Methods for Measuring End-to-End Latency and Scaling SMS based Service with Constant E2E

Gopi Krishnan Nambiar

Intuit Inc, Bangalore, India

Summary
SMS is one of the most widely used data applications in the world. Due to its easy availability and low cost, many enterprises have chosen to use SMS as the primary medium for dissemination of information and alerts. During high loads and traffic scenarios, the SMS latency tends to rise exponentially, which is a concerning situation for these enterprises. Two strategies, namely geo affinity based routing and traffic based load balancing have been proposed in this paper for maintaining constant end-to-end latency values under high load scenarios. We have carried out experiments on a live and popular SMS based service in India 'txtWeb', measured the end-to-end latency using an Android app and have found the latency remained constant even under increasing loads after these methods were implemented in the txtWeb ecosystem.

Key words:
SMS; latency; scaling; latency measurement;

1. Introduction

Short Message Service is one of the most widely used data applications in the world, with 3.6 billion active users or 78% of all mobile phone subscribers. Though smart phones penetration is significant in many of the developed countries of the world, a considerable majority of the mobile devices in the developing regions are still low-cost devices with very minimalistic feature sets and processing power.

According to recent statistics [1], the total number of SMS sent globally in 2010 was 6.1 trillion, a 238.88% increase from the total number of SMS sent in 2007. Around 193000 text messages were sent every second in 2010, which is triple the amount of text messages sent in 2007. This clearly indicates the rapid growth of this mode of communication and its worldwide penetration in the last few years. Due to its accessibility, easy availability and support by low-end handsets and cellular carriers, SMS is a hugely popular means of communication in developing countries. Total mobile-cellular subscriptions reached almost 6 billion [2] by the end of 2011, corresponding to a global penetration of 86%. In fact, according to a recent report [3], the total number of mobile users in India alone is 929.37 million.

As more and more enterprises and users place high expectations on the instant delivery of SMS messages, it becomes imperative to solve the issues of high message delivery latency in general and more importantly during peak load scenarios. Due to its easy availability and low cost, SMS is being considered as a preferred means of communication for mission-critical applications such as emergency alerts [4], authentication mechanisms [5], business alerts and notifications for natural disasters [6] for which timely delivery of the message to the end user is the key.

It has been observed that during peak loads, many SMS based services suffer the issue of high message delivery latency. Another instance where end-to-end latency of such services get drastically impacted is during "flash crowd events" [7].

The data used in this study was collected from an SMS based service that serves more than 5 million cell phone users. The data was collected over an eight week period in 2012 and records over 83.2 million messages.

In this paper, we suggest 2 strategies which would help SMS based services or content providers who wish to maintain constant end-to-end latency during peak loads. We have used various methods such as load balancing user requests and responding to an end user based on his location in order to reduce the end-to-end latency sufferred by the end user.

The rest of the paper is organized as follows. Section 2 provides a brief overview of SMS operation, service classes, the definition of end-to-end latency and how exactly it is calculated and finally, the data set used in this study and its mode of collection using an Android Application. In Section 3, we discuss the two proposed strategies suggested for maintaining constant end-to-end latency, namely geo affinity based SMS routing and traffic based intelligent load balancing in detail. In Section 4, we discuss the related research in this area and compare it with this study. Finally, we conclude in Section 5.

2. Background and Data

2.1 SMS Operation Overview

Signaling System 7 (SS7) is a global standard that defines the procedures and protocols for exchanging information

Manuscript received March 5, 2013
Manuscript revised March 20, 2013

among network elements of wire line and wireless telephone carriers. The SS7 standard is used by network elements to exchange control information for call setup, routing, and mobility management. SMS messages are also transmitted over SS7. The typical network architecture for SMS communication is shown in Fig. 1.

