i MCU’er har rang over gatekeepere, som har rang over gatewayer , som har rang over terminaler. Det gis ”bonuspoeng” for ulike multipoint processors en MC kan ha, slik at den som i størst grad støtter sentraliserte konferanser ”vinner” forhandlingene.

SIP

SIP bruker SDP til medieforhandlinger. SDP er tekstbasert. Dermed kan man i prinsippet beskrive absolutt alle medier og formater så lenge begge parter forstår kodene. I hovedsak vil antakelig RTP-profilen for audio/video over nett (AVT RFC 1890 under oppdatering) brukes. Denne definerer payload-type for alle typer i H.323/H.245 og noen til. Typene blir definert med egne beskrivelser (for det meste tall, og ikke etter den utbredte MIME-standarden.

SDP har ikke samme støtte for å lage alternative valg-grupper som H.245. Dette byr på problemer hvis man skal lage en gateway mellom H.323 og SIP. Man kunne anta at det gikk an å lage flere SDP-meldinger, en for hver alternativ-gruppe, men dette blir vanskelig å få mottakeren til å lese siden all SDP skal utveksles inni en SIP-melding og det ikke er noen mulighet å uttrykke at Body inneholder flere enheter. I praksis blir det umulig å uttrykke alternativ-grupper i SDP/SIP.

SIP tilbyr heller ikke forhandlinger om hvem som skal være ”master” i konferanser, ettersom dette begrepet ikke finnes.

SIP tilbyr kontinuerlige reforhandlinger. En ny INVITE med samme Call-ID skal betraktes som en oppdatering av samtaleopplysninger. En kan her f.eks. legge til nye medier i SDP-meldingen.

QoS og reservasjoner: muligheter og ansvar

I flere år har det vært gjort forsøk med sending av sanntidsdata over internett. Den store forskjellen fra formidling av filer og tekst er at man ønsker at nettet skal kunne garantere en viss kvalitet på gjennomstrømningen m.h.p. overføringstid, båndbredde o.l. Sanntidsdata som skal konsumeres med en gang, som direktesendt lyd og bilde, er uinteressante hvis de blir forsinket i nettet og ikke kommer fram tidsnok. Det er gjort mye grunnforskning på hvordan man kan garantere kvaliteten på tids- og mengde-aspekter i dataoverføringstjenestene (Quality of Service – QoS [13]) på internett. Å garantere en viss gjennomstrømning og maksimal forsinkelse er en forutsetning for å kunne tilby stabile multimediesesjoner der mediene spilles av i sann tid. (Video-)Telefoni er en tjeneste som kan trenge slike garantier.

Forskning hittil har basert seg på at QoS-garantier skal kunne gis til programmer som reserverer båndbredde e.l. Det er interessant å se hvordan de to standardene legger opp til at reservasjoner og kontroll med overføringsressurser skal foregå.

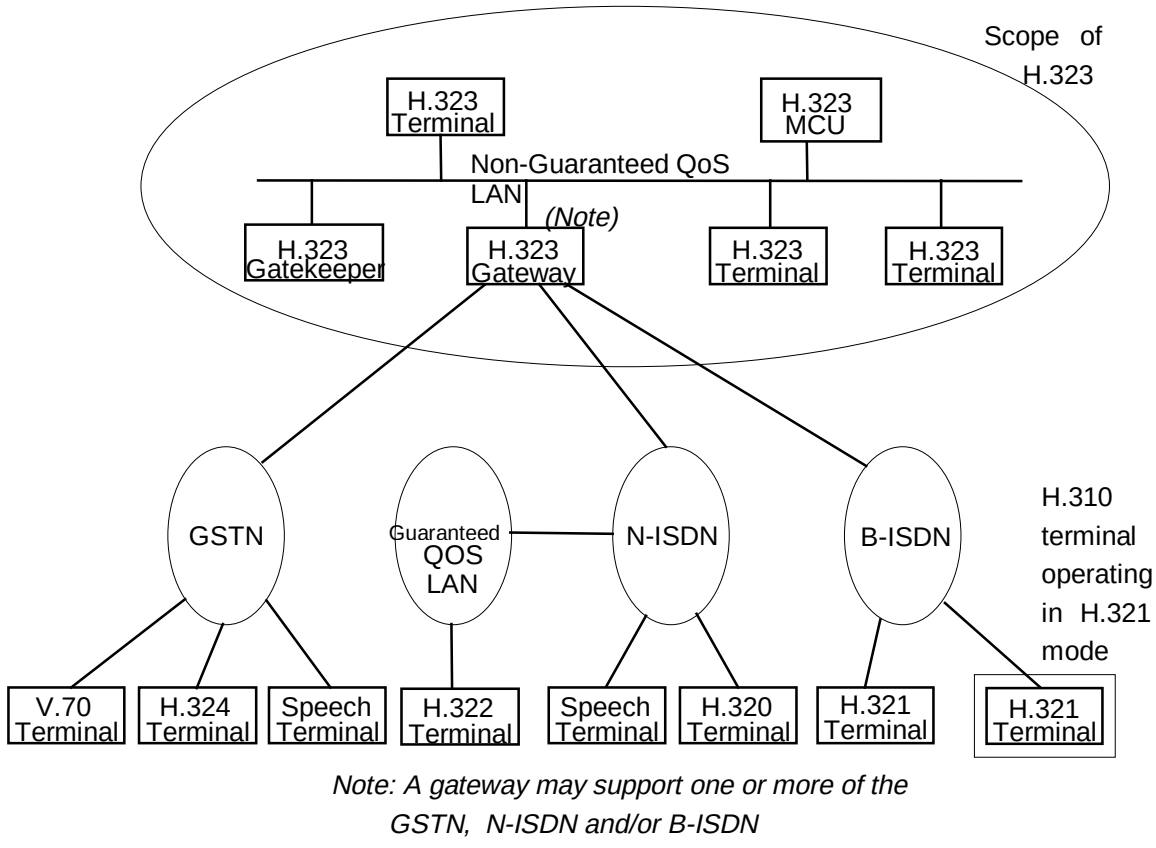
H.323

I H.323 er det spesifisert at alle implementasjoner av Gatekeeper skal svare korrekt på RAS-meldingen Bandwidth Request (BRQ), dvs. med Bandwidth Confirm (BCF) eller Bandwidth Reject (BRJ). Det er ikke krav om at Gatekeeper må reservere og kontrollere båndbredde og andre QoS-resurser, det er tillatt å svare BCF til alle. Hvis det finnes funksjonalitet for å reservere båndbredde i systemet, er det imidlertid Gatekeeper som skal stå for dette. Gatekeeper skal foreta de nødvendige handlingene for å få reservert de etterspurte ressursene på vegne av klientene, og klientene skal kommunisere sine behov med Gatekeeper gjennom BRQ/BCF/BRJ. Standarden legger opp til at IETF-protokollen RSVP (RFC 2205, [14]) skal brukes som reservasjonsprotokoll på nettet.

I SIP-standarden står det eksplisitt at resursreservering er utenfor protokollen. Det antas applikasjonene som foretar selve utvekslingen og avspilling av medier foretar reserveringer på egne vegne.

Interoperasjon med telefonsystemet og andre systemer.

Begge standarder *kan* interoperere med det eksisterende telefonsystemet gjennom såkalte gatewayer. En terminal vil da ringe til gatewayen på samme måte som om den var en terminal eller proxy server. Gatewayen oppleves som endesystem både for H.323/SIP-terminalene og for telefonen i den andre enden. Gatewayen må oversette signaler og lyd-formater mellom de to sidene. Per i dag er det bare for H.323 det finnes standarder for disse terminalene. For SIP er planene mer på skisse-stadiet. I tillegg til SIP arbeides det med en egen standard kalt PINT (PSTN and Internet Interworking, [15]) og en måte å beskrive telefonnummer som URLer [6].



Figur 7 : H.323 med Gateway til andre nett/terminaltyper (fra [1]).

