Accession Testing Automated Tool Specification

Danny Maclean

Overview

This document specifies the functionality and behaviour of the Accession Testing Automated Tool. This will explain the framework, hardware and operation.

The goal of the Accession Testing Automated Tool (ATAT) is to be able to simplify the audio testing process for Accession Mobile. As changes are made to Accession, it is important to ensure voice quality standards are maintained or improved. Currently, there is not an automated tool to be able to do this within Metaswtich. With ATAT, the aim is to alleviate a significant portion of the manual, time consuming testing.

Tool Overview

The Accession Automated Testing Tool is a program that allows playing and recording of audio under varied network conditions. The tool passes input signals through a Wi-Fi repeater that varies the network parameters and records the output. In further stages of this project, the output will be arranged to conduct listening tests and produce objective, analytic analysis of audio quality.

Requirements

One of the main complaints of Accession Mobile is that the voice quality needs improvement. In the past, when a new version of Accession mobile was released, manual audio tests were conducted to confirm that voice quality didn’t regress. However, manual tests are error-prone, time consuming and tedious. Thus, automation is key. The time for setup and user interaction must be kept to a minimum.

Detailed Description

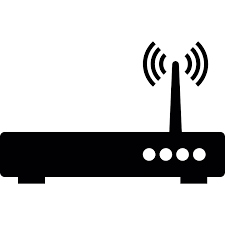
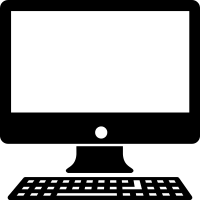
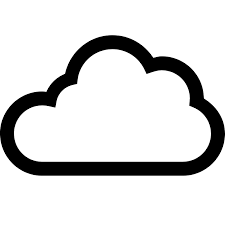


Figure : Functional Flow of the ATAT

Hardware Required

1. 2 USB PnP devices and cables
2. 2 Phones with the Accession Application to be tested downloaded
3. GL Mini Router and accompanying USB cable

Setup

1. Plug the GL Router into a USB port on the PC
2. Plug each USB PnP cable into a USB port on the PC
3. Plug each 3.5-millimetre jack to one of the phones
   1. Plug the grey cable, with 2 rings on the 3.5mm jack, into the phone with the application to be tested
4. Connect the PC and all 3 phones to the GL Router Wi-Fi network
5. Fill in the config.yml file
6. From the command line, type the name of the script

Default Settings

1. Default Input: The default input is 16 distinct WAV files sampled at 16KHz using 16 bit PCM (1 channel) provided from ITU-T (<https://www.itu.int/net/itu-t/sigdb/menu.aspx>). The default input is representative of voices from a variety of backgrounds consisting of different accents, tones, volumes and pronunciations. It is important to note that Accession Mobile also samples audio at 16Khz internally.
2. Default Output: Every output file is also saved as a WAV file, sampled at 16Khz and 256 kb/1(1 channel). The default output directory is in the root directory of the script. Please note
   1. The files within the output directory are organized into sub-directories based on the network parameters that the input file experienced.
   2. If an output directory from a previous run exists in the same location with the same name, it will be deleted entirely before the new output is saved.
3. Default Max\_jb\_size: This parameter specifies the largest time, in ms, that the jitter buffer on the receiving client attains. The default assumption is 250ms. This parameter is important if the jitter buffer will reach a size larger than 250ms. In that scenario, there is a risk that the file will not be recorded entirely and that the ending will be chopped.

