VoIP Workshop
PacNOG3

Rarotonga, Cook Islands

June 2007

Labs 1 - 4, Asterisk

Lab 5, INOC-DBA

Lab 6-7, Cisco Voice Gateways

Lab 8, CODECS

Lab Summary

Server logins are as you have set up in previous labs.

Default gateway: 202.65.202.254.
DNS server: 202.65.202.252

Group	Extensions	Server IP
1	2100 - 2199	202.65.42.1/24
2	2200 - 2299	202.65.42.2/24
3	2300 - 2399	202.65.42.3/24
4	2400 - 2499	202.65.42.4/24
5	2500 - 2599	202.65.42.5/24
6	2600 - 2699	202.65.42.6/24
7	2700 - 2799	202.65.42.7/24
8	2800 - 2899	202.65.42.8/24
9	2900 - 2999	202.65.42.9/24
10	3000 - 3099	202.65.42.10/24
11	3100 - 3199	202.65.42.11/24
12	3200 - 3299	202.65.42.12/24
13	3300 - 3399	202.65.42.13/24
14	3400 - 3499	202.65.42.14/24
15	3500 - 3599	202.65.42.15/24
16	3600 - 3699	202.65.42.16/24
17	3700 - 3799	202.65.42.17/24
18	3800 - 3899	202.65.42.18/24
19	3900 - 3999	202.65.42.19/24
20	4000 - 4099	202.65.42.20/24
21	4100 - 4199	202.65.42.21/24
22	4200 - 4299	202.65.42.22/24
23	4300 - 4399	202.65.42.23/24
24	4400 - 4499	202.65.42.24/24
25	4500 - 4599	202.65.42.25/24
26	4600 - 4699	202.65.42.26/24
27	4700 - 4799	202.65.42.27/24
28	4800 - 4899	202.65.42.28/24
29	4900 - 4999	202.65.42.29/24
30	5000 - 5099	202.65.42.30/24
31	5100 - 5199	202.65.42.31/24

AS5400 Gateway	202.65.42.46	login: pacnog/workshop	enable: rarovoip
VG200 Gateway	202.65.42.47	login: pacnog/workshop	enable: rarovoip
NZ voip server	203.114.130.250		

Dial Plan for each group:

'..' is as per your group number extensions above.

..00 - ..09	SIP phones (password for each, *extension*+passwd, e.g. 2000passwd)
..10 - ..19	Fun with IVRs
..20 - ..29	Music on hold extensions
..30	DB count application
..40 - ..49	Agents
..50	'Helpdesk' queue access
..60	Echo test
..70	Conference bridge
..80 - ..89	Festival text to speech play extensions
..90	Ring the VG200 connected analog phone
..99	Voicemail access

1. Access an outside PSTN line
7. Send calls to the AS5400
8. Access another group
9. Access INOC-DBA

Lab 1: Initial Asterisk Install

1. Install Asterisk

```
apt-get install asterisk
apt-get install asterisk-sounds-extra
apt-get build-dep asterisk
```

We need to download the current Zaptel source and compile it, as the packaged version doesn't work correctly on our lab Ubuntu servers.

```
cd /usr/src
wget http://www.ubuntu.sun.ac.za/asterisk/1.2/libpri-1.2.4.tar.gz
wget http://www.ubuntu.sun.ac.za/asterisk/1.2/libpri-1.2.4.tar.gz
tar -xvf libpri-1.2.4.tar.gz
tar -xvf zaptel-1.2.16.tar.gz

cd /usr/src/libpri-1.2.4.tar
make
make install
```

Before compiling we need modify the zaptel makefile to enable the ztdummy module which will be providing Asterisk a timing signal for things such as music on hold and conferencing:

```
cd /usr/src/zaptel-1.2.16.tar
nano Makefile
```

Search for '# ztdummy' and remove the # (i.e. uncomment ztdummy from that iine)

```
make
make install

modprobe ztdummy

ztcfg -v  # only do this if you have real zaptel interfaces
```

Set RUNASTERISK=yes in /etc/default/asterisk

```
nano /etc/default/asterisk
```

2. Start Asterisk

```
/etc/init.d/asterisk start
```

Have a look at the available startup options:

```
asterisk -h
```

To connect to the Asterisk CLI:

```
asterisk -r
```

3. Edit Configuration Files in /etc/asterisk/

Set up three SIP peers in sip.conf: ..00, ..01, ..02. Add to the bottom of sip.conf, repeating for each of the three SIP peers:

```
[..00]
type=friend
host=dynamic
username=..00
secret=passwd..00
canreinvite=no
nat=yes
context=phones
dtmfmode=rfc2833
allow=all
```

