VoiceTalk: Multimedia-IoT Applications for Mixing Mandarin, Taiwanese and English

YI-BING LIN *

College of Artificial Intelligence and Green Energy, National Yang Ming Chiao Tung University, China Medicine University, Department of Computer Science and Information Engineering, Asia University, College of Humanities and Sciences, Miin Wu School of Computing, National Cheng Kung University, liny@nctu.edu.tw

YUAN-FU LIAO

Department of Electronic Engineering, National Taipei University of Technology, yfliao@ntut.edu.tw

SIN-HORNG CHEN

Dept. of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, schen@mail.nctu.edu.tw

SHAW-HWA HWANG

Dept. of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, hsf@nctu.edu.tw

YIH-RU WANG

Dept. of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, yrwang@mail.nctu.edu.tw

The voice-based Internet of Multimedia Things (IoMT) is the combination of IoT interfaces and protocols with associated voice-related information, which enables advanced applications based on human-to-device interactions. An example is Automatic Speech Recognition (ASR) for live captioning and voice translation. Three major issues of ASR for IoMT are IoT development cost, speech recognition accuracy, and execution time complexity. For the first issue, most non-voice IoT applications are upgraded with the ASR feature through hard coding, which are error-prone. For the second issue, recognition accuracy must be improved for ASR. For the third issue, many multimedia IoT services are real-time applications and, therefore, the ASR delay must be short.

This paper elaborates on the above issues based on an IoT platform called VoiceTalk. We built the largest Taiwanese spoken corpus to train VT-ASR, the ASR model for VoiceTalk, and show how the VT-ASR mechanism can be transparently integrated with existing IoT applications. We consider two performance measures for VoiceTalk: speech recognition accuracy and VT-ASR delay. For the acoustic

The IoTalk and voice-based IoMT applications in this paper were implemented by the members in Lab 117 and the Quanta-NCTU AI Joint Center. This work was supported in part by the Center for Open Intelligent Connectivity from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan; in part by the Ministry of Science and Technology in Taiwan under Grant 108-2221-E-009-047, Grant 107-2221-E-197-006-MY3, Grant 108-2321-B-197-003 and 109-2221-E-027-108; and in part by the Ministry of Economic Affairs in Taiwan under Grant 107-EC-17-A-02-S5-007, Grant 107-EC-17-A-02-S5-007 and Grant 109-EC-17-A-02-S5-008.
tests of PAL-Labs, VT-ASR’s accuracy is 96.47%, while Google’s accuracy is 94.28%. We are the first to develop an analytic model to investigate the probability that the VT-ASR delay for the first speaker is complete before the second speaker starts talking. From the measurements and analytic modeling, we show that the VT-ASR delay is short enough to result in a very good user experience. Our solution has won several important government and commercial TV contracts in Taiwan. VT-ASR has demonstrated better Taiwanese Mandarin speech recognition accuracy than famous commercial products (including Google and Iflytek) in Formosa Speech Recognition Challenge 2018 (FSR-2018) and was the best among all participating ASR systems for Taiwanese speech recognition accuracy in FSR-2020.

CCS CONCEPTS  • Human-centered computing → Human computer interaction (HCI) → Interactive systems and tools

Additional Keywords and Phrases: Automatic Speech Recognition (ASR), Computational Linguistics (CL), Multimedia IoT, IoTtalk


1. INTRODUCTION

Traditional Internet of Things (IoT) approaches support smart applications with narrow-band data. Recently, the Internet of Multimedia Things (IoMT) has become popular, which supports rapid growth in multimedia-on-demand traffic (audio, video, and images). Specifically, IoMT associates multimedia-related information to enable advanced applications for cyber and physical integration. Multimedia formats raise new challenges to data transmission, processing, storage, and sharing [1]. In IoMT, two major multimedia types integrated into IoT are video and audio. Popular video-based IoMT applications are video games, video-based data fusion, and so on. Information fusion of IoMT integrates object detection and sensor feature information including gesture, moving trajectory, and gait, which are retrieved from multiple cameras, sensors, and wearable devices [2], [3], [4]. These IoMT examples integrate various video, audio, and sensor sources to make a more accurate observation of the environment. FusionTalk [2], for example, developed an object identification system using distributed cameras and IoT devices for better visualization and reconfiguration. Experiments were conducted to evaluate the data fusion algorithm, which showed that identification accuracy above 95% can be achieved. Video-based IoMT is out of the scope of this paper.

In this paper, we focus on IoMT applications that integrate voice with IoT. Voice-based IoMT is the combination of IoT interfaces and protocols with associated voice-related information, which enables advanced applications based on human-to-device interactions. The core technology of this integration is Computational Linguistics (CL), which concerns the processing of languages by a computing facility. CL has grown and developed exponentially for over fifty years [5]. However, the utilization of CL services in IoT-based multimedia applications has not received much attention so far. As quoted from [6], “CL has emerged as a new research paradigm for future computing applications.” The future of smart IoMT devices with Automatic Speech Recognition (ASR) is more important in real-time systems such as speech understanding, emotion recognition, and home automation.

Most traditional narrowband data for IoT only need to be handled by simple input and output mechanisms. To manipulate multimedia information in smart IoT applications, ASR techniques are essential for meeting IoMT application requirements for user experience. Three major issues of ASR for IoMT are IoT development cost, speech recognition accuracy, and execution time complexity. For the first issue, most non-voice IoT applications are upgraded with the ASR feature through hard coding, which are error-prone. For the second issue, recognition accuracy must be improved for ASR.
For the third issue, many multimedia IoT services are real-time applications and, therefore, the ASR delay (the voice-to-text translation delay and the transmission delay between the users and the ASR server) must be short. This issue has not been investigated in the literature.

The objectives of this study are to resolve the above three issues. We propose VoiceTalk, an IoT platform that utilizes ASR for various real-time IoT applications. We show how voice-controlled IoT applications can be conveniently developed in the VoiceTalk platform. We then investigate the performance of speech processing for IoT applications including automatic speech recognition. Finally, we study how the time complexity of the ASR service affects the user experience. In this paper, we make the following contributions:

- We propose the VoiceTalk approach, which can graphically, systematically, and transparently integrate the ASR mechanism into existing IoT applications. The idea is to implement the ASR mechanism as an IoT device and manage it like any other IoT device in the VoiceTalk GUI. Without VoiceTalk, a non-voice IoT application requires tedious programming effort to add the ASR feature, and our experience indicates that the resulting application through hard coding is error-prone.
- We build the largest Taiwanese spoken corpus and proposed the novel VoiceTalk ASR (VT-ASR). As compared with existing ASR solutions, VT-ASR has the best speech recognition accuracies for mixing Mandarin, Min-Nan, and English. In the commercial segment, our solution has been used by Public Television and Eastern Television with recognition accuracy higher than 90%, while the accuracies for other commercial solutions were between 60% and 70%.
- We develop a novel analytic model to conduct time complexity analysis. Together with measurements of the VT-ASR delay locally and remotely (in the cloud), our study shows a good user experience of VoiceTalk.

The paper is organized as follows. Section 2 overviews the related work; Section 3 proposes the VoiceTalk platform and elaborates on how traditional IoT applications can be upgraded to voice-based IoMT counterparts without difficulty; Section 4 proposes a novel automatic speech recognition mechanism that is deployed as an IoT device in VoiceTalk, and then evaluates its recognition accuracy for mixing Mandarin, Min-Nan, and English; Section 5 conducts analytic modeling to investigate the time complexity of VoiceTalk.

2. RELATED WORK

This section provides a literature review for voice-based IoMT in the following subsections.

2.1. Provision of an Accurate ASR Mechanism for IoMT

Accuracy for voice-to-text translation is the most important ASR performance index. To fulfill the requirements of real-life human-to-device interactions for Taiwanese people, large-scale Taiwanese multilingual spontaneous speech corpora and a sophisticated Taiwanese-specific ASR system that provides high accuracy and low latency simultaneously are indispensable.

First of all, mixing Mandarin, Min-Nan, and English in daily life speech is popular in Taiwan and China. We note that Taiwanese Mandarin and Min-Nan have many notable differences from the corresponding dialects in China (a.k.a, Putonghua and Hokkien, respectively) in terms of the writing system, pronunciation, accent, wording, and vocabulary. These differences may overlap and enhance each other, but many differences can also be attributed to the influence of Hakka, Formosan, Dutch, and Japanese languages. Therefore, it is well understood that large-scale Taiwanese multilingual spontaneous speech corpora are required to reflect the current status of the language.
There are a few Taiwanese Mandarin speech corpora [7] released by universities and organizations in Taiwan [8], Linguistic Data Consortium (LDC) [9], European Language Resources Association (ELRA) [10], and speech corpus companies such as SpeechOcean [11]. However, most of those corpora are small, containing only speech elicited in a single communicative context (e.g. news) or associated with a particular speech style (e.g. read speech and telephone speech). These corpora may not precisely reflect the current status of Taiwanese Mandarin speech. Furthermore, there are no large Taiwanese Min-Nan speech corpora publicly available. To resolve this problem, we build the two largest spoken corpora of Taiwanese Mandarin and Min-Nan to assist in training a Taiwanese-specific ASR system.

Secondly, it is well understood that a Taiwanese-specific ASR system is essential for better IoMT interaction in Taiwanese people’s daily life. ASR systems are increasingly powerful and accurate. Many commercial ASR systems, including the leading Google’s Cloud Speech-to-Text [12] and iFlyTek’s Open Platform [13] speech dictation services, are general-purpose ASRs that have not been localized properly nor adapted rapidly enough to deal with the dynamically evolving languages used in Taiwan. For example, iFlyTek has not officially supported Traditional Chinese, the majority writing system in Taiwan. Moreover, none of those ASR systems supports the recognition of Min-Nan speech. Therefore, using the two corpora of Taiwanese Mandarin and Min-Nan, we propose VT-ASR, which outperforms these commercial products as follows. We first note that several speech recognition methods have been proposed, including a variation of hidden Markov model/deep neural network (HMM/DNN) [20] [21] and end-to-end-based [14] [15] [16] [17] [18] [19]. Among them, the most common acoustic modeling method is the hybrid HMM/DNN approach. The study in [20] first proposed to use a feed-forward neural network that takes several frames of coefficients as input and produces posterior probabilities over HMM states as output and has been shown to outperform GMMs on a variety of speech recognition benchmarks. Later, [21] reported a conversational speech recognition system that reached human parity via combining convolutional and LSTM acoustic model architectures and recurrent neural network (RNN)-based language modeling approaches.

