**Journal of Kerbala University , Vol. 11 No.4 Scientific . 2013**

# Extending Public Switched Telephone Networks Billing Services Using VoIP Applications

# تعزيز الخدمات التحاسبية لشبكات الهاتف العامة باستخدام تطبيقات الصوت عبر بروتوكول الإنترنت

Saif Ali Abed
Computer Science Department-Kerbala University
Saif.a@uokerbala.edu.iq

## I. ABSTRACT

Public switched telephone networks (PSTN) has been used for decades to provide the telephony service but more and more subscribers are migrating to the modern mobile networks and the Internet. Subscribers are abounding PSTN, use it as a secondary communication option or a media to access the internet, though, legacy PSTN are expected to continue for many years before the complete extinction or total transformation to a pure broadband internet. PSTN operators are under continues pressure to provide new services and billing options rapidly and at low development costs and a possible solution is introduced here. This paper introduces Asterisk, the voice over internet protocol (VoIP) server, and the Personal Home Page(PHP) programing language as a billing service platform to be used in PSTN or any telephony system that can provide the database level access to PHP. The proposed design defines a way to interact with the PSTN database directly using the Asterisk VoIP server as call processor triggers a PHP program to perform the required calculations or transactions for every call.

A system design and call flow diagrams has been presented thenimplemented using Asterisk , MySQL Database management system and PHP among other open source tools. Multiple performance tests have been carried out to highlight the performance bottlenecks and the applicability of the system as well.

**الملخص**

يتم استخدام الشبكات الهاتفية العامة منذ عقود لتوفير خدمات الاتصالالصوتي بين المواطنين ولكن المشتركين يقومون بالهجرة الى شبكات الهاتف المحمول والانترنت بوتيرة متصاعدة حاليا. يقوم المشتركين بترك شبكات الهاتف التقليدية نهائيا او استعمالها كوسيلة اتصال ثانوية او كوسط ناقل لخدمة الانترنت ولكن على الرغم من ذلك فانه من المتوقع ان تستمر الشبكات الهاتفية التقليدية لعدة سنين أخرى قبل الانقراض النهائي او التحول الكلي لشبكات ناقلة لخدمة الانترنت. ان شبكات الهاتف التقليدية تحت ضغط كبير لتوفير خيارات وطرق تحاسب جديدة بسرعة وبكلف تطوير منخفضة وهذا البحث يقدم احدى الطرق الممكنة عمليا لتحقيق هذه الأهداف. يقدم البحث الخادم Asterisk(خادم نقل الصوت عبر بروتوكول الانترنت) ولغة PHP البرمجية كمنصة لتطوير خدمات تحاسبية ليتم استعمالها في شبكات الهاتف العامة او أي شبكة تستطيع توفير قاعدة بيانات يمكن الاتصال بها عن طريق لغة PHP. التصميم المقدم يصف طريقة للتفاعل مع شبكات الهاتف العامة باستعمال الخادم Asteriskكمعالج مكالمات يعمل على تفعيل برامج مطورة بلغة PHP وظيفتها القيام بالعمليات التحاسبية لكل مكالمة مستلمة.تم تقديم تصميم النظام ومخططات تدفق المكالمة ثم تنفيذها باستخدام خادم Asterisk، نظام إدارة قواعد البيانات MySQL ولغة PHP بالإضافة الى مجموعة منالأدوات مفتوحة المصدرالاخرى. تم اجراء اختبارات أداء متعددة لإبراز اختناقات المحتملة واثبات القابلية العملية لتطبيق النظام.

## II.INTRODUCTION

Few years ago, average citizens had no alternatives to public switched telephone networks (PSTN) as the communication solution for the daily usage, though, the speedy rising of the mobile networks and broadband internet is offering more mobility and features which is pushing the subscribers away from PSTN towards these alternatives[1], however, traditional PSTN are expected to continue to exist for many years before the complete extinct or change to some other kind of networks like broadband internet networks.

Session initiation protocol (SIP) and the digital audio codecs arrived alongside broadband internet and quickly become the dominating standards of voice over internet protocol (VoIP) and eventually enable the broadband users to do cheap or free voice calls. VoIP can be seen as the way to use the internet or any packet network for voice transmission.

VoIP's wide adoption triggered the development of all kinds of hardware and software to route, process, monitor and bill SIP calls with a huge variety in features and capacity. As a result, SIP based trunks and devices started to invade the PSTN world as a cheap, flexible alternative for the time division multiplexed (TDM) links. Deeper VoIP applications in the legacy PSTN core switching elements or services is also possible and the proposed work will provide an example for this kind of integration.

