Research Journal of Applied Sciences 7 (8): 382-386, 2012
ISSN: 1815-932X

A Reconfigurable Architecture of SIP in Element Management System

J. Parvathi and B. Karthikeyan
School of Computing, SASTRA University, Thanjavur, Tamil Nadu, India

Abstract: The Element Management System (EMS) is one that is involved in the development of the functionalities of efficient management of network elements like the DSLAMs that handles high speed digital data streams from numerous subscribers, end users and the network service providers. DSLAM is a broadband access network element that supports multiple DSL transmission types (xDSL) and finds advantageous as it provides an efficient way of broadcasting with reliability and scalability and its support to triple play delivery (audio, video and voice supporting). The essence of EMS is to research with network management system (EMS GUI) that provides configuration, maintenance and the FCAPS (Fault management, Configuration management, Accounting management, Security management) management of the network element. This flexibility ensures the enhanced deployment of novel Network Element (NE) releases, enabling high penetration in market for new services. EMS manages the Session Initiation Protocol (SIP) configuration of DSLAMs as being supported by the integration of multiple types of media. There are SIP versions for the hardware of DSLAMs which evolve in parallel, most of the times independently. Generally, the design of the application will have core component and on top of the plugins that are defined for each version of the hardware release. SIP application is like having SIP core and SIP plugins for each DSLAM version which results in making frequent releases of all of these components on every introduction of a new SIP version. This results in lot of cost involvement, every time developing and making releases of the SIP core and plugins. So, the current design of SIP core application and plugins need to be revisited to reduce the redelivery of all of them for each new SIP version thus providing an optimal design of SIP application. The optimal design of SIP application is obtained by moving the SIP components to the SIP core and making a deviant from the plugins that is dependent of each SIP plugins for the new SIP versions that is being introduced. The former is removed from its content thus breaking the link between the plugins and the core for the updation of the new SIP versions. As a result on introduction of every new SIP version the SIP core to be redelivered to the customer instead of redelivering the entire former SIP core and the SIP plugins as both has a dependency for the new SIP version thus resulting in higher performance and cost effective.

Key words: SIP (Session Initiation Protocol), DSLAM (Digital Subscriber Line Access Multiplexer), FCAPS (Fault, Configuration, Accounting and Security Management), EMS (Element Management System), GUI (Graphical User Interface), DSL (Digital Subscriber Line)

INTRODUCTION

Element Management System (EMS) enables telecom operators to do provisioning of the software management of the DSLAMs and also the maintenance purposes and making backups and restoring them. EMS defines three levels for each network element namely core, framework and the plugin. The core layer defines the common functionalities of the network elements upon which the framework which acts as an interface application towards the network element as parsing the code and processing the files of the object models. Each entity in a network is known as managed object. The plugin application defines the functionalities specific to different network elements. EMS provides a Graphical User Interface (GUI) which allows configuring and monitoring DSLAMs in a user-friendly manner and also allows dealing with the DSLAMs configuration database, the so-called MIB. If information is requested by an operator. It is retrieved from the DSLAMs database. The communication between the EMS and the DSLAM is achieved by SNMP request and response design that defines a client/server relationship. SNMP (Simple Network Management Protocol) is a communication protocol which is defined as a method of managing IP-based networks including the network devices (SNMPv3 is preferred as it implements the user-based security). The SNMP manager maintains a database (called the SNMP Component Management Information Base) on the network elements that are fed by means of queries to the SNMP agents distributed throughout the network. SNMP messages are transported

Corresponding Author: J. Parvathi, School of Computing, SASTRA University, Thanjavur, Tamil Nadu, India

using UDP protocol. This has the advantage that control connections will not hang indefinitely when an SNMP agent becomes temporarily unavailable or goes off-line altogether. SNMP carries management information between managers (EMS) and agents (DSLAMs). Each DSLAM has its own Management Information Base (s) with his own set of standard set of statistical and control values but will only communicate upon request from the EMS.

