

Egen VoIP-server på 1-2-3

Lasse Øverlier

Forsvarets forskningsinstitutt (FFI)

15. oktober 2012

FFI-rapport 2012/01106

1126

P: ISBN 987-82-464-2167-4

E: ISBN 987-82-464-2168-1

Emneord

VoIP

Telefoni

Telefonsentral

Godkjent av

Kjell Olav Nystuen

Prosjektleder

Anders Eggen

Avdelingssjef

Sammendrag

Denne rapporten viser hvordan man enkelt kan sette opp en egen server med Voice-over-IP (VoIP) programvare for lab-bruk. Vi gjør dette både på vanlig og på minimalistisk hardware, og viser hvor enkelt oppsettet kan være for at klienter skal ha mulighet til å kommunisere gjennom denne serveren. Tale, video og meldinger er mulige kommunikasjonsmetoder. Vi ser først på bruk av tjenesten for intern kommunikasjon mellom en lukket gruppe personer. Deretter ser vi på oppkobling av vår VoIP-tjeneste til PSTN-nettet for å inkludere samtaler inn/ut til alle vanlige telefoner i tillegg til annen type kommunikasjon mellom klientene. Vi tester også oppsett av klienter på Android, Linux og Windows.

English summary

This report demonstrates the simplicity of setting up a Voice-over-IP (VoIP) server for lab usage. We configure the VoIP-server to be usable both on regular PCs and on minimum hardware. We also show the configuration and setup to allow VoIP-clients to communicate through the server. First we explore the basic setup for a small set of users. Then we extend to include communication with the existing telephony network. Client setup on Android, Linux and Windows is also demonstrated and tested.

Innhold

1	Innledning	7
2	Bakgrunn	7
3	Oppsett av VoIP	9
3.1	Oppsett av Linux	9
3.1.1	På PC	9
3.1.2	På Raspberry Pi	10
4	Eksperimenter	12
4.1	Testoppsett 1 - Intern sentral - Kun for interne samtaler mellom brukerne	13
4.2	Testoppsett 2 - Eksternt tilgjengelig sentral - Intern VoIP-server med tilgang fra utsiden og innsiden	14
4.3	Installasjon og oppsett av Asterisk	15
4.3.1	Installasjon	15
4.3.2	Konfigurasjon	16
4.4	Brukere og telefonnummer	17
4.4.1	Intern sentral - kun intern ringing mellom brukerne	17
4.5	Eksternt tilgjengelig sentral	20
4.5.1	Monitorering og debugging	25
4.6	Programvare på Android	28
4.7	Programvare på Linux	30
4.8	Programvare på Windows	31
4.9	Sikkerhet og VPN	31
5	Konklusjon	32
	Bibliografi	33
	Appendiks A Intern sentral	35
A.1	sip.conf	35
A.2	extensions.conf	36
	Appendiks B Eksternt tilgjengelig sentral	37
B.1	sip.conf	37
B.2	extensions.conf	39

1 Innledning

Motivasjonen for en eksperimentell Voice-over-IP (VoIP) lab har vært å gjøre en vurdering av sikkerheten i signalering og trafikkdata i slike nettverk. For å se på dette i en lab-setting ble det satt opp servere med programvare konfigurert på forskjellige måter. Man vurderte så hvordan man kunne sette opp en fungerende og hensiktsmessig VoIP-lab på en enklest mulig måte. Det ble også gjort en vurdering av forskjellig programvare for å oppnå den ønskede funksjonaliteten. I løpet av denne fasen ble det erfart hvor enkelt det egentlig er å sette opp en brukbar telefonitjeneste for en mindre gruppe mennesker, og det ble besluttet å lage en rapport for senere å kunne gjenskape eksperimentene og/eller oppsettene av lab på en enkel måte.

Denne guiden er laget for å vise hvor raskt dette kan gjøres, hvor lite hardware, og hvor lite konfigurering som må på plass for å sette opp en enkel intern telefonsentral.

Først vil vi fortelle litt om hva slags forkunnskaper vi bygger på og fortsette med å vise hvordan man installerer VoIP-serveren. Deretter setter vi opp serveren for en gruppe personer som kan ringe til hverandre, og etterpå konfigureres den for en gruppe som i tillegg kan ringe til og fra det eksisterende telefonnettet. Deretter vil vi se på installasjonen og konfigureringen av VoIP-klient på Android smarttelefoner og PCer. Til slutt vil vi gjøre en kort vurdering av sikkerhet og konkludere.

2 Bakgrunn

Vi antar at leseren har noe bakgrunn i hva Voice over Internet Protocol (VoIP) og Session Initiation Protocol (SIP) er, og har en viss oversikt over hvordan de fungerer. Dersom man trenger litt mer informasjon bør man starte med tekniske beskrivelser på Wikipedia, rapporter fra FFI[12], bøker[23, 15] og/eller publikasjoner som “An Introduction to Standards-Based VoIP”[28] og “Design and Implementation of a SIP-based VoIP Architecture”[26]. Kort fortalt er SIP en protokoll for å sette opp samtaler for en VoIP-tilbyder, og den er blitt den mest benyttede protokollen for slik signalering. VoIP kan sees på som en fellesbetegnelse på tjenester som tilbyr tale, og ofte utvidelser av dette som meldinger, video, konferansesamtaler, o.l., over Internett. Google voice[10] og Skype[21] er tjenester som blant annet overfører tale mellom brukerne og kan derfor karakteriseres som VoIP-tjenester. Men hverken Google Voice eller Skype benytter SIP for å sette opp samtalene sine.

Det har utviklet seg et enormt stort marked av telefonitjenester som kontinuerlig forsøker å tilpasse seg ny teknologi. Spesielt er det viktig å følge med på de muligheter som ligger i å tilby ekstra tjenester til dagens brukere av tale-, meldings- og videosamtaletjenester. Flere og flere brukere av disse tjenestene forblir “alltid” tilkoblet Internett, og både antallet og mengde tid tilkoblet Internett ser ut til fortsatt å øke[17].

I vår eksperimentering med å se på hvordan SIP og VoIP er implementert, har vi over en

lengre periode forsøkt mange implementasjoner og løsningsoppsett for dette. For å gjøre dette enklere for senere oppsett av lab-løsninger og enkle VoIP-sentraler, har vi derfor skrevet denne introduksjonen i hvordan “alle” kan gjøre dette.

En nyttig bieffekt av et enkelt lab-oppsett, er at man forklarer hvordan mindre grupper som småbedrifter og familier kan sette opp en enkel VoIP-tjeneste. Bruken av en egen VoIP-server muliggjør gratis kommunikasjon med alle andre som er registrert som brukere av samme tjeneste. Dette med forbehold om at man har tilgang på datatrafikk, for eksempel gjennom et trådløst aksesspunkt med sin mobiltelefon. Stort sett alle smarttelefoner har nå mulighet for VoIP-støtte, enten gjennom innebyggede applikasjoner eller gjennom nedlastbare apps.

Under eksperimenteringen fikk vi tilgang til en Raspberry Pi[19] ARM-basert datamaskin som koster cirka NOK 300 med lagringskort og trådløst nettverkskort. Eksperimentering med denne, samt etterfølgende erfaring med hvor mye av eksisterende gratis programvare som allerede fungerer for denne lavkostnads-datamaskinen, gjorde at vi var nødt til å forsøke å kjøre en VoIP-server på den. Vi har ikke gjort ytelsestester for hvor mange samtidige samtaler en slik minimalisert hardware klarer å prosessere, men det er utført tester[14] hvor man holdt ni personer i en konferansesamtale gjennom en Raspberry Pi med 256Mb med minne. For eksempel påstås det støtte av 5+ samtidige samtaler i VoIP-systemet *Incredible Pi* som leveres av IncrediblePBX[13].

Tilgangen på slike ferdige systemer som *Incredible Pi*, forenkler mye manuelt oppsett gjort i denne rapporten, men grafisk grensesnitt og bruk av database stjeler også ressurser fra datamaskinen. Vi har ikke inkludert oppsett og konfigurering av systemer gjennom integrasjoner som Incredible Pi og deres bruk av FreePBX, men holdt oss til manuelt oppsett av den mest benyttede programvaren for gratis VoIP-tjenester - Asterisk[3].

Testoppsettet som vil bli forklart i Kapittel 4, viser hvor enkelt Asterisk kan settes opp. Når lab-oppsettet ble ferdig og vi hadde et fungerende VoIP-system, ble det besluttet at dette burde dokumenteres tidlig slik at andre kan eksperimentere og kanskje også ta mer aktivt i bruk egne VoIP-systemer.

Når det gjelder sikkerhet og sikkerhetsutfordringer i VoIP og SIP har dette ikke vært en viktig del av dette eksperimentet, men vi har med en liten vurdering mot slutten. Ønsker man en bedre oversikt over generelle sikkerhetsutfordringer, bør man se på Geneiatakis et al. [9] som inneholder en oversikt over sikkerhetsmekanismene i SIP. McGann og Sicker [16] og Hagalisletto og Strand [22] tar for seg sikkerhetsutfordringer og angrep på SIP. I tillegg er det gjort flere sikkerhetsanalyser som for eksempel Gupta og Shmatikovs [11] fra 2007. Butcher et al. [5] presenterer en forenklet og grei oversikt over sikkerhetsutfordringer og mulige mottiltak i en VoIP-infrastruktur.

