A Generic Call State Model for SIP Proxy

Hristo Froloshki[1], Evelina Pencheva[2]

Abstract – **Provisioning of value-added services in IP telephone networks requires a generic model of the behavior of SIP server. Definitions of states and transitions in call processing will enable service logic triggering. There are services that deal with outgoing calls and services that deal with incoming calls, so the model distinguishes between processing of the two types of calls. Different traffic cases are considered to synthesize common behavior.**

Keywords – **value-added services in IP telephony networks, behavior of a SIP proxy**

I. Introduction

Today the telecommunications business is almost entirely determined by the services offered and their price. In order to survive in the struggle for peace of the telecom market, network operators need not only the right technology, but the right tools to create value-added services in efficient way. In circuit switched networks services like freephone, premium rate and televoting are implemented centrally on an intelligent network (IN) platform. In the Internet the network intelligence is completely distributed. There is a need for protocols, models and tools for implementing intelligent services in distributed packet switched environment.

A perspective candidate for delivering IP telephony is Session Initiation Protocol (SIP). SIP is an application protocol, utilizing client/server architecture. The protocol is used for multimedia connections in Internet, and also offers status and availability capabilities. In [1] and [2] it is shown how services we are accustomed to use in telephony networks can be implemented in the Internet by the use of SIP signaling. Some of the services are implemented in the end system, while others reside the network nodes (SIP servers). For services implemented in the network it is necessary to describe in a generic way the behavior of SIP server in service provisioning.

The IN conceptual model introduces the concept of half call: basic call state model is defined for originating calls (O_BCSM) and for terminating calls (T_BCSM). Both O_BCSM and T_BCSM can trigger service logic. Following the proved IN concept the paper suggests a generic model of the behavior of a SIP server. This model enables and facilitates additional logical processing in call setup, management and release procedures.

[1]Hristo Froloshki is with the Faculty of Telecommunications, Technical University – Sofia, blvd “Kliment Ohridsky” 8, 1000 Sofia, Bulgaria, hef@tu-sofia.bg
[2]Evelina Pencheva is with the Faculty of Telecommunications, Technical University – Sofia, blvd “Kliment Ohridsky” 8, 1000 Sofia, Bulgaria, enp@tu-sofia.bg

II. IN Basic call state models

In the IN conceptual model the O_BCSM and T_BCSM implement the idea of half-call as state transition diagrams where the points in call (PICs) represent the states through which each call passes.

The transitions are caused by events. An event can have a detection point (DP) associate with it. DPs are the points at which the call processing can be suspended in order to hand over control to service logic. The separation between the originating and termination models allows the definition of different supplementary services, applicable to the correspon-ding party. An example for service, utilizing O_BCSM is freephone, where service logic needs to be activated to translate the number dialed to a local one. An example for service activated by T_BCSM is call forwarding on no answer. In this case service logic should be activated when a call is received and the subscriber hasn’t answered it in a predefined time interval.

The idea of the half call concept is shown on Figure 1. The service switching function (SSF) supports both the O_BCSM and T_BCSM. Apart from this, the calling and the called party are not necessary connected to the same exchange or even to the same network. When the O_BCSM or T_BSCM encounters a DP this triggers the service control function (SCF).

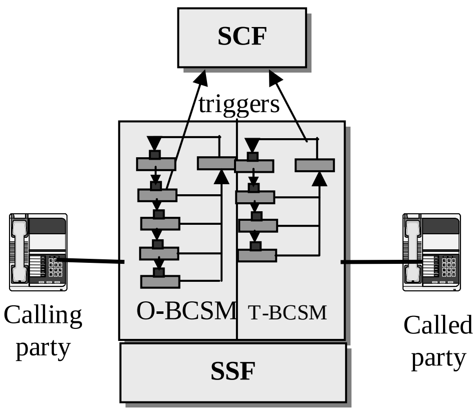

Figure 1 The O_BCSM and T_BCSM defined in the IN conceptual model

The SSF has a mechanism for handling DPs during call processing. There are different types of DPs that require specific actions to be taken. The trigger detection point (TDP) is set statically at the time of deployment of a service or when a user subscribes to it, while the event detection point (EDP) is set dynamically by service logic during the call. If the DP is armed as notification (TDP-N or EDP-N) the BCSM only informs the SCF and continues without waiting the response. If the DP is armed as request (TDP-R or EDP-R) the BCSM suspends call processing, triggers the SCF and waits for instructions on how to proceed.