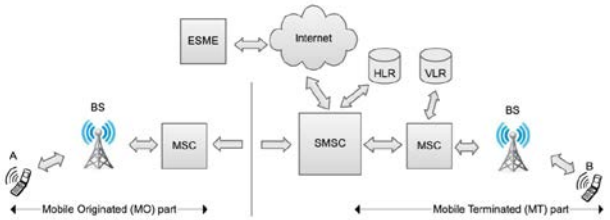

Fig 1. Typical network architecture for SMS

The network architecture consists of two main segments: the Mobile Originating (MO) part and the Mobile Terminating (MT) part. The MO part includes the mobile device (A) of the sender, a base station (BS) which provides all the radio infrastructure necessary for wireless communications, and the Mobile Switching Center (originating MSC) which routes and switches all traffic in and out of the cellular system for the sender. The MT part also includes a base station (BS) and the terminating MSC for the receiver, as well as a centralized store-and-forward server known as SMS Center (SMSC). The SMSC accepts and stores messages, retrieves account status, and forwards messages to the intended recipients. The 2 databases, Home Location Registrar (HLR) and the Visitor Location Registrar (VLR) assist the SMSC. The HLR contains the permanent mobile subscriber information, whereas the VLR contains temporary mobile subscriber information [8].
For a more detailed discussion on SMS operation, we refer readers to [9] and [10].

2.2 Service classes

According to [12], there are primarily two classes of SMS service. The first is guaranteed service. This service class guarantees delivery, offering zero losses. The second is the non-guaranteed service. The rationale for offering two service classes is due to the existence of two different uses of SMS: person-to-person messaging and bulk messaging. While the loss of person-to-person messages in not acceptable, bulk messages, such as promotional messages and messages from information services, can tolerate occasional losses. In this paper we focus on maintaining constant response times under increasing loads for the non-guaranteed service class by making use of the existing infrastructure available.

2.3 End to End Latency

We define End-to-End (E2E) latency or response time, as the total time the user has to wait to get a response for his/her request. This measure is the actual response time from any SMS based service (in this particular case txtWeb), as perceived by the end user. As illustrated in Fig. 2, the end-to-end latency consists of multiple components. The formula for calculating E2E latency for txtWeb is shown in (1).

$$E2E = MO\ latency + Platform\ latency + MT\ latency \quad (1)$$

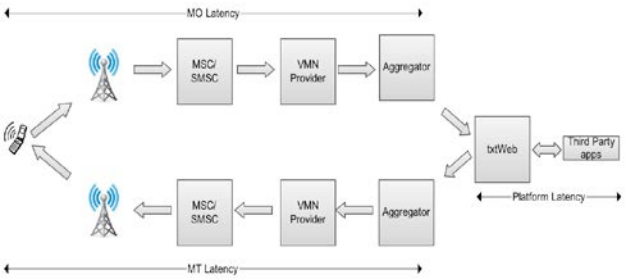

Fig 2. Various components which add to txtWeb's End-to-End latency

Mobile originated latency is the latency that occurs before the SMS request reaches the platform, whereas mobile terminated latency is the latency that occurs after the SMS response has been composed and leaves the platform. The latency introduced into the system due to the processing that happens within the platform is termed as platform latency.
The SMS request (MO) sent by the user travels to the SMSC, which then forwards it to the VMN provider and from there to an aggregator and finally to the txtWeb platform. The platform then has to process this request and make a HTTP call to get the response for the given application identified by the unique keyword. This response is sent as an SMS to the end user via the MT route.

2.4 Data

In 2012, we obtained permission from the concerned authorities and measured the end to end response time data for an Indian SMS based service txtWeb.
txtWeb is defined as a highly efficient real-time communication channel for textual, time-sensitive, non-intrusive, relevant and short updates. It is a global platform where anyone with a mobile phone can discover and consume information and content just by 'texting' keywords, and receive back content. txtWeb aims to provide information accessibility to those population segments that lack internet access by exploiting the universality of text messaging services. It is a popular SMS based service in India, with the total number of users

crossing 5 million (1 million active users per month) by Jan. 2013. There are multiple apps on txtWeb, similar to apps in an app store like Google Play or iTunes App Store. The end user can choose to use any app of his choice by typing a specific keyword, which acts as a unique identifier.

The main data set we collected for this experiment were the E2E records for this service over two specific periods of time: during the month of February 2012, and after the strategies proposed in this paper were implemented, during the month of May 2012.

2.5 End to End Latency Measurement using Android App

We measured the end to end response times by means of an Android application, which was programmed to send an SMS every 5 minutes to the txtWeb platform with the current timestamp recorded, which is treated as the timestamp an MO message was sent. These Android devices are located in various labs distributed in different geographical locations in India, so as to not bias the end-to-end latency values from one particular telecom circle or geographical region.

