Asterisk as a Public Switched Telephone Network Gateway for an IMS Test Bed

Alton MacDonald*, Rodolfo Cartas†, and José Incera‡
Instituto Tecnológico Autónomo de Mexico (ITAM)
Mexico, D.F., Mexico
*Email: alton.macdonald.7886@gmail.com
†Email: rodolfocartas@gmail.com
‡Email: jincera@itam.mx

Abstract—Network operators are in the process of adopting the IP Multimedia Subsystem (IMS) as their main underlying infrastructure which will in turn be used as a Service Delivery Platform (SDP). This process requires the migration of current services to their IMS counterparts, as well as a compatibility layer enabling successful communication between IMS and other network infrastructures. Any and all functionality within IMS is normally available through an Application Server (AS). This paper presents an implementation of a Public Switched Telephone Network (PSTN) Gateway which is able to successfully interface with an IMS test bed. Consequently, it also allows calls to be placed between IMS and other Voice over IP (VoIP) providers. The gateway is enabled as an AS with the help of Asterisk.

I. INTRODUCTION

The IP Multimedia Subsystem (IMS) is the first functional proposal of a Service Delivery Platform (SDP) which makes use of open and generic interfaces to be used in Next Generation Networks (NGN) [1]. Although it was originally conceived by the Third Generation Partnership Project (3GPP) in order to help manage multimedia sessions between users, it now aims at the creation of an architectural independent framework which allows interconnection between traditional and mobile telephone users [2]. IMS is able to create a SDP through the help of Application Servers (AS) which populate the service plane with reusable common enablers and resources [3]. Said AS are able to implement the logic of any function or service within the IMS service plane, thus achieving a rich panoply of independent service delivery blocks to be used in the creation of attractive and innovative services.

Telecommunication service and infrastructure, much like any other competitive market, requires guaranteed compatibility between its individual players. And IMS, as a new contender to this complex playing field, requires the ability to successfully interact with what are slowly becoming legacy telecommunication infrastructures. In order to thrive in this competitive environment, IMS must first be able to seamlessly interact with current telephony operators, whether they be traditional Public Switched Telephone Networks (PSTN) or the more recent Voice over IP (VoIP) players. Herein lies the quintessential need for IMS to communicate with PSTN and exchange traffic with its enterprise level VoIP counterparts.

The IMS is frequently criticized for implementing a non-standard version of SIP [4]. This precise argument is the very cause of many connectivity problems when IMS is forced to exchange data with enterprise VoIP solutions. Other problems, such as SIP routing and registration inconsistencies, also stem from this uneventful side effect. Although steps are being taken to minimize these problems by proposing 3GPP’s SIP modifications to the Internet Engineering Task Force (IETF) for their inclusion in future RFC specifications [3], this is not enough to ensure current compatibility between IMS and PSTN.

This paper focusses on the implementation of a PSTN Gateway used within an IMS test bed and a local telephone network. Specifically, this paper presents the lessons learned from previous attempts of incorporating Asterisk [5] with the Open IMS Core [6]. However, these attempts did not have the desired outcome due to the problems mentioned earlier and were not solved using traditional telephony logic. This paper presents how to overcome such shortcomings by using a different approach, specifically by employing an extension of IMS telephony logic.

This paper is organized as follows: Section two briefly presents an introduction to IMS and how it is able to provide the basic building blocks that form a SDP with the help of Application Servers. Section three presents the problems faced by IMS when trying to implement a PSTN Gateway. Section four shows how these problems can be solved. Then, section five presents future lines of work for improving the current implementation of the PSTN gateway. And finally, section six concludes this paper.

II. IP MULTIMEDIA SUBSYSTEM

The IMS is proposed by the 3GPP as part of their Release 5 and has been improved upon in Releases 6 and 7 [3]. Its growing maturity has adopted four main objectives: (1) to combine the latest technological tendencies; (2) to achieve the paradigm of the mobile internet; (3) to provide a common platform which allows the creation of multimedia services; (4) to create a mechanism which increases income due to its use in mobile networks [3]. Among its many other functions, IMS tries to improve the delivery of Internet enabled services to mobile telephony users with an improved Quality of Service.
(QoS) not achievable with current 3rd Generation Mobile Network (3G) solutions. In essence, IMS aims to provide a SDP capable of handling multimedia streams.

