FFI-rapport 2013/03081

SIP realisert i Asterisk – teori og eksperiment

Bodil Hvesser Farsund og Anne Pernille Hveem

FFI Forsvarets forskningsinstitutt

SIP realisert i Asterisk – teori og eksperiment

Bodil Hvesser Farsund og Anne Pernille Hveem

Forsvarets forskningsinstitutt (FFI)

28. februar 2014

FFI-rapport 2013/03081

1242

P: ISBN 978-82-464-2348-7
E: ISBN 978-82-464-2349-4

Emneord

Voice over IP

Asterisk

Session Initiation Protocol

Real-time Transport Protocol

Godkjent av

Kjell Olav Nystuen	Prosjektleder
Anders Eggen	Avdelingssjef

Sammendrag

Utviklingen innen offentlig elektronisk kommunikasjon gjennomgår i dag en rivende utvikling, samtidig som trenden er at militære systemer bruker de samme teknologiene som sivile systemer. Det har derfor vært interessant for Forsvaret å se på noen av disse nye kommunikasjonsteknologiene. På bakgrunn av dette har det ved Forsvarets forskningsinstitutt blitt gjort studier av noen utvalgte relevante sivile systemer og teknologier for elektronisk kommunikasjon. I denne rapporten beskrives Voice over Internet Protocol (VoIP) med vekt på sikkerhet.

VoIP er tale- og annen multimediakommunikasjon over IP-baserte nett, det vil si over nettverk som er pakkesvitsjet. Vi har satt opp en egen lab, hvor vi har sett på hvordan de mest brukte protokollene for VoIP, Session Initiation Protocol (SIP) og Real-time Transport Protocol (RTP), har fungert ved bruk av tale. Både VoIP-serverne og VoIP-klientene har vært satt opp med gratis programvare. VoIP-serverne benyttet Asterisk programvare.

Vi har først og fremst sett på hvordan VoIP-klienter registrerer seg, og hvordan en talesesjon blir satt opp og avsluttet i flere ulike eksperimenter. Disse eksperimentene har bestått i at:

- VoIP-klientene var registrert hos samme VoIP-server
- VoIP-klientene var registrert hos hver sin VoIP-server og kommuniserte direkte ved hjelp av SIP Unified Resource Identifier (URI)
- VoIP-klientene var registrert hos hver sin VoIP-server og kommuniserte via en ekstern VoIP-tjenesteleverandør
- en VoIP-klient kommuniserte med en Public Switched Telephone Network (PSTN) - eller Global System for Mobile Communication (GSM) -telefon via en ekstern VoIP-tjenesteleverandør

Voip-klientene har i eksperimentene vært koblet opp mot serverne både via internt ”Wireless Local Area Network” (WLAN), eksternt WLAN og Universal Mobile Telecommunications System (UMTS).

Hovedkonklusjonen er at SIP- og RTP-trafikken stort sett følger samme mønster uavhengig av eksperiment og uavhengig av hvordan de er koblet opp mot sin VoIP-server. Trafikken er også som forventet ut fra hva man ser i litteraturen.

Gjennom eksperimentene fikk vi bekreftet en sårbarhet i SIP, nemlig at det ikke skjer noen autentisering dersom en klient prøver å endre på sesjonen underveis, ei heller om han prøver å avslutte sesjonen. Dette er klare sårbarheter i SIP. Det skjer heller ingen autentisering av oppringeren dersom mottakeren kontaktes direkte ved hjelp av SIP URI.

English summary

Today commercial electronic communications are developing fast, at the same time the tendency is that commercial and military systems use the same technologies. Therefore the Norwegian Armed Forces has been interested in taking a closer look at some of these new communications systems. Based on this, the Norwegian Defence Research Establishment has studied some selected relevant commercial systems and technologies for electronic communication. In this report Voice over Internet Protocol (VoIP) is described with focus on security.

VoIP is voice communication or other multimedia sessions over IP based networks, which means networks that are packet-switched. We have set up our own lab, where we have looked at how the most used protocols for VoIP, Session Initiation Protocol (SIP) and Real-time Transport Protocol (RTP) work in the case of voice communication. Both the VoIP-servers and the VoIP-clients have used VoIP-freeware. The VoIP-servers used the Asterisk software.

We have focused on how the VoIP-clients register with the VoIP-servers, and how the voice sessions are being started and terminated in different experimental setups. The different experimental setups were:

- The VoIP-clients were registered with the same server
- The VoIP-clients were registered with different VoIP-servers and were communicating directly using SIP Unified Resource Identifier (URI)
- The VoIP-clients were registered with different VoIP-servers and were communicating using an external VoIP service provider
- A VoIP-client was communicating with a Public Switched Telephone Network (PSTN) or Global System for Mobile Communication (GSM) phone via an external VoIP service provider

The VoIP-clients have been connected to the servers in different ways in the experiments. They have been connected using internal “Wireless Local Area Network” (WLAN), external WLAN and ”Universal Mobile Telecommunications System” (UMTS).

The main conclusion is that the SIP and RTP traffic is mostly independent of the experiments and how the clients are connected to the servers. The traffic is also mostly as expected according to examples found in the literature.

The experiments verified that there was no authentication of the client if he tries to modify session parameters during the session or end the session. This is a vulnerability in SIP. There is neither any authentication of the caller client if he uses the SIP URI when he contacts the receiver.

Innhold

1 Innledning

Det pågår i dag en omfattende utvikling innen offentlig elektronisk kommunikasjon. En klar trend er at Voice over Internet Protocol (VoIP) er i ferd med å ta over for linjesvitsjet telefoni, som har vært benyttet i blant annet Public Switched Telephone Network (PSTN) og Global System for Mobile communication (GSM). VoIP er tale og annen multimediakommunikasjon over Internet Protocol (IP), det vil si over pakkebaserte nett.

Da de militære og sivile systemene for elektronisk kommunikasjon i stadig større grad benytter de samme teknologiene, har det vært interessant for Forsvaret å se nærmere på noen av disse nye sivile systemene. Dette er bakgrunnen for at Forsvarets forskningsinstitutt har valgt å se nærmere på blant annet VoIP. Tidligere har det vært utgitt rapporter som omhandler 4G-teknologiene WiMAX [1] og Long Term Evolution (LTE) [2], samt en teoretisk rapport som omhandler VoIP med fokus på sikkerhet [3].

For å finne ut hvordan VoIP fungerer i praksis, har vi i satt opp en egen VoIP-lab. VoIP kan realiseres med flere ulike protokoller. Blant de åpne standardene kan nevnes ”Session Initiation Protocol” (SIP) i kombinasjon med ”Real-time Transport Protocol” (RTP) samt H.323 [4]. Det finnes også flere proprietære protokoller med høy utbredelse, der kjente eksempler er Skype [5] og Google Hangouts [6] (tidligere Google Talk). Vi har valgt å se nærmere på signaleringsprotokollen, SIP, og transportprotokollen, RTP, da disse er de mest brukte av de åpne standardene.

For å sette opp et SIP- og RTP-basert VoIP-system, har vi benyttet Asterisk [7]. Asterisk fungerer som en telefonsentral, eller såkalt ”Private Branch eXchange” (PBX) i programvare. Den formidler tale, video og meldinger mellom to eller flere klienter. Den kan også koble seg opp mot andre telefonitjenester som blant annet PSTN og andre VoIP-systemer. Den finnes både i en gratis-versjon som vi benyttet, og i en kommersiell versjon med kundestøtte.

1.1 Formålet med rapporten

Formålet med denne rapporten er å dokumentere hvordan SIP- og RTP-protokollene fungerer ved bruk av tale, slik vi har erfart det gjennom våre eksperimenter. VoIP-serverne har vært satt opp som dokumentert i [8], og vi har utført ulike eksperimenter. De ulike eksperimentene omfatter følgende:

- VoIP-klientene var registrert hos samme VoIP-server
- VoIP-klientene var registrert hos hver sin VoIP-server og kommuniserte direkte ved hjelp av SIP ”Unified Resource Identifier” (URI)
- VoIP-klientene var registrert hos hver sin VoIP-server og kommuniserte via en ekstern VoIP-tjenesteleverandør
- en VoIP-klient kommuniserte med en PSTN- eller GSM-telefon via en ekstern VoIP-tjenesteleverandør

VoIP-klientene var i eksperimentene koblet opp mot serverne både via internt ”Wireless Local Area Network” (WLAN), eksternt WLAN og UMTS.

1.2 Rapportens oppbygging

I første del av rapporten blir SIP- og RTP-protokollene kort beskrevet. Dette er gjort i kapittel 2. I kapittel 3 beskrives lab-oppsettet, mens kapittel 4 beskriver den SIP- og RTP-trafikken vi så i de ulike eksperimentene. Til slutt følger oppsummering og konklusjon i kapittel 5.

2 Kort om Session Initiation Protocol (SIP) og Real-time Transport Protocol (RTP)

Som allerede nevnt, er den mest brukte åpne signaleringsprotokollen for VoIP og andre multimediasesjoner i dag “Session Initiation Protocol” (SIP), mens den mest brukte åpne protokollen for transport av tale og video er ”Real-time Transport Protocol” (RTP). SIP og RTP har blitt ”de facto” industristandard for VoIP. Nedenfor følger en kort beskrivelse av protokollene. Innholdet i kapittelet er hovedsakelig hentet fra [9-13].

2.1 Session Initiation Protocol (SIP)

”Session Initiation Protocol” (SIP) er designet og standardisert av ”Internet Engineering Task Force” (IETF). SIP er en signaleringsprotokoll på applikasjonslaget som blir brukt for å sette opp, endre og avslutte multimediasesjoner. Sesjonene kan bestå av en eller flere mediastrømmer mellom to eller flere parter.

SIP er bygget opp rundt en forespørsel- og responsmodell, hvor den ene siden sender en meldingsforespørsel og den andre siden responderer på forespørselen. Hver SIP-enhet har mulighet til å gjøre begge deler avhengig av hvilken enhet som initierer utvekslingen. SIP-meldingene består av ren tekst som gjør at de kan leses av utenforstående. SIP kan bruke både transportprotokollen ”User Datagram Protocol” (UDP) og ”Transmission Control Protocol” (TCP), men bruker vanligvis UDP.

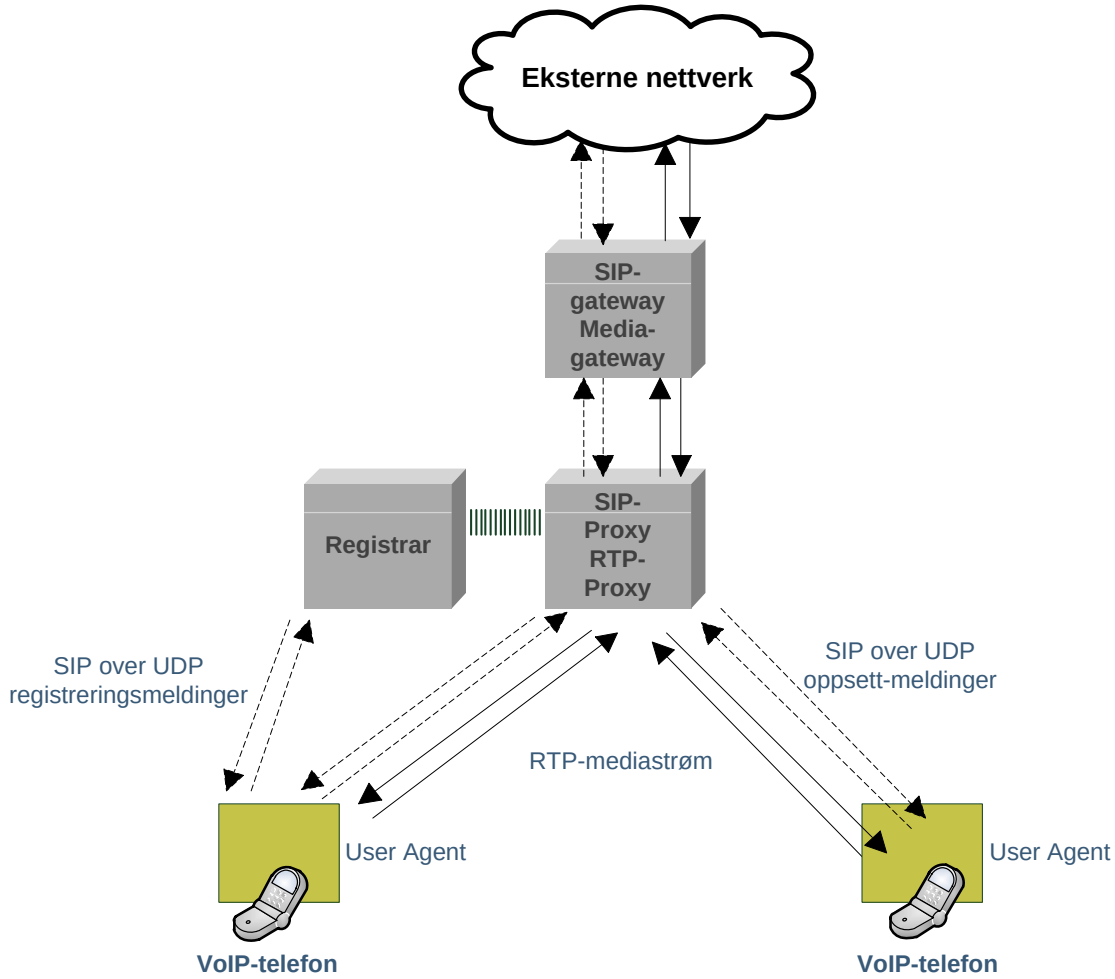
SIP blir ofte beskrevet som ”peer-to-peer”, fordi to SIP-endepunkter kan kommunisere uten mellomliggende infrastruktur. Det er allikevel ikke vanlig at SIP-endepunkter kommuniserer peer-to-peer av forskjellige årsaker. En av årsakene er at det da er mer komplisert for tjenestetilbyder å ta betalt for tjenesten. Det er også sjeldent at den som ringer kjenner IP-adressen til mottakeren, og denne kan også endres ved mobilitet. Det er derfor vanlig å bruke en SIP-proxyserver for å rute signaleringen.

2.1.1 SIP-arkitekturen

En SIP-arkitektur består av disse logiske komponentene:

- User Agent (UA)
- SIP-registrar
- SIP-proxyserver
- SIP-gateway

I Figur 2.1 ser vi et eksempel på en SIP-arkitektur. VoIP-klienter kan ringe hverandre lokalt via en SIP- og RTP-proxyserver. Når de ringer eksterne nummer går dette også via en SIP- og media-gateway. RTP-proxyserver og media-gateway er nærmere forklart i delkapittel 2.2. De stiplede linjene viser signaleringen og de heltrukne linjene viser mediatrafikken.