Figur 7 viser hvordan H.323 tenkes å kunne kommunisere med andre nett gjennom en Gateway. Det er også foreslått Gatewayer for SIP, for eksempel i forbindelse med arbeidet med PINT. Signaliseringen i H.323 bygger delvis på etablerte ITU-standarder, bl.a. Q.931. Hovedforskjellen i signalering mellom SIP/H.323 og vanlige telefonnett er at multimediekonferansesystemer som SIP og H.323 må utveksle mye mere informasjon enn for eksempel telefoner før samtale kan komme igang. Å oversette signalering for å koble opp enkle samtaler er derfor ikke vanskelig. Det er først når signalering skal oversettes mellom to ulike og komplekse standarder, som f.eks. mellom SIP og H.323

at det kan oppstå problemer med datastrukturer og semantikk som ikke kan oversettes og med data som ikke utveksles i samme rekkefølge. Det kommer et eget kapittel om H.323 og SIP sammen lenger bak.

Diskret mobilitet (flytting mellom samtaler)

Med diskret mobilitet menes å ”flytte telefonen” mellom to samtaler (uten at noen samtale pågår). Dette innebærer på en eller annen måte å meddele en eller annen entitet i kommunikasjonssystemet at samtale-/konferanseinvitasjoner til deg ønskes levert ved en spesifikk terminal, gjerne en annen enn den man pleier å sitte ved.

Her må vi merke oss at dette kan gjøres på to måter. Den ene måten, som er ganske utbredt for telefoner, er å flytte seg fra en terminal. Hvis man for eksempel ønsker å sette over telefonen mens man er på ferie, kan man slå en spesiell kode fra den telefonen man flytter fra, og så nummeret til den telefonen man vil flytte innkommende samtaler til.

Den andre måten er å ”ta med seg telefonen”. Dette innebærer å meddele ”nå er jeg her, og nå vil jeg ha alle telefoner hjem sendt hit i stedet”. Støtte for en slik tjeneste må være slik at ikke hvem som helst kan be om å få samtaleinvitasjonene dine sendt til seg i stedet.

Forskjellen mellom disse to måtene er altså med den første måten flytter du samtaler før du drar, fra din vanlige ”Hjemme-terminal”. Med den siste metoden flytter du over samtaler etter at du har kommet fram. Her trenger du med andre ord ikke vite adressen/navnet/nummeret til den nye terminalen før du drar. Her blir autentisering av at den som prøver å ”hente samtaler” virkelig er den han utgir seg for å være, slik at ikke hvem som hels skal kunne be om å overta samtalene dine.

Adresse- og navngivningssystemet vil selvfølgelig ha stor innvirkning på mulighetene til at mobilitet kan være transparent for den som ringer til deg.

H.323

H.323 støtter i utgangspunktet bare ”hent telefon” og ikke ”flytt telefon” . En brukeridentitet (e-mail-type adresse e.l.) er bare assosiert med en terminal så lenge en terminal har registrert seg med den identiteten. En person er i utgangspunktet bare assosiert med en terminal så lenge han er logget inn eller tilsvarende. Hver gang en terminal registrerer seg, må dette kunne ses på som å ”hente telefonen” til den terminalen han sitter ved. Det finnes derfor ingen måte innenfor H.323-standarden f.eks. for en sjef å kunne sette over terminalen sin til sekretærens e.l. for en periode. Dette utelukker selvfølgelig ikke løsninger utenfor standarden hvor applikasjoner blir enige om at sjefens H.323 skal avregistrere seg (med Unregister Request, URQ) og sekretærens applikasjon skal registrere seg med sjefens identitet imens. At sekretærens terminal her vil bli assosiert med flere navn er tillatt.

SIP

SIP støtter på en enkel måte diskret mobilitet. Når en bruker skrur på terminalen og logger inn startes en lokal User Agent Server. Denne sender Requesten REGISTER til SIP-serveren for brukerens hjemmedomene og sier fra hva som er adressen til terminalen brukeren sitter på. Dette er basert på SIP-adresse-URLer og fungerer over hele internett. Brukernavnet behøver ikke engang være det samme. Det er laget utkast til URL’er for telefonnummer [6], så denne ”hentingen” kan fungere hvis brukeren kan registrere at han er tilgjengelig på vanlig telefon gjennom en gateway også.

Til forskjell fra H.323 kan SIP-registreringer også gjelde for perioder der brukeren ikke er logget inn (e. tilsv.). Dette gjør det mulig å registrere seg på en terminal en ikke har logget inn på enda (e.tilsv.), dvs. forhåndsflytning av ”telefonen”. REGISTER gjør det dermed mulig å kunne sette over sine innkommende samtaler til en annen URI, f.eks. til en medarbeiders adresse men du er i møte.

REGISTER tillater mange samtidige registreringer for samme person, samt at hver registrering tilordnes en prioriteringsverdi mellom 0 og 1. Det finnes videre mulighet til å spesifisere metoder utenfor SIP, som `http:` til sider med informasjon, `mailto:` for å legge igjen tekstbeskjeder,

RTSP[16] eller SIP til ”telefon”-svarer (terminalsvarer?) osv. En SIP-klient (eller Proxy) som forstår mange ulike URIer vil her kunne få oppgitt en mengde måter å kontakte personen på, selv om han ikke kan motta samtaleinvitasjoner for øyeblikket.

Kontinuerlig mobilitet (flytting under samtale)

Flytting mens samtale pågår, kontinuerlig mobilitet, er litt vanskeligere enn å sette over telefonen, ettersom hele samtalen (konferansen?) må flyttes med uten å avbrytes. Det enkleste er selvfølgelig å avslutte hele samtalen, for å ringe opp på nytt når man har kommet til den nye terminalen. Brukere vil nok oppfatte dette som litt tungvint, særlig hvis det er snakk om konferanser, som jo tar litt tid å sette opp. Av denne og flere grunner kan det være attraktivt å flytte samtaler uten å måtte ta den ned og sette den opp igjen.

H.323

Flytting av igangværende samtaler i H.323 er spesifisert i [1] kap. 8.1.8 Call Forwarding: Et endepunkt som vil Forward’e en samtale sender Facility til partner med opplysning om den nye adressen. Partner-terminal svarer Release-Complete, tar ned sesjonen og initierer en ny sesjon mot den nye terminalen.

Hvis sesjoner betales omtrent på samme måte som telefonsamtaler i dag vil vel dette bety at kostnadene blir dyttet over på motparten når du flytter deg over til en annen terminal.

Et problem med denne løsningen i H.323 vil være at samme person ikke kan være registrert på to terminaler i samme Zone (dvs. hos samme Gatekeeper) samtidig. Den gamle terminalen må avregistreres og den nye terminalen må registreres før samtalen kan bli tatt imot på den nye terminalen. Noen Gatekeepere vil kanskje tillate at den gamle terminalen sender Unregister Request (URQ) mens samtalen er i gang, altså før ”Facility” er mottatt, selv om dette semantisk sett er helt ulogisk (personen sitter jo vitterlig ved terminalen fortsatt). Hvis ikke dette tillates er funksjonaliteten avhengig av at sekvensen URQ (gammel term.), RRQ (Ny term.) går fort nok til at ikke samtalepartner får timeout på Setup-meldingen, og at Gatekeeper (gitt gatekeeper-routed call-control) ikke mottar Setup før etter at både URQ og RRQ er behandlet. Det finnes ingen metode innenfor standarden å si til den nye terminalen at den må sende RRQ.

Flytting av konferanser gjøres enklest ved å invitere den nye terminalen, for deretter å melde den opprinnelige terminalen ut av konferansen. Standarden beskriver i detalj hvordan utvidelse og innskrenkning av konferanser skal foregå. Den letteste metoden for å flytte en igangværende samtale over til en annen terminal, vil derfor være å gjøre den om til konferanse, invitere den nye terminalen og utmelde den gamle.

SIP

SIP følger samme metode som H.323. Den som flytter seg sender BYE med headeren Contact som beskriver den nye terminalen. Motparten må da initiere kall til den nye terminalen. Dette kan skje automatisk på bakgrunn av Contact. Contact kan da inneholde adressen direkte til User Agent Serveren på den nye terminalen, men alle slags URIer kan plasseres her, akkurat som for REGISTER-meldinger. Automatisk flytting til multicastadresser kan imidlertid ha sikkerhetsmessige problemer, ettersom hvem som helst i prinsippet kan motta data herfra.