Developed with open standards in mind, the 3GPP and the Internet Engineering Task Force (IETF) have been working together to define the technologies behind IMS. With the help of protocols such as IPv6, Session Initiation Protocol (SIP), Real-Time Transport Protocol (RTP), Diameter, and the Session Description Protocol a basic architecture has been defined for IMS consisting of a collection of functions joined together by open standardized interfaces [3]. The IMS architecture is divided into three planes. The first of these is the Access Plane defined by the network access technology used by the clients [3]. The second is the control and connectivity plane which is in charge of establishing delivery and services routes [3]. It is able to reproduce the routing and charging functions required by a telecommunications network operator. The third plane constitutes the service plane which creates a rich panoply of service blocks [3]. This final plane is able to recreate a SDP by providing IMS with modular building blocks to be used in the rapid deployment of innovative services. Figure 1 can be used as a visual representation of the IMS architecture previously described.

Two main functions constitute the core engine behind the IMS control and connectivity plane: the Home Subscriber Server (HSS) and the Call/Service Control Function (CSCF). The HSS serves as a central repository for all information regarding the users and services within an IMS realm [2, 3]. It keeps track of each user’s profile, identity and authorization credentials. It also contains service profiles which can enable or disable the delivery of certain services automatically. The HSS also has the ability to guarantee the successful communication between clients located in different access technologies with the help of additional quality parameters [2].

The CSCF specializes in controlling calls and sessions within an IMS realm. By itself, it is able to setup, destroy, and route SIP sessions, giving it the ability to replicate a traditional PSTN operator. In order to accomplish this, the CSCF is further divided into three subfunctions: the Proxy-CSCF (P-CSCF), the Interrogating-CSCF (I-CSCF), and the Serving-CSCF (S-CSCF). The first of which acts as a SIP proxy between the client and an IMS realm in charge of authenticating the user [3]. The I-CSCF interrogates the SIP requests generated by the users in order to determine the final IMS realm, or S-CSCF, destination which will process the incoming request [3]. In other words, it determines the proper destination service route from the incoming SIP request. Finally, the S-CSCF is in charge of routing the SIP request to its final destination, whether it be another IMS realm or an Application Server. It is responsible for verifying the user’s subscription to the service identified by the SIP request and enabling the delivery of said service if the verification is successful [3].

The IMS service plane is composed primarily of AS which are capable of implementing any service or function not defined in the IMS architecture. They frequently handle a larger array of protocols not supported by IMS in order to facilitate the delivery and migration of foreign services to an IMS realm. Their definition within the IMS architecture is generic in order to propagate diversity and innovation.

Users within an IMS realm possess a public and private identity. The former is used as a presentation card outside the local IMS realm; it is used as a unique identifier for external entities. A user can possess more than one public identity and can register any number of these simultaneously, saving bandwidth and network resources [3]. The most common type of public identities are known as SIP-URI and TEL-URI, both of which are Universal Resource Identifiers (URI). On the other hand, the private identity is known solely by the local IMS realm [2] and is used to authenticate and register a user’s terminal [3].

III. THE PROBLEM

The need for a PSTN gateway within the IMS cannot be stressed enough. This gateway is essential for its guaranteed future adoption, as well as the competitive edge required to dislodge current operator infrastructure. However, the incompatibilities stemming from the unstandardized implementation of SIP given by IMS, has greatly hindered this seemingly simple task.