On the other hand, end-to-end methods now attract a significant amount of attention. The study in [14] introduced a connectionist temporal classification (CTC)-based approach to directly label un-segmented speech sequences. The study in [15] further represented a single Deep Speech 2 ASR pipeline that addresses the entire range of speech recognition contexts, including noisy environments, accents, and different languages handled by humans. Furthermore, the study in [18] presented Listen, Attend and Spell (LAS), an attention-based transformer neural network that learns to directly transcribe speech utterances to characters. The study in [16] described a transformer approach that combines a multi-head attention mechanism with CTC and language model (LM) to achieve better ASR performance. The study in [17] showed that a very deep model with up to 48 Transformer layers for both encoder and decoders can exceed performance from previous end-to-end approaches and even match the conventional hybrid systems. The study in [19] proposed that the convolution-augmented transformer for speech recognition, named Conformer, significantly outperforms previous Transformer-based models. However, the main challenges of end-to-end-based approaches are their high computation complexity and real-time decoding issues, which are usually hungry for advanced GPUs.

Compared with those end-to-end models, the hybrid HMM/DNN approach usually consumes less computing power and could run on a modest CPU. In particular, the approaches in [22] [23] successfully applied SpecAugment-based data augmentation on hybrid HMM/DNN models without increasing model size or training time, and still give state-of-the-art performance as recently shown for benchmarks like Librispeech and Switchboard. Therefore, to simultaneously fulfill the requirement of high accuracy and low latency, this paper proposes VoiceTalk ASR (VT-ASR), a sophisticated hybrid HMM/DNN model. To build an optimal multilingual ASR system with fast decoding, we have evaluated several DNN models including convolutional neural networks (CNN) [24], long short-term memory (LSTM) [25], time-delay neural
network (TDNN) [26], factorized form of TDNN (TDNN-f) [27] and their combinations. Among them, CNN is very good at automatic feature engineering and has been shown to improve ASR robustness [28]. LSTM takes advantage of its recurrent connection to model the context of time-varying speech signals well but is difficult to execute in parallel. On the other hand, TDNN adds delays to their neural network architectures to model temporal pattern/trajectory, which can be nicely executed by the GPUs for acceleration. TDNN-f [27] is structured similarly to TDNN but compresses its connection weights via singular value decomposition (SVD). Compared with conventional TDNNs, TDNN-f uses only one-fourth of parameters with much less required computation power. In this paper, we propose to use the CNN+TDNN-f architecture as the acoustic model (AM) of VT-ASR. For the language model (LM), the 4-gram model is adopted. To further improve the performance of VT-ASR, recognizing Taiwanese Mandarin spontaneous speech, the Chinese Gigaword version 2.0 text corpus (written texts with 1.1 billion characters) [37] and the NER text corpus (spoken texts with 60 million characters) are incorporated into the training of Sub-models 4 and 5 (see Figure 7 in Section 4) for enriching the lexicon and the n-gram model. This version of VT-ASR is denoted as CNN+TDNN-f+Gigaword+NER-text.

We have compared our solution (CNN+TDNN-f+Gigaword+NER-text) with the related studies (including LSTM, TDNN, CNN+TDNN+LSTM, and CNN+TDNN-f). We used combined Mandarin and English language resources (including phones, lexicons, and texts), and conducted Mandarin, English, and mixed Mandarin-English speech recognition experiments for a fair comparison. The training data comprise 8 speech corpora with statistics shown in Table 1.

<table>
<thead>
<tr>
<th>Language</th>
<th>Corpus</th>
<th># of hours</th>
<th># of speakers</th>
<th># of utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandarin</td>
<td>NER-210 [29]</td>
<td>210.02</td>
<td>1,096</td>
<td>35,186</td>
</tr>
<tr>
<td></td>
<td>TCC300 [30]</td>
<td>26.40</td>
<td>300</td>
<td>27,375</td>
</tr>
<tr>
<td></td>
<td>MATBN [31]</td>
<td>127.40</td>
<td>280</td>
<td>29,549</td>
</tr>
<tr>
<td></td>
<td>Aishell-1 [32]</td>
<td>169.00</td>
<td>340</td>
<td>134,424</td>
</tr>
<tr>
<td></td>
<td>Thchs-30 [33]</td>
<td>27.90</td>
<td>50</td>
<td>10,893</td>
</tr>
<tr>
<td>English</td>
<td>Librispeech-clean [34]</td>
<td>100.60</td>
<td>251</td>
<td>28,539</td>
</tr>
<tr>
<td></td>
<td>OC16-CE80 [35]</td>
<td>63.80</td>
<td>1,303</td>
<td>58,132</td>
</tr>
<tr>
<td>Mixed</td>
<td>SEAME [36]</td>
<td>95.10</td>
<td>138</td>
<td>94,034</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>820.22</td>
<td>3,758</td>
<td>418,132</td>
</tr>
</tbody>
</table>

Moreover, eleven corpora are formed and used as the test data. Specifically, to target the performance of Taiwanese Mandarin speech, one training set NER-210 (the first 210 hours of the NER corpus), and three test sets of NER-core (studio-quality segments of the NER corpus), NER-other (non-studio quality), and NER-noisy (corrupted by background noise) were designed. Table 2 lists the statistics of these 11 test sets.

<table>
<thead>
<tr>
<th>Language</th>
<th>Corpus</th>
<th># of hours</th>
<th># of speakers</th>
<th># of utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandarin</td>
<td>NER-core [29]</td>
<td>1.75</td>
<td>35</td>
<td>438</td>
</tr>
<tr>
<td></td>
<td>NER-other [29]</td>
<td>3.23</td>
<td>23</td>
<td>640</td>
</tr>
<tr>
<td></td>
<td>NER-noisy [29]</td>
<td>1.78</td>
<td>23</td>
<td>347</td>
</tr>
<tr>
<td></td>
<td>MATBN</td>
<td>3.06</td>
<td>273</td>
<td>729</td>
</tr>
<tr>
<td></td>
<td>Aishell-1</td>
<td>9.00</td>
<td>20</td>
<td>7,176</td>
</tr>
<tr>
<td></td>
<td>Aishell-2 [38]</td>
<td>29.40</td>
<td>1,991</td>
<td>39,675</td>
</tr>
<tr>
<td></td>
<td>Thchs-30</td>
<td>6.30</td>
<td>10</td>
<td>2,495</td>
</tr>
<tr>
<td>English</td>
<td>Librispeech-other [34]</td>
<td>5.10</td>
<td>33</td>
<td>2,939</td>
</tr>
<tr>
<td></td>
<td>Librispeech-clean [34]</td>
<td>5.40</td>
<td>40</td>
<td>2,620</td>
</tr>
<tr>
<td>Mixed</td>
<td>OC16-CE80</td>
<td>7.93</td>
<td>142</td>
<td>7,099</td>
</tr>
<tr>
<td>Language</td>
<td>Corpus</td>
<td># of hours</td>
<td># of speakers</td>
<td># of utterance</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>SEAME</td>
<td></td>
<td>13.70</td>
<td>18</td>
<td>12,104</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>87.65</td>
<td>2,608</td>
<td>76,262</td>
</tr>
</tbody>
</table>

Figure 1 shows that our solution CNN+TDNN-f+Gigaword+NER-text outperforms other ASR systems for all test sets. Our solution significantly outperforms CNN+TDNN-f by 27.4% and 59.5% of the relative error rate reduction for the two relatively difficult tasks of the NER-other test set and the NER-noisy test set, respectively. These results indicate the importance of the spoken text corpus (i.e. NER-text) for Taiwanese Mandarin spontaneous (or spoken) speech recognition.

![Figure 1: The performance (average Chinese character and English word error rates) of different DNN architectures](image)

**2.2. UPGRADE AN EXISTING NON-VOICE IOT APPLICATION INTO ITS IOMT COUNTERPART CONTROLLED BY VOICE**

Many IoT applications are intelligently controlled by sensors or smartphone soft buttons (see [39] [40] [41] and the references therein). On the other hand, there are many turnkey solutions for voice-based IoT, including Amazon ECHO (Alexa Voice Service) [42], Apple Siri, Google Home [43], and more.

Voice-based IoMT applications can be classified into two categories: voice translation and voice control [39]. Voice translation applications include translations for voice to text, language (text) to language (text) and text to voice. We will focus on the voice-to-text translation mechanism. Examples include live captioning (e.g., for TV or video), voice transcription (voice writer or shorthand writer), and simultaneous interpretation (e.g., for conferences). A voice transcription application for hospitals is illustrated in Figure 2 (a). In this example, the conversations between members of the medical staff and patients are recorded by transcription machines (Figure 2 (1)) and are later stored in the nursing
station (Figure 2 (3)). In this application, a member of the medical staff uses a transcription machine connected to a transcription server in the cloud through the path (1)→(2). After speech recognition processing at the server, transcripts are sent to the nursing station through the path (2)→(3).

Figure 2: Voice translation for hospital and conference meetings.

An example of a simultaneous interpretation application is on-site or remote conferencing. In this application, people speaking different languages participate in a conference. An on-site or a remote participant in the conference meeting uses a voice interpreter to communicate with other participants (Figure 2 (4)). The voice-to-text transcripts are shown on the participants’ computer screens (Figure 2 (5)) or on a big conference projector. A live captioning example will be given in Section 3.

In terms of remote control, there is a five-stage evolution. Two of the five stages use voice-based IoMT technologies.

1. At Stage 1, specific remote-control devices were developed. Eugene Polley (Figure 3 (a)) invented the first wireless TV remote control Flash-Matic in 1955 and was awarded an Emmy in 1997. This innovative invention (Figure 3 (b)) is a flashlight-like device that remotely controls photocells mounted in four corners of a TV screen to switch channels. However, the invention was still somewhat flawed and often misjudged when operating on sunny days [44]. Polley’s idea has been intensively improved, and the resulting wireless (e.g., infrared) remote-control devices have become very popular today [45].

2. Since Stage 2, smartphones have been considered remote-control devices. Back in 1990, remote-controller applications with cellular phones were developed using the short message service [46]. After its invention, the smartphone has become a popular remote-control device [45] [47] [48]. Stages 1 and 2 of the remote-control evolution do not utilize voice-based IoMT technology.