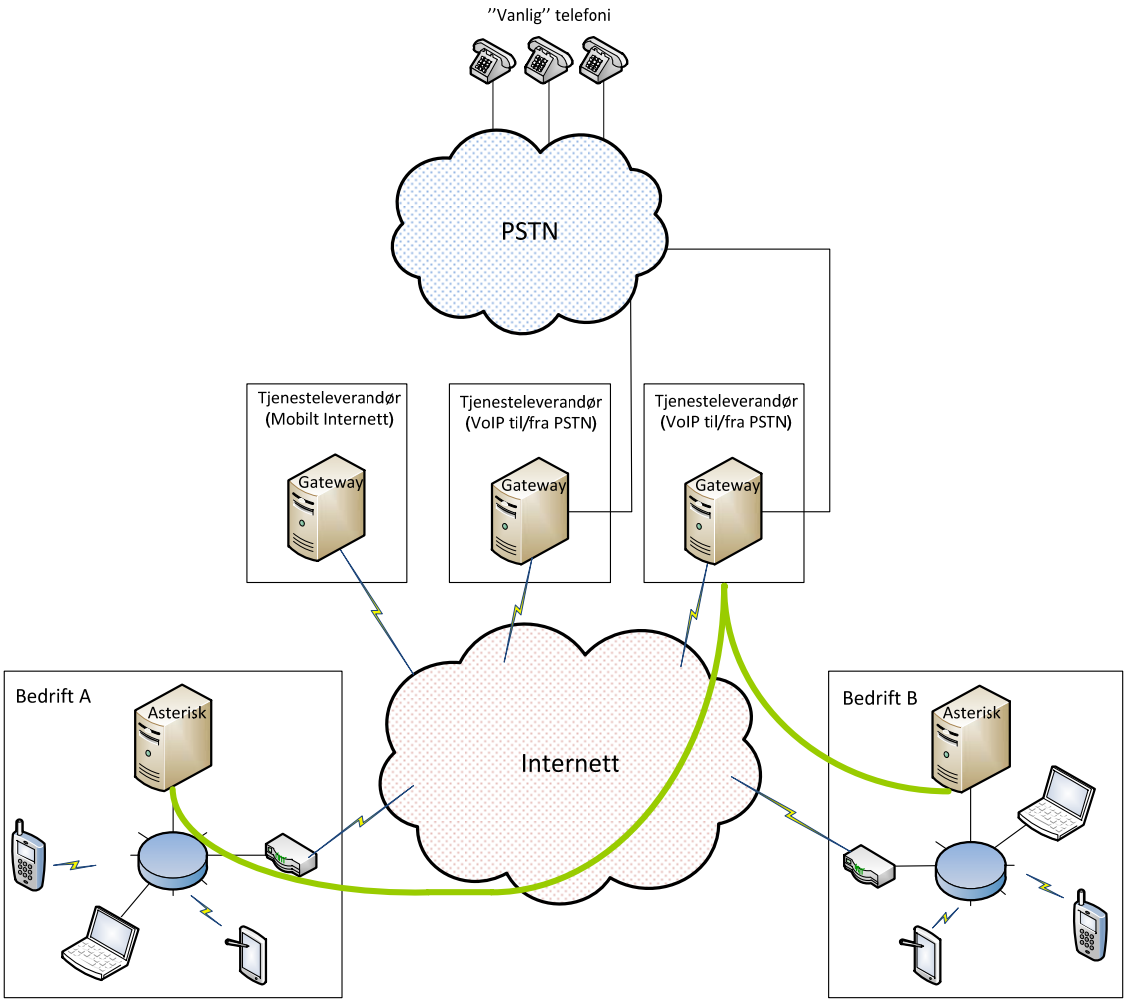
Figur 2.1 Eksempel på en SIP-arkitektur.

En UA er en logisk komponent i en SIP-sesjon som kan sende og motta SIP-meldinger. En SIP UA kan da enten være i rollen som en ”User Agent Client” (UAC) som sender SIP-forespørsler, eller en ”User Agent Server” (UAS) som mottar SIP-forespørsler og returnerer SIP-responser. En VoIP-klient vil inneholde en UA som i en SIP-sesjon vil sende eller motta SIP-forespørsler.

Bruker autentiserer og registrerer sin identitet og lokasjon hos SIP-registrar. Dette skjer når VoIP-klienten slås på, skifter IP-adresse eller nettverk og ved jevne tidsintervall, typisk hvert 5-60 min. SIP-registrar er vanligvis integrert i en SIP-proxyserver, noe som er tilfellet for Astersk-serveren vi har benyttet. SIP-proxyserver ruter og videresender samtaleforespørsler og returnerer responser. En SIP-proxyserver kan derfor ses på som en ruter for signaleringstrafikken. SIP-proxyserveren kan også operere som en SIP-gateway. En SIP-gateway sjekker og ruter SIP-meldinger ut og inn fra et domene. Et domene er en fysisk eller logisk avgrensning, gjerne ett bestemt nettverk eller en bedrift.

SIP-arkitekturen gir et sett med byggeklosser med ulik funksjonalitet. En systemarkitekt kan bygge sin egen installasjon basert på brukerens krav og ønsker, hvor de viktigste momentene er størrelse, sikkerhet og skalerbarhet. SIP-baserte VoIP-installasjoner kan derfor være ulike.

I Figur 2.2 vises et eksempel på et VoIP-basert bedriftssystem. Bedrift A og Bedrift B har installert Asterisk på hver sin fysiske server. Som tidligere nevnt er Asterisk en software-implementasjon av en bedrifts telefonsentral som vi også har benyttet i vårt laboppsett. Asterisk tar seg av både SIP- og RTP-funksjonalitet. VoIP-trafikk som ikke er lokalt mellom ansatte, kan rutes til en VoIP-tjenesteleverandør som bedriftene har en avtale med og som håndterer alle eksterne telefonsamtaler. I dette eksempelet har Bedrift A og Bedrift B samme VoIP-tjenesteleverandør, slik at samtaler som går mellom disse bedriftene vil gå innad i tjenesteleverandørens nett. Hvis ikke telefonnummeret finnes hos tjenesteleverandøren, rutes den til riktig leverandør som har dette telefonnummeret, for eksempel en GSM- eller PSTN-leverandør eller en annen VoIP-leverandør. Det er også mulig for en ansatt i Bedrift A å kontakte en ansatt i Bedrift B direkte hvis han vet IP-adressen til serveren i Bedrift B.



Figur 2.2 Eksempel på et SIP-basert bedriftstelefonsystem.

2.1.2 SIP-adressering

En SIP-bruker adresseres ved hjelp av en SIP URI i SIP-meldingen. SIP kan bruke både tall og tekst som ”telefonnummer”. En typisk SIP URI kan derfor være sip:22123456@servers-IP-adresse eller sip:bob@servers-IP-adresse med “brukernavn@domenenavn” som ved epost. “servers-IP-adresse” er her IP-adressen til brukerens VoIP-server. Adressering ved bruk av tekst har ikke tatt av i industrien, så det er i dag mest vanlig å bruke tall som ”telefonnummer”.

Når for eksempel en VoIP-klient i Bedrift A i Figur 2.2 taster telefonnummeret “22123456”, vil SIP-meldingen først rutes til serveren i Bedrift A, og SIP URI i SIP-meldingen vil være sip:22123456@BedriftAservers-IP-adresse. Dersom dette er et lokalt nummer hos Bedrift A, vil den lokale serveren lage en ny SIP-melding som den sender til mottaker, og SIP URI i den nye SIP-meldingen vil da være sip:22123456@mottakers-IP-adresse. Hvis det er et eksternt nummer, vil Bedrift A sin server lage en ny SIP-melding med SIP URI sip:22123456@voip-tjenesteleverandørens-IP-adresse, som den sender til VoIP-tjenesteleverandøren. VoIP-tjenesteleverandøren vil på samme måte rute SIP-meldingen videre til mottaker.

Hvis en VoIP-klient kjenner IP-adressen til mottakeren sin server, kan han kontakte mottakeren direkte uten å bruke noen VoIP-tjenesteleverandør. Istedenfor å taste for eksempel “22123456”, skrives “22123456@mottakersserver-IP-adresse” direkte inn i VoIP-klienten.

2.1.3 SIP-meldinger

SIP-meldingene som UA’ene utveksler blir delt inn i forespørsler og responser. De vanligste SIP-meldingsforespørslene er:

- REGISTER – registrerer eller avregistrerer identitet og lokasjon til bruker
- INVITE/re-INVITE – henholdsvis initierer eller reforhandler en multimediasesjon
- ACK – bekrefter å ha mottatt en endelig SIP-respons
- BYE – avslutter en multimediasesjon
- CANCEL – kansellerer en forespørsel
- SUBSCRIBE – angir at det ønskes å abonnere på visse typer informasjon som for eksempel varsel om innkommende mail, automatisk tilbakeringing og ”presence information”[1] om gitte brukere.

For å forhandle frem sesjonsparametre i VoIP-sesjonen, benyttes ”Session Description Protocol” (SDP). Eksempler på sesjonsparametre som forhandles frem er mediatype og –format, IP-adresse, portnummer og så videre. SDP blir transportert som nyttelast i SIP-meldingene: “INVITE”, “re-INVITE” og SIP-responsen ”200 OK”.

“re-INVITE”-forespørselen kan bli brukt for å modifisere sesjonsparametre, dialogparametre eller begge deler. Den kan for eksempel endre mediatype, endre IP-adresser eller porter for mottak av mediastrøm, legge til en mediastrøm eller fjerne en mediastrøm. I en konferansesamtale vil deltagere dermed kunne komme til eller forlate samtalen underveis. Re-INVITE kan også bli brukt for å sette den opprinnelige samtalen på vent.

En sårbarhet i SIP er at det ikke kreves autentisering av avsender ved en “re-INVITE”-forespørsel. Hvis angriper har tilgang til nettverkstrafikken og SIP er ukryptert kan angriper sende en falsk “re-INVITE”-forespørsel for å endre sesjonen, slik at sikkerheten svekkes (“down grade”-angrep) eller for å omdirigere trafikken i et avlyttingsforsøk.
De vanligste SIP-meldingsresponsene er:

- ”100 Trying” – forespørsel er mottatt og under behandling
- ”180 Ringing” – det ringer hos ønsket UA
- ”200 OK” – forespørselen er mottatt, forstått og akseptert av ønsket UA
- ”401 Unauthorized” – serveren ber om autentisering av en klient
- ”407 Proxy Authentication Required” – serveren ber om autentisering av en mellomliggende server

[1] “Presence information” er en statusindikator som blant annet angir tilgjengeligheten til en bruker.

De fleste responser (2xx, 3xx, 4xx, 5xx, 6xx) er endelige og avslutter den gjeldende SIP-meldingsutvekslingen, mens 1xx responsene er midlertidige.

”401 Unathorized” blir returnert av enheten som behandler SIP-forespørselen, enten en SIP-registrar eller en SIP-proxyserver, til klienten som sendte meldingsforespørselen med krav om å få autentiseringsinformasjon. ”407 Proxy Authentication Required” blir returnert av mellomliggende SIP-proxyservere som ønsker å autentisere forespørselen før de videresender den til destinasjonsserveren. ”407 Proxy Authentication Required” blir brukt for applikasjoner hvor en behøver en autentisering for å få aksess til en kommunikasjonskanal (for eksempel en telefon gateway) heller enn en mottager [14].

2.2 Real Time Transport Protocol (RTP)

”Real Time Transport Protocol” (RTP) er spesifisert av IETF. RTP transporterer datapakker i sanntid og brukes både for tale og video.

I eksemplet på en SIP-arkitektur vist i Figur 2.1, er det to logiske komponenter som håndterer mediastrømmen, RTP-proxyserver og media-gateway. RTP-proxyserver er en mediaserver som videresender mediastrømmen mellom UA’er, mens en media-gateway formidler mediatrafikk mellom domener. En RTP-proxyserver kan operere som en media-gateway. Mediaforbindelsen som SIP-signaleringen setter opp, kan enten gå direkte mellom sender og mottaker (peer-til-peer) eller via RTP-proxyserver og/eller media-gateway. På grunn av krav til operatører om at det skal være mulig å kunne lovlig avlytte en samtale, vil i praksis all kommersiell kommunikasjon gå via RTP-proxyservere og/eller media-gatewayer. Det finnes allikevel noen VoIP-leverandører som omgår disse kravene.

RTP bruker transportprotokollen UDP, da den gir liten forsinkelse. Ulempen er imidlertid at den ikke gir noen garanti for at pakkene som inneholder mediatrafikken blir levert.

RTP-pakkene består av header og nyttelast, hvor header inneholder informasjon om blant annet:

- type nyttelast
- pakkens sekvensnummer
- tidsstempel
- synkroniseringskilde (”Synchronization Source Identifier”, SSRC)
- bidragskilder (”Contributing Source Identifier”,CSRC)

Type nyttelast sier noe om hva som sendes, for eksempel tale eller video, og hvordan denne er kodet. Mottakeren arrangerer pakkene i riktig rekkefølge ved å bruke sekvensnummer, tidsstempel og synkroniseringskilde. SSRC identifiserer kilden til mediastrømmen. CSRC gir informasjon om hver bidragsyter når mediastrømmen er et samlet resultat av flere mediastrømmer. Dette vil for eksempel være tilfellet ved en talekonferanse med flere bidragsytere.

”Real-time Transport Control Protocol” (RTCP) blir brukt sammen med RTP for å sende kontrollmeldinger til deltagere av en RTP-sesjon. RTCP sin hovedfunksjon er å gi informasjon om kvaliteten på mediastrømmen. Om kvaliteten er dårlig, kan denne informasjonen for eksempel brukes til å endre kodek. RTCP kan også inneholde informasjon om avsender av RTP-pakkene.

Det er forskjellige typer RTCP-pakker avhengig av hva slags informasjon som skal sendes, som for eksempel RTCP “Sender Report”, RTCP “Source Description” og RTCP “BYE”. En RTCP-melding kan bestå av bare RTCP “Sender Report” eller den kan i tillegg inneholde RTCP “Source Description”. Når samtalen avsluttes vil den også inneholde RTCP “BYE”. RTCP “Sender Report” gir blant annet informsjon om:

- round-trip forsinkelse
- jitter
- pakketap
- synkroniseringskilde (SSRC)

RTCP “Source Description” gir informasjon om forskjellige parametre relatert til kilden. Følgende type informasjon finnes, men det er bare “CNAME” som er obligatorisk:

- “CNAME” (Canonical Name)
- “NAME” – vanlig navn på kilden
- “EMAIL”
- “PHONE”
- “LOC” – lokasjon

”Canonical Name” (CNAME) er en unik identifikator som ikke endrer seg under en sesjon. CNAME er en binding mellom SSRC og klienten som er konstant. CNAME skal ha formatet ”user@host” eller bare ”host” dersom det ikke er flere brukere av systemet. For å kunne ha mulighet for monitorering av en tredje-part, skal et program eller en person kunne lokalisere klienten basert på CNAME.

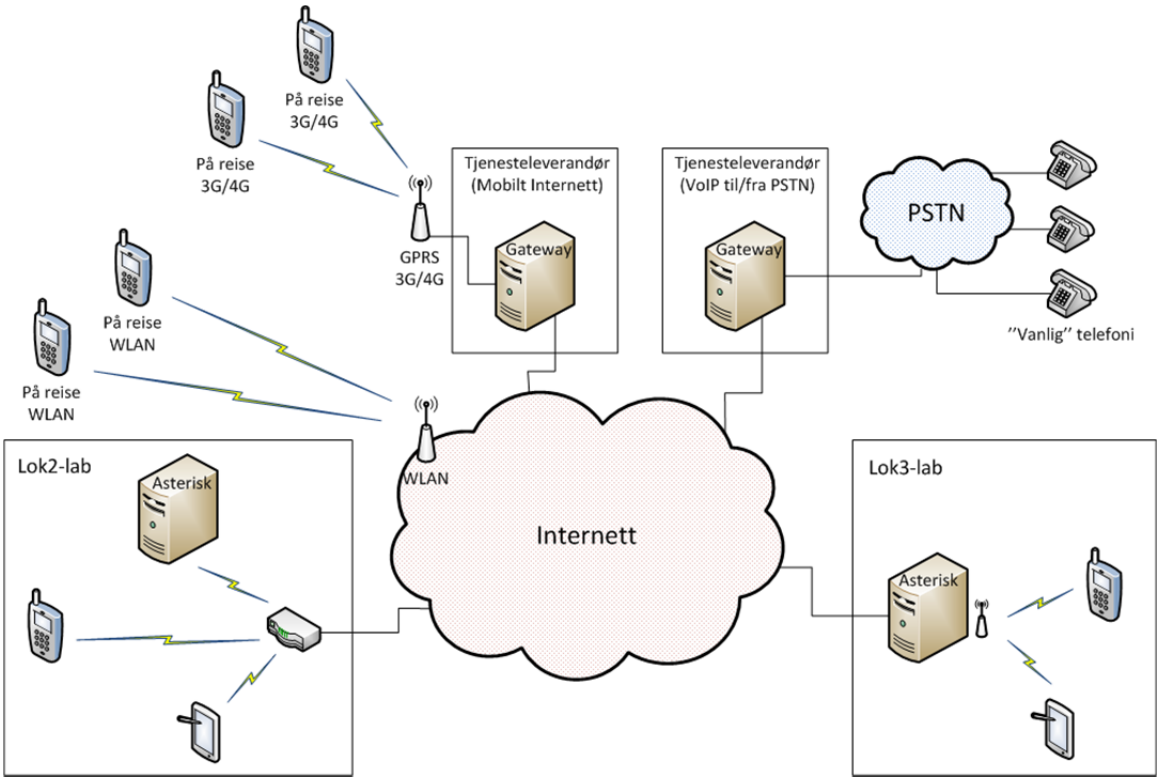
RTP blir tilegnet en liketalls UDP-port mellom klient og server, mens RTCP blir tilegnet den neste oddetalls UDP-porten. Det vil si at hver deltager av en RTP-sesjon bruker minst to UDP-porter hver i området 1024 til 65535.