Since at least 2006, attempts have been made to offer Voice over IP (VoIP) enterprise solutions to IMS users by using Open IMS Core [6] and Asterisk [5] as service providers. Developed by the Fraunhofer Institute for Open Communications Systems (FOKUS), Open IMS Core implements the core functions of an IMS realm by providing the HSS, P-CSCF, I-CSCF, and S-CSCF functions. Figure 2 presents a diagram of the interaction between each function. The main obstacle encountered when Asterisk is introduced to this diagram is where and how to connect it to the IMS realm.

Most of the additional functionality added to IMS can be tacked on through the help of AS. However, when analyzing Figure 1, the gateway components (MGCF and MGW) do not appear to be configured as Application Servers. Therefore, defining service routes can be tricky, if not impossible. The problem becomes whether to enable the PSTN gateway as an AS or a separate component. For both cases, a second problem arises: how to properly configure the desired interaction between IMS and said gateway.
A. Previous Works

In 2006 Lian Wu and Anders Hagelskjær Aasgaard [7] approach the problem of integrating Asterisk and Open IMS Core. Specifically, they analyze the migration of VoIP enterprise solutions towards IMS by focussing on the scenario where a user is registered to an IMS realm and wants to access enterprise SIP solutions. In this scenario they establish two main objectives that must be completed. The first requires the user to register to the VoIP service from within an IMS realm and the second requires the user to successfully establish a call with another user by using said VoIP service. They propose a migration roadmap with four solutions to be implemented in three different stages which depend on the amount of deployed IMS operators. Figure 3 can be used to help visualize the explanation presented in the next paragraph.

![Figure 3. IMS Migration Roadmap and Enterprise VoIP Solutions [7]](image)

The first stage has few IMS operators in place and one proposed solution for migration. They dub the first solution with the name “Forking” [7]. This solution is quite simple, as it proposes that the best operation is to have the IMS operator simply forward all enterprise traffic to the VoIP provider (in this case Asterisk). The second stage has a moderate amount of IMS operators in place and two proposed solutions for migration. These solutions require more features, specifically the use of a presence server, from the IMS operator and a more complex approach to registration and session setup. The second solution, dubbed “Client Based” [7], makes use of an intelligent User Equipment (UE) which is able to detect its presence in an IMS realm and register to the enterprise service dynamically. The third solution, dubbed “Presence” [7], expects the IMS operator and VoIP enterprise to have an agreement to share the user’s presence. This way the network (IMS operator) is able to intelligently detect the user’s presence and register him to the appropriate services. The third, and final stage, is one where all of the operators have an IMS based infrastructure and one final migration solution is presented. The last solution, named “Link Registration” [7], is a much smarter (automated) one which requires features and capabilities not yet available by access technologies. It suggests that the IMS operator keep track of the link registration of the user in order to enable VoIP services which can be delivered to the client UE. Additionally, the enterprise services are enabled and delivered using native IMS service routing policies. In other words, at this stage VoIP enterprise services will be available through IMS without having to enable special treatment for their delivery. Wu and Aasgaard do a good job outlining a migration roadmap. However, they do not implement any of their proposed solutions.

In 2007 Fei Yao [8] tries to implement the second (“Client Based”) solution presented in [7] using Open IMS Core as an IMS operator and Asterisk as a VoIP enterprise service provider. He takes a much more detailed approach by analyzing this solution and the problems it presents in its implementation. He focusses his energy on analyzing Open IMS Core’s robustness and implementing the client registration for said solution. Even though Yao is able to successfully implement the registration part of the Client Based solution, he could not enable the delivery of the enterprise services to the client UE. Yao is able to implement the client registration by duplicating the S-CSCF into two separate entities. One S-CSCF is dedicated to processing normal IMS SIP requests, while the other one acts as a proxy that forwards the enterprise SIP requests to Asterisk [8]. By doing this Yao is able to easily register an IMS user to enterprise services. Unfortunately, this same procedure did not work for the call setup objective. A redirection server was used to forward the session requests to Asterisk, but the P-CSCF would refuse to forward said requests to their enterprise destination [8]. Due to time constraints, Yao was not able to determine the cause of this problem. Figures 4 and 5 show a diagram briefly describing the sequence of SIP messages that Yao tried to implement.