3. In Stage 3, voice-activated remote-control devices were invented. To eliminate the need of pressing buttons, a new class of smart devices entered the home: smart speakers and personal voice assistants [42]. Such services continuously monitor conversations, which are transported to a cloud back-end where they are processed and stored [43]. Many companies use commercial smart speakers to adapt their “smart home” products with voice services. Two examples of smart speaker products are Google Home and Amazon Echo. In [42] the authors argued that smart speakers may create a dangerous situation wherein a hacker may compromise a smart speaker service to disrupt all “smart home” services. Therefore, we should use these commercial smart speaker products carefully.
4. Stage 4 ushers in a new method of automatic remote control, conducted through sensing. At this stage, the remote control system “smartly” learns how to change the statuses of the appliances based on the conditions of the environment [49]. For example, when the CO2 sensor detects that air quality is poor in the room (Figure 3 (c)), the system automatically turns on the purple lights of the plant boxes, which activates the photosynthesis process to improve air quality (Figure 3 (e)). The smart control of this stage does not utilize voice-based IoMT technology.

5. At Stage 5, the interactive remote control is provided through ASR. At Stage 4, machine learning is typically used to “guess” the house owner’s preference of the conditions in a room environment. However, if the prediction is wrong, the system’s actions may disappoint the house owner. To avoid this, the Stage 5 system could ask permission before taking any action. To do so, we have designed an interactive remote-control application for elderly care wherein a cute robot called Zenbo follows the house owner’s movements like a puppy (Figure 4). Zenbo makes environment-sensitive conversations with the homeowner to ascertain his or her desires. This type of IoMT has not been found in the literature, which can be easily implemented in VoiceTalk (the details are given in Appendix A.1).

Figure 3: (a) Eugene Polley (1915-2012); (b) Flash-Matic; (c), (d) and (e) The air quality control application.

For all of the solutions in the first four stages, none of the previous studies elaborate on how to provide an effective and systematic way to integrate the ASR mechanism with existing non-voice IoT applications. We show that this goal can be
achieved by the VoiceTalk platform. We also point out that existing studies have not investigated the effects of the delays for cloud-based ASR, which will be analytically modeled in this paper.

Figure 4: ZenboTalk for interactive remote control.

3. VOICETALK NETWORK ARCHITECTURE

The VoiceTalk approach integrates the ASR mechanism with an IoT application development environment called IoTtalk [50]. In our approach, all hardware devices and software modules connected to VoiceTalk are considered IoT devices. Specifically, we develop a novel ASR mechanism as an IoT device called VoiceTalk ASR (VT-ASR), and manage VT-ASR just like an IoT device in VoiceTalk’s GUI.

We use an automatic live caption example to explain how VoiceTalk works. In this example, the camera in a TV studio (Figure 5 (1)) connected to a cyber IoT device “Multimedia-I” (Figure 5 (2)) consisting of two software modules called the Sensor/Actuator Application (SA; Figure 5 (3)) and Device Application (DA; Figure 5 (4)). The SA is responsible for connecting the physical multimedia or the IoT devices (e.g., cameras, sensors, and so on), and implements the algorithms for the Multimedia IoT device (e.g., video compression and encryption). The DA supports IoT communications to interact with the VoiceTalk server (Figure 5 (5)) based on the HTTPS or the MQTT protocols. The low-layer communications technologies of the DA can be Bluetooth, WiFi, LTE or 5G. The VoiceTalk server consists of two components: the IoTtalk engine (Figure 5 (6)) for processing the IoT data and the VoiceTalk Graphical User Interface (GUI; Figure 5 (14)) for configuring the connections of the IoT devices.

The DA of Multimedia-I provides two one-way data transmission channels Audio-I and Video-I to the VoiceTalk server called input device features (IDFs). An IDF name is ended with “-I” to indicate the data-flow direction from the IoT device to the VoiceTalk server. The VoiceTalk ASR mechanism is implemented as an IoT device VT-ASR (Figure 5 (7)) with two one-way data transmission channels including an IDF Text-I and an output device feature Voice-O. An output device feature (ODF) is ended with “-O” to indicate the data-flow direction from the server to an IoT device. Details of the VT-ASR SA (Figure 5 (9)) are described in Section 4. The device Multimedia-O receives the video (Video-O), the audio (Audio-O) and the texts (Text-O) from its DA (Figure 5 (11)). The SA of the device (Figure 5 (12)) then inserts the text into the appropriate position of the video/audio stream (Figure 5 (13)). The VoiceTalk GUI (Figure 5 (14)) is used to configure the connections from the IDF to the ODFs. The developer can remotely access the web-based VoiceTalk GUI through any computing device with a browser (such as a desktop or a smartphone; see Figure 5 (15)). The example in Figure 5 can be created in VoiceTalk GUI as a project VoiceTalk1 illustrated in Figure 6 (a).
In the VoiceTalk GUI, each of the input and the output parts of an IoT device is represented by an icon; e.g., (B) and (C) for VT-ASR in Figure 6 (a). Inside a device icon, there are one or more small icons that represent the DFs, where all DFs are grouped in an input device icon placed on the left side of the GUI window; e.g., (1) and (2) are grouped in (A) of Figure 6 (a). Similarly, all DFs are grouped in an output device icon placed on the right side of the GUI window; e.g., (4), (5) and (6) are grouped in (D) of Figure 6 (a). To connect a data transmission channel of an input device (e.g., Video-I in (A)) with another channel of an output device (e.g., Video-O in (D)), one drags a line to connect the IDF icon to the ODF icon (e.g., Join 2). In this configuration, the video and the audio streams will go to Multimedia-O directly through Joins 1 and 2, respectively; see the paths (1)→(4) and (2)→(5). The audio stream is also directed to VT-ASR through Join 1 (see the path (1)→(7)). The SA of VT-ASR (Figure 5 (9)) processes the data received from Video-O (through the internal transition (7)→(12)), and the voice-to-text transcript are then sent to Multimedia-O through Join 6 (see the path (12)→(6)).

Through VoiceTalk GUI, an example of voice transcription for hospital application (Figure 2 (a)) can be straightforwardly implemented as the VoiceTalk2 project illustrated in Figure 6 (b), which is a simple extension of VoiceTalk1 by adding Voice-I in icon (B), and Text-O in icon (C). Details of adding DFs in an IoT device can be found in [50]. In this application the web-based transcription machine is a Multimedia-I IoT device ((A) in Figure 6 (b)), which receives the voice inputs through Audio-I, short video clicks through Video-I and text inputs from the keyboard through Text-I. In VoiceTalk2, the nursing station is represented as the Multimedia-O IoT device ((D) in Figure 6 (b)). The original audio, video and text inputs can be sent directly to the nursing station through Join 1 ((1)→(4)), Join 2 ((2)→(5)) and Join 3 ((3)→(6)). The conversations, for example, between a patient and a nurse are sent to the output part of the VT-ASR IoT device through Join 1; i.e., (1)→(7). Similarly, the texts are sent to VT-ASR through Join 3; i.e., (3)→(9). VT-ASR performs voice-to-text translation in its SA (through the internal transition (9)→(12)) and sends the results from Text-I of the input part of VT-ASR ((12) in Figure 6 (b)) to Text-O of Multimedia-O ((6) in Figure 6 (b)).

In a simultaneous interpretation application (Figure 2 (b)), people speak different languages in a conference using voice translators. Like the hospital scenario in Figure 2 (a), this application can be directly implemented in VoiceTalk2 with smartphones. In this scenario, an instance of Audio-I in the Multimedia-I device ((1) in Figure 6 (b)) is the microphone of a smartphone, and an instance of Text-O ((6) in Figure 6 (b)) in the Multimedia-O device is the display of a smartphone.
VoiceTalk allows the smartphones of all conference participants to connect to the same Multimedia-I and Multimedia-O for multicast communications. An on-site or a remote participant in a conference simply uses his/her smartphone to scan the conference QR code, then the smartphone is automatically bound to VoiceTalk2; that is, the smartphone becomes a web-based voice translator for this conference (the bounding process is described in [51]). The smartphones of all participants in the conference can be bound to Multimedia-I simultaneously to serve as the translators. In this way, no specific personal voice assistant devices are required. Similarly, the big screen and the computer screens of the participants are bound to Multimedia-O. Therefore, all paths ending at (4), (5), and (6) in Figure 6 (b) are multicast paths. For example, the speech of a participant is interpreted into English to play to the speakers of all participants through the path (1)→(7)→(10)→(4) in Figure 6 (b) or shown in the screens of all smartphones through the path (1)→(7)→(12)→(6). A participant can also use the keyboard of a smartphone to type the texts. Then VT-ASR translates these texts into the languages of the receiving participants. The path is (3)→(9)→(12)→(6).

Figure 6: The configurations for voice translation: (a) automatic live captioning and (b) hospital and conference applications

We note that the beauty of VoiceTalk is that we only need to develop a general configuration of IoT device icon connections, and then by binding the icons to various physical IoT devices (the microphones in a TV studio, the web-based transcription machines in a hospital, and the web-based voice translators in a conference meeting), many services are immediately created without the need of writing new application programs.

In Appendix A.1, we will elaborate on how the IoMT applications for voice control can be implemented in VoiceTalk. In particular, we will show that existing IoT applications in IoTtalk can be upgraded to voice-controlled IoMT applications with low effort. The demonstrated examples include interactive swing light poles [52], interactive hollow light globe [53], PlantTalk [54][55], ToiletTalk [56] and ZenboTalk [57].