3 Oppsett av VoIP

Vi har forsøkt å gjøre dette så enkelt som mulig. Lab-oppsettet, som også vil kunne fungere som en “hjemmesentral” for en familie eller mindre bedrift, består av:

- En datamaskin som kan kjøre Linux. Dette kan være en gammel PC, men som minimumskrav holder det med en Pentium 4 eller nyere, 512Mb RAM og 10Gb harddisk. Vi benyttet også en Raspberry Pi [19] med 256Mb minne, et minnekort på 4Gb som benyttes som lagringsplass og et trådløstkort.
- En Linux-installasjon. For både PC og Raspberry Pi ble det benyttet en Debian “squeeze” minimum installasjon. Denne er klar til bruk med all programvaren som vi har benyttet. Det er kun behov for nedlasting og konfigurasjon.
- Asterisk[3] er state-of-the-art for VoIP programvare. Denne benyttes på alt fra større kommersielle telefonisentraler til små gruppesentraler lik den som vi skal beskrive her.
- En forbindelse til Internett for PC-en dersom denne skal benyttes også på reise, hvilket er oftest hvor besparelsene kommer inn.

3.1 Oppsett av Linux

Vi vil kort beskrive oppsett av Linux på både PC og på Raspberry Pi, ettersom de aller fleste enkelt vil kunne finne en eldre PC å kjøre programvaren på og dermed slipper å kjøpe inn en ny enhet.

Man kan benytte ferdigløsninger som IncrediblePBX[13] og *AsteriskNOW*. *AsteriskNOW* er en ferdigversjon av Asterisk som installerer et Linux system med Asterisk og grafisk brukergrensesnitt. Eventuelt kan man benytte *3CX for Windows*[1] som finnes i en gratis versjon med redusert funksjonalitet.

Fordelene med å installere Linux selv og gjøre konfigurasjonen manuelt, er høyere grad av fleksibilitet for spesialtilpasninger. Dette er viktig for lab-bruk.

3.1.1 På PC

Mange installasjonsveiledninger finnes til Linux. Vi valgte Debian[7] “squeeze” edition i386 ettersom denne har et enkelt installasjonsgrensesnitt, samt at den fungerer på stort sett alle eldre PC-er.

Installasjon av Debian squeeze kan gjøres på følgende måte:

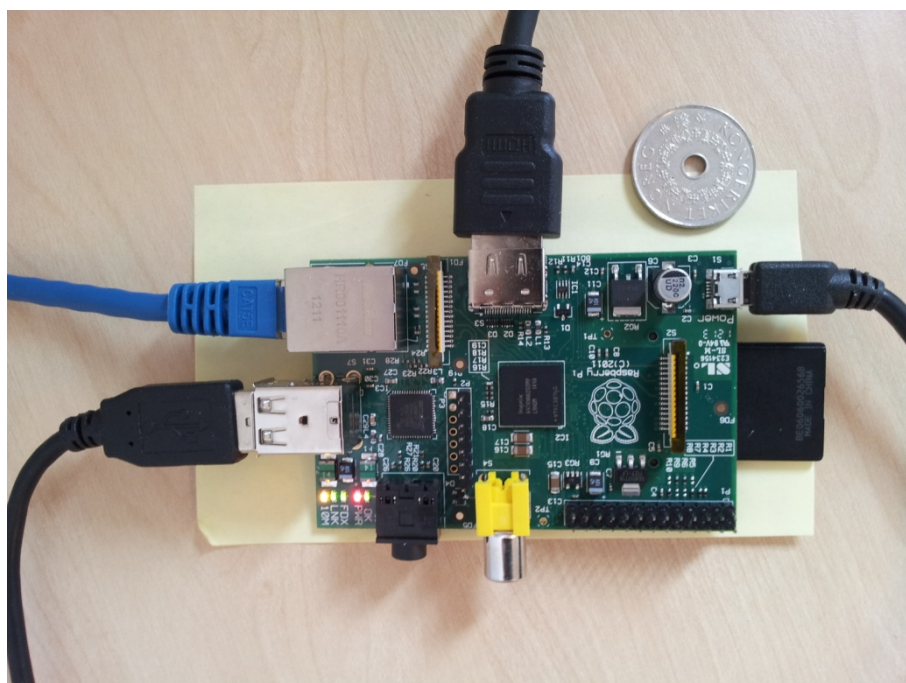
1. Last ned Debian squeeze 6.0.5 eller senere. Brenn CD eller lag USB-minnepinne for installasjon. Boot fra CD/USB.
2. Velg minimumsinstallasjon og automatisk partisjonering av harddisk.
3. Velg alt slik som det foreslås med mindre man skal ha flere operativsystemer på samme maskin. I få fall regner vi med at man klarer å sette opp Debian selv.

4. Når minimumsinstallasjonen er gjennomført og man har restartet maskina logger man inn som den nye brukeren man lagde.

3.1.2 På Raspberry Pi

På hjemmesidene til Raspberry Pi organisasjonen finner man ferdige installasjonsimage for flere Linuxbaserte operativsystemer. Debian “squeeze”, Arch Linux ARM[2] og Qt-on-Pi[18] var de distribusjonene som det ble referert til når dette ble skrevet. Vi har benyttet oss av Debian “squeeze” ettersom dette gjør resten av oppsettet tilnærmet identisk for både PC og ARM-kort.

Etter de initielle eksperimentene og under skriving av denne rapporten, ble det også gjort tilgjengelig en variant av Debian som var spesialtilpasset til Raspberry Pi, kalt Raspbian[20]. Denne fungerer på samme måte som beskrevet under og har en del optimalisering i forhold til hardwaren som er benyttet.



Figur 3.1 Størrelsen til Raspberry Pi gjør den perfekt for enkle og små serveroppgaver.

Først må vi laste ned en ferdig installasjon av Debian for Raspberry Pi og sjekke at den ble korrekt nedlastet. Ved skriving av denne rapporten var det versjonen ‘debian6-19-04-2012.zip’ som var siste tilgjengelige versjon med en sjekksum¹ - 1852df83a11ee7083ca0e5f3fb41f93ecc59b1c8 - som ble kontrollert som vist under:

```
user:~/Downloads$ ls -l debian6-19-04-2012.zip
```

¹SHA1 sjekksommen var lik ‘1852df83a11ee7083ca0e5f3fb41f93ecc59b1c8’, men dette lå på samme side som nedlastingen så sikkerhetsmessig har den ingen betydning. Vi benytter den for å sjekke at vi har lastet ned en komplett kopi.

```

-rw-r--r-- 1 user user 464583238 May  4 16:37 debian6-19-04-2012.zip
user:~/Downloads$ shasum debian6-19-04-2012.zip
1852df83a11ee7083ca0e5f3fb41f93ecc59b1c8  debian6-19-04-2012.zip
user:~/Downloads$ unzip debian6-19-04-2012.zip
Archive:  debian6-19-04-2012.zip
  creating: debian6-19-04-2012/
    inflating: debian6-19-04-2012/debian6-19-04-2012.img.sha1
    inflating: debian6-19-04-2012/debian6-19-04-2012.img
user:~/Downloads$ cd debian6-19-04-2012/
user:~/Downloads/debian6-19-04-2012$ shasum debian6-19-04-2012.img
e1a6f8695be719c83ac7f2595220c36ecf0a58cf  debian6-19-04-2012.img
user:~/Downloads/debian6-19-04-2012$ cat debian6-19-04-2012.img.sha1
e1a6f8695be719c83ac7f2595220c36ecf0a58cf  debian6-19-04-2012.img

```

Deretter må vi skrive til minnekortet. Installering av Debian på minnekortet gjøres ved å benytte en Linux-maskin og kommandoen 'dd', eller en Windows-maskin og programmet Win32DiskImager[6].

På vår Linux-PC var minnekortskriveren tilgjengelig som en enhet kalt '/dev/mmcblk0'. Denne finner man lettest ut av ved å sette inn et minnekort, kjøre kommandoen 'df' for å se hvilken enhet som ble montert, og til slutt fjerne siste del av navnet - oftest '...p1'. Vår device het '/dev/mmcblk0p0' og enheten vi skulle skrive til ble da '/dev/mmcblk0'. På noen maskiner dukker den også opp som '/dev/sdc1', '/dev/sdd1', eller lignende og da er det viktig å benytte henholdsvis '/dev/sdc' eller '/dev/sdd'. Dette er godt forklart på installasjonssidene til Raspberry Pi.

MERK! Vær100% sikker på at man ikke skriver over harddisken på PC-en! Dersom man tror man skal skrive til /dev/sda, /dev/hda,... eller liknende, må man dobbeltsjekke og trippelsjekke at dette ikke er harddisken som datamaskinen benytter til sine filsystemer (bruk kommandoen 'df').

```

$ eject /dev/mmcblk0p1
$ sudo dd if=debian6-19-04-2012.img of=/dev/mmcblk0 bs=64k
29754+1 records in
29754+1 records out
1950000000 bytes (2.0 GB) copied, 277.696 s, 7.0 MB/s
$ sync

```

Deretter flyttes minnekortet over i Raspberry Pi og denne skrus på for første gang med denne disken. Når man starter opp Raspberry Pi ser det ut som vist i Figur 3.3.

Etter at datamaskinen har bootet kan man logge inn som bruker "pi" med passord "raspberry" og man skifter selvfølgelig umiddelbart dette passordet til noe nytt! Kjente passord og



Figur 3.2 Etter at Raspberry Pi er installert, trenger den kun strøm og nettverk.

standardpassord ved installasjon er et av de mest sårbare sikkerhetshull/svakheter ved enhver nyinstallasjon av en datamaskin.

```
pi@raspberrypi:~$ passwd
Changing password for pi.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

4 Eksperimenter

Flere forskjellige eksperimenter og oppsett ble testet, men det var flere felles elementer. Alle brukere var lokale og vi la manuelt til brukere som beskrevet i kapitlet om Asterisk. Vi benyttet kun serveren som en frittstående Asterisk VoIP server. Alle kobler opp mot denne. Denne må ha egen fast IP på Internett som alle kan nå². Det er derfor svært viktig at man tester at maskinen kan nås fra Internett og at ingen brannmur, NAT-gateway, eller annet stopper trafikken til/fra serveren. Figur 4.2 inneholder litt beskrivelse av hvordan dette kan foregå og hvilke porter som må være åpne, men vi forutsetter her at serveren står koblet rett på Internett. Det finnes mange oppskrifter på hvordan man setter opp brannmur/NAT/DMZ til å slippe igjennom

²Dersom den kun er ment å være for internt bruk kan man ha den på en intern IP-adresse, men da blir det tilsvarende en intern telefonsentral.

