Network Effects on the Functionality of Voice-Controlled Systems and Applications

Yasser Gadallah and Ikbal Taleb
College of Information Technology, UAE University
P.O. Box 17551, Al Ain, UAE.
{ygadallah,italeb}@uaeu.ac.ae

Abstract

The use of voice to control devices and applications provides a flexible, convenient and secure method for operating these systems. Users can control these systems using voice commands while they are in the proximity of the system to be controlled. Attempting to apply such control over the internet may be affected by several network impairments, e.g. jitter, errors, losses, etc. The effects of these impairments, while may not be felt by the normal human ear, can impact the operation of the software that processes these commands. This processing system is usually based on speaker identification and speech recognition components. In this paper, we study these network effects on the correct operation of such systems. We experiment with different types of network impairments and measure the effects on the voice-controlled operation. Our experiments show that network impairments affect the processing of voice commands to varying degrees. They also provide a comparative view between the behavior of command processing modules and the perceived quality of voice.

1. Introduction

Using voice to control systems and applications is proliferating at an increasing pace due to the flexibility and security that it provides to its users. It is much more convenient for many users to “speak” for some action to be taken than having to learn to use control buttons, a computer mouse, a joystick or some combination of complex configurations. When secure access is required, it is more secure to rely on voice stamps to verify if a user is authorized to perform a certain action or operation than it is to use the more traditional authentication methods. It is a lot harder to falsify a voice signal than cracking a regular text-based password. Voice-controlled applications span many aspects of life. This includes household appliances [1], door control [4], medical applications [3] and even controlling robots [2].

There are many applications for controlling household systems over the internet. For example, a user may need to operate air-conditioning or heating systems while they are away from home. They may need to turn lights on/off while they are traveling as means of deterring intruders. There are also many emerging medical applications where control needs to be applied over the internet [5]. Voice control may therefore prove quite attractive as a robust and secure form of system control. This is in addition to the ease of use for end users who may not be technology-savvy.

Systems that can be used for this purpose are usually based on the speaker identification and voice recognition technologies. These technologies are usually sensitive to the variations in the signal patterns to varying degrees. Therefore, while a certain spoken sentence may be possible to recognize by the human ear, it may not be recognizable by the voice processing software. This means that an acceptable perceived voice quality may not be sufficient to guarantee successful voice commands processing by the software. Issuing voice control commands over a network exposes the transmitted voice to a variety of network impairments. These impairments may not always affect the perceived voice quality in a significant way. However, they may render the commands unrecognizable by the voice processing software.

There have been several studies that dealt with network effects on the perceived quality of voice, e.g. [9]. On the other hand, [9] discussed some aspects of the network effects on voice recognition of the transmitted VoIP traffic.

In this study, we perform several experiments to examine the effects of the main network impairments that may affect voice transmission on the performance of voice-controlled applications. We measure the
ability of the voice commands processing software to correctly react to voice commands that are issued to the system remotely over an IP network. To achieve this, we subject the voice streams of these commands to a variety of network impairments. We then measure the resulting effects. We use simple commands that are used for purposes such as turning on/off appliances or opening/closing doors. Our aim is to determine the differences between the perceived voice quality and the actual quality as sensed by these processing software modules. We use an experimental setup that mimics realistic scenarios that are similar to what would be encountered with real remotely voice-controlled applications.

The rest of this paper is organized as follows. In Section 2, we describe the structure of the voice-controlled system under study. In Section 3, we discuss the main network impairments that can potentially affect the functionality of a remotely voice-controlled system. In Section 4, we present the experimental setup as well as the results of the different experiments that we performed. In Section 5, we conclude the study.

2. System structure
We consider a generic voice-controlled system in which a user issues voice commands over an IP-based network in order to invoke an action in a physically remote place. For example, a traveler in Paris can issue a voice command over the internet to turn the light on/off occasionally in his house in New York as a means to deter intruders while he is away from his home. A microphone that is connected to a networked computer system is used to issue a voice command. This command is sent over the network to a designated destination where another computer system processes the voice command. If the command is processed successfully, the required action is taken, otherwise the command is rejected. Figure 1 gives an illustration of the system structure.