The proposed work shows one possible approach to use VoIP standards and equipment to develop three services to PSTN and demonstrates how Asterisk VoIP server can be used to power a low cost service development environment. These new service will enable the PSTN user to query the bill ,check how much a call will cost and what was the cost of the last call. The proposed work can be used as a way to enhance PSTN customers' billing and management experience without the need to change anything in the existing PSTN installation. The basic concept is to transfer the call flow from PSTN to Asterisk, execute some call processing codeand play the results to the caller then hang-up.This is possible even if the PSTN switch doesn't support SIP by transforming the call back and forth between SIP and TDM via voice gateways or TDM cards.

## III. REVIEW

### A. PSTN Billing

PSTN switching cores evolved to fully automatic computerized routing elements capable of serving huge numbers of subscribers and capable of establishing thousands of concurrent calls with smaller space and power consumption. Billing methods and usage metering started to shift from periodic pulse metering(PPM)to the modern and more flexibleintelligent networks(IN)based billing.

In PPM, the charge is determined by charging pulses given to the counter of the A subscriber during the call. First pulse may be given as soon as the destination telephone is answered, and after that with predetermined time intervals [2].

Intelligent Network billing works differently,when a user calls a destination within the IN switch,the service switch point (SSP) will send a request to the service control point(SCP).The SCP usually responds with the charging and routing needed for the call [3]. When the call ends, SSP will signal SCP that the call has been ended; the SCP will start a sequence of processes which normally include the creation of a call detail record (CDR).

The generated CDRs normally are saved in text or binary files which can be converted laterto database records. Each CDR contains at a minimum the following field: the number making the call, the number receiving the call, when the call started (date and time), how long the call was (duration)[4]. More fields are normally included like call charge and some charging calculation details. These CDRs will be used to calculate the total monthly bill for the post-paidsubscribers; they are also exchanged with the PSTN service providers to calculate and audit the arrangement between the interconnected providers.

### B. VoIP and SIP

VoIP usesinternet protocol (IP) based networks like Internet, Intranets and local area networks (LAN) as a media to transmit voice between two points. The voice will be digitalized to pulse code modulation (PCM), compressed using voice codecs, and transmitted as an IP packets from the first point to be reconstructed at the second point. This means that voice communication will be treated as just another application running on the internet instead of a service running over a dedicated network.VoIP has been implemented in various ways using both proprietary and open protocols and standards. The most extended are H.323 and SIP[5].

SIP has been standardized by the internet engineering task force (IETF) for establishing VoIP connections [6]. It's an application layer protocol based on the client/server model. The protocol itself is very similar to HTTP in the way the two nodes communicate throw a series of textual representation request-response messages.SIP represents the signalling part of the call and it doesn't carry the voice packets which are transmitted over UDP.SIP signalling commands are sent in clear text between the clients and the server to establish the media channel link and only after that point the media will be transmitted from the caller and the called clients.

### C. Asterisk

Asterisk is an open source VoIP server which can offer communication channels using almost all the VoIP protocols including SIP. It supports SIP, H.323, Media Gateway Control Protocol (MGCP), inter-Asterisk exchange (IAX2) and T1/E1 lines when equipped with the required TDM interfacing cards. Asterisk is a powerful free alternative for the commercial VoIP serversas a result of many reasons, mainly because of the fact that it is being developed and operated over the open source Linux operating system beside the active community of users and developers.

The main applications of Asterisk are deploying a low cost communication system for enterprises in the form of IP private branch exchange (PBX) or as a gateway to convert voice calls from one protocol to another. Combined by a billing software, Asterisk has been used as prepaid calling systems. It has been also used as a custom interactive voice response (IVR) server, conferencing server,number translation,predictive dialler, call queuing with remote agents and remote offices for existing PBX [7]. Another powerful usage of Asterisksis to buildthe routing element in low cost GSM networks like OpenBTS which uses the Asterisk as a VoIP backhaul to build a cellular network that can be deployed and operated at substantially lower cost[8].

All calls directed to Asteriskare processed inside a dial plan; the dial plan is a prioritized list of instructions or steps that Asterisk will follow when the calls arrive to the system[9]. Dial plans are divided into contexts completely isolated of each other. Eachcaller (SIP client or SIP peer) will be assigned context for its calls and each context should be capable of handling all the valid extensions or destinations for that caller including the special extensions like voicemail and IVR menus.

Applications are the main building blocks of the contexts instructions; they perform various kinds of functionality like playing sound files, getting input from callers or calling a SIP client. Most Asterisk applications accept arguments and return some values to reflect the success or the failure of the application execution.