III. MODELLING APPROACH

The SIP proxy is modeled with a finite state machine, following the proven modeling approach used for description of SSF behavior in the IN conceptual model. Two groups of different call scenarios (Figure 2 a,b,c,d,e,f), including SIP, PSTN[1] and H.323 networks, are considered. The first group consists of scenarios where a call originates from the SIP network and terminates in the same or an external network. The second group of scenarios presents a call terminating in the SIP network. Every SIP phone runs a user agent, consisting of user agent client (UAC), responsible for establishing outgoing calls and user agent server (UAS), which accepts incoming calls.

A SIP proxy server is an essential part of the SIP network, for it provides functionality needed for establishing calls. It processes SIP requests and can perform user location, address translation, security screening or any other processing on the SIP requests.

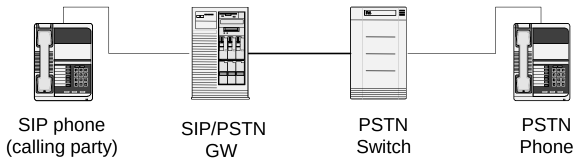

Figure 2 a SIP-to-PSTN scenario

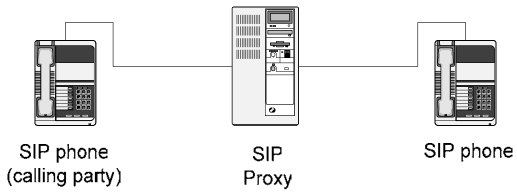

Figure 2 b SIP-to-SIP scenario

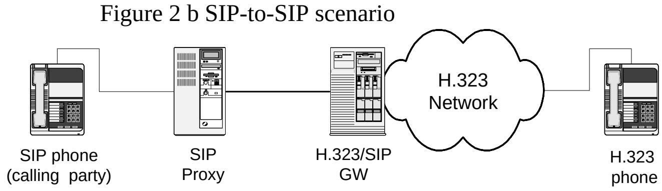

Figure 2 c SIP-to-H.323 scenario

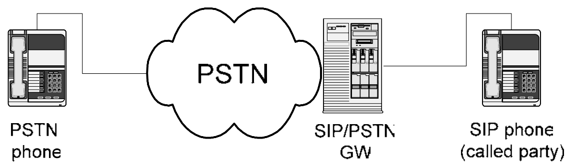

Figure 2 d PSTN-to-SIP scenario

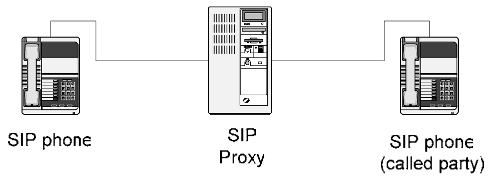

Figure 2 e SIP-to-SIP scenario

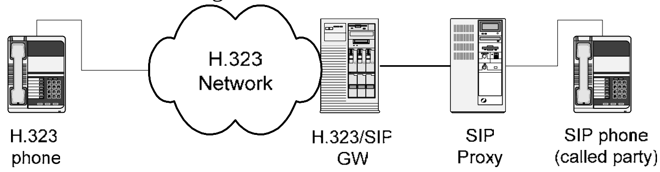

Figure 2 f H.323-to-SIP scenario

Actually, a SIP server can perform any of IN features, but the mechanism of control is different. In IN the SSF keeps tight control of the call state through BCSM. When the SCF interferes in the call processing the control is always returned to the SSF. In SIP there is no central entity that keeps tight control of call signalling. The request routed to a SIP proxy can be sent onward after processing or just transparently routed to another proxy. Nevertheless the behaviour of SIP proxy in call processing resembles the mechanism for service triggering in the BCSM. This server demonstrates similar behavior in the different scenarios and therefore the proposed state model is based on its states.