B. Current Shortcomings

Both [7] and [8] approach the problem of Asterisk and Open IMS Core integration in a scenario where legacy VoIP solutions must adapt to IMS. If, however, a different approach is used where IMS is now a majority and must react to legacy VoIP enterprise solutions, the incompatibility issues previously presented can be solved as has been demonstrated in [9]. Under this new scenario, the VoIP enterprise service provider is not considered a separate entity; it now forms part of the IMS logic. Therefore, it can be implemented within IMS as an AS.
This setup closer reflects the “Presence” solution presented by [7], in the sense that the intelligence resides within the IMS operator network and not the client UE.

The client UE no longer requires special treatment (of SIP messages) in order to register to enterprise services because these are now directly enabled through the IMS registration and service profiles. The registration to enterprise VoIP solutions and call setup to external SIP users is accomplished with the help of a softswitch Private Branch eXchange (PBX), otherwise known as Asterisk. This solves the problem of connecting Asterisk to IMS, which in turn allows call placement to IMS users originating from a VoIP provider. Calls from IMS users to their SIP counterparts present unexpected complications; call setup is not symmetric. Furthermore, the registration to more sophisticated enterprise services (such as voicemail, directory services and calls to other legacy telephony systems) is unavailable under this rudimentary setup.

Even though the P-CSCF successfully forwards SIP requests to the enterprise service provider, it refuses to attend such requests. The reason why this occurs does not have to do with how Open IMS Core and Asterisk are connected. Instead, it turns out that their SIP implementations are incompatible. Thus, further expanding upon the problems that arise from IMS using a non-standard SIP implementation. Specifically, Open IMS Core uses additional SIP headers for routing and multimedia capabilities that Asterisk does not support. Since Asterisk does not know how to interpret said headers it simply rejects all incoming IMS traffic. This could very well have been the same problem encountered by [8] and incorrectly labeled as a fault with the P-CSCF.

IV. THE SOLUTION

In practical terms, both previous works show that Asterisk can be used in varying degrees of success together with Open IMS Core. Registering Asterisk as an AS helps prove that this solution both makes use of IMS logic and is a step closer to fulfilling the migration roadmap presented in [7]. It is also able to rid itself of the duplicated S-CSCF proposed by [8]. Additionally, by knowing that the problem presented in this last scenario resides in the exchange of incompatible SIP implementations, a temporary solution can be presented with relative ease.

The first step in identifying where to apply the temporary solution resulted in analyzing the packets exchanged between Open IMS Core and Asterisk during call setup. This resulted in the realization that Asterisk consistently answered all IMS INVITE messages with status 402 Bad extension (unsupported). This led to believe that IMS users were using a wrong header extension during calls. This possibility, however, was ruled out after noticing that the same message was not generated during the SIP-IMS calls. Therefore, the problem must reside in Asterisk, possibly it was compiled without support for certain SIP functions.

The second step resulted in debugging Asterisk. Everytime the same 402 message was generated, the line Received SIP INVITE with unsupported required extension: precondition was again registered in the Asterisk log files. Knowing that log files are frequently generated by predefined statements in source code, the next logical step was to find the exact conditions that trigger this output in the log files and error messages in SIP responses.

Consequently, the third step made use of linux bash routines in order to filter through thousands of lines of code. This method found just one occurrence of the line Received SIP INVITE with unsupported required extension:. What followed was an analysis of the surrounding source code, which led to the realization that Asterisk tries to check for supported SIP headers before continuing with call setup. Again, since IMS makes use of additional SIP headers, Asterisk simply rejected all incoming calls assuming that it could not process them.

The easiest and fastest solution is to modify the Asterisk source code. By simply commenting out the section that checks for additional support in the SIP headers of incoming messages, Asterisk is able to reply to SIP requests originated from IMS users [9]. Table I shows the exact lines (14085 and 14090) that have to be modified in Asterisk source code version 1.4.22, file
This fix enables the delivery of VoIP enterprise services to IMS users. Among other services, calls to PSTN users are also enabled. Therefore, this fix not only is able to solve the problems presented in [8], but it can also add the functionality of a PSTN Gateway to an IMS test bed.