SIP er litt uklar på betydning og formidling av adresser i konferanser, men baserer seg uansett på bruk av Mbone multicast. Antakelig vil den enkleste måten å flytte over en konferanse på bli som i H.323: meld inn den nye terminalen og avslutt den gamle.

Dataapplikasjoner/delt whiteboard

H.323

Som det går fram av Figur 2, er det klart spesifisert at delte applikasjoner skal følge T.120. Dette er et hierarkisk sett protokoller som definerer bl.a. synkroniseringspunkter for meldinger, kontroll-hierarkier mellom partene, m.m. for delte-data-applikasjoner. At et produkt for distribuert datasamarbeid er konformt med T.120 betyr imidlertid ikke at det kan brukes sammen med H.323. Det er H.323-signalering som foretas for å avtale når og hvor T.120-data skal utveksles.

Bruk av dataapplikasjoner avtales under forhandlingene om medietyper og terminalegenskaper. Å åpne T.120-sesjoner i en samtale som er igang, vil si det samme som å reforhandle medier og kapabiliteter. Det finnes også mulighet for å starte en H.323-sesjon og koble den til en T.120-sesjon som eksisterte på forhånd.

SIP

Alle typer medier som kan spesifiseres i SDP kan utveksles på alle måter over. Dette inkluderer dataapplikasjoner. Det finnes ingen standard i SDP for hvordan disse applikasjonene skal være (pakker, dataformater, kontroll, synkroniseringsmekanismer osv.). Dette antas allerede avtalt mellom partene hvilket program man skal bruke.

Samtaler med mer enn 2 deltakere (konferanser)

H.323

Samtaler med mer enn 2 parter må koordineres av MC-enheten som kan være i et av endesystemene eller en gatekeeper, eller av en MCU. Det er definert prosedyrer for hvordan man kan invitere flere parter til en konferanse eller hvordan de kan spørre om å få være med. En og bare en MC er Master for konferansen. Terminalene (og evt. gatekeepere) forhandler seg imellom om hvem som skal være Master. Hvis en gateway eller en gatekeeper er med i signaliseringsveien mellom partenes terminaler vil disse bli Multipoint Controller (MC) for konferansen, ellers vil en av terminalene velges. Dette blir avgjort under oppsettingen av samtalen. H.323 spesifisere at mediestrømmene i en konferanse kan gå gjennom en sentral Multipoint Controller Unit, men at desentraliserte og blandete konferanser også er lov.

Konferanser kan settes opp med (sentralisert) eller uten en MCU. Uten en MCU må alle sende alle data til alle. En MCU kan brukes som ”distribusjonssentral”, og hvis den inneholder en Multipoint Processor, kan den mikse alle lyd- og bildedata til ett lydspor/videospor. Mange terminaler vil nok ikke ha ytelse nok til å kunne dekode flere videoer samtidig. Det går også an å ha blandede konferanser der en MCU mikser noen av deltakerne, mens de andre sender alle-til-alle. H.323 har veldig god støtte for denne type ting. Se også utvidelse av samtale til konferanse.

Konferanse gjennom en MCU antas oppsatt ved at den første deltakeren ringer til MCUen. Deretter kan han få MCUen til å invitere de andre partene. Hvordan blandede konferanser skal foregå er i liten grad beskrevet.

I tillegg til de planlagte konferansene finnes såkalte Ad-hoc-konferanser der en vanlig samtale mellom to parter utvides til en konferanse for å kunne ta med flere parter. Å åpne for konferanse betyr å sende en Setup-melding inneholdende konferanse-id (CID) og feltet ConferenceGoal satt til "create".

Under konferansen vil MC kontrollere alle innmeldinger og utmeldinger av konferansen. Hvis noen av terminalene som ikke er MC mottar en samtaleinvitasjon må den svare at den er i konferanse og hvem som er MC. Den oppringende part kan da ringe til MC'en og spørre om å få lov til å bli med i konferansen (conferenceGoal=join). Bare MC kan invitere nye parter til konferansen. Hvis noen av de andre terminalene vil invitere en ny part sender de Setup-meldingen til MC med adressen til den som

skal inviteres (og av hvem). MC vil da sende riktig Setup videre. All koordinering, forhandling etc. med den nye terminalen gjøres av MC på vegne av konferansen.

MC sender melding til alle parter i konferansen når en terminal har kommet til eller forlatt konferansen. Når en ny kommer til i konferanser hvor alle sender data til alle, må de gamle terminalene få beskjed om å sende data også til den nye terminalens adresse.

SIP

Samtaler med flere parter settes opp ved hjelp av internett multicast-adresser. Så kan man invitere alle parter til "samtale" med multicastadressen. Dette forutsetter at alle parter kan ta imot flere lyd-/bildesesjoner samtidig, og kan komme i konflikt med funksjoner som er forberedt på automatisk køing av invitasjoner som kommer til en bruker som allerede er med i en sesjon (se eksempel på oppsetting av samtale under SIP).

Også SIP-meldingene (ikke bare mediestrømmene) kan sendes over multicast. Ettersom SIP mangler noen standard for sentralisert konferanse-kontroll, er dette eneste måte som er beskrevet i standarden til å opplyse alle om alle deltakere og mediestrømmer.

Multicast har litt uheldige sikkerhetsmessige konsekvenser, ettersom hvem som helst kan da det imot bare de vet adresse og portnummer. Det foreslås i SDP-protokollen å kryptere denne informasjonen (Body i SIP-pakkene) for å holde dette hemmelig, men det garanterer jo ikke at uvedkommende kommer over det på slump.

I utvidelsen org.ietf.sip.cc [8] foreslås headerene "Also" og "Requested-By" for å lage konferanser basert på punkt-til-punkt-forbindelser mellom alle deltakere. Dette vil gjøre konferanser lettere, men krever mye ressurser.

At sentralisert konferansekontroll ikke er beskrevet i standarden, betyr ikke at det ikke er mulig å bygge det. Sentralisert konferansekontroll kan lages med en spesiallaget proxy som alle deltakere har punkt-til-punkt-samtale med. Proxyen kan kombinere alle mediestrømmene den får inn til en, eller den kan sende alle mediestrømmene til alle klientene. Den eneste forskjellen klientene vil merke er hvor mange separate mediestrømmer de må håndtere og hvor ofte "samtalepartneren" (konferanse-proxyen) reforhandler antall mediestrømmer. Dette er delvis beskrevet i [17].

Utvide samtale til konferanse

H.323

Et av endepunktene eller en Gatekeeper må ha en MC. Samtaler kan bli gjort om til konferanser med 3 eller flere parter. Enten skjer dette fordi en av partene inviterer en tredje person, eller fordi denne tredjemann ringer til en som allerede er i en samtale og blir invitert til konferanse av partner som alt snakker med noen.

Hvis en konferanse inviterer en part som allerede er med i en annen konferanse, kan konferansene slås sammen.

Det er definert detaljerte regler for oppretting, utvidelser, sammenslåing osv. av konferanser.

SIP

SIP spesifiserer at meldingen BYE innebærer overflytting av sesjonen hvis den inneholder headeren "Contact". BYE med overflytting til multicastadresse kan brukes til å "sette over" en samtalesesjon til en konferanse. Utvidelsen "org.ietf.sip.call" definerer headeren "Also" som gir støtte for å invitere flere personer i alle-til-alle-unicast-konferanser, men det er usikkert hvor mange som kommer til å implementere støtte for denne utvidelsen.

Andre telefoni-relaterte tilleggstjenester

Vi har i dette kapittelet sett på noen viktige tilleggstjenester som opprinnelig har blitt utviklet for telefonnettet. Det finnes imidlertid flere.

