The SIP Express Router

An Open Source SIP Platform

Y. Rebahi, D. Sisalem, J. Kuthan, A. Pelinescu-Oncicul, B. Iancu, J. Janak, D. C. Mierla

Fraunhofer Institut Fokus, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
{rebahi, sisalem, kuthan, Pelinscu-oncicul, iancu, janak, mierla}@fokus.fraunhofer.de

Abstract

The session initiation protocol (SIP) is constantly gaining in popularity and acceptance as the signaling protocol for next generation multimedia communication. This paper describes a scalable and reliable open source SIP platform called the SIP Express Router (SER). SER does not only support basic SIP features but also advanced features such as messaging and presence, translation between SIP and SMS or Jabber as well as full featured application programming interfaces. In this paper we will describe the architecture of SER, its different features and technical specifications.

I. Introduction

VoIP and multimedia communication services have been hyped as the next revolution in the Internet since the mid nineties. However, due to low communication quality, complicated and expensive technologies and vague advantages this promised revolution has been postponed over and over again. Only now, with the gradual move towards next generation networks and increased availability of high speed access networks VoIP and multimedia communication services are finally on the rise with various providers of different sizes offering their users VoIP services.

The session initiation protocol (SIP) is a signaling protocol that is often used to establish Voice over IP calls, advertise their presence status, send and receive instant messages, and maintain any kind of session including games and chats. A major benefit of SIP is, it creates an open framework for composing services out of multiple components. The same argumentation mentioned above regarding the speed of deployment of VoIP technology applies for the session initiation protocol (SIP) as well. While SIP has been hailed as the driving force behind the VoIP revolution and the basis for future multimedia communication, the deployment of SIP has been only very slow as well. While SIP was promising a low cost alternative to other well-established solutions such as intelligent telecommunication services on the one side and H.323 on the other, SIP-based products were up-to recently expensive and often immature. With the availability of lower cost and mature products on the one side and the increased interest in VoIP we are finally watching a rapidly increasing number of large deployments of SIP based service offers in various places in the world.

With available SIP implementation, including commercial and open source products either too expensive, immature, not-extendable or all of those together, research and development work to be conducted in the area of SIP-based services in next generation networks, such as the work conducted in the Evolute project, was close to impossible for research labs, universities and small enterprises.

With this background, the SIP Express Router was designed and realized to fill in the gap and provide the research community with a powerful toll for experimenting with SIP services and usage and for providers with a full-featured SIP infrastructure at open-source conditions.

In this paper we give a brief description of the architecture of SER in Sec.II. In Sec.III, we describe the different features and components supported by SER. The technical specification of SER is described in Sec.IV. Sec.V gives an overview of the usage of SER in the research community and Sec.VI summarizes the current usage of SER and next developments planned.

II. General Architecture

The SIP Express Router was designed in a highly modular manner as depicted in Figure 1, SER consists of a highly efficient core that is responsible for receiving, parsing SIP and forwarding messages.

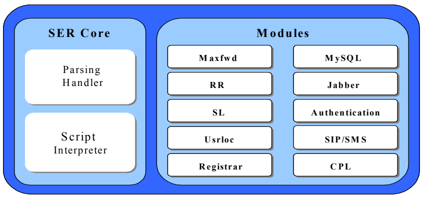

Figure 1: General Architecture of SER

The core is also responsible for invoking certain procedures that are provided as extension modules. These modules are dedicated to providing certain features. Such modules include:

- **Registration and user location:** This part is responsible for handling registration requests and managing the user location database.
- **Transaction management:** when acting in a statefull mode SER must maintain per transaction state, generate replies, match replies to requests and deal with forking.
- **User authentication:** SIP uses HTTP digest for authenticating user requests. This part of SER interacts with the database, which maintains the user's identity and password and is responsible for checking the credentials of the users.
- **SIP handling:** While the core deal with the message processing, the transaction management deal with state handling and taking the appropriate SIP actions, this type of modules deals with additional processing logic such as handling of record-route headers, loop detection or support for ENUM.

- **Application modules:** These modules provide some application level services such as SIP to Jabber translation for example.
- **Application programming modules:** To enable external programmers to use the features of functionalities of this class of modules provides a clear and simple to use interface that allows the separation between the SER and the application code.