Asterisk also provide an Asteriskgateway interface (AGI) which is an interface for adding functionality to Asteriskusing many programming languages like Perl, PHP, C, Pascal, Bourne Shell[9]. Using AGI application in a dialplan will provide the call channel parameters (caller number, called number, channel type, etc.) to an external program write in any supported language like PHP or Perl[10]. Some Functions and classes will be used inside the program (the PHP file for example) to get the channel variables or to instruct Asterisks to do some actions with the call. Using this architecture, the flexibility of the mature programing language will be added to Asterisk capabilities in call management to provide powerful telephony applications like the one presented in this work.

## IV. RELATED WORK:

The ideas of extending Asterisk functionality and extending legacy telephony systems have been proposed in different contexts. Hitchcock[11] investigated Asterisk capability to provide the depth of services that will be required for a VoIP solution and the available options to extend the core system functionality by using four methods of modifying the core system functionality were presented: dial plans, external interfaces, AGI, and the Application API. His study focused and provided detailed experiments using the AGI interface. Heexaminedthese methods and investigates the advantages and disadvantages for each. Two simple example services have been developed; AGI weather service to provide local temperature information on demand, and a service application to play music.

In [12]Femminella et al used JAVA Application program interface for integrated networks (JAIN) to reduce time-to-make and extend the service portability over multiple networks and devices. The work provided a system design and performance tests for three different service architecture alternatives.

## V. ASSUMPTIONS:

An imaginary core of a PSTN system will be used to demonstrate the methods and design introduced in this paper. We will assume that this PSTN system consists of a switching element and a simplified databases schema of three tables.

The first table is in the PSTN database is the charging matrix table shown inTable 1; it basically holds the prefix-rate information and it will hold the charge of a 1 minute call to any number start with the specified prefix.

The second table holds the CDRs of the system. A new record will be added to this table after each call,

Table **2**shows example CDRs for the PSTN system.The CDR format will be assumed to have the caller number, called number, timestamp, call duration, and call cost. At the CDR creation phase a matching prefix should be found for the called number to calculate the call cost for that destination. The call duration will be rounded to minutes and multiplied by the rate to calculate the cost for this particular call.

The last tableis the billing table which will save the billing and invoicing history for each client (PSTN subscriber). This table will have a new record whenever the system generates a new bill and the subscriber pays that bill; a sample data is shown in Table 3.

The billing database will contain a record for the last payment for each user and thetimestamp of bill generation.

Table 1. Charging matrix table

| prefix | Cost (cents) |
|---|---|
| 0770 | 15 |
| 0780 | 13 |
| 0790 | 10 |
| 001 | 100 |
| 00962 | 110 |

Table 2. CDR table

| caller | called | timestamp | Call_duration (minutes) | Call_cost (cents) |
|---|---|---|---|---|
| 032348622 | 07706227766 | 2012-09-07 01:22:00 | 3 | 45 |
| 032346373 | 07807384950 | 2012-09-07 12:24:03 | 1 | 13 |
| 032355631 | 07807356345 | 2012-09-07 13:10:29 | 2 | 26 |
| 032355631 | 00962567694345 | 2012-09-07 14:42:07 | 10 | 1100 |
| 032387634 | 07709278346 | 2012-09-07 16:33:21 | 4 | 60 |
| 032383484 | 07709349784 | 2012-09-07 16:34:07 | 2 | 30 |

Table 3. Billing table

| Phone_number | bill_generation_timestamp | amount (cents) |
|---|---|---|
| 032387634 | 2012-09-07 10:42:07 | 5010 |
| 032387634 | 2012-08-07 11:02:33 | 7652 |
| 032387634 | 2012-11-07 10:21:53 | 23411 |
| 032387634 | 2012-10-07 10:22:55 | 7682 |
| 033587243 | 2012-08-07 12:44:22 | 6345 |

## VI. SYSTEM DESIGN:

The system design is shown in

Figure (**1**).The system consists of an Asterisk server connected to the PSTN main switch via a voice trunk. A voice gateway or a TDM interfacing will be used to convert the E1 link to a VoIP link if the PSTN switch doesn't support VoIP trunks. This link is required to receive the incoming calls from the PSTN subscribers by the Asterisk server which will trigger all the database query operations involved to provide the caller with the required service. The PSTN switch should be configured to forward the calls from the subscribers to the Asterisk server if they call a predefined service number or a short code, 777 for example. The forwarded call must contain information about the caller id which will act as input variables for Asterisks dial plans.Asterisk will be also connected to the database server which contains the CDR, billing and charging matrix tables via a LAN.