H.323

H.323 bygger i hovedsak på de tjenestene som er presentert hittil i kapittelet. I tillegg kan ytterligere tjenester bygges ut på nettverkslaget av en Service Provider. I den grad Ericssons Supplementary Services Execution Environment (SSEE) [17] eller lignende blir utbredt, vil også kunder kunne spesifisere tjenester de ønsker levert via Gatekeeper.

SIP

Med utvidelsen org.ietf.sip.cc [8], støtter SIP veldig mange av det som ellers er tilleggstjenester man ofte må betale ekstra for i telefonisammenheng. Forskjellen er at SIP bygger på at mye av logikken ligger i klientprogrammet, og at dette i større eller mindre grad må forholde seg til komplekse tilbakemeldinger med flere handlingsalternativer. I tillegg til det som tidligere er angitt, foreslår [19] at en entitet som mottar en innkommende samtale, eksekverer et script for å finne ut hva som skal gjøres. Disse kan settes opp på servere og User Agent'er og eksekvere på bakgrunn av mottakers personlige konfigurering, oppringers adresse osv. Dette er imidlertid langt fra standardisert, og vil antakelig heller aldri bli det.

Oppsummering av tjenester

Det finnes en mengde tjenester definert for telefoni. I [20] finner vi følgende oppsummering, gitt SIP-implementasjon som inneholder støtte for "org.ietf.sip.cc":

FEATURE	SIP	H.323
Blind transfer	Yes	Yes
Operator Assisted Transefer	Yes	No
Hold	Yes, through SDP	Not yet
Multicast Conferences	Yes	Yes
Multi-unicast Conferences	Yes	Yes
Bridged Conferences	Yes	Yes
Forward	Yes	Yes
Call Park	Yes	No
Directed Call Pickup	Yes	No

Som vi ser av tabellen og av sammenligningen forøvrig støttes omtrent de samme tjenestene av SIP og H.323. Hvordan de utføres, og hvor implementasjonen av dem finnes, kan imidlertid variere.

IV. SIP og H.323 sammen

Så lenge begge standardene ser ut til å ha oppbacking er det sannsynlig at implementasjoner av begge standarder kommer til å få en viss utbredelse. H.323 ligger litt foran i løypa, og det er det flere årsaker til. For det første bygger det på eksisterende standarder for (video-)telefoni og er til en viss grad interoperatibelt med disse. For det andre er det etablert modeller for å ta betalt, slik at det kan bli lønnsomt å tilby H.323-tjenester. For det tredje er H.323 backet opp av tunge aktører i telekommunikasjonssektoren. SIP har imidlertid en viss støtte i internett-miljøer. Også aktører som konsentrerer seg om H.323 har fått øynene opp for sider der SIP er sterkere (f.eks. [21]). En rekke implementasjoner av SIP er dessuten ferdig eller underveis, og det er grunn til å tro at disse vil få en viss utbredelse i det minste i akademiske miljøer. Da blir det straks interessant å se om interoperasjon mellom systemer og komponenter fra forskjellig standard er mulig.

Interoperasjon kan man tenke seg på to måter. Enten ønsker man å sette opp et system som inneholder "det beste" fra hver standard eller eliminerer ulemper i den ene. Eller så ønsker man en slags gateway eller lignende som gjør konferanser og samtaler mulig mellom H.323-utstyr og SIP-utstyr.

DET BESTE FRA HVER STANDARD

Som vi har sett i (de to) forrige kapittel/kapitlene, er de funksjonelle forskjellene på de to standardene ikke så stor. I hovedsak er forskjellene i signalering og i den kontrollfunksjonen gatekeeper utgjør i H.323. Dessuten har H.323 en eksplisitt "sjef" (Master) i samtaler og konferanser for å gjøre sentralisert kontroll mulig.

Det finnes en del funksjonalitet som ikke er den samme i begge standardene. Brukere vil kanskje ønske seg å sette sammen ”bokser” fra begge standarder for å få til for eksempel gatekeeper-type kontroll i et SIP-miljø. Så lenge det er forskjeller i signalering, kan man selvfølgelig ikke bare sette gjenstander/entiteter fra den ene standarden inn i et system fra den andre standarden og få det til å virke. Da blir det interessant om det går an å implementere tilsvarende funksjonalitet innenfor den andre standarden. Ellers må man ha en slags adapter/oversetter/gateway mellom de to standardene, noe som kommenteres i neste avsnitt.

Raskere SIP-type signalering i H.323

Vi har valgt å anse SIP-standardens signalering for en fordel i forhold til H.323. Det kan diskuteres om å basere standarden på tekstmeldinger er å sløse med båndbredde eller prosessering i endesystemene. Vi velger å betrakte denne som ubetydelig. Helt sikkert er det at SIP har en fordel i at programvare for håndtering av tekstmeldinger er mer fleksibel og lettere å debugge. SIPs styrke ligger i hovedsak i at de omfattende tekstmeldingene (og protokollen forøvrig) gir drastisk kortere sekvens av meldingsutvekslinger enn det H.323 legger opp til.

H.323v2 har forsøkt å korte ned meldingsutvekslingen ved å definere Fast Start jumbo-meldinger som inneholder alle opplysninger som skal utveksles i forhandlingene i fase B og C av sesjonsoppsettet. For implementasjoner som støtter dette kan oppkoblingstiden reduseres betydelig. Mye av pakkeutvekslingen skyldes imidlertid at kontrollkanalen for samtalen benytter TCP-kanaler som må settes opp. Dette gjør at forbedringspotensialet er begrenset. Det står eksplisitt at denne må gå over pålitelig dataoverføringskanal. Dermed er UDP utelukket. En viss begrensning kan man likevel få ved å la signaliseringen gå direkte mellom endepunktene, og ikke gjennom Gatekeeper. Dette er beskrevet i standarden. Ulempen med dette er at man mister de sikkerhetsmessige fordelene en Gatekeeper kan gi. Dessuten krever det at man vet adressen (host og portnummer) til mottakerterminalen.

Gatekeeper-funksjonalitet i SIP.

Bruk av gatekeeper i H.323 kan gi visse sikkerhetsmessige og administrative fordeler. Kan hende vil noen ønske tilsvarende funksjonalitet i SIP.

Admission Control

Kan delvis implementeres i SIP ved hjelp av brannmur og en proxy. Dette gjør at signaliseringen tar lenger tid, men tillegget blir mye mindre enn gatekeeper-routed call-control i H.323, siden UDP benyttes. I de fleste lokalnettmiljøer er det ikke mulig å forhindre at en datamaskin sender data til en annen på lokalnettet. Dette gjelder selvfølgelig også om de utveksler H.323-data. Å sikre tilgangskontroll vil da bety å sikre seg at ingen terminaler inneholder programvare som kan brukes til "uønsket" datautveksling.

Adresseoversettelser

Adresseoversettelse gjøres allerede av eksisterende systemer i SIP. Adressehåndteringen (oversette fra navn til transportadresse) for adresser til terminaler i andre Zoner i H.323 er dårligere enn i SIP, og særlig er den lite skalerbar. Faktisk snakkes det i H.323-kretser om å bruke SIP til adresseutveksling mellom Gatekeepere. Det er med andre ord ingen grunn til å bruke H.323s adressesystem i SIP.

Båndbredde- og QoS-reservasjoner

Det finnes i liten grad støtte for dette i dag. På sikt vil resursallokeringsprotokoller som f.eks. RSVP være mer utbredt. Det er antakelig ingen ting i veien for å lage programvare som gir Gatekeeperliknende grensesnitt mot RSVP, men det spørs hvor hensiktsmessig det er.

Autorisasjonskontroll

Det kan argumenteres for at dette hører sammen med admission control. Så lenge admission control gjennomføres ved å bruke en SIP-server som proxy kan proxyen kreve autorisering. Allerede dekket av standarden.

Generell kontroll i Zonen

Det er ikke klart hva dette skal være. Soner er et ukjent begrep i SIP-verden. Så lenge det refererer til et lokalnett kan vi anta dette er ivaretatt av driftsavdelingen og programvare i nettsystemet.