Each module exports a set of functional procedures and utilizes procedures exported by other modules. The integration between the different modules and the core is realized through a configuration language. The SER configuration language (SCL) is a rich and flexible scripting language. With SCL the administrator of the SER platform can specify the actions to take when certain events occur. That is, with the rule specified in the figure below, the administrator can for example specify that after receiving a registration request, the core should invoke the authentication module before handing the registration to the registration module for example.

```
If(method==”REGISTER”)
www_challenge(“iptel.org”/*realm*/, “0”);
break;
```

III. Goals and Functional Architecture

To provide the research community as well as service providers with a flexible and powerful SIP platform the SIP Express Router was designed to fulfil the following goals:

- Offer service providers with a fully featured SIP infrastructure supporting powerful control interfaces allowing the utilization of SER under various usage scenarios and provider policies.
- Allow application programmers and research units to introduce new features and experiment with novel ideas by designing SER in a modular manner with open extension interfaces.
- Provide the users of the SIP technology with simple interfaces for indicating their own preferences and preferred services in a simple an intuitive manner.

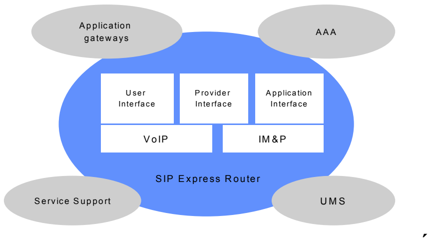

Figure 2: Conceptual model of the SIP Express Router

In this context and as depicted in Figure 2, SER was designed to offer different functional components ranging from basic SIP services up to application programming interfaces and unified messaging features.

A. Support of Basic SIP Services

The first task of any SIP implementation is to provide the basic SIP services as described in [1]. In conformance with the SIP standard [1], SER offers the basic functionalities of SIP proxy, redirect and registrar. To support messaging and presence, SER was extended to support the SIMPLE model [2]. Thereby, SER not only supports message routing but also presence management. In this context SER can act also as a message inbox, managing messages sent to a user currently offline and enabling the user to retrieve his messages after logging in.

B. Programming and Service Creation Interface

One of the biggest promises of SIP was to provide a unified basis for different communication modes such as audio, video, gaming and messaging and to ease the introduction of novel services. By supporting messaging and presence in addition to the routing and handling of SIP messages, SER already supports the first part of the SIP promise. To ease the introduction of novel services and the realization of intelligent services, SER further provides different extension and programming interfaces covering the needs of various user communities.

a. User service creation interfaces

To enable users to specify their own call preferences, SER supports the usage of CPL [3] and offers the users a simple graphical interface for creating their own intelligent call management logic.
The call processing language (CPL) is a tool that gives the user the possibility to implement his own call routing logic. It is meant to be simple, extensible, easily edited by graphical clients, and independent of operating system or signaling protocol. CPL is based on XML, which is a widely accepted industry standard.

A CPL service consists of a CPL server that interprets the CPL scripts, a CPL editor for composing the scripts, a database for managing the scripts and the signaling infrastructure (e.g., SIP proxy) that consults the CPL server about the actions to take after receiving a signalling message. SER integrates a CPL server as an extension module.

b. *Application Programmer Interface*

To enable application programmers to utilize the different features provided by SER in a language independent manner, SER provides a textual interface allowing external applications to communicate with the server called FIFO. This interface can be used to manipulate user contacts, send instant messages, watch server health, etc. As a result, SER can be easily integrated with a lot of existing applications even if they are SIP unaware. With the FIFO interface, the application programmer can implement his application in any language he wishes and simple execute various procedures provided by SER through this interface. In this context, the FIFO interface acts as an inter-process communication mechanism allowing the application to be run in a process independent from SER but still use the SER features.

As an example, the PHP based SER's web inteface – serweb- launches the FIFO interface when displaying and changing the user location information stored in the server's local store. Serweb also leverages the FIFO interface to send instant messages without any knowledge of the underlying SIP stack. Another application using the FIFO interface is the ser management utility command-line (serctl) which browses the server's in-memory user location database and display running processes as well as operational statistics.

c. *Extension Interfaces*

To enable the extension of the functionality of SER as well as experimenting with novel ideas concepts, SER provides an open programming interface that enables the user to utilize the current functionalities of SER and add novel ones.