```
Starting NFS common utilities: statd.  
Cleaning up temporary files....  
Setting up ALSA...done (none loaded).  
fuse init (API version 7.17)  
Setting console screen modes.  
Skipping font and keymap setup (handled by console-setup).  
Setting up console font and keymap...done.  
Setting sensors limits.  
Setting kernel variables ...done.  
INIT: Entering runlevel: 2  
Using makefile-style concurrent boot in runlevel 2.  
Network Interface Plugging Daemon...skip eth0...done.  
Starting NFS common utilities: statd.  
Starting portmap daemon...Already running..  
Starting enhanced syslogd: rsyslogd.  
Starting system message bus: dbus.  
Starting Hardware abstraction layer: hald/etc/dahdi/system.conf not found.  
Starting DNS forwarder and DHCP server: dnsmasq.  
Starting Asterisk PBX: asterisk.  
  
Starting periodic command scheduler: cron.  
Starting NTP server: ntpd.  
Starting OpenBSD Secure Shell server: sshd.  
Starting internet superserver: xinetd.  
My network IP address is 172.27.27.27  
  
Debian GNU/Linux 6.0 raspberrypi tty1  
raspberrypi login: _
```

Figur 3.3 Raspberry Pi med loginprompt. Vises her etter oppsett av nettverk og installasjon av Asterisk.

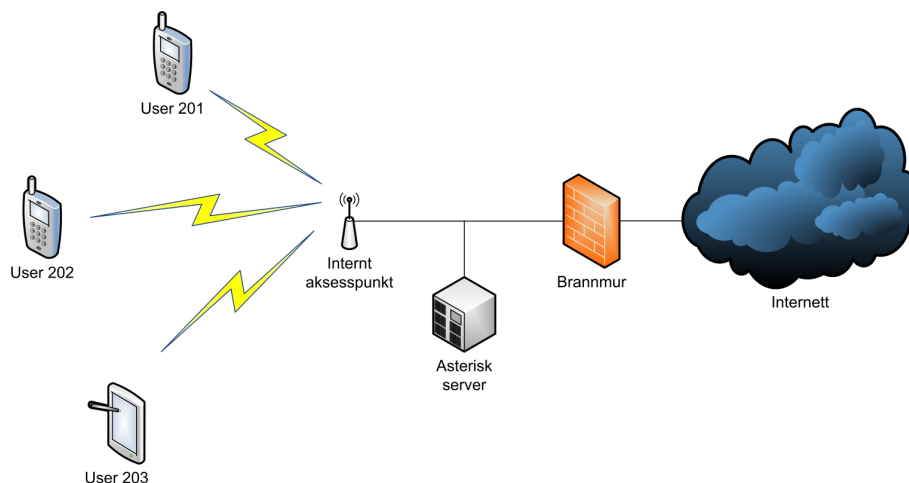
SIP-trafikk og vi gjennomgår ikke dette her. Det som er viktig er å stanse alle tjenester på maskina som ikke er i bruk.

Eksperimenter som forklares i denne rapporten er:

- 1. Intern sentral - kun interne brukere.** Her er serveren frittstående og kun for lokale brukere hvor alle kan ringe alle. Den er tilgjengelig for alle som skal ha tilgang til tjenesten, og har ikke tilgang til å ringe ut eller motta samtaler fra eksterne telefontjenester.
- 2. Eksternt tilgjengelig sentral - interne brukere med eksternt tilgang.** Her er serveren koblet mot en eksternt tjenesteleverandør som benyttes for telefoni til/fra vanlig telefoninettet. Her kan alle brukerne ringe hverandre, men de kan også nås på vanlige telefonnummer gjennom ett eller flere eksterne telefonnummer. I tillegg kan denne eksterne tjenesteleverandøren benyttes til å ringe ut til vanlige brukere av telefoninettet.

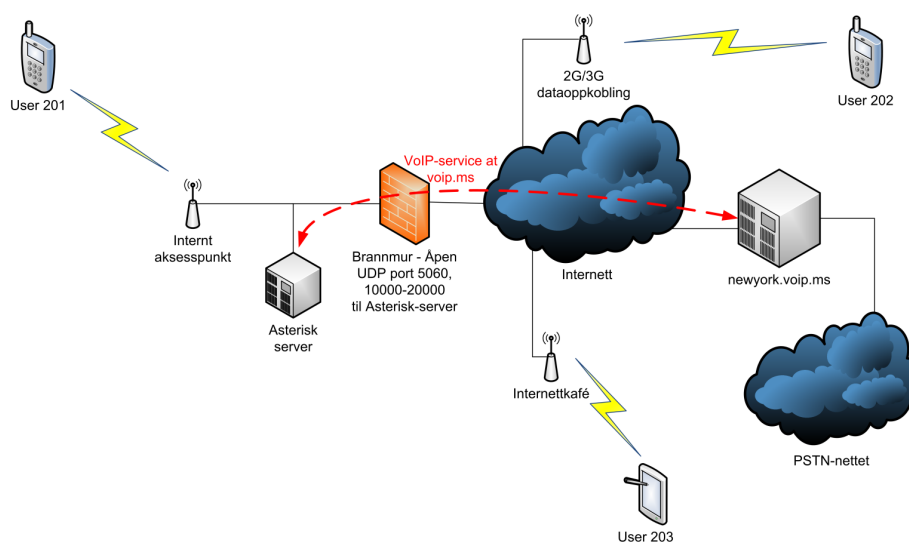
4.1 Testoppsett 1 - Intern sentral - Kun for interne samtaler mellom brukerne

I Figur 4.1 viser vi hvordan lab-oppsettet har vært i testoppsettet "Intern sentral". Her har brukerne kun tilgang på VoIP-server gjennom internt nettverk, og kan kun ringe mellom hverandre. Brukerne har vanlig tilgang på nettverkstjenester på smarttelefonene sine, men VoIP er nå ikke konfigurert til å ha kontakt med det vanlige telefonnettet (Public Switched Telephone Network - PSTN).



Figur 4.1 Intern Sentral

4.2 Testoppsett 2 - Eksternt tilgjengelig sentral - Intern VoIP-server med tilgang fra utsiden og innsiden



Figur 4.2 Eksternt tilgjengelig sentral

I Figur 4.2 viser vi hvordan lab-oppsettet har vært i testoppsettet “Eksternt tilgjengelig sentral”. Her har brukerne tilgang på VoIP-server gjennom internt nettverk og gjennom Internett. De kan ringe både til hverandre, motta samtaler fra resten av telefonnettet, samt ringe til alle andre også.

Her kan både “User 203” og “User 202” logge seg på Asterisk-serveren selv om de er utenfor hjemmesonen, samt at Asterisk-serveren har tilgang til PSTN-nettet gjennom en tjenesteleverandør med adresse “newyork.voip.ms”.

4.3 Installasjon og oppsett av Asterisk

Her følger en gjennomgang av oppsettet av Asterisk-server på en datamaskin som kjører Debians versjon av GNU/Linux. Vi starter med installasjon og nettverksoppsett, deretter brukere, telefonnummer og konfigurasjonsfiler, og til slutt litt om monitorering og kontroll med Asterisk mens den kjører på serveren.

4.3.1 Installasjon

Etter at vi har skiftet passord på standardsbrukeren (kommandoen 'passwd') installerer vi SSH-server som tillater oppkobling for innlogging på datamaskinen over nettet. Vi anbefaler å installere og benytte kun sertifikatbasert innlogging over SSH ved å sette inn linjen "PasswordAuthentication no" i filen /etc/ssh/sshd_config, og opprette et privat/offentlig nøkkelpar for innlogging. Se Appendix A for mer detaljer ved installasjon og oppsett av openssh-server.

Vi må også gi Asterisk-serveren en fast IP-adresse. Vi benytter gjennom denne rapporten IP-adressen 172.27.27.27 for Asterisk-serveren og har satt den på et eget subnet med DNS og gateway på 172.27.27.1. Vi legger derfor inn følgende nettverksoppsett i '/etc/network/interfaces' (i stedet for linjen med 'iface eth0 inet dhcp'):

```
auto eth0
iface eth0 inet static
address 172.27.27.27
netmask 255.255.255.0
broadcast 172.27.27.255
gateway 172.27.27.1
dns-nameservers 172.27.27.1
```

Deretter tar vi nettverket ned og opp igjen og sjekker at vi har nettverkskontakt:

```
pi@raspberrypi:~$ sudo ifdown eth0
pi@raspberrypi:~$ sudo killall dhclient
pi@raspberrypi:~$ sudo ifup eth0
pi@raspberrypi:~$ ping www.google.com -c 1
PING www.l.google.com (173.194.71.103) 56(84) bytes of data.
64 bytes from lb-in-f103.1e100.net (173.194.71.103):\
  icmp_req=1 ttl=48 time=26.6 ms

--- www.l.google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 26.641/26.641/26.641/0.000 ms
```

Da er vi klar for å installere Asterisk på serveren. En nokså enkel oppgave når man benytter seg av Debian-distribusjonen:

```
pi@raspberrypi:~$ sudo apt-get update
pi@raspberrypi:~$ sudo apt-get install asterisk
```

Her må man under installasjonen sette inn korrekt landskode for telefoni “ITU-T telephone code”, som for Norge er '47'.

