8th International Conference from "Scientific Computing to Computational Engineering"
8th IC-SCCE
Athens, 4-7 July, 2018

ASSISTIVE TECHNOLOGY TO SUPPORT PEOPLE WITH SPEECH IMPAIRMENT IN PHONE COMMUNICATION

Machigar Ongtang[1] and Nattanun Thatphithakkul*

[1]College of Innovative Technology and Engineering
Dhurakij Pundit University, Bangkok, Thailand
e-mail: machigar.ong@dpu.ac.th

* National Electronics and Computer Technology Center (NECTEC)
Thailand Science Park, Pathumthani, Thailand
e-mail: nattanun.thatphithakkul@nectec.or.th

Keywords: Assistive Technology, Accessibility, Softphone, SIP, RTP, Text-to-Speech

Abstract. *People with impaired speech encounter difficulties communicating especially over phone calls and thus require proper accessibility support. To advance the area of assistive technology for communication, we have designed and developed a Softphone SDK which makes and receives SIP calls, using Real-time Transport Protocol (RTP) as its media protocol. Unlike traditional Softphones, our Softphone SDK accepts audio stream from other applications or audio sources in addition to conventional device microphone. For demonstration, our prototype system comes with a demo application which utilizes Text-to-Speech to allow speech-impaired user to type sentences instead of speaking. The synthesized voice is converted into appropriate format, passed to our Softphone SDK, inserted to the call's outgoing audio buffer, and transmitted to the other end of the line. Our system has been evaluated to be practical for use and is planned to be deployed to assist people with disabilities in Thailand.*

1 INTRODUCTION

Hearing and speech disorders are one of the main causes of communication problems. Approximately 40 million Americans have communication disorders such as speech or hearing impairments. Approximately 7.5 million people in the United States have trouble using their voices, while between 6 and 8 million people in the U.S. have some form of language impairment [1]. People with this type of disability suffer lifetime costs due to inability or limited ability to study and work. However, there are limited number of accessibility solutions to assist this group of people to communicate over phone calls.

In this paper, we propose the development of an SDK that can be integrated smoothly with phone application. It enables the phone application to accept audio stream from other applications or audio sources, such as speech synthesizer, and use such audio stream to compensate the user's impaired voice.

Nevertheless, due to the security policy and mechanism embedded in most popular smartphone platforms, injecting audio stream from sources external to the platform is considered untrusted and thus disallowed. Therefore, we decide to apply our approach to a softphone application, which makes and receives SIP calls, using Real-time Transport Protocol (RTP) as its media protocol. Our softphone SDK is now available for both Android and iOS.

The rest of this paper is organized as follows. Section 2 outlines related research and industry contributions. Our system design and architecture are detailed in section 3. Section 4 describes how our system is implemented. Section 5 evaluates and discusses the outcome of our system. Finally, section 6 draws the paper to conclusion and presents some potential future work.

2 BACKGROUND AND RELATED WORKS

Assistive technology is technology used by individuals with disabilities in order to perform functions that might otherwise be difficult or impossible. It can include mobility devices such as walkers and wheelchairs, as well as hardware, software, and peripherals that assist people with disabilities in accessing computers or other information technologies [2]. For people with speech disorder, speech synthesizer, which performs artificial production of human speech, is widely used to assist their communication.

There are a broad variety of applications with Augmentative and Alternative Communication (AAC) features for the speech impaired. Predictable [3] is text-to-speech application for Android and iOS platforms. It simply speaks out the messages that the user types and also offers customizable AAC functions with the social media integration. More importantly, Predictable supports word prediction engine and voice recording. Touch Voice Gold [4] is another mobile application with AAC features for the speech impaired. By touching buttons on the screen, the application speaks for the user with a customizable human sounding computerized voice.

Proloquo2Go and Proloquo4Text provide voice synthesis for people who cannot or have difficulty speaking [5]. Proloquo2Go is a symbol-supported communication application to promote language development and grow communication skills, applicable to children or emerging communicators. On the other hand, Proloquo4Text is also a text-based AAC application but is designed for literate users.

Clearly, the aforementioned AAC applications only assist local communication. They do not provide accessibility support for communication over the phone. On the contrary, RogerVoice[6] uses automatic speech recognition and speech synthesis to provide call captioning for user with hearing impairment, and to allow users to write their own message to their correspondent, who will receive a voice response. However, RogerVoice only allows for VoIP outgoing calls.