Sentralisert konferansekontroll i SIP

H.323 definerer programvare kalt Multipoint Controller som finnes i de fleste entiteter. I sesjoner/samtaler velges MC'en i en av de deltakende entitetene til "master", de andre blir "slave". Konferanser er koordinert av MC'en som er master. I sentraliserte konferanser sendes all lyd (og evt. bilde) på punkt-til-punkt-kanaler mellom master og hver av deltakerne.

I SIP finnes det ikke slik sentralisert konferansestruktur. Konferanser er i hovedsak bygget på bruk av Mbone multicast til distribusjon av mediestrømmer. Alle som er med kan invitere nye til å delta i sesjonen. Det går an å konfigurere alle parter så all SIP-signalisering går gjennom en sentral proxy, som dermed får kontroll, men konferansen forblir åpen så lenge mediestrømmene sendes via multicast og hele konferansen er en samtale.

Konferanser med sentralisert kontroll, som i H.323, kan etterlignes ved å la deltakerne "snakke" med en fiktiv bruker på konferanse-proxyen med punkt-til-punkt-forbindelser, der alle brukere har forskjellig Call-ID. Dette er ikke nevnt i standarden, og vil kreve spesiell programvare i proxyen. Konferanse ved punkt-til-punkt-forbindelser krever mye båndbredde samlet sett, og dette kan være et problem. En kan tenke seg å lage en konferanse-proxy med spesiell hardware for å mikse mediestrømmene fra alle deltakerne, akkurat som MCU'er i H.323 kan inneholde såkalte Multipoint Processors.

GATEWAY MELLOM SIP OG H.323

For å kommunisere mellom eller gjennom utstyr fra ulike standarder kreves en gateway for å oversette informasjonen som utveksles. Funksjonaliteten til gatewayer mellom hver av de to standardene og telefonnettet er allerede beskrevet i standardene eller dokumenter i tilknytning til dette. Gatewayer mellom de to systemene er derimot ikke definert.

Gatewayer kan brukes på to forskjellige måter. Enten kan man sette en gateway "midt i nettet" for å kunne formidle samtaler mellom et SIP-basert system på den ene siden og et H.323-basert system på den andre siden. Eller så kan man lage gatewayer nær terminalene for å sette en SIP-terminal inn i et H.323-mlijø eller omvent. Funksjonalitet i SIP-H.323-gatewayen blir litt forskjellig avhengig om den stå innenfor eller utenfor gatekeeper/SIP-server.

Gateway "midt i nettet"

En gateway ”midt i nettet” for å oversette samtalekontrollmeldinger er den mest åpenbare løsningen for å få til kommunikasjon mellom H.323-terminaler og SIP-terminaler. Med ”midt i nettet” mener jeg at den bygger bro mellom et fullstendig og selvstendig H.323-system på den ene siden, og et fullstendig SIP-system på den andre siden. Vi regner her med H.323-utstyret allerede kan brukes til å ringe til annet H.323-utstyr, og at SIP-utstyr kan brukes mellom SIP-terminaler. Gatewayen blir dermed en bro for å koble disse systemene sammen og muliggjøre samtaler og konferanser der både H.323-terminaler fra H.323-siden deltar og SIP-terminaler fra SIP-siden av gatewayen deltar.

En slik gateway må operere som endesystem (eller proxy) for terminalene som er med i samtalen, og må oversette signaliseringen fra det ene systemet til det andre. Det er grunn til å anta at mediene ikke behøver å konverteres til andre formater.

Utviklingen av en slik gateway vil særlig møte tre typer problemer:

1. Signaliseringsprotokollene er ikke 100% oversettbare. For eksempel kan H.245 uttrykke komplekse sett av preferanser og kapabiliteter for medieformater det ikke er mulig å uttrykke i SDP.
2. Funksjoner fra den ene standarden finnes ikke eller kan ikke etterlignes på standard utstyr i den andre. Et eksempel er ”master” for konferanser i H.323, som ikke finnes i SIP.
3. Under oppsett av samtale utveksles ikke informasjon i samme rekkefølge på begge standarder. H.323 gjør forhandling om medier etter at samtalen er etablert, mens SIP bygger på at initiativtaker sender informasjon om sine preferanser før samtalen etableres.

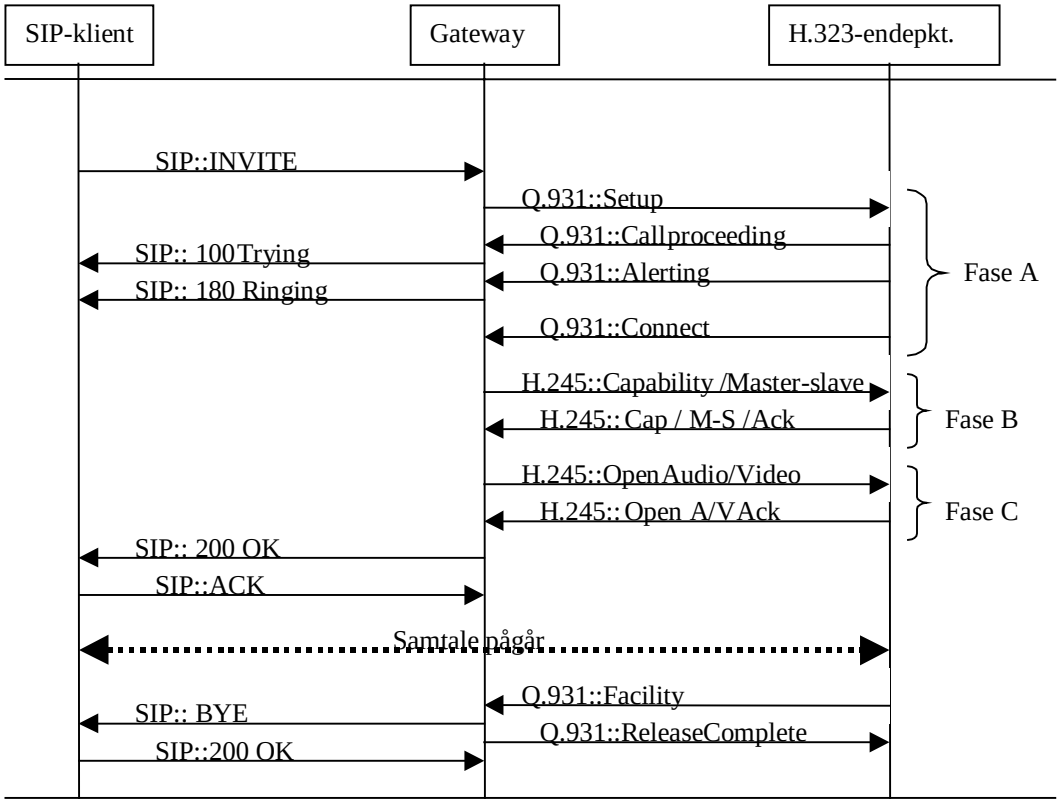
For å få til full interoperasjon mellom SIP og H.323 må en gateway løse alle disse problemene. Enkelte av problemene, f.eks. de komplekse medie-kapabilitetene det er mulig å uttrykke i H.245, skyldes imidlertid muligheter som i praksis ikke benyttes. Det vil dermed være mulig å etablere kommunikasjon mellom SIP- og H.323-terminaler i de aller fleste tilfeller selv om ikke alle problemer er fullstendig eliminert.

Samtale fra SIP-terminal til H.323

Å sette opp en samtale fra en SIP-klient til en H.323-terminal møter få problemer. Signalering foregår i forskjellig rekkefølge, men siden SIP i hovedsak utveksler informasjon tidlig i under oppsettingen, vil gatewayen kunne sitte på informasjonen fra SIP-klienten og sende denne til H.323-terminalen etterhvert.

