Extending a 3GPP Prepaid Protocol to Improve Credit Pre-reservation Mechanism

Natal Vieira de Souza Neto,
Flávio de Oliveira Silva
and Pedro Frosi Rosa

Faculty of Computing
Federal University of Uberlândia
Uberlândia, MG, Brazil
Email: natal@mestrado.ufu.br,
flavio@facom.ufu.br,
pfrosi@ufu.br

Marcos Guimarães de Medeiros
and João Henrique de Souza Pereira

Innovation, Research and Development
Algar Telecom
Uberlândia, MG, Brazil
Email: marcosgm@algartelecom.com.br,
joaohs@algartelecom.com.br

***Abstract*—Online Charging System (OCS) uses credit pre-reservation to rating prepaid services. As the current standardized protocols present some problem during the pre-reservation phase, a new service must be introduced to address this kind of problem. In this paper, we propose a protocol for the communication between OCS modules. Running at the application layer, the protocol focuses on the definition of Refuse message, by regarding the credit reservation for call service, to be used when Rating Function module has some performance problem. We formalize the protocol messages and present its design rules and procedures. As results, the protocol was implemented into production in a carrier, and all calls were completed after the new feature.**

Keywords–Credit reservation; Telecommunication protocol; Online charging; Prepaid service.

I. Introduction

Online Charging System (OCS) [1] is a platform for charging, rating and controlling calls and other services like Short Message Service (SMS), Multimedia Message Service (MMS), General Packet Radio Service (GPRS) and Value-Added Service (VAS). In a good architecture for telecommunication enterprise, all prepaid services are charged by OCS. Prepaid service is the service for which the payment is made before the use [1].

In typical OCS, as recommended by 3rd Generation Partnership Project (3GPP) technical specifications, the module of call control must be separated from the module which processes the specified rules. The main reason is to avoid the overhead for the the processing module [1]. The module for call control is named Session Based Charging Function (SBCF) or Event Based Charging Function (EBCF). For the purpose of this work, we use the name Based Charging Function (BCF), because the most important here is the communication protocol and not the module development. The module for processing pricing and prepaid rules is named Rating Function. Call control with other application platforms can be built on BCF. Prepaid rules, pricing rules, Charging Detail Record (CDR) [2] records and database interaction can be built on Rating Function.

The communication between BCF and Rating Function must be made in real time, as the session is spread into two modules. A protocol is necessary for the communication to keep the call. This protocol is described in [3], but it does not address the Rating Function performance problem. Regarding the performance, a specific primitive is required, indicating to Rating Function that a problem happened and has been handled. This paper proposes an extension to the 3GPP protocol to address the issues described above, introducing the Refuse primitive, that besides the protocol specified here, could be used in other protocols for unit reservation, for instance.

A protocol that solves the problem of Rating Function and BCF communication is a protocol which ensures the session of the call on BCF and Rating Function. When a network element on telecommunication architecture, for instance Mobile Switching Center (MSC), communicates with OCS, the protocol used for mobile call is Camel Application Part (CAP), which initializes and finalizes a call. Newer architectures, for instance IP Multimedia Subsystem (IMS), use Diameter protocol and the communication is made by Diameter Credit Control (DCC). The CAP or DCC communicates with BCF, but the communication between BCF or Rating Function normally is made by Diameter [4], or proprietary protocols.

With a scale-out architecture using commodities hardware to build Rating Function, for instance, timeouts can occur when the call is initialized. The communication with database or other module made by Rating Function can be delayed. In this scenario, the primitive proposed in this paper ensures error control, which is not provided by 3GPP Prepaid Protocol specification.

Several protocols have been used to address this issue, but no one, as far as we know, has been successfully applied when subscriber's refund is necessary due to production environment. Based on it, a new service named Refuse is proposed. This paper focuses on call service, and not on GPRS, SMS, MMS or VAS.

The remaining of the paper is structured as follows: in Section II, we present related works; in Section III, we detail the credit pre-reservation mechanism; in Section IV, we describe our protocol specification and design, i.e., its environment, encoding, vocabulary, services and procedure rules; in Section V, we present the implementation details behind our approach; and finally, in Section VI, we make our final remarks.

 ISBN: 978-1-61208-360-5

II. Related Work

Credit pre-reservation is not a new ideia. According to 3GPP TS 32296 [1], an EBCF requests Rating Function for unit reservation. Unit reservation and credit pre-reservation are similar ideas.