Our Softphone SDK aims to provide assistive technology for communication over the phone. Our solution supports both outgoing and incoming calls to provide speech-impaired users with better communication experience.

3 SYSTEM DESIGN AND ARCHITECTURE

Our softphone SDK is designed and implemented on the top of Linphone [7], an open source SIP Phone. It comes in two separate versions, one for Android OS and the other for iOS platform. While different techniques are used to develop the required functionality for these two platforms, both of the SDK versions work nearly identically to provide the following functional requirements.

- The softphone SDK is developed as module that supports API call from other applications
- The softphone SDK aims to assist voice calls by accepting audio stream from other applications or audio sources in addition to conventional device microphone. Then, this stream is sent as outgoing audio stream to the remote phone client.
- Incoming audio stream from remote phone client is passed to the default call audio output (e.g. phone speaker).
- Outgoing call request from other modules is also supported to allow external module to initiate outgoing call via API call.

3.1 System Overview for Softphone SDK on Android

Our softphone SDK on Android consists of seven main components as shown in figure 1.

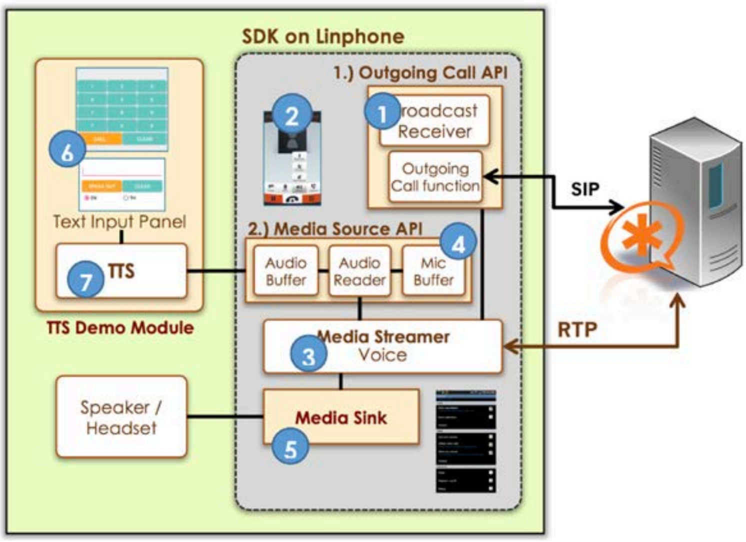

Figure 1: Architecture of Softphone for Speech Impaired on Android

1. Outgoing Call API to support outgoing call request from other modules. It receives call parameters, namely, outgoing call number.
2. In-Call Screen to show ongoing call status

3. Media Streamer that receive and send voice stream between the softphone SDK and other modules. It also communicates with the SIP server to send and receive voice stream via RTP protocol.
4. Media Source API receives voice stream from other module to deliver it to the media streamer
5. Media Sink API to send voice stream from the media streamer to another module

For demonstration, we use Text-to-Speech (TTS) as the demo to show how assistive audio stream from another source can be passed to our softphone SDK. The TTS demo module includes:

6. Text input panel to accept text typed by the user
7. Text-to-Speech (TTS) module to synthesize speech from the input text and generate voice stream

Outgoing Call API on Android

While incoming call works as in the original Linphone SIP phone application, the outgoing call includes support for API call from external module. Two components are built to enable this functionality.

- *Outgoing Request Broadcast Receiver* is created to listen to call request from other modules. It receives Intent (Android-specific messaging object) with a predefined action string. The SIP address for the outgoing call must be included in this Intent.
- *Outgoing call function* utilizes Linphone's outgoing call function. It accepts SIP address from the Outgoing Request Broadcast Receiver for making outgoing call. Then, it returns the in-call screen to show the call status to the user.

As show in Figure 2, to make an outgoing call request, the calling module must send out an Intent with the predefined action string and desired SIP address. The Outgoing Request Broadcast Receiver activates the outgoing call function to make the call and return the in-call screen.