For live captioning and simultaneous interpretation, it is important that the voice-to-text translation results are synchronized with the translated sources. "Live" television may be delayed a few seconds because the TV station needs a little window to add close captioning. Such a process is done automatically in the SA of Multimedia-O (Figure 5 (12)). The time complexity for the speech recognition and the communications of VoiceTalk is analyzed in Section 5.
4. VT-ASR: THE ASR MECHANISM FOR MIXING MANDARIN, TAIWANESE AND ENGLISH

The SA of VT-ASR provides a streaming multilingual automatic speech recognition (ASR) service targeting Taiwanese Mandarin, English, Taiwanese Min-Nan, and their mixture. To fulfill the requirements of multilingual ASR for Taiwanese people, we have built the two largest spoken corpora of Taiwanese Mandarin and Min-Nan to assist in training VT-ASR as a Taiwanese-specific ASR system. The first corpus is a real-life, multi-genre and spontaneous Taiwanese Mandarin broadcast speech corpus with manual transcription from the digital archive of Taiwan’s National Education Radio (NER) [58] [59][60]. The NER corpus was publicly released in 2020. Table 3 summarizes the statistics of the NER corpus. The table indicates that the 21-volume, 3200-hour NER corpus is the largest Taiwanese Mandarin spoken corpus. This corpus is also the largest Chinese spoken text (instead of writing text) database with about 60 million traditional Chinese characters.

The second corpus is a multi-channel Taiwanese Min-Nan read-speech corpus. This corpus is called TAT (Taiwanese Across Taiwan) corpus [61]. The TAT corpus contains 300 hours x 6 channels of read speech produced by 600 speakers. It is the largest Taiwanese Min-Nan speech corpus. The first two volumes of the TAT corpus have been well-transcribed and publicly released in 2021. Table 4 shows their statistics.

Table 3: Summary of the NER corpus

<table>
<thead>
<tr>
<th>Sets</th>
<th>#. of hours</th>
<th>#. of files</th>
<th>#. of words</th>
<th>#. of characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>clean</td>
<td>624.71</td>
<td>89,908</td>
<td>6,638,286</td>
<td>12,983,842</td>
</tr>
<tr>
<td>other</td>
<td>2,581.07</td>
<td>400,495</td>
<td>19,362,579</td>
<td>47,800,272</td>
</tr>
<tr>
<td>Total</td>
<td>3,205.78</td>
<td>490,403</td>
<td>26,000,865</td>
<td>60,784,114</td>
</tr>
</tbody>
</table>

Table 4: Summary of the first two volumes of the TAT corpus

<table>
<thead>
<tr>
<th>Sets</th>
<th>#. of speakers</th>
<th>#. of hours</th>
<th>#. of files</th>
<th>#. of characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAT-Vol1</td>
<td>100</td>
<td>51.94</td>
<td>28,833</td>
<td>339,592</td>
</tr>
<tr>
<td>TAT-Vol2</td>
<td>100</td>
<td>52.42</td>
<td>28,978</td>
<td>340,607</td>
</tr>
<tr>
<td>Total</td>
<td>200</td>
<td>104.36</td>
<td>57,811</td>
<td>680,199</td>
</tr>
</tbody>
</table>

By incorporating the NER and the TAT corpora into the ASR model training, VT-ASR is developed to be a Taiwanese-specific multilingual ASR system. The system is discussed as follows. Given input speech signals \( A \) (Voice-O in Figure 6 (7)), VT-ASR transforms \( A \) as a feature vector sequence \( O \) to find in real-time the most likely Chinese, English, Taiwanese Min-Nan or mixed Chinese-English word sequence \( \hat{W} \) (Text-I in Figure 6 (12)). VT-ASR uses several speech and text corpora including the NER and TAT corpora to train an HMM/DNN model [24] [27][28], and computes \( \hat{W} \) as

\[
\hat{W} = \arg\max_w P(W|O) = \arg\max_w P(O|W)P(W) = \arg\max_w P(O|Q)P(Q|L)P(L|W)P(W)
\]

to maximize a posteriori probability \( P(W|O) \) by 5 sub-models illustrated in Figure 7. In this figure, Sub-model 1 performs speech feature vectors extraction, which transforms an input speech signal \( A \) to mel-scaled cepstral coefficients (MFCCs) \( O \). Sub-model 2 (Figure 7 (2)) performs frame classification by dividing each phoneme from all languages into several tri-phone HMM states and applying DNN to compute frame-level state observation probabilities \( P(O|Q) \). Sub-model 3 (Figure 7 (3)) builds tri-phone HMMs by accumulating the state likelihoods of each tri-phone for calculating segment-level phone scores \( P(Q|L) \). Sub-model 4 (Figure 7 (4)) builds word models to compute word hypotheses scores \( P(L|W) \), which applies multilingual pronunciation lexicon for word hypotheses evaluation. Sub-model 5 (Figure 7 (5))
constructs the language model for scoring $P(W)$, which adopts a mixed-language n-gram model for optimal Chinese, English, Taiwanese Min-Nan, and their mixture word sequence decoding.

Sub-model 2 merits further discussions. To build an optimal multilingual ASR system with fast decoding, we have evaluated several DNN models described in Section 2.1, and adopted the CNN+TDNN-f architecture as the acoustic model (AM) of the system. After some preliminary experiments, we cascade a 6-layer CNN and a 12-layer TDNN-f to establish our multilingual ASR system. In the design, the Inverse Discrete Transform (IDCT) block (Figure 8 (1)) first converts MFCCs back to spectrogram to provide time-frequency cues. The 6-layer CNN (Figure 8 (2)) then automatically learns how to extract and concentrate optimal speech features layer-by-layer using trainable convolutional filters and down-sampling operations. The 6 layers of CNN have 64, 64, 128, 128, and 256 filters, respectively. Each layer also has 2560 Rectified Linear Units (RELU’s) with Batch Normalization (BN). In this way, CNNs improve ASR robustness [28]. In the 12-layer TDNN-f (Figure 8 (3)), each layer has an SVD-based compression block with an input/output dimension of 1536 and a hidden bottleneck block with a dimension of 160 (except that the first layer is 256) to reduce the number of parameters. Each layer also has a resnet-style bypass-connection from the output of the previous layer, and a “continuous dropout” schedule. The TDNN-f is trained using the Lattice Free Maximum Mutual Information (LF-MMl) criterion to result in high-accuracy performance with fast decoding capability [27]. Finally, a Softmax unit (Figure 8 (4)) is applied to compute tri-phone state posterior probabilities of all target languages. As pointed out in Section 2.1, the Chinese Gigaword text corpus and the NER text corpus are incorporated into the training of Sub-models 4 and 5 for enriching the lexicon and the n-gram model. This version of VT-ASR is denoted as CNN+TDNN-f+Gigaword+NER-text.

The effectiveness of the VT-ASR on Taiwanese Mandarin and Min-Nan speech recognition was then examined. For Mandarin speech recognition, the VT-ASR performed very well to reach a traditional Chinese character recognition accuracy of 90.5% [59] in the Formosa Speech Recognition Challenge 2018 (FSR-2018, task: building an ASR system for recognizing real-life, multi-genre broadcast Taiwanese Mandarin speech) [62]. In the same challenge, the accuracies of the two general-purpose commercial ASRs of Google and iFlyTek were 79.4% and 81.8%, respectively. In the 2020 Formosa Speech Recognition Challenge (FSR-2020) [63], two test sets of the TAT corpus and the corpus of Taiwan Public Television Service (PTS) Taigi Channel TV Shows (Table 5) were used to evaluate Taiwanese Min-Nan speech recognition systems. The challenge had two tracks (for recognizing spontaneous and read Taiwanese Min-Nan speech to output Taiwanese Min-Nan Hanji character and syllable, respectively) and was open to both the academic and industrial sectors without any language resource limit. As the challenge host, we also evaluated VT-ASR in parallel using the same CNN+TDNN-f architecture and mainly the TAT-Vol1 and TAT-Vol2 corpora.
Table 5: Summary of the FSR-2020 test corpora

<table>
<thead>
<tr>
<th>Track</th>
<th>Corpus</th>
<th># of hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taiwanese Min-Nan Hanji Character</td>
<td>PTS Taigi Channel TV Shows</td>
<td>20.5</td>
</tr>
<tr>
<td>Taiwanese Min-Nan Syllable</td>
<td>TAT-test</td>
<td>10.5</td>
</tr>
</tbody>
</table>

Figures 9 and 10 show the Taiwanese Min-Nan Hanji Character Error Rates (CERs) and Syllable Error Rates (SERs) of all ASR participants on the two challenge tracks, respectively. The best systems achieved 37.5% Taiwanese Min-Nan Hanji CER and 10.2% SER on the two tracks, respectively. Compared with the best ASR systems participated in FSR-2020, VT-ASR has shown better performance than CER (34.98%) and SER (9.83%). In conclusion, VT-ASR is better than all participating ASR systems tested in the FSR-2020 Taiwanese Min-Nan speech recognition competition.

We used AI, IoT and analytic modeling to develop VoiceTalk. Then our study bridged the gap between theory and practice by commercializing VoiceTalk with excellent customer feedback. VT-ASR has won several important government contracts in Taiwan. Depending on the applications of the contracts, several versions of VT-ASR were trained using the NER corpus and some field data. In each task, iterative model updating procedures were continuously applied to fine-tuning Sub-models 2, 4, and 5 while the accumulated field data, i.e., speech and their transcriptions, reached a certain amount. For example, Sub-model 2 was retrained when every 100 hours task-specific speech data were collected. On the other hand, Sub-models 4 and 5 were adapted more often to follow language evolution along the time by adding out-of-vocabulary (OOV) words into the lexicon, such as new person’s names, proper nouns or slang mentioned in recent TV News or online social media (for Sub-model 4 adaptation), and by interpolating pre-trained general-purpose and new created field-specific n-gram models (for Sub-model 5 adaptation), respectively. In one contract, our solution provided live captioning for the 2020 Taiwan presidential debates to engage an audience of millions [64]. In another contract, VT-ASR...
provided live captioning for the Legislature of Taiwan. Figure 11 shows the recognition accuracies of VT-ASR for the speech recordings of 19 legislators in the 10th term of Legislative Yuan in Taiwan [65].

![Figure 9: Performance (CER for Taiwanese Hanji Characters) of the participated ASR systems in FSR-2020](image)

![Figure 10: Performance (SER for Taiwanese Syllable) of the participated ASR systems in FSR-2020](image)

VT-ASR has also been used by the Ministry of Health and Welfare (MOHW) for various services including the Q&A tasks for the long-term care customer services in the call centers of 21 counties in Taiwan and the captioning tasks for the TV broadcasting of the daily CDC press conference for COVID-19. For the call center service, the MOHW conducted a third-party verification executed by the PAL-Labs, an independent acoustic test company [66]. The PAL-Labs used randomly sampled recordings of real live inbound telephone conversations handled by call center agents for the testing, where the VT-ASR’s accuracy is 96.47% and the Google’s accuracy is 94.28%. The recognition accuracy for the daily CDC TV captioning from April to June 2020 is on average 92.3% (Figure 12). It is interesting to note that the lowest point (83.7%) in Figure 12 was due to the event of “磐石艦” (pan shi jian, Panshi Battleship) COVID-19 group infection, where “磐石艦” is an OOV.