Config Information:

Audio\_config.yml:

1. Input\_path: The absolute path for the user chosen input directory. This is optional. The default is detailed above and located in the root directory of the repository. Moving the default input will cause an error.
2. Output\_path: The absolute path for the user chosen output directory. This is optional. The default output is in the root directory and detailed above
3. Max\_jb\_size: The largest size that the Jitter Buffer will reach on the receiving client. (Note the receiving client is the device with a grey 3.5mm cable plugged in with 2 rings on the jack). This is important to prevent the program not recording the end of the call.
4. Login\_info: the IP address, username and password are required
5. Command\_1: The file has space for 5 commands. The structure for each is identical.
   1. Custom\_string: Enter a complete netem compatible command.
      1. Syntax: beginning with “tc qdisc change dev wlan-sta root …” and the custom command to be executed.
      2. This has highest priority, if entered, the rest of the command parameters are ignored
   2. Gemodel: this specifies the Gilbert Elliot model scheme with which to drop packets.
      1. Syntax: “1 10 70 0.1” (p r 1-h 1-k) move-to-burstmode (p) of 1%, move-to-gapmode (r) of 10%, drop-in-burstmode (1-h) of 70% and drop-in-gapmode (1-k) of 0.1%
      2. This has higher priority than packet\_loss\_percent. If a Gemodel is specified, the packet\_loss\_percent is ignored
   3. Packet\_loss\_percent: This emulates distributed packet loss and is a single number
      1. Syntax: “0.1” indicates to drop packets randomly with a probability of 0.1%
   4. Delay: This is a uniform delay applied to all packets
      1. Syntax: “100” indicated a uniform delay of 100ms on all packets
   5. Jitter\_normal\_distribution: This emulates jitter on all outgoing packets.
      1. Syntax: “10” indicates that the standard deviation of delay on every packet will vary by 10ms

Run Time Instructions

1. From the command line enter the repository and run ATAT.py
2. The script will SSH onto the router.
   1. If an error is encountered, the prompt “Could not connect to router, program will now exit” is displayed and the error message is displayed and the program exits
3. The script will then query the 2 USB PnP devices connected
   1. If 2 USB PnP’s are not detected, a prompt will be displayed detailing the error and the program will exit
4. The script will then play audio from one USB and listen on the other USB.
   1. If no audio is detected, the script will prompt you to start a call. Start a call and press enter.
   2. Type “okay” to ignore this message.
5. An example message is then played.
   1. In Audacity, set the device to monitor to USB PnP 1 or 2 and listen for the example message. If no audio can be heard: confirm the devices are using the 3.5mm jacks as the audio source, alternate the device that Audacity is monitoring audio from
   2. If the audio is distorted: turn down the volume on all devices, play that message again, repeat
6. A prompt to confirm the output directory is then displayed
   1. Press enter to continue or control-c to exit
7. Automated playing will now begin

Post Run Time Information

* The output directory is located at the path specified in the config file
* If audio distortion was detected, the name of the directory is labelled “corrupt” plus the standard name
* Inside a corrupt directory will be at least one file labelled “corrupt” plus the standard name.
* Inside the runtime\_log.yml file (located in the output directory) will be more information about the reason the file is corrupt

Diagnostics

Please check the runtime\_log.yml file located in the output directory for diagnosing issues and analysing statistics. For every file played, this includes

* Length of input file in 16 bit PCM samples
* Length of output file in 16 bit PCM samples
* Custom command
* Jitter Normal Distribution
* Packet Loss
* Gilbert Elliot Model statistics
* Latency between input and output
* Transmission Status (success, failure, clipped)

Performance Considerations

The runtime complexity of the program scales proportionally to the length of the input files and amount of conditions to be tested.

Deliverables

Upon completion, the input files, program and documentation will be available.

Rejected Alternative Approaches

1. Emulation of Android devices and Accession Mobile
   1. The android emulator does not allow the emulated phone to take .WAV files as input to the device.
   2. Emulating the VoIP call uses significant amount of disk space and slows down the computer drastically.
   3. Emulating the VoIP call also proved troublesome because the emulated device was not correctly connecting to the network and thus failing to receive or make calls.
2. An interactive GUI that users can use to specify input and output paths along with netem parameters
   1. This option has been neglected for now because of time constraints and is still open to be revisited
3. A Google Play application that can directly interface with Accession mobile
   1. This option has been neglected for now because of time constraints and is still open to be revisited