For the purpose of this study, we only focus on the system component that processes the commands in order to take the appropriate action. This component consists of two main processing stages. The first stage is the speaker identification unit which is responsible for user authentication. It is based on comparing the voice command with speech patterns of users who are authorized to control the system. The second stage is invoked only upon the successful completion of the first stage, i.e. when the authentication is successful. In this stage, the command is interpreted in order to determine the required action. Voice recognition techniques are used to determine the action that is specified by the incoming voice command. Once identified successfully, the action is executed. If this stage of processing is not able to identify the action, the system issues a failure message that is sent back over the network to the user who issued the command.

Figure 1. Structure of the system under study

The system works correctly, according to the above description, as long as there are no external effects that influence the operation of one or both of the command processing stages. However, since the commands are sent over the network, it is expected that network impairments will affect the voice packets that constitute the different commands. This, in turn, may affect the ability of the voice processing modules to interpret these commands correctly and hence its ability to invoke the desired action. While the human ear may be able to tolerate some of the network impairment types that we discuss in the next section, the voice processing modules may not be as tolerant, and this is what we explore in this study.

3. Voice-affecting network impairments
There are many impairment types that can potentially affect voice transmission over the network. These impairments range from packet delays to actual errors that affect the data content within these packets. Using the TCP protocol usually takes care of correcting some of these errors. However, since voice and video traffic has to be transmitted in real-time, these corrective measures cannot be applied in such cases. Therefore, the transmission of voice traffic is normally done using the UDP protocol which does not
have such corrective measures. The drawback of this is that voice traffic will be affected by any network impairments that it encounters with no corrective measures being applied. The resulting voice, in many cases, can still be comprehended by the human ear. The same may not be true for the software modules that we use to process the voice commands in the systems that we consider in this study.

There are many issues that can affect data transmission over IP networks [6]. Some packets may get lost, distorted, delayed, etc. In this section we discuss few of these types.

3.1 Jitter
It is the variation in the inter-arrival times of the data packets. This results from the different packets of a voice steam taking different routes over the network. Jitter inter-arrival times can be represented by different probability distributions, with the closest distribution to the actual behavior encountered over the internet being the exponential distribution.

3.2 Packet losses
This network impairment results when the network or parts of the network are overloaded. When this occurs, some packets get severely delayed, or may even be dropped as some of the router queues of the affected parts of the network overflow. For voice traffic, the lost packets will not get retransmitted as we need to transmit in real-time as we explained earlier, which may affect the perceived voice quality. Our goal is to measure the effect on voice processing software and compare it to the effect on perceived voice quality.

3.3 Packet errors
There are several sources of packet errors. In case of voice traffic, one of the reasons could be the manipulation of this kind of traffic in order to use the available bandwidth efficiently. This is caused by applying coding/decoding techniques which at the end may result in packet distortion to varying degrees.

3.4 Packet doubling
This impairment usually results from wrong configurations which may result in sending the same packet more than once. Voice receivers normally ignore repeated packets which helps preserve perceived voice quality in many cases.

4. Experimental setup and results
We use a test bed setup that consists of two Cisco Catalyst 3560G switches and three personal computers (PCs). One of the PCs is used to initiate and transmit voice commands. The second PC is used to receive and process these voice commands. This PC hosts the voice processing software which is responsible for authenticating the sender and recognizing the commands. The third PC is used to emulate the network. It hosts commercial network impairment emulation software. It is equipped with two Ethernet cards. The setup is illustrated in Figure 2. The commands flow from the command sender PC through the emulator PC, where the network impairment under experimentation gets injected into the voice stream. Then the voice stream gets transmitted to command processing PC where it gets processed. If processing is successful, a screen is shown to indicate a successful operation and a success message is sent back to the sender. If it is not, a failure message is sent back to the sender, and no action is taken. The failure could be either due to an authentication failure, or a command recognition failure. We use simple voice commands for our experiments such as “open the door”. The software that we use for speaker identification is based on [7]. The software that is used for voice command recognition is based on [8]. We run each experiment 10 times and the results that we report are the averages of the 10 experimental runs. When processing failures are encountered, we include a comparison with the perceived quality of the voice which depends on listening to the commands and determining if they can be recognized with sufficient clarity through hearing.

4.1 Behavior with no network impairments
We issue voice commands over our network setup without network impairments to verify the system functionality. For all experiment runs, all stages of the processing were completed with no failures.