This is achieved by implementing a new module that provides these new features. Such a new module would use the exported procedures of the other modules as well as functionalities of the core. The new module in its turn exports new procedures that could be invoked by the core and other modules as well.

As already mentioned in Sec.II, the SER scripting language (SCL) is then used for integrating the new functionalities in the general functionality of SER. Using SCL, the administrator of the SER platform can define the exact behavior of the platform based on available modules and needed actions. In this context SCL is similar to shell languages with support for regular expressions and allowing the administrator to define certain rules in the manner of

```
If (event) then (action);
```

Using SCL the administrator can also define arbitrary routing decisions, e.g.

```
If (method=="MESSAGE" && uri=~"sip:0179")
    forward(sms_gw);
```

Using an expression such as

```
if (method=="REGISTER") {
www_challenge(  "iptel.org" /* realm */,"0" /*  noqop -
- M$ can't deal with it */); break; };
```

the administrator can define for example that if a registration message is received then the authentication procedures of the authentication modules should be invoked before processing the registration.

C. Service Support

Besides the support for basic SIP features as defined in Sec.I.A, a SIP platform that is to be deployed in a large commercial environment needs to support additional features such as ENUM, traversal of NATs or IPv6.

In this context SER supports the following features:

- ENUM: To enable simpler representation of SIP addresses, ENUM [4] provides the possibility of using E.164 numbers for representing SIP users. In this case, SER is capable of communicating with ENUM servers and resolving the addresses in the appropriate manner.

- Firewall and NAT traversal: The basic NAT [6] operation is the translation and mapping of one set of IP addresses (usually private) to another (usually public). It was originally developed to hide private addresses from being seen on public networks. Unfortunately, this translation is not compatible with SIP and the mechanisms to handle NATs traversal have some limitations. The easiest way to establish SIP sessions through NATs is to use a STUN [7] enabled phone, however, the traversal is not guaranteed. For the case of user agents that do not support STUN, SER is capable of rewriting the contact and SDP [8] headers. That is, SER would rewrite the included private addresses with the public address included by the NAT in the IP header of the SIP message. Further, SER maintains this binding alive by periodically pinging the user agents.

- Support for IPv6: Besides IPv4, SER is one of the few available implementations that support IPv6 as well as the translation between IPv4 and IPv6 SIP messages, SER supports thereby a smooth transition between IPv4 and IPv6 networks.

Besides the support for additional features, SER provides the users and administrators with an intuitive web interface called SERWeb. Using SERWeb, the user can check any missed calls, look at the call logs, indicate further SIP aliases under which he is available and so on.

D. Authentication, Authorization and Accounting

To be able to support SIP services in a professional manner, service providers need an efficient infrastructure for authenticating users, authorizing the requested services and account their usage.

To comply with these needs SER supports:

- Authentication mechanisms as defined by [9]
- Authorization of users based on admission control lists. In this context the administrator can define different control lists for certain services such international. National and local calls for example. A user can only request and international call if he is included in that list.
- SER provides the needed mechanisms for collecting detailed information about the used services by a user and to save this information in a call detail record (CDR). Such CDRs list then the duration, destination and identity of the callers and callee in addition to other information.

To further comply with the varying needs of the wide scope of possible users, SER supports also different AAA databases and protocols. This includes simple text files and MySQL databases up to ISP level protocols such as RADIUS [10]. Work on DIAMETER [11] is on its way as well.

E. Application Gateways

The SIP Express Router allows a communication establishment even in a heterogeneous environment. This is realized by using translation gateways between SER and the other messaging systems.

- **SMS Gateway**: This provides a way of communication between SIP networks (via SIP MESSAGE) and GSM networks (via Short Message Service). Communication will be possible from SIP to SMS and also in the revert direction.
- **JABBER Gateway**: JABBER ([12]) is currently being discussed as an approach for providing translation between the heterogeneous messages and presence systems available in the market. This is achieved by translating the features of the different IM&P protocols into an XML-based form. With the SER JABBER gateway, SER translated between SIMPLE and Jabber IM (XMPP) ([13]). This gateway supports SIP/Jabber conference, sends received SIP Message messages to different IM systems (Jabber, ICQ, AIM, Yahoo) using Jabber server and the incoming instant messages as SIP Message messages. The Jabber gateway is also able to manage all kinds of Jabber messages (message/presence/iq) and can provide reliability, e.g the ability to detect if an IM gateway failed.