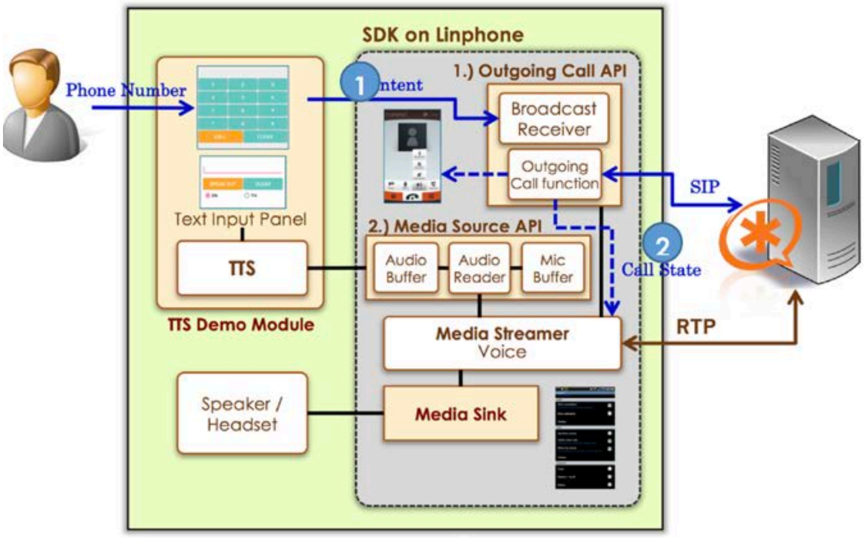

Figure 2: Outgoing Call API of Softphone for Speech Impaired on Android

Outbound Voice Streaming on Android

When the call is established, either by outgoing or incoming call, another application or module can deliver audio stream to the remote end via outbound streaming as shown in Figure 3. From the figure, part of the Media Source API that interacts with the external module consists of three main components as shown in Figure 4.

- *Audio Reader* reads binary audio data and delivers it to the media streamer. By default, this audio reader reads audio data from the phone's microphone buffer. For our assistive support, we modified its functionality to enable it to accept audio data from different audio buffer. In this case, it reads data into a byte array.
- *Mic Buffer* is the default data source for the audio reader. On Android, it reads from the device's microphone hardware via audio recorder with data source set to MIC or VOICE_COMMUNICATION.
- *Audio Buffer* accepts general audio data to changes data to byte array format to be readably by the audio reader.

Our softphone SDK reads binary data from audio buffer in form of byte array. Therefore, our mechanism is flexible enough to accept general forms of audio. However, the limitation comes from the format of the audio the other modules can acquire and send to our softphone SDK.

In the case of our demonstration, the TTS engine that comes with Android OS supports audio output only to hardware speaker or to file (.wav). For this reason, our demo module can only send .wav file to our softphone SDK. To support such input format, our softphone SDK supports .wav file reading and converting to byte array of binary data. This array of binary data is then passed to the audio buffer as outlined above.