```
...
Setting up asterisk (1:1.6.2.9-2+squeeze5) ...
Adding system user for Asterisk
Adding user 'asterisk' to group 'dialout' ...
Adding user asterisk to group dialout
Done.
Adding user 'asterisk' to group 'audio' ...
Adding user asterisk to group audio
Done.
Starting Asterisk PBX: asterisk.
pi@raspberrypi:~$
```

Asterisk kjører nå på serveren, men er ikke satt opp med hverken brukere eller telefonnummer. Vi er klare for å starte med å konfigurere Asterisk til vårt enkle bruk.

4.3.2 Konfigurasjon

Alle konfigurasjonsfilene vi trenger for å sette opp Asterisk ligger i katalogen '/etc/asterisk'. De eneste filene vi trenger å se på for å få et enkel VoIP-server opp og klar for å formidle samtaler er:

sip.conf³ som er konfigurasjonsfilen for Asterisk sine SIP-kanaler, både innkommende og utgående.

extensions.conf⁴ som er konfigurasjonsfila for nummeroppsett og hvordan forbindelser kobles av Asterisk-serveren.

For å få til en minimumsinstallasjon har vi satt opp et system med tre brukere mot denne sentralen

³http://www.asterisk.org/doxygen/trunk/Config_sip.html og <http://www.voip-info.org/wiki/view/Asterisk+config+sip.conf>

⁴http://www.asterisk.org/doxygen/trunk/Config_ext.html og <http://www.voip-info.org/wiki/view/Asterisk+config+extensions.conf>

4.4 Brukere og telefonnummer

Vi har definert tre brukere av systemet: “user201”, “user202” og “user203”. Disse kan nås på henholdsvis internnummer 201, 202 og 203. Dette medfører at dersom man “tar av røret” for VoIP-telefonen og slår dette nummeret, skal man komme direkte til den andre personen dersom denne også er “online”. Dersom man har en eksternt linje og slår ’0’ først, skal man få ringt det etterfølgende nummeret til “utsiden” av Asterisk-serveren.

Hoveddelene av oppsettet for ’sip.conf’ og ’extensions.conf’ vises nedenfor, mens de komplette versjonene er gitt i Appendix Appendiks A og Appendix Appendiks B. Vi tar for oss de samme oppsettene som er forklart i detalj tidligere:

- Intern sentral
- Eksternt tilgjengelig sentral.

Det er det siste alternativet som er mest realistisk for en vanlig småbedrift eller for en familie som vil kombinere og utnytte sine telefonressurser. Her vil de benytte VoIP ved å ringe hverandre og kun hverandre. Eller de kan i tillegg ha ett eller flere offentlig tilgjengelige telefonnummer for å bli kunne motta innkomne samtaler, samt et valgfritt antall linjer til å ringe eksternt, det vil si ut på det gamle (“vanlige”) telefonnettet. Vi vil ta for oss begge alternativene som har hver sine utfordringer i oppsett og konfigurasjon.

4.4.1 Intern sentral - kun intern ringing mellom brukerne

```
;  
; sip.conf for intern sentral  
;  
[general]  
port=5060  
bindaddr=0.0.0.0  
context=1242domain  
tos=0x18  
nat=yes  
externip=128.39.x.y  
disallow=all  
allow=all  
alwaysauthreject=yes  
  
[user201]  
regexten=201  
type=friend  
host=dynamic
```

```

context=1242domain
secret=12341234
callerid="User 201"
dtmfmode=rfc2833
nat=yes
;directmedia=no
disallow=all
allow=all
registertrying=yes

[user202]
regexten=202
...

[user203]
regexten=203
...

```

Som vist over i 'sip.conf' er det mange konfigurasjonsparametre. Vi forsøker her å gi en kort forklaring på parametrene. Områdene "[user202]" og "[user203]" er lik "[user201]" bare vi bytter ut '201' med hhv. '202' og '203'. I feltet [general] med generelle konfigurasjonssettinger finner vi:

- "port=5060" - forteller at vi skal lytte på registreringsmeldinger på port 5060
- "bindaddr=0.0.0.0" - bind lyttesocketet mot alle interface og IP-adresser på serveren
- "context=1242domain" - hva heter dette domenet - dette benyttes også i andre konfigurasjonsfiler (som for eksempel 'extensions.conf') og skal være likt for alle brukerne
- "tos=0x18" - for å sette bitverdier i "Type of Service". Hver bit har sin verdi - 0x10=low delay, 0x08=high throughput, 0x04=high reliability, 0x02=ECT bit set, 0x01=CE bit set. Hvilket betyr at vår innstilling er "low delay" og "high throughput".
- "nat=yes" - vil (kunne) være bak en brannmur som kjører Network Address Translation (NAT)
- "externip=128.39.x.y" - dette er labens eksterne IP-adresse (utenfor NAT-brannmuren)
- "disallow=all" - ikke tillat noen som standard, tillat i preferert rekkefølge de som følger med "allow="
- "allow=all" - tillat alle
- "alwaysauthreject=yes" - alltid svar med samme feilmelding selv om brukernavn ikke stemte eller det var passordfeil eller en annen feil. Bedre sikkerhet mot scanning av SIP-konti.

I feltet "[user201]" (og de andre to områdene for de andre brukerne) har vi følgende innstillin-

ger:

- “[user201] ” - login-navn på denne brukeren
- “regexten=201 ” - hvilket nummer/extension som skal benyttes av denne brukeren
- “type=friend ” - type kan være peer, user eller friend. “peer” er kun for faste IP/port installasjoner, “user” er kun for innkommende samtaler, “friend” vil være en kombinert “user” og “peer”.
- “host=dynamic ” - benyttes ved type=friend, og gjør at klienten må registrere seg for å kunne motta samtaler fra vår Asterisk server
- “context=1242domain ” - hvilken sammenheng i ’extensions.conf’ vi skal benytte for denne brukeren
- “secret=12341234 ” - passordet ved innlogging/registrering
- ‘callerid=”User 201”’ - Hva som skal vises når brukeren ringer en annen
- “dtmfmode=rfc2833 ” - hva slags type Dial Tone Multi Frequency (DTMF) som skal benyttes. Kan være “inbound”, “outbound”, “rfc2833” eller “info”. Dette er standard innstilling⁵.
- “nat=yes ” - vil sannsynligvis være bak en NAT-gateway/brannmur
- “;directmedia=no ” - er standard “yes” og vil dermed forsøke å koble direkte mellom klientene. Dersom man vet at man er bak en NAT vil det sannsynligvis være bedre å kommentere inn igjen denne innstillingen slik at den settes til “no”. Het tidligere “canreinvite”.
- “disallow=all ” - kodingsinnstillinger for lyd for denne brukeren (som under [general], men overstyres her)
- “allow=all ”
- “registertrying=yes ” - send en “100 Trying” melding når SIP-klienten registrerer seg.

Innstillingene i ’extensions.conf’ er en god del enklere når vi kun har samtaler mellom de interne brukerne. Det er lagt til en echo-test konfigurasjon på internnummer 299 slik at man kan teste om en SIP-telefon fungerer mot sentralen ved å ringe dette nummeret.

```
;
; extentions.conf for intern sentral
;
[general]
static=yes
writeprotect=no

[1242domain]
; First a playback/echo on internal 299
```

⁵Se <http://support.draytek.net.au/index.php?Knowledgebase/Article/View/124/0/what-dtmf-mode-should-i-use> for mer informasjon om DTMF-mode.

```
exten => 299,1,Playback(demo-echotest)
exten => 299,2,Echo
exten => 299,3,Playback(demo-echodone)
; This is the users
exten => 201,1,Dial(SIP/user201,20,tr)
exten => 202,1,Dial(SIP/user202,20,tr)
exten => 203,1,Dial(SIP/user203,20,tr)
exten => 203,2,Dial(SIP/user201,20,tr)
```

Filen 'extensions.conf' skal som tidligere forklart sette opp koblingene mellom telefonnummer og brukere.

Først har vi generelle konfigurasjonsparametre i "[general]".

- "static=yes" - sier at fila ikke skal overskrives automatisk dersom noen forandrer på extensions direkte inne i Asterisk. Kun "yes" er implementert.
- "writeprotect=no" - forteller om man kan overskrive ringeoppsettet under bruk av Asterisk. Dette skjer gjennom å skrive "dialplan save" i konsollet til Asterisk.

Vi legger merke til "[1242domain]" som inneholder hvordan de forskjellige nummerene skal benyttes dersom de blir ringt av noen av klientene. En linje er på formen:

```
exten => nummer,prioritet,applikasjon(argument1,argument2,...)
```

hvor 'nummer' er internnummer eller en søkestreng som innkommende ringenummeret skal matche, 'prioritet' spesifiserer rekkefølge aksjonene for 'nummer' dette er, og 'applikasjon(arg1, arg2, ...)' forteller hva som skal utføres.

I våre fire nederste linjer vil man ved å ringe 201 komme til bruker "user201", 202 til "user202" og 203 til "user203". Siste linje forteller at dersom "user203" ikke er pålogget, avviser å svare, eller ikke svarer innen 20 sekunder (argument nummer 2 - prioritet) blir samtalen forsøkt koblet opp til "user201".

De tre øverste 'exten' linjene forteller at man ved å ringe '299' vil få en beskjed om at en ekko-test vil starte - 'Playback(demo-echotest)', deretter starter testen - 'Echo', og til slutt avsluttes testen med en beskjed - 'Playback(demo-echodone)'.

4.5 Eksternt tilgjengelig sentral

For oppsettet med eksternt tilgjengelig sentral, har vi kjøpt inn to eksterne telefonnummer fra eksternt leverandør, VoIP.ms[17], som er en av mange operatører som nå selger tjenester for oppkobling av interne/private VoIP-servere. De to nummerene for bruk i denne laben er:

- Norsk nummer: +47 67555307 som er registrert som et Akershus/Lørenskog nummer
- Amerikansk nummer : +1 703-555-4512 som er registrert i Dulles-området i Virginia, USA
- To andre nummer er også innkjøpt for bruk inn i systemet, men disse benyttes kun for eksperimentering og flyttes til stadighet til de forskjellige lokasjonene.

Username	Description	Protocol	Authentication Type	CallerID Number	Internal Extension
141188_loc2	Location 2	SIP	Register	+1703-555-4512	none
141188_loc3	Location 3	SIP	Register	+4767-555-307	none

Figur 4.3 Lokasjonene hos tjenesten voip.ms for henholdsvis det amerikanske og det norske kontaktpunktet.

For tilgang til PSTN-nettet og mulighetene til å ringe ut gjennom voip.ms sine tjenester, har vi satt opp flere bruker/passord-kombinasjoner. En bruker eksisterer for hver lokasjon som skal kobles opp mot deres tjeneste. Hovedlokasjonen er for eksperimentell bruk, men lab-opsettene benytter brukerid lik 141188_loc2 og 141188_loc3 som vist i Figur 4.3. Passordet for alle oppkoblinger er midlertidig satt til 'SECretPassword21', slik at det er lett å identifisere i konfigurasjonsfilene.

Description	Number	Options	Routing	Note	VM	TO	POP
DULLES, VA	703-555-4511	cnam	[SIP] Main Account	none	90	New York	
DULLES, VA	703-555-4512	cnam	[SIP] 141188_loc2	none	60	New York	
NORWAY AKERSHUS	476-7-555-303	cnam	[SIP] Main Account	none	90	New York	
NORWAY AKERSHUS	476-7-555-307	cnam	[SIP] 141188_loc3	none	60	New York	

Figur 4.4 Oppsett over telefonnummer for en spesifikk lokasjon.

Lokasjon 2 har det amerikanske nummeret +1 703-555-4512 koblet opp mot seg og lokasjon 3 har det norske nummeret +47 67555307 koblet opp mot seg som vist i Figur 4.4.

Merk at det er mange måter å sette opp brukere og tjenester på i Asterisk. Vi vil kun ta for oss mindre systemer og dermed holde oss til filbaserte konfigurasjoner. Databaseløsninger er selvfølgelig å foretrekke for større enheter, men da må man se på brukerdokumentasjonen⁶ til Asterisk.

Merk at selv om vi har tilknyttet oss et norsk telefonnummer for å bli oppringt på, vil våre eksterne forbindelser ut fra vårt VoIP-nett bli lokalisert ut på telefonnettet i USA - nærmere bestemt i New York i vårt eksempel.

Vi vil her ta for oss forskjellene fra intern sentral mot et oppsett for en eksternt tilgjengelig server som har norsk telefonnummer +47 67555307.

```
;
; sip.conf for eksternt tilgjengelig sentral
;
[general]
...
context=voipmscontext
...
register => 141188_loc3:SEcRetPassword21@newyork.voip.ms:5060
...

;
; config at location 3
;
[voipms]
context=voipmscontext
host=newyork.voip.ms
secret=SEcRetPassword21
type=peer
username=141188_loc3
disallow=all
allow=ulaw
allow=g729
fromuser=141188_loc3
trustpid=yes
sendrpid=yes
insecure=invite
```

⁶<http://www.asterisk.org/docs>

```
nat=yes

[user201]
..
context=voipmscontext
callerid="User 201" <+4767555307>
...
```

Som vist over i 'sip.conf' er det ikke mange forskjeller fra en intern server. Vi har satt opp et nytt VoIP-domene som vi har kalt "voipmscontext" (etter tjenesteleverandøren voip.ms) og dette benyttes også i 'extensions.conf'. Det er viktig at alle brukere også settes opp i "voipmscontext" og at man forandrer "callerid" til det som gjelder hver enkelt bruker, som vist i eksempelet for "user201".

Andre viktige forskjeller er at vi registrerer oss mot en ekstern leverandør av VoIP-tjenester og definerer et nytt domene/context som kobler sammen informasjon som vist i kommentarene under:

- "context=voipmscontext" - nytt context som skal benyttes nå i 'extensions.conf'
- "register =>141188_loc3:SECretPassword21@newyork.voip.ms:5060" logg inn på serveren, newyork.voip.ms, port 5060, med brukernavn "141188_loc3" og passord "SECretPassword21". Vi har tidligere vist at vi har tilgjengelig et norsk nummer, 67555307 på denne brukerens innlogging hos voip.ms.

For vårt nye domene, "[voipms]", har vi noen nye innstillinger for utgående samtaler:

- "context=voipmscontext" - samme context som andre
- "host=newyork.voip.ms" - maskin som man skal koble seg opp mot
- "secret=SECretPassword21" - passord for oppkobling
- "type=peer" - vi kobler opp mot en peer
- "username=141188_loc3" - brukernavn, nå benyttes defaultuser
- "disallow=all" - som tidligere...
- "allow=ulaw"
- "allow=g729"
- "fromuser=141188_loc3" - brukeren å logge inn med....
- "trustpid=yes" - stol på Remote-Party-ID ved samtaler
- "sendrpid=yes" - Remote-Party-ID headeren skal sendes
- "insecure=invite" - ikke krev autentisering ved innkommende INVITE-meldinger
- "nat=yes" - ofte bak en NAT

Under brukeroppsettet til "user201", må vi også forandre context til nytt domene, samt at vi kan legge inn et gyldig eksternt tilgjengelig telefonnummer under "callerid":

- context=voipmscontext - ny context for denne brukeren
- callerid="User 201"<+4767555307>- har fått inn telefonnummer som callerid og som dermed vil bli vist hos mottaker dersom den støtter mottak og visning av dette.

For 'extensions.conf' blir forandringene litt mer kompliserte ettersom vi nå skal ha muligheten til å ringe eksternt i tillegg til å rute innkommende samtaler korrekt.

```

;
; extentions.conf for Eksternt tilgjengelig sentral
;
[general]
static=yes
writeprotect=no

[1242domain]
...som tidligere...

[voipmscontext]
include => 1242domain
include => voipms-inbound
include => voipms-outbound

[voipms-inbound]
exten => 4767555307,1,Dial(SIP/user201,20,tr)
exten => 4767555307,2,Dial(SIP/user202,20,tr)
exten => 4767555307,3,Dial(SIP/user203,20,tr)

[voipms-outbound]
exten => _1NXXNXXXXXX,1,Dial(SIP/${EXTEN}@voipms)
exten => _1NXXNXXXXXX,n,Hangup()
exten => _NXXNXXXXXX,1,Dial(SIP/1${EXTEN}@voipms)
exten => _NXXNXXXXXX,n,Hangup()
exten => _011.,1,Dial(SIP/${EXTEN}@voipms)
exten => _011.,n,Hangup()
exten => _00.,1,Dial(SIP/${EXTEN}@voipms)
exten => _00.,n,Hangup()

```

Vi kan legge merke til at vi ved å innføre context [voipmscontext], så kunne vi fortsatt ha konfigurasjonen under 1242domain som tidligere og inkludere denne sammen med de nye områdene [voipms-inbound] og [voipms-outbound]. På denne måten *kan* man se for seg noen

interne brukere som kun kan ringe internt mellom hverandre, og noen som også kan ringe eksternt.

[voipms-inbound] definerer hvordan innkommende samtaler skal routes. Av linjene og forklaringsene fra tidligere oppsett, ser vi at innkommende samtaler til det norske nummeret, 67555307 først vil ringe 20 sekunder hos "user201", deretter 20 sekunder hos "user202", og til slutt 20 sekunder hos "user203".

I [voip-outbound] er det en del mer å sette opp. Først må vi forstå hvordan mønstergjenkjenning av nummer foregår, samt huske på at det er en amerikansk leverandør av disse tjenestene. Dersom vi benytter et amerikansk telefonnummer og ringer/mottar samtaler med denne, er tabellen som vist i [voip-outbound], hvor mønstergjenkjenningen er som følger:

- " _ " forteller at tallene i telefonnummeret begynner etter denne "markøren"
- X er ethvert siffer 0-9
- Z er ethvert siffer 1-9
- N er ethvert siffer 2-9
- ". " passer med alt (Det vil si at "_011." betyr at strengen starter med "011" og hva som helst som kommer etterpå er OK. Dette er den vanlige måten man ringer ut av USA på.

Ettersom vi har ønske om å ringe norske nummer med åtte siffer direkte kan vi sette opp noen innstillinger for eksempel som følger:

- "exten =>_82XXXXXX,1,Congested ikke ring 82x xxxxx nummer..."
- "exten =>_XXXXXXXX,1,Dial(SIP/01147\${EXTEN}@voipms)" - alle åttesifrede forsøk på å ringe vil gå ut av USA (011) og til Norge (47) før det ringte åttesifrede nummeret benyttes.

Men dette er en lite hensiktsmessig måte å ta i bruk en slik tjeneste på, ettersom vi ringer norske nummer fra USA. I dette eksempelet har vi benyttet oss av et amerikansk nummer for både innkommende og utgående samtaler og vi betaler derfor utenlandstakst (for tiden \$0.10/min fra USA til Norge) for alle samtaler både ut og inn.

Vi vil i senere rapporter se på beste mulige måter å ringe til og fra PSTN-nettet avhengig av lokasjonen til brukeren og VoIP-serveren.

4.5.1 Monitorering og debugging

Ved å starte monitorering og debugging i Asterisk kan vi følge hva som skjer på serveren, samt få ut informasjon om hvem som er tilkoblet, hvem som ringer, hvem som snakker sammen m.m. For å starte Asterisk med mye debuggingsinformasjon til konsollet benyttes kommando-linjeopsjon '-v'. Jo flere v'er jo mer debuggingsinformasjon får vi ut. Opsjonen -r er for å koble oss til en allerede kjørende asteriskprosess.