SIP signaling procedures for the above mentioned scenarios [3], [4] are considered. The communication between functional entities is broken down into elements and relations are found between the different traffic cases. The aim is to synthesize abstract generic models for SIP UAC and SIP UAS. Based on these models generic signaling flows between SIP UAC, SIP proxy and SIP UAS are developed. Having defined common signaling flows the behavior of SIP proxy is described by the use of a finite state machine.

Figure 3 shows a generic description of the behavior of a SIP UAC. The SDL[2] is used as a modeling tool.

In order to create a generic model of a SIP UAC common communication elements and signaling procedure parts are abstracted and put together. The model is described by the use of states and transitions. It starts with sending the *Invite* message to the called party. If authorization is not required, the transition to the next state is up to the type of the terminating network. The *Wait Progress* state is entered while checking the status of the called PSTN party. In the *Wait OK* state a positive response from the called party is anticipated. The *OK* message may be generated either by a SIP gateway, or by a SIP UAS. The *Alerting* state is where the calling party is notified that the *Invite* request has been received and alerting is taking place. The *RTP Session* state represents the active phase of the call. The *Authorizatio*n state is the point in session where the identity of the calling party is verified by proxy or registrar server.

The *Error Handling* state is used for processing of exceptions that might occur during call setup or release. Implementation details such as error identification codes, and their processing are not in the scope of this paper.

[1] Public Switched telephone Network

[2] Specification and Description Language

In the *Idle* state the UAS listens for incoming calls. After an *Invite* request is received, the server sends *Ringing* message, alerts the user (not shown on the figure) and sends *OK* response to the *Invite* request. The *Wait ACK* state is where final confirmation (*ACK*) for the establishing of media session is received.

The *RTP Session* state is where the actual conversation between parties takes place. Any party may terminate the communication by sending *BYE* message. An acknowledgment response for successful termination of the session is expected in *Wait OK* state.

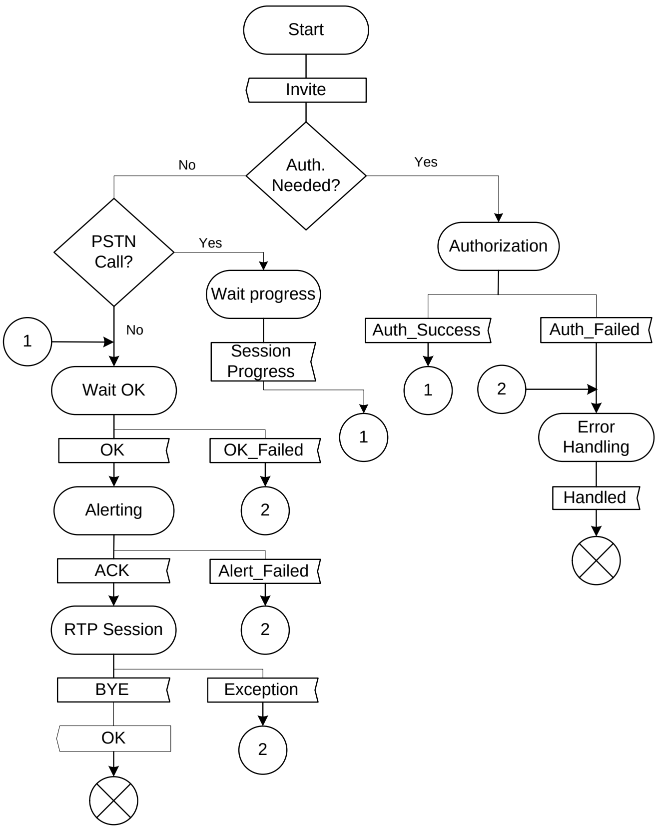

Figure 3 SDL description of a generic behavior of a SIP UAC

Figure 4 shows a generic description of the behavior of a SIP UAS.