Asante networks, Inc.:

- SNMP get: retrieves the value of an object at the agent
- SNMP set: sets the value of an object at the agent Only alarms are sent by the DSLAM to the EMS as an initiative action
- SNMP trap: allows an agent to notify the management station of significant events. In fact what is needed just like for the EMS is an IP connection with the DSLAM

MATERIALS AND METHODS

EMS functionality: Element management layer lies at the bottom of the Telecommunication Management System (TMS) also known as fifth layer of TMS takes care in management of the network elements say its configurations, correcting the faults (Kawagoe *et al*., 2011). EMS manages specific types of network elements. The EMS follows up the server-client architecture say EMS server and the EMS client (nCite technology). Server side is the use of Java platform; Java EE provides web services, component model, management and communications APIs that makes it a standard for implementing enterprise-class Service-Oriented Architecture (SOA) and next generation applications.

JBOSS is an open source application server that implements the Java Enterprise specifications and is used to run enterprise applications and the client side is the eclipse rich client platform. The server is for the session management, connection management, etc. and the client side is for the viewing and the editors' support. EMS GUI (Graphical User Interface) functions as an interface. The EMS communicates to the network element through SNMP.

Thus on the whole EMS maintains a real time monitoring of the network elements, setting and configuring the network elements equipments, store all the system and monitoring information in the database. One of the configuring and setting of the network element is the SIP configuration which has an architecture that contains two layers say sip core and the sip plugins. The function for SIP configuration is to administer service provisioning process, to manage and configure network element.

EMS SIP architecture: SIP is an application layer control protocol used to establish, modify and terminate multimedia sessions. It is run on either of the underlying layers TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). The SIP protocol is used as a signaling when the multimedia sessions is to be made between the two end points or multiple end points such as providing the functionality for VoIP (Voice over Internet Protocol) between two or multiple participants. The SIP supports name mapping and the redirection of services, allowing users to maintain a single, externally visible identifier, regardless of their network location, capability negotiation, allowing users to have appropriate usage of media, participants' management, allowing users to have a control of the incorporation of new user entries into the session (Bao *et al*., 2009).

The components of SIP are the SIP Client (UC) which sends a request for the initiation of a multimedia session, SIP Server (US) which responses to the requests and a SIP proxy server which acts as a mediator between the UC and US in handling the request and responses thus accomplishing the multimedia sessions between the users (Nahum and Wright, 2007).

EMS provides the call setup on the network element and is achieved by providing SIP facility to the network element. Thus, the whole call setup is made available by downloading the sip configuration file application on to the particular ONT (Optical Network Terminal) that terminates at the home premises thus activating the call setup on that particular port of the network element. The EMS support for VoIP services is handled by the EMS SIP and the services are made available on the ONT at the home premises using SIP-FTP. The SIP application contains the following components.

SIP core: As per the Fig. 1 the SIP core component contains the generic SIP configuration, management of the SIP profiles and the generic network element attributes for the services (say, POTS port (Plain Old Telephone

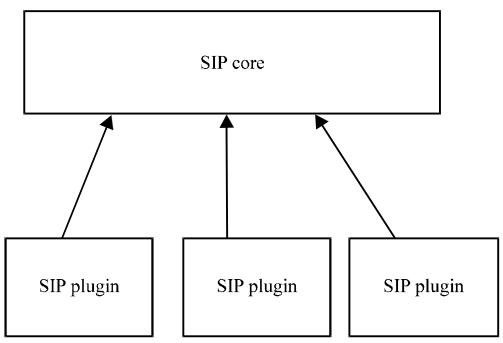

Fig. 1: EMS SIP architecture

System) and Voice services). The SIP core defines/creates the sip client and subscriber profiles for the supported SIP versions, defines/creates the objects (SIP objects) for the services (POTS and VoIP services) to handle the SIP attributes. It provides the management of the SIP client/Subscriber Profiles. It also takes care of the configuration of the SIP file generation job which is to be downloaded to the ONT. A client is a VoIP service client supported by an ONT and a subscriber is a VoIP service subscriber supported by a POTS Port. The client and subscriber profiles define the parameters related to the ONT VoIP and the ONT POTS.