Table I

<table>
<thead>
<tr>
<th>Modification of Asterisk Source Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>14079 /* Find out what they require */</td>
</tr>
<tr>
<td>14080 required = get_header (req, &quot;Require&quot;);</td>
</tr>
<tr>
<td>14081 if (!last_strlen (required)) {</td>
</tr>
<tr>
<td>14082 required_profile = parse_sip_options (NULL,</td>
</tr>
<tr>
<td>14083 if (required_profile &amp; required_profile &amp; !</td>
</tr>
<tr>
<td>14084 /* At this point we only support REPLACES */</td>
</tr>
<tr>
<td>14085 // transmit_response_with_unsupported (p, &quot;420</td>
</tr>
<tr>
<td>14086 Bad extension (unsupported)&quot;, req, required);</td>
</tr>
<tr>
<td>14087 ast_log (LOG_WARNING, &quot;Received SIP INVITE</td>
</tr>
<tr>
<td>14088 with unsupported required extension: %s\</td>
</tr>
<tr>
<td>14089 p-&gt;invitestate = INV_COMPLETED;</td>
</tr>
<tr>
<td>14090 if (!p-&gt;lastinvite)</td>
</tr>
<tr>
<td>14091 sip_scheddestroy (p,</td>
</tr>
<tr>
<td>14092 DEFAULT_TRANS_TIMEOUT);</td>
</tr>
<tr>
<td>14093 }</td>
</tr>
<tr>
<td>14094 return -1;</td>
</tr>
</tbody>
</table>

However, this fix is not sufficient for future deployments of a PSTN Gateway within IMS. As can be appreciated in Table I, Asterisk at this point only supports the SIP REPLACES header. A more elaborate action is necessary if proper support for additional SIP headers is to be implemented in the PSTN Gateway. Fortunately, by knowing exactly where the problem resides and how to administer a temporary fix, establishing a permanent solution should not be that difficult.

In order to provide full support for VoIP enterprise services for IMS users, such as a directory and voicemail services, a few minor adjustments must be made to Asterisk configuration files. Since IMS is now in charge of registration, Asterisk no longer has control over the users that access its services. Therefore, if a user wants to enable advanced enterprise services, his user data has to be repeated in Asterisk’s configuration files. Additionally, due to the fact that Open IMS Core has problems registering more than one public identity implicitly, any public identity used by Asterisk must also be registered as a separate user. The best way to organize this information in a non redundant manner is to create all public identities in Open IMS Core. Then identities which contain a SIP-URI should be handled by the IMS realm, while identities containing a TEL-URI should be handled by Asterisk. This way the IMS realm can function normally with an easily identified SIP-URI and Asterisk will have a list of IMS users that are assigned a TEL-URI. Therefore, the users with a TEL-URI public identity have access to a broader range of VoIP enterprise services, such as support for routing of incoming calls from the PSTN operator.

Obviously, Asterisk requires the use of a Peripheral Component Interconnect (PCI) device which allows it to properly interface with the PSTN operator. Support for traditional telephony communication, such as Dual-Tone Multi-Frequency (DTMF) and Signalling System 7 (SS7), can be added through the use of appropriate modules [10]. These modules are undergoing constant improvements and it is only a matter of time before a complete PSTN Gateway can be available through Asterisk.

The proper functioning of the gateway was testing in [9] by successfully establishing calls between the different communication protocols supported by Asterisk. The IMS test bed communicates with the PSTN through a traditional telephone line. A direct consequence of this is that only one call can be established at any given time between IMS and PSTN users. In order to deal with this limitation, incoming PSTN calls are screened by an operator that makes use of the directory functions available in Asterisk in order to route incoming calls to their IMS destination. It is important to note that a jitter buffer had to be placed on the analog channel in order to counteract the delay experienced between IMS and PSTN users [9]. Said delay is caused by the creation of an additional service route used by an IMS session. Without the jitter buffer it is impossible to comprehend audio exchanged between the clients.