Create backup of original extensions.conf:

```
mv extensions.conf orig_extensions.conf
```

Create new extensions.conf with the following:

```
[general]
static=yes
writeprotect=no
autofallthrough=yes
clearglobalvars=no
priorityjumping=yes

[phones]
; remember to replace .. with your group's numbers!
exten => ..00,1,Dial(SIP/..00)
exten => ..01,1,Dial(SIP/..01)
exten => ..02,1,Dial(SIP/..02)

exten => ..60,1,Answer()
exten => ..60,2,Playback(demo-echotest)
exten => ..60,3,Echo
exten => ..60,4,Playback(demo-echodone)
exten => ..60,5,Hangup
```

Connect to Asterisk (asterisk -r), up the debug output (set verbose 10), and reload the config (reload).

4. Configure Softphone

Download and configure the Xten Xlite Softphone - (http://www.xten.com/index.php?menu=download)

Input SIP settings in *Main Menu* > *System Settings* > *SIP Pro..* > *Default*

Enabled:	Yes
Username:	SIP extension you are configuring (e.g. 2000)
Authorization User:	Same as *Username*
Password:	extensionpasswd, e.g. 2000passwd
SIP Pro..:	The address of your Asterisk server
OutBound Pro..:	Same as *SIP Pro..*

You should now be able to call between your three phones.

Call the echo test on ..60 and you should be able to hear yourself!

Lab 2: Basic Asterisk Config

Configure the following, using the extensions given in the Lab summary:

- voicemail for each extension
- a sample IVR
- a meetme conference
- a sample MOH stream

Here's a start on the configuration files:

voicemail.conf

```
[default]
..00 => 1234,User 1,user1@email.address
..01 => 1234,User 2,user2@email.address
..02 => 1234,User 3,user3@email.address
```

extensions.conf

```
[phones]
; configure pattern match for local extensions
; e.g. _200X
exten => _..0X,1,Dial(SIP/${EXTEN},15)
exten => _..0X,n,Voicemail(u${EXTEN})
exten => _..0X,n,Hangup()

; allow checking of voicemails. try it out!
exten => ..99,1,VoicemailMain()

; extension to allow dialling the IVR
exten => _..10,1,Goto(ivr-test,s,1)

[ivr-test]
; based on the slides, create an IVR which allows you to
; ring your extensions
```

If you're not sure about how specific applications work, from the Asterisk CLI try:

```
show applications
show application goto
```

Lab 3: Advanced Asterisk Configuration

1. Asterisk Database

Implement the following in extensions.conf:

```
[phones]
; start counting and store count progress in astdb

; check if DB key exists, if not, jump to key_no_exist
; function DB_Exists returns 1 if the key exists, 0 if not
exten => 30,1,GotoIf(DB_Exists(test/count)?key_no_exist)

; begin the counting!
exten => 30,n(start),Set(COUNT=${DB(test/count)})
exten => 30,n,SayNumber(${COUNT})
```

```
exten => 30,n,Set(COUNT=$[${COUNT} + 1])
; update the DB
exten => 30,n,Set(DB(test/count)=${COUNT})
exten => 30,n,Goto(start)

; if we got here it is because the key didn't exist in the DB
; create the key
exten => 30,n(key_no_exist),Set(DB(test/count)=1)
; and jump back to the start to begin counting
exten => 30,n,Goto(start)
```

Reload Asterisk, and have a look at the Asterisk DB

```
reload
database show
```

Now dial ..30, and look at the DB again. You should see a new key (test/count) in the DB containing the current count.

2. Implement Nightmode

We want the nightmode to work as follows:

We want to create an extension called ..50 for our 'main number'
We will create two new keys in the DB:
nightmode/open_time, and nightmode/close_time
When a call comes in, we will check to see if we are currently between those two times, and if so ring all three phones. If not, go straight to voicemail

Hints:

To manually set a DB key from the CLI:

```
database put family key value
```

Time based branching:

```
show application gotoif
```

Dialling multiple channels simultaneously:

```
Dial(SIP/1000&SIP/2000&SIP/3000)
```

3. Extension Macro

Look in the original /etc/extensions.conf (you should have moved it to orig_extensions.conf), and use it as a guide.