3 Lab-oppsett

Lab-oppsettet har vært forsøkt gjort så enkelt som mulig. Det har bestått av to VoIP-servere som har vært koblet opp mot en ekstern VoIP-tjenesteleverandør for å kunne ringe inn i PSTN- og GSM-nettet, samt til enheter tilknyttet andre servere. Klientene har vært smarttelefoner og nettbrett. All programvare på servere og klienter har vært gratis, men å koble seg til en ekstern VoIP-tjenesteleverandør måtte vi betale for.

For å se på SIP- og RTP-trafikken ble programvaren ”Wireshark” benyttet. Wireshark er en åpen og gratis pakke-analysator som brukes for å se på datatrafikk [15].

Laboppsettet er vist i Figur 3.1. Nedenfor følger en kort oppsummering av oppsettet.



Figur 3.1 Oversikt over laboppsettet.

3.1 Asterisk-serverne

VoIP-serverne som ble brukt var to datamaskiner hvor vi installerte Debian 6 (squeeze) og Asterisk (versjon 1.6.2.9-2). Disse ble lastet ned gratis fra Internett. Se [8] for nærmere beskrivelse av hvordan dette ble gjort. Som tidligere nevnt inneholder en SIP-arkitektur både en SIP-registrar og en SIP-proxyserver. I vårt tilfelle er SIP-registrar og SIP-proxyserver samme maskin, og blir siden omtalt som Asterisk-server, VoIP-server eller bare server. Asterisk er opprinnelig designet for Linux som vi har benyttet, men støtter også andre operativsystemer.

For at VoIP-serverne skulle kunne ringe inn i PSTN- og GSM-nettet, var det behov for å koble seg opp mot en ekstern tjenesteleverandør som tilbyr slike tjenester. Det er flere slike tjenesteleverandører å velge blant, og vi benyttet oss av en kalt ”voip.ms” [16]. Vi kunne nå koble VoIP-serverne opp mot en av ”voip.ms” sine servere, og vi ble tildelt telefonnumre som fungerte som gatewayer mellom de tradisjonelle PSTN-nettene og våre servere. Våre klienter kunne nå nås av GSM- og PSTN-brukere og motsatt.

Vi konfigurerte 2 voip-servere som sammen med klienter dannet henholdsvis ”Lok2-lab” og ”Lok3-lab”. De var koblet opp på denne måten:

- Lok2-lab
  - Tilkoblet et WLAN-knutepunkt (Netgear WNR 2200), hvor knutepunktet hadde ekstern IP-adresse 193.156.31.128 og intern IP-adresse 10.0.0.1.
  - Koblet opp mot ”voip.ms” sin server ”newyork.voip.ms”
  - Tilkoblet PSTN med USA-nummeret +1-703-996-4512
  - Wireshark installert på samme maskin som Asterisk-serveren med intern IP-adresse 10.0.0.2
- Lok3-lab
  - Eget WLAN-knutepunkt på samme maskin som Asterisk-serveren med ekstern IP-adresse 193.156.31.129 og intern IP-adresse 172.30.201.1.
  - Koblet opp mot ”voip.ms” sin server ”london.voip.ms”
  - Tilkoblet PSTN med det norske nummeret +47 67 20 93 07
  - Wireshark installert på samme maskin som WLAN-knutepunktet og Asterisk-serveren

På grunn av oppsettet, var det på Lok2-lab bare mulig å logge trafikken som gikk til og fra Asterisk- serveren på internt WLAN. På Lok3-Lab derimot var det mulig å logge trafikk både til og fra Asterisk-serveren på internt og eksternt WLAN, samt trafikk rutet til WLAN-knutepunktet både via knutepunktets eksterne og interne IP-adresse.

3.2 VoIP-klienter

VoIP-klientene vi brukte var følgende:

- Iphone 4 (smarttelefon) med operativsystemet ”iOS” og SIP-applikasjonen ”Media5-fone”.
- Samsung Galaxy SII (smarttelefon) med ”Android” operativsystem og SIP-applikasjonen ”CSipSimple”.
- Samsung Galaxy Tab 10.1 (nettbrett) med ”Android” operativsystem og SIP-applikasjonen ”CSipSimple”.

Avhengig av hvilke forsøk som skulle gjøres, ble disse klientene konfigurert med ulike internnummer og brukernavn. På Lok2-lab hadde klientene internnumrene ”221”, ”222” og ”223”, med tilhørende brukernavn ”user221”, ”user222” og ”user223”. Tilsvarende hadde klientene som var regsitrert hos Asterisk-serveren på Lok3-Lab internnumrene ”331”, ”332” og ”333”, med tilhørende brukernavn ”user331”, ”user332” og ”user333”.

Klientene hadde mulighet for både WLAN- og UMTS-tilkobling. De kunne være tilkoblet internt WLAN, det vil si at klienten var tilkoblet samme WLAN som den VoIP-serveren den var tilknyttet. Eller de kunne være tilkoblet eksternt WLAN, det vil si at klienten var koblet til et

annet WLAN enn den VoIP-serveren den var tilknyttet var koblet til. Ved UMTS-tilkobling ble mobilnettets (”Universal Mobile Telecommunications System”, UMTS) dataforbindelse benyttet.

4 SIP- og RTP-trafikk i de ulike eksperimentene

I denne rapporten har vi sett på ordinær SIP- og RTP-trafikk mellom to VoIP-klienter. VoIP-klientene har vært koblet opp på ulike måter, både via internt WLAN, eksternt WLAN og UMTS. Vi har sett på eksperimenter der klientene er registrert hos samme Asterisk-server (delkapittel 4.1), hos hver sin server (delkapittel 4.2 og 4.3), og hvor VoIP-klientene har kommunisert med brukere i PSTN-nettet (delkapittel 4.4). I tilfellet hvor klientene var koblet opp mot hver sin voip-server, ble det gjort målinger der klientene kontaktet hverandre henholdsvis direkte og via en VoIP-tjenesteleverandør, nærmere bestemt ”voip.ms” [16].

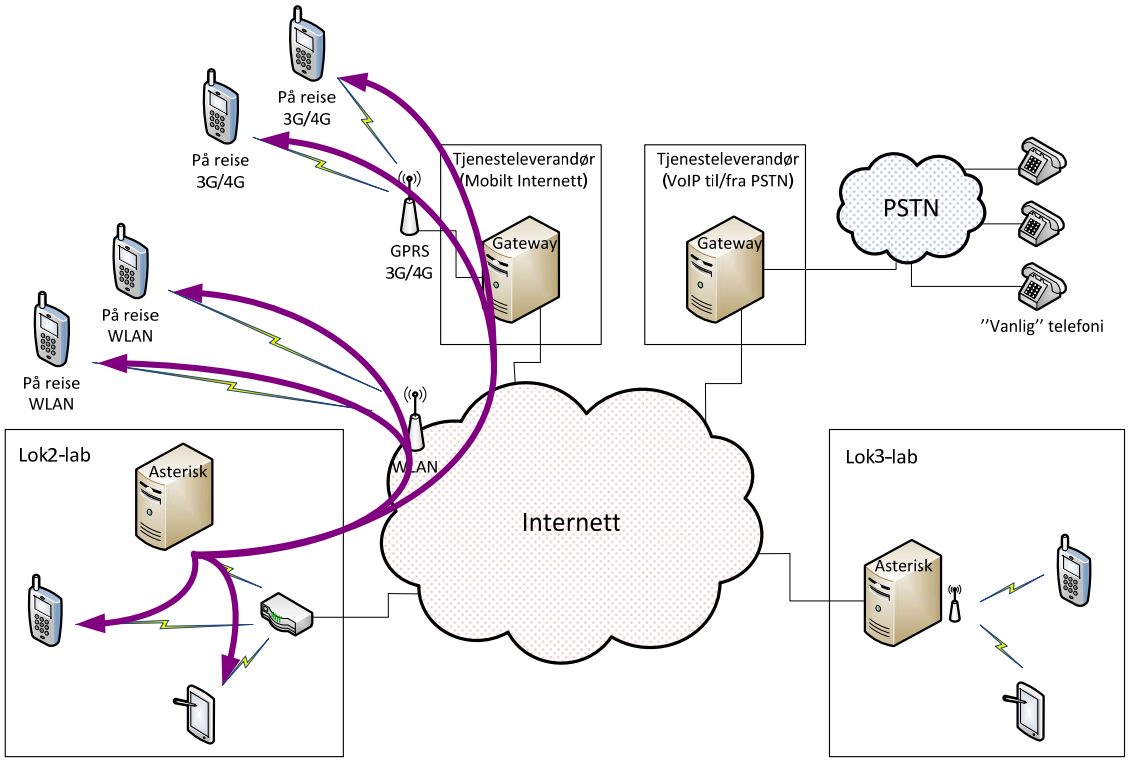
Når vi undersøkte SIP- og RTP-trafikk under alle disse oppsettene, fant vi raskt ut at mye av trafikken var lik, uavhengig av hvordan klientene var tilknyttet og uavhengig av om det var en eller to servere som var involvert. På bakgrunn av dette har vi valgt å dele trafikken inn i ulike ”steg”, og å vise typiske eksempler på disse. Disse stegene har vi kalt ”registrering av klient”, ”oppstart av sesjon”, ”media-trafikk”, ”avslutning av sesjon” og ”avregistrering av klient”. Alle disse stegene er vist i delkapittel 4.1, der klientene er tilknyttet samme VoIP-server. For de andre eksperimentene er bare avvik fra dette vist.

4.1 SIP- og RTP-trafikk for klienter tilknyttet samme VoIP-server

Vi startet med å se på SIP- og RTP-trafikk mellom to klienter som var registrert hos samme Asterisk-server. De kunne imidlertid være tilknyttet serveren via internt WLAN, eksternt WLAN eller via UMTS, hvor de to sistnevnte tilfellene vil være tilfellet ved for eksempel reiser. Vi testet ut ulike kombinasjoner av hvordan klientene var tilknyttet VoIP-serverne, og totalt ble det foretatt følgende 9 deleksperimenter:

- Klient tilknyttet internt WLAN kontakter klient tilknyttet internt WLAN
- Klient tilknyttet internt WLAN kontakter klient tilknyttet eksternt WLAN
- Klient tilknyttet internt WLAN kontakter klient tilknyttet UMTS
- Klient tilknyttet eksternt WLAN kontakter klient tilknyttet internt WLAN
- Klient tilknyttet eksternt WLAN kontakter klient tilknyttet eksternt WLAN
- Klient tilknyttet eksternt WLAN kontakter klient tilknyttet UMTS
- Klient tilknyttet UMTS kontakter klient tilknyttet internt WLAN
- Klient tilknyttet UMTS kontakter klient tilknyttet eksternt WLAN
- Klient tilknyttet UMTS kontakter klient tilknyttet UMTS

Figuren nedenfor illustrerer de ulike konfigurasjonene:



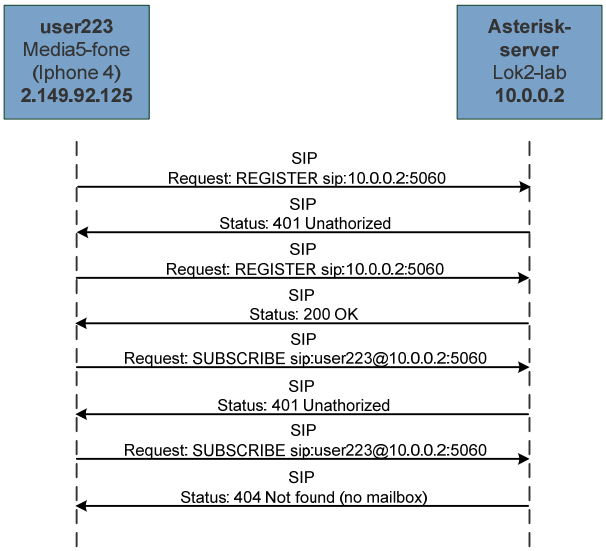
Figur 4.1 Oversikt som viser mulige laboppsett når to klienter tilknyttet samme Asterisk-server kommmuniserer.

Som ventet viste det seg at SIP- og RTP-trafikken som gikk mellom klient og server var uavhengig av hvordan klienten var koblet opp mot Asterisk-serveren, bortsett fra at IP-adressene det ble kommunisert mellom var ulike. Nedenfor er det vist eksempler på typisk trafikk for de ulike stegene omtalt i starten av kapittelet.

4.1.1 Registrering av klient

Det første som skjer når du starter en VoIP-applikasjon er at klienten registrerer seg hos Asterisk-serveren. Dette skjer på samme måte uavhengig av om klienten er tilknyttet serveren via internt WLAN, eksternt WLAN eller UMTS. Nedenfor er det vist et eksempel der klienten ”user223” er tilknyttet serveren på Lok2-lab ved bruk av UMTS. ”user223” har fått ip-adressen 2.149.92.125, som er en av Telenor sine adresser. Dette er fordi smarttelefonen som er brukt, en ”Iphone 4”, har et abonnement fra Telenor.

Klienten sender først en ”REGISTER”-forespørsel til Asterisk-serveren, se Figur 4.2. Ved behandling av ”REGISTER”-forespørselen trenger Asterisk-serveren mer informasjon for å autentisere klienten. Serveren svarer derfor med ”401 Unauthorized”. Klienten sender så en ny forespørsel om å registrere seg, og sender da med autentiseringsinformasjon. Asterisk-serveren har nå fått den informasjonen den trenger, og svarer med ”200 OK”.



Figur 4.2 Typisk eksempel på SIP-trafikk ved registrering av klient. Her er ”user223” tilknyttet serveren ved bruk av UMTS, og har blitt tildelt en ip-adresse fra Telenor.

Litt uventet inneholder SIP-forespørselen den interne IP-adressen til serveren. Dette er fordi ruteren på Lok2-lab endrer IP-adressen i SIP-headeren fra 193.156.31.128, som er den eksterne IP-adressen til 10.0.0.2 som er den interne IP-adressen. Det var overraskende at en rimelig ruter, Netgear WNR 2200 til under 500 kr, hadde slik funksjonalitet. Denne funksjonaliteten kunne imidlertid skrus av.

Muligheten til å sende og motta varsler ved spesielle hendelser støttes i SIP ved forespørslene SUBSCRIBE og NOTIFY. Det som skjer i vårt tilfelle er at klienten ber om å bli varslet ved endringer i meldingsboksen. På samme måte som ved registreringen trenger serveren mer informasjon for å kunne autentisere klienten. ”user223” sender denne informasjonen i en ny SUBSRIBE-melding, men får da til svar at serveren ikke finner meldingsboksen: ”Status: 404 Not found (no mailbox)”. Dette stemmer med at vi ikke har satt opp noen meldingsboks for klienten.