V. Future Work

As has been previously mentioned throughout this document, the current PSTN gateway is not ready for enterprise deployment since it has many rough edges that need to be ironed out beforehand. The following is a brief recap of the PSTN gateway’s deficiencies and proposals for their future improvement.

First, a reevaluation of the faulty P-CSCF [7] is to be accessed. It was stated in this paper that possibly the P-CSCF was in fact not faulty and the symptoms it presented were due to the incompatibility issues stemming from IMS and Asterisk using different implementations of SIP. The true cause of the connectivity problems should be further investigated in order to determine its origin and strengthen its corresponding weakness.

Second, after proposing a temporary fix to Asterisk source code, an elegant and robust solution should be implemented which adds support to additional SIP headers. A proposed starting point for this task would be to construct a list of headers used by IMS and incorporate their added functionality into Asterisk’s application logic. This would turn the commented lines into possibly another module for Asterisk in charge of solely handling IMS headers.

Third, although the duplicated S-CSCF was removed, an additional level of configuration and administration complexity was added to Asterisk configuration files, as well as Open IMS Core data integrity. This solution was able to simulate an intelligent network. The next step would be to implement said intelligence in an additional component, such as an AS dedicated to enforcing data integrity between user profiles in IMS and Asterisk. Furthermore, the simultaneous registration
of TEL-URI and SIP-URIs should be incorporated to the IMS Core.

Fourth, the PSTN gateway was only tested under specific conditions; those where only DTMF compatibility was necessary. Although Asterisk already has the ability to interface with SS7, this functionality should be extensively tested within an IMS context. Instead of connecting the PSTN gateway to a local telephony service, further testing should be done with Asterisk connected directly to a real telephone network operator.

In conclusion, Asterisk can be successfully used within an IMS test bed in order to guarantee compatibility with VoIP enterprise services, as well as function as a PSTN Gateway. The reason why this paper limits the use of Asterisk within the confines of an IMS test bed and not an enterprise solution is because a few kinks have to be resolved before the latter can be true. Specifically, a more rigorous testing is required to certify SS7 compatibility. Additionally, user configurations should not have to be repeated within Asterisk configuration. Open IMS Core should be able to implicitly register all public identities simultaneously and enable TEL-URI specific services without having to manually add such configurations to Asterisk’s user profiles.

VI. Conclusion

This paper presents an introduction to the IMS, as well as its main concern for ensuring compatibility with legacy telephony networks. Then, IMS is further explored; presenting how Application Servers are used in conjunction with IMS in order to create a Service Delivery Platform. Subsequently, this paper presents previous encounters between Asterisk and Open IMS Core, as well as their varying degrees of successful compatibility. This information is in turn used to incorporate Asterisk as an AS within an IMS realm. By using this approach, this paper is able to demonstrate the necessary steps required to enable VoIP enterprise services for IMS registered users, as well as the deployment of a PSTN Gateway for an IMS realm.

This paper’s findings reside in two main focal points. The first changes the approach used when incorporating Asterisk with Open IMS Core. A slew of problems resulted from previously analyzing them as separate operators. When Asterisk is treated as an AS the incompatibilities between said entities become easier to manage. The second finding discovers and temporarily fixes the incompatibility issue where Asterisk simply rejects all IMS SIP session requests. This last finding is essential in fixing and adding future compatibility between IMS and Asterisk enterprise solutions.

Then, future lines of work are presented to help achieve a much more robust solution than the one currently presented here. This paper can be used as a starting point for future development and support of SIP headers within Asterisk that will enable a successful communication between VoIP and IMS operators. It also points out a few faults incurred by the Open IMS Core implementation. In summary, it provides a possible solution for implementing a PSTN Gateway within an IMS test bed.

References