SIP plugin: On top of the core is the plugins defined for each hardware release and each plugin contains specific attributes and handlings for the sip services as POTS/VoIP services for each component plug release as shown in the Fig. 1. It does the generation of the SIP configuration files and sets the file name SNMP object on the network element and also packaging the list of scripts, hardware configuration files which is made available to the ONT (El-Sawda *et al*., 2010).

RESULTS AND DISCUSSION

Current EMS SIP architecture: In the current architecture, the SIP versions are defined on each and every plugins. The SIP functionality depends on the SIP core and the SIP plugins. SIP plugins by taking care of packaging provides network specific configuration data's for the ONT and the handling of the VoIP and POTS services the scripts and hardware configuration files. Thus, a client profile, subscriber profile and a hardware profile on the whole constitutes as input to the network element provided the uploading of the configuration file to the ONT is conventional. SIP application causes the code update for number of components as part of a new GSIP version support or any change required in the existing scripts or hardware configuration files which causes an redelivery of all gsip plugs in case of new SIP version support.

SIP configuration in EMS: Element Management System configures a whole setup of VoIP call by provisioning SIP configuration on ONT. The ONT (Optical Network Terminal) serves as SIP client and terminates at the home/premise. The ONT is supported by SIP as well as POTS services for the happening of whole call setup. The provisioning of SIP to the ONT results in the availability of VoIP services like call waiting and call forwarding, etc. The SIP accessibility is done as a generation of configuration files that contains parameters/values so as to activate the VoIP session on to the ONT. The scripts describes the generation of the final SIP configuration file to be downloaded to the ONT through ftp. The configuration file which contains scripts describes the properties of the call services. The properties are grouped as client properties and subscriber properties. The ONT registers the voice services with the SIP Proxy server to use the voice capabilities. An ONT POTS is given dial tone by the ONT when the SIP application is loaded into it.

The SIP application is downloaded to the ONT by the activation of ftp server, i.e. the ONT uses ftp to get the SIP configuration file. Two types of traffic will be routed from the ONT to the corporate network. First, the ONT needs to find the ftp server on the EMS machine) and get the configuration file. Second, once the configuration phase is done, the ONT (SIP client) needs to find the SIP Proxy and bind gateway with client and will be in running on the same service VLAN (Virtual Local Area Network). SIP is provisioned on the ONT by providing a SIP configuration file containing SIP properties and using SNMP to communicate the file's name and location to the ONT. This involves the generation of configuration files per subscriber and making it available on the particular network element.

EMS generates the SIP configuration file and the ONT will retrieve the configuration file either from an external ftp server or an EMS server as shown in the Fig. 2. For the generation of the script the EMS will allow the user to enter the ftp server IP address of the machine which will be the IP address of the machine where EMS runs. The EMS will always check if the ftp server IP address entered in the VoIP service window belongs to the station on which it executes. If it doesn't, then it will not only generate the SIP configuration file but it will also invoke a script that will ftp this configuration file over the

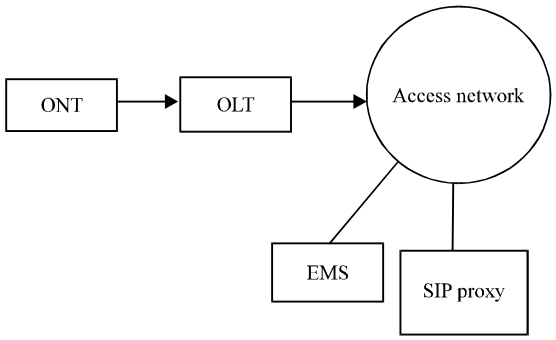

Fig. 2: Provisioning SIP configuration file in EMS

firewall with user id and password id required to the ftp server machine. At the end, it will of course let the ONT know that the file is ready.

Once the configuration file is generated, EMS will invoke script along with the ftp server ip address, username, password, configuration file as input. The script establishes the ftp connection with external server and the file will be uploaded. ONT's requires a file which contains the subscriber, client hardware specific file to activate the service. The file name/ftp server information designate in the VoIP service level via SNMP attributes.

Then, the ONT will act as an ftp client to pickup the configuration file from ftp server when the data path is established between ONT and ftp server.

