Command Line Control of an Asterisk Confbridge

Darryn Anton Jordan
jrddar001@myuct.ac.za

June 25, 2015

Contents

0.1 Introduction

This document details the method used to configure an asterisk server and describes a C++ program used to record calls made in a confbridge.
The aim is to create a platform where node operators can group call through an Asterisk confbridge.

0.1.1 Software and Hardware Used

- Ubuntu 15.04
- Asterisk 13.1.0
- Code::Blocks 13.12
- SFLphone 1.4.1

0.2 Preparation and Set-up

0.2.1 Overview of system

Each node is assigned a SIP account and an extension number. This number can be used for one-on-one calls. Furthermore, a confbridge is created. This is essentially a conference call with a specific extension. Recording of the confbridge is achieved using a C++ program.

0.2.2 Asterisk Server

The server is required to have Asterisk installed:

```
sudo apt-get install asterisk
```

Once installed, the command line interface can be accessed as follows:

```
sudo asterisk -r
```

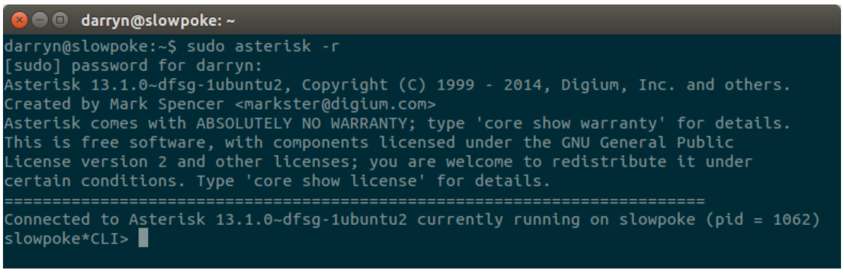

Figure 1: Asterisk CLI

The Asterisk server is configured by editing three *.conf* files:

- sip.conf - manage SIP accounts and set the server IP.
- extensions.conf - control what happens when extensions are dialled.
- confbridge.conf - configure a confbridge.

These files are located at:

```
/etc/asterisk
```

These files must be replaced with the provided ones, in order to allow recording. It is recommended to make a copy of the original contents before replacing.

0.2.3 Important CLI Commands

- Whenever a *.conf* file is altered, the *reload* command must be used to refresh the server.
- To restart the server, use *core restart now*.
- View users present in a confbridge, *confbridge list*.
- Commands can be made from terminal, without entering the CLI:

  ```
  sudo asterisk -rx "command"
  ```

  ```
  For example:
  sudo asterisk -rx "confbridge list"
  ```

0.2.4 Configuration File Descriptions

sip.conf

```
[general]
bindaddr=0.0.0.0:5060             ;listen on IPv4 wildcard, UDP default port
localnet=127.0.0.1/255.255.255.0 ;server IP, must be changed accordingly

[111]                             ;account username/extension number
type=friend                       ;account can make and recieve calls
host=dynamic                      ;dynamic IP address
secret=123                        ;account password

[222]                             ;create as many accounts as needed
type=friend
host=dynamic
secret=234

[333]
type=friend
host=dynamic
secret=345

[444]
type=friend
host=dynamic
secret=456
```

extensions.conf

```
[default]

exten => 100,1,Answer()                  ;if 100 is dialed, server answers
;ask node opperator for name and announce arrival to others
exten => 100,2,Set(CONFBRIDGE(user,announce_join_leave)=yes)
exten => 100,3,ConfBridge(100,NeXtRad)  ;link opperator to confbridge


;if other extension called, dial that number (one-on-one calls).
exten => _XXX,1,Dial(SIP/${EXTEN})
```

confbridge.conf

```
[general]

; --- ConfBridge User Profile Options ---
[default_user]
type=user
music_on_hold_when_empty=yes

; --- ConfBridge Bridge Profile Options ---
[default_bridge]
type=bridge

[NeXtRad]                                       ;confbridge name
type=bridge
record_file=/var/spool/asterisk/NeXtRAD.wav     ;save location
record_conference=no                            ;no record from start

; --- ConfBridge Menu Options ---
[sample_user_menu]
type=menu
*=playback_and_continue(conf-usermenu)
*1=toggle_mute
```

```
1=toggle_mute
*4=decrease_listening_volume
4=decrease_listening_volume
*6=increase_listening_volume
6=increase_listening_volume
*7=decrease_talking_volume
7=decrease_talking_volume
*8=leave_conference
8=leave_conference
*9=increase_talking_volume
9=increase_talking_volume
```

0.2.5 SIP Clients

SFLphone 1.4.1

This client was used during testing of the asterisk server. Note that the status must be registered in order to work.