Create a simple extension macro to dial our extensions and branch to voicemail if not answered.

4. Set up Agents

Edit agents.conf - add three agents for you group to the bottom of the existing file:

```
agent => ..40,1234,Agent one
agent => ..41,1234,Agent two
agent => ..42,1234,Agent three
```

To enable Agent login and logout, add to extensions.conf:

```
[phones]
; hint in CLI, show application AgentCallbackLogin
exten => ..59,1,AgentCallbackLogin()
```

Reload Asterisk, then check the state of Agents before and after a login:

```
show agents
```

5. Set up a Queue
Edit queues.conf - use the existing defaults as a guide. Call the queue helpdesk (this is at the start of the file in [ ]). The important piece is to add to the bottom of queues.conf:

```
member => Agent/..40
member => Agent/..41
member => Agent/..42
```

And in Extensions.conf create a means to enter the queue:

```
[phones]
exten => ..50,1,Queue(helpdesk)
```

Ring the queue with Agents all logged out, and all logged in.

6. Install Festival text to speech
Exit out of Asterisk and install Festival:

```
apt-get install festival
```

Configure Festival for Debian / Ubuntu. Make /etc/festival.conf look like the following:

```
;; Enable access to localhost (needed by debian users)
(set! server_access_list '("localhost\\.localdomain" "localhost"))

;; set italian voice (comment the following 2 lines to use british_ameri-
can)
(language_italian)
(set! voice_default 'voice_pc_diphone)

;;; Command for Asterisk begin
(define (tts_textasterisk string mode)
    "(tts_textasterisk STRING MODE)
  Apply tts to STRING.  This function is specifically designed for
  use in server mode so a single function call may synthesize the string.
  This function name may be added to the server safe functions."
      (utt.send.wave.client (utt.wave.resample (utt.wave.rescale (utt.synth
                  (eval (list 'Utterance 'Text string))) 5) 8000)))
;;; Command for Asterisk end
```

To use Festival:

```
exten => 123,1,Festival('Hello World')
exten => 123,2,SetVar(speech='Hello World by variable')
exten => 123,3,Festival('${speech}')
```

Lab 4: Asterisk Exercises

1. Another Extensions Macro
Write an extension macro which looks up a database to get the following information:

callerID name
callerID number
Voicemail box
do not disturb flag

If the do not disturb flag is set, playback a prompt saying (sorry, <name> doesn't want to be disturbed). Make sure the macro correctly set the CallerID name and number.

2. DB lookup for incoming calls
Write a piece of code that does a DB lookup on inbound calls into the [incoming] context, looks up the number in the database, and uses the result to branch into the appropriate location in the dial plan.

In what circumstances do you think this would be handy?

3. Write a Prompt recording Macro.
This macro will need to take as input the filename to record, and optionally the format to record it in.

The macro needs to:

1. record the prompt
2. let the user play it back
3. let the user confirm they wish to use that prompt
4. save the prompt in the correct location

Note, Festival text to speech is handy to provide instructions here!

4. Write an application to ping a device
Create a context (starting with the 's' extension) which allows you to ping a device.

You'll need to work out how to accept DTMF input, run a ping command external to asterisk, and read the result back to the caller.

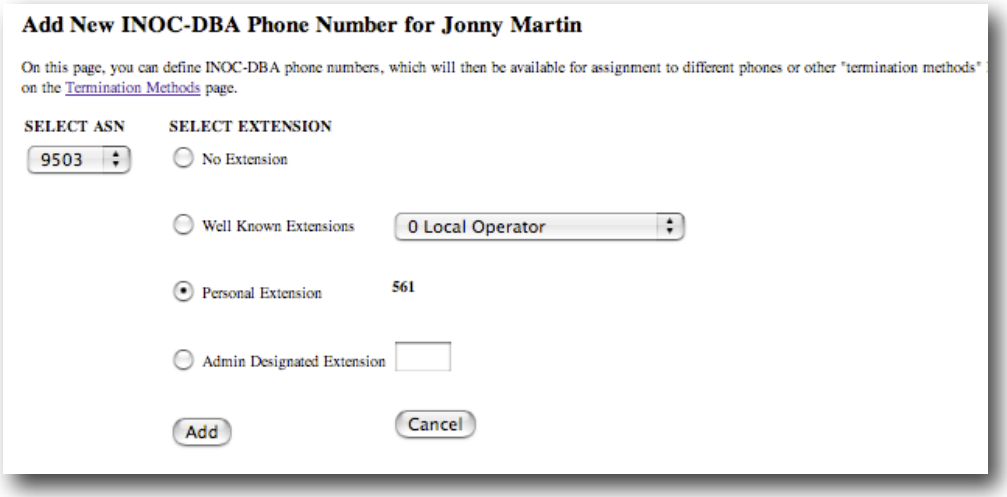
Lab 5: Connecting Asterisk to INOC-DBA

You will need to have set up an account and log in to the INOC-DBA administration system to do this.