Each SMS request travels by means of a telecom carrier (e.g. Airtel or Vodafone) to an SMS aggregator, which then converts this SMS request to a HTTP request. This HTTP request then gets forwarded by the SMS aggregator to the txtWeb servers. The txtWeb servers use the keyword in the request and map it to a third party application URL. Next, a HTTP request is made to the above application which echoes back the same message it got as the request. Finally, the response from the application is sent back to the Android device as an MT message, and as soon as the response is received by the Android application, the timestamp is logged, which is treated as the time an MT was received. We consider the difference between the MT timestamp and the MO timestamp as the E2E latency.

We have built an Android application for calculating the end-to-end latency based on the above mentioned logic. The end-to-end latency values are measured as and when a request comes to the platform and then this record is stored in the database. A single database record contains information about the MO and MT latency, which when added to platform latency, forms the end-to-end latency. This data was then displayed on a web application dashboard where the stats such as end-to-end latency and best and worst response times were highlighted for each of the 27 txtWeb numbers.

There were multiple challenges faced during the implementation of this Android application. The Android Operating System imposes a hard limit on the total number of SMS that can be sent. A total of 100 SMS messages can be sent per hour as per the Android OS. The minimum number of messages we needed to send every hour for the purpose of end-to-end latency measurement was 324, since we had 27 local numbers which were monitored every 5 minutes. In order to achieve this, we used the Android Debug Bridge (ADB), a command line tool. Using ADB, we modified the values of constants such as 'sms_outgoing_check_interval_ms', which control the time interval between the SMS that are sent by the device. Another challenge we faced was the loss of data due to device crashes. Due to the devices being switched on and constantly sending SMS 24*7, after a certain period of time we observed device crashes which lead to loss of end-to-end monitoring data for a particular time period. Since this was a critical part of the monitoring setup, we decided to push values into our dashboard by means of a HTTP request, as soon as a request response flow was compelete. In this manner, even if the device crashes, we lose no transactional data, since the data has already been captured in the dashboard and the database associated with it.

We recorded multiple values in the database for the purpose of tracking the end-to-end latency, some of which include sendingTime, ReceivedTime, ResponseTime and Status. We used the end-to-end latency records stored in the above format between the time range 01-02-2012 00:00:00 and 29-02-2012 23:59:59 and the second set between the time range 01-05-2012 00:00:00 and 31-05-2012 23:59:59 for our analysis.

During the above mentioned time periods, the traffic on the txtWeb platform was extremely high and the volumes could be equated to that of a 'flash crowd' event. These specific date ranges were chosen, as there was a cricket series going on in these two months respectively, which lead to very high traffic on the platform. Also, this would give us a clear comparison of the difference in end-to-end latency values before (month of February) and after (month of March) the strategies proposed in this paper were implemented in the txtWeb ecosystem. We picked a few specific date samples where the traffic patterns clearly depict the general traffic trend in these two months and have drawn some conclusions based on these.

We try to address the problem of sustaining constant end to end latency even during times of very high load for SMS based services by means of the two strategies described below.

3. Two Proposed Strategies

3.1 Geo affinity based SMS routing

Majority of the SMS based services in India today work on a pull based model, owing to the recent government clampdown on SMS spam, or the push model. The user sends a request for content to a particular number and gets back the response from the service concerned. This

MO-MT cycle characterizes a typical person-to-application and application-to-person scenario. This interaction does not receive the same priority as that of a person-to-person SMS and hence is at a higher risk of being delivered late by the operators, after the person-to-person traffic is delivered. In this paper, we propose using geo affinity based routing as a strategy to maintain constant end-to-end latency in these cases.
Consider the scenario where only a single user facing number exists for an SMS service in the entire country. During peak loads, every single user request (MO) will bombard this single VMN (Virtual Mobile Number), and inevitably leave it clogged. We propose the introduction of local numbers for every province, where the service has high usage. This would help in solving the latency issues, as a single number does not get bombarded by the entire SMS traffic from all the users of the SMS service in the country. This load would then be distributed among VMNs in different geographical regions. This can be achieved by means of geo-affinity based SMS routing.