When a call or other service is initialized, OCS works while the call is on. The performance is analyzed in others papers, like [5]. It means that OCS performance is a feature to be carefully analyzed. The Refuse primitive is necessary when the pre-reservation phase presents some problem.

Several services that are controlled by the OCS need credit pre-reservation. One example is Universal Mobile Telecommunications System (UMTS) prepaid service, exposed in [6]. In this scenario, the DCC protocol provides pre-reservation. To facilitate the understanding, some primitives in this paper are similar to those from DCC. We introduce the INIT_CALL_FIXED and INIT_CALL_MOBILE, similar to DCC INITIAL_REQUEST message. The DCC TERMINATE_REQUEST is similar to FINISH message described here. A telecommunication platform, like General Packet Radio Service (GGSN), can communicate with OCS using DCC. Into the OCS, BCF will communicate with Rating Function or other modules, and this communication uses the services INITIAL_REQUEST and TERMINATE_REQUEST. Then, BCF uses primitives presented here to keep this flow inside OCS. This paper introduces Refuse service to assure that the process will finish properly.

IMS aims to evolve network core from circuit-switching to packet-switching. IMS, standardized by 3GPP, becomes real in many networks. OCS is an element embedding in an IMS architecture. Communication in IMS often uses Diameter, and DCC, as described previously, has a flow for the credit reservation. The protocol proposed here is used inside OCS [7].

The 3GPP has standardized it by regarding flexibility and to provide all telecommunications functions over IP. OCS is an element embedding in an IMS architecture. Communication in IMS often uses Diameter, and DCC, as described previously, has a flow for the credit reservation. The protocol proposed here is used inside OCS [7].

A single module used for rating function was designed by Oumina and Ranc [8]. Rating Function integrates complex applications and allows the specification of quality of service parameters. Since it is a real time system, timeouts between Rating Function and BCF could happen.

III. Credit pre-reservation mechanism

The communication with OCS platform is made in telecommunications architecture based on real time systems. When a subscriber makes a call, the session request is sent to OCS, which returns complete or not complete. There are two types of terminals: fixed (wired) or mobile (wireless). The fixed architecture, for standard, uses Session Initiation Protocol (SIP) [9], and mobile architecture uses CAP protocol [10]. Furthermore, modern architectures use Diameter protocol. The basic difference between the three protocols is the way they establish the session. SIP sends the call session request to OCS, while CAP maintains the session on MSC, and only triggers OCS. The subscriber's money amount is controlled by OCS, but the session is controlled by MSC. The diameter uses DCC to trigger the OCS, and DCC controls the session [11][12].

In all cases described, the trigger or session come on BCF, which will control the session or just receive the trigger. If it receives a call attempt, it could make a numerical analysis, for instance, push an insertion network of number A or number B, or others [13]. Then, it sends the call informations to Rating Function, where the charging processing is made. When Rating Function receives a call attempt, it will analyze if the number can (or cannot) make the call, which is named Init. The matter here is the time. When MSC (or other telecommunication element) initializes a call, it needs the time that call will last. As the call is prepaid, the charging must be made at this moment. However, it is highly network consuming to trigger the Rating Function every second, thus Rating Function must decide the time granted for the call [3].

After the call, the BCF communicates to the Rating Function the duration of the call. If the duration of the call is less than the granted time, the Rating Function returns the difference. In the previous example, the time reserved was five minutes, supposing the call duration was two minutes, Rating Function returns three minutes to be refunded to the subscriber [8].

However, in some cases, the call is established and its duration is greater than the granted time. For instance, if Init granted five minutes, but the call took six minutes, the BCF will ask for more time. This is done through a service named Continue Call. Mandatorily, Init and Finish services are requested once, but Continue Call could be requested zero, one or more times [12]. The described mechanism is the credit pre-reservation.

IV. Protocol Specification and Design

For the protocol specified in [1], the pre-reservation mechanism will take place on BCF modules. In this design, the protocol will reserve a balance on Rating Function because performance evaluation shows that doing it on BCF module takes more time than doing it on Rating Function. As BCF should always complete the calls, independently of Rating Function time settings, all rules should be implemented on the Rating Function. Furthermore, the name of the proposed services also changes, because the communication flows are changed so that only the Rating Function will handle the balance.