Så lenge klienten er påkoblet vil den sende regelmessige REGISTER- og SUBSCRIBE-forespørsler. Ut fra hva vi har observert i våre eksperimenter, ser det ut til å være ulik praksis hos de ulike SIP-klientapplikasjonene for hvor ofte dette gjøres. ”CSipSimple” ser ut til å sende en REGISTER-forespørsel omtrent hvert 15. minutt (hvert 895 sekund) med tilhørende SUBSCRIBE-forespørsel, mens kun SUBSCRIBE-forspørsler alene i tillegg sendes hvert 5. minutt.

Det vi observerer stemmer med at parameteren ”Expires” har blitt satt til 900 i ”CsipSimple” sin REGISTER-forespørsel, og at serveren svarer med det samme i sin ”200 OK”-respons. Det vil si at registreringen utløper etter 900 sekunder eller 15 minutter. Klienten registrerer seg derfor på nytt etter 895 sekunder, 5 sekunder før registreringen utløper. Den sender deretter en tilhørende SUBSCRIBE-melding. I SUBSCRIBE-meldingene blir parameteren ”expires” satt til 3600. Det vil si at en ny SUBSCRIBE-medinger bør sendes før det har gått 3600 sekunder eller 1time.

Det er derfor litt uklart hvorfor det blir sendt egne SUBSCRIBE-meldinger i tillegg hvert 5. minutt, men grunnen kan være at klienten ikke får noen 200 OK-melding tilbake på SUBSCRIBE-meldingen som bekrefter utløpstiden på 1 time.

”Media5-fone” derimot, ser ut til å sende REGISTER-forespørsler hyppigere, hvert 90. sekund, og da uten noen tilhørende SUBSCRIBE-melding, mens REGISTER- og SUBSCRIBE-meldinger sammen blir sendt sjeldent, og ved svært varierende intervall. I våre forsøk varierte dette mellom 30 og 150 minutter.

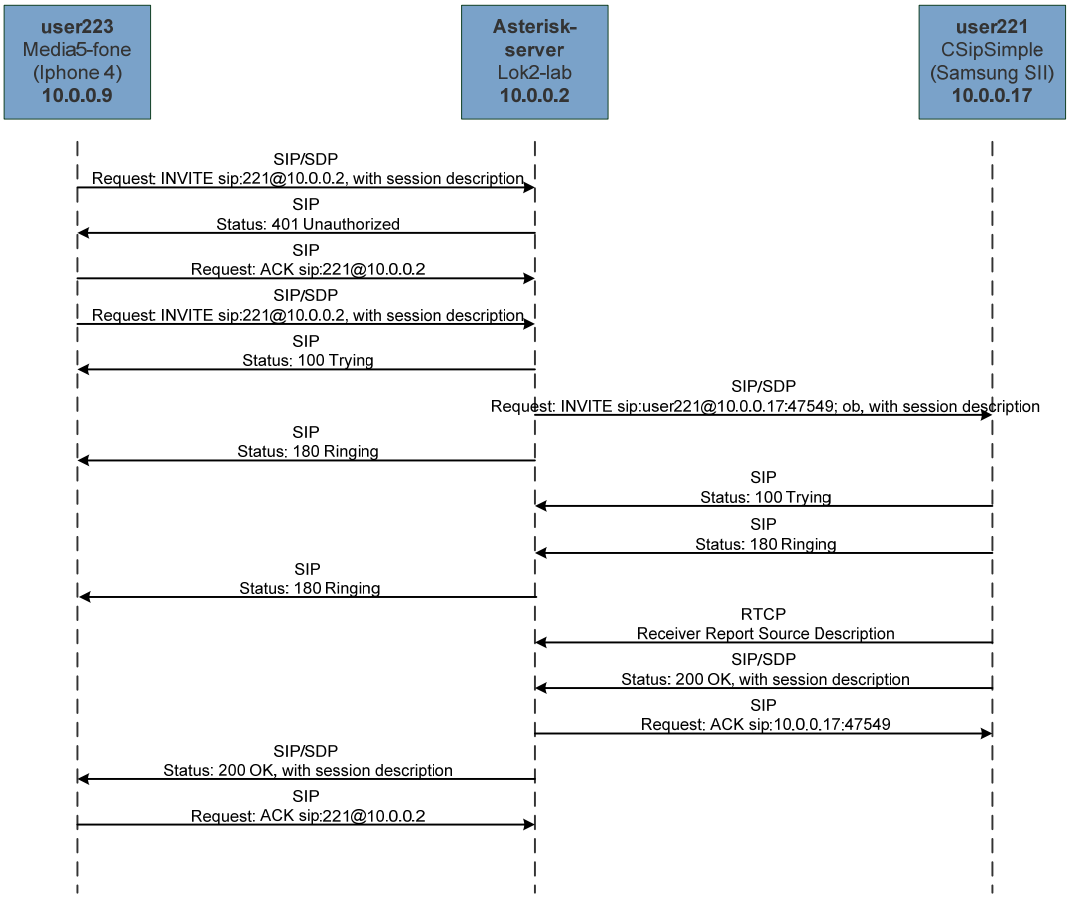
”Media5-fone” setter ”Expires” til 120 sekunder i sin REGISTER-melding, og serveren svarer med det samme i sin ”200 OK”-melding. Denne klienten velger derfor å registrere seg på nytt 30 sekunder før registreringen utløper. Hvor lenge før registreringen utløper man velger å registrere seg på nytt er derfor ulik i våre to applikasjoner.

Noe av årsaken til at REGISTER- med tilhørende SUBSCRIBE-meldinger blir sendt ved så varierende intervall i ”Media5-fone” kan være at klienten heller ikke med denne applikasjonen får noen ”200 OK”-melding som bekrefter utløpstiden.

Kort oppsummert kan en si at den første registreringen ser lik ut ved våre to applikasjoner, men hvordan applikasjonene velger å vedlikeholde sine registreringer er ulik.

4.1.2 Oppstart av sesjon

Oppstart av en sesjon skjer når en klient initierer en samtale med en annen klient. I Figur 4.3 er det vist et eksempel der ”user223” kontakter ”user221”, og hvor begge er registrert hos den lokale serveren på Lok2-lab. Begge klienter er koblet opp via internt WLAN, og ”user223” har lokal ip-adresse 10.0.0.9, ”user221” har lokal ip-adresse 10.0.0.17 og Asterisk-serveren har som tidligere nevnt lokal ip-adresse 10.0.0.2.



Figur 4.3 Typisk eksempel på SIP-trafikk ved oppstart av sesjon. Her er begge klienter registrert hos Asterisk-serveren på Lok2-lab og tilknyttet serveren ved bruk av internt WLAN.

Her er det altså klient ”user223” som ønsker en samtale med klient ”user221”. ”user223” vet ikke ip-adressen til ”user221”, men til serveren hvor den er registrert. Den sender derfor en INVITE-forespørsel til serveren. INVITE-forespørselen inneholder også en SDP-nyttelast som gir ”user221” en del informasjon som den trenger for å sette opp en RTP-mediastrøm til ”user223”.

På samme måte som ved forespørslene REGISTER og SUBSCRIBE, trenger Asterisk-serveren mer informasjon for å kunne autentisere klienten før den kan behandle INVITE-forespørselen. Asterisk-serveren svarer da med responsen ”401 Unauthorized” for å få autentiseringsinformasjonen den trenger. Når den har fått denne informasjonen, svarer den med en ”100 Trying”-melding, og sender forespørselen videre til klient ”user221”. Asterisk-serveren vet IP-adressen til ”user221”, som den fikk da denne klienten registrerte seg.

I praksis lager Asterisk-serveren en ny INVITE-forespørsel som den sender til ”user221”. I den nye INVITE-forespørselen har Asterisk-serveren endret mediaport og ip-adresse i SDP-nyttelasten fra ”user223” sin til sin egen. På denne måten får ”user221” ingen informasjon om ”user223” sin ip-adresse og mediaport, og den vil dermed sende RTP-pakker til Asterisk serveren, som sender disse videre til ”user223”.

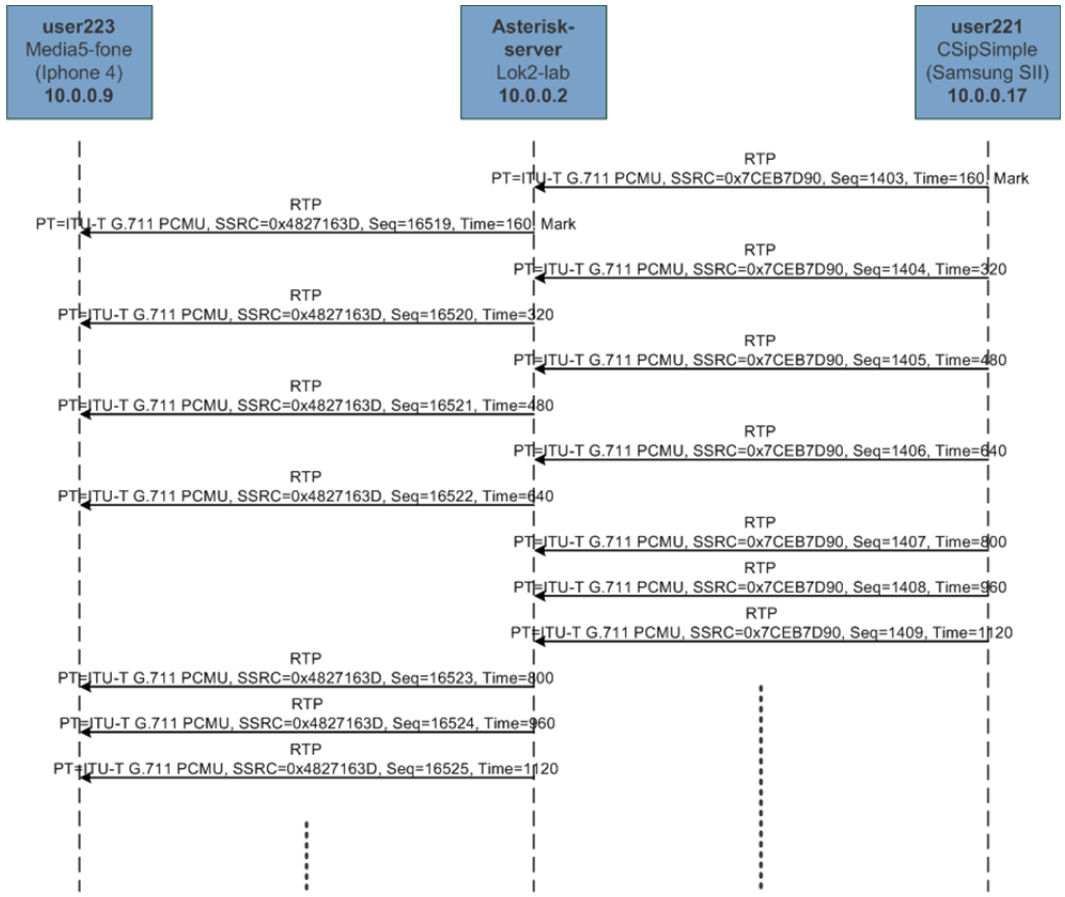
Når ”user221”-klienten får INVITE-forespørselen, svarer den med en ”100 Trying”-respons. Den starter så å ringe, og sender samtidig responsen ”180 Ringing” til Asterisk-serveren, som videreformidler denne til ”user223”. Hos mottakeren av anropet står det nå ”user221” på displayet som angitt i ”callerid” i ”sip.conf”-filen hos Asterisk-serveren, se [8]. Når brukeren av ”user221”-klienten aksepterer anropet, sender denne klienten en ”200 OK”-melding til serveren. Denne meldingen inneholder også en SDP-nyttelast med informasjon om hvor blant annet oppringeren skal sende RTP-trafikken. Serveren svarer med en ACK-melding. Asterisk-serveren sender så en ny ”200 OK”-melding til ”user223”, hvor den igjen har endret ip-adresse og mediaport i SDP-nyttelasten til sin egen. Oppringeren bekrefter mottaket av ”200 OK”-meldingen med en ACK-melding.

Asterisk-serveren har dermed satt opp mediastrømmen til å gå via seg, det vil si at RTP-pakkene går fra ”user223” til serveren og så videre til ”user221” og motsatt.

Noe som er litt spesielt i dette tilfellet er at serveren sender en ”180 Ringing”-melding til oppringeren før den har mottatt noen respons på INVITE-forespørselen fra mottakeren. Denne ”180 Ringing”-meldingen er helt lik den som blir sendt etterpå. Denne første meldingen blir ikke sendt dersom klientene er registrert hos hver sin server. Hvorfor serveren gjør det på denne måten er litt uklart, men antakelig er det fordi den vet at mottakeren er registrert, og antakelig ikke har noen samtale gående. Sistnevnte kan ikke serveren vite sikkert, da det er mulig for en klient å ha en samtale uten at oppringerens server ser dette. Dette er beskrevet nærmere i delkapittel 4.2 og 4.3.

4.1.3 Media-trafikk

Når SIP-meldingsutvekslingen har etablert mediaforbindelsen vil RTP-protokollen bli brukt for å transportere tale-trafikken. I alle tilfellene vi testet hvor klientene var tilknyttet samme server, gikk RTP-trafikken via Asterisk-serveren. Det stemmer med hva vi så under oppstart av en sesjon vist i avsnittet over. Et eksempel på RTP-trafikk er vist i Figur 4.4.



Figur 4.4 Eksempel på RTP-trafikk mellom ”user223” og ”user221”, der begge klienter er på samme WLAN.

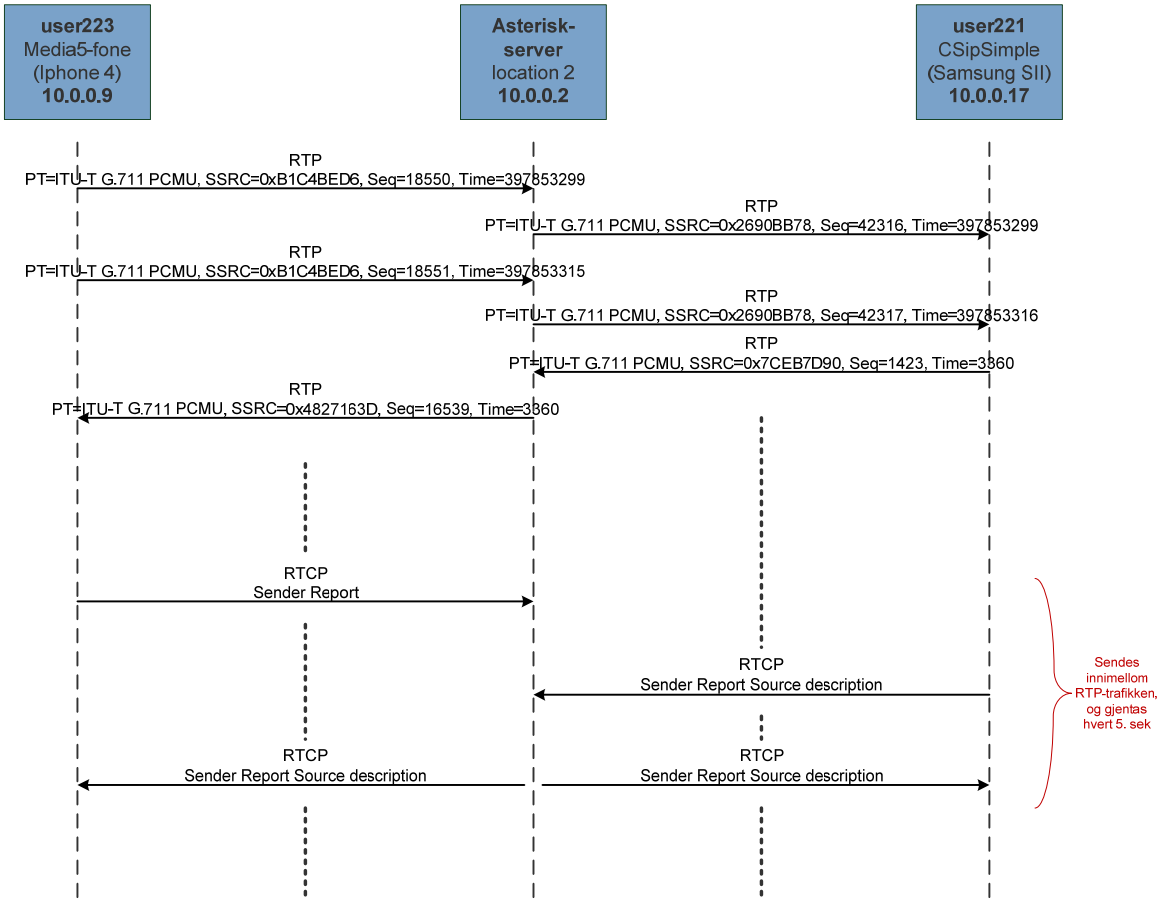
I headeren på RTP-pakkene blir det blant annet spesifisert hvilken type nyttelast den inneholder, og hvordan den er kodet. I vårt tilfelle er dette med standarden G.711, som er en ITU-T-standard for tale, også kalt ”Pulse Code Modulation (PCM) of voice frequencies”. Videre blir synkroniseringskilden, SSRC, identifisert. Dette er en verdi som velges vilkårlig for hver kilde, og hensikten er at to synkroniseringskilder i samme sesjon skal ha ulik identitet. Pakkene får videre et sekvensnummer, som øker med en for hver RTP-pakke som blir sendt fra en gitt sender til en gitt mottaker, se Figur 4.4. I tillegg får de et tidsstempel. Sekvensnummeret blir brukt sammen med tidsstempel og synkroniseringskilde til å sortere mottatte pakker, og til å detektere tapte pakker, som tidligere nevnt i kapittel 2. Pakketap kan forekomme da RTP bruker UDP, og det dermed ikke er noen garanti for at alle pakkene kommer frem til mottakeren.