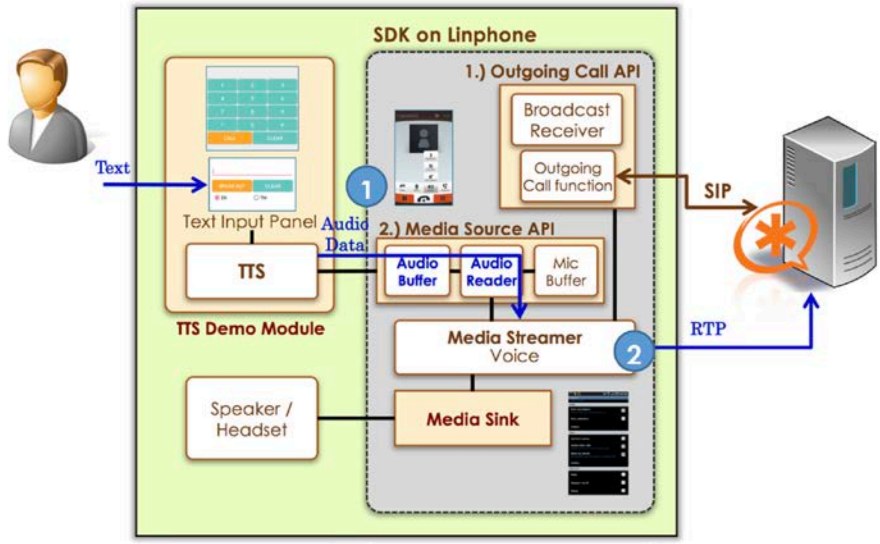

Figure 3: Outbound Voice Streaming of Softphone for Speech Impaired on Android

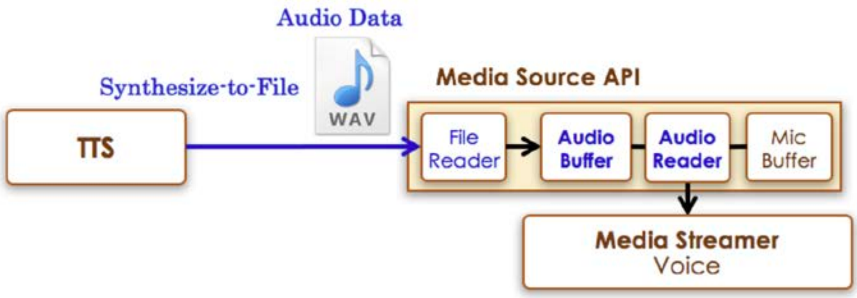

Figure 4: How audio data from TTS is delivered to Softphone SDK on Android

Note that our softphone SDK still allows reading of audio data from the microphone hardware. If there is audio data in the audio buffer, such data is delivered to the remote end; otherwise, the audio data from microphone buffer is delivered as usual.

Incoming Voice Streaming on Android

Inbound audio stream is accepted as displayed in Figure 5. The media streamer receives audio stream from the remote end via RTP protocol. It passes the stream to the Media Sink API which connects to the default call audio output of the device. This stream is then sent out as normal such as to the speaker, receiver, or the phone's headset.

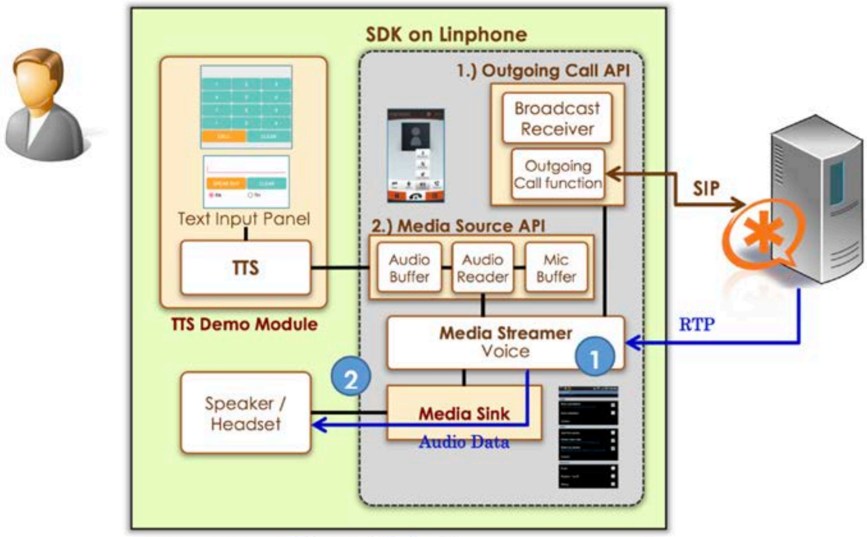

Figure 5: Inbound Voice Streaming of Softphone for Speech Impaired on Android

3.2 System Overview for Softphone SDK on iOS

Our softphone SDK on iOS consists of eight main components as shown in figure 6.

