Uncovering Twilio: Insights into Cloud Communication Services

Ramnatthan Alagappan
ra@cs.wisc.edu

Sourav Das
souravd@cs.wisc.edu

1 Abstract

We study the protocols and architecture of a popular cloud communication service *Twilio* through gray box methods. We develop a simple VoIP service atop *Twilio* for our study. We provide insights into how different components in a cloud communication service interact with each other in various scenarios. We also measure some guarantees with respect to call and message dequeuing provided by Twilio. Our analysis reveals a number of interesting aspects about the Twilio ecosystem and have strong implications for developers who build applications on top of *Twilio* APIs.

2 Introduction

Cloud communication services (CCS) are an upcoming service model which provide sophisticated APIs for enterprises to develop applications, offload communication related tasks from enterprise applications. CCS host switching, telepony infrastruture and storage in the cloud. The services are offered through simple REST APIs for the applications to make use of them. Cloud communication services have a lot of advantages compared to *on-premise* communication infrastructure. Firstly, it takes the burden of communication from the applications and separates it into a separate service. The applications can seamlessly interact with the communication APIs to accomplish complex communication tasks like sending promotional messages to customers, providing critical real-time information to mobile phones etc. Secondly, the development effort involved in integrating an application with CCS is very less compared to developing and maintaining a home-brewed communication infrastructure. Thirdly, enterprises can build communication applications in a cost effective way becasue of the pay-per-use model provided by most the CC services.

Twilio is one of the prominent players in the CCS space.*Twilio* has a huge customer base including popular enterprises like *Coca-Cola*, *WalmartLabs*, *Intuit*, *Box* use *Twilio* to accomplish a wide variety of tasks ranging from two-step authentication, powering lending machines with music, secure file sharing, delivering deals and ads to mobile phones etc. *Twilio* enables lot of new scenarios for businesses that were rather cumbersome to implement in the past. The APIs are simple and intuitive to understand and develop. For our study, we developed a simple VoIP service atop Twilio APIs using a *Twilio* free-account. The entire development and testing took just two developer-weeks. Twilio also provides rich documentation and code examples to accomplish simple things like VoIP communication. We mostly tweaked the code provided in developer forums to get our VoIP service working.

3 Motivation and Related Work

Businesses want to delegate communication from their applications and services because of the advantages provided by CCS. The number of enterprises using CCS has seen a steady increase since its advent. Though there are no enough evidences that this pattern is going to continue, because cloud commucation services provide an attractive cost-model and easy-to-use APIs, we strongly believe that this trend is going to continue in the future. As more and more enterprises start using CCS like *Twilio*, there is a good chance that this traffic may contribute to a good fraction of internet traffic in the future.

There have been lot of studies on VOIP service like Skype in the past [1,2]. There have been recent studies on cloud storage service like *Dropbox* [3]. Best to our knowledge, we are the first to study cloud communication services. Similar studies have been carried out previously - [3] provides a thorough characterization of the Dropbox protocol and traffic patterns. They provide insights into how traffic to *Dropbox* varies across four different networks including home and campus networks. Our study on the other hand does not deal with traffic analysis since we did not have the sophistication of collecting the packet traces in the campus network. Our study aims at studying the architecture and protocols of the entire *Twilio* ecosystem. Our study also aims at finding some possible enhancements to the entire *Twilio* ecosystem. [1] provides a detailed packet level study of the *Skype* protocol. The authors also provide deep insights into the *Skype* architecture. Our study involves studying the architecture of the system and the protocols involved using a suite of gray box tools that we have developed.

The reminder of the paper is organized as follows. In secion 3, we provide a detailed analysis of the *Twilio* ecosystem, architecture of the system, high level protocols for some key scenarios and packet level analysis for two scenarios. We give some detailed measurements with respect to call dequeuing rates in section 4. Then, we present some oddities that we found in the entire ecosystem in section 5. In section 6, we discuss some of the pos-

sible enhancements to the system. Finally we conclude by presenting the implications of our study and our future work.

4 Twilio Ecosystem and Protocol Study

5 Measurements

6 Oddities

7 Discussion

8 Future Work and Conclusions

References