”RTP Control Protocol” er som tidligere nevnt en protokoll som kjøres parallelt med RTP, og dens primærfunksjon er å evaluere og gi tilbakemelding på den tjenesten som RTP leverer. Dette skjer ved å periodevis sende statistisk informasjon til deltakerne av sesjonen. Slik informasjon kan være pakketap, jitter og round-trip forsinkelse. Denne informasjonen kan brukes til for eksempel å endre talekode.

I våre målinger mottok Asterisk-serveren ”RTCP Sender Report” eller ”RTCP Sender Report Source description” fra klientene som kommuniserte ca hvert 5. sekund, hvor den statistiske

informasjonen lå. Vi er litt usikre på hvorfor klientene noen ganger bare sender ”RTCP Sender Report”. I tillegg sender serveren ut ”RTCP Sender Report Source description” nøyaktig hvert 5. sekund. ”Sender Report Source discription”-meldinger inneholder som tidligere nevnt i tillegg til statistisk informasjon om kanalen også blant annet ”Canonical Name” (CNAME), som er en helt unik identifikator som ikke endrer seg under sesjonen. Den er blant annet nyttig hvis SSRC-identiteter tilfeldigvis skulle bli den samme og må endres, eller hvis et program restartes. CNAME er en binding mellom SSRC og klienten som er konstant. I våre eksperimenter er CNAME den lokale IP-adressen, også dersom serveren og klientene er på ulike WLAN. CNAME blir bare brukt av klientene. Serveren legger ikke inn informasjon i CNAME i våre eksperimenter.

Syklusen med at oppringeren sender en RTCP Sender Report, og at serveren og klienten som blir oppringt sender en “RTCP Sender Report Source description” gjentok seg hvert 5. sekund. Dette er vist i eksempelet i Figur 4.5.



Figur 4.5 Eksempel på hvordan RTCP-pakker periodevis sendes mellom ”user223” og ”user221”, der begge klienter er på samme WLAN.

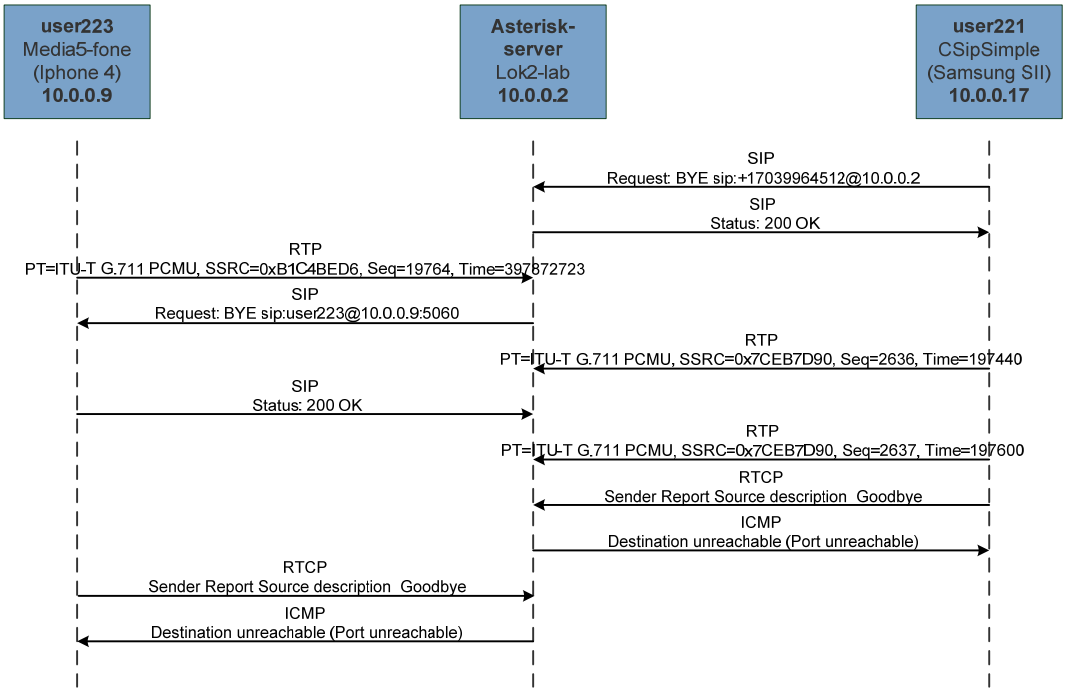
4.1.4 Avslutning av sesjon

Når en klient ønsker å avslutte en samtale og ”legger på røret”, blir det sendt en BYE-melding fra klienten til Asterisk-serveren. Serveren svarer med en ”200 OK”-respons, og videresender BYE-

meldingen til den andre klienten. Denne svarer tilbake til Asterisk-serveren med en ”200 OK”-respons. Ved avslutning av en sesjon skjer det ingen autentisering av klienten som initierer dette.

RTP-trafikken avsluttes ikke med en gang. Dette skjer først når RTCP sender en ”RTCP Sender Report Source description Goodbye”.

Et eksempel på avslutning av en sesjon er vist nedenfor. Her er det ”user221” og ”user223”, begge koblet til serveren ved bruk av internt WLAN, som kommuniserer. Det er ”user221” som initierer avslutningen av samtalen.

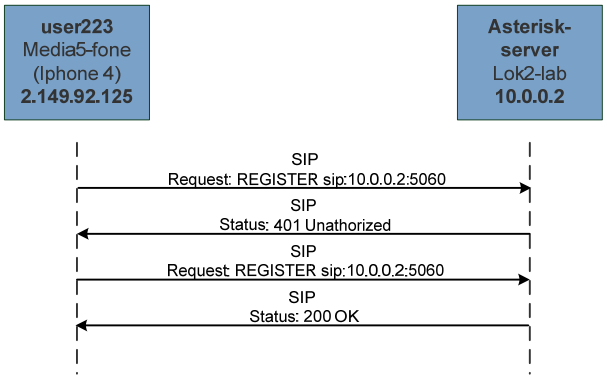


Figur 4.6 Eksempel på SIP-trafikk ved avslutning av sesjon. Her er det ”user221” som avslutter samtalen.

4.1.5 Avregistrering av klient

Når VoIP-applikasjonen avsluttes blir det sendt en REGISTER-melding. Parameteren ”Expires” er nå satt til 0. Det vil si at registreringen avsluttes om 0 sekunder, altså der og da. På samme måte som når klienten registrerer seg, må klienten autentiseres. Når dette er gjort mottar klienten en ”200 OK”-respons fra serveren. Denne inneholder på samme måte ”Expires” satt til 0. Meldingsutvekslingen er vist i eksempelet under, der ”user223” er tilkoblet via UMTS, og dermed har blitt tildelt en ip-adresse, 2.149.92.125, fra teleoperatøren Telenor.

Om telefonen bare blir slått av uten å avslutte VoIP-applikasjonen, skjer det ingen avregistrering ved vårt laboppsett.

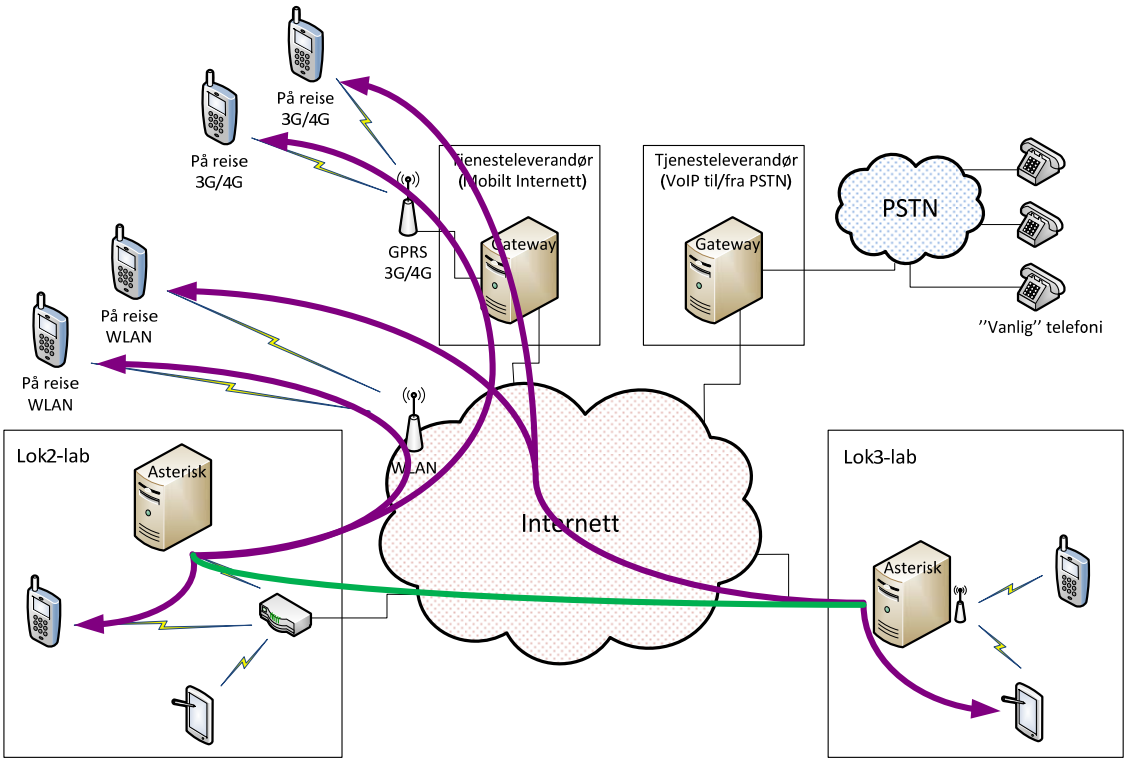


Figur 4.7 Eksempel på SIP-trafikk ved avregistrering av en klient. Her er ”user223” koblet opp mot serveren ved bruk av UMTS.

4.2 SIP- og RTP-trafikk for klienter tilknyttet ulike VoIP-servere som kontakter hverandre direkte

Etter å ha undersøkt hvordan SIP- og RTP-trafikken går mellom klienter registrert hos samme Asterisk-server, foretok vi målinger der klientene er tilknyttet to ulike servere. På samme måte som under forsøket med bare en server, kunne klientene være tilknyttet serveren både via internt WLAN, eksternt WLAN og UMTS. Kommunikasjon mellom de to lokasjonene kan enten gå direkte mellom lokasjonene med for eksempel URI:221@193.156.31.128 eller via ”voip.ms” med for eksempel URI:7039964512@newyork.voip.ms. I dette delkapittelet vises SIP- og RTP-trafikk for klienter som er tilkoblet ulike VoIP-servere, og hvor kommunikasjonen går direkte uten bruk av ”voip.ms”

Laboppsettet er vist i Figur 4.8. Fiolette piler viser mulige alternativer for klientene å være tilkoblet serverne på, mens den grønne direktelinjen illustrerer at det ikke finnes noen mellomliggende voip-infrastruktur.



Figur 4.8 Oversikt som viser mulige laboppsett når to klienter tilknyttet hver sin Asterisk-server kommuniserer direkte uten bruk av "voip.ms".

Vi testet ulike kombinasjoner av hvordan klientene var koblet opp mot sin server, på samme måte som i forsøket med klienter tilknyttet en og samme server. Det viste seg også her at SIP- og RTP-trafikken som går mellom klient og server er uavhengig av om klienten er tilkoblet server via internt WLAN, eksternt WLAN eller UMTS.

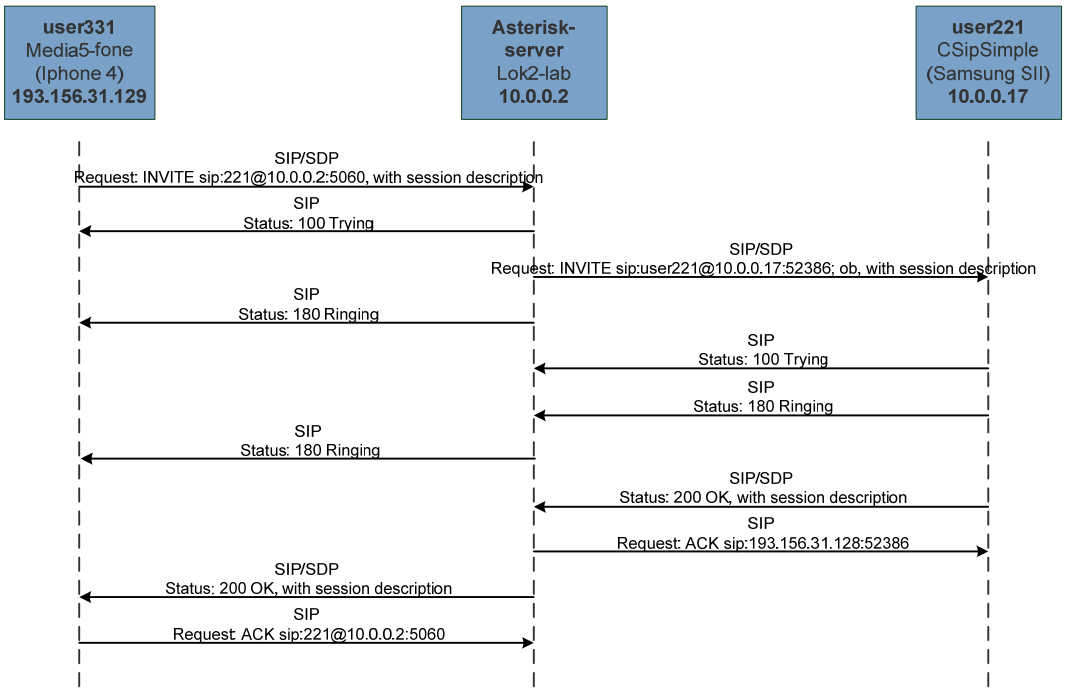
Når det gjelder registrering og avregistrering av klienter så skjer dette på samme måte som med tilfellet hvor begge klienter er tilknyttet samme server, vi har derfor ikke med noe nytt eksempel på dette.