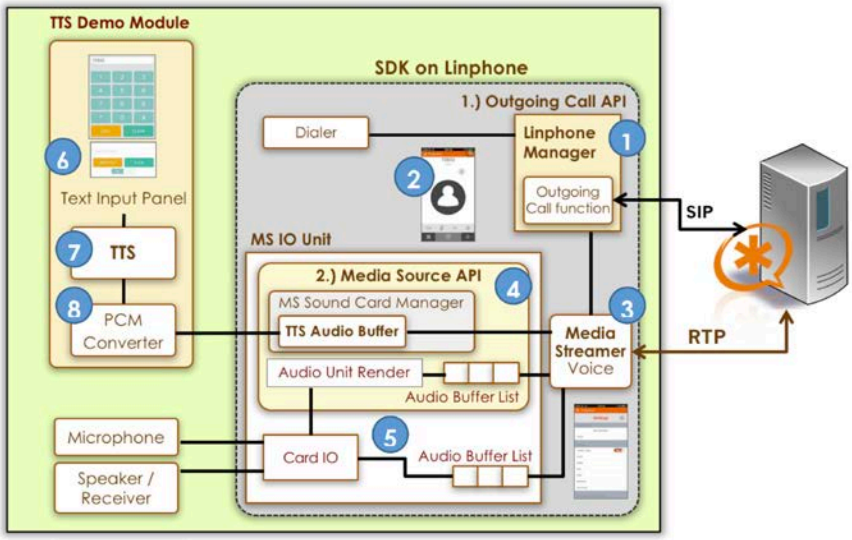

Figure 6: Architecture of Softphone for Speech Impaired on iOS

1. Outgoing Call API to support outgoing call request from other modules
2. In-Call Screen to show ongoing call status
3. Media Streamer that receive and send voice stream between the softphone SDK and other modules. It also communicates with the SIP server to send and receive voice stream via RTP protocol.
4. Media Source API in MS IO Unit receives voice stream from other module to deliver it to the media streamer
5. MS IO Unit sends voice stream from Card IO to media streamer and receives voice stream from media streamer to default outgoing bus of the Card IO.

The TTS demo module includes:

6. Text input panel to accept text typed by the user
7. Text-to-Speech (TTS) module to synthesize speech from the input text and generate voice stream
8. PCM adapter to convert audio data from mp3 format to PCM format

Outgoing Call API on iOS

Similar to Android, the incoming call works as in the original Linphone, while the outgoing call includes support for API call from external module. Two components are built to enable this functionality.

- *Dialer Function* is created to receive SIP address for the outgoing call and pass it to the outgoing call function.

- *Outgoing call function* uses Linphone's outgoing call function. It works in the similar way as in Android.

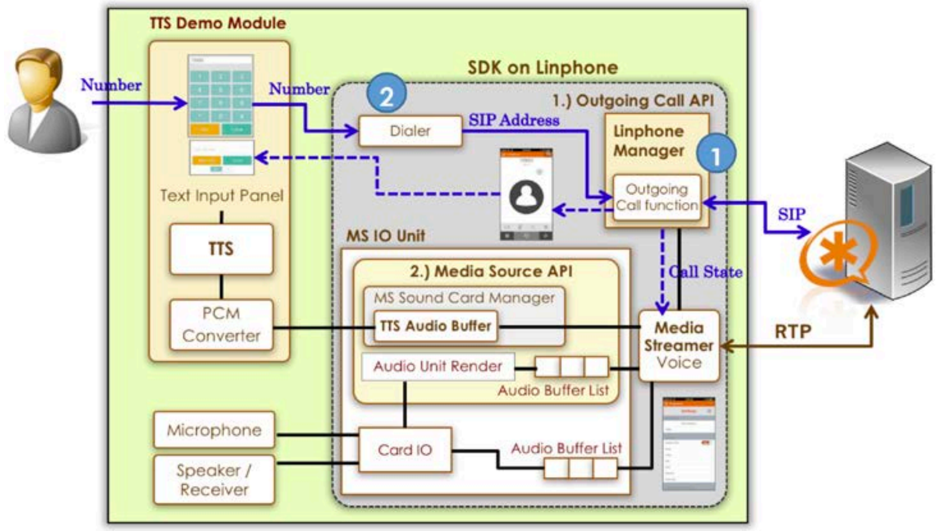

Figure 7: Outgoing Call API of Softphone for Speech Impaired on iOS

As show in Figure 7, to make an outgoing call request, the calling module must send out the desired SIP address to our dialer function.

Outbound Voice Streaming on iOS

When the call is established, either by outgoing or incoming call, another application or module can deliver audio stream to the remote end as shown in Figure 8 and 9. Main components are as follows.

- *PCM Converter:* The MS IO Unit receives audio data in PCM format. Thus, the PCM converter reads audio data from the TTS module and determines the original format. If it is not in PCM format (e.g. mp3), it converts the data into 16-bit PCM data format.
- *Audio Unit Render* is an original component in Linphone. It reads audio data from microphone hardware via Card IO and places the data on the Audio Buffer List. This list contains audio buffer in PCM format to be delivered to the media streamer.
- *TTS Audio Buffer* receives audio data in PCM format and stores in form of byte data. An API is created to accept audio data from other modules (e.g. TTS). When it is detected that there is audio data in TTS Audio Buffer, this audio data will be delivered to the media streamer instead of that in the Audio Buffer List of the Audio Unit Reader.

In our demonstration, we use Google TTS engine to synthesize audio stream in mp3 format. The PCM converter then converts it into PCM format and in turn sends it to the TTS Audio Buffer as outlined in Figure 9.