The location of the end user can be detected via various means, based on the type of mobile handset owned by the user, as illustrated below :

- The user sets his location either while registering for the SMS based service or by means of an IVR (Interactive Voice Response) service during the registration process
- The user sets his location by sending a text message to a preconfigured application to the SMS based service. The service then associates the user set location with his mobile number and uses this for all practical purposes wherever location is required
- The user's smartphone capabilities are utilized to detect the current location of the end user and SMS routing logic is based on this detected location

Based on the user set location, the SMS based service can intellgently decide the VMN that should be used to serve the user's request as shown in Fig. 3.
In case of txtWeb, a single number was advertised as the primary number (which belonged to Karnataka, a province in India). Once a user from a different province accesses the primary number, the replies from txtWeb service were delivered using the user's local number (based on his location). This change in the VMN from where the user's response was originating, was seamless to the end user, and the user would be prompted to hit the reply button in his/her SMS inbox and continue interacting with txtWeb using the local number.

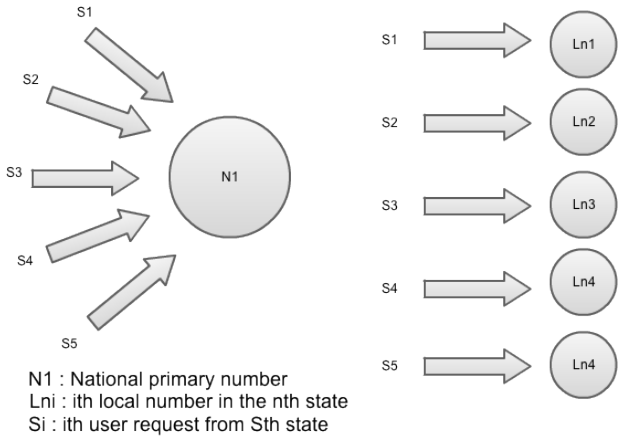

Fig 3. Visualization of the concept of geo affinity based routing

The proposed solution would help any SMS service to handle its MO traffic more efficiently as the load to a single VMN will be reduced by many folds, since the users would reply to the local number that they got a response from. Following this step, the MT would also be routed from the same local VMN the MO was requested from, which in most cases would be a VMN closest to the physical location of the user. This solution is extremely beneficial to the end user as well, since the end users from different provinces can access the local numbers of the SMS based service and hence avoid paying national SMS charges.

3.2 Traffic based intelligent SMS routing

As highlighted in 3.1, the local numbers would solve the issue of routing traffic from different states to different VMNs. This solution cannot however, reduce the E2E latency for high volume of incoming traffic from a single province. Some services, such as txtWeb are extremely popular in a particular province, due to which a large chunk of the traffic will be from that province. This may account for a large chunk of the total traffic for the service, and introducing local numbers will not solve this specific issue.
For this purpose, we have introduced the concept of traffic based intelligent SMS routing. For every province, we propose the introduction of multiple VMNs (Virtual Mobile Numbers). The response to a user's request can be served from a different VMN, based on the user's mobile number. A simple logic, for example would be routing based on the total number of VMNs in a region and using the last digit of the user's mobile number to decide which VMN the user receives a response from, as shown in (2).

$$LB_u = LB_N \quad (2)$$

Where $N = (N_u \% 10)\% \Sigma V_i$

LB_u *: Load balanced VMN for a user u*

N_u *: Mobile number of a user u*

V_i *: Total number of VMNs in a particular province*

We perform a modulo operation on the last digit of a user's mobile number and the total number of VMNs available in a particular province to get the VMN allotted to a particular user.
This helps to reduce the MO clogging on a single number as well, as when the user hits reply to a txtWeb response, he/she would automatically use the load balanced number, and hence reduce the number of requests to a single number in a province where the SMS based service is widely popular and heavily used as illustrated in Fig. 4.