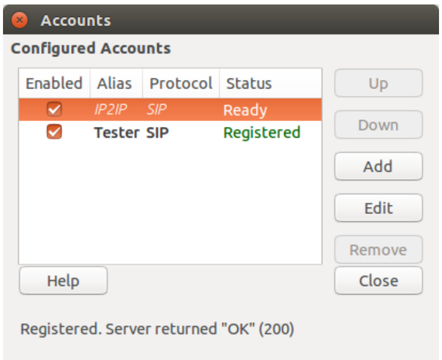

Figure 2: Account Settings

Figures 3 and 4 display the account settings of a functioning account.

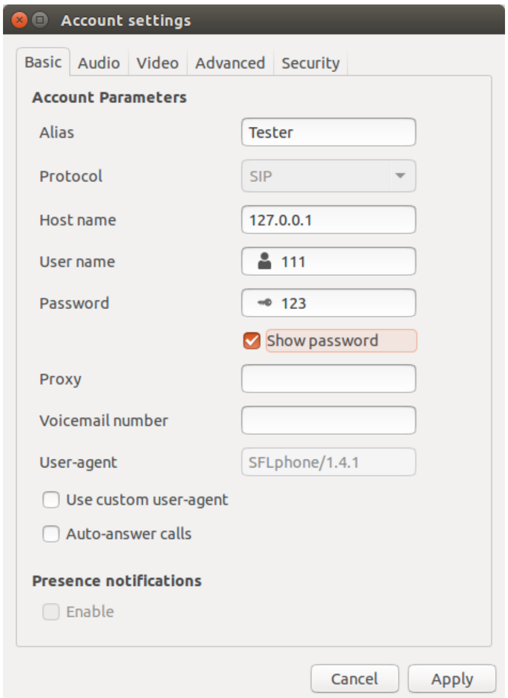

Figure 3: Basic Account Settings

NB Note that the port number is different to the one specified in the *sip.conf* file.

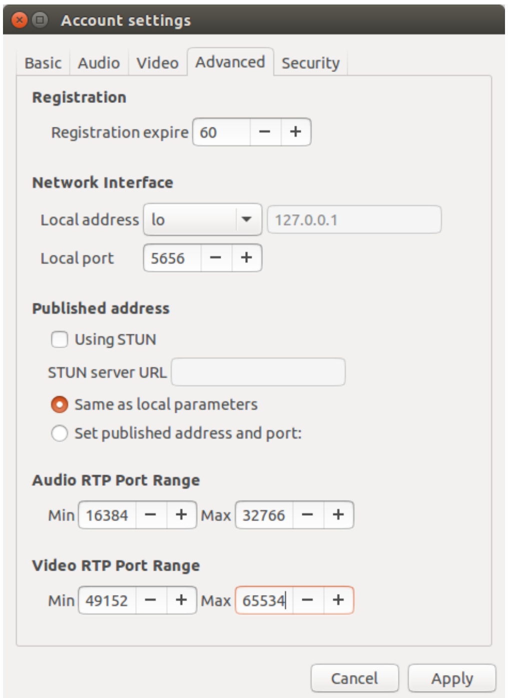

Figure 4: Advanced Account Settings

To test the server confbridge, dial 100 using SFLphone. The user is prompted to speak his/her name and then press the hash key. The user is announced to any other users present in the confbridge and the conference call commences.