```

$ sudo asterisk -rvvvv
Asterisk 1.6.2.9-2+squeeze5, Copyright (C) 1999 - 2010 Digium,\
  Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show\
  warranty' for details.
This is free software, with components licensed under the GNU\
  General Public
License version 2 and other licenses; you are welcome to\
  redistribute it under
certain conditions. Type 'core show license' for details.
=====
== Parsing '/etc/asterisk/asterisk.conf': == Found
== Parsing '/etc/asterisk/extconfig.conf': == Found
Connected to Asterisk 1.6.2.9-2+squeeze5 currently running\
  on raspberrypi (pid = 6803)
Verbosity is at least 4

```

For å laste konfigurasjonsfilene på nytt benyttes kommandoen 'reload':

```

raspberrypi*CLI> reload
.... Masse debuggingsinformasjon ....

====MERK OPPDATERING HER=====

```

For å se maskiner vi har oppkobling mot benytter vi 'sip show registry':

```

raspberrypi*CLI> sip show registry

====MERK OPPDATERING HER=====

```

For å se brukere registrert i systemet benytter vi 'sip show users':

```

raspberrypi*CLI> sip show users
Username  Secret    Accountcode  Def.Context  ACL  NAT
user201   12341234          1242domain  No  Always
user203   12341234          1242domain  No  Always
user202   12341234          1242domain  No  Always

```

Og for å se hvilke brukere og eksterne stasjoner som er oppkoblet benytter vi 'sip show peers':

```
raspberrypi*CLI> sip show peers
Name/username      Host           Dyn Nat ACL Port  Status
user201/user201 172.27.27.103 D   N     35502 Unmonitored
user202/user202 172.27.27.103 D   N     45744 Unmonitored
user203/user203 172.27.27.103 D   N     48072 Unmonitored
3 sip peers [Monitored: 0 online, 0 offline\
Unmonitored: 3 online, 0 offline]
```

For mer informasjon om en spesifikk bruker benyttes 'sip show user *username*':

```
raspberrypi*CLI> sip show user user201
* Name           : user201
Secret           : <Set>
MD5Secret        : <Not set>
Context          : 1242domain
Language         :
AMA flags        : Unknown
Transfer mode    : open
MaxCallBR        : 384 kbps
CallingPres      : Presentation Allowed, Not Screened
Call limit       : 0
Callgroup        :
Pickupgroup      :
Callerid         : "User 201" <>
ACL              : No
Sess-Timers      : Accept
Sess-Refresh     : uas
Sess-Expires     : 1800 secs
Sess-Min-SE      : 90 secs
Codec Order      : (ulaw:20,g729:20)
Auto-Framing     : No
```

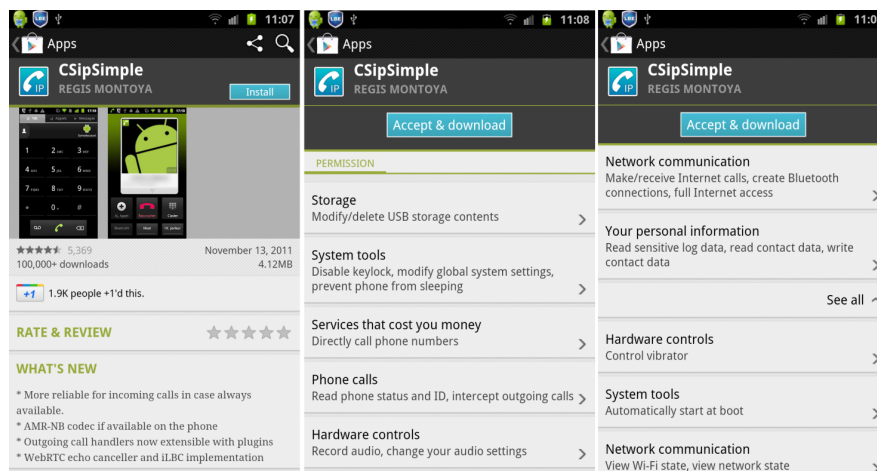
Når "user203" ringer "user201" vil følgende vises i monitoreringsvinduet til Asterisk:

```
== Using SIP RTP CoS mark 5
-- Executing [201@1242domain:1] Dial("SIP/user203-00000008",\
"SIP/user201,20,tr") in new stack
== Using SIP RTP CoS mark 5
-- Called user201
-- SIP/user201-00000009 is ringing
```

```
-- SIP/user201-00000009 answered SIP/user203-00000008
== Spawn extension (1242domain, 201, 1) exited non-zero\
on 'SIP/user203-00000008'
```

4.6 Programvare på Android

For å teste VoIP og oppsettet av Asterisk på Androidklienter benyttet vi oss av programmet *CSipSimple*⁷. Dette var det programmet som viste seg å være enklest å konfigurere, samt at det enkelt kunne integreres helt med telefonen.



Figur 4.5 CSipSimple og krav til rettigheter på telefonen.

Start Google Play (tidligere Android Market) og søk etter CSipSimple og man får opp skjermen som vist i Figur 4.5.

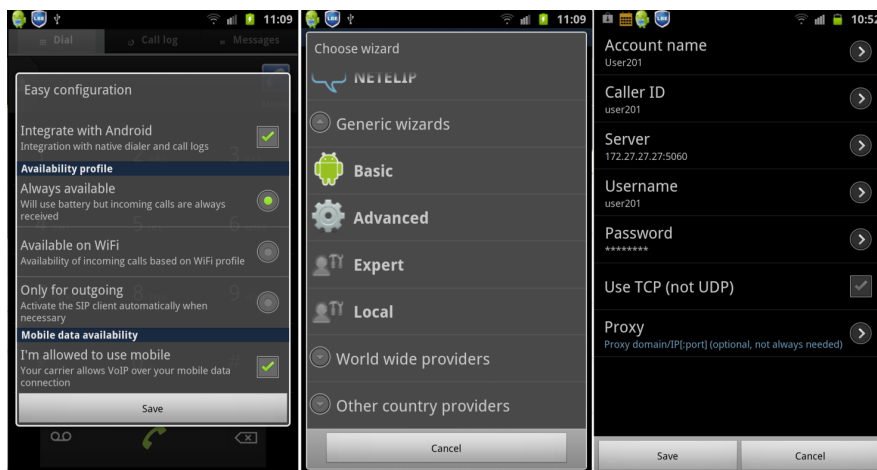
Som vi kan se av Figur 4.5, så stilles det mange krav til tilgang og tillatelser på smarttelefonen. Dersom man forsøker å unngå å ha full tillit til villkårlige programmerere, bør man kanskje undersøke innstillingene som omfatter mulighet for misbruk av opptak av lyd, lese logg- og kontaktdata, m.m. Men applikasjonssikkerhet er ikke vårt fokus så vi fortsetter med å teste VoIP på denne klienten.

Når vi starter opp CSipSimple for første gang får vi en oppstartsskjerm som vist til venstre i Figur 4.6. Dersom vi ønsker å la VoIP-programmet fungere sammen med vanlig mobiltelefonringing av telefonnummer så merker vi av øverste boks “Integrate with Android”. Dette medfører at man blir spurt om man skal ringe via VoIP eller vanlig GSM når man ringer ut.

Dersom vi kun ønsker at VoIP skal være tilgjengelig (for innkommende VoIP-samtaler) når vi er på et trådløst nettverk, eller kun for utgående VoIP-samtaler, må vi krysse av korrekt i “Availability profile”.

Dersom vi ønsker at VoIP-serveren også skal være tilgjengelig, både for innkommende og utgående samtaler, når vi benytter oss kun av datatrafikk og ikke har noe lokalt trådløst nett-

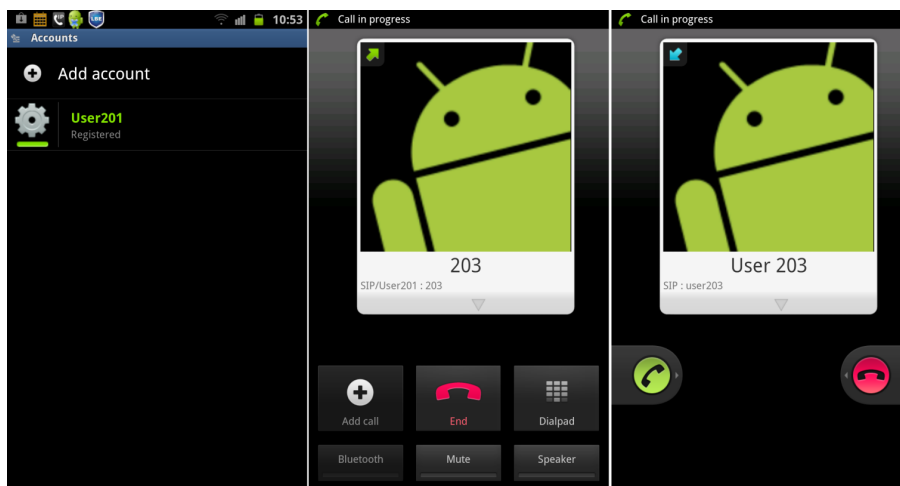
⁷<http://code.google.com/p/csipsimple/>



Figur 4.6 CSipSimple – oppstart og innstillinger på klienten.

verk, må vi huke av siste boksen også “I’m allowed to use mobile”. (OBS. Dette kan bli ganske mye dyrere enn vanlige telefonsamtaler ettersom vi må betale for trafikkdata i tillegg til tjenesten når vi ringer. Men for brukere med ubegrenset data er det jo ingen problemer - kostnadsmessig.)

Når vi starter programmet for første gang har den ingen brukere og konfigurasjoner klare for VoIP-tilgang. Midt i Figur 4.6 vises valget man får ved å velge “Add account” i menyen etter oppstart, og velger man “Advanced” her kommer man til vinduet til høyre i Figur 4.6. Her fyller vi ut med data som vist i figuren og med passord “12341234” som man har benyttet i ‘sip.conf’ sin brukerseksjon.



Figur 4.7 CSipSimple. Status på konti og innlogginger, samt utgående og innkommende samtale til/fra internummer '203'.

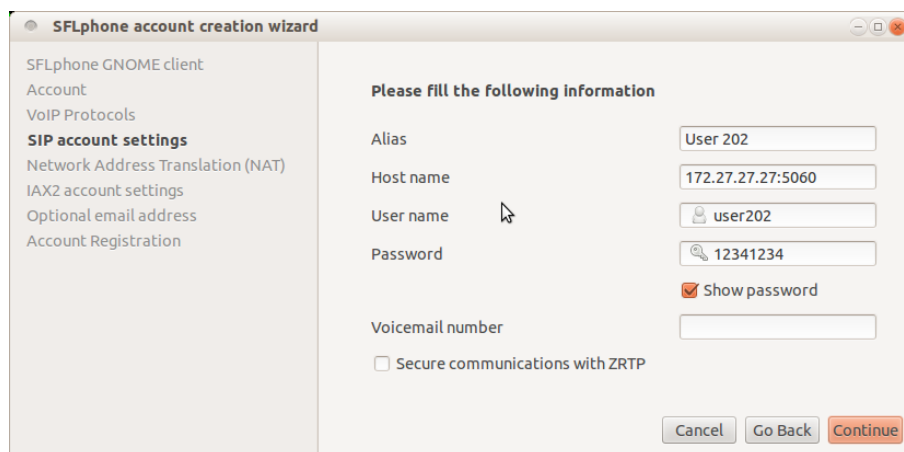
Dersom alt går greit vil man komme til en statusside med oversikt over alle konti i VoIP-programmet som vist til venstre i Figur 4.7. Her viser grønn lampe på kontoen at vi er oppkoblet mot VoIP-server. Vi viser i Figur 4.7 også hvordan det ser ut på telefonen når man ringer

'203' i CSipSimple, samt hvordan mottak av samtale fra '203' ser ut. Legg merke til at vi for internnummer '203' har definert CallerID til å være "User 203".

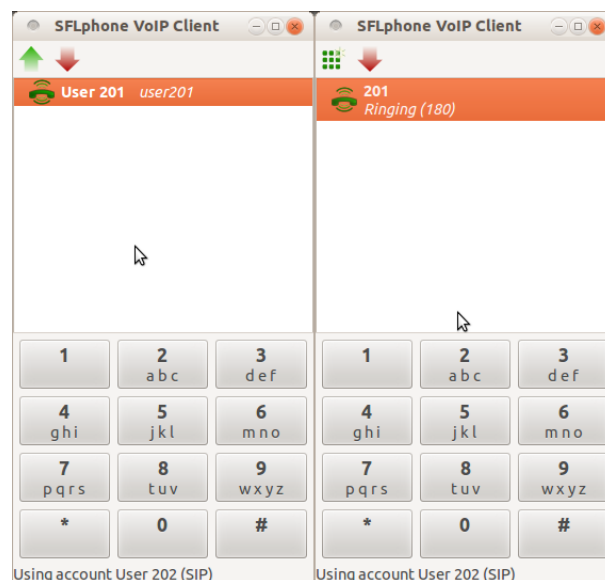
4.7 Programvare på Linux

På Linux ble programmet sflphone-gnome⁸ testet. Dette installeres gjennom å kjøre kommandoen:

```
$ sudo apt-get install sflphone-gnome
```



Figur 4.8 Konfigurering av sflphone for intern server.



Figur 4.9 Bruk av sflphone til utgående og innkommende samtaler.

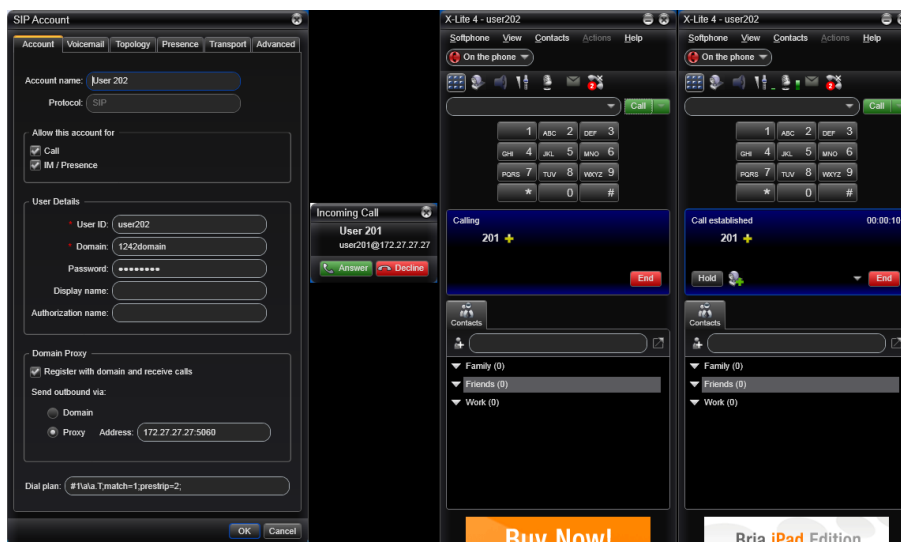
Figur 4.8 viser hvordan den ble satt opp på tilsvarende måte som Androids CSipSimple. Forsøk på ringe ut til nummer 201, samt innkommende fra 201, fungerte tilfredstillende og vises i

⁸<http://sflphone.org/>

Figur 4.9.

Det var ingen problemer med oppsettet og telefonidelen fungerte akkurat på samme måte som ved bruk av Android-programvaren.

4.8 Programvare på Windows



Figur 4.10 Bruk av X-Lite4 under Windows 7.

På Microsofts Windows plattform ble programvaren X-Lite⁹ benyttet. Figur 4.10 viser fra venstre konfigurasjon, mottak av samtale, utgående oppringing og pågående samtale når vi har satt den opp mot den interne serveren.

4.9 Sikkerhet og VPN

Sikkerhet i SIP og VoIP er behørlig beskrevet i tidligere publikasjoner som FFI-rapport[12] og andre publikasjoner vi viser til i Kapittel 2.

Metodene vi har satt opp i våre eksperimenter og oppsett, beskytter hverken innhold eller trafikkmønster, det vil si hvem som ringer til hvem. Dersom man alltid bruker VPN mot sentralen (for eksempel OpenVPN¹⁰ for en VPN-løsning som er billig og god) så vil man i alle fall beskytte alle interne samtaler, og innhold i samtaler. Vel og merke inntil de forlater din server og sendes ukryptert mot mottaker på telefoninettet.

Kryptering og beskyttelse mot ekstern avlytting av eksterne samtaler og mottatte samtaler fra “utsiden”, dvs. beskyttelse mot avlytting av oppkoblingen til voip.ms, er ikke en del dette oppsettet. I selskapet VoIP.ms sin Wiki gir de noen råd til sikring av server og brukere ved bruk av deres tjenester, men uten å løse de overnevnte problemer.

⁹<http://www.counterpath.com/x-lite.html>

¹⁰<http://www.openvpn.net/>

Vi vil i en senere rapport komme tilbake til bruk av sikkerhetsmekanismer i VoIP som protokollene TLS[8], SRTP[4, 25] og ZRTP[27, 24].

5 Konklusjon

Vi har vist at det er enkelt å sette opp en VoIP-tjeneste for internt bruk. Vi har også vist at det er enkelt å utvide en slik tjeneste til å inkludere tilgang til det vanlige telefoninettet ved å kjøpe tjenester fra en tredjepart. Med dette kunne vår “interne sentral” ringe ut og motta innkommende samtaler på både amerikansk og norsk nummer. Vi har i tillegg vist hvordan man kan sette opp slik tjeneste både på en Intel- og en ARM-basert datamaskin, ved å benytte programvaren Asterisk.

Vi har testet samtaler mellom brukerne i to scenarier, med og uten ekstern oppkobling. Dermed har vi sett på oppringing av interne brukere gjennom eksterne telefonnummer vi leier både i Norge og i USA. Vi har også testet oppringing av eksterne telefonnummer med de interne brukerne - oppringing av både norske og amerikanske nummer er testet.

Vi håper at våre erfaringer med å sette opp enkle VoIP-systemer kan gi andre med behov for lab-oppsett eller mindre VoIP-sentraler en lettere start. Det er mange videre veier for å sette opp større tjenester og et mangfold av programvare finnes for å kunne gjøre dette. Vi har valgt å holde oss til Asterisk ettersom det i dag benyttes i alt fra enkle små oppsett, til meget avanserte telefonsentraler. Asterisk fungerer godt både for bedrifter og teleleverandører, og kan sannsynligvis fint benyttes til de mest kompliserte lab-oppsett.

Referanser

- [1] 3CX. 3cx phone system - software based voip ip pbx / pabx for windows. <http://www.3cx.com/>, 2012. [Online; hentet 15. oktober 2012].
- [2] Arch Linux ARM. Arch linux arm - raspberry pi. <http://archlinuxarm.org/platforms/armv6/raspberry-pi>, 2012. [Online; hentet 15. oktober 2012].
- [3] Asterisk/Digium. Asterisk - the open source telephony projects. <http://www.asterisk.org/>, 2012. [Online; hentet 15. oktober 2012].
- [4] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The secure real-time transport protocol (srtp). <http://www.ietf.org/rfc/rfc3711.txt>, March 2004. [Online; hentet 15. oktober 2012].
- [5] D. Butcher, Xianyang Li, and Jinhua Guo. Security challenge and defense in voip infrastructures. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, pages 1152–1162, 2007.
- [6] Michael Casadevall. Image writer for windows. <http://launchpad.net/win32-image-write>, 2012. [Online; hentet 15. oktober 2012].
- [7] Debian. Debian - the universal operating system. <http://debian.org/>, 2012. [Online; hentet 15. oktober 2012].
- [8] T. Dierks and C. Allen. The tls protocol - version 1.0. <http://www.ietf.org/rfc/rfc2246.txt>, January 1999.
- [9] Dimitris Geneiatakis, Georgios Kambourakis, Tasos Dagiuklas, Costas Lambrinouidakis, and Stefanos Gritzalis. Sip security mechanisms: A state-of-the-art review. *Proceedings of the Fifth International Network Conference (INC 2005)*, 2005.
- [10] Google. Call phones from gmail. <http://www.google.com/chat/voice/>, 2012. [Online; hentet 15. oktober 2012].
- [11] Prateek Gupta and Vitaly Shmatikov. Security analysis of voice-over-ip protocols. pages 49–63, 2007.
- [12] Anne Pernille Hveem. Sikkerhet i voice over ip og andre multimediasesjoner basert på sip og rtp. *FFI Report*, June 2012.
- [13] IncredibePBX. Incrediblepbx - voip telephony on steroids. <http://incrediblepbx.com/>, 2012. [Online; hentet 15. oktober 2012].
- [14] IncredibePBX. Voip on steroids: Introducing incredible pbx 3.3 for the usd 35 raspberry pi. <http://nerdvittles.com/?p=3026>, 2012. [Online; hentet 15. oktober 2012].

- [15] Leif Madsen, Jim Van Meggelen, and Russel Bryant. *Asterisk: The Definitive Guide*. O'Reilly Media.
- [16] Shawn McGann and Douglas C. Sicker. An analysis of security threats and tools in sip-based voip systems. *2nd Annual Workshop VoIP Security*, June 2005.
- [17] mobiThinking.com. Global mobile statistics 2012 home: all the latest stats on mobile web, apps, marketing, advertising, subscribers and trends... <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats>, 2012. [Online; hentet 15. oktober 2012].
- [18] Nokia. Qt on pi. <http://qt-project.org/wiki/Qt-RaspberryPi>, 2012. [Online; hentet 15. oktober 2012].
- [19] RaspberryPi. Raspberry pi - an arm gnu/linux box for usd 25. take a byte! <http://www.raspberrypi.org/>, 2012. [Online; hentet 15. oktober 2012].
- [20] Raspian. Raspian - an unofficial port of debian wheezy armhf. <http://www.raspbian.org/>, 2012. [Online; hentet 15. oktober 2012].
- [21] skype.com. Skype. <http://skype.com/>, 2012. [Online; hentet 15. oktober 2012].
- [22] Lars Strand and Anders Moen Hagalisletto. Designing attacks on sip call setup. *International Journal of Applied Cryptography*, 2(1):13–22, July 2010.
- [23] Ted Wallingford. *Switching to VoIP*. O'Reilly Media.
- [24] Wikipedia. Zrtp. <http://en.wikipedia.org/wiki/ZRTP>, 2012. [Online; hentet 15. oktober 2012].
- [25] Wikipedia.org. Secure real-time transport protocol. http://en.wikipedia.org/wiki/Secure_Real-time_Transport_Protocol, 2012. [Online; hentet 15. oktober 2012].
- [26] Sherali Zeadally and Farhan Siddiqui. Design and implementation of a sip-based voip architecture. *Proceedings of the 18th International Conference on Advanced Information Networking and Applications*, 2004.
- [27] P. Zimmermann. Zrtp: Media path key agreement for unicast secure rtp. <http://www.ietf.org/rfc/rfc6189.txt>, April 2011. [Online; hentet 15. oktober 2012].
- [28] Theo Zourzouvilys and Eric Rescorla. An introduction to standards-based voip: Sip, rtp, and friends. *IEEE Internet Computing*, vol. 14, nr. *IEEE Internet Computing*, pages 69–73, March 2012.

Appendiks A Intern sentral

Her er de komplette filene *sip.conf* og *extensions.conf* slik som de ble benyttet i testoppsettet kalt "Intern sentral".

A.1 sip.conf