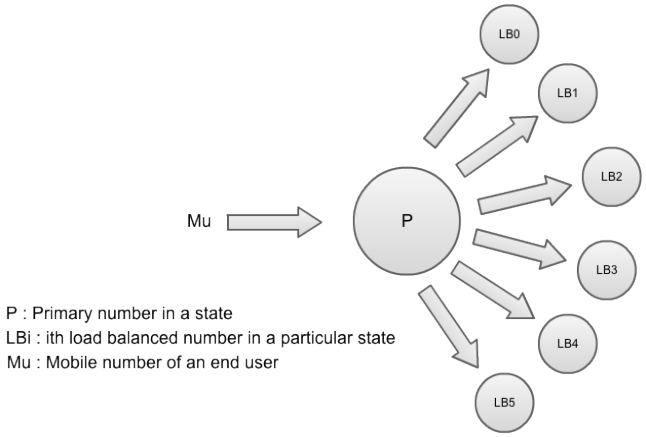

Fig 4. Visualization of traffic based intelligent routing

Fig. 5 and Fig. 6 depict the RequestCount (total number of requests received by the txtWeb Platform) vs. E2E latency graph before and after the strategies related to geo affinity based routing and traffic based routing were introduced in the txtWeb ecosystem. In Fig. 5, we can clearly observe that the E2E values were rising at an alarming rate when the RequestCount was around 1.3 million, but after the changes were implemented, the system was easily able to handle RequestCounts upto 3.5 million without much increase in the E2E values (Fig. 6). The E2E value recorded at peak load in case of Fig. 5 was 61.66 seconds. However after the strategies proposed in this paper were implemented, the E2E values reduced to around 45.74 seconds even at peak loads of 3.479 million requests.

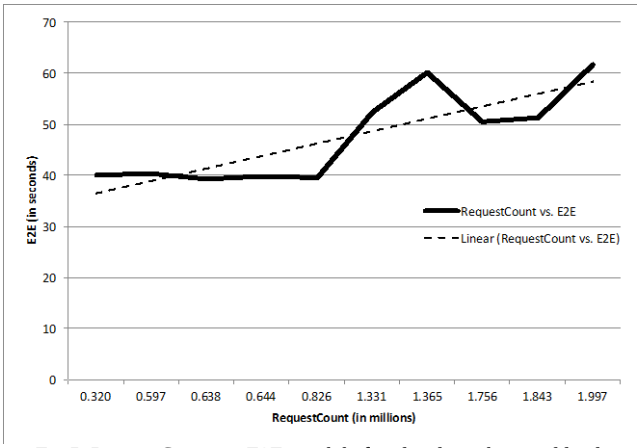

Fig 5. RequestCount vs. E2E graph before local numbers and load balancing were introduced

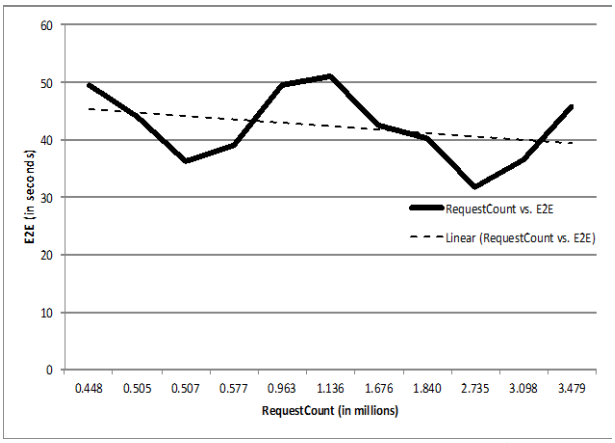

Fig 6. RequestCount vs. E2E graph post implementation of local numbers and load balancing

4. Related Work

There has been a lot of research interest, especially in the reliability aspects of this medium of SMS, since it is such a ubiquitous and easily accessible medium of communication. Meng et. Al in [8] analyzed and characterized a baseline for reliability of SMS in terms of message failure ratio and latency. They have concluded that flash crowd events are one of the primary reasons for the high end-to-end latency observed in SMS networks. In this paper, we attempt to solve the problem of achieving reliable end-to-end response times and maintaining constant end-to-end latency during peak loads with the existing SMS infrastructure by using geo affinity based routing and traffic based load balancing.
They have analyzed SMS data sets collected from a nationwide cellular carrier. The presented results are based on measurement and data from a live and popular SMS based service in India, txtWeb. This traffic does not

include any user-to-user traffic, only traffic which from a user to a system and system back to the user. We are focusing on the specific use case of maintaining constant SMS latency for all SMS based services, which usually tend to be very high for services that work over VMNs. In this work, we analyze incoming messages from all kinds of handsets, which is to the tune of more than 4 million requests per day, ranging from most basic phones to the highest end smartphones. The end users of this service are from various parts of India, and this ensures that there is no location bias as such.

Prieto et al in [12] have suggested means to alleviate the network congestion by making fundamental changes to the existing SMS infrastructure. This would mean significant amount of architectural changes at the SMS gateways, which involve cost and take considerable amount of time to implement across the entire ecosystem. Due to the closed nature of cellular networks and unavailability of real data, the authors in this study are forced to make assumptions about traffic loads, service delivery times, message expiration times and other parameters that are used in their models. The changes proposed in this paper are based on end-to-end latency from a live SMS based service and would require no architectural changes to implement, which would eliminate a significant amount of cost and time.