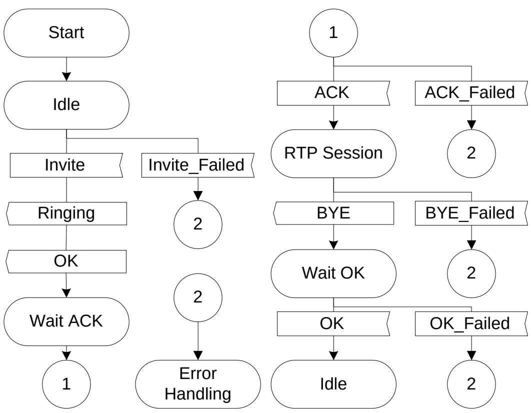

Figure 4 SDL description of a generic behavior of a SIP UAS

IV. SIP PROXY BEHAVIOUR MODEL

Based on the abstract models for SIP user agents generic signaling flows between user agents through SIP proxy servers are defined. The proposed message sequence in case of a stateful proxy is shown on Figure 5.

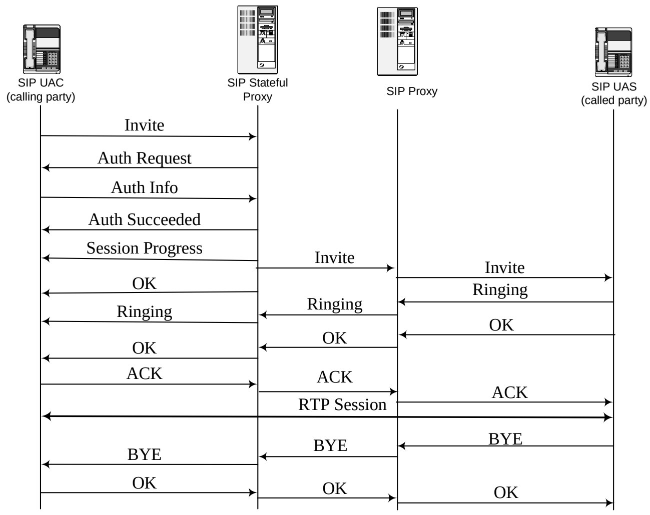

Figure 5 Generic signaling flows

The behavior of a SIP proxy can be described by states and transitions derived from the generic signaling flows. Following the half call concept in IN conceptual model two SIP proxy models are proposed: one for the SIP outgoing calls and one for the SIP incoming calls. There is a striking resemblance between the states and transitions in these models and the appropriate PICs and DPs in O_BSCM and T_BCSM.

Figure 6 presents the SIP proxy model handling outgoing calls. An *Invite* request is expected in the *O_Idle* state. The *Authorization* state performs identity verification procedures the aim is to acknowledge that the calling party is authorized to make a call. The *Address Analysis* state uses information received from the *Invite* message to identify destination address and additional parameters, needed for call setup. If the address information is complete, then the SIP proxy sends an *Invite* message to the called party. In case of SIP-PSTN call the *Session progress* message is sent to inform the calling party about the call progress and also the *OK* message acknowledges the establishment of an additional media channel. In the *Routing* state the receiving of a *Ringing* message indicates that the called party is alerted and then a *Ringing* message is sent to the calling party. An answer form the called party is expected in the *O_Alerting* state. The SIP proxy is informed when the called party answers with the *OK* message and it causes sending of the OK message to the calling party. The acknowledgment for established end-to-end session is received in the *O_Setup_Confirmation* state. In the *O_RTP session* state the actual data transfer takes place. Note that a SIP proxy server has no media capabilities, so the RTP stream passes transparently through the SIP proxy. The *O_Release* state is entered when a party ends the communication. The acknowledgement for call released causes the transition back to the *O_Idle* state.

All exceptions during the normal SIP outgoing call processing are processed in the *O_Error_Handling* state.