Figure (2) shows the call flowchart for the design. The flowchart is divided into two parts,the upper part is executed inside the PSTN switch and the lower part by the Asterisk server using AGI and PHP.

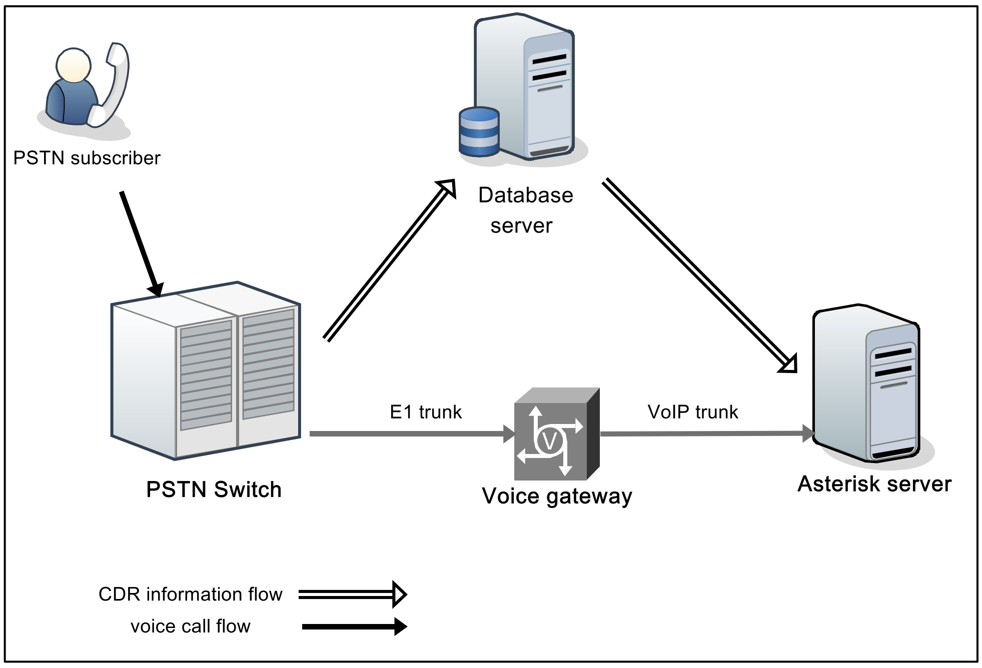

Figure (1) System design

The subscribers' calls will be forwarded to the VoIP gateway trunk only if the call destination is 777. The gateway job is to transform the TDM calls sent from the PSTN side in to a VoIP calls directed to the Asterisk server. When the call reaches Asterisk the caller will hear a welcome announcement generated by the server and he/she will be asked to choose a service code number then press the hash sign key in the phone. Depending on the user input, the required service will be triggered.

If the user choose the call cost calculator service by pressing 1 then he/she will be asked to enter the destination number and then press the hash sign key. A php code will be executed to access the charging matrix and execute a query against Table 1 to find the call cost per minute for this particular destination.

By pressing 2 the caller will choose the last call cost query which will trigger a PHP code to access the CDR database and execute a query against CDR table shown in

Table **2** to find the CDR of the last call made by the caller and play that particular call cost to the caller.

Choosing the current bill calculation service by pressing 3 will trigger a PHP code to access the CDR database and execute a query against CDR table shown in

Table **2** and the billing table shown in Table 3 to calculate the current bill for the user by aggregating the calls cost for all the calls made after the last payment.

## A. Test Bed Setup

A simulation environment has been prepared to test the applicability and performance of the proposed system. The following tools have been used:

- mysqlslap, a benchmark tool designed to simulate workloads and measure MySQL database server performance.
- SIPp, SIP traffic generator and analyser.
- vmstat, system resources monitoring tool.

MySQL 5.5.24 has been installed on a 64 bit windows 7 machine with Intel i5-2430@2.40GHz and 4GB RAM. The server has been loaded with a CDR table identical to

Table **2** in structure and contains 100000 random call records belongs to one thousand random subscribers. Asterisk 1.8.11.0 has been installed on another machine with Intel i5-2450@2.50GHz and 4GB RAM running 64 bit Centos 5.7. Fast Ethernet network has been used to connect all the nodes including a third machine used to generate the SIP traffic.