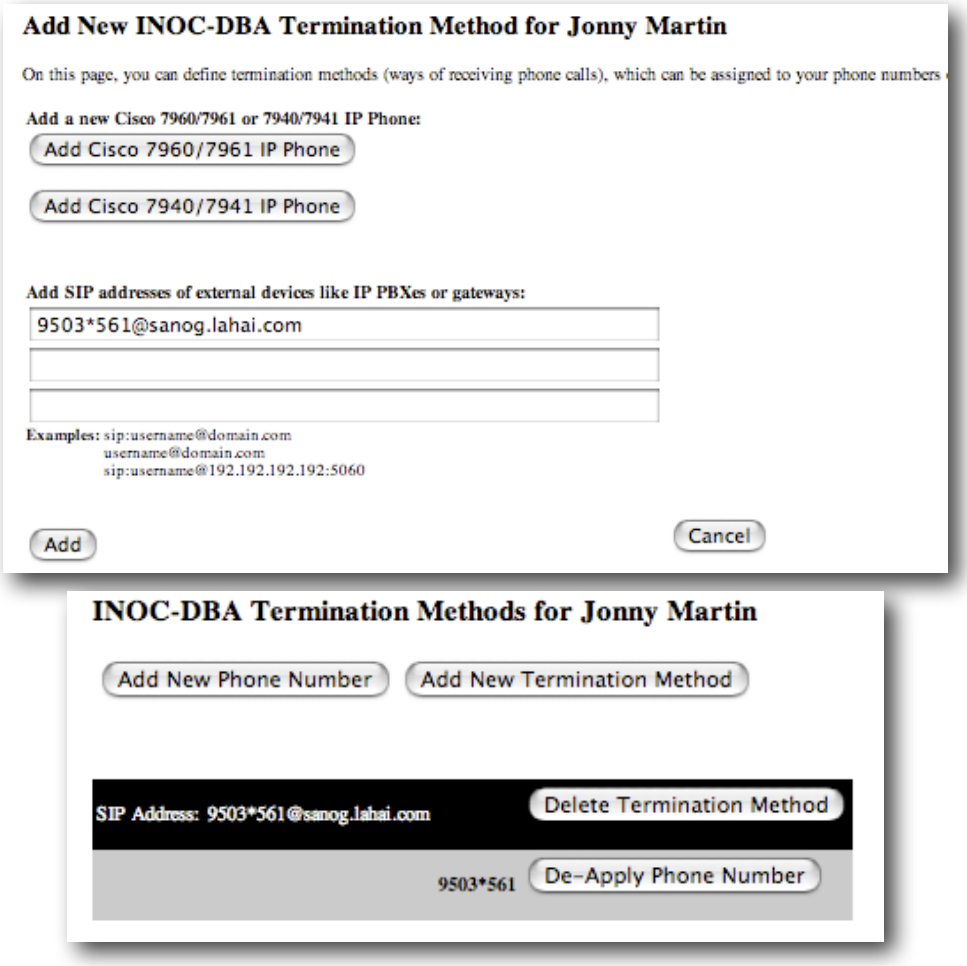
http://www.pch.net/inoc-dba/

1. Set up INOC-DBA to send calls to your Asterisk server.
You need to set up a termination method through the INOC-DBA system to deliver calls to your asterisk server.

For this lab exercise we will will set up your INOC-DBA personal extension to terminate calls on your lab asterisk server. Select 'My Phone Numbers' from the menu and populate it accordingly:



Select 'Termination Methods' from the menu and add the IP address of your lab server, then select that as the termination method for your personal extension.



2. Configure Asterisk sip.conf

Asterisk needs to be configured to SIP REGISTER itself with the INOC-DBA servers. Add the following to the [general] section of sip.conf:

```
register => 9503*561:password:jonny@inoc-dba.pch.net/9503*561
```

Replacing 9503*561 with your INOC-DBA extension, and password:jonny@ with your password and login name.