The basic elements to specify a protocol are: the services provided by the protocol, the assumptions about the environment where the protocol will be executed, the message vocabulary used, message format and procedure rules that ensures consistence [14].

With the specification, it is possible to validate the protocol. This is done by creating the validation model, where the following features will be modeled: timings, flow control, message channel, process type, variables and data types, state execution, procedure model and recursion. After validation model, there are correctness requirements, which define the behavior, assertion, deadlocks, and so on. After the correctness requirements are defined, we are ready to design the protocol [14].

For the protocol proposed in this paper, the client is the BCF and the server is the Rating Function. It is a typical client-server relationship.

The assumptions about the environment start by considering that all of the actions (for instance credit reservation) should be completed before the call establishment. As the telecommunication industry is highly regulated, many of these actions should happen within a time interval.

As protocols are essentially event driven (services), it is important to specify what are the names of these services, in our case, there are nine services.

A. INIT_CALL_FIXED

this service is invoked when a call is started from a fixed terminal. BCF receives an indication to initialize a call, typically through SIP protocol. Then, BCF sends this service request to Rating Function, indicating that the call control is waiting for a response from the Rating Function whether this call can (positive) or cannot (negative) be completed.

B. INIT_CALL_MOBILE

this is similar to the previous service, but for a mobile terminal. It is necessary because there are different parameters which are sent when the terminal is mobile, like International Mobile Subscriber Identity (IMSI).

C. INIT_CALL_RESPONSE

upon INIT_CALL_FIXED or INIT_CALL_MOBILE indication, Rating Function processes all of the rules are applied to the requester terminal and returns to BCF the Call Id and the granted time, if BCF successfully completes the call. Otherwise, a negative answer is returned containing the error code.

D. CONTINUE_CALL

if the time granted is not enough to support the call, BCF can ask for additional time by requesting CONTINUE_CALL service informing the Call Id. Upon its receive, the Rating Function applies the charging rules and decides to grant, or not, a new time interval for the call.

E. CONTINUE_CALL_RESPONSE

upon receiving a CONTINUE_CALL, Rating Function processes the request, all of the rules are applied to the requesting Call Id, and, if successful, a new granted conversation period is returned to BCF for this Call Id. The call is ended if the granted period is zero.

F. FINISH_CALL

this service is invoked when the call is terminated by subscriber's terminal or the time is insufficient to keep the call.

G. FINISH_CALL_RESPONSE

it informs to BCF that the call has been correctly finished and, thus, its session removed and the balance updated. As it is a response, if timeout occurs, BCF generates a CDR ERROR, indicating the charging has been made incorrectly.

H. INIT_REFUSE_RESPONSE

this message has been introduced in this work to address performance. When BCF sends INIT_CALL_FIXED or INIT_CALL_MOBILE, it waits for INIT_CALL_RESPONSE. Depending on timing parameters configured in other telecommunications elements (like MSC), if there is a delay in the response, affecting other elements, it could disturb the Finite State Machine (FSM) in all of the telecommunication architecture elements. We implemented INIT_REFUSE_RESPONSE to address this issue. If timeout occurs between BCF and Rating Function, BCF completes the call (in an offline manner), independently of the Rating Function's response. However, due to the asynchronous behavior of the network, BCF could receive, after timeout, the previously expected INIT_CALL_RESPONSE. As BCF initialized an offline call, it sends to Rating Function an INIT_REFUSE_RESPONSE, instructing Rating Function to refund customer the pre-reserved credit.

I. FINISH_REFUSE_RESPONSE

this message has been introduced in this work to improve the call error control. When BCF sends FINISH_CALL, it waits for a FINISH_CALL_RESPONSE, indicating that Rating Function correctly updated the subscriber's credit. BCF sends to Rating Function a FINISH_CALL, and if the timeout occurs, BCF considers that Rating Function failed, and generates a CDR ERROR, ie, a CDR indicating that the call could not be finished properly, and it sends to Rating Function a FINISH_REFUSE_RESPONSE, informing the Rating Function that call may have had charging problems.

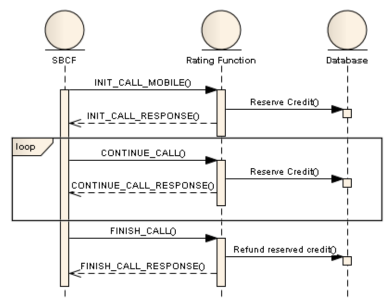

Figure 1. Call flow.