Figur 8 viser hvordan dette kan gjøres (åpning av logiske kanaler og aksesskontroll med gatekeeper utelatt).Fra denne illustrasjonen av samtaleoppsett kan vi merke oss det følgende:

- Meldingen ”Connect” fra H.323-terminalen ikke kan videresendes (200 OK) før etter at alle forhandlinger er gjort. ACK fra SIP-klienten blir bare brukt til å fortelle gatewayen at samtalen er



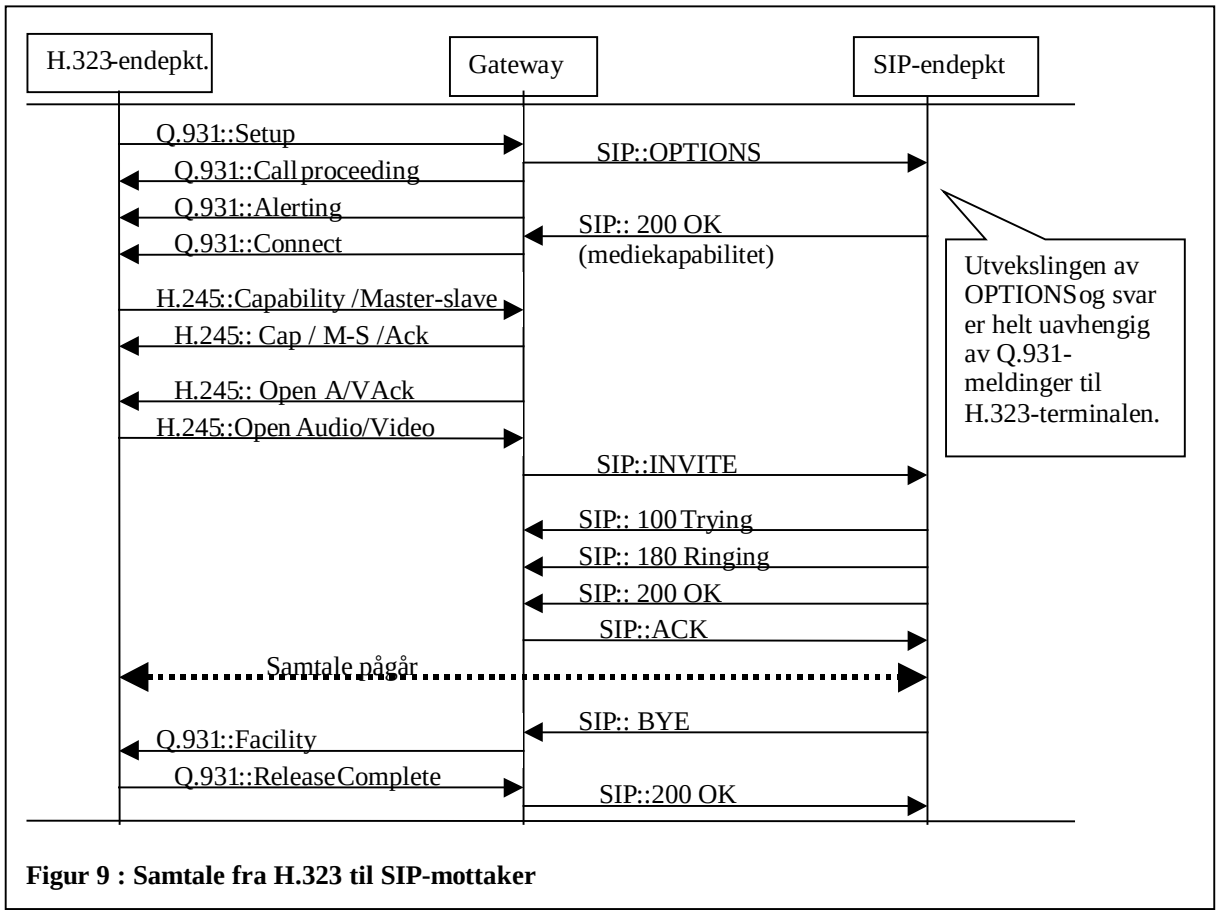
Figur 8: Samtale fra SIP til H.323-system

igang. Å bruke ACK'en til ytterligere medieforhandlinger (slik det er lov i SIP) er helt nytteløst, siden medieforhandlingene må være ferdige før "200 OK" sendes klienten. Hvis medieforhandlingene ikke var løst måtte en feilmelding klasse 4xx, f.eks. "415 Unsupported Media Type" sendes i stedet for "200 OK".

- Gatewayen fungerer som en Proxy på SIP-siden. Autentisering kan fungere som normalt for SIP. Gatewayen vil også være et endesystem på H.323-siden, og vil måtte forholde seg til Gatekeepere på samme måte som terminaler.
- Fase B: Master – Slave determination blir et problem, siden SIP ikke inneholder noe begrep om konferanse-"master". H.323 spesifiserer at Gatewayer skal ha rang over terminaler, så gatewayen må være master. Vi ser på hvilke følger dette gir for konferanser lenger ned. Legg merke til at hvis enten gatewayen eller terminalen bruker Gatekeeper-routed call control vil en Gatekeeper ha rang over gatewayen.
- Oppsettingen av samtalen blir selvfølgelig triviell hvis H.323-terminalen støtter "Fast-start". Da vil all informasjonen fra SIP INVITE-pakken bli sendt i en "jumbo-melding" til H.323-terminalen.
- Det blir Gatewayen som bestemmer hvilke medier som skal utveksles. Den vil oversende SIP-klientens mediekapabilitetssett til H.323-endepunktet, oversatt til H.245. Det finnes medieformat som ikke kan uttrykkes i H.245, men som har standardiserte koder for SDP. Med andre ord kan ikke alt oversettes. Gatewayen vil motta H.323-terminales kapabilitet, oversette denne til SDP og sende den over i 200 OK-meldingen. Siden begrepene for mediekapabilitet i H.245 har større uttrykksmuligheter (se "Forhandlinger" s.29), vil heller ikke alt kunne oversettes denne veien. Dette er imidlertid et større problem når initiativet til samtalen går den andre veien (se neste avsnitt).

Samtale fra H.323 til SIP-terminal

Hvis en SIP-H323-gateway skal formidle en samtale fra en H.323-terminal til en SIP-terminal får vi større problemer med rekkefølgen på datautveksling enn når samtalen går den andre veien. Nå vil SIP-User Agent'en som tar imot INVITE-Requesten fra Gatewayen forvente mer informasjon i den enn det som er utvekslet mellom H.323-terminalen og Gatewayen. Det blir et problem at SIP-User Agent'en ikke kan svare "200 OK" før endepunktene har blitt enige om medier og formater, mens Gatewayen ikke kan få tak i H.323-terminalens medie-ferdigheter før etter at "200 OK" er mottatt og sendt til H.323-terminalen (som Q.931::Connect). Gatewayen får med andre ord ikke kjennskap til SIP-terminalens medieferdigheter tidsnok.



Figur 9 : Samtale fra H.323 til SIP-mottaker

Det finnes to tilnærminger til dette problemet. Ingen er garantert å virke. Den første forsøket på løsning er å utnytte det faktum at SIP bruker "3-way handshake" til opprettelse av samtale, og at alle tre meldinger (også ACK) kan brukes til å utveksle medieferdigheter. Da kan Gatewayen sende INVITE uten noen SDP-melding med H.323-terminalesn medieferdigheter (for det vet jo ikke Gatewayen noe om enda). Når "200 OK" kommer tilbake kan mediene forhandles med H.323-terminalen, ettersom "200 OK" inneholdt SIP-terminalens preferanser/ferdigheter. Til sist kan den endelige beslutningen meddeles SIP-terminalen i ACK-meldingen rett før samtalen startes.

Problemet med denne første løsningen er at SIP bygger på at mediekapabilitet i utgangspunktet skal utveksles i INVITE. Å passe sammen flere parters preferanser for medier er ikke en helt triviell oppgave. Hvis mediekapabiliteten SIP-User Agent'en returnerer i "200 OK" er det subsett av formater i SDP-mediebeskrivelsen i INVITE som støttes, vil denne metoden åpenbart ikke føre frem. "200 OK" vil heller ikke inneholde noen medier eller formater, og samtalen kan bare opprettes hvis ingen medier utveksles (logisk sett vil det være en samtale siden man har ende-til-ende samtalekontroll).