In the commercial segment, VT-ASR has been used by Public Television [67] and Eastern Television to generate subtitles for TV talk shows (mostly debates). The average recognition accuracy was higher than 90%. On the other hand,
the accuracies for other commercial solutions were between 60% and 70%, since those services may not deal well with various phenomena of Taiwanese spontaneous speech, such as interjections, filled pauses, ill-formed sentences, and new words.

![Figure 11: The recognition accuracies (average Chinese character and English word recognition rates) of VT-ASR for the speech recordings of 19 senators in the 10th term of the Legislative Yuan in Taiwan.](image)

5. THE VT-ASR DELAY ANALYSIS

In the voice-to-text translation applications, the VT-ASR delay for speech recognition of VoiceTalk is a key performance index for user experience. In a remote conference meeting, multiple participants take turns to speak. The VT-AST delay should be short so that the first speaker is complete before the second speaker starts talking. Otherwise, the captions will overlap with wrong speeches. This section proposes an analytic model to investigate the above phenomenon. Consider the timing diagram in Figure 13. Suppose that a participant (denoted as the 0-th speaker) speaks for a speech period that ends at time $T_0$. After a pause period $t_{0,2}$, the following participant speaks for another period $t_{1,1}$ and then another pause period $t_{1,2}$. Let the speech period of the $i$-th subsequent participant be $t_{i,1}$ and the following pause period be $t_{i,2}$. Let $t_d$ be the VT-ASR delay to translate a sentence in $t_{0,1}$ (i.e., the delay for the path (1)→(7)→(12)→(6) in Figure 6 (b)). Note that when the 0-th speaker talks, the voice is simultaneously sent to the VoiceTalk server for ASR processing. Therefore, the
complete speech transcript of the 0-th speaker is received by other participants (or the conference projector) at time $T_0 + t_d$. By convention, $t_0 = t_{0.2}$ in Figure 13, and $t_{i-1} \leq t_d \leq t_i$ for some $i$. We say that the 0-th speech period and its voice-
to-text translation are “synchronized” if and only if $t_d = t_0 = t_{0.2}$. Clearly, if $t_d \geq \tau_i$, the user experience becomes worsen for a larger $i$. Typically, the user experience is very good if $t_d \leq \tau_0$. The user experience is acceptable if $\tau_0 \leq t_d \leq \tau_1$. The listeners can tolerate interpretation delay for the 0-th speaker when the first speaker is talking. However, the user experience is NOT acceptable if $t_d \geq \tau_2$. In this section, we derive the probability $\Pr[t_d \geq \tau_j]$. This probability is used to evaluate the time complexity of VoiceTalk, and a good user experience is achieved if $\Pr[t_d \geq \tau_0]$ approaches 0.

**Figure 13:** The timing diagram.

In Figure 13, we have

$$\tau_i = t_{0.2} + \sum_{j=1}^{i} (t_{j,1} + t_{j,2}) \quad (1)$$

Let $t_{i,1}$ be i.i.d. random variables with the probability density function (PDF) $g_1(t_{i,1})$, and $t_{i,2}$ be i.i.d. random variables with the PDF $g_2(t_{i,2})$. Let the Laplace transform of $g_1(t_{i,1})$ and $g_2(t_{i,2})$ be $g_1^*(s)$ and $g_2^*(s)$, respectively. Let $t_i = t_{i,1} + t_{i,2}$, then $t_i$ is a random variable with the PDF $g(t_i)$ and the Laplace transform $g^*(s)$, where

$$g(t_i) = \int_{t_{i,1}=0}^{\tau_i} g_2(t_i - t_{i,1})g_1(t_{i,1})dt_{i,1} \quad \text{and} \quad g^*(s) = g_1^*(s)g_2^*(s) \quad (2)$$

Let $\tau_i$ be a random variable with the PDF $h_i(\tau_i)$ and the Laplace transform $h_i^*(s)$, respectively. Then from Eq. (1), for $i \geq 0$,

$$h_i(\tau_i) = \int_{\tau_{i-1}=0}^{\tau_i} g(t_i)h_{i-1}(\tau_{i-1})dt_{i-1}$$

For $i = 0$, $h_0^*(s) = g_2^*(s)$. For $i \geq 1$,

$$h_i^*(s) = g_2^*(s)[g^*(s)]^i = g_2^*(s)[g_1^*(s)g_2^*(s)]^i \quad (3)$$

and the probability $\Pr[t_d \leq \tau_i]$ is expressed as

$$\Pr[t_d \leq \tau_i] = \int_{\tau_i=0}^{\infty} h_i(\tau_i) \int_{t_d=0}^{\tau_i} f_d(t_d) dt_d dt_i$$
\[
\frac{1}{\pi T} \lim_{\tau \to \infty} \int_{-\tau}^{\tau} h''(\sigma) \left( \frac{f^*(s - \sigma)}{s - \sigma} \right) d\sigma \\
\frac{1}{\pi T} \lim_{\tau \to \infty} \int_{-\tau}^{\tau} g''(\sigma) \left[ g(\sigma) g^*(\sigma) \right]^i \left( \frac{f^*(s - \sigma)}{s - \sigma} \right) d\sigma
\]

(4)

In the above equation, the integral is performed along with the vertical line \( \text{Re}(\sigma) \) that lies entirely within the region of convergence of \( h'' \). If \( t_d \) is an Erlang random variable with the shape parameter \( n \) and the scale parameter \( \lambda \), then its PDF and the Cumulative Distribution Function (CDF) are

\[
f_d(t_d) = \frac{\lambda^n t_d^{n-1} e^{-\lambda t_d}}{(n-1)!} \quad \text{and} \quad F_d(t_d) = 1 - \sum_{j=0}^{\infty} \frac{\lambda^j t_d^j e^{-\lambda t_d}}{j!}
\]

(5)

Note that the Erlang distribution or a mixture of Erlang distributions are typically used to model IoT message delays and CPU processing times [68][69]. Further assume that \( g_1(t_{12}) \) and \( g_2(t_{12}) \) are Gamma PDFs with the shape parameters \( \alpha_1 \) and \( \alpha_2 \) and the scale parameters \( \beta_1 \) and \( \beta_2 \), respectively. A Gamma distribution is a generalization of an Erlang distribution. Let \( E[t_{12}] = \eta_1 E[t_{24}] \) and \( E[t_{14}] = \eta_2 E[t_{24}] \). We use Eq. (5) to derive equations (6)-(27) in Appendix A.2, and finally obtained \( \text{Pr}[t_d \geq \tau_k] \) and \( \text{Pr}[t_d \geq \tau_0] \) as

\[
\text{Pr}[t_d \geq \tau_k] = \frac{\alpha_1 \lambda^{\kappa_1} \alpha_2 (k+1)^{\alpha_2}}{(\eta_1 + \alpha_1)^{\kappa_1} (\eta_2 + \alpha_2)^{k+1}\alpha_2}
\]

(28)

and

\[
\text{Pr}[t_d \geq \tau_0] = \left( \frac{\alpha_2}{\eta_2 + \alpha_2} \right)^{\alpha_2} \sum_{j=0}^{\infty} \left( \frac{\alpha_2 + j - 1}{\eta_2 + \alpha_2} \right) \left( \frac{\alpha_2}{\eta_2 + \alpha_2} \right)^j
\]

(29)

Eqs. (28) and (29) are validated by Simulation following the approaches in [70][71]. The discrepancies between the analytic model and simulation are within 1%.

We have measured \( t_d \) in commercially operated VoiceTalk environments where the VoiceTalk server and the VT-ASR server are collocated. Two network configurations are considered. In the local network configuration, these two servers and the users of the IoT applications reside on the same site. The histogram of the measured \( t_d \) in this configuration (named \( t_{L,d} \)) is illustrated in Figure 14 (a), where \( E[t_{L,d}] = 0.1723 \) sec and \( V[t_{L,d}] = 0.0346^2 \). In the remote network configuration, the VoiceTalk/VT-ASR servers are running in a medium-end virtual machine (VM) in Amazon Web service located in San Francisco and the users of the IoT applications reside in Taiwan. The VM’s hardware specifications include Intel Xeon W2123 CPU (4 Cores/3.6GHz), 16GB memory (DDR4 2666 ECC RDIMM) and 512GB storage (M.2 PCI-E). The histogram of the measured \( t_d \) in this configuration (named \( t_{R,d} \)) is illustrated in Figure 14 (b), where \( E[t_{R,d}] = 0.2044 \) sec and \( V[t_{R,d}] = 0.1099^2 \).