As shown in Fig. 1, BCF sends INIT_CALL_MOBILE (or INIT_CALL_FIXED), upon receiving a call request from other telecommunication elements. BCF is responsible for the call control for mobile calls, and maintains the session control for fixed calls. In fixed calls, BCF counts the duration, but for mobile calls, it is made by other elements, because CAP has parameters informing BCF of the duration of the call. BCF sends to Rating Function the Init, an Init for each call, independent the calling number. In modern platforms, a single calling number can make several calls at a time. Upon receiving INIT_CALL_MOBILE, the Rating Function creates a session with the rating information.

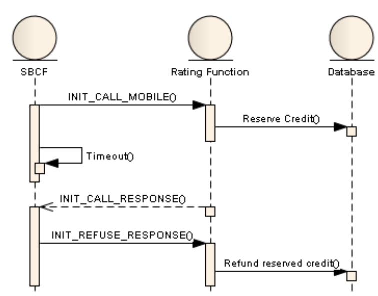

Figure 2. Call flow with Init Refuse.

In Fig. 1, upon receiving INIT_CALL_MOBILE, the Rating Function returns an INIT_CALL_RESPONSE which contains the duration for the call and, if needed, BCF could

 ISBN: 978-1-61208-360-5

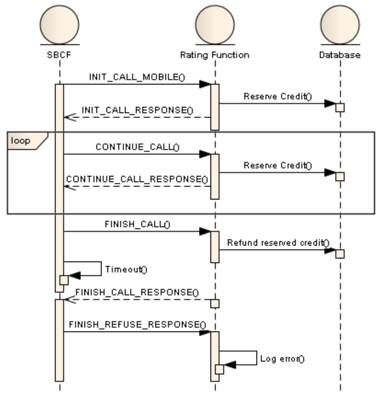

Figure 3. Finish Refuse service behavior.

ask for more time. If the duration is finishing, BCF requests a CONTINUE_CALL service for the particular call id. In architectures with modules installed in several machines, a CONTINUE_CALL must be sent to the specific Rating Function which received the INIT_CALL_MOBILE. Then, Rating Function returns more duration or not continues the call. Several CONTINUE_CALL messages can be issued, until the subscriber ends the call, or there is no more money in the account.

Fig. 1 also shows the FINISH_CALL service, where subscriber ends the call, or the subscriber has insufficient balance. BCF requests FINISH_CALL by indicating the call id and the call duration. With these pieces of information, the Rating Function can finish the call, crediting money (if the credit reservation was not fully used), and generating CDR.

When some problem happens at the beginning of the call, the flow is similar to Fig. 2, when BCF receives a request to initialize a call, and starts the call control. BCF sends INIT_CALL_MOBILE (or INIT_CALL_FIXED), as shown in Fig. 2, some problem happens by regarding Rating Function. BCF starts an offline call and the customer will not be charged. An offline call is released with unlimited duration and BCF will wait for another platform (like MSC) to finish the call. Even delayed, the Rating Function will reserve credit and answer to the BCF, and BCF will receive after the timeout. As BCF started an offline call, Rating Function response is not applicable. Upon this, BCF sends an INIT_REFUSE_RESPONSE telling Rating Function to refund subscriber the reserved credit. At the end of the call, BCF generates an offline CDR which will be eventually manually tariffed.

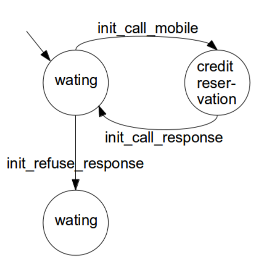

Figure 4. Rating Function Finite State Machine for Init Call service.

Fig. 3 represents the FINISH_REFUSE_RESPONSE which is used when a FINISH_CALL_RESPONSE delays or does not come to BCF. BCF informs the Rating Function that the call is facing a charging problem and the call is logged for future manual analysis. When FINISH_REFUSE_RESPONSE is used, BCF generates a CDR ERROR, for further analysis.

Now, we will define the services. In this paper, three services will be analyzed: Init, Continue and Finish. The Continue service does not have refuse message, as described previously.