Proposed EMS SIP architecture: SIP plug is dependent on NE plug, SIP core and framework. These components should have been built and deployed before deploying SIP plug. In the existing testing stage, the result is that the compilation and the deployment of the core and the plugs of GSIP on inclusion of a new SIP version to the SIP core and SIP plug researchers fine. The proposed design is that in the testing stage, on inclusion of a new SIP version to the core, the build and the redeployment of the core and the framework to be a success without building the SIP plug and by just performing a deploy action on SIP plug made to research as backward compatibility and the deployment to be a success. In the deployment of the SIP architecture, the SIP core is deployed and then the SIP plugins for the SIP components to be configured for the new SIP version introduced.

On the introduction of a new SIP version there is a necessity of redelivering all the plugins in this case as there is a dependency for the new SIP versions of SIP plugins to the SIP core. An improved solution is to move the package of the SIP components, the scripts and the hardware configuration files external to the plugins but this leads to the redelivering of the all the plugins as each plugin expresses a dependency on the SIP component. As this becomes a drawback, a more cost effective way would be to move the package of the SIP components that is fed as input to the network element, the scripts and the hardware configuration files and the processes involved to the SIP core rather than to the plugins externally. This leads to the redelivering of the SIP core rather than redelivering each plugins when a new SIP version is introduced. By moving the SIP components to the SIP core does not make all the sip plugins to find a dependency on the SIP components when a new SIP version is being introduced as the whole processes involved is deployed in the core itself. Only the SIP core needs to be redelivered.

Reliablity: The movement of the SIP components to the core is considered as the best way of redesigning as the changes are to be done only in the sip core and not in the SIP plugins as it just take care in the updation of VoIP services and the ont services. These changes are done at the interior portion of the SIP architecture thus making it reliable as it does not affect the functionality of the SIP application.

Performance: The support of the new SIP version to the SIP core makes it to be redelivered by updating the core with the new SIP version and without any related changes done in plugs with the build a success. Thus, the whole SIP package on introduction of new SIP version is made available to the customer by just redelivering the SIP core instead of the entire SIP core and plugin components due to the dependency of SIP plugin to the core making the proposed architecture cost effective.

CONCLUSION

EMS manages the SIP configuration of DSLAM for difference SIP versions as it is supported in the DSLAM. Evolution of SIP versions are happening very frequently. Current design of the SIP application is having SIP core and SIP plugins for each DSLAM version which is resulting in making frequent releases of all of these component which results in lot of cost involved every time developing and making releases of the SIP core and plugins. So, the current design of SIP core application and plugins is redesigned to reduce the redelivery of all of them for each SIP version.

The design changes to SIP architecture in order to optimize development of the application for every SIP release is done by analyzing existing design proposals and finding the best one or come up with a completely new design proposal. The new design proposed is by moving the SIP functionality to the SIP core and redelivering it if incase of introducing new SIP releases, the SIP plugs works as backward compatible to the previous releases. The updation of the new SIP version in the SIP folder which is present in the SIP core is done by making it available to the core its related support and no such updation in the SIP plugs ended with the redelivering of the SIP core thus, breaking an interlink between the core and the plugins for the new SIP versions support.

REFERENCES

Bao, D., D.L. Carni, L. de Vito and L. Tomaciello, 2009. Session intiation protocol automatic debbuger. IEEE Trans. Instru. Measur., 58: 1869-1877.

El-Sawda, S., P. Urien and R. El-Sawda, 2010. A trust communication with sip protocol. Proceedings of the IEEE/ACS International Conference on Computer System and Application, May 16-19, 2010, Hammamet, Tunisia, pp: 1-6.

Kawagoe, T., H. Kawakami, K. Soga, K. Tanaka H. Okazaki and S. Hasegawa, 2011. On the design and an implementation of broadband access management systems C and C product technologies development laboratories. NEC Corporation C and C Research Laboratories, NEC Corporation.

Nahum, E. and T.C. Wright, 2007. Evaluating SIP proxy server performance. Proceedings of the 17th International workshop on Network and Operating Systems Support for Digital Audio and Video, June 4-5, 2007, Urbana-Champaign, IL., USA.