## A. Performance and Numerical Results

Figure (3) shows the results of a performance test has been applied to the MySQL server to investigate the variation of query execution time compared to the number of the simultaneous connections trying to execute a read query on the CDR table. This performance matric has been used to investigate the existence of a bottleneck in the database query operations required during every call made to the service short codes. Mysqlslap has been used to simulate 25 to 1300 concurrent connection (in steps of 25) to the MySQL server.Each connection is trying execute the structured query language (SQL) query identical to the one that will be used by Asterisk to find the cost of the last call from the CDR table.

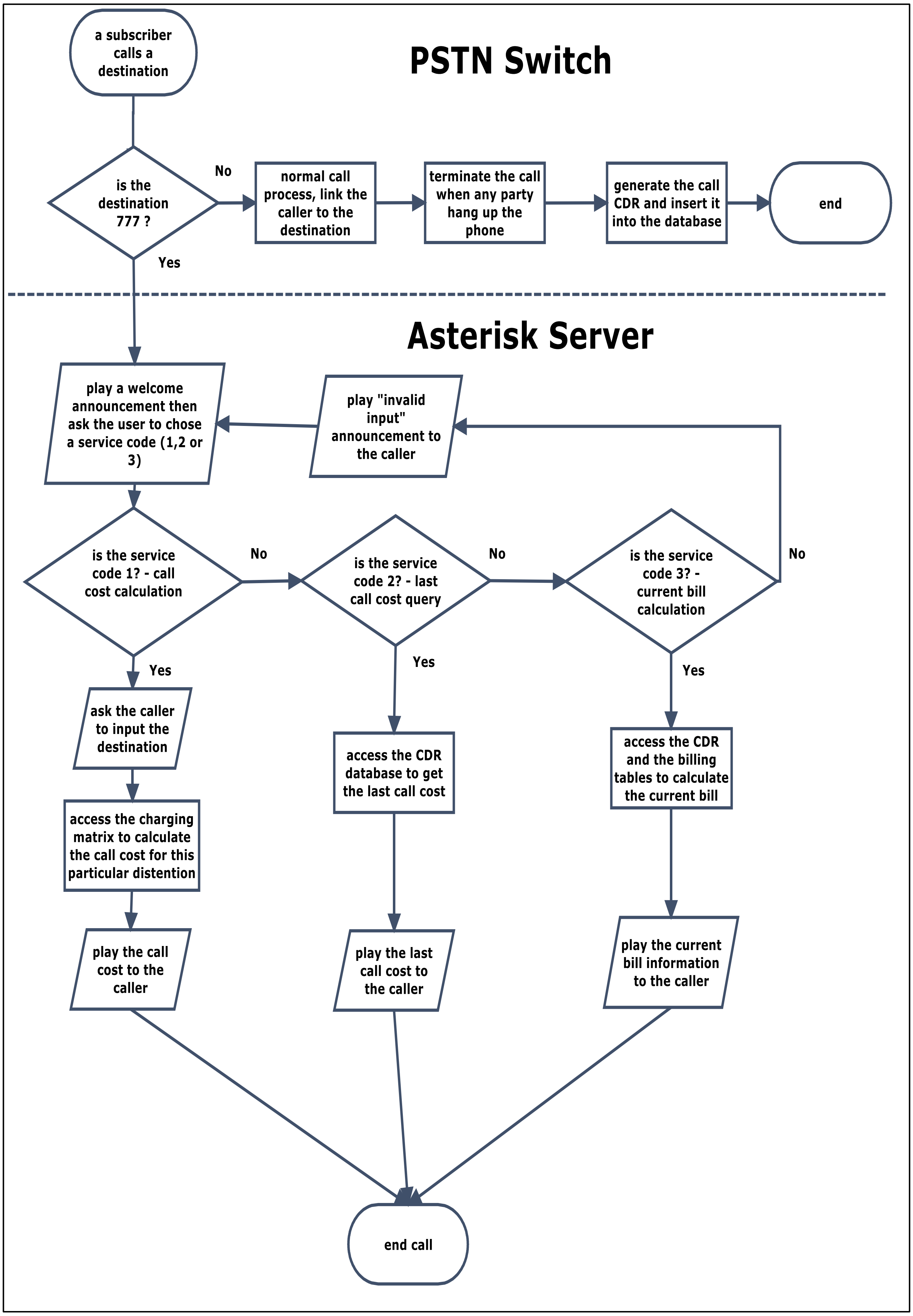
PSTN Switch
a subscriber calls a destination
is the destination 777 ?
No
normal call process, link the caller to the destination
terminate the call when any party hang up the phone
generate the call CDR and insert it into the database
end
Yes
Asterisk Server
play a welcome announcement then ask the user to chose a service code (1,2 or 3)
play "invalid input" announcement to the caller
is the service code 1? - call cost calculation
No
is the service code 2? - last call cost query
No
is the service code 3? - current bill calculation
No
Yes
ask the caller to input the destination
access the charging matrix to calculate the call cost for this particular distention
play the call cost to the caller
Yes
access the CDR database to get the last call cost
play the last call cost to the caller
Yes
access the CDR and the billing tables to calculate the current bill
play the current bill information to the caller
end call