Based on the above \( t_d \) measurements, our experiences with pause and speech periods and the studies in [72][73], we obtained the delays in two application scenarios. In a company meeting scenario, we compute the expected value and variance of \( t_{1,2} \) as \( E[t_{C,1}] = 20.0 \) sec and \( V[t_{C,1}] = 0.92^2 \). The expected value and variance of \( t_{1,2} \) are \( E[t_{C,2}] = 0.610 \) sec, and \( V[t_{C,2}] = 0.589^2 \). In a TV interview scenario, we compute the expected values of \( t_{1,1} \) and \( t_{1,2} \) as \( E[t_{T,1}] = 0.610 \) sec, and \( V[t_{T,1}] = 0.589^2 \).
180.0 sec and $E[t_{\tau,2}] = 4.0$ sec. The combinations of the two network configurations and the two application scenarios result in four cases:

**Case 1** (a company meeting with the local network): From Eqs. (26) and (27) in Appendix A.2, the $n$, $\alpha_1$, $\alpha_2$, and $\eta_2$ values are derived as $n_{1,1} = 25$, $\alpha_{1,1} = 640.923$, $\alpha_{1,2} = 1.073$, $\eta_{1,1} = 116.077$ and $\eta_{1,2} = 3.540$. Denote $t_d$ and $t_l$ in this case as $t_{L,d}$ and $\tau_{C,l}$ respectively. Then from Eqs. (28) and (29), we have

$$\Pr[t_{L,d} \geq \tau_{C,0}] = 8.797 \times 10^{-3} \quad \text{and} \quad \Pr[t_{L,d} \geq \tau_{C,1}] = 2.036 \times 10^{-48}$$

**Case 2** (a TV interview with the local network): The $n$, $\alpha_1$, $\alpha_2$, and $\eta_2$ values are derived as $n_{2} = 25$, $\alpha_{2,1} = 38279.773$, $\alpha_{2,2} = 46.12$, $\eta_{2,1} = 1044.689$ and $\eta_{2,2} = 23.215$. Denote $t_d$ and $t_l$ in this case as $t_{L,d}$ and $\tau_{T,l}$ respectively. Then from Eqs. (28) and (29), we have

$$\Pr[t_{L,d} \geq \tau_{T,0}] = 1.888 \times 10^{-51} \quad \text{and} \quad \Pr[t_{L,d} \geq \tau_{T,1}] = 1.110 \times 10^{-464}$$

**Case 3** (a company meeting with the remote network): The $n$, $\alpha_1$, $\alpha_2$, and $\eta_2$ values are derived as $n_{3} = 3$, $\alpha_{3,1} = 640.923$, $\alpha_{3,2} = 1.073$, $\eta_{3,1} = 97.847$ and $\eta_{3,2} = 2.984$. Denote $t_d$ and $t_l$ in this case as $t_{R,d}$ and $\tau_{C,l}$ respectively. Then from Eqs. (28) and (29), we have

$$\Pr[t_{R,d} \geq \tau_{C,0}] = 0.103 \quad \text{and} \quad \Pr[t_{R,d} \geq \tau_{C,1}] = 1.634 \times 10^{-41}$$

**Case 4** (a TV interview with the remote network): The $n$, $\alpha_1$, $\alpha_2$, and $\eta_2$ values are derived as $n_{4} = 3$, $\alpha_{4,1} = 38279.773$, $\alpha_{4,2} = 46.12$, $\eta_{4,1} = 880.626$ and $\eta_{4,2} = 19.569$. Denote $t_d$ and $t_l$ in this case as $t_{R,d}$ and $\tau_{T,l}$ respectively. Then from Eqs. (28) and (29), we have

$$\Pr[t_{R,d} \geq \tau_{T,0}] = 8.296 \times 10^{-15} \quad \text{and} \quad \Pr[t_{R,d} \geq \tau_{T,1}] = 5.165 \times 10^{-393}$$

In Cases 1, 2 and 4, $\Pr[t_d \geq \tau_0]$ approach 0, which imply that the user experience is very good. In Case 3 (a company meeting), $\Pr[t_d \geq \tau_0]$ is about 0.1, which cannot be ignored. Fortunately, $\Pr[t_d \geq \tau_1]$ approaches 0. That is, the user experience for 90% of the delays is very good, and 10% of them are acceptable. To reduce $\Pr[t_d \geq \tau_0]$ in Case 3, local VoiceTalk/VT-ASR deployment is essential. We further analyze the timing parameters as follows. We first note that in Eqs. (28) and (29), a small $n$ means a large variance of the VT-ASR delay $t_d$. A small $\alpha_2$ means a large variance (unstable)
of the pause period $\tau_0$, and a large $\eta_2$ means a long $\tau_0$ period. Figure 15 illustrates the impact of $t_d$'s variance on $\Pr[t_d \geq \tau_0]$. Since the variance of $t_d$ increases as $n$ decreases, the figure shows that $\Pr[t_d \geq \tau_0]$ exponentially drops as the variance of $t_d$ decreases. The largest variance occurs when $n = 1$. In all cases of our measurements, $n \geq 3$, and the resulting $\Pr[t_d \geq \tau_0]$ values are sufficiently small. Figure 15 also indicates that $\Pr[t_d \geq \tau_0]$ increases as $\eta_2$ and/or $\alpha_2$ decrease (when long and/or unstable $\tau_0$ are observed).

By considering the worst case $n = 1$, Figure 16 plots $\Pr[t_d \geq \tau_k]$. The figure indicates that the effects of $\alpha_2$, and $\eta_2$ in Figure 16 are the same as that in Figure 15. Figure 16 shows that $\Pr[t_d \geq \tau_k]$ increases as $\eta_1$ and/or $\alpha_1$ decrease (when long and/or unstable $\tau_k$ are observed). Also $\Pr[t_d \geq \tau_k]$ drops exponentially as $k$ increases. To conclude, for all cases investigated in this paper, VoiceTalk shows very good user experience in terms of the VT-ASR delay performance.
6. CONCLUSIONS

This paper proposed the VoiceTalk approach to develop voice-based IoMT applications as well as to upgrade existing IoT applications to voice-based IoMT. For ASR of mixing Mandarin, Min-Nan, and English, we have deployed the largest Taiwanese spoken corpus and use the corpus to train the novel automatic speech recognition solution VT-ASR based on a hybrid hidden Markov model/deep neural network and an n-gram language model. In Formosa Speech Recognition Challenge 2020 (FSR-2020), the Taiwanese Min-Nan recognition accuracy of VT-ASR is better than all participating ASR systems. For Taiwanese Mandarin in FSR-2018, the accuracy of VT-ASR is 90.5%. In the same challenge, the accuracies of Google and Iflytek are 79.4% and 81.8%, respectively [59]. We also developed an analytic model to investigate the probability that the voice-to-text translation/transmission for the first speaker is complete before the second speaker starts talking. From the measurements and the analytic modeling, we showed that the VT-ASR delay is short enough to result in very good user experience.

The VT-ASR mechanism is implemented as an IoT device in VoiceTalk. With such implementation, existing IoT-based applications can be almost transparently upgraded to voice-based IoMT. The demonstrated examples include interactive swing light poles [52], interactive hollow light globe [53], PlantTalk [54][55], ToiletTalk [56] and ZenboTalk [57]. We have made three important contributions described below:

- We proposed the VoiceTalk approach, which can graphically, systematically, and transparently integrate the ASR mechanism into existing IoT applications. The idea is to implement the ASR mechanism as an IoT device and manage it like other IoT devices in the VoiceTalk GUI. Without VoiceTalk, a non-voice IoT application requires tedious programming effort to add the ASR feature, and our experience indicates that the resulting application through hard coding is error-prone.
- We built the largest Taiwanese spoken corpus and proposed the novel VT-ASR. As compared with existing ASR solutions, VT-ASR has the best speech recognition accuracies for mixing Mandarin, Min-Nan, and English.
- We developed a novel analytic model to conduct time complexity analyses. Together with the measurements of the VT-ASR delay locally and remotely (in the cloud), our study shows good user experience of VoiceTalk; that is, the translated captions seldom overlapped incorrectly with spoken phrases. Such analysis has not been found in the literature.

VT-ASR has won several important government contracts in Taiwan. Our solution provided live captioning for the 2020 Taiwan presidential debates, engaging an audience of millions, and has also been chosen to provide live captioning for the Legislature of Taiwan. VT-ASR has also been used by the Ministry of Health and Welfare for various services, including call centers for long-term care customer service of 21 counties in Taiwan and CDC TV broadcasting for COVID-19. For the acoustic tests of PAL-Labs, VT-ASR’s accuracy is 96.47%, while the Google’s accuracy is 94.28%. In the commercial sector, our solution has been used by Public Television and Eastern Television with recognition accuracy higher than 90%, while the accuracies for other commercial solutions were between 60% and 70%. In the future, we will enhance VT-ASR for other Chinese dialects including Cantonese, Hakka, and more.

A APPENDICES

A.1 VoiceTalk Examples

As we described in Section 2, the evolution of wireless remote control has five stages. This section uses examples to describe how VoiceTalk upgrades IoT applications to their voice-based IoMT counterparts in these evolution stages.
For Stage 2 of the remote-control evolution, Figure 17 shows an example to use a smartphone as the remote control for an art work called swing light balls [52]. In this example, the light balls supported by thin poles (Figure 17 (1)) swing with the wind. This swing light ball application is configured as the IoTtalk3 project in the GUI (Figure 18 (a)). The bending degree of a light ball is measured by an acceleration sensor ((1) in Figure 18 (a)), and the measured degree is sent to the light ball actuator ((2) in Figure 18 (a)) through the control path (1)→(2). The SA of the light ball actuator determines the light color based on the inputs received from its DA. Specifically, a light ball may change its color depending on the degree it bends. The swing light balls can also be controlled by the soft keyboards (Figure 17 (2)) and soft switches (Figure 17 (3)) of a smartphone through the control path (3)→(2) in Figure 18 (a). By scanning a QR code, the web-based control keyboard for the swing light ball is shown in the browser of a smartphone [51].

We note that in the IoTtalk3 project, all actuators are controlled by the sensors and the smartphone keyboards without voice control. The swing light ball application can be conveniently upgraded to voice-controlled IoMT to support Stage 3 remote control. The resulting configuration is VoiceTalk3 in Figure 18 (b). In VoiceTalk3, we add the VT-ASR device ((5) and (6) in Figure 18 (b)) and the voice input Voice-I in the Smartphone device ((4) in Figure 18 (b)). When one says...
“blue”, the color of the lights turns blue through the control path \((4)\rightarrow(5)\rightarrow(6)\rightarrow(2)\) in Figure 18 (b). The transition \((5)\rightarrow(6)\) occurs in the SA of VT-ASR (Figure 5 (9)) where the voice is translated to a text control instruction to drive the light balls.

Figure 19: The hollow light globe interactive application: (a) Night control in 2019 and (b) Day view in 2021.

Another interesting interactive art work is the hollow light globes in a lakeshore (Figure 19). In the IoT application version, one uses an arbitrary smartphone to turn on the lights of the hollow globes through Keyboard-I in IoTalk3; see \((3)\rightarrow(8)\) in Figure 18 (a). Through the path \((1)\rightarrow(8)\) in Figure 18 (a), the globes can also be turned on and off when the temperature changes. Similar to the swing light ball application, the voice-based IoMT project VoiceTalk3 allows voice interaction between the microphone of a smartphone (Figure 19 (1)) and the hollow light globes (Figure 19 (2)). In the demo video [53], when a girl says “Prof. Lin I love you” through her smartphone, the hollow light globes turn on through the path \((4)\rightarrow(5)\rightarrow(6)\rightarrow(8)\) in Figure 18 (b). The louder the voice, the more the hollow globes light on. When she says “I don’t love you anymore”, the light globes turn off. The lights of the hollow globes can also be controlled by the soft switches of the smartphone as illustrated in Figure 19 (b).