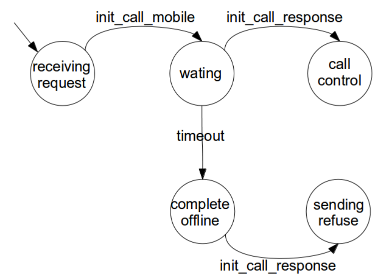

Figure 5. BCF Finite State Machine for Init service.

The Init service is shown in Fig. 4 which represents the FSM of Rating Function. Basically, Rating Function waits INIT_CALL_FIXED or INIT_CALL_MOBILE (as shown) to proceed with the credit reservation and responses with a INIT_CALL_RESPONSE message. If Rating Function receives an INIT_REFUSE_RESPONSE, it must refund the pre-reserved money. Fig. 5 represents Init service at BCF. In the first state, BCF is waiting a request from customer terminal (Calling User). Upon receiving it, it initializes the call control, sends to Rating Function a INIT_CALL_MOBILE and waits for a response. If it receives the INIT_CALL_RESPONSE primitive, the call is completed and becomes under control. However, if timeout occurs, BCF completes the call in offline mode, and if INIT_CALL_RESPONSE comes after timeout, the INIT_REFUSE_RESPONSE primitive is sent to Rating Function.

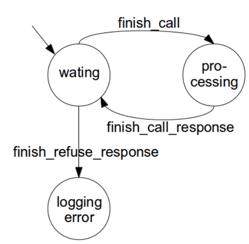

Figure 6. Rating Function Finite State Machine for finish service.

The Finish service at Rating Function is represented in Fig. 6. First, the Rating Function is waiting for the end of the call. When the call ends, the Rating Function receives a FINISH_CALL primitive and processes it (updating subscriber's accounts and generating CDR). However, if Rating Function receives a FINISH_REFUSE_RESPONSE, it logs the information, as it is impossible to know if the call was correctly or incorrectly charged.

Fig. 7 shows the FSM at BCF, where if a timeout on finish

 ISBN: 978-1-61208-360-5

service happens, a FINISH_REFUSE_RESPONSE is sent to Rating Function. A CDR ERROR is generated which will be manually treated later.

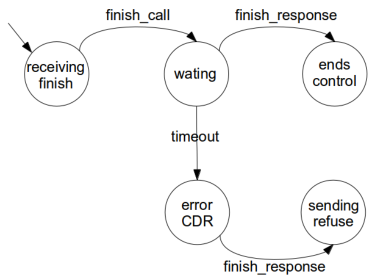

Figure 7. BCF Finite State Machine for Finish service.

V. Implementation Details

The first development of the protocol, to build BCF Module, was made using Java language and JAIN SLEE Mobicents [15]. The Rating Function Module also was built in Java, by using Spring Framework [16]. The main reasons for this are due to the fact that Mobicents has an architecture designed to create, deploy and manage services and applications integrating voice, and other services. Mobicents has been largely used for telecommunication companies around the world. Spring allows several programming techniques, which is helpful because the Rating Function has several rules. All rules were tested and deployed in a real Brazilian telecommunications company.

The formatting for the protocol communication is made by using Protocol Buffers (protobuf). It is a well tested framework for the encoding of structured data [17]. The primitives were packed/unpacked using message concept of protobuf which allows the focus on the design and implementation of the protocol. Protobuf converts the defined messages to Java classes which can be used on BCF (with mobicents) or Rating Function (with Spring). The messages contain typed fields. Each message will be described below.

A. INIT_CALL_FIXED message fields

1) Call Id: this parameter is a string that contains the session's Id.

2) Insertion_Network: this paramenter is a string and for fixed calls, Next Generation Networks (NGN) could insert a prefix in calls, to distinguish some services. It may contain digits, characters and special symbols.

3) Original_Number_A: this parameter is a string and it contains the original calling number. It may contain digits, characters and special symbols.

4) Tariff_Number_A: this parameter has the same type as Original_Number_A which is modified by BCF before sending it to Rating Function.

5) Original_Number_B: this parameter is a string and it contains the original called number. It may contain digits, characters and special symbols.

6) Tariff_Number_B: this parameter has the same type as Original_Number_B which is modified by BCF before sending it to Rating Function.

7) CSP: this paramenter is a string and it contains the Carrier Service Prefix, used to make long-distance calls. It may contain digits, characters and special symbols.