Figure (2) System flowchart

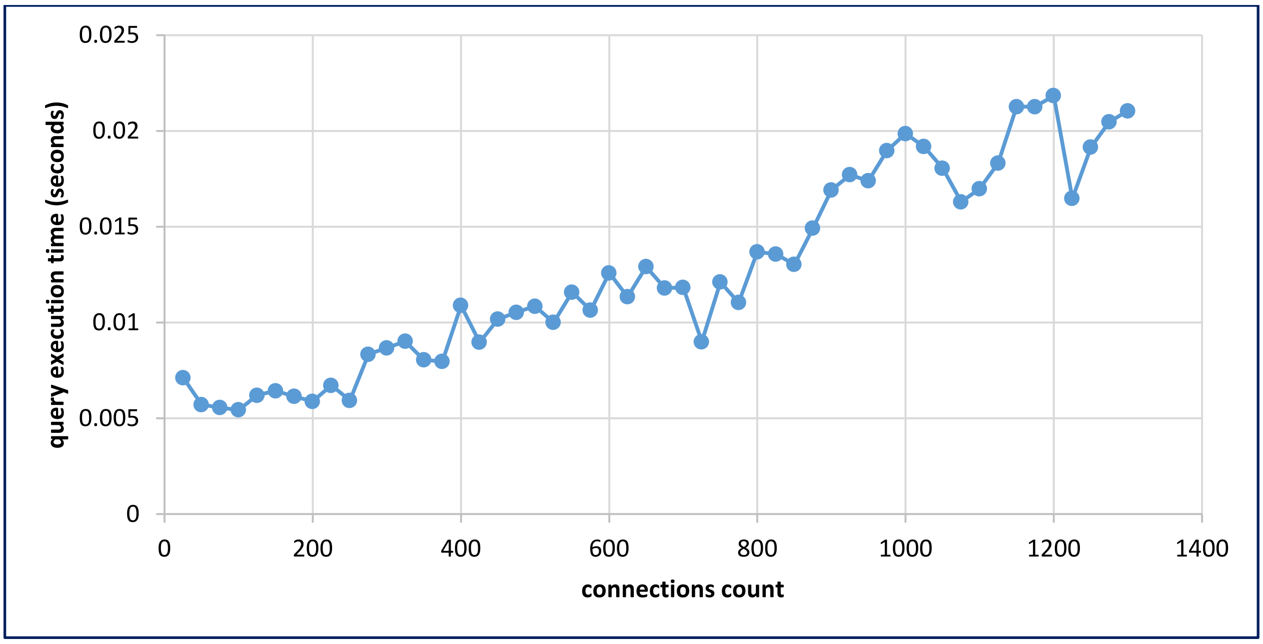

Figure (3) Query execution time vs. concurrent connections

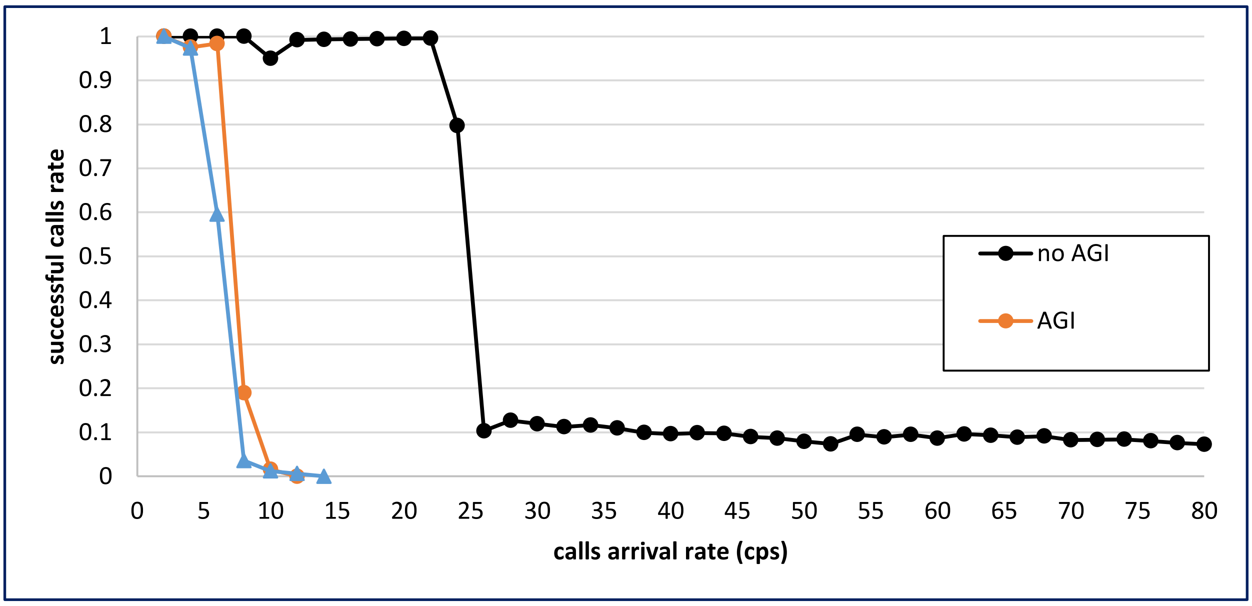

Figure (4 ) Throughput vs. calls arrival rate.

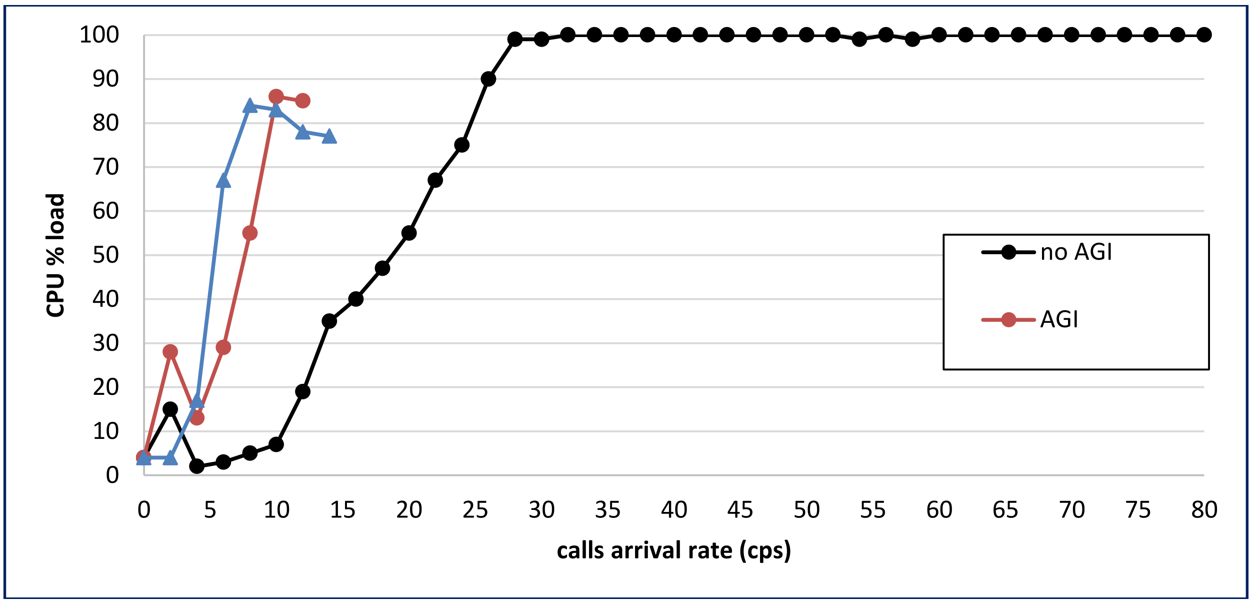

Figure (5 ) Percentage CPU usage vs. calls arrival rate.

## VII. Performance Evaluation

SIPp has been configured to generate 15 second callsat avariable call arrival rate rangesfrom 2 to 80 calls per second (cps) in steps of 2 cps. Each step last for 5 minutes in which SIPp calculated the call success rate and vmstat recorded the system resources usage. Three short codes has been defined inAsteriskto answer the calls and play a sound file for 1 second then wait for the caller to end the call or forcibly end it after 16 seconds.