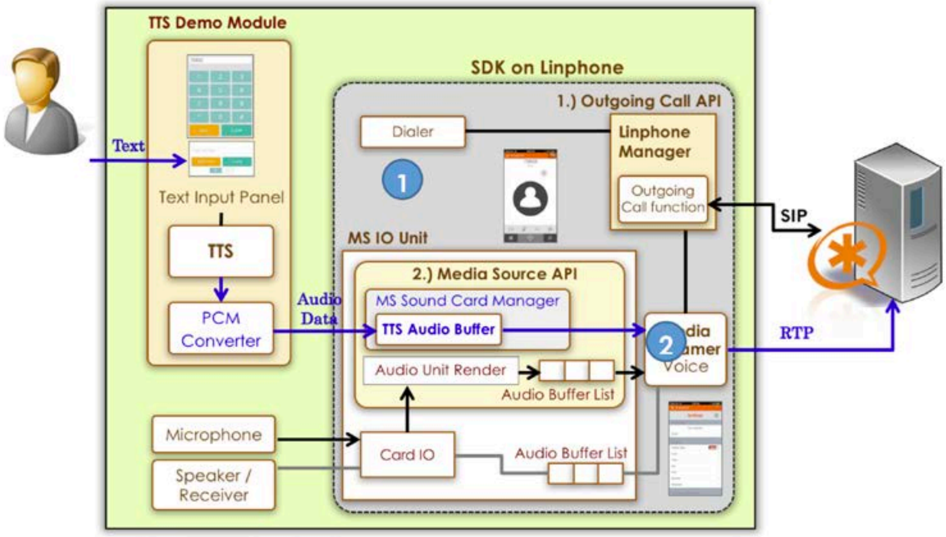

Figure 8: Outbound Voice Streaming of Softphone for Speech Impaired on iOS

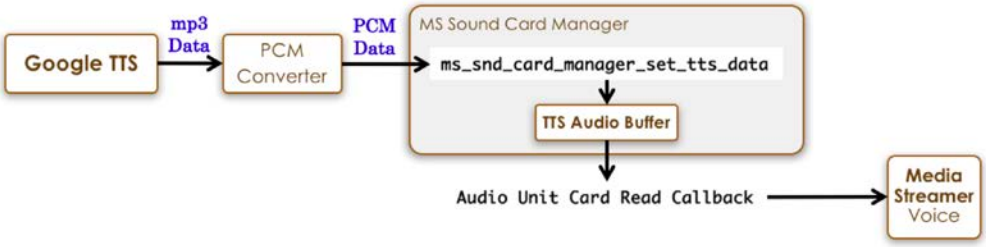

Figure 9: How audio data from TTS is delivered to Softphone SDK on iOS

Incoming Voice Streaming on iOS

From Figure 10, the media streamer receives audio stream from the remote end via RTP protocol. It passes the stream to the MS IO Unit to store to in Audio Buffer List. By default, the incoming audio data is sent to the speaker, receiver, or the phone's headset.

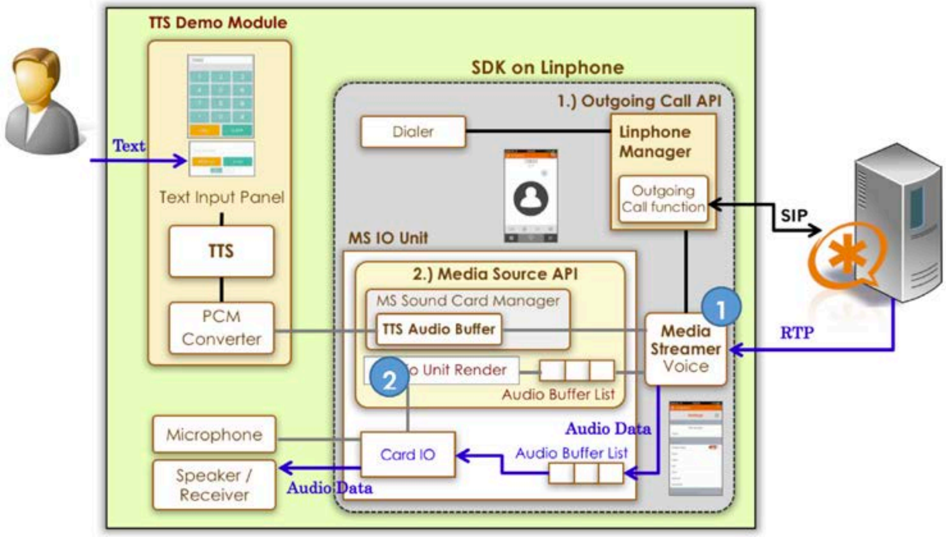

Figure 10: Inbound Voice Streaming of Softphone for Speech Impaired on iOS

4 SYSTEM IMPLEMENTATIONS

Due to space limitation, we show only our implementation on Android platform. The user interface of our application on iOS is nearly identical to that on the Android platform.