Figure 20: (a) A plant box controlled by Google Home; (b) A smart toilet application.
Another example of Stage-2 remote control is a plant box application (Figure 20 (a)) configured in the IoTtalk4 project (Figure 21 (a)) [54]. In IoTtalk4, a plant box is represented by two icons in the GUI. The measured data of the sensors for pH, temperature, humidity, O₂, water level, and CO₂ (i.e., Sensor-I; see (1) in Figure 21 (a)) are sent to the output part of the PlantBox IoT device (i.e., Sensor-O; see (2) in Figure 21 (a)), and the actuator controls of the plant box (for reverse osmosis, fan, sprinkler, drain pump, suction pump, nutrition dripper, white LED, red LED, and blue LED) are grouped at the input part of the PlantBox IoT device (i.e., Control-I; see (7) in Figure 21 (a)). The actuators ((8) in Figure 21 (a)) are automatically controlled by the sensors through the path (1)→(2)→(7)→(8), which exercises Stage 4 remote control.

![Diagram of IoTtalk4 and VoiceTalk4](image.png)

Figure 21: Upgrading the plant box application from IoT to IoMT: (a) IoTtalk4 (sensor control) and (b) VoiceTalk4 (sensor control and voice control)

VoiceTalk can easily upgrade the IoTtalk4 applications to voice-based IoMT (VoiceTalk4; see Figure 21 (b)) using a commercial smart speaker. In the demo video [55], the plant box is controlled by Google Home. In the VoiceTalk4 project, we add Text-I and Control-O to the PlantBox device; see (5) and (4) in Figure 21 (b). The Google Home device is bound to the SmartSpeaker IoT device; see (3) and (6) in Figure 21 (b). We have implemented the Google Home SA by using the Google assistant SDK. Google Home translates the voice received by the microphone into the text, and the SmartSpeaker SA sends the text strings to the VoiceTalk server through Text-I ((3) in Figure 21 (b)). On the other hand, when the SmartSpeaker SA receives a text string, Google Home translates it into the voice (Voice-O; (6) in Figure 21 (b)). Then the SmartSpeaker SA plays the voice through the speaker. In the demo video [55], after the opening remark initiated by Google Home, the house owner asks for the humidity value of the plant box. The PlantBox device obtains the humidity data from the path (1)→(2) in Figure 21 (b). The voice received by the microphone of Google Home is translated into a text instruction and is forwarded to PlantBox through the path (3)→(4). The SA of PlantBox retrieves the concentration of humidity (which is 54) and sends it to Google Home through the path (5)→(6). Google Home translates the text string into voice and plays it through the speaker. When the home owner asks to sprinkle water, the instruction is sent to PlantBox through (3)→(4) again. The SA of PlantBox sends the control message to the sprinkler through the path (7)→(8), and asks Google Home to say “Sure. Start sprinkling water” through (5)→(6). The subsequent interactions in the demo video are similar and are omitted.

Many IoT applications can be upgraded with voice capabilities like the smart plant box. For example, in a smart toilet application (Figure 20 (b)), the toilet is typically controlled by a digital panel mounted on the wall next to the toilet. Such
application can be implemented in IoTtalk3 where the digital panel is bound to the Keyboard-I icon ((3) in Figure 18 (a)), and the toilet is bound to the Actuator device ((7) in Figure 18 (a)). Voice-controlled smart toilet application can be achieved by upgrading the IoTtalk3 project to VoiceTalk4 in Figure 21 (b), where the PlantBox device is replaced by the Toilet device. As illustrated in the demo video [56], one gives verbal instructions (open, close the toilet lid and flush the toilet) through the microphone of a smart speaker. The voice is translated into a text-based instruction ((3) in Figure 21 (b)) to control the toilet through the path (3) → (4) → (7) → (8).

Figure 22: Replacing the smart speaker in VoiceTalk: (a) VoiceTalk5 (smartphone as smart speaker) and (b) VoiceTalk6 (Zenbo control)

As mentioned in Section 2, a Stage 4 remote-control system “smartly” learns how to change the statuses of the appliances ((d) and (e) in Figure 3) based on the conditions of environment ((c) in Figure 3) [49]. Such application can be effortlessly implemented in VoiceTalk4. When the CO2 sensor ((1) in Figure 21 (b)) detects that air quality is poor in the room, the system automatically turns on the purple lights of the plant boxes, which activates the photosynthesis process to improve the air quality ((8) in Figure 21 (b)). The control path is (1) → (2) → (7) → (8). When the CO2 reading is high, the system also opens the window (Figure 3 (d)) to bring fresh air into the room. The control path is (1) → (8) in Figure 21 (b).

We can use a smartphone to replace specific hardware (e.g. Google Home) to serve as a smart speaker by modifying the VoiceTalk4 project to VoiceTalk5 in Figure 22 (a). Specifically, we replace the SmartSpeaker device by the Smartphone and the VT-ASR devices. VoiceTalk5 is basically the same as VoiceTalk3 except that VT-ASR in VoiceTalk5 also performs text-to-voice translation.

As for Stage-5 remote control, we use Zenbo (Figure 4) as an example to illustrate how VoiceTalk can straightforwardly implement interactive remote control through dialogs. The VoiceTalk GUI configuration for Zenbo is illustrated in Figure 22 (b), where the voice received by Zenbo’s microphone is sent out by its SA through Voice-I ((3) in Figure 22 (b)). Zenbo’s SA can also send out control signals and texts through Text-I and Control-I; see (5) and (7) in Figure 22 (b), respectively. Zenbo’s SA receives the sensor measurements, the voice and the control signals through Sensor-O, Control-O and Voice-O; see (2), (4) and (6) in Figure 22 (b), respectively. In [57], when the room is dark, the light sensor triggers
Zenbo to ask if the home owner wants to turn on the light. The control path is (1) → (2) → (5) → (12) → (9) → (6). When the home owner answers through the microphone of Zenbo, the control path is (3) → (11) → (10) → (4). If the home owner’s answer is “yes”, the light is turned on through the control path (7) → (8), and Zenbo replies to the home owner that the light is turned on through the control path (5) → (12) → (9) → (6). When the room has enough sun light, Zenbo asks if the home owner wants to turn off the light. If not, Zenbo does nothing. Similarly, when the room is hot, Zenbo asks to turn on the fan following the same procedure as the light control.

A.2 Analytic Modeling

This appendix derives the probabilities \( \Pr[t_d \geq \tau_i] \) and \( \Pr[t_d \geq \tau_k] \). Let \( t_d \) be an Erlang random variable with the shape parameter \( n \) and the scale parameter \( \lambda \). Then its PDF and the CDF are

\[
f_d(t_d) = \frac{\lambda^n t_d^{n-1} e^{-\lambda t_d}}{(n-1)!} \quad \text{and} \quad F_d(t_d) = 1 - \sum_{j=0}^{n-1} \frac{\lambda^j t_d^j e^{-\lambda t_d}}{j!}
\]  

(5)

Substitute Eq. (5) into Eq. (4) to yield

\[
\Pr[t_d \geq \tau_i] = \int_{\tau_i=0}^{\infty} h_i(\tau_i) \int_{t_d=\tau_i}^{\infty} f_d(t_d) \, dt_d \, d\tau_i \\
= \sum_{j=0}^{n-1} \left( \frac{\lambda^j}{j!} \right) \int_{\tau_i=0}^{\infty} \tau_i^j h_i(\tau_i) e^{-\lambda \tau_i} \, d\tau_i \\
\]  

(6)

From frequency domain general derivative of Laplace transform, Eq. (6) is re-written as

\[
\Pr[t_d \geq \tau_i] = \sum_{j=0}^{n-1} \left( \frac{\lambda^j}{j!} \right) \left[ \frac{h_i^{(j)}(s)}{ds^j} \right]_{s=\lambda} \\
= \sum_{j=0}^{n-1} \left( \frac{-\lambda^j}{j!} \right) \left[ \left\{ g_2^{(0)}(s) [g_1^{(0)}(s) g_2^{(0)}(s)]^{(j)} \right\} \right]_{s=\lambda} \\
\]  

(7)

For an arbitrary Laplace transform function \( f^*(s) \), denote its \( j \)-th derivative as

\[
f^{(j)}(s) = \frac{f^{(j)}(s)}{ds^j}
\]

Then from Eq. (7), we have

\[
h_i^{(j)}(s) = \left\{ g_2^{(0)}(s) [g_1^{(0)}(s) g_2^{(0)}(s)]^{(j)} \right\}
\]

(8)

In Eq. (8), let \( G_i^{(0)} = \left[ g_1^{(0)} g_2^{(0)} \right]^i \). Then for \( i = 1 \),

\[
G_1^{(j)} = \sum_{k=0}^{j} \binom{j}{k} g_1^{(k)} g_2^{(j-k)}
\]

(9)
For $i > 1$, 

$$G_i^{(1)} = i \left[ g_1^{(0)} g_2^{(0)} \right]^{i-1} \left[ g_1^{(0)} g_2^{(0)} \right]^{(1)} = i G_{i-1}^{(0)} G_i^{(1)} \quad (10)$$

From Eq. (10), 

$$G_i^{(k)} = \left[ G_i^{(1)} \right]^{(k-1)} = i \left[ G_{i-1}^{(0)} G_i^{(1)} \right]^{(k-1)} \quad (11)$$

Applying general Leibniz rule to Eq. (11), for $i > 1$ we have 

$$G_i^{(j)} = i \left\{ \sum_{k=0}^{j-1} \binom{j-1}{k} G_{i-1}^{(k)} \left[ \sum_{m=0}^{j-k} \binom{j-k}{m} g_1^{(m)} g_2^{(j-k-m)} \right] \right\} \quad (12)$$

We can recursively apply general Leibniz rule to Eq. (12) to yield 

$$G_i^{(j)} = i \left\{ \sum_{k=0}^{j-1} \binom{j-1}{k} G_{i-1}^{(k)} \left[ \sum_{m=0}^{j-k} \binom{j-k}{m} g_1^{(m)} g_2^{(j-k-m)} \right] \right\} \quad (13)$$