8) Flow: this paramenter is integer, and specifies the call flow, indicating if the BCF received a calling or a called number.

9) Call_Type: a string, indicating if the call is local (same region) or not (other region).

10)Flag_Portability_Number: this parameter is a boolean and it indicates if the calling or called number is a ported number (originally from other operators).

11)CNL_A: a string, indicates the locality of calling number. The price can be determined based on the distance between the calling number and called number, and this parameter is used for this.

12)CNL_B: a string, indicates the locality of called number, also used to determinate the distance between calling and called numbers.

13)Portability_Prefix: a string, indicating portability of the calling/called number. When a subscriber migrates of operator, but maintains the number, this parameter is needed to define to which carrier the number belongs to.

B. INIT_CALL_MOBILE message fields

This service has the same INIT_CALL_FIXED message fields, except CNL_A and CNL_B needed only by fixed number. Besides INIT_CALL_FIXED fields, INIT_CALL_MOBILE has the following exclusive fields:

1) Is_Video_Call: this parameter is a boolean and indicates if the call is a video call. CAP provides this information. In case of other protocols that BCF receives, the information does not exist. Therefore, is a optional field, and the default value is false.

2) IMSI: a string, indicating the IMSI. If the flow is calling, it is the calling number IMSI, if the flow is called, it is the called number IMSI.

3) Cell_Global_Id: a string, indicating the Cell Id. A cell is the sector of the Base Transceiver Station (BTS) or NodeB where the mobile is registered.

With the messages described above, Rating Function can handle and charge the call.

C. INIT_CALL_RESPONSE message fields

The INIT_CALL_RESPONSE is the same for both fixed and mobile calls. It is described below:

1) Call Id: a string, indicating the id, same id received by Rating Function on INIT_CALL_MOBILE or INIT_CALL_FIXED. It is necessary for the BCF to identify for which call it is receiving response.

2) Action: a string, it indicates to BCF if the call must be completed (or not), or if a ring tone must be played before answer.

3) Quantity: a long value, it is the duration released, ie, how long BCF must allow the call.

4) account: a vector of numerical types, with the accounts. Some money was reserved on these accounts. This field is important for BCF to maintain information about charging, for a possible refuse.

D. INIT_REFUSE_RESPONSE message fields

When necessary, BCF sends a Init Refuse service. The following fields are used:

1) Call Id: a string, indicating the id. It is necessary for the Rating Function to know which call was refused.

2) Action: a string, indicating the action received on INIT_CALL_RESPONSE. If the call was not completed, it will not be in Rating Function session.

3) Account: a vector of long, the account's vector received on INIT_CALL_RESPONSE. It is important because refuse indicates a problem in Rating Function, and maybe Rating Function has no call information. In this case, it returns balance for accounts on this vector.

E. CONTINUE_CALL message fields

Continue Call service, implemented with the CONTINUE_CALL message. It has the following fields:

1) Call Id: a string, the same id used on Init call. It is necessary for the Rating Function to know which call is requesting more time.

F. CONTINUE_CALL_RESPONSE message fields

The CONTINUE_CALL_RESPONSE message has only new quantity released and the accounts vector.

G. FINISH_CALL message fields

The Finish Call service is made with the FINISH_CALL message. This primitive has the following fields:

1) Call Id: a string, indicating the call id. It is necessary for the Rating Function to know which call has finished.

2) Start_Time: a date and time, indicating the start time of the call, this time is stored by BCF, and sent to Rating Function only on FINISH.

3) Total_Duration: a long value, in seconds, indicating the total duration, considering wait time of subscriber's equipment and the call duration.

4) Call_Duration: a long value, in seconds, indicating the total duration without the wait time.

5) Call_Charge: this is a boolean field and indicates if it is a charged (or not charged) call.

6) Finish_Cause: a string. Same protocols inform BCF of the finish cause, and BCF must send this information to the Rating Function. The Rating Function can use this information in CDR or in a specific rule.

H. FINISH_CALL_RESPONSE message fields

The FINISH_CALL_RESPONSE message has two fields: call id; and processed, a boolean field indicating if Rating Function finished correctly.

I. FINISH_REFUSE_RESPONSE message fields

The FINISH_REFUSE_RESPONSE message can be implemented with one unique field, the call id, indicating to Rating Function that a refuse happened.