```
;
; sip.conf for intern sentral
;
[general]
port=5060
bindaddr=0.0.0.0
context=1242domain
tos=0x18
nat=yes
externip=128.255.x.y
disallow=all
allow=all
alwaysauthreject=yes

[user201]
regexten=201
type=friend
host=dynamic
context=1242domain
secret=12341234
callerid="User 201"
dtmfmode=rfc2833
nat=yes
disallow=all
allow=all
registertrying=yes

[user202]
regexten=202
type=friend
host=dynamic
context=1242domain
secret=12341234
callerid="User 202"
dtmfmode=rfc2833
```

```
nat=yes
disallow=all
allow=all
registertrying=yes
```

```
[user203]
regexten=203
type=friend
host=dynamic
context=1242domain
secret=12341234
callerid="User 203"
dtmfmode=rfc2833
nat=yes
disallow=all
allow=all
registertrying=yes
```

A.2 extensions.conf

```
;
; extentions.conf for intern sentral
;
[general]
static=yes
writeprotect=no

[1242domain]
; First a playback/echo on internal 299
exten => 299,1,Playback(demo-echotest)
exten => 299,2,Echo
exten => 299,3,Playback(demo-echodone)
; The users
exten => 201,1,Dial(SIP/user201,20,tr)
exten => 202,1,Dial(SIP/user202,20,tr)
exten => 203,1,Dial(SIP/user203,20,tr)
exten => 203,2,Dial(SIP/user201,20,tr)
```

Appendiks B Eksternt tilgjengelig sentral

Her er de komplette filene *sip.conf* og *extensions.conf* slik som de ble benyttet i testoppsettet kalt "Eksternt tilgjengelig sentral".

B.1 sip.conf

```
;
; sip.conf for eksternt tilgjengelig sentral
;
[general]
port=5060
bindaddr=0.0.0.0
context=voipmscontext
tos=0x18
nat=yes
externip=128.255.x.y
disallow=all
allow=all
alwaysauthreject=yes
register => 141188_loc3:SEcRetPassword21@newyork.voip.ms:5060

;
; config at location 3
;
[voipms]
context=voipmscontext
host=newyork.voip.ms
secret=SEcRetPassword21
type=peer
username=141188_loc3
disallow=all
allow=ulaw
allow=g729
fromuser=141188_loc3
trustrpid=yes
sendrpid=yes
insecure=invite
nat=yes

[user201]
```

```
regexten=201
type=friend
host=dynamic
secret=12341234
context=voipmscontext
callerid="User 201" <+4767555307>
dtmfmode=rfc2833
nat=yes
disallow=all
allow=all
registertrying=yes
```

```
[user202]
regexten=202
type=friend
host=dynamic
secret=12341234
context=voipmscontext
callerid="User 202" <+4767555307>
dtmfmode=rfc2833
nat=yes
disallow=all
allow=all
registertrying=yes
```

```
[user203]
regexten=203
type=friend
host=dynamic
secret=12341234
context=voipmscontext
callerid="User 203" <+4767555307>
dtmfmode=rfc2833
nat=yes
disallow=all
allow=all
registertrying=yes
```

B.2 extensions.conf

```
;
; extensions.conf for eksternt tilgjengelig sentral
;
[general]
static=yes
writeprotect=no

[1242domain]
; First a playback/echo on internal 299
exten => 299,1,Playback(demo-echotest)
exten => 299,2,Echo
exten => 299,3,Playback(demo-echodone)
; The users
exten => 201,1,Dial(SIP/user201,20,tr)
exten => 202,1,Dial(SIP/user202,20,tr)
exten => 203,1,Dial(SIP/user203,20,tr)
exten => 203,2,Dial(SIP/user201,20,tr)

[voipmscontext]
include => 1242domain
include => voipms-inbound
include => voipms-outbound

[voipms-inbound]
exten => 4767555307,1,Dial(SIP/user201,20,tr)
exten => 4767555307,2,Dial(SIP/user202,20,tr)
exten => 4767555307,3,Dial(SIP/user203,20,tr)

[voipms-outbound]
exten => _1NXXNXXXXXX,1,Dial(SIP/${EXTEN}@voipms)
exten => _1NXXNXXXXXX,n,Hangup()
exten => _NXXNXXXXXX,1,Dial(SIP/1${EXTEN}@voipms)
exten => _NXXNXXXXXX,n,Hangup()
exten => _011.,1,Dial(SIP/${EXTEN}@voipms)
exten => _011.,n,Hangup()
exten => _00.,1,Dial(SIP/${EXTEN}@voipms)
exten => _00.,n,Hangup()
```