Den andre mulige tilnærmingen til problemet er hvis Gatewayen skaffer seg oversikt over SIP-terminalens mediekapabilitet før samtalen settes opp ved hjelp av OPTIONS-meldingen. Da kan den

late som samtalen blir satt opp overfor H.323-klienten. I virkeligheten sendes ikke INVITE til SIP-User Agent'en før etter at H.323-terminalens mediepreferanser er utvekslet.

Samtale kan da settes opp som i figur 9. Det går an å påpeke at denne løsningen innebærer å lure både H.323-terminalen og dens eier til å tro at samtalen er igang før det er avklart om den i det hele tatt kan opprettes. Hvis evt. tellerskritt eller andre kostnader begynner å løpe på H.323-siden allerede ved "Q.931::Connect" går det an å ha etiske innvendinger mot denne løsningen.

Løsningen forutsetter at Gatewayen avgjør medieforhandlingene mellom de to partene, eller egentlig på vegne av SIP-terminalen, etter å ha fått svar på OPTIONS. Det kan dessuten hende at løsningen ikke virker, ettersom minimumsimplementasjoner av SIP-agenter bare anbefales, ikke kreves, å støtte OPTIONS. Det er heller ikke gitt at H.323-endepunktet vil akseptere at det går lang tid fra den meddeler "la oss starte" og til SIP-klienten faktisk begynner å sende data. Her kan det være snakk om mange sekunders venting, ettersom INVITE må finne den riktige terminalen, varsle brukeren som i sin tur må "ta av røret" osv., før samtalen faktisk kan starte.

H.245-mediekapabilitetsmeldinger som inneholder flere alternative samtidigliste-strukturer kan ikke uten videre oversettes til SDP. Et forslag til å overkomme problemet har vært å først oversette den første samtidig-lista til SDP og prøve INVITE til SIP-endepunktet med denne. Hvis svaret blir "415 Unsupported Media Type" prøves neste. Dette fortsetter til svaret blir noe annet enn 415 eller det ikke er flere samtidig-lister igjen. Dette vil ta noe tid, og forutsetter at H.323-klienten har bygget opp lista i foretrukket rekkefølge. I praksis vil nok dette problemet sjelden oppstå.

Problemer i forbindelse med konferanser

I forbindelse med konferanser kan det oppstå visse vanskeligheter knyttet til de svært forskjellige konferansemodellene i de to standardene. På H.323-siden forhandles det om hvem som skal være "Master" for konferansen (MC). Såfremt ingen Gatekeeper eller MCU er innblandet, vil Gatewayen "vinne" over alle terminaler som måtte være med på denne siden. Vinne betyr å få ansvaret for å koordinere konferansen som MC. Man må imidlertid regne med at alle profesjonelle og/eller kommersielle installasjoner vil bruke Gatekeeper og Gatekeeper-routed call-control. I såfall blir ikke Gatewayen MC.

Alle H.323-terminaler som vil være med i konferanse må spørre MC om lov. Hvis det kommer inn samtaleforsøk til terminaler som er "slaver" (ikke MC), må disse svare at de er i konferanse og oppgi MC'ens adresse. Tilsvarende må alle invitasjoner til nye konferansepartnere gå gjennom MC. MC har ansvaret for å kringkaste til alle deltakende terminaler når en terminal kommer til eller forlater konferansen. Dette vil bli et problem for samvirke med SIP, hvor det ikke finnes noen sentral konferansekontroll.

I en konferanse som passerer gjennom en SIP-H.323-gateway vil det være en MC i en Gatekeeper eller MCU på H.323-siden som forventer å ha full kontroll over konferansen og dens deltakere. Dette gjør at H.323-siden av konferansen kan gå som normalt så lenge det bare en SIP-partner. Problemene oppstår hvis SIP-partneren inviterer flere SIP-deltakere. I følge H.323 må invitasjon av nye deltakere i konferanser gå gjennom MC'en. I følge SIP kan imidlertid hvem som helst invitere hvem som helst når som helst.

Problemene er imidlertid ikke større enn at de antakelig kan løses. Det enkleste er hvis klientene på H.323-siden inviterer de nye partene til samtalen. Da vil MC på en triviell måte forholde seg til Gatewayen som et vanlig H.323-endepunkt, og Gatewayen må "oversette" et vanlig H.323-til-SIP-kall. Nedenfor ser vi på mulige løsninger hvis klientene på SIP-siden begynner å invitere nye parter.

Konferanse sentralisert med MCU på H.323-siden

I en sentralisert konferanse har alle punkt-til-punkt-samtale med MCUen som er Master i konferansen, også eventuelle SIP-klienter, siden Gatewayen med stor sannsynlighet bare behøver å oversette samtalekontrollmeldinger. I SIP er det i utgangspunktet to måter å lage konferanser på. Den

ene er ved bruk av headeren ”Also” i utvidelsen SIP-Call-control. Den andre er å overføre data og signalering til multicast. I begge tilfeller vil Gatewayen kunne fungere som Proxy mot MCUen. Bruk av Also vil føre til at konferansen blir punkt-til-punkt mellom alle parter på SIP-siden, men MCUen og H.323-klientene vil ikke affekteres av dette. I multicast-konferanser blir det imidlertid ikke så lett for Gatewayen å avvise terminaler på SIP-siden hvis Master på H.323-siden (MCUen) bestemmer at de ikke får være med.

Hvis en terminal blir invitert av en SIP-klient med Also-metoden (Also: gateway-id, sip-term1, sip-term2…), vil den sende INVITE til Gatewayen med samme Call-ID som resten av konferansen. Gatewayen vil oversette dette til ”Setup” på vegne av SIP-terminalen (men med en adresse i Gatewayen som avsender, såklart). Den nye terminalen blir en ny terminal på H.323-siden, representert ved en port i Gatewayen til Call-control.

Ved multicast call-control på SIP-siden kan Gatewayen fungere som proxy mot MCUen. Her må Gatewayen enten opprette en port (Proxy-adresse) for hver SIP-klient eller virke som en samordnende enhet for alle SIP-klienter. I det siste tilfellet vil MCUen oppfatte Gatewayen som en enkelt terminal som stadig ber om reforhandlinger av antall parallelle mediestrømmer den sender og mottar. I det første tilfellet (Proxy-per-SIP-terminal) vil den oppfattes som ulike terminaler. Det siste vil nok være enklest på H.323-siden, og skal være mulig med ”tag”-tillegget i headerene To og From på SIP-siden.

Konferanser som er desentralisert på H.323-siden

Desentraliserte konferanser på H.323-siden har en bestemt MC, antakelig en Gatekeeper, men sender i utgangspunktet alle mediedata på multicast. Uten å ha studert dette nærmere antar vi (fortsatt) at SIP-terminaler kan motta mediepakkene som sendes av H.323-terminaler og vice versa. Gatewayen må da kunne oversette signalering (som over). Dette kan antakelig enkelt gjøres med Proxy-per-SIP-terminal (som over).

SIP-terminal i en Gatekeeper-Zone

Å bruke en SIP-terminal på et nett hvor alt resten følger H.323 vil kreve en gateway mellom SIP-terminalen og Gatekeeper. For samtaler vil denne bli veldig lik den som er skissert i forrige avsnitt. En viktig forskjell er at den må håndtere oversettelse av meldinger som utveksles mellom lokal Gatekeeper og terminal. Dette vil særlig si meldinger for å registrere terminalen/brukeren på terminalen.

Når en person logger inn på en SIP-terminal vil det normalt bli sendt en REGISTER-melding til den lokale SIP-registrar (sannsynligvis domenets SIP-server). En SIP-terminal på et H.323-nett må sende denne til gatewayen, som omgjør dette til en RAS/RRQ (Register request) som sendes Gatekeeper. Hvilken adresse som skal settes i denne meldingen er ikke umiddelbart klart. Det må være en adresse Gatekeeper kan forholde seg til, og H.323 har langt strengere forhold til adresser enn SIP. SIP tillater jo mailto og alle andre URIer som adresser. Adressen må være et endepunkt i Gatewayen, og det blir opp til Gatewayen å forholde seg til problemene hvis adressen på SIP-siden er en URL for en annen tjenestetype enn SIP (f.eks. mailto:).