Linphone 3.6.1

Linphone is a cross-platform SIP client (iOS, Android, Windows, OS X and Linux). It has been tested to work perfectly with this system on iOS and Linux. The set-up is extremely similar to the method described above. Friendly reminder: port number is 5656. (NOT 5060)

0.3 C++ Confbridge Recorder

Upon compiling the code in Code::Blocks, the following console application will appear:

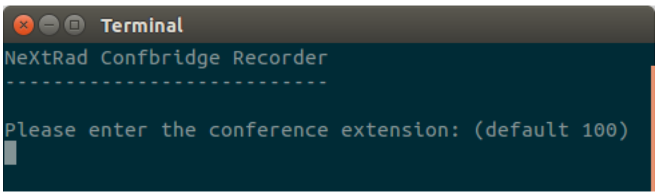

Figure 5: Welcome Screen

The user is then prompted to enter his/her password. If the program is successful in initiating recording, the output shown in figure 6 will appear. During development, the only location which Asterisk had permission to store recorded files was: /var/spool/asterisk

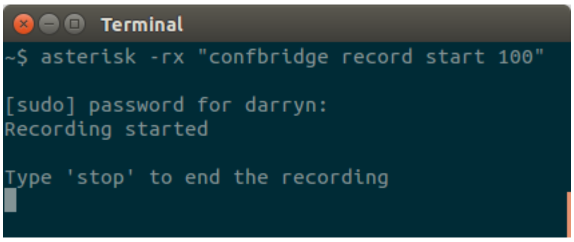

Figure 6: Recording Success

NB Note that a confbridge only exists once a user is present. Thus, a user must be in the confbridge before recording can take place. The following error occurs if no users are present:

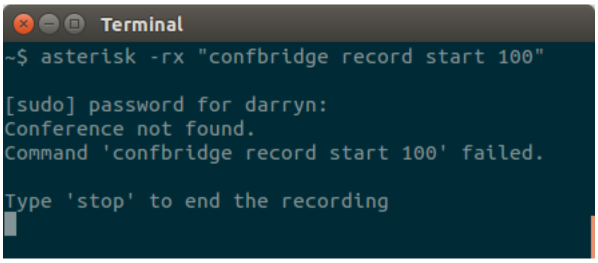

Figure 7: Confbridge Error

Upon typing *stop*,

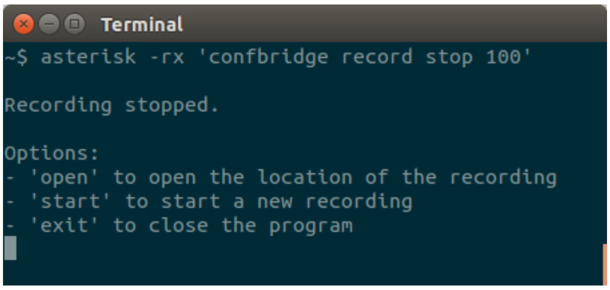

Figure 8: Final Options

Appendix A: Code Listing

```
//includes
#include <iostream>
#include <string>
#include <stdlib.h>

//namespaces
using namespace std;

//global variables
string confNumber;
string command;
string option;

//functions
void welcome();
void stop();
void start();

void start()
{
    system("clear\n"); //clear console
    cout << "~$ asterisk -rx \"confbridge record start " + confNumber + "\""
         << endl << endl; //print command to terminal
    command = "sudo asterisk -rx \"confbridge record start " + confNumber +
        "\""; //create string of command
    system(command.c_str()); //run command in terminal

    cout << "\nType 'stop' to end the recording" << endl;

    while(true)
    {
        cin >> option;
        if (option == "stop")
        {stop();}
    }
}

void stop()
{
    system("clear\n"); //clear console
    cout << "~$ asterisk -rx 'confbridge record stop " + confNumber + "'" <<
         endl << endl; //print command to terminal
    command = "sudo asterisk -rx \"confbridge record stop " + confNumber + "
        \""; //create string of command
    system(command.c_str()); //run command in terminal

    cout << "\nOptions: " << endl;
    cout << "- 'open' to open the location of the recording" << endl;
    cout << "- 'start' to start a new recording" << endl;
    cout << "- 'exit' to close the program" << endl;

    while(true)
    {
        cin >> option;
        if (option == "start")
        {start();} else
        if (option == "exit")
        {exit(0);} else
        if (option == "open")
        {system("cd /var/spool/asterisk");}  //unable to gain access
    }
```

```
}

void welcome()
{
    system("clear\n");
    cout << "NeXtRad Confbridge Recorder" << endl;
    cout << "---------------------------" << endl << endl;

    cout << "Please enter the conference number: (default 100)" << endl;
    cin >> confNumber;
}

int main()
{
    welcome();
    start();

    return 0;
}
```