Ved oppstart og avslutning av sesjon vil all SIP- og RTP-trafikk gå direkte fra oppringeren sin klient til mottakerens server og videre til mottakeren. Det vil si at SIP- og RTP-trafikken går utenom oppringerens server. Dette får konsekvenser for oppstart av sesjonen og media-trafikken, og blir beskrevet videre i dette delkapittelet. Bortsett fra at oppringerens server ikke er involvert er SIP- og RTP-trafikken ved avslutning av sesjonen den samme som vist i delkapittel 4.1.4. Vi tar derfor ikke med noe nytt eksempel på dette.

I dette tilfellet opplevde vi noen ganger at det ble sendt “re-INVITE”-forespørsler under sesjonen. Bruk av “re-INVITE”-forespørsler er behandlet i kapittel 4.5.

4.2.1 Oppstart av sesjon

Oppstart av en sesjon skjer på en litt annen måte når klientene er registrert hos hver sin server og kontakter hverandre direkte, enn om klientene er tilkoblet samme server. Dette er forsøkt vist i eksempelet i Figur 4.9. Her er det “user331” som kontakter “user221” direkte ved hjelp av URI: 221@193.156.31.128. “user331” er koblet opp via internt WLAN mot serveren på Lok3-lab og har lokal IP-adresse; 172.30.201.206. “user221” er koblet opp via internt WLAN mot serveren på Lok2-lab og har lokal IP-adresse; 10.0.0.17.



Figur 4.9 Eksempel på SIP-trafikk ved oppstart av sesjon der klientene er tilknyttet to forskjellige VoIP-servere og klientene kontakter hverandre direkte. Her er det ”user331” som ønsker en samtale med ”user221”.

Som tidligere nevnt sender “user331” ved Lok3-lab en INVITE-melding direkte til serveren på Lok2-lab. Forskjellen fra tidligere er at når oppringeren sender en INVITE-melding utenom sin egen server og direkte til mottakerens server, blir det ikke gjort noen autentisering av oppringeren. Serveren hos mottakeren sender ingen ”401 Authentication Required”-melding tilbake til “user331”. Autentisering av oppringeren skjer vanligvis ved den lokale SIP-serveren som oppringeren er registrert hos. Det at oppringeren ikke blir autentisert er en klar sårbarhet som kan utnyttes av en eventuell angriper.

På displayet til mottakeren av samtalen står det nå det som er angitt som ”callerid” på telefonen.

I noen få tilfeller der mottakeren var tilkoblet serveren på Lok3-lab, ble det ikke sendt noen ”100-Trying”-melding fra mottakeren til serveren. Vi vet ikke årsaken til dette. Vi er også litt usikre på hvorfor ”ACK”-meldingen fra serveren til ”user221” bruker den eksterne IP-adressen til klienten.

Som tidligere nevnt endrer ruteren ved Lok2-Lab IP-adressene i SIP-headeren til den lokale IP-adressen. Dette er årsaken til at log-pakkene fra ”user331” til serveren er adressert til IP-adressen 10.0.0.2 og ikke 193.156.31.128.

4.2.2 Media-trafikk

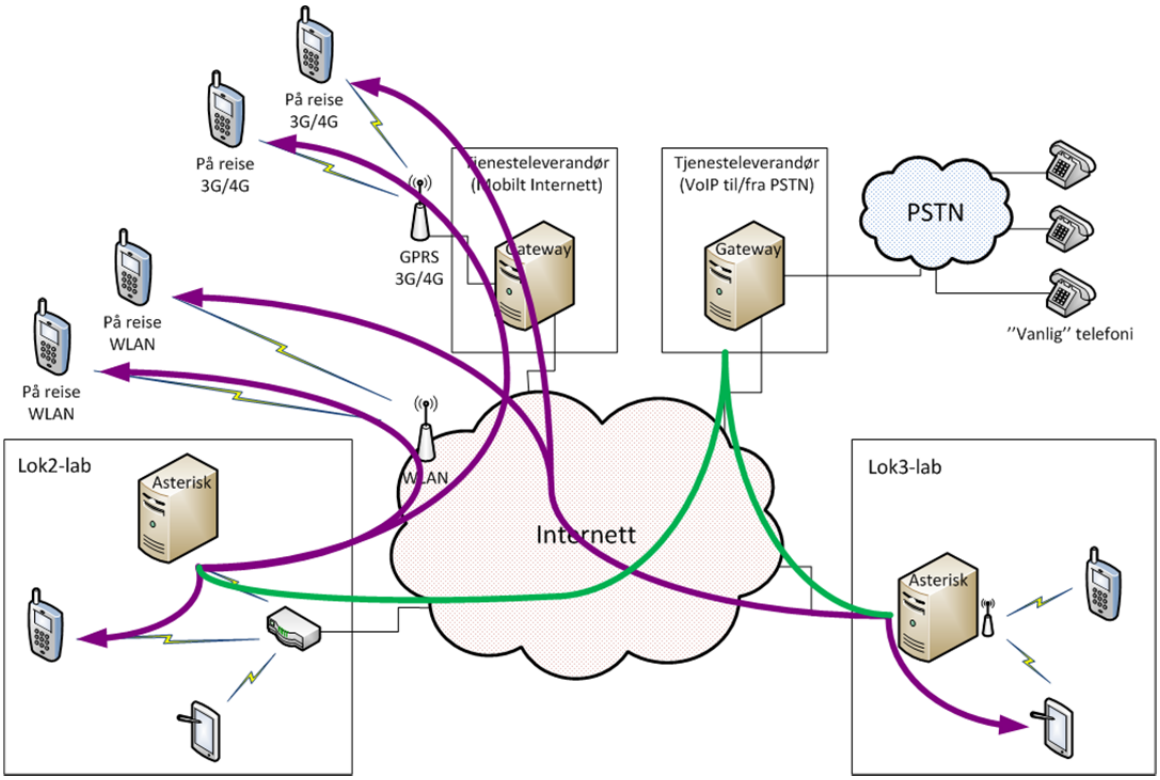
Når oppringeren bruker SIP URI direkte til mottakerens server, vil også RTP-trafikken gå utenom oppringerens server. INVITE-meldingen som oppringeren sender direkte til mottakerens server vil inneholde IP-adresse og mediaport til oppringeren i SDP-nyttelasten, slik at RTP-trafikken vil gå direkte til oppringeren. Mediatrafikken går fortsatt via mottakerens server i våre forsøk, fordi mottakerens server lager en ny SIP-INVITE-melding hvor den endrer IP-adresse og mediaport i SDP-nyttelasten til sin egen IP-adresse. På denne måten vil RTP-pakkene gå via denne serveren, og ikke direkte til oppringeren.

Også i dette tilfellet så vi noen ganger at “re-INVITE”-meldinger ble benyttet. Dette er videre behandlet i kapittel 4.5.

4.3 SIP- og RTP-trafikk for klienter tilknyttet ulike VoIP-servere som kontakter hverandre via ”voip.ms”

Etter å ha undersøkt hvordan SIP- og RTP-trafikken går mellom klienter registrert hos samme Asterisk-server, og mellom klienter registrert hos ulike servere som kommuniserer direkte, foretok vi målinger der klientene er registrert hos ulike servere, og hvor kommunikasjonen foregår via ”voip.ms”. På samme måte som ved de tidligere forsøkene, kunne klientene være tilknyttet serveren både via internt WLAN, eksternt WLAN og UMTS.

Laboppsettet er vist i Figur 4.10. Fiolette piler viser mulige alternativer for klientene å være tilkoblet serveren på, mens de grønne linjene viser hvordan Asterisk-serverne forbindes via ”voip.ms”. Når for eksempel “user331” på Lok3-lab ønsker kontakt med “user221” på Lok2-lab, kan han taste ”17039964512” på tastaturet til VoIP-applikasjonen eller bruke URI: 7039964512@newyork.voip.ms. Dersom “user221” på Lok2-lab ringer til “user331” på Lok3-lab, vil den på samme måte taste ”011476720930 7” på tastaturet eller bruke URI: 4767209307@london.voip.ms. Å kunne bruke URI på denne måten, er ikke nødvendigvis noe en voip-tjenesteleverandør vil støtte, selv om dette var tilfellet med ”voip.ms”. Med vårt laboppsett, måtte telefonnumrene tastes med landskode og retningsnummer som om de ble ringt fra USA.



Figur 4.10 Oversikt som viser mulige laboppsett når to klienter registrert hos hver sin Asterisk-server kommuniserer via "voip.ms".

Selv om begge klientene befant seg på FFI sitt område på Kjeller, så gikk SIP- og RTP-trafikken nå via "voip.ms" sine servere i London og New York.

På samme måte som i tilfellene hvor klientene kommuniserte uten bruk av "voip.ms", var det ingen forskjell i SIP-meldingsutvekslingen selv om klientene var koblet opp mot Asterisk-serveren via internt WLAN, eksternt WLAN eller UMTS.

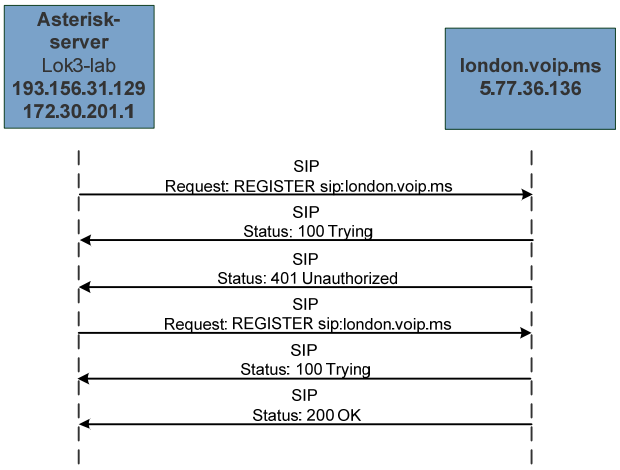
Registrering og avregistrering av klient skjer på samme måte som i tilfellet hvor begge klienter er tilknyttet samme server, og denne trafikken involverer heller ikke "voip.ms" sin server. Vi har derfor ikke tatt med noe nytt eksempel på dette. Avslutning av sesjonen er også helt lik, bortsett fra at meldingene blir sendt via "voip.ms" sine servere. Vi viser derfor heller ikke noe eksempel på dette.

Derimot trenger Asterisk-serverne å koble seg opp mot "voip.ms" sine servere. Dette har vi tatt med og vist et eksempel på. Vi har også vist et eksempel på oppstart av en sesjon, både ved bruk av telefonnummer og ved bruk av URI, da dette skjer på en litt annen måte enn i de foregående eksemplene.

I dette tilfellet opplevde vi ofte at det ble sendt "re-INVITE"-meldinger, som oftest fordi oppringerens server ønsket å rute trafikk utenom oppringerens server. Dette er nærmere beskrevet i delkapittel 4.5.

4.3.1 Registrering av serveren hos voip.ms

Når Asterisk-serveren skal registrere seg opp mot ”voip.ms” følger SIP-trafikken noe av det samme mønsteret som når en klient skal registrere seg opp mot en Asterisk-server. Et eksempel på dette er vist i Figur 4.11 under. Her er det Astersik-serveren ved Lok3-lab med intern ip-adresse 172.30.201.1 og ekstern ip-adresse 193.156.31.129, som registrerer seg hos ”london.voip.ms” med ip-adresse 5.77.36.136.



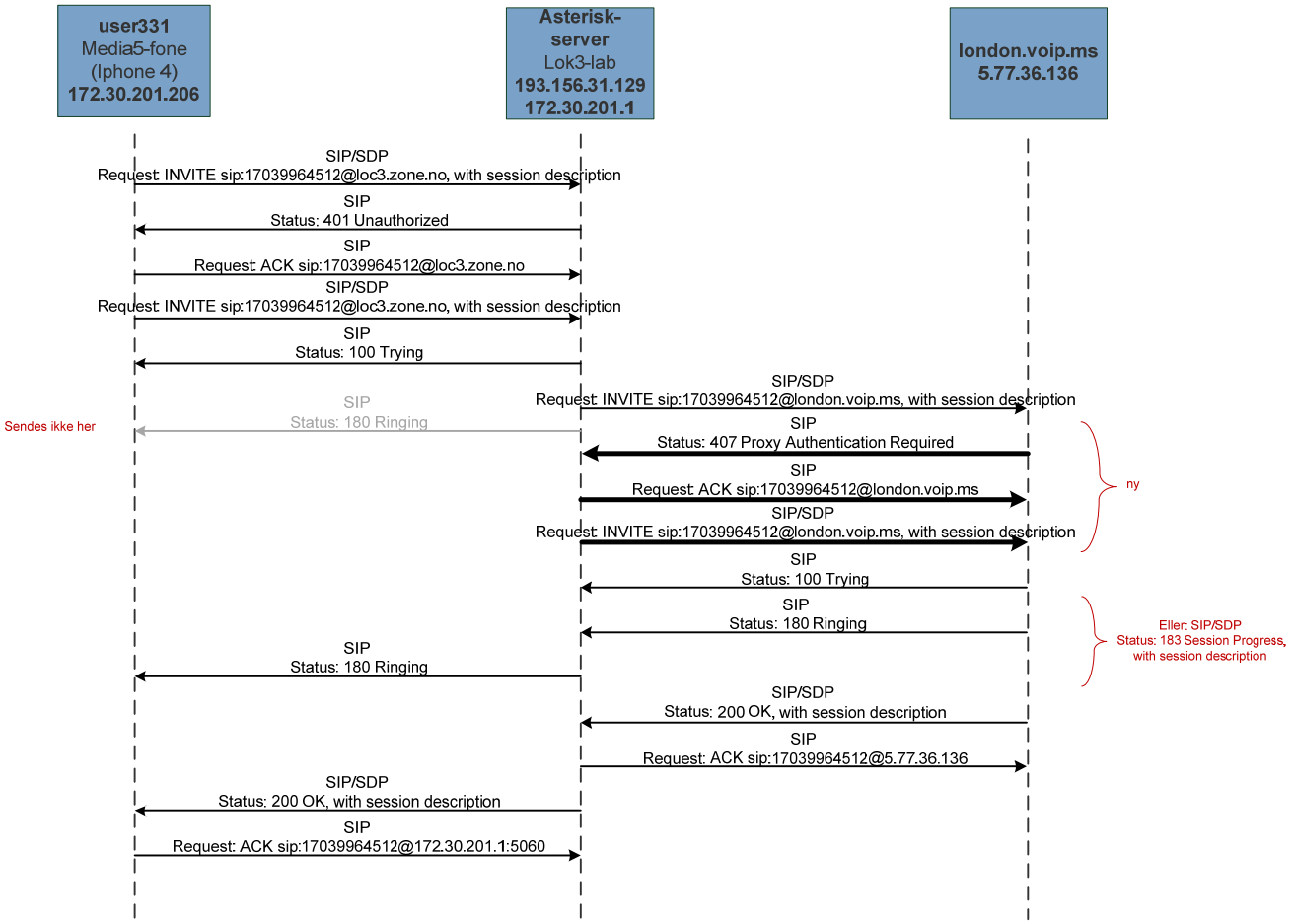
Figur 4.11 Eksempel på SIP-trafikk ved registrering av Asterisk-serveren hos ”voip.ms”.

Etter å ha fått den første REGISTER-forespørselen fra Asterisk-serveren, svarer ”london.voip.ms” først med en ”100 Trying”-melding, og deretter at den trenger mer autentiseringsinformasjon. Asterisk-serveren sender da en ny forespørsel med denne informasjonen. ”Voip.ms”-serveren svarer da igjen først med en ”100 Trying”-melding, og deretter med en ”200 OK”-melding. Det at det sendes ”100 Trying”-meldinger avviker fra SIP-trafikken vi ser når klienter registreres opp mot Asterisk-serverne. I tillegg blir det her ikke sendt noen SUBSCRIBE-meldinger i dette tilfellet.