4.2 Effect of network jitter
We apply an exponentially distributed, with parameter $\lambda$, jitter to the voice stream. We vary the parameter $\lambda$, and measure the resulting effect on the functionality of the system. We use the values that will result in jitter values ranging, in a random fashion, between a minimum of 0 ms and a maximum of 26, 52, 104, 208 and 1040 ms, respectively. The results are shown in Figure 3. We see from the figure that the system functions properly in all trials regardless of the value of the jitter. This is due to the fact that voice receivers utilize jitter buffers in order to alleviate the perceived jitter effect on the human ear. This seems to have had a positive effect on the speech processing modules, where they were able to process the commands successfully in all cases.
4.3 Effect of packet losses

We apply packet losses in such a way that a certain percentage of the voice stream is randomly lost. We experiment with losses that range from 10-60% of the stream and measure the effect on the processing of the voice commands. Figure 4 shows that the success rate while processing the command falls from about 80% when the percentage of lost packets is at 10%, to 0% when the losses reach 50% of all stream packets. This occurs gradually and shows that the tolerance of the voice processing modules to packet losses decreases as the lost packet rate increases. When we consider the perceived voice quality which is measured through hearing, we find that command recognition is better than the case of software processing. Recognition that is based on perceived voice quality is 100% successful up to 30% packet loss, and it decreases gradually but never reached 0% success rate, in our experiments, as it is the case with software processing.

4.4 Effect of packet errors

We apply packet errors to a certain percentage of the voice command stream and then measure the effect in terms of the percentage success in processing the commands. We see from Figure 5 that the tolerance to packet errors is high initially when the percentage of the erroneous packets does not exceed 10%. In this case, the processing is 100% successful. However, when the percentage of packets in error increases, the success ratio drops rapidly and reaches 0% when the percentage of error reaches 40%. When we compare this behavior to that of recognition on the basis of perceived voice quality, we find that with perceived quality and as the error rate increases, the success rate is considerably higher and does not reach 0% as in the case of software processing. This means that for relatively small percentage of packet errors, the voice processing software remains capable of authenticating and recognizing the voice commands properly similar to the ability of the human ear. However, as the error ratio increases, this ability diminishes in a rapid pace despite the fact that the perceived quality of the voice enables the human ear to still recognize it.

4.5 Effect of packet doubling

In this experiment, a certain percentage of the voice stream is subjected to packet doubling. Packet doubling is done in such a way that the affected packets are repeated randomly between 1 and 3 times. Figure 6 shows the results. It is clear from this figure that this kind of impairments has no effect on the ability of the voice processing modules to handle the commands successfully as the receiver ignores duplicate packets.

4.6 Effect of packet doubling combined with network jitter

We also experimented with the case where packet doubling and network jitter occur simultaneously, to gauge the ability of the system to process the commands in this case. We used a value of $\lambda = 0.1$, and changed the packet doubling values as per the previous experiment. In this case, as shown in Figure 7, we see that this combination of network impairments has no effect on the processing of the voice commands where it all reaches 100% success for all cases.

It is worth mentioning that in all experiments the speaker identification module of the voice processing system was able to function correctly in all cases. All the processing failures were due to the failure of the voice recognition module to correctly recognize the
command. This is due to the fact that speaker identification, unlike command recognition, does not require a full voice command in order to be able to tie the pattern to a certain speaker.

5. Conclusions
The use of voice as means to control systems and applications has many merits such as the ease of use and the secure operation. Using this method over an IP-based network has some implications that stem from the fact that transmission over the network exposes the voice stream to numerous network impairments. In this study, we explored the effects of some of the main network impairments on the processing of such voice commands. Our experiments showed that command processing is particularly vulnerable to packet losses and packet errors. These types of impairments affect the ability of the control module to process the commands to completion successfully. When comparing this behavior to the perceived voice quality, our experiments showed that the perceived quality of voice is considerably higher than that sensed by the voice commands processing software. Moreover, we found that the speaker identification component of the software is not as sensitive to network impairments as it is the case with the command recognition software.

6. Acknowledgement
The authors would like to acknowledge the use of the prototype that was developed by A. Alhamoudi, M. Alshamsi and M. Mazrooei as a starting point in the test bed used to conduct the experiments for this study.

7. References