Substitute Eq. (9) into Eq. (7) to yield 

$$h_i^{(j)} = \left\{ g_2^{(0)} G_i^{(0)} \right\}^{(j)} \quad (14)$$

Applying general Leibniz rule to Eq. (14) and using Eq. (13) [74], we have 

$$h_i^{(j)} = \sum_{l=0}^{j} \binom{j}{l} g_2^{(l)} G_i^{(j-l)}$$

$$= \sum_{l=0}^{j} \binom{j}{l} \left[ \sum_{k=0}^{j-l-1} \binom{j-l-1}{k} G_{i-1}^{(k)} \left[ \sum_{m=0}^{j-l-k} \binom{j-l-k}{m} g_1^{(m)} g_2^{(j-l-k-m)} \right] \right] \quad (15)$$

Substitute Eq. (15) into Eq. (7) to yield 

$$\Pr[t_d \geq t_i] = \sum_{j=0}^{n-1} \left( \frac{- \lambda}{j!} \right)^j \left[ \frac{h_i^{(j)}(s)}{ds^j} \right]_{s=\lambda}$$

$$= \sum_{j=0}^{n-1} \left( \frac{- \lambda}{j!} \right)^j \left[ g_2^{(j)}(s) \right]_{s=\lambda} \prod_{i=1}^{k} \left[ g_1^{(i)}(s) \right] \left[ g_2^{(i)}(s) \right]_{s=\lambda} \quad (16)$$

Eq. (16) can also be derived from Eq. (6) as follows: 

$$\Pr[t_d \geq t_k] = \int_{t_2=0}^{\infty} g_2(t_2) \prod_{i=1}^{k} \int_{t_i=0}^{\infty} g_1(t_i) \int_{t_{i+2}=0}^{\infty} g_2(t_{i+2})$$

\[27\]
× \sum_{j=0}^{n-1} \left( \frac{\lambda}{j!} \right)^j \sum_{l=0}^{k} \left( t_{l,1} + t_{l,2} \right)^j e^{-\lambda \left[ t_{l,1} + \sum_{i=1}^{k} \left( t_{i,1} + t_{i,2} \right) \right]} dt_{l,0} \prod_{i=1}^{k} dt_{l,1} dt_{l,2}

= \sum_{j=0}^{n-1} \left( \frac{\lambda}{j!} \right)^j \int_{t_{l,0}}^{\infty} g_2(t_{l,2}) e^{-\lambda t_{l,2}} \prod_{i=1}^{k} \left[ \int_{t_{l,1}}^{\infty} g_1(t_{l,1}) e^{-\lambda t_{l,1}} \int_{t_{l,2}}^{\infty} g_2(t_{l,2}) e^{-\lambda t_{l,2}} dt_{l,2} \right] \times \left[ t_{l,0} + \sum_{i=1}^{k} (t_{i,1} + t_{i,2}) \right]^j dt_{l,0} \prod_{i=1}^{k} (dt_{l,1} dt_{l,2}) \quad (17)

By generalizing the Binomial theorem, we have
\[ \left[ t_{l,0} + \sum_{i=1}^{k} (t_{i,1} + t_{i,2}) \right]^j = \sum_{j_{0,2}+\sum_{i=1}^{k} j_{i,1}+j_{i,2}=j} \sum_{l_{0,2}} \left( \begin{array}{c} j_{0,2} \\ j_{i,0,2} \end{array} \right) \prod_{i=1}^{k} (t_{i,1}+t_{i,2}) \quad (18) \]

From Eq. (18), Eq. (17) is rewritten as
\[
\Pr[t_d \geq \tau_k] = \sum_{j=0}^{n-1} \left( \frac{\lambda}{j!} \right)^j \int_{t_{l,0}}^{\infty} g_2(t_{l,2}) e^{-\lambda t_{l,2}} \prod_{i=1}^{k} \left[ \int_{t_{l,1}}^{\infty} g_1(t_{l,1}) e^{-\lambda t_{l,1}} \int_{t_{l,2}}^{\infty} g_2(t_{l,2}) e^{-\lambda t_{l,2}} dt_{l,2} \right] \times \left[ t_{l,0} + \sum_{i=1}^{k} (t_{i,1} + t_{i,2}) \right]^j dt_{l,0} \prod_{i=1}^{k} (dt_{l,1} dt_{l,2})
\]
\[
= \sum_{j=0}^{n-1} \left( \frac{\lambda}{j!} \right)^j \sum_{j_{0,2}+\sum_{i=1}^{k} j_{i,1}+j_{i,2}=j} \left[ \left. \frac{g_2(s)}{ds} \right|_{s=\lambda} \right]^{j_{0,2}} \prod_{i=1}^{k} \left[ \left. \frac{g_2(s)}{ds} \right|_{s=\lambda} \right]^{j_{i,1}} \quad (19)
\]

For \( n = 1 \), Eq. (19) is expressed as
\[
\Pr[t_d \geq \tau_k] = g_2^\lambda(\lambda) [g_2^\lambda(\lambda)]^k \quad (20)
\]

For \( k = 0 \), Eq. (19) is expressed as
\[
\Pr[t_d \geq \tau_0] = \sum_{j=0}^{n-1} \left( \frac{\lambda}{j!} \right)^j \left. \frac{g_2^j(s)}{ds} \right|_{s=\lambda} \quad (21)
\]

If \( g_1(t_{i,1}) \) and \( g_2(t_{i,2}) \) are Gamma PDFs with the shape parameters \( \alpha_1 \) and \( \alpha_2 \) and the scale parameters \( \beta_1 \) and \( \beta_2 \), respectively, then the \( j \)-th derivatives of their Laplace transforms are
\[
g_1^{(j)} = \frac{(-1)^j \Gamma(\alpha_1 + j) \beta_1^{\alpha_1}}{\Gamma(\alpha_1)} \text{ and } g_2^{(j)} = \frac{(-1)^j \Gamma(\alpha_2 + j) \beta_2^{\alpha_2}}{\Gamma(\alpha_2)} \quad (22)
\]

For \( n = 1 \), we substitute Eq. (22) into Eq. (20) to yield
\[
\Pr[t_d \geq \tau_k] = \frac{\beta_1^k \alpha_1 \beta_2^{(k+1)\alpha_2}}{(\lambda + \beta_1)^{k\alpha_1}(\lambda + \beta_2)^{(k+1)\alpha_2}} \quad (23)
\]

For \(k = 0\), we substitute Eq. (22) into Eq. (21) to yield

\[
\Pr[t_d \geq \tau_0] = \sum_{j=0}^{n-1} \left( \frac{-\lambda}{j!} \right)^j \left[ \frac{(-1)^j \Gamma(\alpha_2 + j)\beta_2^{\alpha_2}}{\Gamma(\alpha_2)(s + \beta_2)^{\alpha_2+j}} \right] = \sum_{j=0}^{n-1} \left( \alpha_2 + j - 1 \right) \left[ \frac{\lambda^j \beta_2^{\alpha_2}}{(\lambda + \beta_2)^{\alpha_2+j}} \right] \quad (24)
\]

If \(\alpha_1 = \alpha_2 = 1\), then Eq. (22) is simplified as

\[
g_1^{(j)} = \frac{(-1)^j(j!)\beta_1}{(s + \beta_1)^{j+1}} \quad \text{and} \quad g_2^{(j)} = \frac{(-1)^j(j!)\beta_2}{(s + \beta_2)^{j+1}} \quad (25)
\]

Therefore, for \(\alpha_1 = \alpha_2 = 1\), we substitute Eq. (22) into Eq. (19) to yield

\[
\Pr[t_d \geq \tau_k] = \sum_{j=0}^{n-1} \left( \frac{-\lambda}{j!} \right)^j \sum_{i_1 \in \nu_{01},j_1 \in \nu_{11},j_2 = j} \left[ \frac{(-1)^{j_2}(j_2!)\beta_2}{(s + \beta_2)^{j_2+1}} \right] \left[ \frac{(-1)^{j_1}(j_1!)\beta_1}{(s + \beta_1)^{j_1+1}} \right] \left\{ \frac{(j_1,j_2)!}{(\lambda + \beta_2)^{j_2+1}} \right\} \left\{ \frac{(j_1,j_2)!}{(\lambda + \beta_2)^{j_1+1}} \right\}
\]

Eqs. (23) and (24) merit further discussion. For Erlang \(t_d\) and Gamma \(t_{d,1}\) and \(t_{d,2}\), we have \(E[t_d] = \frac{n}{\lambda}, E[t_{d,1}] = \frac{\alpha_1}{\beta_1}, E[t_{d,2}] = \frac{\alpha_2}{\beta_2}\), and \(V[t_d] = \frac{\alpha_1^2}{\beta_1^2}\) and \(V[t_{d,1}] = \frac{\alpha_1^2}{\beta_1^2}\) and \(V[t_{d,2}] = \frac{\alpha_2^2}{\beta_2^2}\) or

\[
\lambda = \frac{n}{E[t_d]}, V[t_d] = \left( \frac{E[t_d]}{n} \right)^2, \text{and} \quad V[t_d] = \frac{\eta^2(E[t_d])^2}{n} \quad (26)
\]

Let \(E[t_{d,1}] = \eta_1 E[t_d]\) and \(E[t_{d,2}] = \eta_2 E[t_d]\), then

\[
\beta_1 = \frac{\alpha_1}{\eta_1 E[t_d]} \quad \text{and} \quad \beta_2 = \frac{\alpha_2}{\eta_2 E[t_d]} \quad (27)
\]

Substitute Eq. (27) into Eq. (23) to yield

\[
\Pr[t_d \geq \tau_k] = \frac{\alpha_1^{k\alpha_1} \alpha_2^{(k+1)\alpha_2}}{(\eta_1 + \alpha_1)^{k\alpha_1}(\eta_2 + \alpha_2)^{(k+1)\alpha_2}} \quad (28)
\]

Similarly, substitute Eq. (27) into Eq. (24) to yield

\[
\Pr[t_d \geq \tau_0] = \left( \frac{\alpha_2}{n\eta_2 + \alpha_2} \right)^{\alpha_2} \sum_{j=0}^{n-1} \left( \frac{\alpha_2 + j - 1}{\frac{n\eta_2 + \alpha_2}{n\eta_2 + \alpha_2}} \right)^j \quad (29)
\]
REFERENCES