I våre målinger skjer denne registreringen hvert 105. sekund. Dette harmonerer bra med at parameteren ”Expires” blir satt til 120 sekunder i REGISTER- og ”200 OK”-meldingene. Det vil si at det må skje en ny registrering innen to minutter om serveren skal fortsette å være registrert.

4.3.2 Oppstart av sesjon ved bruk av voip.ms og tildelt telefonnummer

Etter at klientene er registrert hos hver sin Asterisk-server, og Asterisk-serverne er registrert hos ”voip. ms”, er systemet klart til å sette opp en forbindelse mellom klienter på Lok2-lab og Lok3-lab via ”voip.ms”. Å starte en sesjon ved oppringt ”voip.ms”-nummer skjer på omtrent samme vis som ved en server, men noen forskjeller er det, og disse er forsøkt illustrert i Figur 4.12.



Figur 4.12 Eksempel på SIP-trafikk ved oppstart av sesjon der klientene er tilknyttet to forskjellige servere og hvor trafikken går via "voip.ms" ved bruk av telefonnummer. Forskjellene i forhold til tilfellet der klientene er tilknyttet samme server er angitt. Her er SIP-trafikk mellom oppringeren "user331" og "voip.ms" sin server i London vist.

I dette eksemplet er det "user331" som er registrert hos serveren på Lok3-lab som prøver å ringe "user221" som er registrert hos serveren på Lok2-lab. Begge er tilkoblet via internt WLAN.

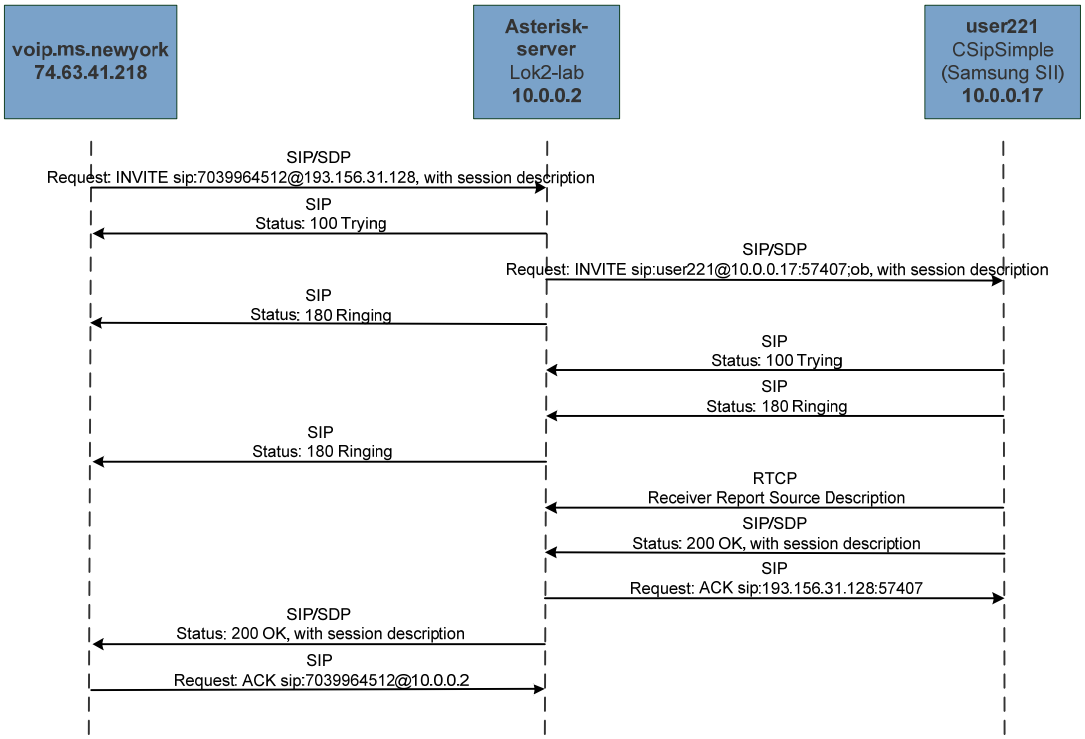
På samme måte som i det første eksempelet, må oppringeren "user331" sende to INVITE-forespørsler, for å bli autentisert hos Asterisk-serveren på Lok3-lab. Når dette er gjort svarer den samme Asterisk-serveren bare med en "100 Trying"-melding, men ikke med en "180 Ringing"-melding som i tilfellet hvor begge klienter er tilknyttet samme server. Dette skjer antakelig fordi serveren i dette tilfellet ikke har mottakeren registrert hos seg, og dermed ikke vet om det er mulig å få kontakt med denne klienten.

INVITE-forespørselen blir videresendt til "london.voip.ms", se Figur 4.12. "london.voip.ms" svarer da med en "407 Proxy Authentication Required", det vil si at den trenger mer autentiseringsinformasjon av Asterisk-serveren ved Lok3-lab. "407 Proxy Authentication Required" blir returnert av "london.voip.ms" som ønsker å autentisere INVITE-forespørselen før

den videresender den til ”newyork.voip.ms”. Asterisk-serveren ved Lok3-lab svarer med en ACK-melding og en ny INVITE-forespørsel som også inneholder autentiseringsinformasjon. ”london.voip.ms” svarer da med en ”100 Trying”-melding og en ”180 Ringing”-melding, eller en ”100 Trying”-melding og en ”183 Session Progress”-melding. ”180 Ringing”- eller eventuelt ”183 Session Progress”-meldingen blir videresendt av Asterisk-serveren på Lok3-lab til ”user331”. På displayet til mottakeren står det nå det som er angitt som ”callerid” i ”sip.conf”-fila i Asterisk-serveren, se [8].

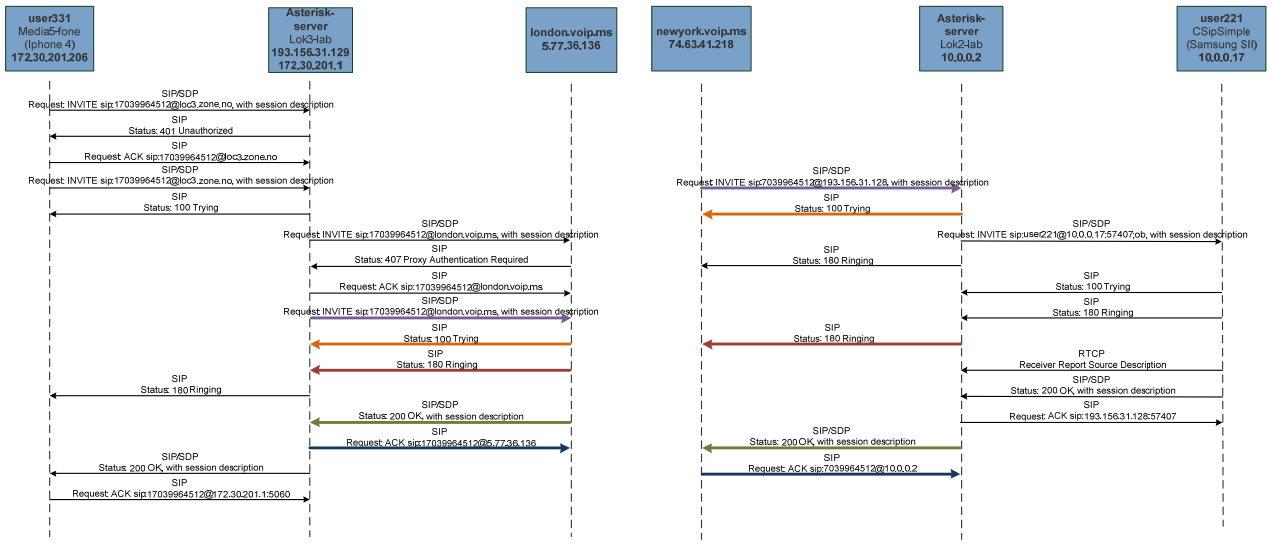
”183 Session Progress”-meldingen vil si at det blir satt opp en foreløpig sesjon hvor mediatrafikk som angir om mottakeren er opptatt, ”ventemusikk” osv blir overført. Ved bruk av ”180 Ringing”-meldingen er serveren derimot sikker på at det ringer hos mottaker. Det er den lokale ringetonen hos serveren som blir benyttet. Se for eksempel [17].

Når det gjelder SIP-trafikken som går mellom ”newyork.voip.ms”, Asterisk-serveren ved Lok2-lab og ”user221”, altså på mottakersiden, så er dette den samme SIP-meldingsutvekslingen som vi så i tilfellet hvor begge klienter var registrert hos samme server, se Figur 4.3 og Figur 4.13.



Figur 4.13 Eksempel på SIP-trafikk ved oppstart av sesjon der klientene er tilknyttet to forskjellige servere, og hvor trafikken går via ”voip.ms” ved bruk av telefonnummer. Her er SIP-trafikk mellom ”voip.ms” sin server i New York og mottakeren ”user221” vist.

Sammenhengen mellom det som skjer på Lok2-lab og Lok3-lab-siden i eksperimentet over er forsøkt vist i Figur 4.14 under. Samme farge på piler indikerer at dette er ”samme” SIP-melding, det vil si at i enkelte tilfeller kan noe av innholdet være endret.

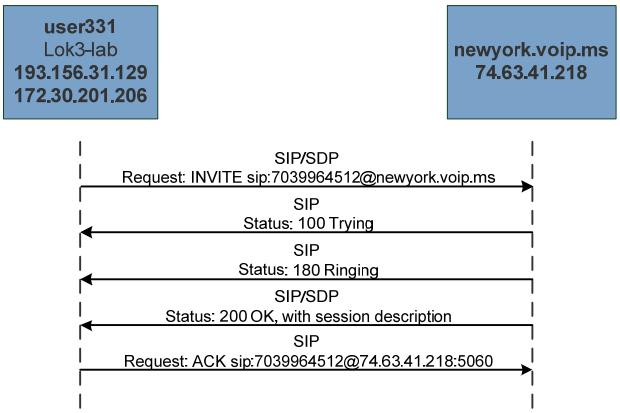


Figur 4.14 Eksempel på SIP-trafikk ved oppstart av sesjon mellom to klienter som er tilknyttet forskjellige servere, og hvor trafikken går via ”voip.ms” sine servere i London og New York ved bruk av telefonnummer. Samme farge på pilene indikerer at dette er ”samme” melding som blir videresendt, i enkelte tilfeller med noe endring av innhold.

4.3.3 Oppstart av sesjon ved bruk av voip.ms og SIP URI

Som nevnt tidligere i dette delkapittelet, kan man i tilfellet hvor klientene er registrert hos hver sin server også kontakte den andre klienten ved hjelp av ”voip.ms” sin SIP URI. Et eksempel på SIP URI vil være 7039964512@newyork.voip.ms.

På samme måte som i tilfellet hvor klientene var registrert hos hver sin server, men kontaktet hverandre direkte, gikk trafikken utenom oppringerens server og det skjedde heller ingen autentisering av klienten. Et eksempel er vist under i Figur 4.15. Der ser vi at etter at ”user331” har sendt en INVITE-forespørsel, svarer ”newyork.voip.ms” med en ”100 Trying”-melding, deretter en ”180 Ringing”-melding og en ”200 OK”-melding når mottakeren aksepterer anropet. Oppringeren svarer på ”200 OK”-meldingen med en ”ACK”-melding. På displayet til mottakeren står det nå det som er angitt som ”callerid” i oppsettet av VoIP-applikasjonene på klientene.



Figur 4.15 Eksempel på SIP-trafikk ved oppstart av sesjon der klientene er tilknyttet to forskjellige servere og hvor trafikken går via "voip.ms" ved bruk av SIP URI. Her er SIP-trafikk mellom oppringeren "user331" og "voip.ms" sin server i New York vist.

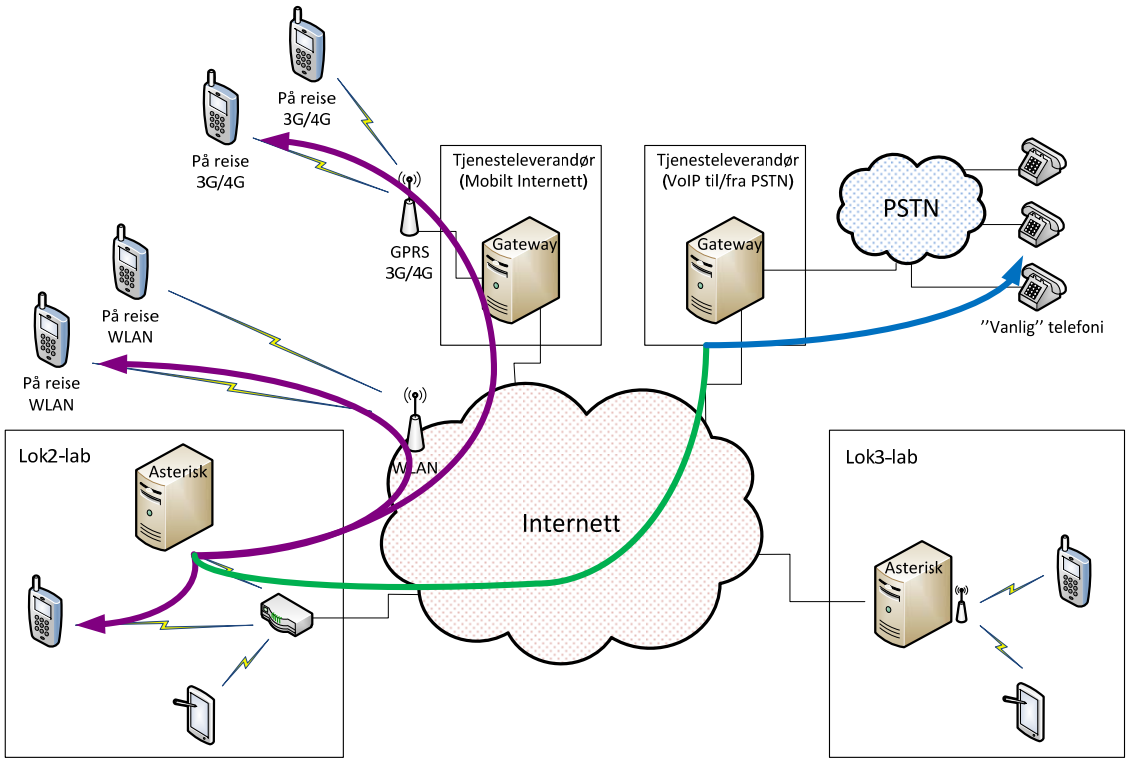
På mottakersiden er trafikken den samme som ved bruk av "voip.ms" og tasting av telefonnumer. Et eksempel på dette er vist tidligere i Figur 4.13.

4.3.4 Media-trafikk via voip.ms

I tilfellet ved bruk av SIP URI gikk all mediatrafikken utenom oppringeren sin server. Ved bruk av telefonnummer, så vi at Asterisk-serveren hvor oppringeren er registrert som oftest prøvde å sette opp mediastrømmen direkte mellom oppringeren og "voip.ms" sin server. For å få til dette brukte den "re-INVITE"-forespørsler, se delkapittel 4.5. Generelt er hensikten med dette å avlaste Asterisk-serveren på oppringeren sin side og redusere RTP-pakkenes forsinkelse. Asterisk-serveren på mottakeren sin side sender ingen "re-INVITE"-forespørsel, så der går trafikken som vanlig via serveren.