In [13], the author evaluated Quality of Service (QoS) of an international SMS, based on a test SMS generator connected to an aggregator. Due to the nature of the tests conducted, the QoS estimation was not carried out at peak loads, but only in normal or test loads. In our work, all of the end-to-end latency measurements were carried out in varying load scenarios, both at minimal load and at peak loads, when the system was bombarded with millions of SMS requests in a single day. Moreover, the measurement was over a broad period of time, in order to normalize the end-to-end latency values and to ensure that the good or bad performance of an aggregator on a particular day when the tests were carried out, does not lead to a bias in the end-to-end latency measurement.

5. Conclusion

SMS based services and text messaging in general is an extremely popular and indispensible tool for billions of people across the world. It is widely used by various businesses and the government as the primary means of communication of various alerts, emergency messages and other important communication. Hence, we believe it is of paramount importance that during the high volume and high traffic scenarios, the SMS services should respond to the end user's request with constant E2E latency. We have also surveyed and monitored various other SMS based services in India and found that E2E latency is a common issue for such services in general.

We conducted a data based study of the end-to-end latency for the SMS service - txtWeb, and found that the latency values remain constant under increasing loads as depicted by the charts and data, after geo affinity based routing and traffic based intelligent load balancing were implemented, in comparison to the earlier results (Fig. 5).

We have gone ahead and proposed two strategies in Section III to circumvent the exponential increase in the latency suffered during flash crowd events and other such high traffic scenarios. By means of live experiments we conducted on the txtWeb service, we show how the sudden and exponential increase in SMS volumes was handled seamlessly. In case of this particular experiment, the proposed solution handled up to 1.7 times the normal load, maintaining constant E2E latency values throughout the time frame of the experiment.

Finally, we would like to emphasize that this study is based on measurements obtained from three major SMS aggregators in India. It would be useful to collect data sets from other aggregators, similar SMS based services from different parts of the world and conduct experiments using international SMS to further test and validate the generality of the results presented in this work.

References

[1] The World in 2010: ICT Facts and Figures http://www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf (2010)

[2] ICT Statistical Highlights 2012, http://www.itu.int/ITU-D/ict/statistics/material/pdf/2011%20Statistical%20highlights_June_2012.pdf (2012)

[3] Telecom Regulatory Authority of India Press Release No. 143/2012 http://www.trai.gov.in/WriteReadData/PressRealease/Document/PR-TSD-May12.pdf (May 2012)

[4] Office of the Manager, "SMS over SS7," Tech. Rep. NCS TID 03-2, National communications system, December 2003

[5] Y. Feng, Y. Liu, J. Chen, & K. Liu. "Statistic analysis and automated diagnosis of a short-message-based authentication system", in Communication Technology (ICCT), 2011 IEEE 13th International Conference on (pp. 692-697). IEEE.

[6] H. Rheingold, "SMS disaster warning system," December 2004

[7] J. Jung, B. Krishnamurthy and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for CDNs and websites," in Proc. of World Wide Web (WWW), 2002.

[8] X. Meng, P. Zerfos, V. Samanta, S. H. Wong, & S. Lu (2007). "Analysis of the reliability of a nationwide short message service," in INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE (pp. 1811-1819)

[9] G. Peersman, S. Cvetkovic, P. Griffiths and H. Spear, “The global system for mobile communications short message service,” IEEE Personal Communications, June 2000.
[10] Telecommunications Industry Association, “TIA/EIA-637-A, Short Message Service.” December 1999.
[11] P. Zerfos, X. Meng, V. Samanta and S. Lu. “A Study of the Short Message Service of a Nationwide Cellular Carrier,” in Proc. of ACM SIGCOMM Internet Measurement Conference (IMC), 2006
[12] A.G. Prieto, R. Cosenza and R. Stadler. “Policy-based congestion management for an SMS gateway in Policies for Distributed Systems and Networks,” 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on (pp. 215-218)
[13] N. Mulkijanyan. “Evaluation Procedure for QoS of Short Message Service: International SMS Route Analysis” (Doctoral dissertation, KTH, 2011).

Gopi Krishnan Nambiar received his B.E. degree, from P.E.S. Institute of Technology in 2011. He has been working on the txtWeb Project, at Intuit India Development Centre since 2011. His research interests include mobile computing, natural language processing and machine learning.