All fields described above were used on a real development of an OCS platform. Depending on the rules used by OCS, new fields need to be added.

In tests in a real production environment, the rate of refused messages on the network is approximately 100 to 600 calls daily, in a universe of 4 million of calls daily. It is a small number, but refuse services are necessary so that the subscriber is not harmed.

VI. CONCLUDING REMARKS AND FUTURE WORK

In this work, a communication protocol between ECF (or SBCF) and Rating Function modules has been designed in OCS platforms. This protocol has introduced the Refuse services on the initiation/end of calls. The Refuse services can be used when the Rating Function delays answers or has problems, including being out of service. With these, the subscriber has its call established, without problem. The protocol was implemented and used, with heavy load tests.

We used Rating Function to reserve and to debit balance. The main reason for this was because the protocol proposed is similar to 3GPP protocol, but our design considers that the calls will be always completed. Then, SBCF or EBCF only make call control, becoming lightweight to control the call.

The result is a platform currently in use, which charges 900,000 subscribers, considering both fixed and mobile terminals, and controls Rating Function timeout, allowing calls to always be completed.

For future works, it is possible to see the same features provided through Refuse services to other platforms like GPRS, SMS, MMS and VAS. These services have different flows, and refuse messages must be designed considering other parameters and services.

REFERENCES

[1] 3GPP, "Online Charging System (OCS): Applications and interfaces," 3rd Generation Partnership Project (3GPP), TS 32.296, Sep. 2011.

[2] ——, "Charging Data Record (CDR) parameter description," 3rd Generation Partnership Project (3GPP), TS 32.298, Dec. 2013.

[3] ——, "Charging architecture and principles," 3rd Generation Partnership Project (3GPP), TS 32.240, Dec. 2013.

[4] ——, "Diameter charging applications," 3rd Generation Partnership Project (3GPP), TS 32.299, Dec. 2013.

[5] T. Grgic and M. Matijasevic, "Performance metrics for context-based charging in 3gpp online charging system," in Telecommunications (ConTEL), 2013 12th International Conference on, June 2013, pp. 171–178.

[6] H.-Y. Lee and Y.-B. Lin, "Credit pre-reservation mechanism for umts prepaid service," Wireless Communications, IEEE Transactions on, vol. 9, no. 6, June 2010, pp. 1867–1873.

[7] H. Oumina and D. Ranc, "Specification of rating function of online charging system in 3gpp ip multimedia system (ims) environment," in New Technologies, Mobility and Security, 2008. NTMS '08., Nov 2008, pp. 1–5.

[8] ——, "Designing the rating function of 3gpp online charging system for ip multimedia subsystem," in Internet Multimedia Services Architecture and Applications, 2008. IMSAA 2008. 2nd International Conference on, Dec 2008, pp. 1–6.

[9] J. Rosenberg et al. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard). Internet Engineering Task Force. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141, 6665, 6878. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt [retrieved: Jun., 2002]

[10] 3GPP, "CAMEL Application Part (CAP) specification," 3rd Generation Partnership Project (3GPP), TS 29.078, Dec. 2012.

[11] ——, "Advice of Charge (AoC) service," 3rd Generation Partnership Project (3GPP), TS 32.280, Dec. 2013.

[12] J. Bhosale and P. Pawar, "Credit pre-reservation mechanism for mobile prepaid service," International Journal of Innovations in Engineering and Technology, vol. 9, no. 6, Dec 2013, pp. 82–87.

[13] 3GPP, "IP Multimedia Subsystem (IMS) charging," 3rd Generation Partnership Project (3GPP), TS 32.260, Dec. 2013.

[14] G. J. Holzmann, Design and Validation of Computer Protocols. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1991.

[15] M. Femminella, R. Francescangeli, E. Maccherani, and L. Monacelli, "Implementation and performance analysis of advanced it services based on open source jain slee," in Proceedings of the 2011 IEEE 36th Conference on Local Computer Networks, ser. LCN '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 746–753.

[16] Spring. Springframework reference manual 3.1. [Online]. Available: http://spring.io/docs [retrieved: May, 2014]

[17] K. Varda. Protocol buffers. http://code.google.com/apis/protocolbuffers/. [Online]. Available: http://code.google.com/apis/protocolbuffers/ [retrieved: May, 2014]

 ISBN: 978-1-61208-360-5