4.4 SIP- og RTP-trafikk for klienter som kommuniserer med telefoner i PSTN-nettet

I dette eksperimentet ringer en VoIP-klient registrert hos en Asterisk-server ut til en telefon i PSTN-nettet, eller motsatt. Det vil si at VoIP-klienten blir oppringt fra en telefon i PSTN-nettet. Dette er vist i Figur 4.16. Her er på samme måte fiolette piler mulige alternative måter for klientene å være tilkoblet. Grønn linje er forbindelsen mellom Asterisk-serveren og" voip.ms" sin server, mens den blå pilen her indikerer forbindelsen mellom voip.ms og ut i PSTN-nettet. Den blå pilen indikerer altså linjesvitsjet telefoni.



Figur 4.16 Oversikt som viser mulige laboppsett når en VoIP-klient skal ringe ut i PSTN-nettet, eller motsatt.

I dette tilfellet ser vi kun trafikk som passerer VoIP-delen av forbindelsen. Trafikken som går i denne delen av nettverket er helt lik den som vi så i eksperimentet hvor klientene var registrert hos hver sine servere, og kontaktet hverandre via ”voip.ms” ved bruk av telefonnummer. Det vil si at hvis Voip-klienten er oppringeren, blir SIP-trafikken som vist i eksempelet i Figur 4.12. INVITE-forespørslene inneholder da telefonnummeret til den som blir ringt opp, og med landskode som om samtalen blir ringt fra USA. Det er da ikke ”180 Ringing”-responser, men ”183 Session Progress”-responser som blir sendt fra ”voip.ms” via Asterisk-serveren til VoIP-klienten.

Noen ganger kommer det ”180 Ringing”-reponser rett etter ”183 Session Progress”-responser. Vi har ikke funnet noen dokumentert forklaring på dette.

Hvis derimot VoIP-klienten blir ringt opp fra PSTN-nettet, det vil si at vi ringer ett av våre to kjøpte ”voip.ms”-nummer fra en fast- eller mobiltelefon, så ser SIP-trafikken mellom ”voip.ms” sin server og VoIP-klienten ut som vist tidligere i Figur 4.13.

Når klientene mottar anrop fra PSTN-nettet vises PSTN-nummeret på displayet.

4.5 Bruk av re-INVITE-forespørsler

Relativt ofte under utførelsen av eksperimentene så vi at det ble sendt “re-INVITE”-forespørsler. Dette skjedde spesielt når vi ringte via “voip.ms”. Da ble forespørslene brukt for å rute trafikken utenom oppringeren sin server. Noen ganger så det også ut som om serveren sendte “re-INVITE”-forespørsler for å endre talekode. Dette blir nærmere beskrevet under.

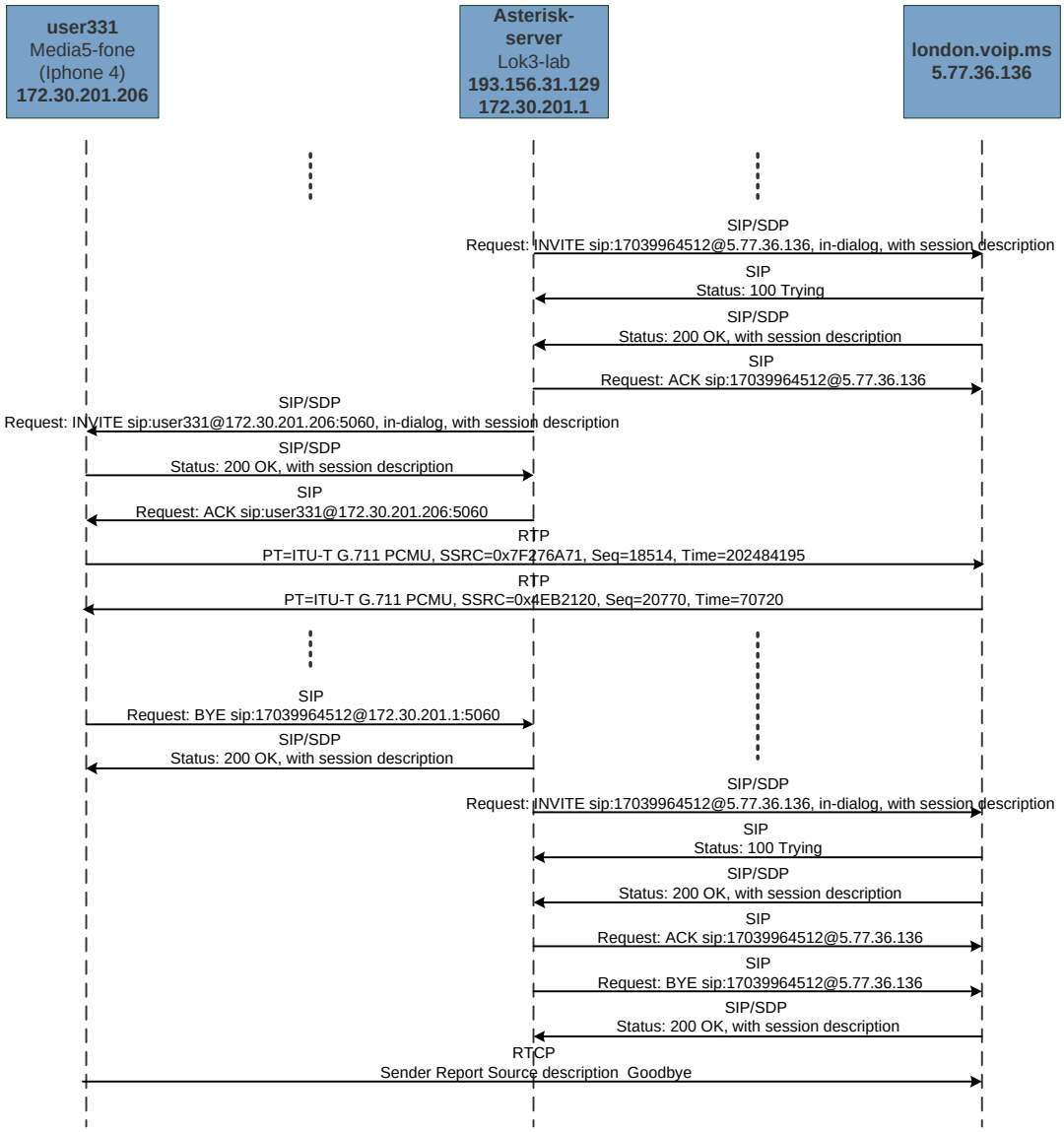
I motsetning til ved en INVITE-forespørsel, blir klienten ikke autentisert ved en “re-INVITE”-forespørsel. Dette er en sårbarhet, da det er mulig for uvedkommende å rute om eller endre talekode på en pågående sesjon.

4.5.1 Bruk av re-INVITE for å rute trafikk utenom oppringerens server

Når SIP setter opp en samtale vil SDP-nyttelasten i INVITE-forespørselen og “200 OK”-responsen inneholde informasjon om hvor mediastrømmen skal sendes. Asterisk-serveren setter seg selv som mottakeren av mediastrømmen når den etablerer en samtale mellom klienter registrert ved samme server og når den etablerer en samtale mellom klienter registrert ved hver sin server via “voip.ms”. RTP-pakkene går da via Asterisk-serveren.

I scenarioet der klientene er registrert hos hver sin server og kommuniserer via ”voip.ms”, ser vi at Asterisk-serveren på oppringerens side sender en ”re-INVITE”-forespørsel til både oppringeren og “voip.ms” sin server. ”re-INVITE”- forspørslene blir sendt rett etter at den initielle INVITE-sesjonen er ferdig behandlet og oppringeren har fått responsen “200 OK”. SDP-nyttelasten i ”re-INVITE”-forespørselen inneholder IP-adressen og mediaporten til endepunktene. Asterisk-serveren bruker ”re-INVITE”-meldinger til å sette opp RTP-trafikken direkte mellom oppringeren og “voip.ms” sin server. På mottakersiden sendes ingen ”re-INVITE”-meldinger og RTP-trafikken går via Asterisk-serveren. Når oppringeren sender en BYE-forespørsel til Asterisk-serveren, sender Asterisk-serveren en ny ”re-INVITE”-forespørsel til “voip.ms” med sin egen IP-adresse og sitt eget portnummer. Asterisk-serveren tar da tilbake kontrollen over RTP-pakkene fra “voip.ms”, før den sender en BYE-forespørsel til “voip.ms” sin server.

Figur 4.17 viser et utsnitt av SIP- og RTP-trafikken ved en samtale hvor Asterisk-serveren bruker ”re-INVITE”-meldinger for å sette opp RTP-trafikk direkte mellom endepunktene “user331” og ”london.voip.ms”. “user331” ved Lok3-lab ringer via “voip.ms” “user221” ved Lok2-lab, hvor begge henger på internt WLAN. Utsnittet i figuren er tatt på oppringerens side etter at den ordinære INVITE-sesjonen er ferdig behandlet og “user331” har fått responsen “200 OK” tilbake. I figuren ser vi at Asterisk-serveren først sender en ”re-INVITE”-forespørsel (INVITE in-dialog) til ”london.voip.ms” og så til “user331”. Når ”re-INVITE”-forespørselen er ferdig behandlet og “200 OK- responsene er mottatt av Asterisk-serveren vil RTP-trafikken gå direkte mellom “user331” og “london.voip.ms”. SIP-trafikken vil derimot fortsatt gå via server.



Figur 4.17 Eksempel på SIP- og RTP-trafikk der Asterisk-serveren bruker en "re-INVITE"-forespørsel for å sette opp RTP-trafikk direkte mellom "user331" og "london.voip.ms".

Figur 4.17 viser også at Asterisk-serveren sender en "re-INVITE"-forespørsel til "london.voip.ms" før den sender en BYE-melding.

Vi ser at "london.voip.ms" ikke ber om autentisering av avsender av "re-INVITE"-forespørselen. Dette er som sagt en sårbarhet i SIP, da det er mulig for andre enn deltakerne i samtalen å endre hvor trafikken skal sendes.

4.5.2 Bruk av re-INVITE ved dårlig kvalitet

Uavhengig av eksperiment ser vi at oppringeren sender en eller flere "re-INVITE"-forespørsler til Asterisk-serveren, hvor oppringeren endrer mediaport for mottak av RTP-trafikk og begrenser mulige talekoder. Dette gjøres i SDP-nyttelasten i "re-INVITE"-meldingen. Mediaporten blir i de fleste tilfeller ikke endret i praksis. Dette tror vi er fordi oppringeren sender RTP-trafikk fra den

opprinnelige mediaporten og ikke fra den som den selv oppgir i SDP-nyttelasten i ”re-INVITE”-forespørselen. I noen tilfeller endrer mediaporten seg for en liten periode, men bare til oppringeren sender fra den opprinnelige mediaporten igjen. RTP-trafikken fra oppringeren har i disse tilfellene noe dårlig kvalitet, så dette kan være årsaken til at oppringeren sender ”re-INVITE”-forespørsler til serveren.

5 Oppsummering og konklusjon

På bakgrunn av den raske utviklingen vi ser innen offentlig elektronisk kommunikasjon, samt at sivile og militære systemer i stadig større grad anvender samme teknologi, er det interessant for Forsvaret å se nærmere på en del av de sivile teknologiene, deriblant VoIP.

I denne rapporten har vi beskrevet de resultatene som fremkom da vi så nærmere på de mest anvendte protokollene for VoIP, nemlig SIP og RTP. I dette arbeidet ønsket vi å se nærmere på hvordan disse protokollene fungerte i praksis ved bruk av tale.

Vi satt opp en lab som bestod av to servere, og ulike klienter, og det ble utført ulike eksperimenter. Disse eksperimentene omfattet tilfellene hvor VoIP-klientene var registrert hos samme server, VoIP-klientene var registrert hos ulike servere men kommuniserte direkte, VoIP-klientene var registrert hos ulike servere men kommuniserte via en ekstern VoIP-gateway og tilslutt at VoIP-klienten kommuniserer ut i PSTN-nettet. I alle eksperimentene ble klientene koblet til serverne via internt WLAN, eksternt WLAN eller UMTS, for å undersøke om dette hadde innvirkning på SIP- eller RTP-trafikken vi så.

Serverprogramvaren som ble benyttet var Asterisk, mens klientene benyttet ”CSipSimple” og ”Media5-fone”. All programvare som ble benyttet var gratis. Å benytte en ekstern voip-gateway for å blant annet kunne kommunisere ut i PSTN-nettet, var imidlertid en tjeneste vi måtte kjøpe fra en kommersiell leverandør.

Eksperimentene viste som ventet at SIP- og RTP-trafikken er lik uavhengig av om klintene er koblet opp mot VoIP-serveren via internt WLAN, eksternt WLAN eller UMTS. Den er også slik man stort sett kunne forvente seg ut fra de eksemplene på SIP-trafikk man finner i ulike bøker og så videre.

Det at en oppringer kan benytte SIP URI for å kontakte mottaker direkte medfører at oppringeren ikke blir autentisert når han starter en sesjon. I alle våre forsøk er det oppringeren sin server som foretar autentiseringen, og når SIP-trafikken går utenom denne serveren blir dette utelatt. Det skjedde heller ingen autentisering ved ”re-INVITE”- eller BYE-forespørslene, noe som gjør det enkelt for uvedkommende for eksempel å endre talekode, rute om RTP-trafikken eller avslutte en sesjon. Dette er klare sårbarheter i våre ukrypterte SIP-implementasjoner. Konfidensialitets- og integritetsbeskyttelse av SIP-trafikken med for eksempel Transport Layer Security (TLS) vil løse dette problemet. Konfidensialitets- og integritetsbeskyttelse forekom imidlertid ikke hos de kommersielle leverandørene vi vurderte. Dette kommer vi nærmere inn på i en senere FFI-rapport.

Referanser

[1] B. H. Farsund, "WiMAX - teknologi, funksjonelle egenskaper og sikkerhet," in *FFI-rapport 2010/01347* 2010.

[2] A. P. Hveem, "Mobilt bredbånd med LTE - teknologi, sikkerhet, tjenester og utbygging," in *FFI-rapport 2011/00709* 2011.

[3] A. P. Hveem, "Sikkerhet i Voice over IP og andre multimediasesjoner basert på SIP og RTP," in *FFI-rapport 2012/00521* 2012.

[4] "http://www.itu.int/rec/T-REC-H.323/en/," 2013.

[5] "http://www.skype.com/no/," 2013.

[6] "http://www.google.com/hangouts/," 2013.

[7] "http://www.asterisk.org/," 2013.

[8] L. Øverlier, "Egen VoIP-server på 1-2-3," in *FFI-rapport 2012/01106* 2012.

[9] Internet Engineering Task Force, "RFC 3261: SIP - Session Initiation Protocol," 2002.

[10] S. Ganguly and S. Bhatnagar, "VoIP: Wireless, P2P and New Enterprise Voice over IP," Wiley, 2008.

[11] J. Epstein, "Scalable VoIP Mobility - Integration and Deployment," Newnes, 2009.

[12] P. Thermos and A. Takanen, "Securing VoIP Networks - Threats, Vulnerabilities and Countermeasures," Addison-Wesley, 2007.

[13] O. Hersent, "IP Telephony - Deploying VoIP Protocols and IMS infrastructure," Wiley, 2010.

[14] "https://lists.cs.columbia.edu/pipermail/sip-implementors/2001-October/001949.html," 2013.

[15] "http://www.wireshark.org/," 2013.

[16] "http://www.voip.ms/," 2013.

[17] "http://wiki.freeswitch.org/wiki/180_vs._183_vs._Early_Media," 2013.

Forkortelser

CNAME	Canonical Name
CSRC	Contributing Source Identifier
GSM	Global System for Mobile communication
IETF	Internet Engineering Task Force
IP	Internet Protocol
LTE	Long Term Evolution
PBX	Private Branch eXchange
PSTN	Public Switched Telephone Network
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SSRC	Synchronization Source Identifier
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
URI	Unified Resource Identifier
VoIP	Voice over IP
WLAN	Wireless Local Area Network