F. Unified Messaging Applications

As an example for some of the applications that might run on top of SER and provide users with value added features, the SER platform supports voicemail-2-mail services. Here unanswered calls are directed to a voicemail server that records the voice message. The recorded message is then sent as an Email-attachment to the user. This not only eases the retrieval of voice messages but enables the user to get a notification about the availability of new messages as soon as he is on-line.

IV. Technical Specification

The SIP Express Router is available as open source software under the GNU GPL licence [17]. It is implemented in conformance with [1] in ANSI-C and is ported and tested under all flavours of UNIX such as Linux, BSD and Solaris.
SER has been tested both in public by powering the SIP service offer of **www.iptel.org** as well as in various SIP interoperability tests.
Performance tests have shown that SER is capable of handling up to 5000 calls per second on a dual 1GHz processor with 250 Mbytes of memory.

V. SER in the European Research Community

One of the major development goals of SER was to provide the research community in general and the European in specific with a powerful and extendable open-source SIP platform. SER has been developed partially under the umbrella of different IST funded projects such as EVOLUTE and 6NET. In the context of the EVOLUTE project, SER was extended with messaging and AAA mechanisms namely support for RADIUS and HTTP digest. In the context of 6NET, SER was enhanced with IPv6 capabilities.
In the international level, SER is being deployed in the Internet2 ([16]) community as well as various universities such as Yale.
With the increased interest in SIP-based technologies, the usage of SER in the European research community will be further intensified in the upcoming projects.

VI. Summary and Next Steps

With its high level of configurability, open interfaces and scalability SER is turning into one of the most successful SIP implementations used in both academic and commercial environments. As most prominent deployments we mention the free world dialup service ([15]) and SIPPhone.com in the USA as well as Vozetele in Spain. SER is also providing the basis for ENUM trials in Germany, Austria and Spain. From the academic point of view, SER is being used in different universities and research labs for experimentation, teaching as well as service provisioning.
Based on the current implementation SER provides an ideal candidate for experimenting with 3GPP [14] technologies. Due to its configurability, SER can easily resume the different roles of the various service control functions defined in 3GPP. Additional required features such as enhanced routing capabilities, AKA based user authentication [18] or DIAMETER based AAA can be added to SER as further extension modules.

Remark: The work presented here was partially supported by the IST project EVOLUTE (seamlEss multimedia serVices Over alL IP-based infrastructures) under contract IST-2001-32449

Formatiert
Formatiert
Formatiert
Formatiert
Formatiert

VII. References

[1] J. Rosenberg, et al, "SIP: Session Initiation Protocol", RFC 3261, June 2002

[2] B. Campbell, et al, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002

[3] J. Lennox, et al, "Call Processing Language Framework and Requirements", RFC2824, May 2000

[4] P. Faltstrom, et al, "E.164 number and DNS", RFC 2916, September 2000

[5] K. Egevang, et al, "The IP Network Address Translator (NAT)", RFC 1631, May 1994

[6] K. Egevang, et al, "The IP Network Address Translator (NAT)", RFC 1631, May 1994

[7] J. Rosenberg, et al, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 20043

[8] M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998

[9] C. de Laat, et al, "Generic AAA Architecture", RFC 2903, August 2000

[10] C. Rigney, et al, "Remote Authentication Dial In User Service (RADIUS)", RFC 2138, April 1997

[11] P. Calhoun, et al, "Diameter Base Protocol", RFC 3588, September 2003

[12] P. Saint-Andre, J. Miller, "XMPP Instant Messaging", draft-ietf-xmpp-im-17

[13] D. Mierla, "SIMPLE-XMPP Interworking", draft-mierla-simple-xmpp-interworking-00

[14] 3rd Generation Partnership Project (3GPP), http://www.3gpp.org

[15] Free World Dialup, http://www.pulver.com/fwd/

[16] Internet2, http://www.internet2.edu/

[17] GNU General Public License, http://www.gnu.org/licenses/licenses.html

[18] A. Niemi, et al, "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)", RFC 3310, September 2002