Hvis gatewayen fungerer som bro for bare én SIP-terminal kan den bruke seg selv som H.323-adresse for invitasjoner til brukeren som sendte REGISTER. Hvis den skal være bro for flere trengs en måte å skille hvilken terminal som forsøkes nådd når det kommer en samtale inn. Her er det mulig å benytte forskjellig portnummer for de forskjellige terminalene, siden RRQ skal inneholde transportadresse, ikke nettverksadresse.

Hvis SIP-terminalen bare kan kommunisere med andre terminaler gjennom gatewayen, vil problemene med konferanser og master-slave-forhold i forrige avsnitt kunne overkommes. Det vil ikke være mulig for SIP-klienter å snakke med hverandre uten at MC’en får vite om det.

I tillegg til registrering må RAS-meldinger for aksesskontroll og båndbreddekontroll oversettes. Det siste blir vanskelig, ettersom det eksplisitt står i SIP-standarden at båndbreddereservasjon er utenfor standarden, det vil si opp til de applikasjonene som står for utveksling av mediedata. Disse vil, slik det ser ut nå, kanskje basere seg på RSVP. Gatewayen vil ikke få forespørsler om båndbredde fra klienten, men Gatekeeper vil forvente at alle båndbredde klienten trenger blir reservert ved forespørsler til Gatekeeper (gjennom gatewayen). Så enten må gatewayen bestille masse båndbredde på eget initiativ. Det går an, for den har jo oversikt over hvilke medier som skal utveksles. Eller så må RSVP-systemet og Gatekeeper settes opp så de kan samarbeide. H.323 antar at gatekeepere vil bruke RSVP til resursreservering, så dette vil nok være mulig.

H.323 terminal i et SIP-domene

Å benytte en H.323-terminal i et SIP-miljø vil kreve en gateway for å oversette signalering. Gatewayen vil fungere som en terminal eller proxy i SIP-systemet. Hvis H.323-terminalen er laget for å fungere sammen en Gatekeeper må det lages en entitet til å ta imot RAS-meldinger. Det kan nok være lurt å legge denne funksjonen inn i gatewayen også. Registrering vil da kunne gå enkelt. Forespørsler om båndbreddereservasjon skal ikke oversettes, men må besvares. Å svare ja på alle båndbredde-forespørsler er imidlertid tillatt, så disse funksjonene vil nok ikke være vanskelige å håndtere.

Legg merke til at denne kombinasjonen vil kunne tillate Registrering (flytting) av terminal i hele SIP-domenet (altså hele internett). Men det spørs om H.323-klienten er i stand til å produsere en slik RRQ-melding.

V. Konklusjoner

Vi ser at SIP og H.323 tilbyr den samme grunnleggende funksjonalitet og stort sett de samme tjenestene. Alle vanlige moderne telefontjenester støttes.

H.323 støttes opp om og leveres av mange tunge aktører innen telekommunikasjon. På bakgrunn av deres lange erfaring innen telefoni, har de også etablerte mekanismer for å ta betalt for samtaler og tjenester. Disse aktørene har antakelig den tilstrekkelige tyngden i markedet som skal til for at H.323 skal bli en etablert og utbredt standard.

SIP tilbyr større fleksibilitet og muligheter for integrasjon av applikasjoner og tjenester på klientsiden. SIP gjør stor nytte av eksisterende tjenester i internett, som DNS til adresseoppslag.

Begge standardene fokuserer på å ta i bruk kommende internett-tjenester og –standarder for å oppnå QoS. Det er forholdsvis sterke sikkerhetsmekanismer i begge systemene.

Per dags dato er det bare H.323 som har implementert standardiserte løsninger for interoperasjon med det eksisterende telefonsystemet. Det finnes imidlertid skisser for hvordan tilsvarende løsninger kan lages for SIP, dessuten er det en egen arbeidsgruppe i IETF som jobber med interoperasjon mellom internett og telefonsystemet.

Interoperasjon mellom de to standardene er mulig. Dette kan antakelig gjøres med å lage en gateway mellom de to systemene. Siden SIP-systemer utveksler mere informasjon i begynnelsen av samtaleoppsettet, er det lettere å sette opp samtaler fra SIP-klienter til H.323-terminaler enn den andre veien. Oppsett av samtale fra H.323-terminal til SIP-terminal må ”lure” H.323-siden til å tro at samtalen er igang før den i det hele tatt er initiert på SIP-siden. Dette for å få tak i tilstrekkelig informasjon om medier og adresser til å kunne initiere SIP-siden av samtalen.

På grunn av veldig forskjellige begrep om konferanser i de to standardene, blir det veldig vanskelig, men antakelig mulig, å lage Gatewayer som gir mulighet for konferanser mellom SIP-terminaler og H.323-terminaler.

VI. Referanser

[1] H.323 standard: Draft ITU-T Recommendation H.323V2 pr. 5. Aug.-97

[2] H.245 standard: 24 mars 1997

[3] H.235 standard: Draft H.Secure March 1997

[4] HTTP/1.1 : Hypertext Transfer Protocol v1.1, IETF RFC 2068, *T. Berners-Lee m.fl., Januar 1997*

[5] SDP : Session Description Protocol, IETF RFC 2327, *M. Handley, V.Jacobson, April 1998*

[6] Telefon-URL: draft-antti-telephony-url-04.txt (work in progress), *A. Vaha-Sipila, 23. Feb. 1998,* fra http://www.ietf.org/internet-drafts/

[7] SIP : Session Initiation Protocol: draft-ietf-mmusic-sip-09.ps (work in progress), *Handley, Schulzrinne, Schooler, Rosenberg,* fra http://www.ietf.org/internet-drafts/

[8] SIP Call Controll Services: (utvidelsen "org.ietf.sip.cc", work in progress), *Schulzrinne, Rosenberg,* draft-ietf-mmusic-sip-cc-00.txt fra http://www.ietf.org/internet-drafts/

[9] H.323-primer fra Databeam (http://www.databeam.com)

[10] e-post-diskusjoner på confctrl@ISI.EDU

[11] SIP-info-sider: http://www.columbia.edu/~hgs/sip

[12] RTP : Real Time Protocol (oppdatering av RFC 1890, work in progress), *Schulzrinne m.fl. 5. Desember 1997*

[13] Cristina Aurrecoechea, Andrew Campbell, Linda Hauw, *A Survey of Quality of Service Architectures,* i *ACM/Springer Verlag Multimedia Systems Journal, Vol. 6 nr. 3.*

[14] RSVP : Resource Reservation Protocol, IETF RFC 2205, *R. Branden m.fl. September 1997.*

[15] PINT: PSTN-Internet Interworking: draft-ietf-pint-inweb-00.txt (work in progress) *M.Krishnaswamy, November 1997.*

[16] RTSP: Real Time Streaming Protocol, IETF RFC 2326, *Schulzrinne/Rao/Lanphier, April 1998*

[17] *Signaling for Internet Telephony,* Henning Schulzrinne and Jonathan Rosenberg, Columbia University Technical Report CUCS-005-98; submitted for publication, January 1998.

[18] *ImiS-Ericsson* NR-rapport nr. 926, Peter D. Holmes m.fl., Mai 1998

[19] *Requirements for SIP Servers and Agents,* Schulzrinne og MCI, 7. Mars 1998

[20] *A Comparison of SIP and H.323 for Internet Telephony,* Schulzrinne & Rosenberg, Network and Operating System Support for Digital Audio and Video (NOSSDAV), (Cambridge, England), July 1998

[21] APC-1382 om endringer i Gatekeeper discovery-prosedyrer, Innspill fra Nokia til ITU-Study Group 16-møtet Cannes i juni-98.

Bilder/illustrasjoner hentet fra andre kilder:

Figur 1 og 2 : Fra [11].
Figur 3, 5 og 6 : SIP-info-sider: http://www.columbia.edu/~hgs/sip (5 og 6 også i [7])
Figur 4 laget selv etter figur 3 og etter figur 21 i [1].
Figur 7 fra [1].
Figur 8 og 9 laget selv.