This statement registers our Asterisk box with INOC-DBA. Inbound calls are sent to the default context.

3. Configure inbound calls

Inbound calls land in the default context. We want these calls to ring a phone, so add something like the following into the [default] context, substituting your details for SIP/2000 and 9503*561.

```
exten => 9503*561,1,Dial(SIP/2000,15)
exten => 9503*561,n,Voicemail(u2000)
exten => 9503*561,n,Hangup()
```

You may want to have inbound NOC calls ring multiple phones. Configure your INOC-DBA extension to ring multiple phones at once.

A nicer way to implement this is to use a GoTo statement in the default context to send inbound calls to 9503*561 to an extension elsewhere in your dialplan, enabling you to easily change the destination for inbound calls. Use this method to send a call to one of your existing extensions. This could be a phone, voicemail, conference, etc.

4. Configure outbound calls
Calls prefixed with a 9 will be sent out to INOC-DBA. We need to first strip the 9, and then set our outgoing callerID correctly. Add the following to the appropriate context in extensions.conf:

```
; This extension is for outgoing calls to inoc-dba
; 9 for an outside-inoc-dba-line
exten => _9.,1,SetCIDName(Jonny Martin)
exten => _9.,n,SetCIDNum(9503*561)
exten => _9.,n,Dial(SIP/${EXTEN:1}@inoc-dba)
exten => _9.,n,Hangup
```

Lab 6. Cisco IOS Voice Gateway i.

In this lab we are going to setup a VG200 with one FXS interface such that an attached telephone can dial all of the allocated extensions on each asterisk server.

1. Configure SIP peer on Asterisk server
You need to configure a SIP peer for both in and outbound calls to the VG200 gateway. In sip.conf add the following peer definition:

```
[vg200]
type=friend
disallow=all
allow=ulaw
canreinvite=no
context=from-vg200
host=202.65.42.47
dtmfmode=rfc2833
insecure=very
```

Note, if you don't have a context [from-as5400] in extensions.conf you will need to create it, and put some extension statements in it!

2. Configure extensions.conf
Inbound calls will land in the [from-vg200] context. Use a GoTo statement to send these calls to the context where you have all your useful extensions defined.

We want to create an extension that when dialled from our Asterisk system will dial the VG200 analog phone. Add the following the appropriate context to extensions.conf. This example is for group one, using extension 2190

```
exten => 2190,1,Dial(SIP/2190@vg200)
```

The IOS gateway is simply another SIP peer, so you can send calls to it in the same fashion you send calls to any other peer.

3. Configure the VG200
The FXS voice port is already configured for you. In this case, the default IOS settings are all that is required.

Create a dial-peer to ring the analog phone. Login to the gateway and configure the following (changing 2190 for your group's extension, e.g. 2190, 2290):

```
dial-peer voice 2190 pots
 port 1/0/0
```

Create a dial-peer to send calls to your asterisk server (again, an example for group 1):

```
dial-peer voice 1 voip
 description calls to group 1
 destination-pattern 21..
 session protocol sipv2
 session target ipv4:202.65.42.1
 dtmf-relay rtp-nte
 codec g711ulaw
 no vad
```

Test that you can make calls to and from the analog phone.

Lab 7. Cisco IOS Voice Gateway ii.

In this lab we are going to setup an AS5400 multiple E1/T1 interfaces on it as an outbound PSTN gateway.

In this example, calls from your softphone to the asterisk server will be prefixed with a 7. Asterisk will send these calls to the AS5400 without stripping the 7.

In our fictitious example, calls will be going out to the local PSTN and requires to be prefixed by a 0. The AS5400 will first strip the 7, and append a 'telco' access code of 0 before sending the call out one of the E1/T1 interfaces.

The digit stripping is performed on the inbound dial-peer (VoIP in this case), and the prepend on the outbound dial-peer (POTS in this case).

1. Configure SIP peer on Asterisk server
You need to configure a SIP peer for both in and outbound calls to the AS5400 gateway. In sip.conf add the following peer definition:

```
[as5400]
type=friend
disallow=all
allow=ulaw
canreinvite=no
context=from-as5400
host=202.65.42.46
dtmfmode=rfc2833
insecure=very
```

Note, if you don't have a context [from-as5400] in extensions.conf you will need to create it, and put some extension statements in it!

2. Configure extensions.conf
Inbound calls will land in the [from-vg200] context. Use a GoTo statement to send these calls to the context where you have all your useful extensions defined. In this case we want inbound calls to ANY number to be met with music on hold.