Three performance tests has been done on the proposed system, the first test include calling a short code to activate anAsterisk dial plan without AGI functionality. The second test has been applied using a short code to trigger a dial plan which includes a call to a minimum PHP AGI code to measure the impact of using AGI on the system performance without using any functions in the PHP code itself. The third test has been made to a short code to activate the actual last call cost service that includes a PHP AGI call and a MySQL query. Figure (4) shows the successful call rate for the three tests.The system was able to handle 22, 6 and 4 cps for the three tests respectively and then reached an unstable state.Introducing the PHP AGI to the dial plan produced significant overhead and reduced the system performance, the system was capable of handling 6 cps only for the calls to the short code with no PHP AGI calls in its dial plan and become unstable at higher call rates. Adding the functionality of querying the CDR database and playing the last call cost to the caller to the PHP script reduced the system performance further to 4 cps which is by far less than the capacity of the MySQL server since it is clear from

Figure (3) that the database server is capable of handling more than 1200 query per second without introducing any significant delay, therefore, the database does not create a bottleneck in the simulation environment.

Figure (**5**) shows the central processing unit (CPU)utilizationpercent of the VoIP server during the load tests described above. It shows that the system bottleneck is the processing power of the CPU. Increasing the call rate at each test lead the system to unstable state and increased the CPU utilization further thereforethe VoIP server started to drop the calls since the CPU was unable to process the new calls and keep the previously established calls alive.

## VIII. CONCLUSION:

This paper successfully introduced the open source VoIP server "Asterisk" as service creation and deployment platform for the legacy phone systems. Three services to the PSTN subscribers has been developed as an example of how to build new services over the existing PSTN infrastructure using Asterisk as the core call processing element. A simulation environment has been prepared and multiple performance tests has been done against the proposed system. The simulation was able to process scale 4 cps for the 15 second calls and up to 60 concurrent calls (4 cps* 15 second) which can be seen as an acceptable rate compared to the limited hardware resources used in the tests. The paper also highlighted the performance impact of using PHP AGI in Asterisk dial plans and how it reduce the server capabilities by approximately 80% due to the cost of starting the PHP interpreter and starting the execution of the PHP code.

Although the work focused on PSTN but the same ideas and methods can be ported to any mobile networks as long as these network can provide the required database level access to the CDRs and charging calculation related tables. This will enable the mobile operators to develop new services rapidly and at low cost compared to the usually expensive and complicated alternatives offered by the commercial vendors.

## IX. REFERENCES

[1] I. Nurhalim and D. Gunawan, "PSTN VoIP Application Support System Design using mobile Short Message Service (SMS): Case Study of PSTN VoIP Missed Call Notification to mobile phone by SMS," in Electrical Engineering and Informatics (ICEEI), 2011 International Conference on, 2011, pp. 1 –4.

[2] Teemu Makela, "Accounting in the PSTN," Department of Computer Science; Helsinki University of Technology, 1999.

[3] Samer Jamous, "The Evolution of Switch Charging - Multimetering Method," presented at the Poznan Telecommunications Workshop, 2003.

[4] D. Su and F. Qi, "An Approach for Ensuring the Reliability of Call Detail Records Collection in Billing System," 2009, pp. 100–103.

[5] L. Tian, N. Dailly, Q. Qiao, J. Lu, J. Zhang, J. Guo, and others, "Study of SIP protocol through VoIP solution of 'Asterisk'," in Mobile Congress (GMC), 2011 Global, 2011, pp. 1–5.

[6] R.Arora,"Voice over IP:Protocols and standards," Network Magazine,, 23rd of November, 1999.

[7] M. Spencer, M. Allison, and C. Rhodes, "The asterisk handbook," Asterisk Documentation Team, 2003.

[8] M. Cabral, I. Almeida, C. Melo, and A. Klautau, "Low-cost GSM telephony in the Amazon region based on open-source/open-hardware projects," in Communications, 2009. LATINCOM'09. IEEE Latin-American Conference on, 2009, pp. 1–6.

[9] Jim Van Meggelen,Leif Madsen,and Jared Smith,Asterisk,the futuer of telephony.O'reilly, 2007.

[10] "Asterisk AGI - voip-info.org." [Online].http://www.voip-info.org/wiki/view/Asterisk+AGI. [Accessed: 03-Jul-2012].

[11] J. Hitchcock, "Decorating asterisk: Experiments in service creation for a multi-protocol telephony environment using open source tools," Rhodes University, 2006.

[12] M. Femminella, R. Francescangeli, F. Giacinti, E. Maccherani, A. Parisi, and G. Reali, "Design, implementation, and performance evaluation of an advanced SIP-based call control for VoIP services,"in Communications,2009.ICC'09.IEEE International Conference on,2009, pp.1–5.