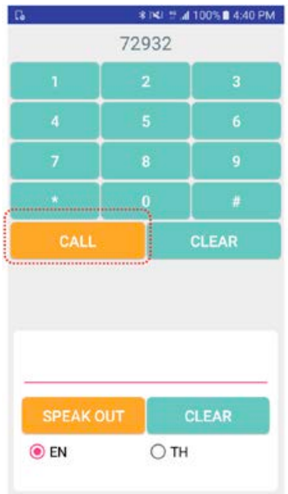

Figure 11: Make outgoing call

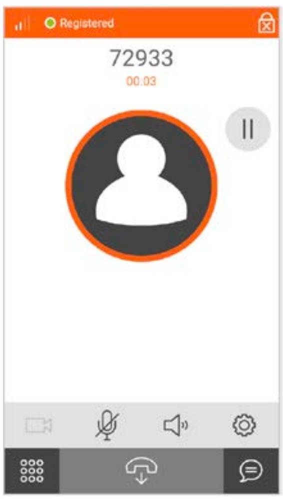

Figure 12: In-call screen

Figure 13: TTS activation with English input

Figure 14: TTS activation with Thai input

After initializing the softphone, the demo application can be launched. From the demo application, the user can input the SIP address and make outgoing call as shown in Figure 11. The in-call screen as in Figure 12 is shown when the call is established.

On the main screen of the TTS demo application, the user can type either English message or Thai message. Appropriate option must be selected to allow the TTS engine to work correctly. After submitting the message, the TTS engine will perform its task and the output synthesized audio will be passed to the remote end via our softphone SDK.

5 EVALUATION AND DISCUSSION

We have successfully developed and experimented our prototype system in our lab environment. Our system's operational model has been designed to be practical for real-world usage as it is part of an accessibility service project which has already been partially deployed. While more fine-tuning is likely to be required to support the streaming of long and continuing TTS audio data, the approach proposed in this paper is fairly promising to be included in our total communication service for accessibility.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we demonstrate the design and development of a softphone SDK and a demo module as assistive technology to support people with speech disorder in phone communication. The proposed system aims to make and receive SIP calls, using Real-time Transport Protocol (RTP) as its media protocol. Our Softphone SDK accepts audio stream from other applications or audio sources in addition to conventional device microphone. For demonstration, we use TTS as audio data source to allow speech-impaired user to type sentences instead of speaking. The synthesized voice is converted into appropriate format, passed to our Softphone SDK, inserted to the call's outgoing audio buffer, and transmitted to the remote end.

While our prototype system properly achieves the required functions, we still continuously improve the quality, smoothness, and continuity of the audio stream to make it ready for real-world deployment.

REFERENCES

[1] American Speech-Language Hearing Association (2013), *Communication Disorders: Prevalence and Cost in the United States*.

[2] National Center on Accessible Information Technology in Education (2011), *What is assistive technology?*.

[3] Therapy Box, Ltd. (2018), Predictable: Text based communication app, retrieved from https://itunes.apple.com/us/app/predictable-text-based-communication-app/id404445007 and https://play.google.com/store/apps/dev?id=4917466754585552320

[4] Touch Voice Business (2018), *Touch Voice Gold*, retrieved from https://itunes.apple.com/us/app/touch-voice-gold/id1072460911 and https://play.google.com/store/apps/details?id=com.touch_voice.touchvoicegold

[5] Origin Instruments Corporation (2018), *Proloquo2GO and Proloquo4Text*, retrieved from http://www.orin.com/access/Proloquo2Go/

[6] RogerVoice (2018), *RogerVoice Caption Call*, retrieved from https://itunes.apple.com/app/rogervoice/id1033113354 and https://play.google.com/store/apps/details?id=com.rogervoice.app

[7] Linphone, VoIP Software, http://www.linphone.org/.