We want to create an extension match that matches strings starting with a 7, and send the call to the AS5400:

```
exten => _7.,1,Dial(SIP/${EXTEN}@as5400)
```

Why do we use ${EXTEN} in the dial string in this case?

3. Configure the AS5400
The E1/T1 port will already be configured for you. First we need to create a translation rule to strip the leading 7 of incoming VoIP calls. This translation rule will be called 10+<group number> e.g. 101 for group one.

```
translation-rule 101
```

```
 Rule 1 ^71.% 1
 Rule 2 ^72.% 2
 Rule 3 ^73.% 3
 Rule 4 ^74.% 4
 Rule 5 ^75.% 5
 Rule 6 ^76.% 6
 Rule 7 ^77.% 7
 Rule 8 ^78.% 8
 Rule 9 ^79.% 9
```

The translation rule for prepending a 0 for outbound POTS calls is a little simpler. This rule will be called 20+<group number> e.g. 201 for group one.

```
translation-rule 201
 Rule 1 ^.% 0
```

Create a dial-peer to match incoming VoIP calls from your asterisk server (remember that a call consists of two dial-peers. This dial-peer will be called 100+<group number>, e.g. 1001 for group one:

```
dial-peer voice 1001 voip
 description calls from group 1
 answer-address 21..
 destination-pattern 21..
 translate-outgoing called 101
 session protocol sipv2
 session target ipv4:202.65.42.1
 dtmf-relay rtp-nte
 codec g711ulaw
 no vad
```

The answer-address line tells this dial-peer to match incoming calls from only your asterisk server. In this case, session protocol and target is correctly setup, however at this stage it is not being used as no inbound calls are configured.

Create a dial-peer to send calls out the first E1/T1 interface on the AS5400. This dial-peer will be called <group number>, so substitute your group's number when configuring this. We will be using the the translation pattern to prepend a 0 that we configured up earlier. For group one:

```
dial-peer voice 1 pots
 destination-pattern T
 translate-outgoing calling 201
 direct-inward-dial
 port 1/0:D
```

Once you have completed this, try making a call. If it fails, perform some debug both on your Asterisk server, and on the gateway to try and work out why.

Some useful IOS voice commands:

```
show call active voice brief     ! show voice calls currently in progress
show call history voice brief    ! show recent voice calls

show controller controller-id    ! show details of interface controller
show voice port                  ! show voice port configuration

show dial-peer voice [summary | id#]

test translation-rule            ! test a translation pattern
     e.g. test translation-rule 201 64212304323
```

Lab 8. CODECs

This lab aims to give you real world experience making calls over real satellite IP links with different CODECs and settings.

We will be making calls to an asterisk server back in New Zealand to an echo test, and the PSTN.

1. Asterisk configuration
You'll need the following SIP peer configured on your asterisk server (use pacnog<group number> for the peer name):

```
[nz]
type=friend
disallow=all
allow=alaw
callerid=your name <2100>
dtmfmode=rfc2833
canreinvite=no
nat=no
host=203.114.130.250
username=pacnog1
fromuser=pacnog1
secret=pacnog1
```

We'll se our dialplan up such that calls starting with 1 are sent to NZ, after having the 1 stripped:

In extensions.conf add a route to the NZ asterisk box:

```
exten => _1.,1,Dial(SIP/${EXTEN:1}@nz)
```

2. Check connectivity to NZ server
We want to see what the connectivity to the NZ server looks like, a ping and traceroute will give us an idea of what this is like. From a shell prompt on your asterisk server:

```
ping 203.114.130.250
mtr 203.114.130.250
```

What does the path look like?

What is the round trip latency? What might a voice call sound like?

3. Make some calls
Make a call to the NZ server. The NZ server is configured with the following dialplan:

200	echo test
222	music on hold
0...	domestic NZ numbers (e.g. 021 2304323 to dial Jonny's cellphone)
00...	international numbers (e.g. 00 682 xxxxx to dial Rarotonga)

(Careful when making PSTN calls, as call quality will be highly variable due to the limited bandwidth out of the workshop, and the number of simultaneous call attempts!)

Once you have successfully made a call, try further calls to the echo test and music on hold using other codecs. Change the 'allow=' line in your nz SIP peer to try the following codecs:

gsm	- GSM codec
ilbc	- Internet low bandwidth codec
ulaw	- G.711 ulaw
speex	- Speex codec
g726	- G.726.1 codec