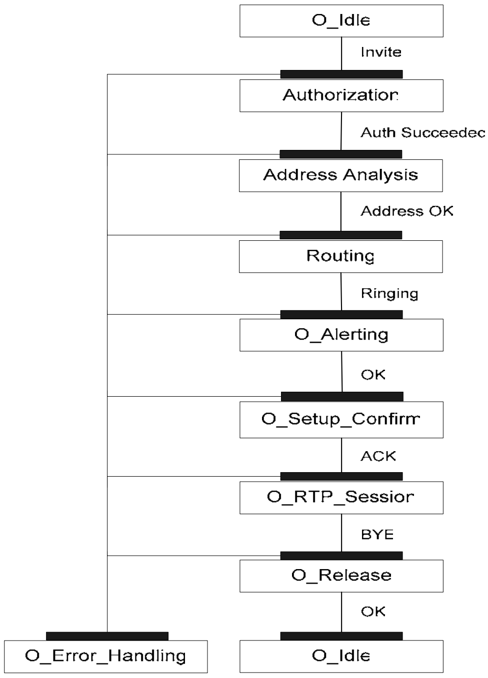

Figure 6 A generic stateful SIP proxy model handling outgoing calls

The generic behavior of SIP proxy server in case of SIP incoming call is shown in Figure 7.

The *Invite* request is received in the *T_Idle* state and this causes the transition to the *Present_call* state where the called party status is examined. If the called party is idle then a *Ringing* message is sent and the SIP proxy enters the *T_Alerting* state. The answer from the called party generates an *OK* message and a corresponding *OK* message is sent to the calling party. An acknowledgement for the established end-to-end connection is received in the *T_Setup_Confirmation* which causes the transition to *T_RTP_session* state. When a party hangs, the SIP server enters the *T_Release* state waiting for cancel acknowledgement.

All exceptions during the normal SIP incoming call processing are processed in the *T_Error_Handling* state.

The events causing transitions can have associate DPs. An event in an armed DP can trigger a service logic located on the same SIP server or on an external application server.

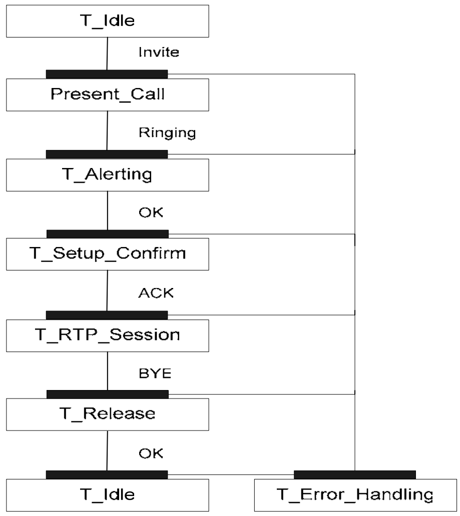

Figure 7 A generic SIP proxy model handling incoming

V. CONCLUSION

A generic model of a SIP proxy server is suggested. As it is in the IN conceptual model, the behavior of the SIP server is considered from two points of view: processing of outgoing calls and processing of incoming calls. The separation is required because there are services that deal with outgoing calls (for example originating call screening) and services that deal with incoming calls (blocking of malicious calls). The clear definition of the states and detection points enables service logic to interfere in the call processing.

REFERENCES

[1] Lennox I J., T Schulzrinne, F. La Porta, *Implementing Intelligent Network Services with the Session Initiation Protocol*, Tech-Report Number CUCS-002-99

[2] Pencheva E., V. Georgieva, *Modelling of Intelligence in Internet Telephone System*, ELECTRONICS'2004, Sozopol, Bulgaria Proceedings, book 3, pp 112-118.

[3] Alan B. Johnston, *Session Initiation Protocol*, Artech House Inc., 2004

[4] Rosenberg, J. et al., SIP: Session Initiation Ptorocol, RFC 3261, 2002

[5] Johan Zuidweg, *Next Generation Intelligent Networks*, Artech House Inc., 2002